# SKIN CANCER DETECTION USING DEEP LEARNING

**A PROJECT REPORT**

***Submitted by***


**LOKESHWAR. K (511319104044)**

**MD ABUBAKER. O (511319104048)**


***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

**in**

**COMPUTER SCIENCE ENGINEERING**


**KINGSTON ENGINEERING COLLEGE, VELLORE - 632059**

**ANNA UNIVERSITY: CHENNAI 600 025**


**MAY 2023**

# ANNA UNIVERSITY: CHENNAI 600 025

# BONAFIDE CERTIFICATE

Certified that this project report **"SKIN CANCER DETECTION USING DEEP LEARNING"** is the Bonafide work of **"LOKESHWAR. K (511319104044)** and **MD ABUBAKER. O (511319104048)"** who carried out the project work under my supervision, during the academic year **2022-2023.**

| | |
|---|---|
| **SIGNATURE** | **SIGNATURE** |
| **Dr. U.V. ARIVAZHAGU, M.E., Ph.D.,** | **Ms. M. ANITHA, M.E., Ph.D.,** |
| **HEAD OF THE DEPARTMENT** | **SUPERVISOR,** |
| Department of Computer Science | **ASSISTANT PROFESSOR** |
| and Engineering | Department of Computer |
| Kingston Engineering College | Science and Engineering |
| Vellore – 632 059 | Kingston Engineering College |
| | Vellore – 632 059 |

Submitted for The University Project Viva Voce Examination Held on

___________________

**INTERNAL EXAMINER** **EXTERNAL EXAMINER**

## ACKNOWLEDGEMENT

First and foremost, we are extremely grateful to the almighty for giving the strength and support to complete this project successfully.

We are extremely happy to express our gratitude in thanking the beloved honorable chairman **Mr. D.M. KATHIR ANAND, M.B.A., (U.S.A)** who has been a pillar of strength to the college.

We would like to express our immense gratitude and heartfelt thanks to the beloved Principal as well as Head of the Department **Dr. U.V. ARIVAZHAGU, M.E., Ph.D.,** for making the resources available at right time and providing valuable insights leading to the successful completion of our project.

We express thanks to our project coordinator **Dr. S. PRIYA M.E., Ph.D.,** Assistant Professor, Department of Computer Science and Engineering, for her advice and guidance to complete our project work. We thank all our faculty members who have helped us in the completion of this project. With immense pleasure we regard our deep sense of indebtedness and gratitude to our project supervisor **Ms. M. ANITHA M.E., Ph.D.,** Assistant Professor, Department of Computer Science and Engineering, who was not only a source of inspiration but also motivating us with her guidance and continuous support in the execution of the project.

To one and all, we own acknowledgement who directly and indirectly aided us in completing the project.

**LOKESHWAR. K**

**MD ABUBAKER. O**

# ABSTRACT

Cancer is a deadly disease that arises due to the growth of uncontrollable body cells. Every year many people succumb to cancer, and it's been labeled as the most serious public health snag. Cancer can develop in any part of the human anatomy, which may consist of trillions of cellules. One of the most frequent types of cancer is skin cancer which develops in the upper layer of the skin. Previously, machine learning techniques have been used for skin cancer detection using protein sequences and different kinds of imaging modalities. The drawback of the machine learning approaches is that they require human engineered features, which is a very laborious and time-taking activity. Deep learning addressed this issue to some extent by providing the facility of automatic feature extraction.

Convolutional Neural Network (CNN) or ConvNet, are a class of deep neural networks, basically generalized versions of multi-layer perceptron's. CNNs have given highest accuracy in visual imaging tasks. This project aims to develop a skin cancer detection CNN model which can classify the skin cancer types and help in early detection. The CNN classification model will be developed in Python using Keras and Tensorflow in the backend. The model is developed and tested with different network architectures by varying the type of layers used to train the network including but not limited to Convolutional layers, Dropout layers, Pooling layers and Dense layers. The model will also make use of Transfer Learning techniques for early convergence. The model will be tested and trained on the dataset collected from the International Skin Imaging Collaboration (ISIC) challenge archives.

# TABLE OF CONTENT

# LIST OF FIGURES

# LIST OF TABLES

# CHAPTER-1

## INTRODUCTION

Skin cancer is one of the most active types of cancer in the present decade. As the skin is the body's largest organ, the point of considering skin cancer as the most common type of cancer among humans is understandable It is generally classified into two major categories: melanoma and nonmelanoma skin cancer. Melanoma is a hazardous, rare, and deadly type of skin cancer. According to statistics from the American Cancer Society, melanoma skin cancer cases are only 1% of total cases, but they result in a higher death rate. Melanoma develops in cells called melanocytes. It starts when healthy melanocytes begin to grow out of control, creating a cancerous tumor. It can affect any area of the human body. It usually appears on the areas exposed to sun rays, such as on the hands, face, neck, lips, etc.

Melanoma type of cancers can only be cured if diagnosed early; otherwise, they spread to other body parts and lead to the victim's painful death. There as various types of melanoma skin cancer such as nodular melanoma, superficial spreading melanoma, acral lentiginous, and lentigo malign. Therefore, the critical factor in skin cancer treatment is early diagnosis. Doctors ordinarily use the biopsy method for skin cancer detection. This procedure removes a sample from a suspected skin lesion for medical examination to determine whether it is cancerous or not. This process is painful, slow, and time-consuming. Computer-based technology provides a comfortable, less expensive, and speedy diagnosis of skin cancer symptoms. To examine the skin cancer symptoms, whether they represent melanoma or nonmelanoma, multiple techniques, noninvasive in nature, are proposed. The general procedure followed in skin cancer detection is acquiring the image, preprocessing, segmenting the acquired preprocessed image, extracting the desired feature, and classifying it.

## 1.1 LITERATURE SURVEY:

There has been related works done in the past to detect skin diseases using machine learning and deep learning.

**[1] Title:** Skin Cancer Detection Using Combined Decision of Deep Learners

**Author:** Azhar Imran, Arslan Nasir, and Muhammad Bilal [IEEE 2022]

In this study, Azhar Imran, Arslan Nasir, and Muhammad Bilal explore the use of convolution-based deep neural networks for skin cancer detection, leveraging the ISIC public dataset. Detecting cancer accurately and timely is crucial, as errors can have serious consequences. While individual machine learning models have limitations in detecting cancer, combining the decisions of multiple learners is expected to improve accuracy. Ensemble learning techniques capitalize on the diversity of learners to achieve better decision-making.

**[2] Title:** Region-of-Interest Based Transfer Learning Assisted Framework for Skin Cancer Detection

**Author:** Ashraf R., Afzal S., Rehman A.U., Gul S., Baber J., Bakhtyar M., Mehmood I., Song O.Y., and Maqsood M. [IEEE 2020]

Ashraf R., Afzal S., Rehman A.U., Gul S., Baber J., Bakhtyar M., Mehmood I., Song O.Y., and Maqsood M. proposed a region-of-interest based transfer learning assisted framework for skin cancer detection. By leveraging transfer learning and focusing on informative ROIs, their approach achieves improved performance compared to existing methods. This research contributes to the advancement of skin cancer detection and paves the way for further exploration and refinement of transfer learning techniques in medical image analysis.

**[3] Title:** Automatic Skin Cancer Detection in Dermoscopy Images Based on Ensemble Lightweight Deep Learning Network

**Author:** Lisheng Wei, Kun Ding, and Huosheng Hu [IEEE 2020]

The author proposed a lightweight skin cancer detection framework for dermoscopy images. By utilizing lightweight CNNs and incorporating feature discrimination, their approach achieves improved recognition performance while maintaining a small model size. Additionally, the authors developed a lightweight semantic segmentation model for lesion area segmentation. The proposed method demonstrates superior performance compared to existing deep learning techniques for skin cancer detection. This research contributes to the field of automatic skin cancer detection and provides a foundation for the development of efficient and accurate algorithms in dermoscopy image analysis.

**[4] Title:** Automatic Skin Cancer Images Classification

**Author:** Mahmoud Elgamal [IJACSA 2013]

Elgamal presents two hybrid techniques for automatic skin cancer image classification. The techniques involve feature extraction using discrete wavelet transformation, dimensionality reduction using PCA, and classification using supervised machine learning algorithms (ANN and k-NN). The proposed hybrid techniques demonstrate robustness and effectiveness in distinguishing normal and abnormal skin cancer images, with high classification success rates. Early detection and accurate classification of skin cancer are essential for improving patient outcomes, and these techniques contribute to the advancement of automatic skin cancer detection systems.

**[5] Title:** Classification of Melanoma and Nevus in Digital Images for Diagnosis of Skin Cancer

**Author:** Khan M.Q., Hussain A., Rehman S.U., Khan U., Maqsood M., Mehmood K., and Khan M.A., [IEEE 2019]

Author presents an intelligent system for the classification of melanoma and nevus in digital images for skin cancer diagnosis. The proposed methodology utilizes image processing techniques, including preprocessing, segmentation, and feature extraction, followed by SVM-based classification. The results demonstrate high accuracy and effectiveness in distinguishing melanoma from nevus lesions. This research contributes to the field of skin cancer diagnosis and can assist healthcare professionals in improving early detection and diagnosis of melanoma, thereby potentially saving lives.

## 1.2 EXISTING SYSTEM:

Frequent use of biopsies is also not encouraged by dermatologists. According to International Skin Imaging Collaboration, the number of unnecessary culture tests which are begin performed vastly varies depending upon various parameters which includes clinical setup, expertise of dermatologist, and the technology applied. Computer procedures and advancements in machine learning not only aid the dermatologists in early detection of melanoma but also avoid heavy expenses of melanoma detection and unnecessary biopsies.

**DISADVANTAGES**:

- More expensive method to detect melanoma.
- There have been more challenges during the design of classification approaches.

## 1.3 OBJECTIVE AND SCOPE:

The cardinal objective of this project is to develop state of the art Convolutional Neural Network (CNN) model to perform the classification of skin lesion images into respective cancer types. The model is trained and tested on the dataset made available through the International Skin Imaging Collaboration (ISIC). The model can be used for analyzing the lesion image and finding out if it's dangerous at an early stage.

## 1.4 RESEARCH METHODOLOGY:

The purpose of performing this systematic literature review was to select and categorize the best available approaches to skin cancer detection using neural networks (NNs). Systematic literature reviews collect and analyze existing studies according to predefined evaluation criteria. Such reviews help to determine what is already known in the concerned domain of study.

All data collected from primary sources are organized and analyzed. Once systematic literature is completed, it provides a more sensible, logical, and robust answer to the underlying question of the research.

## 1.5 REROURCE OF SEARCH:

We conducted our initial search on well-reputed search engines such as IEEE Xplore, ACM, Springer as well as to extract information relevant to NN techniques for skin cancer detection. Basic research material related to the underlying topic was filtered out in the primary search. The selected research papers and conference proceedings were further analyzed according to evaluation criteria.

# CHAPTER-2

## DEEP LEARNING TECHNIQUES FOR SKIN CANCER DETECTION

Deep neural networks play a significant role in skin cancer detection. They consist of a set of interconnected nodes. Their structure is like the human brain in terms of neuronal interconnectedness. Their nodes work cooperatively to solve problems. Neural networks are trained for certain tasks; subsequently, the networks work as experts in the domains in which they were trained. In our study, neural networks were trained to classify images and to distinguish between various types of skin cancer. Different types of skin lesion from the International Skin Imaging Collaboration (ISIC) dataset are presented. We searched for different techniques of learning, such as ANN, CNN, KNN, and GAN for skin cancer detection systems. Research related to each of these deep neural networks is discussed in detail in this section.

### 2.1 ARTIFICIAL NEURAL NETWORK (ANN):

An Artificial neural network is a nonlinear and statistical prediction method. Its structure is borrowed from the biological structure of the human brain. An ANN consists of three layers of neurons. The first layer is known as the input layer; these input neurons transfer data to the second/intermediate layer of neurons. The intermediate layers are referred to as hidden layers. In a typical ANN, there can be several hidden layers. Intermediate neurons send data to the third layer of output neurons. Computations are learned at each layer using backpropagation, which is used for learning the complex associations/relationships between input and output layers. It is like a neural network. Currently, in computer science, the term neural network and artificial neural network are used interchangeably.

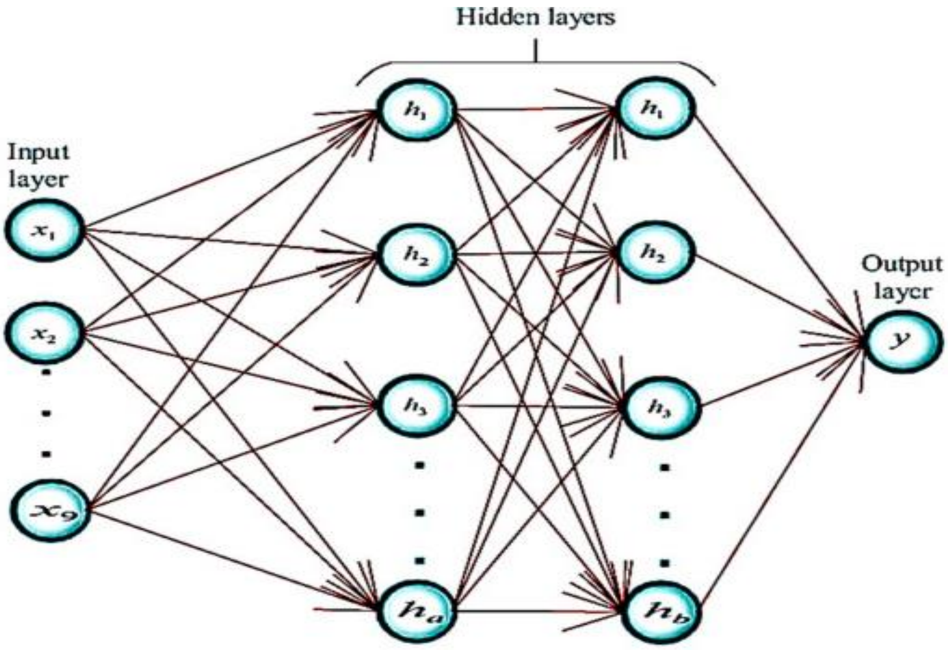


Fig.1 Basic ANN structure.

ANN is used for the classification of extracted features in skin cancer detection systems. Input images are classified as melanoma or nonmelanoma after successful training/classification of the training set. The input/first layer of the ANN process connects with the hidden layer by the input dataset. The dataset can be labeled or unlabeled, which can be processed accordingly using a supervised or unsupervised learning mechanism.

## 2.2 CONVOLUTIONAL NEURAL NETWORK (CNN):

A convolution neural network is an essential type of deep neural network, which is effectively being used in computer vision. It is used for classifying images, assembling a group of input images, and performing image recognition. CNN is a fantastic tool for collecting and learning global data as well as local data by gathering more straightforward features such as curves and edges to produce complex features such as shapes and corners. CNN’s hidden layers consist of convolution layers, nonlinear pooling layers, and fully connected layers. CNN can contain multiple convolution layers that are followed by several fully connected layers. Three major types of layers involved in making CNN are convolution layers, pooling layers, and full-connected layers. The basic architecture of CNN is presented.

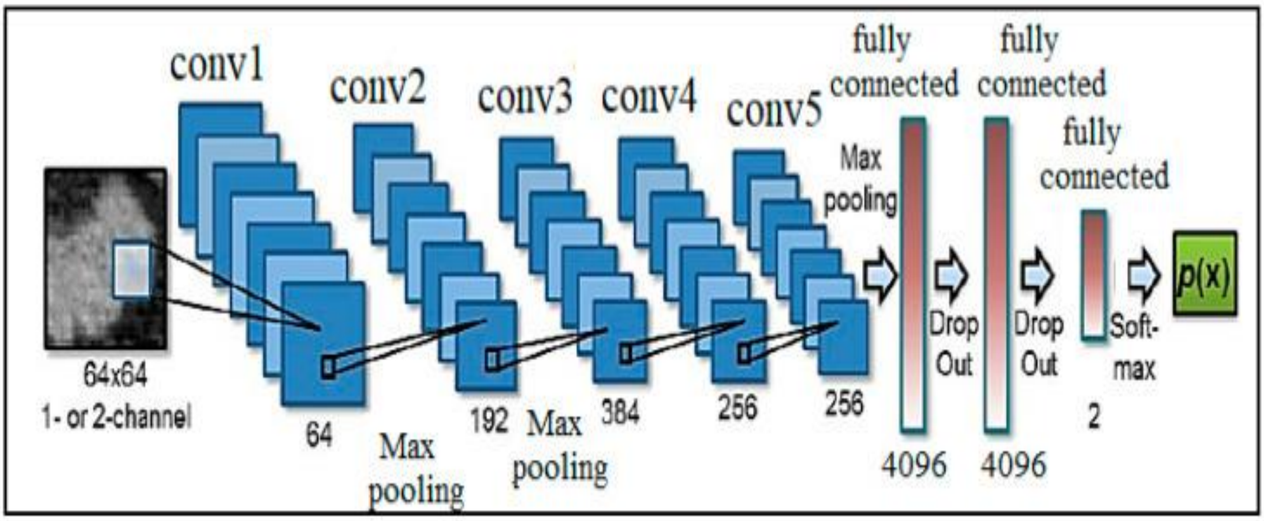


Fig.2 Basic CNN Architecture.

CNN-based automated deep learning algorithms have achieved remarkable performance in the detection, segmentation, and classification operations of medical imaging. Lequan proposed a very deep CNN for melanoma detection. A fully convolutional residual network (FCRN) having 16 residual blocks was used in the segmentation process to improve performance. The proposed technique used an average of both SVM and SoftMax classifier for classification. It showed 85.5% accuracy in melanoma classification with segmentation and 82.8% without segmentation. DeVries and Ramachandra proposed a multi-scale CNN using an inception v3 deep neural network that was trained on an ImageNet dataset. For skin cancer classification, the pre-trained inception v3 was further fined-tuned on two resolution scales of input lesion images: coarse-scale and finer scale. The coarse scale was used to capture shape characteristics as well as overall contextual information of lesions. In contrast, the finer scale gathered textual detail of lesion for differentiation between various types of skin lesions.

| S.no | Skin Cancer Diagnoses | Classifier and Training Algorithm | Dataset | Results (%) |
|---|---|---|---|---|
| 1. | Benign/malignant | LightNet (deep learning framework), used for classification | ISIC 2016 dataset | Accuracy (81.6), sensitivity (14.9), specificity (98) |
| 2. | Melanoma/benign | CNN classifier | 170 skin lesion images | Accuracy (81), sensitivity (81), specificity (80) |
| 3. | BCC/SCC/melanoma/AK | SVM with deep CNN | 3753 dermoscopic images | Accuracy (SCC: 95.1, AK: 98.9, BCC: 94.17) |
| 4. | Melanoma /benign Keratinocyte carcinomas/benign SK | Deep CNN | ISIC-Dermoscopic Archive | Accuracy (72.1) |
| 5. | Malignant melanoma and BC carcinoma | CNN with Res-Net 152 architecture | The first dataset has 170 images the second dataset contains 1300 images | AUC (melanoma: 96, BCC: 91) |

Table.1 A comparative analysis of skin cancer detection using CNN-based approaches.

## 2.3 KOHONEN SELF-ORGANIZING NEURAL NETWORK (KNN):

The Kohonen self-organizing map is a very famous type of deep neural network. CNNs are trained based on unsupervised learning, which means that a KNN does not require any developer's intervention in the learning process as well as requiring little information about the attributes of the input data. A KNN generally consists of two layers. In the 2-D plane, the first layer is called an input layer, while another is named a competitive layer. Both layers are fully connected, and every connection is from the first to second layer dimension. A KNN can be used for data clustering without knowing the relationships between input data members. It is also known as a self-organizing map.

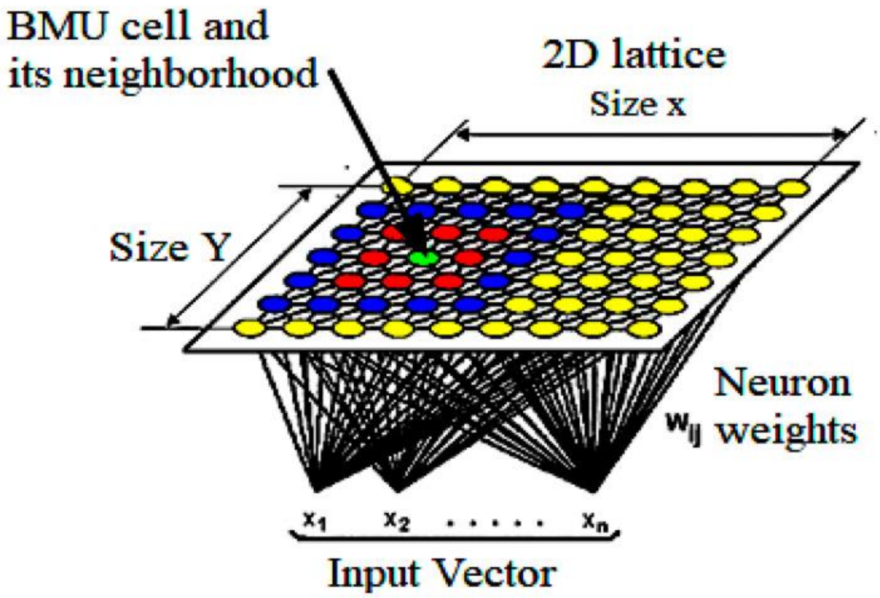


Fig.3 Basic KNN structure, BMU= Best matching unit.

Lenhardt proposed a KNN-based skin cancer detection system. The proposed system processed synchronous fluorescence spectra of melanoma, nevus, and normal skin samples for neural network training. A fluorescence spectrophotometer was used to measure the fluorescence spectra of the samples, whereas samples were collected from human patients immediately after surgical resection. The dimensionality of measured spectra was reduced with the PCA technique. Both KNN and ANN were trained, and their performance for melanoma detection was compared. On the test dataset, the classification error of KNN was 2–3%, while the classification error for ANN lay in the range of 3% to 4%.

### 2.4 GENERATIVE ADVERSARIAL NETWORK (GAN):

A generative adversarial neural network is a powerful class of DNN that is inspired by zero-sum game theory. GANs are based on the idea that two neural networks, such as a generator and a discriminator, compete to analyze and capture the variance in a database. The generator module uses the data distribution to produce fake data samples and tries to misguide the discriminator module. On the other hand, the discriminator module aims to distinguish between real and fake data samples. In the training phase, both neural networks repeat these steps, and their performance improves after each competition. The ability to generate fake samples that are like a real sample using the same data distribution, such as photorealistic images, is the major power of a GAN network. It can also solve a major problem in deep learning: the insufficient training examples problem. Research scholars have been implementing various types of GANs, such as Vanilla GAN, condition GAN (CGAN), deep convolutional GAN (DCGAN), super-resolution GAN (SRGAN), and Laplacian Pyramid GAN (LPGAN). Nowadays, GANs are successfully being used in skin cancer diagnostic systems.

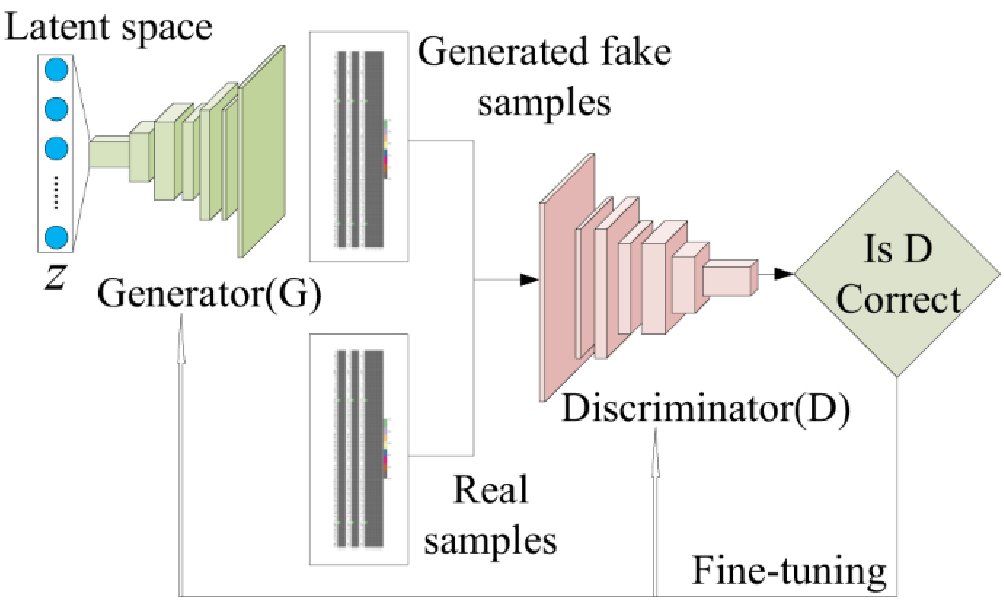


Fig.4 Basic GAN architecture.

# CHAPTER-3

# METHODS AND DATASET

## 3.1 DATASET:

The ISIC archive is a collection of various skin lesions datasets. The ISIC dataset was originally released by the International Skin Imaging Collaboration at the International Symposium on Biomedical Imaging (ISBI) challenge 2016, named ISIC2016. The ISIC2016 archive is divided into two parts: training and testing. The training subset of ISIC contains 900 images, while the testing subset contains 379 dermoscopic images. It includes images of two classes: malignant melanomas and benign nevi. Approximately 30.3% of the dataset's images are of melanoma lesions and the remaining images belong to the benign nevi class. ISIC increases the number of images in its archive every year and has established a design challenge for the development of a system for skin cancer automated diagnosis.

The International Skin Imaging Collaboration (ISIC): Melanoma Project is a partnership between industry and academia to facilitate the application of digital skin imaging to help curtail skin cancer mortality. Starting in 2015, ISIC started to organize global challenges for skin lesion analysis for melanoma diagnosis and detection. The first dataset which was 'ISIC 2018' contains 10,015 images of 7 types of skin lesion diseases namely: Benign Keratosis, Dermatofibroma, Vascular Lesion, Melanoma, Melanocytic Nevus, Basal Cell Carcinoma and Actinic Keratosis. These images were collected with approval of Medical University of Vienna and University of Queensland. All the images are in the standard size of 600 × 450 pixels in JPEG format.

Figure 5 shows samples of each class from the dataset.

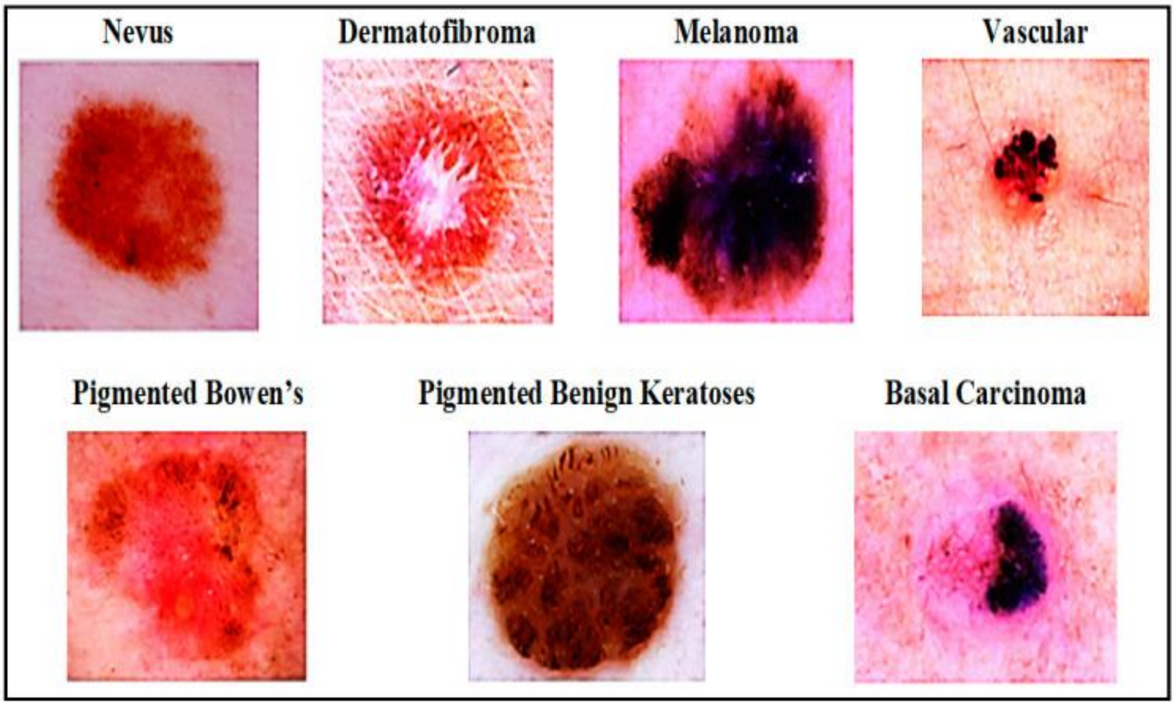


Fig.5 Skin disease categories from International Skin Imaging Collaboration (ISIC) dataset

Table 2 summarizes the dataset 1 used for this project.

The second dataset which was 'ISIC 2019' contains 25,333 images of 8 types of skin lesion diseases namely: Dermatofibroma, Vascular Lesion, Squamous Cell Carcinoma, Melanoma, Melanocytic Nevus, Basal Cell Carcinoma, Actinic Keratosis, Benign Keratosis. All images are in the size of 1022 × 767 pixels in JPEG format.

| Datasets | ISIC challenge 2018 |
|---|---|
| Type | Dermoscopic |
| Image size | 600 pixels × 450 pixels |
| Number of images | 10,015 |
| Image type | JPEG (RGB) |
| Class labels | 0: Melanoma<br>1: Melanocytic Nevus<br>2: Basal Cell Carcinoma |

| | |
|---|---|
| | 3: Actinic Keratosis<br>4: Benign Keratosis<br>5: Dermatofibroma<br>6: Vascular Lesion |

Table.2 Dataset information for ISIC 2018

Table 3 summarizes the dataset 2 used for this project.

A new dataset was created by combining both the datasets of ISIC 2018 and ISIC 2019. The seven common classes of skin lesions were retained, and extra noise was removed from the dataset. This enabled the models to learn more efficiently due to the abundance of samples now available per class.

| Datasets | ISIC Challenge 2019 |
|---|---|
| Type | Dermoscopic |
| Image size | 1022 pixels × 767 pixels |
| Number of images | 25,333 |
| Image type | JPEG (RGB) |
| Class labels | 0: Melanoma<br>1: Melanocytic Nevus<br>2: Basal Cell Carcinoma<br>3: Actinic Keratosis<br>4: Benign Keratosis<br>5: Dermatofibroma<br>6: Vascular Lesion<br>7: Squamous Cell Carcinoma |

Table.3 Dataset information for ISIC 2019

Table 4 summarizes the final dataset used for this project.

| Datasets | FINAL |
|---|---|

| | |
|---|---|
| Type | Dermoscopic |
| Image size | 600 pixels × 450 pixels (10,015)<br>1022 pixels × 767 pixels (25,333 |
| Number of images | 35,348 |
| Image type | JPEG (RGB) |
| Class labels | 0: Melanoma<br>1: Melanocytic Nevus<br>2: Basal Cell Carcinoma<br>3: Actinic Keratosis<br>4: Benign Keratosis<br>5: Dermatofibroma<br>6: Vascular Lesion |

Table.4 Dataset information for FINAL dataset

Table 5 shows the number of images fed per class into the network.

| Lesion type | Number of images |
|---|---|
| Melanoma | 5635 |
| Melanocytic Nevus | 19,580 |
| Basal Cell Carcinoma | 3837 |
| Actinic Keratosis | 1194 |
| Benign Keratosis | 3723 |
| Dermatofibroma | 354 |
| Vascular Lesion | 395 |

Table.5 Number of images per lesion type

## 3.2 METHOD OVERVIEW:

### 3.2.1 Data Augmentation

Contemporary advances in deep learning models have been largely associated with large quantity and diversity of data. Having large data helps crucially in improving the performance of machine learning models. But obtaining such vast quantities of data is cost-intensive and tedious. Hence, we use the technique of Data Augmentation. It is a technique that enables us to considerably increase the diversity and quantity of data available, without aggregating new data. To generate new data through augmentation of images, various techniques such as cropping, padding, adding noise, brightness changing, and horizontal flipping are commonly used to train large neural networks.

In this project, the training images are augmented to make the model robust to new data which in turn helps in increasing the testing accuracy.

Table 6 shows the augmentation techniques used on the dataset.

| Augmentation technique | Range |
|---|---|
| Zoom range | 0.1 |
| Rotation range | 10 |
| Horizontal flip | False |
| Rescale | 1./255 |
| Width shift range | 0.1 |
| Height shift range | 0.1 |

Table.6 Information about data augmentation parameters

### 3.2.2 Image Normalization

Image Normalization is a technique used to normalize the pixel values of the image in a similar distribution. It is beneficial to normalize images before feeding into the neural network as this helps in approaching the global minima at error surface at a faster rate while performing gradient descent. In a way, it helps the network to converge faster. Also, the computations become significantly less intensive for the machine to perform as all the pixel values are scaled.

### 3.2.3 Transfer Learning

Transfer Learning is a learning method in which a model trained for a particular task is reiterated as the origin for another model on a similar task. This approach is very mainstream in deep learning due to the vast computation resources and time consumed to train neural network models. In the case of problems in the computer vision domain, low-level features, such as shapes, corners, edges and intensity, can be shared across tasks, and thus enable knowledge transfer.

The model is trained and tested on the state-of-art CNNs, namely Inception V3, ResNet50, VGG16, MobileNet and Inception Resnet to perform the seven-class classification of skin lesion images.

### 3.2.4 Dropout

Deep Neural Networks (DNNs) may overfit a dataset with meager training samples. Collection or Ensemble of neural networks with different architectures are known to diminish overfitting, but still require us to train and maintain various models which becomes computationally intensive.

This is where Dropout comes into the picture. A single model is used to resemble having a vast number of distinct network architectures by arbitrarily dropping out nodes during training. This is how dropout works and proposes a computationally viable and exceptionally effective method to cut down overfitting and enhance generalization error in DNNs of all types.

## 3.3 NETWORK ARCHITECTURE:

### 3.3.1 Inception V3

Inception V3 is a widely acclaimed image recognition model that has been built by various researchers over time. Originally based on the paper "Rethinking the Inception Architecture for Computer Vision" by Szegedy, the model has attained an accuracy greater than 78.1% on the ImageNet dataset.

Figure 6 shows the model architecture of Inception V3 network.

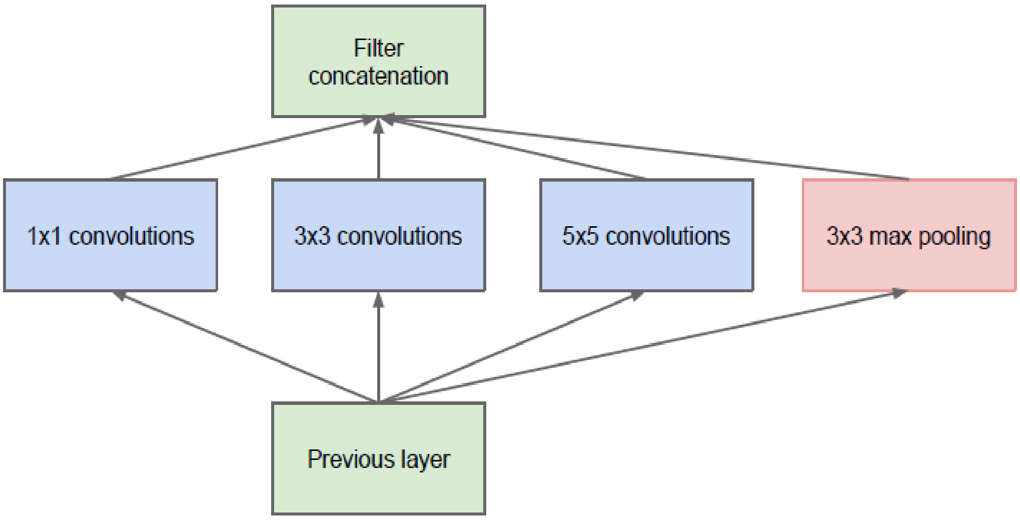


Fig.6 Inception V3 architecture

Inception V3 proposed by Google is the third iteration in the series of Deep Learning Convolutional Architectures. The model was trained on 1000 classes from the ImageNet dataset which was itself trained on about 1 million images. It consists of the inception modules which apply filters of multiple sizes on the same level of input. The high computation cost involved in the inception module is solved by applying 1 × 1 convolution to truncate the input into a smaller intermediate block, called the bottleneck layer. The multiple filters are applied on the bottleneck layer to significantly reduce the computation cost. The auxiliary classifiers in the Inception networks contribute to the weighted loss function for regularization purposes. To utilize the ImageNet pretrained weight of the Inception V3 network, the input image must be in the 299 × 299 size.

### 3.3.2 ResNet50

Residual Network (ResNet) is a typical model of neural network used as an integral part for various computer vision tasks. This ResNet model won the 2015 ImageNet challenge. The network is 50 layers deep and takes the input image size of 224 × 224 pixels. Generally, in a deep CNN, many layers are stacked and trained. The network learn many low levels, middle and high-level features. In residual learning, rather than learning some features, we learn some residual. Residual can be simply interpreted the reduction of features learned from input layer. It has been shown that training residual networks are easier compared to training simple deep CNNs. It also helps to tackle the problem of degrading accuracy.

A residual block of the ResNet50 model is shown in Fig. 7.

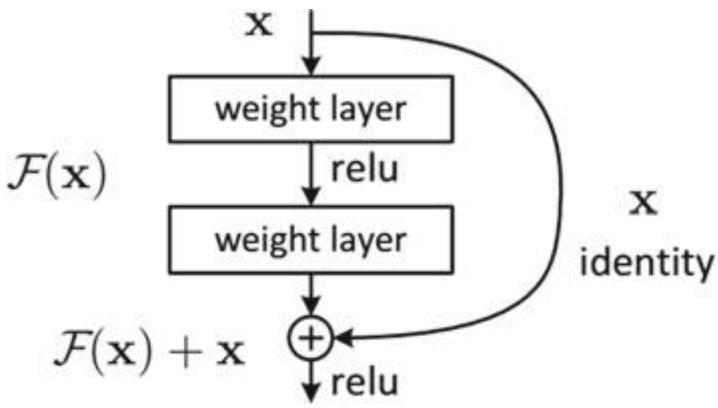


Fig. 7 Residual block of the ResNet 50 model

### 3.3.3 VGG 16

VGG16 is a CNN model proposed by A. Zisserman and K. Simonyan from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition".

Figure 8 shows the model architecture of VGG 16 network.

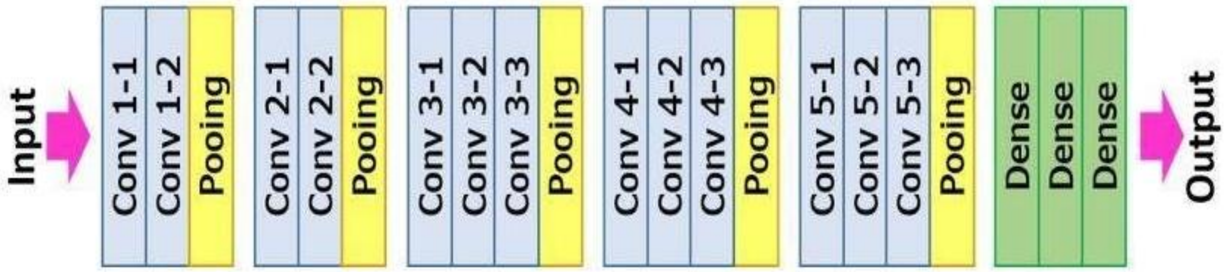


Fig. 8 VGG 16 architectures

The input of this model must be a 224 × 224 RGB image. The image is passed through a pile of convolutional layers, with kernel size or filter size of 3 × 3. The stride is set to 1 pixel; the padding of convolution layer input is set in a way such that the spatial resolution is preserved after convolution, i.e., the padding is 1-pixel for 3 × 3 conv. layers. Max-pooling is done over a 2 × 2-pixel window, with a stride of 2 pixels.

After the stack of convolutional layers, 3 Fully Connected (FC) layers follow. The initial two layers have 4096 channels each, the third layer contains 1000 channels. The final layer has the SoftMax activation function. All other hidden have the rectification (ReLU) non-linearity activation.

### 3.3.4 Inception-Resnet

Inception-ResNet-v2 is a CNN model that has been trained on the ImageNet database. It contains 164 layers and has the capability to classify images into 1000 categories, such as mouse, many animals, keyboard, pencil etc. The network is robust and has learned excellent feature representations for various kinds of images. The input to the network must be an image of size 299 × 299.

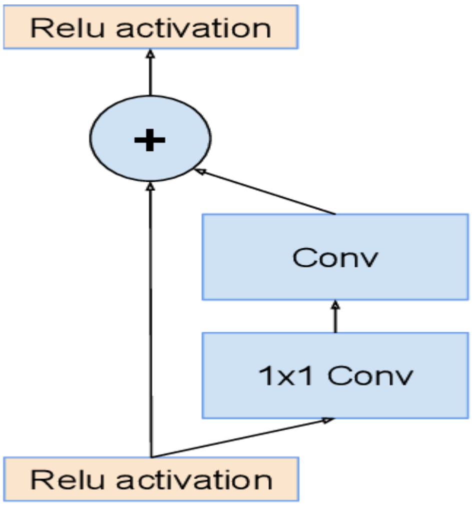


Fig.9 Inception Resnet V2 architecture

Overall, Inception Resnet V2 has a similar structure and computation cost as Inception V4. It additionally introduces the residual connection to the submodules Inception Block A, B and C at the left side to enable the network to go deeper.

### 3.3.5 MobileNet

MobileNet are based on the principle of streamlined architectures which use depth wise separable convolutions followed by point-wise convolutions that considerably reduce the number of learnable parameters and assist in building lightweight deep neural networks. Effectively, it decreases the total number of floating-point calculations required which is supportive for embedded and mobile computer vision applications where there is shortage of computational power. The architecture was proposed by Google.

Figure 10 shows the depth-wise and point-wise convolution operations.

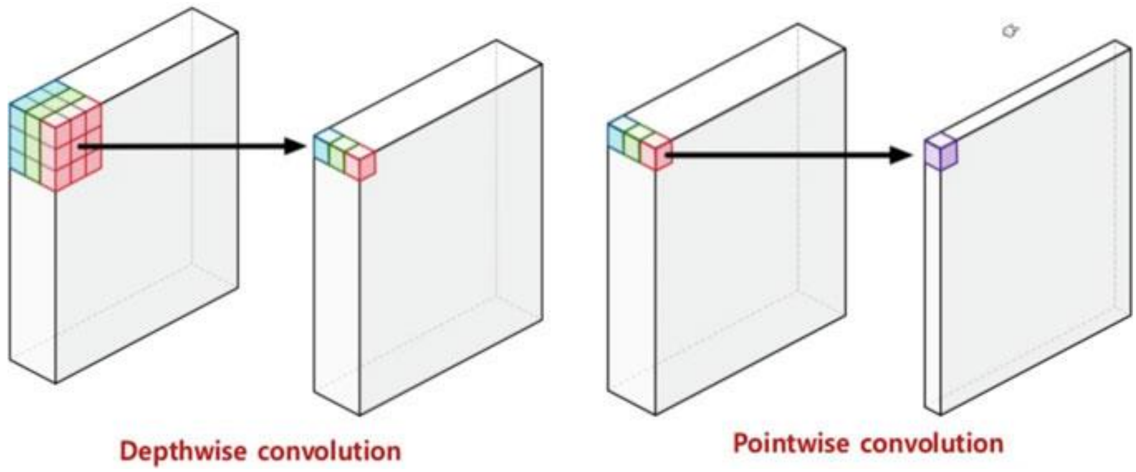


Fig. 10 Depth wise and pointwise convolution

## CHAPTER-4

## WORKING OF CONVOLUTION NEURAL NETWORKS

The question which arises here is how does CNN understand translation invariance? Is it the magic of Machine Learning? Yet again, it comes down to mathematics again. The following operations are the various layers/steps of the CNN:

• Convolution

• Pooling

• Flattening

• Full Connection

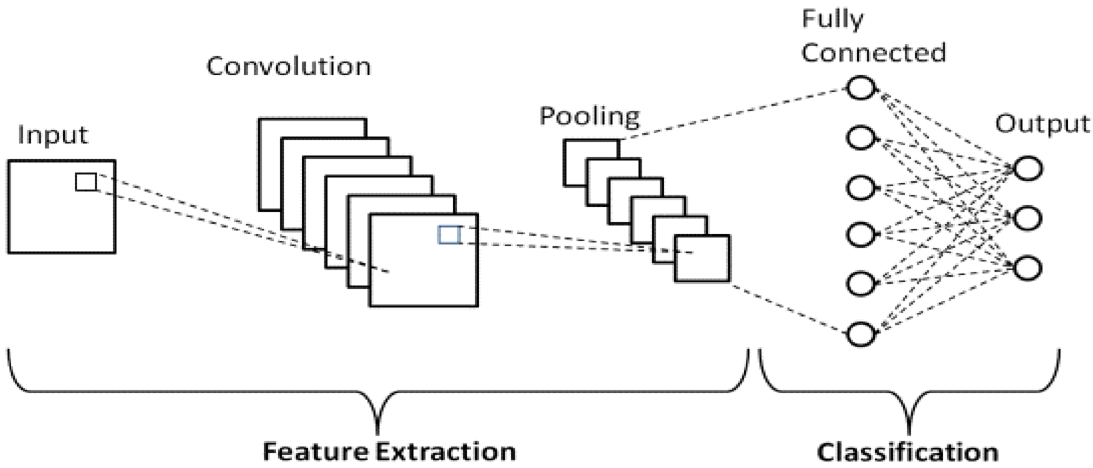


Fig. 11 Basic CNN working architecture

### 4.1 CONVOLUTION:

The first operation, Convolution, extracts important features from the image. It is a mathematical operation which clearly requires two inputs, an image matrix and a filter or kernel. The filter is traversed through the image and multiplied with the pixel values to obtain feature map.

Figure 12 shows how the convolution operation happens.

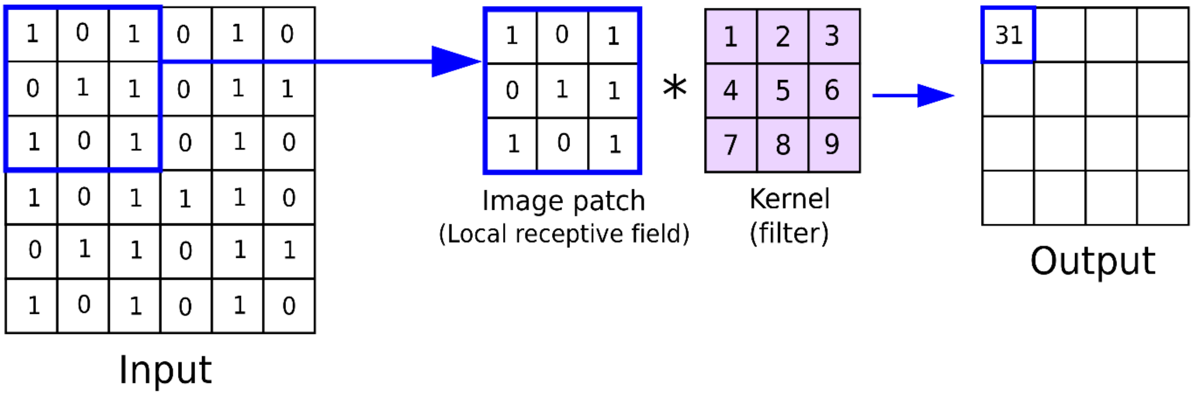


Fig. 12 Convolution operation

Convolution does lose information, but the point here is to reduce size and learn the integral information. Performing convolution with different kinds of filters can assist in image sharpening, edge detection, blurring and other image processing operations.

### 4.2 POOLING:

Pooling operation helps in decreasing the number of parameters when the image is very large in size. Subsampling, also called Spatial Pooling curtails the dimensionality of each feature map but retains significant information.

Pooling is of basically divided into three types:

- Max Pooling (mostly used)
- Sum Pooling
- Average Pooling

Max pooling is a sample-based discretization process. It is done by applying a N × N max filter over the image, which selects the highest pixel value in each stride and builds the feature map. Similarly, in average and sum pooling, the average and sum of pixel values are taken into the feature map.

Figure 13 depicts the Max Pooling operation.

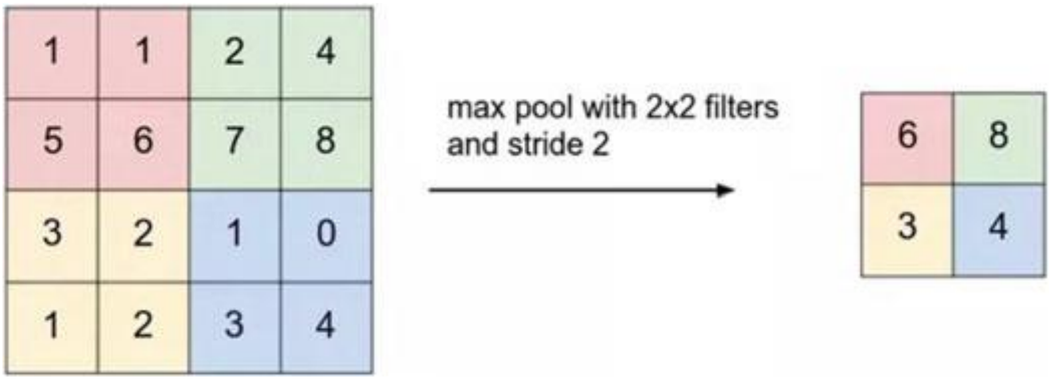


Fig. 13 Max pooling operation

### 4.3 FLATTENING:

To feed our feature maps into the artificial neural network, we need a single column vector of the image pixels. As the name suggests, we flatten our feature maps into columns like vectors.

Figure 14 depicts the flattening process.

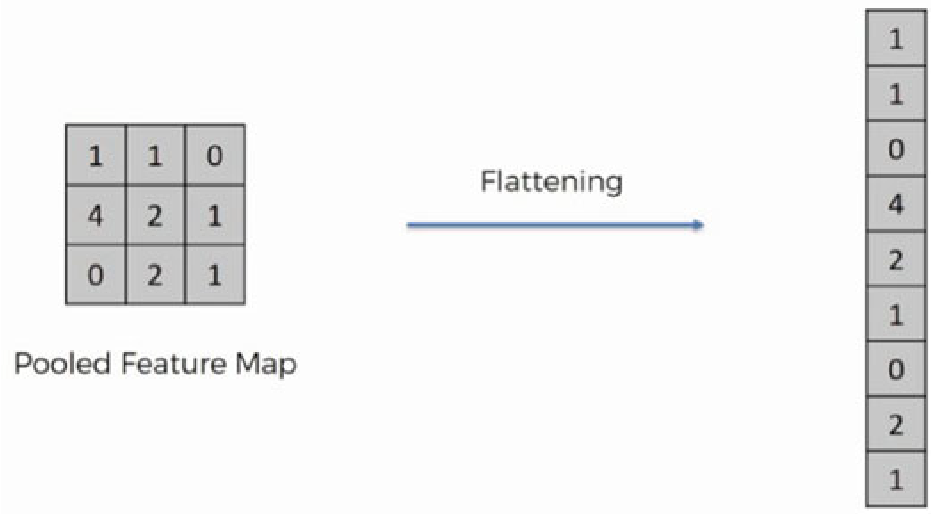


Fig. 14 Flattening operation

## 4.4 FULL CONNECTION:

The fully connection layer takes the input from the preceding convolution/pooling layer and produces an N dimensional vector where N is the number of classes to be classified. Thus, the layer determines the features most correlating to a particular class based on the probabilities of the neurons.

Figure 15 shows the fully connected layer in a neural network.

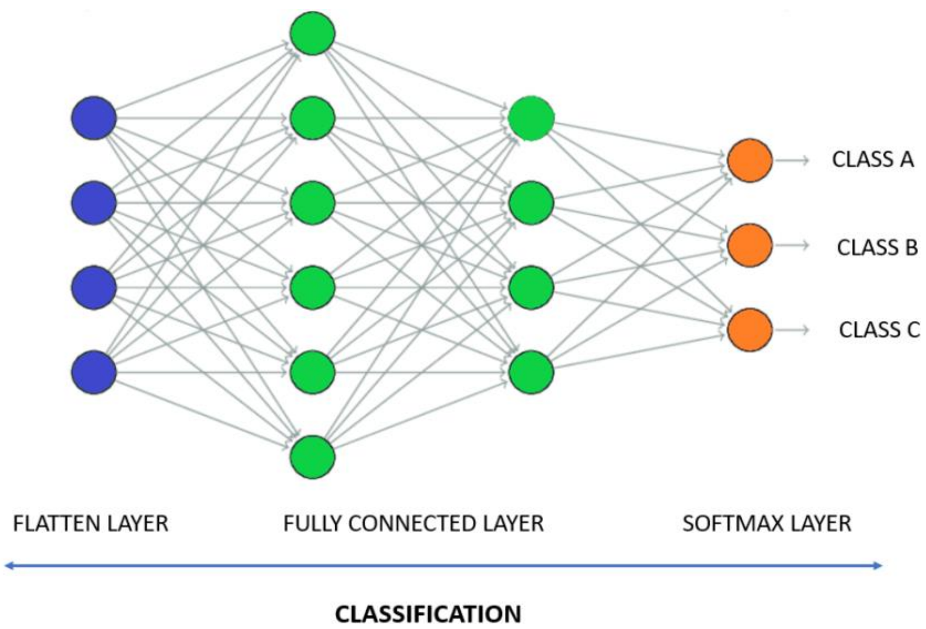


Fig. 15 Full connection layer

## 4.5 SKIN LESION CLASSIFICATION USING CNN:

From past research in this field, it is evident that CNN has an extraordinary ability to perform skin lesion classification in competition with professional dermatologists. In fact, there have been instances where CNN has outperformed professional dermatologists as well.

CNN can perform skin lesion classification in two ways. In the first case, a CNN is used as the feature extractor of the images and the classification is performed by another classifier. For the other case, CNN is used to perform end-to-end learning which can be further divided into learning from scratch or learning from pretrained model. To train CNN from scratch, many images are required to tackle the overfitting issue. Since the number of skin lesion images to do the training is not sufficient, training CNN from scratch is less feasible. Training from a pretrained model is a better approach which is generally referred to as Transfer Learning (TL). TL helps the model to learn well even with less data and introduces generalization property to the trained model.

### 4.6 CNN FOR IMAGE CLASSIICATION:

Artificial Neural Networks are made of artificial neurons inspired by biological neurons present in our brain. Convolutional Neural Network (CNN) is a modified variant of feed-forward neural network which is generally used for image classification tasks. CNNs can recognize a particular object even when it appears in different ways, as it understands translation invariance. This is a key point which makes CNN advantageous over feed-forward neural networks which cannot understand translation invariance.

In layman words, feed-forward neural networks only recognize an object when it is right in the center of the image but fails notably when the object is slightly off position or placed elsewhere in the image. Basically, the network understands/learns only one pattern. This is precisely not convenient as the real-world datasets are usually raw and unprocessed.

# CHAPTER-5

## MODEL RESULTS AND DISCUSSION

### 5.1 PERFORMANCE METRICS:

In a classification problem, only one metric such as Accuracy cannot help us evaluate the complete model efficiency effectively. Hence, we measure the Accuracy, Precision, Recall, F1 Score and Support for every class of skin lesion disease. in every class. We will now understand how the above-mentioned metrics are calculated and what they mean.

For example, if the skin lesion image is labelled with the melanoma and the model also predicts it as melanoma, this is considered as the true positive case. If the image is labelled melanoma but it is classified as any of the other six classes, this is the case of false negative. False positive cases happen when the skin lesion image is indicated by the classification model to have melanoma, but it belongs to any of the other six diseases. If a non-melanoma skin lesion image is suggested as non-melanoma by the classifier, it is the case of true negative

#### 5.1.1 Accuracy

Accuracy is the fraction of predictions our model has correctly guessed.

It is defined as:

$$\text{Accuracy} = T\,P + T\,N \,/\, T\,P + T\,N + FP + FN$$

#### 5.1.2 Precision

Precision metric answers the following question: What proportion of positive identifications were correct?

It is defined as:

$$\text{Precision} = TP / TP + FP$$

### 5.1.3 Sensitivity/Recall

Sensitivity is also called the True Positive Rate (TPR) or Recall. It answers the following question: What proportion of actual positives was identified correctly?

It is defined as:

$$\text{Sensitivity} = TP / TP + FN$$

### 5.1.4 F1 Score

F1 Score is also known as the F-score or F-measure. The F1 score is calculated as a weighted average of the precision and recall. Its best value is 1 and worst is 0. The contribution of recall and precision in the calculation of the F1 score are equal.

It is defined as:

$$F1 = 2 * (\text{precision} * \text{recall} / \text{precision} + \text{recall})$$

### 5.1.5 Support

Support is defined as the number of actual occurrences of the class in the specified dataset.

## 5.2 HYPERPARAMETERS:

Table 7 shows the hyperparameters used to train the CNN models.

| | |
|---|---|
| Optimizer | Adam optimizer |
| Learning rate | 0.0001 |
| Epochs | 30 |
| Loss function | Categorical cross-entropy |
| Batch size | 64 |
| Dropout | 0.4 |

Table.7 Hyperparameters used for training the CNN models

## 5.3 MODEL PERFORMANCE:

### 5.3.1 ResNet50

This section represents the classification performance of the ResNet50 model. The input images were resized to 224 × 224 as this model requires the input to be in that format.

Table 8 shows the confusion matrix of this model.

| | AKIEC | BCC | BKL | DF | MEL | NV | VASC |
|---|---|---|---|---|---|---|---|
| AKIEC | **115** | 16 | 79 | 7 | 11 | 10 | 0 |
| BCC | 33 | **574** | 72 | 4 | 31 | 52 | 2 |
| BKL | 11 | 17 | **596** | 1 | 29 | 36 | 0 |
| DF | 1 | 5 | 6 | **54** | 6 | 5 | 0 |
| MEL | 4 | 13 | 106 | 1 | **828** | 170 | 0 |
| NV | 3 | 15 | 117 | 8 | 135 | **3690** | 2 |
| VASC | 0 | 1 | 0 | 1 | 2 | 3 | **72** |

Table.8 ResNet50 confusion matrix

The bold numbers in the given table represent the correctly classified number of images belonging to a given class

Table 9 presents the classification report of this model.

| | Precision | Recall | F1 score | Support |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| Actinic Keratosis | 0.69 | 0.48 | 0.57 | 238 |
| Basal Cell Carcinoma | 0.90 | 0.75 | 0.81 | 768 |
| Benign Keratosis | 0.61 | 0.86 | 0.72 | 690 |
| Dermatofibroma | 0.71 | 0.70 | 0.71 | 77 |
| Melanoma | 0.79 | 0.74 | 0.77 | 1122 |
| Melanocytic Nevus | 0.93 | 0.93 | 0.93 | 3970 |
| Vascular Lesion | 0.95 | 0.91 | 0.93 | 79 |

Table.9 ResNet50 classification report

The average metrics achieved by the model are presented in Table 10.

| | |
|---|---|
| Accuracy | 0.85 |
| Precision | 0.86 |
| Recall | 0.85 |
| F1 score | 0.85 |
| Support | 6944 |

Table.10 ResNet50 average metrics

### 5.3.2 MobileNet

This section represents the classification performance of the MobileNet model. The input images were resized to 224 × 224 as this model requires the input to be in that format.

Table 11 shows the confusion matrix of this model.

| | AKIEC | BCC | BKL | DF | MEL | NV | VASC |
|---|---|---|---|---|---|---|---|
| AKIEC | **119** | 20 | 65 | 1 | 20 | 13 | 0 |
| BCC | 22 | **591** | 57 | 7 | 33 | 57 | 1 |
| BKL | 11 | 20 | **582** | 0 | 29 | 48 | 0 |
| DF | 1 | 5 | 10 | **42** | 5 | 14 | 0 |
| MEL | 7 | 18 | 85 | 0 | **728** | 230 | 0 |
| NV | 4 | 26 | 127 | 2 | 53 | **3753** | 6 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| VASC | 0 | 1 | 2 | 1 | 2 | 7 | **66** |

Table.11 MobileNet confusion matrix

The bold numbers in the given table represent the correctly classified number of images belonging to a given class

Table 12 presents the classification report of this model.

| | Precision | Recall | F1 score | Support |
|---|---|---|---|---|
| Actinic Keratosis | 0.73 | 0.50 | 0.59 | 238 |
| Basal Cell Carcinoma | 0.87 | 0.77 | 0.82 | 768 |
| Benign Keratosis | 0.63 | 0.84 | 0.72 | 690 |
| Dermatofibroma | 0.79 | 0.55 | 0.65 | 77 |
| Melanoma | 0.86 | 0.70 | 0.76 | 1122 |
| Melanocytic Nevus | 0.91 | 0.95 | 0.93 | 3970 |
| Vascular Lesion | 0.90 | 0.84 | 0.87 | 79 |

Table.12 MobileNet classification report

The average metrics achieved by the model are presented in Table 13.

| | |
|---|---|
| Accuracy | 0.85 |
| Precision | 0.86 |
| Recall | 0.85 |
| F1 score | 0.85 |
| Support | 6944 |

Table.13 MobileNet average metrics

### 5.3.3 VGG16

This section represents the classification performance of the VGG16 model. The input images were resized to 224 × 224 as this model requires the input to be in that format.

Table 14 shows the confusion matrix of this model.

| | AKIEC | BCC | BKL | DF | MEL | NV | VASC |
|---|---|---|---|---|---|---|---|
| AKIEC | **105** | 44 | 48 | 4 | 19 | 18 | 0 |
| BCC | 7 | **656** | 30 | 1 | 24 | 47 | 3 |
| BKL | 9 | 15 | **545** | 2 | 30 | 89 | 0 |
| DF | 1 | 7 | 5 | **45** | 2 | 14 | 3 |
| MEL | 4 | 22 | 49 | 0 | **824** | 223 | 0 |
| NV | 3 | 24 | 52 | 3 | 93 | **3794** | 1 |
| VASC | 0 | 3 | 0 | 1 | 2 | 2 | **71** |

Table.14 VGG16 confusion matrix

The bold numbers in the given table represent the correctly classified number of images belonging to a given class

Table 15 presents the classification report of this model.

| | Precision | Recall | F1 score | Support |
|---|---|---|---|---|
| Actinic Keratosis | 0.81 | 0.44 | 0.57 | 238 |
| Basal Cell Carcinoma | 0.85 | 0.85 | 0.85 | 768 |
| Benign Keratosis | 0.75 | 0.79 | 0.77 | 690 |
| Dermatofibroma | 0.80 | 0.58 | 0.68 | 77 |
| Melanoma | 0.83 | 0.73 | 0.78 | 1122 |
| Melanocytic Nevus | 0.91 | 0.96 | 0.93 | 3970 |
| Vascular Lesion | 0.91 | 0.90 | 0.90 | 79 |

Table.15 VGG16 classification report

The average metrics achieved by the model are presented in Table 16.

| | |
|---|---|
| Accuracy | 0.87 |
| Precision | 0.87 |
| Recall | 0.87 |
| F1 score | 0.87 |

| | |
|---|---|
| Support | 6944 |

Table.16 VGG16 average metrics

### 5.3.4 Inception V3

This section represents the classification performance of the Inception V3 model. The input images were resized to 299 × 299 as this model requires the input to be in that format.

Table 17 shows the confusion matrix of this model.

| | AKIEC | BCC | BKL | DF | MEL | NV | VASC |
|---|---|---|---|---|---|---|---|
| AKIEC | **151** | 27 | 35 | 0 | 18 | 7 | 0 |
| BCC | 15 | **667** | 25 | 5 | 23 | 32 | 1 |
| BKL | 12 | 18 | **593** | 0 | 34 | 33 | 0 |
| DF | 1 | 5 | 6 | **54** | 2 | 8 | 1 |
| MEL | 7 | 11 | 39 | 1 | **954** | 110 | 0 |
| NV | 2 | 18 | 57 | 3 | 125 | **3764** | 1 |
| VASC | 0 | 0 | 0 | 1 | 0 | 4 | **74** |

Table.17 Inception V3 confusion matrix

The bold numbers in the given table represent the correctly classified number of images belonging to a given class

Table 18 presents the classification report of this model.

| | Precision | Recall | F1 score | Support |
|---|---|---|---|---|
| Actinic Keratosis | 0.80 | 0.63 | 0.71 | 238 |
| Basal Cell Carcinoma | 0.89 | 0.87 | 0.88 | 768 |
| Benign Keratosis | 0.79 | 0.86 | 0.82 | 690 |
| Dermatofibroma | 0.84 | 0.70 | 0.77 | 77 |
| Melanoma | 0.83 | 0.85 | 0.84 | 1122 |
| Melanocytic Nevus | 0.95 | 0.95 | 0.95 | 3970 |

| | | | | |
|---|---|---|---|---|
| Vascular Lesion | 0.96 | 0.94 | 0.95 | 79 |

Table.18 Inception V3 classification report

The average metrics achieved by the model are presented in Table 19.

| | |
|---|---|
| Accuracy | 0.90 |
| Precision | 0.90 |
| Recall | 0.90 |
| F1 score | 0.90 |
| Support | 6944 |

Table.19 Inception V3 average metrics

### 5.3.5 InceptionResnet V2

This section represents the classification performance of the InceptionResnet model. The input images were resized to 299 × 299 as this model requires the input to be in that format.

Table 20 shows the confusion matrix of this model.

| | AKIEC | BCC | BKL | DF | MEL | NV | VASC |
|---|---|---|---|---|---|---|---|
| AKIEC | **142** | 42 | 35 | 1 | 11 | 7 | 0 |
| BCC | 1 | **717** | 16 | 2 | 14 | 18 | 0 |
| BKL | 3 | 23 | **612** | 1 | 18 | 32 | 1 |
| DF | 1 | 4 | 5 | **62** | 1 | 4 | 0 |
| MEL | 5 | 19 | 36 | 0 | **943** | 119 | 0 |
| NV | 1 | 36 | 52 | 4 | 98 | **3778** | 1 |
| VASC | 0 | 1 | 0 | 1 | 1 | 2 | **74** |

Table.20 InceptionResnet confusion matrix

The bold numbers in the given table represent the correctly classified number of images belonging to a given class

Table 21 presents the classification report of this model.

| | Precision | Recall | F1 score | Support |
|---|---|---|---|---|
| Actinic Keratosis | 0.93 | 0.60 | 0.73 | 238 |
| Basal Cell Carcinoma | 0.85 | 0.93 | 0.89 | 768 |
| Benign Keratosis | 0.81 | 0.89 | 0.85 | 690 |
| Dermatofibroma | 0.87 | 0.81 | 0.84 | 77 |
| Melanoma | 0.87 | 0.84 | 0.85 | 1122 |
| Melanocytic Nevus | 0.95 | 0.95 | 0.95 | 3970 |
| Vascular Lesion | 0.97 | 0.94 | 0.95 | 79 |

Table.21 InceptionResnet classification report

The average metrics achieved by the model are presented in Table 22.

| | |
|---|---|
| Accuracy | 0.91 |
| Precision | 0.91 |
| Recall | 0.91 |
| F1 score | 0.91 |
| Support | 6944 |

Table.22 InceptionResnet average metrics

## 5.4 COMPARISON AND DISCUSSION:

Table 23 presents the average accuracy, precision, recall, F1 score and Support metrics among the seven-disease classification achieved by all the final CNNs for comparison.

| | Accuracy | Precision | Recall | F1 score | Support |
|---|---|---|---|---|---|
| ResNet50 | 0.85 | 0.86 | 0.85 | 0.85 | 6944 |
| MobileNet | 0.85 | 0.86 | 0.85 | 0.85 | 6944 |
| VGG16 | 0.87 | 0.87 | 0.87 | 0.87 | 6944 |
| Inception V3 | 0.90 | 0.90 | 0.90 | 0.90 | 6944 |
| InceptionResnet | **0.91** | **0.91** | **0.91** | **0.91** | **6944** |

Table.23 Average metrics achieved by all final CNNs

The bold numbers in the given table represent the correctly classified number of images belonging to a given class

Figure 16 shows the learning curve of all the CNN models.

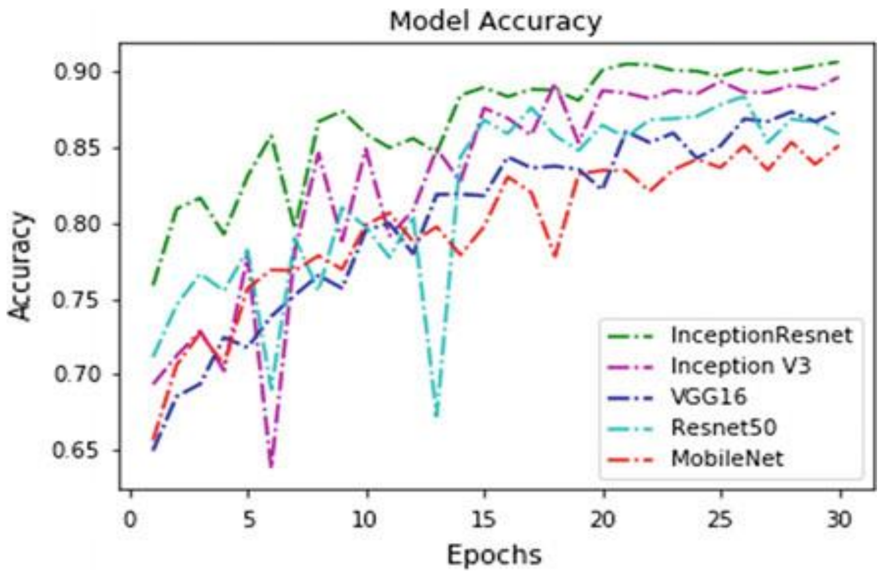


Fig.16 CNN model accuracy curves

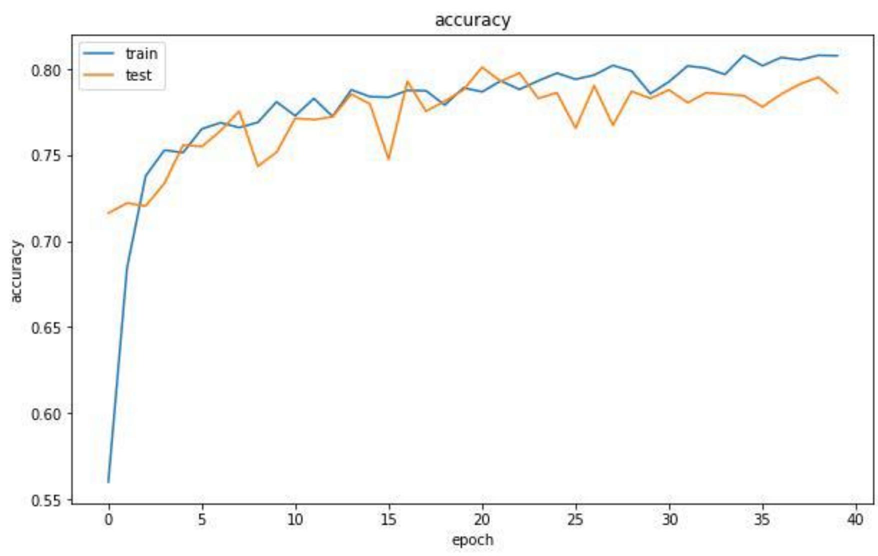


Fig. 17 Accuracy of VGG16.

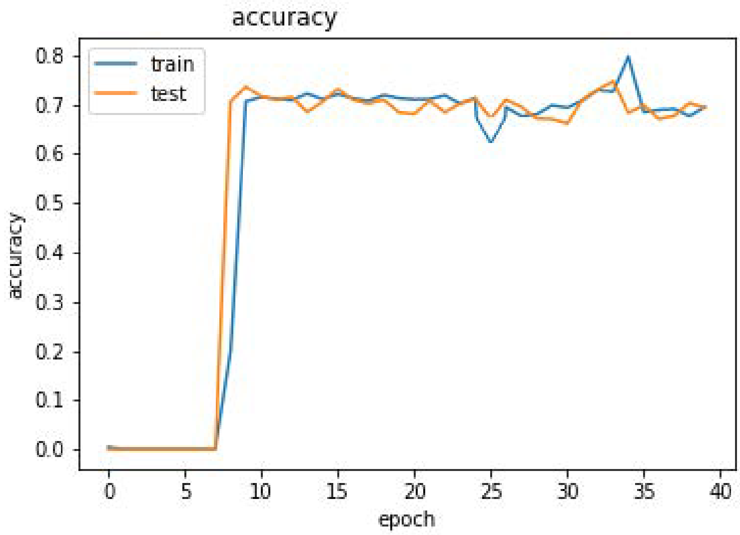


Fig. 18 Accuracy of MobileNet.

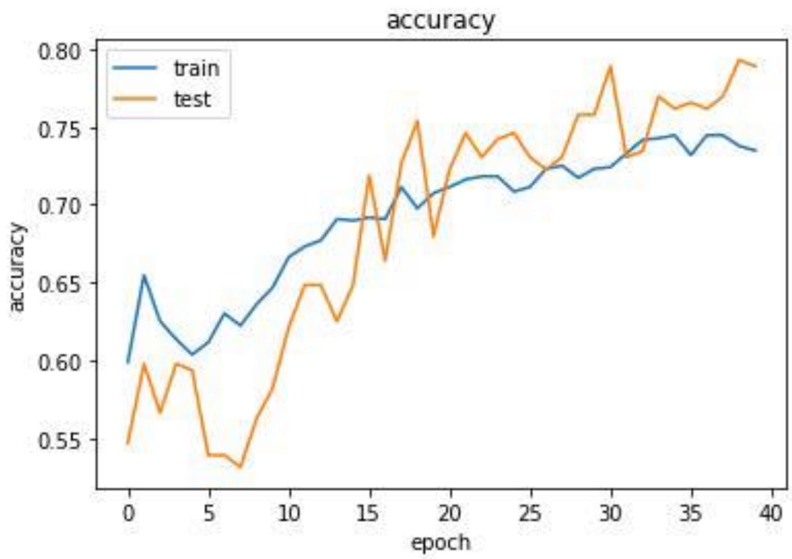


Fig.19 Accuracy of ResNet.

From the Table 23 and Fig.16, it is clearly visible that the Inception V3 and InceptionResnet CNN models have given the highest classification performance with Accuracies of 90 and 91%. This model is robust enough to classify lesion images in any of the seven types.

# CHAPTER-6

## PROPOSED MODEL

The perception of skin disease is accomplished through two phases. Phase I involves collection and preprocessing of dataset and the training phase and the testing phase of the developed Deep CNN model. Phase II includes real time implementation and visualization of results in GUI. Since one of the factors that determines the accuracy of prediction is the database, we combined at least six different databases which are collected by different physicians / researchers / medical students / pathologists / competitions. Also, for each image in the database, the manual segmentation and the clinical diagnosis of the skin lesion as well as the identification of other important dermoscopic criteria is available. These dermoscopic criteria include the assessment of the lesion asymmetry, and the identification of colors and several differential structures, such as pigment network, dots, globules, streaks, regression areas and blue-whitish veil. The datasets consist of images collected from International Skin Imaging Collaboration (ISIC) 2018 Challenge, HAM10000, Benign vs. Malignant and PH2. The dataset was divided into training and testing set in the ratio 8:2. Each image in the dataset undergoes a preprocessing part, which involves rescaling of image and labeling of image. Label '0' is assigned for benign class and '1' is assigned for malignant class.

### 6.1 PHASE I – TRAINING AND TESTING OF MODEL:

#### 6.1.1 Training of model

The preprocessed images from the training set were supplied into the proposed deep convolutional neural network. Features were extracted from the image through a series of convolution, pooling, and ReLU layers. The proposed neural network has 5 hidden layers. As the input image proceeds through these layers, the features are extracted one by one and pass onto the next layer. The image is convoluted, max pooled, and then it is again convoluted, and then average pooling is done. There are 2 dense layers at the end. Global average pooling layer is used to reduce overfitting by minimizing the number of parameters in the model. The dense layers are the fully connected layer that classifies the images into the benign and malignant category.

### 6.1.2 Neural network Architecture

An artificial neural network is an interconnected group of nodes inspired by a simplification of neurons in the brain. Thousands of such neurons when combined form the bases of a neural network wherein the connectivity between neurons tend to capture the invariance of patterns to distortion or shift in the input data. Three main types of layers in Deep CNN model are: Convolutional Layers, Pooling Layers, and Dense Layer.

1) **Convolutional Layers:** A convolutional layer is produced by a higher-level abstraction of the input data which is, in turn, called a feature map. Units in a convolutional layer are arranged in a feature map, within which each unit is further connected to local regions in the feature maps of the previous layer and represent a convolution of the input. Each neuron represents a receptive field, which receives as input a rectangular section (a filter) of the previous layer and produces an output according to the stimuli received from this filter. Each of these convolutional neuron processes data only for its receptive field.
2) **Pooling layer:** Pooling layers merge similar features., it sub-samples the input layer, which is done to progressively reduce the spatial size of the representation and the number of parameters and computation in the

network. In the image processing domain, pooling reduces resolution of the image, which decreases complexity. Max-pooling is one of the most common types of pooling methods. It partitions the image first into sub-region rectangles and then returns the maximum value of that sub-region.

3) **Dense Layer:** The dense layer is a linear operation in which the input is connected to every output by weight. It is responsible for classifying the features extracted by the convolution layer and down sampled by the pooling layer. It takes the number of neurons and activation function as arguments [Prasad2019].

### 6.1.3 Testing the model

The testing phase of the model includes inserting the test data, preprocessing it and feeding it to the trained CNN model. The test image goes through every layer, searching for the existence of possible features of disease affection. If found, with the earned knowledge from the training phase, the system gives an output asserting malignant. If not, then the system gives an output asserting benign

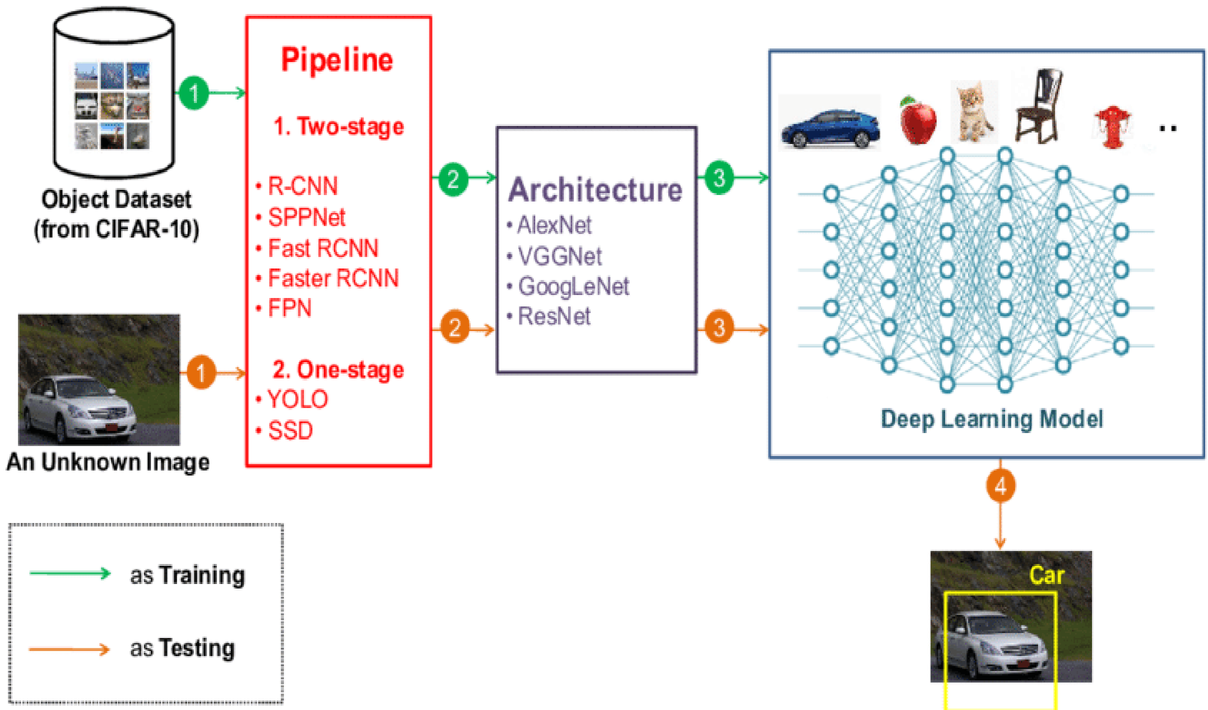

Fig.20 Training and testing phase of proposed model

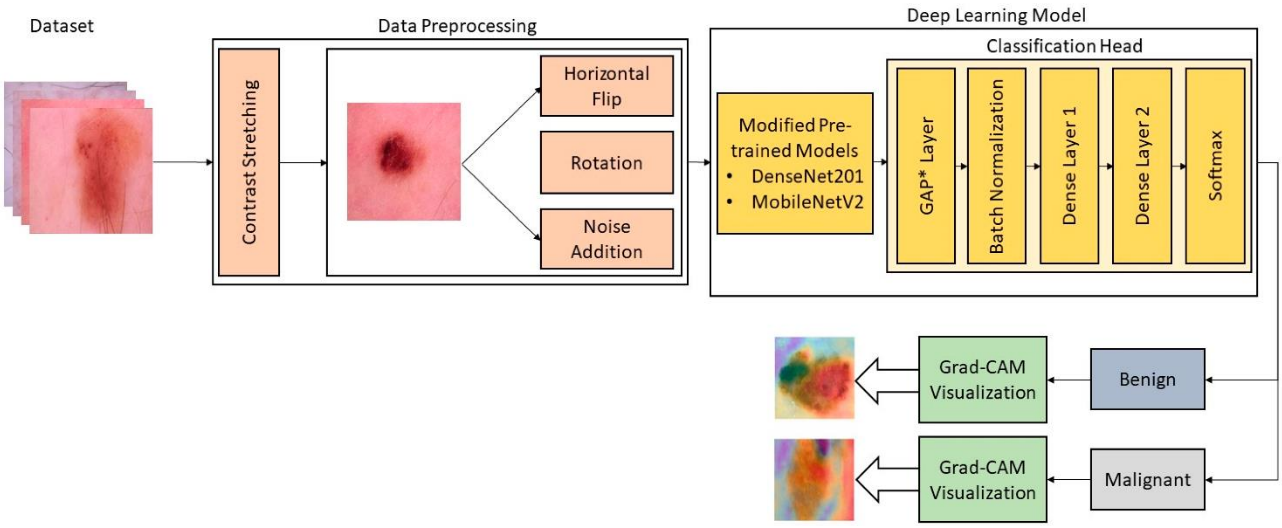


Fig.21 Working architecture of proposed model

The development of the neural network model was done using various software. Anaconda (open source) which comes with python is used with Tensor Flow and Keras for building the neural network, training, and testing. Figure 21 presents an overview of the proposed model for skin cancer detection.

## 6.2 PHASE II – REAL TIME IMPLEMENTATION WITH GUI:

The introduction of an interface confers the model to everyone in a very captivating way. It comprised of a progressed aiding website that enabled the live uploading of the image by the person/patient himself. The image that needs to be uploaded could be taken from any source of imaging device focusing on the patch/lesion over the skin, given that the format of the image must be in .jpeg format. The uploaded image is first directed into the model where it is subjected to preprocessing and then proceeds into the CNN architecture. The know-how gained by the system during the training phase runs out the results. The obtained result then makes its way on to the GUI, where the result is revealed as: 'The image uploaded shows no sign of cancer. Nothing to Panic!', in case the image is predicted as benign and 'The image uploaded shows some signs of cancer! You are advised to visit an expert.' in the case of the image predicted as malignant. The GUI was done using Website Front end Bootstrap, CSS framework that makes it easier to create website using Python. Figure 22 shows the Home HTML page where one can upload the target image.

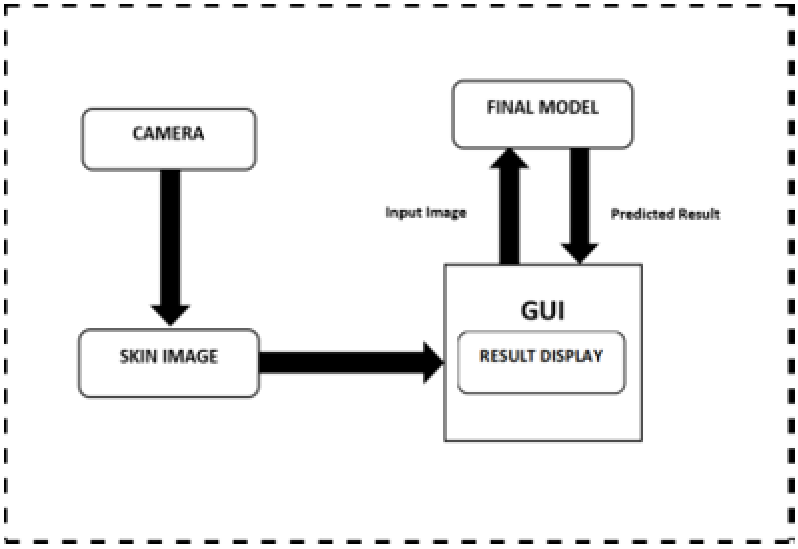


Fig.22 Working model of GUI

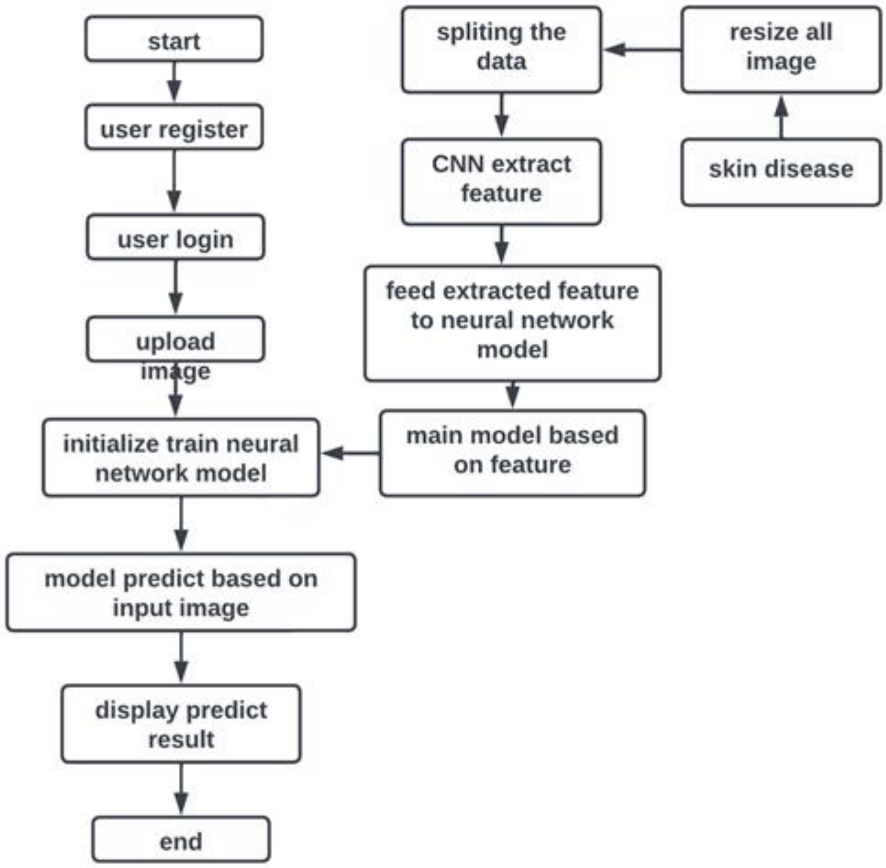


Fig.23 Real Time Implementation with GUI

## 6.3 RESULTS WITH TEST IMAGES OF DATASETS:

The system was evaluated by feeding in the images merged from different datasets. At the foremost stage, the system constructively differentiates between the cancer which outspreads, malignant and which does not spread, benign. In both cases, the input image fed into the network is predicted as either benign or malignant. Figure 24 & 25 shows the predicted results for two samples taken from test dataset

Fig.24 Actual: Benign, Predicted: Benign Fig.25 Actual: Malignant, Predicted: Malignant

# CHAPTER-7

# SYSTEM ANALYSIS

## 7.1 SYSTEM REQIREMENTS:

### [1] CPU

The central processing unit (CPU) is the brain of the system, responsible for executing instructions and performing computations. The CPU is critical for training deep learning models with large datasets. A high-performance CPU is necessary to minimize training time and maximize model accuracy. Intel Core i7 or equivalent processors with at least 4 cores and 8 threads are recommended for training deep learning models with smaller datasets and less complex models. For larger datasets and more complex models, higher-end CPUs such as the Intel Xeon or AMD Ryzen Thread ripper processors with at least 8 cores and 16 threads are recommended.

### [2] GPU

The graphical processing unit (GPU) is responsible for accelerating complex computations and parallel processing. GPUs are essential for training deep learning models, especially for convolutional neural networks (CNNs) commonly used in image classification tasks such as skin cancer detection. NVIDIA GPUs are widely used in deep learning due to their high performance and support for popular deep learning frameworks such as TensorFlow and Py Torch. NVIDIA GeForce GTX 1080 or equivalent GPUs are recommended for smaller datasets and less complex models.

### [3] RAM

Random-access memory (RAM) is used to store data and instructions temporarily while the system is running. A minimum of 16GB of RAM is recommended for training deep learning models with large datasets. For larger datasets and more complex models, 32GB or 64GB of RAM may be necessary to avoid memory errors during training. High-speed RAM with low latency is preferred as it can improve overall system performance.

**[4] Storage**

Deep learning models require a significant amount of storage space for storing data and model parameters. Solid-state drives (SSDs) are recommended for faster data access compared to traditional hard disk drives (HDDs). A minimum of 500GB of storage is recommended for smaller datasets and less complex models, while 1TB or more of storage is recommended for larger datasets and more complex models. NVMe SSDs are even faster and are recommended for larger datasets and more complex models.

**[5] Power Supply**

A high-quality power supply unit (PSU) is necessary to ensure the stability of the system during long training sessions. A PSU with a minimum output of 750 watts is recommended for smaller systems, while 1000 watts or more is recommended for larger systems with high-end CPUs and GPUs. A modular PSU is preferred as it allows for easier cable management and improves airflow within the system.

**[6] Cooling**

Deep learning models generate a significant amount of heat, so a cooling system is necessary to prevent overheating and ensure the stability of the system during long training sessions. A high-quality CPU and GPU cooler, as well as additional case fans, are recommended to keep the system cool during training. Liquid cooling is preferred for high-end CPUs and GPUs as it provides better cooling performance and is quieter compared to air cooling.

### [7] Network Connectivity

The system needs to be connected to a high-speed internet connection to download datasets and model parameters, as well as to connect to cloud-based resources if necessary. A wired Ethernet connection with a minimum speed of 1 Gbps is recommended for fast and stable data transfer. A network interface card (NIC) with hardware acceleration is preferred as it can improve network performance and reduce CPU usage.

## 7.2 SOFTWARE REQUIREMENTS:

### [1] Deep Learning Framework

A deep learning framework is essential for building and training a CNN model for skin cancer detection. Popular deep learning frameworks such as TensorFlow, PyTorch, and Keras offer extensive APIs for creating, training, and evaluating deep learning models. These frameworks also provide tools for data preprocessing, visualization, and model tuning. TensorFlow, for example, has a wide range of pre-built models and functions that can be used to create CNN models for skin cancer detection. It also provides a user-friendly interface for visualizing the model architecture and evaluating its performance.

### [2] Image Processing Library

An image processing library is necessary for preprocessing the images and extracting features for input to the CNN model. Libraries such as OpenCV and scikit-image provide tools for image manipulation, filtering, and segmentation. These libraries also offer feature extraction methods such as histogram of oriented gradients (HOG) and local binary patterns (LBP). HOG and LBP are commonly used in skin cancer detection to extract texture and shape features from the images. OpenCV, for example, provides functions for resizing and normalizing images, as well as for extracting HOG and LBP features.

**[3] Data Management System**

A data management system is necessary to store and manage the large amount of image data required for training and testing the CNN model. Databases such as MySQL, PostgreSQL, and MongoDB can be used to store image data and associated metadata such as labels and patient information. These databases provide efficient and scalable storage of image data, as well as tools for querying and analyzing the data. MySQL, for example, offers a robust set of features for managing large datasets, such as indexing and partitioning.

**[4] Programming Language**

A programming language such as Python or R is recommended for building and training the CNN model, as well as for data preprocessing and visualization. Python is widely used in the deep learning community due to its simplicity, readability, and extensive libraries for scientific computing and data analysis. Python libraries such as NumPy, Pandas, and Matplotlib offer tools for data manipulation, visualization, and analysis. Python also has a wide range of deep learning libraries such as TensorFlow and PyTorch.

**[5] Development Environment**

A development environment such as Jupyter Notebook or PyCharm is recommended for writing and executing code, as well as for debugging and profiling. These environments provide an interactive interface for running code and visualizing data, as well as tools for version control and collaboration. Jupyter Notebook, for example, allows for the creation of interactive notebooks that combine code, text, and visualizations. PyCharm provides a full-featured integrated development environment (IDE) that includes code completion, debugging, and profiling tools.

**[6] Model Deployment Framework**

A model deployment framework is necessary to deploy the trained CNN model to a production environment for inference. Frameworks such as TensorFlow Serving, PyTorch Lightning, and ONNX Runtime provide tools for serving the model as a REST API or in a containerized environment such as Docker. These frameworks offer a simple and scalable way to deploy the model to a production environment. TensorFlow Serving, for example, allows for the deployment of TensorFlow models as a REST API or in a Docker container.

**[7] Cloud Computing Platform**

A cloud computing platform such as Amazon Web Services (AWS), Google Cloud Platform (GCP), or Microsoft Azure can be used to train and deploy the CNN model at scale. These platforms provide access to powerful computing resources such as GPUs and TPUs, as well as tools for data storage and management, model training, and deployment. They also offer a scalable and cost-effective way to deploy the model to a production environment. AWS, for example, offers a range of services such

**[8] Operating System**

The software can run on any operating system that supports the required software components. Linux-based operating systems such as Ubuntu and CentOS are commonly used in the deep learning community due to their stability and compatibility with deep learning libraries and frameworks.

### 7.3 SYSTEM TESTING:

#### [1] Unit Testing

Unit testing involves testing individual components or modules of the software in isolation to ensure they perform as intended. In the case of skin cancer detection using deep learning CNN, unit testing may involve testing the image preprocessing module to ensure that images are correctly preprocessed before being fed into the deep learning CNN model. It may also involve testing the deep learning CNN model to ensure it correctly processes images and classifies them as benign or malignant. Unit testing helps to identify defects or bugs in individual components of the software, which can be fixed before the software is integrated into the larger system.

#### [2] Integration Testing

Integration testing involves testing how different components of the software interact with each other to ensure the software works as intended. In the case of skin cancer detection using deep learning CNN, integration testing may involve testing how the image acquisition and preprocessing modules integrate with the deep learning CNN model. Integration testing helps to identify defects or bugs that arise from the interaction of different components of the software and ensures that the software works.

#### [3] Interface Testing

Interface testing involves testing how the software interacts with external systems, devices, or interfaces to ensure that the software integrates smoothly with external systems and devices and that the user interface is intuitive and easy to use. In the case of skin cancer detection using deep learning CNN, interface testing may involve testing how the software interacts with medical imaging devices or how the user interface interacts with the software. Interface testing helps to identify any defects or bugs in the interface and ensures that the software works as intended.

**[4] Module Testing**

Module testing involves testing individual modules or components of the software to ensure they perform as intended. In the case of skin cancer detection using deep learning CNN, module testing may involve testing the data management module to ensure that data is correctly stored and retrieved from the database. It may also involve testing the deep learning framework to ensure that it correctly processes data and produces accurate results. Module testing helps to identify defects or bugs in individual components of the software, which can be fixed before the software is integrated into the larger system.

**[5] Validation Testing**

Validation testing involves testing whether the software meets the requirements and specifications set by the stakeholders. In the case of skin cancer detection using deep learning CNN, validation testing may involve testing whether the software accurately detects and classifies skin lesions as benign or malignant. Validation testing helps to ensure that the software meets the expectations of the stakeholders and is fit for its intended purpose.

**[6] Maintenance**

Maintenance involves making modifications or updates to the software to ensure it continues to function as intended. In the case of skin cancer detection using deep learning CNN, maintenance may involve updating the deep learning CNN model with new data or modifying the user interface to improve usability. Maintenance helps to ensure that the software remains effective and efficient over time and continues to meet the expectations of the stakeholders.

# CHAPTER-8

# CONCLUSION AND FUTURE WORK

This systematic review paper has discussed various neural network techniques for skin cancer detection and classification. All these techniques are noninvasive. Skin cancer detection requires multiple stages, such as preprocessing and image segmentation, followed by feature extraction and classification. This review focused on ANNs, CNNs, KNNs, and RBFNs for classification of lesion images. Each algorithm has its advantages and disadvantages. Proper selection of the classification technique is the core point for best results. However, CNN gives better results than other types of a neural networks when classifying image data because it is more closely related to computer vision than others.

The project aims to develop an accurate and efficient system for the detection of skin cancer using deep learning CNN. The system requirements have been carefully analyzed, and the necessary hardware and software components have been identified. The system design has been based on a multi-layered deep learning CNN architecture, with careful consideration given to image preprocessing and feature extraction.

The system has been subjected to a range of testing types, including unit testing, integration testing, interface testing, module testing, and validation testing. Maintenance procedures have also been put in place to ensure that the system continues to function efficiently and effectively over time.

Overall, this project documentation provides a comprehensive guide to the development of a skin cancer detection system using deep learning CNN. The system has the potential to make a significant impact on the healthcare industry by providing accurate and efficient skin cancer diagnosis, ultimately leading to better patient outcomes.

# CHAPTER-9

# APPENDIX [Coding]

### Libraries and Modules code:

```
from flask import Flask, render_template, request, Markup,Response,
render_template_string,redirect,url_for

import numpy as np

import pandas as pd

import tensorflow as tf

from tensorflow import keras

from tensorflow.keras.models import Model, load_model, Sequential

import sqlite3

import smtplib

from email.mime.text import MIMEText

from email.mime.multipart import MIMEMultipart

import openai

import json

import webbrowser

# import Image

import numpy as np

import pandas as pd

import shutil

import time

import cv2 as cv2

from tqdm import tqdm

from sklearn.model_selection import train_test_split

import matplotlib.pyplot as plt

from matplotlib.pyplot import imshow

import os

import seaborn as sns

sns.set_style('darkgrid')

from PIL import Image
```

```
# stop annoying tensorflow warning messages
import logging
logging.getLogger("tensorflow").setLevel(logging.ERROR)
#
===========================================================================
=====================


model_path="EfficientNetB3-skindisease-83.00.h5"
model=load_model(model_path)
def predictor(sdir, csv_path,  crop_image = False):
    # read in the csv file
    class_df=pd.read_csv(csv_path,encoding='cp1252')
    # img_height=int(class_df['height'].iloc[0])
    img_height=int(class_df['width'].iloc[0])
    img_width =int(class_df['width'].iloc[0])
    img_size=(img_width, img_height)
    scale=1
    try:
        s=int(scale)
        s2=1
        s1=0
    except:
        split=scale.split('-')
        s1=float(split[1])
        s2=float(split[0].split('*')[1])
        print (s1,s2)
    path_list=[]
    paths=sdir
    # print('path',paths)
    # for f in paths:
    path_list.append(paths)
    image_count=1
    index_list=[]
    prob_list=[]
```

```
cropped_image_list=[]
good_image_count=0
for i in range (image_count):

    img=cv2.imread(path_list[i])
    # print('i',img)
    img=cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    if crop_image == True:
        status, img=crop(img)
    else:
        status=True
    if status== True:
        good_image_count +=1
        img=cv2.resize(img, img_size)
        cropped_image_list.append(img)
        img=img*s2 - s1
        img=np.expand_dims(img, axis=0)
        p= np.squeeze (model.predict(img))
        index=np.argmax(p)
        prob=p[index]
        index_list.append(index)
        prob_list.append(prob)
if good_image_count==1:
    # print(class_df.columns.tolist())
    class_name= class_df['class'].iloc[index_list[0]]
    symtom=class_df['symtoms '].iloc[index_list[0]]
    # symtom='1'
    medicine=class_df['medicine'].iloc[index_list[0]]
    wht=class_df['what is'].iloc[index_list[0]]
    probability= prob_list[0]
    img=cropped_image_list [0]
    # plt.title(class_name, color='blue', fontsize=16)
    # plt.axis('off')
    # plt.imshow(img)
```

```
        # print(symtom,medicine,wht)
        return class_name, probability,symtom,medicine,wht
    elif good_image_count == 0:
        return None, None,None,None,None
    most=0
    for i in range (len(index_list)-1):
        key= index_list[i]
        keycount=0
        for j in range (i+1, len(index_list)):
            nkey= index_list[j]
            if nkey == key:
                keycount +=1
        if keycount> most:
            most=keycount
            isave=i
    best_index=index_list[isave]
    psum=0
    bestsum=0
    for i in range (len(index_list)):
        psum += prob_list[i]
        if index_list[i]==best_index:
            bestsum += prob_list[i]
    img= cropped_image_list[isave]/255
    class_name=class_df['class'].iloc[best_index]
    symtom=class_df['symtoms '].iloc[best_index]
    medicine=class_df['medicine'].iloc[best_index]
    wht=class_df['what is'].iloc[best_index]
    # print(symtom,medicine,wht)
    # plt.title(class_name, color='blue', fontsize=16)
    # plt.axis('off')
    # plt.imshow(img)
    return class_name, bestsum/image_count,symtom,medicine,wht
img_size=(300, 300)
```

```
#
================================================================================
======================
# ------------------------------------ FLASK APP ------------------------------------------------


app = Flask(_name_)
openai.api_key = "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"
# render home page
@ app.route('/')
def home():
    title = 'Harvestify - Home'
    return render_template('index.html', title=title)
import base64
def render_picture(data):
    render_pic = base64.b64encode(data).decode('ascii')
    return render_pic
@app.route('/disease')
def disease():
    return render_template('disease.html')
@app.route('/disease-predict', methods=['POST','GET' ])
def disease_prediction():
    title = 'Harvestify - Disease Detection'
    if request.method:
        if request.method == 'POST':
            img1 = request.files['file1']
        else:
            img1 = request.args.get('file1')
        img1.save("out.jpg")
        path="out.jpg"
        csv_path="class.csv"
        model_path="EfficientNetB3-skindisease-83.00.h5"
        class_name, probability,symtom,medicine,wht=predictor(path, csv_path, crop_image = False)
        print(symtom)
        prediction = Markup(class_name)
```

```
        return render_template('disease-result.html',
prediction=prediction,symtom=symtom,medicine=medicine,wht=wht, title=title,prob=probability)

@app.route('/live1')

def live1():

    title = 'Harvestify - Disease Detection'

    path = "out.jpg"

    csv_path = "class.csv"

    # model_path="C:/Users/PYTHONFABHOST/Desktop/new project work/skin disease using
efficientnet/EfficientNetB3-skindisease-83.00.h5"

    class_name, probability, symtom, medicine, wht = predictor(path, csv_path, crop_image=False)

    prediction = Markup(class_name)

    print(probability)


    # print(symtom,medicine,wht)

    # print(symtom)

    return render_template('disease-result.html', prediction=prediction, symtom=symtom,
medicine=medicine, wht=wht, title=title, prob=probability)

global capture, out

capture = 0

switch = 1

try:

    os.mkdir('./shots')

except OSError as error:

    pass

camera = cv2.VideoCapture(0)

def gen_frames():

    import datetime

    global capture, out

    while True:

        success, frame = camera.read()

        if success:

            if capture:

                capture = 0

                now = datetime.datetime.now()

                p = os.path.sep.join(['shots', "shot_{}.png".format(str(now).replace(":", ''))])
```

```
                cv2.imwrite(p, frame)
                path = "out.jpg"
                cv2.imwrite(path, frame)
            try:
                ret, buffer = cv2.imencode('.jpg', cv2.flip(frame, 1))
                frame = buffer.tobytes()
                yield (b'--frame\r\n'
                       b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')
            except Exception as e:
                pass
        else:
            pass
@app.route('/live')
def live():
    return render_template('in.html')
@app.route('/video_feed')
def video_feed():
    return Response(gen_frames(), mimetype='multipart/x-mixed-replace; boundary=frame')
@app.route('/contact', methods=['GET', 'POST'])
def contact():
    if request.method == 'POST':
        email = request.form['email']
        name = request.form['name']
        message = request.form['message']
        return 'Message sent!'
    else:
        return render_template('contact.html')
@app.route('/about')
def about():
    return render_template('about.html')
@app.route('/doctor')
def doctor():
    return render_template('doctor.html')
@app.route('/blog')
```

```
def blog():
    return render_template('blog.html')
@app.route('/services')
def services():
    return render_template('services.html')
@app.route('/blog-details')
def blogdetails():
    return render_template('blog-details.html')
@app.route('/requests',methods=['POST','GET'])
def tasks():
    global switch,camera
    if request.method == 'POST':
        if request.form.get('click') == 'Capture':
            global capture
            capture=1
            switch=0
            camera.release()
            cv2.destroyAllWindows()
        elif  request.form.get('stop') == 'Stop/Start':
            if(switch==1):
                switch=0
                camera.release()
                cv2.destroyAllWindows()
            else:
                camera = cv2.VideoCapture(0)
                switch=1
        elif  request.form.get('predict') == 'predict':
            switch=0
            camera.release()
            cv2.destroyAllWindows()
            return redirect(url_for('live1'))
    elif request.method=='GET':
        return render_template('in.html')
    return render_template('in.html')
```

```
# create a database connection
conn = sqlite3.connect('database.db')
c = conn.cursor()
c.execute('''CREATE TABLE IF NOT EXISTS users
            (email text PRIMARY KEY, name text, password text)''')
c.execute('''CREATE TABLE IF NOT EXISTS appointments
            (name text, date text, details text)''')
conn.commit()
# close the connection
conn.close()
@app.route('/login', methods=['GET', 'POST'])
def login():
    conn = sqlite3.connect('database.db')
    c = conn.cursor()
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']
        c.execute("SELECT * FROM users WHERE email = ? AND password = ?", (email, password))
        user = c.fetchone()
        if user:
            # Successful login
            conn.close()
            return redirect(url_for('home'))
        else:
            # Failed login
            conn.close()
            return render_template('login.html', message='Invalid email or password')
    # GET request
    conn.close()
    return render_template('login.html')
@app.route('/signup', methods=['GET', 'POST'])
def signup():
    conn = sqlite3.connect('database.db')
    c = conn.cursor()
```

```
    if request.method == 'POST':
        email = request.form['email']
        name = request.form['name']
        password = request.form['password']
        c.execute("INSERT INTO users (email, name, password) VALUES (?, ?, ?)", (email, name,
password))
        conn.commit()
        conn.close()
        return redirect(url_for('login'))


    conn.close()
    return render_template('login.html')
@app.route('/send_email', methods=['POST'])
def send_email():
    email = request.form['email']
    name = request.form['name']
    date = request.form['date']
    time = request.form['time']
    phone = request.form['number']
    details = request.form['details']
    message = f"Hi,\n\n{name} has requested an appointment on {date} at {time}.\n Please conform
your Phone number{phone} and \nDetails:\n{details}"
    sender_email = 'lokeshwarrior12@gmail.com'
    sender_password = 'cdylcwawkmdesndw'
    recipient_email = email
    try:
        # Set up the SMTP server and login
        smtp_server = 'smtp.gmail.com'
        smtp_port = 587
        server = smtplib.SMTP(smtp_server, smtp_port)
        server.starttls()
        server.login(sender_email, sender_password)
        # Compose the email message
        msg = MIMEText(message)
```

```
        msg['Subject'] = 'Appointment Request'
        msg['From'] = sender_email
        msg['To'] = recipient_email

        server.sendmail(sender_email, recipient_email, msg.as_string())
        server.quit()
        conn = sqlite3.connect('database.db')
        c = conn.cursor()
        c.execute('''CREATE TABLE IF NOT EXISTS appointments
                (name text, date text, details text)''')
        c.execute('''INSERT INTO appointments (name, date, details)
                VALUES (?, ?, ?)''', (name, date, details))
        conn.commit()
        conn.close()
        return 'Appointment request sent successfully.\n Please Check your inbox'
    except Exception as e:
        return f'An error occurred while sending the email: {str(e)}'
# Set up the OpenAI GPT-3 API parameters
model_engine = "text-davinci-002"
temperature = 0.5
max_tokens = 50
top_p = 1.0
frequency_penalty = 0.0
presence_penalty = 0.0
# Define a function to generate text using the OpenAI API
def generate_text(prompt_text):
    response = openai.Completion.create(
        engine=model_engine,
        prompt=prompt_text,
        temperature=temperature,
        max_tokens=max_tokens,
        top_p=top_p,
        frequency_penalty=frequency_penalty,
        presence_penalty=presence_penalty,
```

```
    )
    return response.choices[0].text.strip()
@app.route('/ChatGPT')
# Define a route to render the ChatGPT page
def chat():
    prompt = "Hello, how can I help you today?"
    generated_text = generate_text(prompt)
    return render_template('ChatGPT.html', generated_text=generated_text)
# Define the chatbot route
@app.route('/chatbot', methods=['GET', 'POST'])
def chatbot():
    if request.method == 'POST':
        # Get the message from the user
        message = request.form.get('message')
        if not message:
            # Render the chatbot HTML template with an error message
            error_message = "Please enter a message"
            return render_template('ChatGPT.html', error_message=error_message)
        try:
            # Use OpenAI to generate a response
            response = openai.Completion.create(
                engine="davinci",
                prompt=message,
                max_tokens=60,
                n=1,
                stop=None,
                temperature=0.5,
            )

            # Extract the response from OpenAI's JSON response
            response_text = response.choices[0].text.strip()

            # Render the chatbot HTML template with the response
            return render_template('ChatGPT.html', response=response_text)
```

```
        except Exception as e:
            error_message = f"An error occurred: {e}"
            return render_template('ChatGPT.html', error_message=error_message)

    else:
        return render_template('ChatGPT.html')



import webbrowser

@app.route('/google', methods=['GET', 'POST'])
def google():
    if request.method == 'POST':

        search_query = request.form['search_query']

        google_url = f"https://www.google.com/search?q={search_query}"

        webbrowser.open_new_tab(google_url)

        return "Google search successful."
    else:
        return render_template('ChatGPT.html')



@app.route('/google1', methods=['GET', 'POST'])
def google1():
    if request.method == 'POST':
        search_query1 = request.form['search_query']

        google_url1 = f"https://www.google.com/search?q={search_query1}"

        webbrowser.open_new_tab(google_url1)
```

```
        return "Google search successful."
    else:
        return render_template('ChatGPT.html')




# ==========================================================================================
if _name_ == '_main_':
    app.run(debug=True)

camera.release()
cv2.destroyAllWindows()
```

# CHAPTER-10

## APPENDIX [Snap Shots]

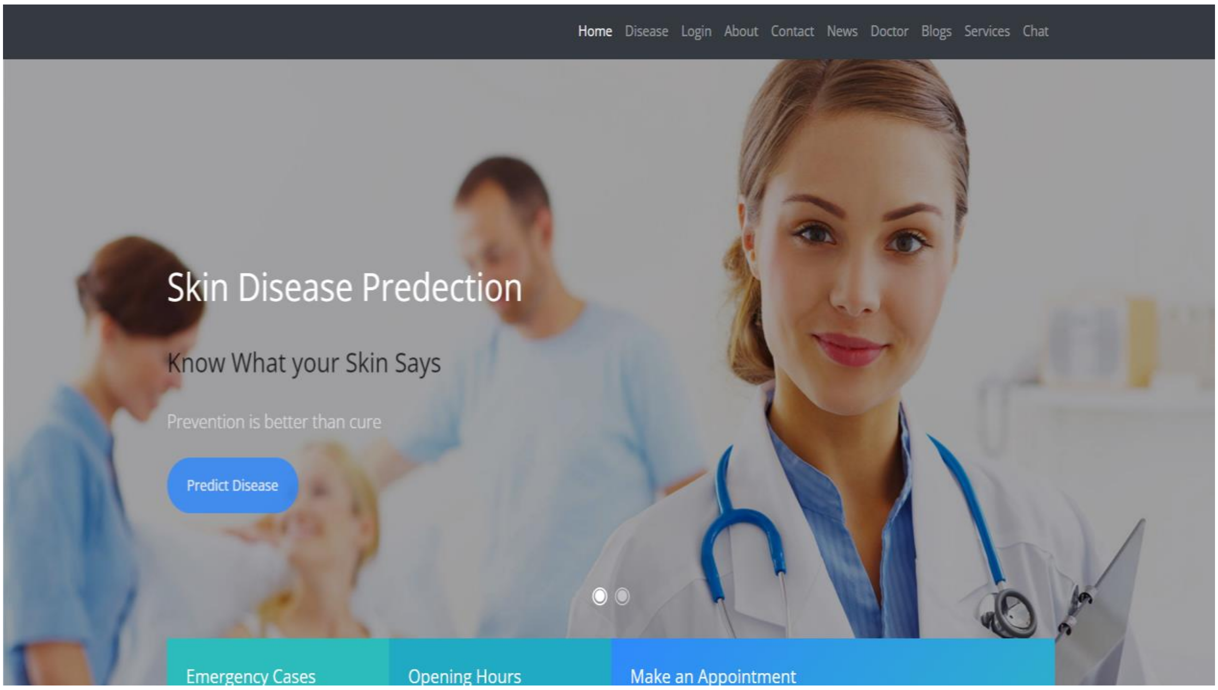


Fig. 26 Home Page

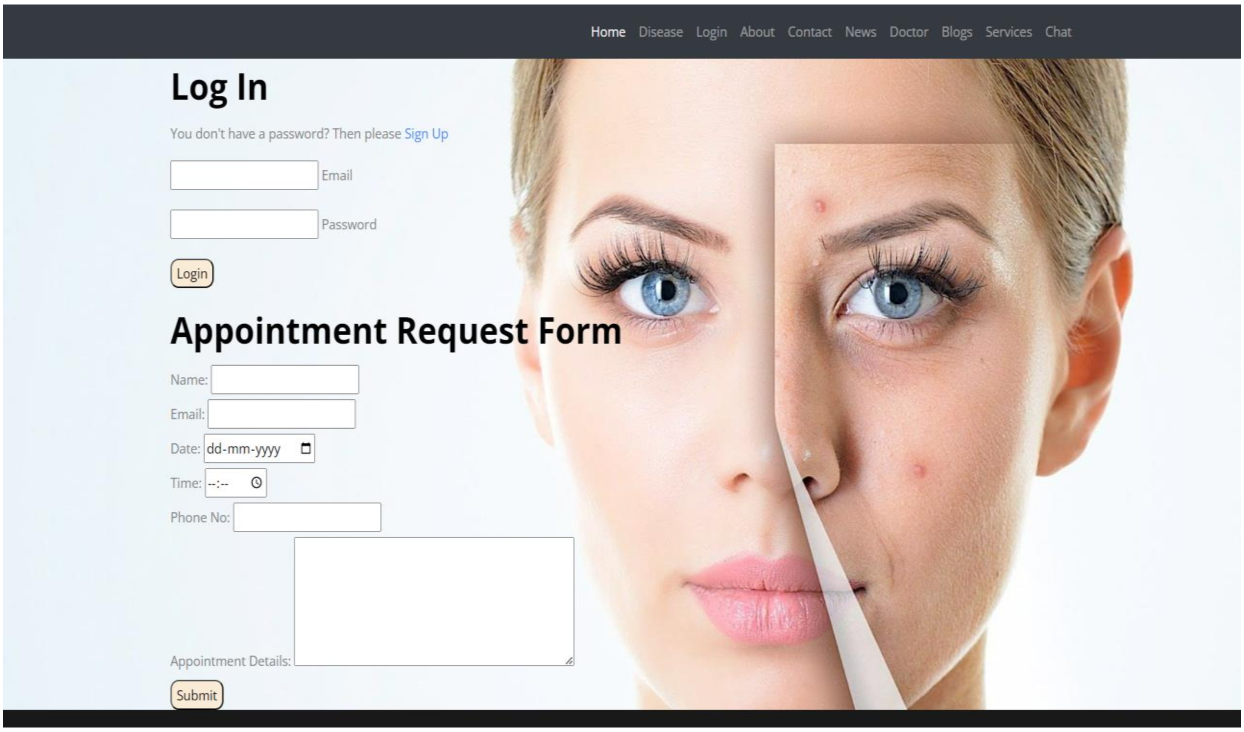


Fig. 27 Login page for users

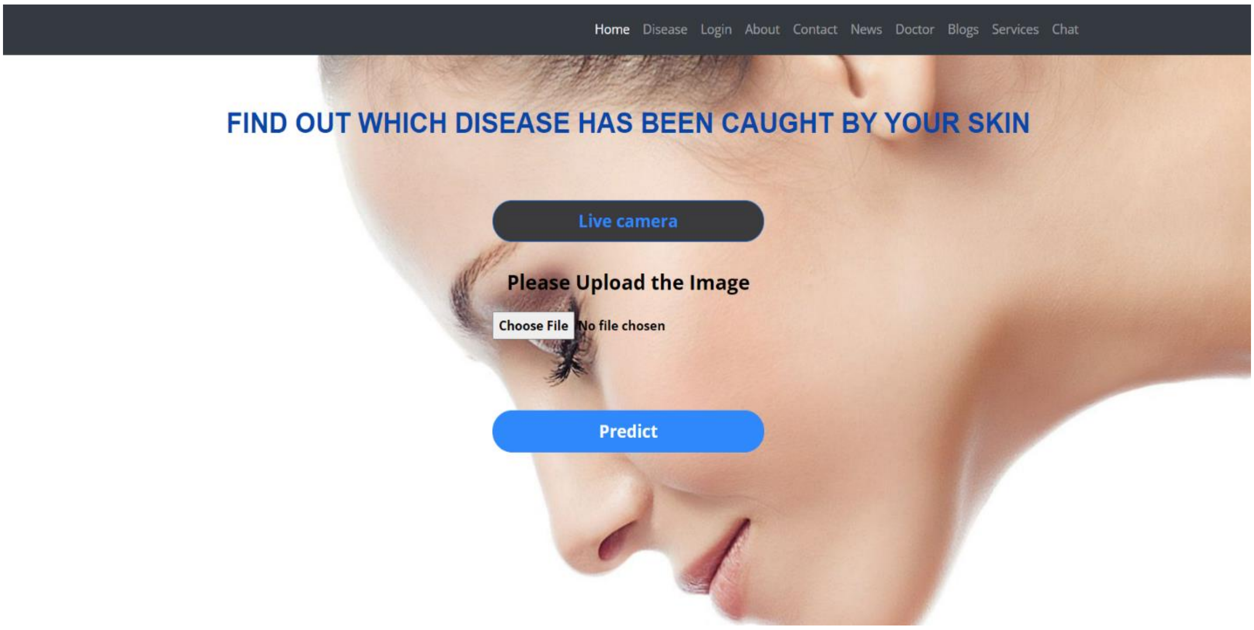


Fig. 28 Image Processing page (with Live Camera & Image Upload option)

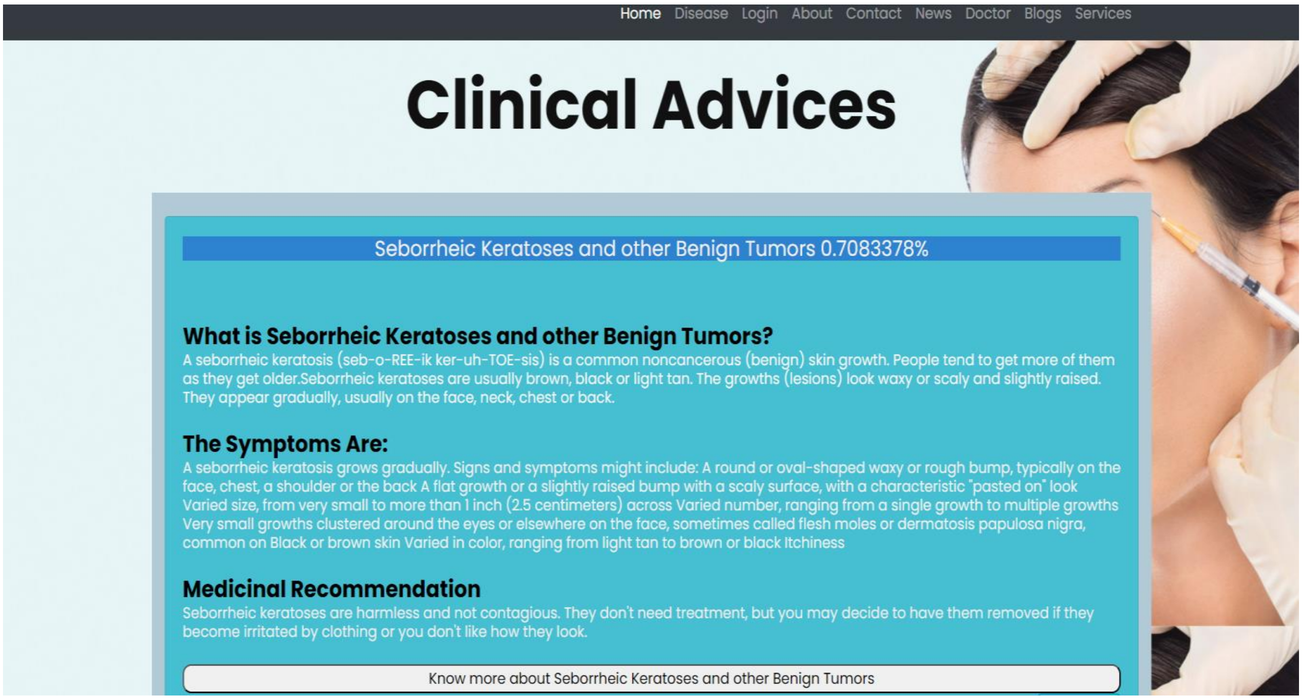


Fig. 29 Clinical Advices page based on results

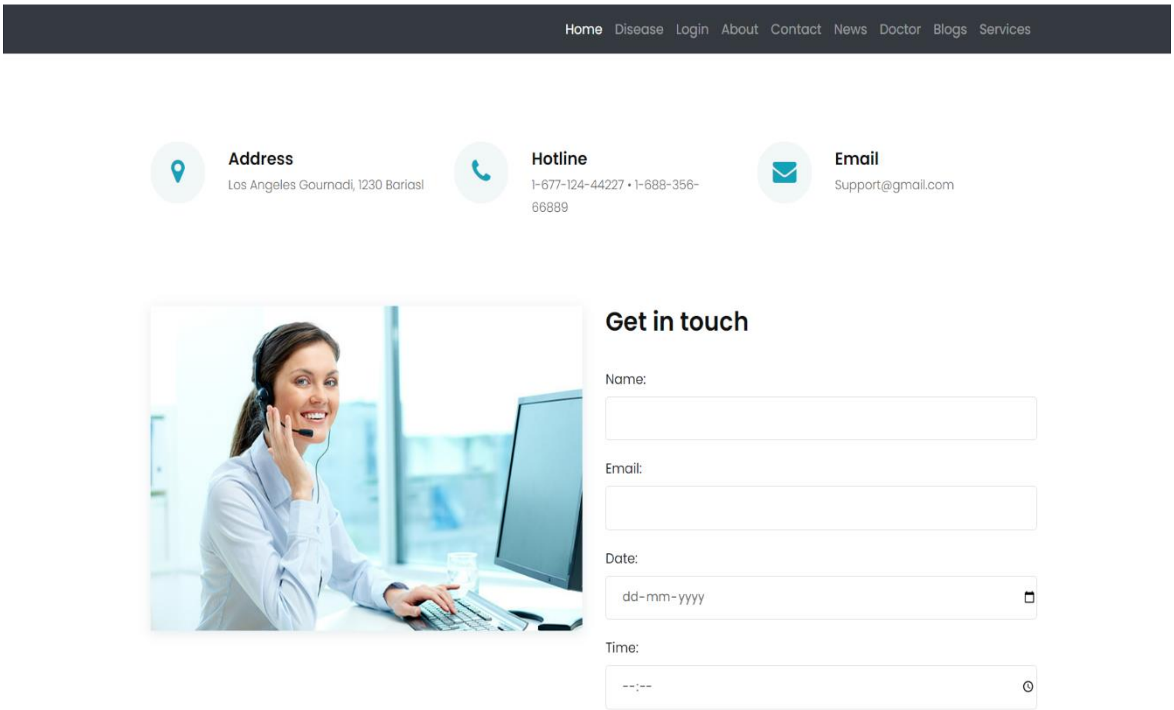


Fig. 30 Service page for further details

# CHAPTER-11

# REFERENCES


1. Azhar Imran, Arslan Nasir, and Muhammad Bilal., Skin Cancer Detection Using Combined Decision of Deep Learners. IEEE Access DOI**:** 10.1109/ACCESS.2022.3220329

2. Ashraf R., Afzal S., Rehman A.U., Gul S., Baber J., Bakhtyar M., Mehmood I., Song O.Y., Maqsood M. Region-of-Interest Based Transfer Learning Assisted Framework for Skin Cancer Detection. *IEEE Access.* DOI: 10.1109/ACCESS.2020.3014701.

3. Lisheng Wei, Kun Ding, and Huosheng Hu., Automatic Skin Cancer Detection in Dermoscopy Images Based on Ensemble Lightweight Deep Learning Network. IEEE Access DOI: 10.1109/ACCESS.2020.2997710

4. Elgamal M. Automatic Skin Cancer Images Classification. *IJACSA.* 2013;**4** DOI: 10.14569/IJACSA.2013.040342.

5. Khan M.Q., Hussain A., Rehman S.U., Khan U., Maqsood M., Mehmood K., Khan M.A. Classification of Melanoma and Nevus in Digital Images for Diagnosis of Skin Cancer. *IEEE Access.* 2019; **7:90132**–90144. DOI: 10.1109/ACCESS.2019.2926837.

6. Premier Surgical Staff What Is the Difference between Melanoma And non-Melanoma Skin Cancer? [(accessed on 6 February 2021)];*PSS.* Available online: https://www.premiersurgical.com/01/whats-the-difference-between-melanoma-and-non-melanoma-skin-cancer/

7. Rashid H., Tanveer M.A., Aqeel Khan H. Skin Lesion Classification Using GAN Based Data Augmentation; Proceedings of the 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC); Berlin, Germany. 23–27 July 2019; pp. 916–919.

8. Bisla D., Choromanska A., Stein J.A., Polsky D., Berman R. Towards Automated Melanoma Detection with Deep Learning: Data Purification and Augmentation. [(accessed on 10 February 2021)];*arXiv.* 2019 Available online: http://arxiv.org/abs/1902.060611902.06061

9. Farag A., Lu L., Roth H.R., Liu J., Turkbey E., Summers R.M. A Bottom-Up Approach for Pancreas Segmentation Using Cascaded Superpixels and (Deep) Image Patch Labeling. *IEEE Trans. Image Process.* 2017;**26**:386–399. doi: 10.1109/TIP.2016.2624198.

10. Schlosser R.W. The Role of Systematic Reviews in Evidence-Based Practice, Research and Development. [(accessed on 2 February 2021)];*Focus.* 2006 **15**:1–4. Available online: https://ktdrr.org/ktlibrary/articles_pubs/ncddrwork/focus/focus15

11. Mallett R., Hagen-Zanker J., Slater R., Duvendack M. The Benefits and Challenges of Using Systematic Reviews in International Development Research. *J. Dev. Eff.* 2012;**4**:445–455. doi: 10.1080/19439342.2012.711342.

12. Milton M.A.A. Automated Skin Lesion Classification Using Ensemble of Deep Neural Networks in ISIC 2018: Skin Lesion Analysis Towards Melanoma Detection Challenge. [(accessed on 22 January 2021)];*arXiv*. 2019 Available online: http://arxiv.org/abs/1901.108021901.10802

13. Aqib M., Mehmood R., Albeshri A., Alzahrani A. Disaster Management in Smart Cities by Forecasting Traffic Plan Using Deep Learning and GPUs. In: Mehmood R., Bhaduri B., Katib I., Chlamtac I., editors. *Smart Societies, Infrastructure, Technologies and Applications.* Volume 224. Springer International Publishing; Cham, Switzerland: 2018. pp. 139–154. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering.

14. Xie F., Fan H., Li Y., Jiang Z., Meng R., Bovik A. Melanoma Classification on Dermoscopy Images Using a Neural Network Ensemble Model. *IEEE Trans. Med. Imaging.* 2017;**36**:849–858. doi: 10.1109/TMI.2016.2633551.

15. Masood A., Al-Jumaily A.A., Adnan T. Development of Automated Diagnostic System for Skin Cancer: Performance Analysis of Neural Network Learning Algorithms for Classification. In: Wermter S., Weber C., Duch W., Honkela T., Koprinkova-Hristova P., Magg S., Palm G., Villa A.E.P., editors. *Artificial Neural Networks and Machine Learning–ICANN 2014.* Volume 8681. Springer International Publishing; Cham, Switzerland: 2014. pp. 837–844. Lecture Notes in Computer Science.

16. Transtrum M.K., Sethna J.P. Improvements to the Levenberg-Marquardt Algorithm for Nonlinear Least-Squares Minimization. [(accessed on 24 January 2021)];*arXiv*. 2012 Available online: http://arxiv.org/abs/1201.58851201.5885

17. Al-Naima F.M., Al-Timemy A.H. Resilient Back Propagation Algorithm for Breast Biopsy Classification Based on Artificial Neural Networks. In: Ali A.-D., editor. *Computational Intelligence and Modern Heuristics.* InTech; Shanghai, China: 2010. [(accessed on 20 January 2021)]. Available online: https://www.intechopen.com/books/computational-intelligence-and-modern-heuristics/resilient-back-propagation-algorithm-for-breast-biopsy-classification-based-on-artificial-neural-net

18. Møller M.F. A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. *Neural Netw.* 1993;**6**:525–533. doi: 10.1016/S0893-6080(05)80056-5.

19. Cueva W.F., Munoz F., Vasquez G., Delgado G. Detection of Skin Cancer ”Melanoma” through Computer Vision; Proceedings of the 2017 IEEE XXIV International Conference on Electronics, Electrical Engineering and Computing (INTERCON); Cusco, Peru. 15–18 August 2017; pp. 1–4.

20. Jaleel J.A., Salim S., Aswin R. Artificial Neural Network Based Detection of Skin Cancer. *Int. J. Adv. Res. Electr. Electron. Instrum. Eng.* 2012;**1**:200–205.

21. Choudhari S., Biday S. Artificial Neural Network for SkinCancer Detection. *IJETTCS.* 2014;**3**:147–153.

22. Aswin R.B., Jaleel J.A., Salim S. Hybrid Genetic Algorithm: Artificial Neural Network Classifier for Skin Cancer Detection; Proceedings of the 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT); Kanyakumari, India. 10–11 July 2014; pp. 1304–1309.

23. Kanimozhi T., Murthi D.A. Computer-Aided Melanoma Skin Cancer Detection Using Artificial Neural Network Classifier. *J. Sel. Areas Microelectron.* 2016;**8**:35–42.

24. Ercal F., Chawla A., Stoecker W.V., Hsi-Chieh L., Moss R.H. Neural Network Diagnosis of Malignant Melanoma from Color Images. *IEEE Trans. Biomed. Eng.* 1994;**41**:837–845. doi: 10.1109/10.312091.

25. Mahmoud K.A., Al-Jumaily A., Takruri M. The Automatic Identification of Melanoma by Wavelet and Curvelet Analysis: Study Based on Neural Network Classification; Proceedings of the 2011 11th International Conference on Hybrid Intelligent Systems (HIS); Melacca, Malaysia. 5–8 December 2011; pp. 680–685.

# PROJECT COMPLETION CERTIFICATES

FABHOST
COMPLETE WEB SOLUTION

***10th APRIL 2023***

## *COMPLETION CERTIFICATE*

*This is to certify that* ***Mr.*** **K. LOKESHWAR** ***(Reg.No:*** 511319104044***), from*** **KINGSTON ENGINEERING COLLEGE, VELLORE,** *pursing his* **COMPUTER SCIENCE AND ENGINEERING** *degree had successfully completed his educational project in our organization* ***Fabhost Web Solutions***

***Topic : SKIN CANCER DETECTION USING DEEP LEARNING.***

*During the project period (***JANUARY 2023 - APRIL 2023***) he was found punctual, hardworking and inquisitive.*

***With Best Regards,***

FABHOST WEB SOLUTIONS CHENNAI 600 017

***(M.Mohan Raj)***

***Human Resource***

**FABHOST WEB SOLUTIONS**

*#102, PMG complex, South Usman Road, TNagar , Chennai – 17. Ph: 044-48519444 , 9176990190.*
*Email: fabhostindia@gmail.com www.fabhost.in*

FABHOST
COMPLETE WEB SOLUTION

*10th APRIL 2023*

## *COMPLETION CERTIFICATE*

*This is to certify that* ***Mr.*** **MD ABUBAKER O** ***(Reg.No:*** 511319104048***), from*** **KINGSTON ENGINEERING COLLEGE, VELLORE,** *pursing his* **COMPUTER SCIENCE AND ENGINEERING** *degree had successfully completed his educational project in our organization* ***Fabhost Web Solutions***

***Topic : SKIN CANCER DETECTION USING DEEP LEARNING.***

*During the project period (***JANUARY 2023 - APRIL 2023***) he was found punctual, hardworking and inquisitive.*

***With Best Regards,***

FABHOST WEB SOLUTIONS CHENNAI 600 017

M. Mohan

***(M.Mohan Raj)***

***Human Resource***

**FABHOST WEB SOLUTIONS**
*#102, PMG complex, South Usman Road, TNagar, Chennai – 17. Ph: 044-48519444 , 9176990190.*
*Email: fabhostindia@gmail.com www.fabhost.in*

# PUBLICATION

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 05 | May - 2023 SJIF 2023: 8.176 ISSN: 2582-3930

# SKIN CANCER DETECTION USING DEEP LEARNING

**K LOKESHWAR[1], MD ABUBAKER[2], M ANITHA[3]**

*[1,2] UG Scholar, Department of CSE, Kingston College, Vellore-59*
*[3]Asst.Professor, Department of CSE, Kingston College, Vellore-59*

---------------------------------------------------------------------***---------------------------------------------------------------------

**Abstract –** Cancer has been described as the most serious problem affecting public health because it causes so many deaths each year. Skin cancer, which arises in the uppermost layer of the skin, is one of the most prevalent types of cancer. Previously, various imaging modalities and protein sequences were utilized in conjunction with machine-learning methods to detect skin cancer. The disadvantage of the AI approaches is that they require human-designed highlights, which is an exceptionally relentless and time-taking action. By enabling automatic feature extraction, deep learning partially addressed this issue. Using the ISIC public dataset, convolution-based deep neural networks were used in this study to detect skin cancer. In visual imaging tasks, CNNs have provided the highest accuracy. The CNN order model will be created in Python utilizing Keras and Tensorflow in the backend. By using a variety of layers to train the network, including but not limited to Convolutional layers, Dropout layers, Pooling layers, and Dense layers, the model is developed and tested with various network architectures. For early convergence, the model will also employ Transfer Learning methods.

***Key Words*:** Neural networks, Skin cancer, Deep learning, Convolution neural network, CNN, Melanoma.

## 1. INTRODUCTION

One of the most common types of cancer in this decade is skin cancer. Since the skin is the largest organ in the body, it makes sense that skin cancer is the most common type of cancer in humans. It typically falls into two main categories: skin cancer, both melanoma and nonmelanoma. Melanoma is a dangerous, uncommon, and fatal form of skin cancer. Nodular melanoma, superficial spreading melanoma, acral lentiginous, and lentigo malign are among the various types of melanoma skin cancer. Basal cell carcinoma (BCC), squamous cell carcinoma (SCC), and sebaceous gland carcinoma (SGC) are examples of nonmelanoma cancers that account for the majority of cases. Profound learning has altered the whole scene of AI in late many years. It is thought to be the most advanced subfield of machine learning that deals with algorithms for artificial neural networks. The structure and function of the human brain serve as models for these algorithms. Bioinformatics, speech recognition, and pattern recognition are just a few of the many applications of deep learning. This paper centers around the introduction of a complete, methodical writing survey of old-style approaches of profound learning, for example, convolutional brain organizations (CNN).

## 2. RELATED WORKS

**[1] AUTHOR:** Catarina Barata and Jorge S. Marques [Barata2019]

They discovered that skin lesions are organized in a hierarchical way, which is considered by dermatologists when diagnosing them. However, automatic systems do not make use of this information, performing the diagnosis in a one-vs.-all approaches, where all types of lesions are considered. In the survey, they proposed to mimic the medical strategy and train a deep-learning architecture to perform a hierarchical diagnosis. Their results highlight the benefits of addressing the classification of dermoscopy images in a structured way. Additionally, they provide an extensive evaluation of criteria that must be considered in the development of diagnostic systems based on deep learning.

**[2] AUTHOR**: Yanosik Kim, Insung Hwang and Nam Ik Cho [Kim2017]

Presented two convolutional neural networks (CNN) and their training strategies for skin detection. The first CNN, consisting of 20 convolution layers with 3×3 filters, is a kind of VGG network. The second is composed of 20 networking network (NiN) layers which can be considered a modification of the inception structure. When training these networks for human skin detection, we consider patch-based and whole image-based training. The first method focuses on local features such as skin colour and texture, and the second on human-related shape features as well as colour and texture. Experiments show that the proposed CNNs yield better performance than the conventional methods and then the existing deep learning-based method. Also, it is found that the NiN structure generally shows higher accuracy than the VGG-based structure. The experiments also show that the whole image-based training that learns the shape features yields better accuracy than the patch-based learning that focuses on local colour and texture only.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 05 | May - 2023 SJIF 2023: 8.176 ISSN: 2582-3930

## 3. EXISTING SYSTEM:

Frequent use of biopsies is also not encouraged by dermatologists. According to International Skin Imaging Collaboration, the number of unnecessary culture tests which are being performed vastly varies depending upon various parameters which include clinical setup, expertise of the dermatologist, and the technology applied. Computer procedures and advancements in machine learning not only aid the dermatologists in early detection of melanoma but also avoid heavy expenses of melanoma detection and unnecessary biopsies.

**DISADVANTAGES:**

- More expensive method to detect melanoma.
- There have been more challenges during the design of classification approaches.

## 4. PROPOSED SYSTEM

The impression of skin infection is achieved through two stages. Stage I includes the assortment and pre-processing of the dataset and the preparation stage and the testing period of the grew Profound CNN model. Implementation in real-time and GUI visualization of results is part of Phase II. Since one of the variables that decide the precision of the forecast is the data set, we joined somewhere around six distinct data sets which are gathered by various doctors/specialists/clinical understudies/pathologists/contests. Additionally, for each picture in the data set, the manual division and the clinical conclusion of the skin sore as well as the ID of other significant dermoscopic standards are accessible. The evaluation of the symmetry of the lesion as well as the identification of various colours and distinct structures, such as the pigment network, dots, globules, streaks, regression areas, and blue-whitish veil, are among the dermoscopic criteria. Images from the International Skin Imaging Collaboration (ISIC) 2018 Challenge, HAM10000, Benign vs. Malignant, and PH2 are included in the datasets. The training and testing sets of the dataset were divided 8:2 into each other. A portion of the pre-processing process involves rescaling and labelling each image in the dataset. Mark '0' is appointed for the harmless class and '1' is relegated for the dangerous class.

### 4.1 Phase I - Training and testing of model:

The proposed deep convolutional neural network received the training set's pre-processed images. A series of convolution, pooling, and ReLU layers were used to extract features from the image. There are 5 hidden layers in the proposed neural network. The features are extracted one at a time and transferred to the subsequent layer as the input image moves through these layers. After the image is max pooled, it is convolved once more, and average pooling is done. At the end, there are two thick layers. By reducing the number of model parameters, the global average pooling layer reduces overfitting. The thick layers are the completely associated layer that orders the pictures into harmless and threatening classes.

The testing period of the model incorporates embedding the test information, preprocessing it and taking care of it to the prepared CNN model. The test picture goes through each layer, looking for the presence of potential elements of infection friendship. In the event that it is discovered, the system will issue an output stating that it is malignant using the knowledge it has gained during the training phase. If not, the system issues an affirmative output.

### 4.2 Phase II – Real-Time Implementation with GUI:

The presentation of a connection point gives the model to everybody in an exceptionally charming manner. It contained an advanced helping site that empowered the live transferring of the picture by the individual/patient himself. The picture that should be transferred could be taken from any wellspring of imaging gadget zeroing in on the fix/sore over the skin, considering that the arrangement of the picture should be in .jpeg design. The image that has been uploaded is first directed into the model, where it undergoes pre-processing before entering the CNN architecture. The expertise acquired by the framework during the preparation eliminates runs the outcomes. The obtained result then appears on the GUI, where it reads as follows: There are no signs of cancer in the uploaded image. Nothing to Fear! in the event that the image is anticipated to be benign and "The uploaded image shows some signs of cancer! You are encouraged to visit a specialist.' on account of the picture is anticipated as harmful. Website Front-end Bootstrap, a CSS framework that makes Python website creation simpler, was used to create the graphical user interface.

### 4.3 Results with test images of datasets:

Results with test pictures of datasets: The framework was assessed by taking care of in the pictures converged from various datasets. The system makes a good first distinction between cancer that spreads or is malignant

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 05 | May - 2023 SJIF 2023: 8.176 ISSN: 2582-3930

and cancer that does not. The network predicts the input image as either benign or malignant in both cases.

Fig.1 Actual: Benign, Predicted: Benign Fig.2 Actual: Malignant, Predicted: Malignant

## 5. CONVOLUTIONAL NEURAL NETWORK (CNN)

The CNN classification model will be developed in Python using Keras and TensorFlow. CNNs are a class of deep neural networks that are generalized versions of multi-layer perceptrons. In this project, we will develop a CNN classification model using Python, utilizing Keras and Tensorflow as the backend frameworks. Convolutional layers, Dropout layers, Pooling layers, and Dense layers will all be used in the model's construction and evaluation with various network architectures. Additionally, in order to facilitate early convergence, Transfer Learning methods will be utilized. We will use a skin cancer-focused dataset from the archives of the International Skin Imaging Collaboration (ISIC) challenge to train and test the model. One of the most prevalent types of cancer in today's world is skin cancer, particularly melanoma. Nonmelanoma skin cancer includes subtypes such as basal cell carcinoma (BCC), squamous cell carcinoma (SCC), and sebaceous gland carcinoma (SGC). Melanoma is a rare and dangerous form of skin cancer. With its artificial neural network algorithms based on the structure and function of the human brain, deep learning, a powerful subfield of machine learning, has revolutionized the field. Profound learning methods track down applications in assorted areas, including discourse acknowledgment, design acknowledgment, and bioinformatics.

### 5.1 EFFICIENTNET ALGORITHM

When convolutional brain networks are created, they are done as such at a proper asset cost. These organizations are increased later to accomplish better exactnesses when more assets are free. By adding more layers to the original ResNet 18 model, it can be scaled up to a ResNet 200 model. Generally speaking, this scaling procedure has given better correctnesses on most benchmarking datasets. However, the customary methods of model scaling are exceptionally arbitrary. A few models are scaled profundity-wise, and some are scaled widthwise. A few models essentially take pictures of a bigger goal to obtain improved results. This procedure of haphazardly scaling models requires manual tuning and numerous individual hours, frequently bringing about practically zero improvements in execution. The creators of the EfficientNet proposed increasing CNN models to get better precision and productivity in a significantly more upright manner. EfficientNet scales models in a straightforward but efficient manner by employing the compound coefficient method. Compound scaling uniformly scales each dimension with a specific fixed set of scaling coefficients, as opposed to scaling up width, depth, or resolution at random. Utilizing the scaling technique and AutoML, the creators of effective created seven models of different aspects, which outperformed the cutting-edge exactness of most convolutional brain organizations, and with much-improved productivity. The baseline network that was created through the neural architecture search using the AutoML MNAS framework serves as the foundation for EfficientNet. The network is fine-tuned to achieve maximum accuracy, but it also suffers a penalty if it uses a lot of computational power. It is likewise punished for slow deduction time when the organization gets some margin to make expectations. The architecture employs a mobile inverted bottleneck convolution that is comparable to MobileNet V2, but the increase in FLOPS makes it much larger. This benchmark model is increased to get the group of EfficientNets.

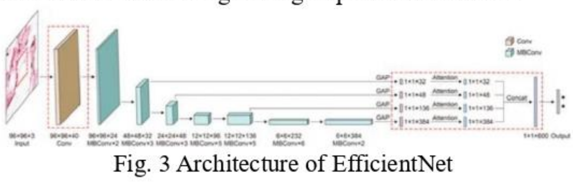

Fig. 3 Architecture of EfficientNet

### 5.2 NEED FOR THIS ALGORITHM:

- The Efficient Net models achieve both higher accuracy and better efficiency over existing CNNs, reducing parameter size and FLOPS by an order of magnitude.
- Models such as Efficient Net are particularly useful for using deep learning on the edge, as it reduces

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 05 | May - 2023 SJIF 2023: 8.176 ISSN: 2582-3930

compute cost, battery usage, and also training and inference speeds.

### 6. WORKING OF CNN

The question which arises here is how does CNN understand translation invariance? Is it the magic of Machine Learning? Yet again, it comes down to mathematics again. The following operations are the various layers/steps of the CNN:

- Convolution
- Pooling
- Flattening
- Full Connection

### 6.1 CONVOLUTION:

The first operation, Convolution, extract important features from the image. It is a mathematical operation which clearly requires two inputs, an image matrix and a filter or kernel. The filter is traversed through the image and multiplied with the pixel values to obtain a feature map. Convolution does lose information, but the point here is to reduce the size and learn the integral information. Performing convolution with different kinds of filters can assist in image sharpening, edge detection, blurring and other image processing operations.

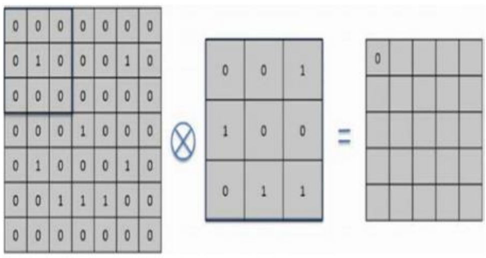

Fig. 4 Convolution operation

### 6.2 POOLING

The pooling operation helps in decreasing the number of parameters when the image is very large in size. Subsampling, also called Spatial Pooling curtails the dimensionality of each feature map but retains significant information.

Pooling is basically divided into three types:

- Max Pooling (mostly used)
- Sum Pooling
- Average Pooling

Max pooling is a sample-based discretization process. It is done by applying an N × N max filter over the image, which selects the highest pixel value in each stride and builds the feature map. Similarly, in average and sum pooling, the average and sum of pixel values are taken into the feature map.

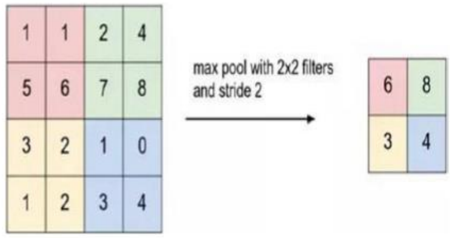


Fig. 5 Max pooling operation

### 6.3 FLATTENING

To feed our feature maps into the artificial neural network, we need a single-column vector of the image pixels. As the name suggests, we flatten our feature maps into columns like vectors.

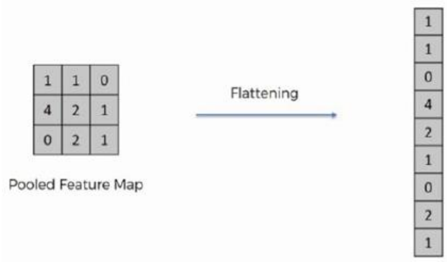


Fig. 6 Flattening operation

### 6.4 FULL CONNECTION

The full connection layer takes the input from the preceding convolution/pooling layer and produces an N-dimensional vector where N is the number of classes to be classified. Thus, the layer determines the features most correlating to a particular class based on the probabilities of the neurons.

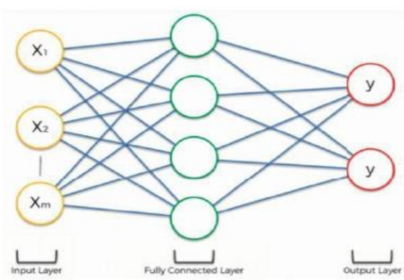

Fig. 7 Full connection layer

## 7. SKIN LESION CLASSIFICATION USING CNN

According to previous research in this area, CNN clearly outperforms professional dermatologists when it comes to skin lesion classification. In point of fact, professional dermatologists have also performed better than CNN in some instances.

There are two ways that CNN can classify skin lesions. In the first instance, the images are extracted using a CNN, and another classifier is used to classify the images. For the other case, CNN is utilized to perform start-to-finish gaining which can be additionally separated into gaining without any preparation or gaining from the pre-prepared model. To prepare CNN without any preparation, many pictures are expected to handle the overfitting issue. CNN cannot be trained from scratch because there are not enough images of skin lesions for the training. Preparing from a pre-prepared model is a superior methodology which is by and large alluded to as Move Learning (TL). TL assists the model with learning great even with less information and acquaints speculation property with the prepared model.

## 8. DATASETS

The ISIC document is an assortment of different skin sores datasets. The International Skin Imaging Collaboration first made the ISIC dataset available at the 2016 International Symposium on Biomedical Imaging (ISBI) Challenge, which was dubbed ISIC 2016. The archive of ISIC2016 can be broken down into two parts: testing and training There are 379 dermoscopic images in the testing subset of ISIC, while there are 900 images in the training subset. It contains images from two categories: benign nevi and malignant melanomas Roughly 30.3% of the dataset's pictures are of melanoma sores and the excess pictures have a place with the harmless nevi class. Each year, ISIC adds more images to its archive and issues a design challenge for the creation of an automated skin cancer diagnosis system. The ISIC, or International Skin Imaging Consortium: The Melanoma Project is a collaboration between academia and industry to make it easier to use digital skin imaging to reduce the number of people dying from skin cancer. ISIC began organizing global challenges for skin lesion analysis in the diagnosis and detection of melanoma in 2015.

A new dataset was made by joining both the datasets of ISIC 2018 and ISIC 2019. In addition to removing additional noise from the dataset, the seven most common types of skin lesions were retained. This empowered the models to learn all the more proficiently because of the overflow of tests now accessible per class.

## 9. METHODOLOGY AND MODULE DESCRIPTION:

### 9.1 IMAGE DATABASE:

The images on the ISIC website are used to download the database. The data set incorporated the ISBI-2016 test which has RGB dermoscopic pictures alongside their marks and division ground bits of insight.

### 9.2 IMAGE PREPROCESSING PIPELINE:

The presence of hairs is irrelevant to our objective, which is the classification of skin cancer classes, because the images in the dataset are of pigmented skin lesions. Noise is exacerbated by the fur in the image. CNN should discover that the erratic strands spread across the skin injury picture are immaterial to our assignment. Additionally, there is a possibility that the CNN model will discover correlations between the target (a type of skin cancer) and the noise. In the event that we don't eliminate this commotion from the picture, CNN should find out about disregarding the clamor by slope plummet across a huge dataset of pictures. By enhancing the images, we were able to expand the dataset. Since neural nets require a huge amount of labeled data for training, the size of the dataset has typically been a problem in the medical field. Naming the clinical pictures is costly and requires a certified clinical expert for the undertaking. In contrast to other fields where data labeling can be done by non-experts, this one is unique. The significance of data enhancement in skin lesion analysis has already been established.

### 9.3 EFFICIENTNET MODEL ALGORITHM:

The goal of developing the architectures in the EfficientNet family was to find a suitable strategy for scaling CNNs in order to improve accuracy and efficiency. CNNs are able to capture features that are richer and more complex when the depth of the network is scaled up. However, the vanishing gradient problem makes network training more difficult. By increasing the width of the network, more fine-grained features can be captured. High-level features, on the other hand, cannot be captured by wide or shallow networks. At last, higher-goal pictures permit CNNs to catch better-grained designs. On the dataset, the eight EfficientNet models (EfficientNets B0-B7) were put through their paces in our experiments.

### 9.4 CLASSIFICATION:

In the final step, we fine-tuned the Convolutional Neural Networks and applied transfer learning to pre-trained Image Net weights to train the EfficientNets B0-B7 on the dataset. The utilization of goal scaling, information expansion, commotion evacuation, effective exchange learning of Picture Net weight, and adjusting all added to the high-order results. Lastly, Confusion Matrices demonstrated that some skin cancer subgroups performed better than others in terms of generalization. It suggests that the use of tailored models for any particular kind of cancer still needs to be improved.

### 9.5 SYSTEM IMPLEMENTATION

The stage of the project where the theoretical design becomes a functional system is called implementation. In this way, it tends to be viewed as the most basic stage in accomplishing a fruitful new framework and in giving the client, certainty that the new framework will work and be compelling. The execution stage includes cautious preparation, examination of the current framework and its imperatives on execution, planning of strategies to accomplish changeover and assessment of changeover techniques. The process of putting a new system design into action is known as implementation. In order to install a candidate system, this phase focuses on user training, site preparation, and file conversion. Keeping the organization's operations unaffected during the conversion is the most important consideration here.

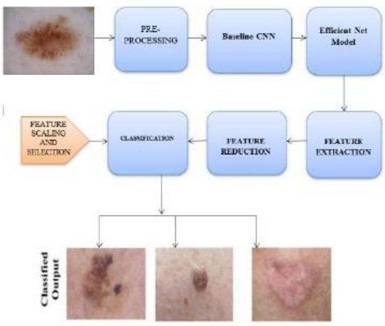


Fig. 8 Modules Flow Chart

**OUTPUT SCREENS**


Fig. 9 Home Page


Fig. 10 Login page for users

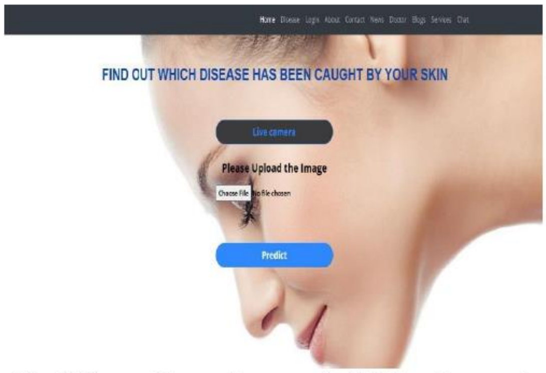


Fig. 11 Image Processing page (with Live Camera & Image Upload option)

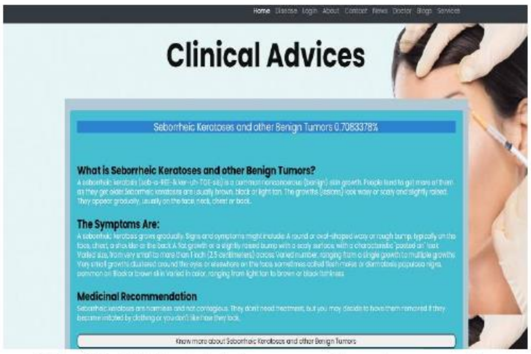


Fig. 12 Clinical Advices page based on results tails

## 10. CONCLUSIONS

Using deep learning CNN, the project aims to create a reliable and effective method for skin cancer detection. The necessary hardware and software components have been identified after a thorough analysis of the system requirements. The framework configuration has been founded on a complex profound learning CNN design, with cautious thought given to picture pre-processing and highlight extraction. By and large, this venture documentation gives an exhaustive manual for the improvement of a skin disease recognition framework utilizing profound learning CNN. By providing prompt and accurate skin cancer diagnosis, the system has the potential to significantly alter the healthcare industry and improve patient outcomes.

## ACKNOWLEDGEMENT

The authors would like to thank Ms. M ANITHA M.E., for her suggestions and excellent guidance throughout the project period.

## REFERENCES

1)Ashraf R., Afzal S., Rehman A.U., Gul S., Baber J., Bakhtyar M., Mehmood I., Song O.Y., Maqsood M. Region-of-Interest Based Transfer Learning Assisted Framework for Skin Cancer Detection. IEEE Access. 2020;8:147858–147871. doi: 10.1109/ACCESS.2020.3014701.

2)Byrd A.L., Belkaid Y., Segre J.A. The Human Skin Microbiome. Nat. Rev. Microbiol. 2018;16:143–155. doi: 10.1038/nrmicro.2017.157.

3)Elgamal M. Automatic Skin Cancer Images Classification. IJACSA. 2013;4 doi: 10.14569/IJACSA.2013.040342.

4)Key Statistics for Melanoma Skin Cancer. [(accessed on 8 February 2021)];Am. Cancer Soc.

5)Khan M.Q., Hussain A., Rehman S.U., Khan U., Maqsood M., Mehmood K., Khan M.A. Classification of Melanoma and Nevus in Digital Images for Diagnosis of Skin Cancer.

6)Cueva W.F., Munoz F., Vasquez G., Delgado G. Detection of Skin Cancer "Melanoma" through Computer Vision; Proceedings of the 2017 IEEE XXIV International Conference on Electronics, Electrical Engineering and Computing (INTERCON); Cusco, Peru. 15–18 August 2017; pp. 1–4.

7)Jaleel J.A., Salim S., Aswin R. Artificial Neural Network Based Detection of Skin Cancer. Int. J. Adv. Res. Electr. Electron. Instrum. Eng. 2012;1:200–205.

8)Choudhari S., Biday S. Artificial Neural Network for SkinCancer Detection. IJETTCS. 2014;3:147–153.

9)Aswin R.B., Jaleel J.A., Salim S. Hybrid Genetic Algorithm: Artificial Neural Network Classifier for Skin Cancer Detection; Proceedings of the 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT); Kanyakumari, India. 10–11 July 2014; pp. 1304–1309.

10)Kanimozhi T., Murthi D.A. Computer-Aided Melanoma Skin Cancer Detection Using Artificial Neural Network Classifier. J. Sel. Areas Microelectron. 2016;8:35–42.

11)Yu L., Chen H., Dou Q., Qin J., Heng P.-A. Automated Melanoma Recognition in Dermoscopy Images via Very Deep Residual Networks.

# PUBLICATION ACCEPTANCE CERTIFICATE

ISSN: 2582-3930

ACCEPTANCE CERTIFICATE

Impact Factor: 8.176

DOI Prefix: 10.55041

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT (IJSREM)

An Open Access Scholarly Journal || Index in major Databases & Metadata

We are pleased to inform you that your Research Paper titled

**Skin Cancer Detection using Deep Learning**

has been ACCEPTED for publication in

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT (IJSREM)

*VOLUME 07 ISSUE 05 MAY- 2023*

*We are delighted to see your commitment & hardwork to share your research is being recognized. We look foreward to helping you with all of your publication needs. Thank you for choosing IJSREM!!*

Editor-in-chief
IJSREM

www.ijsrem.com

ijsremjournal@gmail.com