Comparative Analysis of CNN Architectures for Skin Cancer Detection and Classification

Mukesh Mann, Prince Kumar , Mohit Kukreja and Rakesh P. Badoni

Indian Institute of Information Technology, Sonepat, Haryana – 131029, India & École Centrale School of Engineering, Mahindra University, Hyderabad –500043, India
Cooresponding Author Email- Mukesh.maan@iiitsonepat.ac.in, and Rakesh Badoni (Rakeshbadoni@gmail.com)

Abstract :

Skin cancer diagnosis traditionally requires patients to visit a dermatologist, who then conducts a thorough visual examination of the lesion while also considering the patient's medical history. While this process is effective, it can be time and resource-consuming. In this study, we propose an automated approach to skin cancer classification using machine learning models, including a custom convolutional neural network (CNN) and pre-trained ResNet50, MobileNet, and DenseNet architectures. Using the HAM10000 dataset, we classify skin cancer into seven different categories. Our custom 20-layer CNN model achieved the highest performance with a test accuracy of 84%, while DenseNet and MobileNet achieved 84% and 78% accuracy, respectively. In contrast, the ResNet50 model struggled with generalization on this dataset and showed significantly lower accuracy. These results suggest that a customized CNN model could be a viable solution for more efficient and accurate skin cancer detection.

1. Introduction

Skin cancer and other dermatological conditions often requires time consuming investigation and specialised medical expertise for accurate diagnosis, which is not really available to everyone in need. Traditional methods for detecting skin disease can lack precision, and the reluctance of many individuals to visit a doctor for regular checkups exacerbates the problem. To address this, we harness the power of AI, offering a faster and more accessible solution for early skin disease detection.

Our AI-based system focuses on identifying seven specific skin diseases namely, "Melanocytic Nevi", "Melanoma", "Benign Keratosis like lesion", "Basal Cell Carcinoma", "Actinic Keratoses", "Vascular Lesion", "Dermatofibroma", along with providing actionable guidance to users. While this tool cannot replace a doctor's expertise, it empowers individuals to take proactive steps by offering preliminary insights and recommending professional consultation when necessary.

Prior research has demonstrated that machine learning and deep learning methods, particularly Convolutional Neural Networks (CNN), can achieve impressive accuracy in automatic skin disease detection, with some models reaching up to 100% accuracy. However, many existing models are trained on controlled or unrealistic datasets, which limits their practical use in real-world scenarios.

Our approach differs: we have trained our model using real-world data sourced from Kaggle, ensuring it is more robust and applicable to actual patient cases. By providing users with early-stage identification of these diseases, our solution bridges the gap between routine self-assessment and clinical diagnosis, helping to promote earlier medical intervention and better outcomes for skin health.

1.1 Convolutional Neural Network (CNN) models

CNN models consist of at least three layers of three types, a convolutional layer, a pooling layer, and a fully connected layer (Awati, 2022). In each of these layers, different features of an input image are detected. The output of one layer becomes the input of the next layer. From layer to layer, the CNN becomes more and more complex until it can identify the complete object. This multi-layer network provides more accurate results than single-layer networks. Fig-1 represents a Convolution Layer. Besides these accurate results, especially with a high amount of data, CNN models also have the advantage that the model learns the object's features. This makes manual feature extraction not needed.

2. Related Work

In an early study, Dorj et al. [1] achieved a notable accuracy of 94.2% on a dataset containing approximately 3,700 images across four types of skin cancer. This result was obtained by combining a deep convolutional neural network (DCNN) with an error-correcting output codes support vector machine (ECOC-SVM). To reduce noise, the images in this study were cropped, and features were extracted using a pre-trained model based on AlexNet. The results showed that the model achieved maximum average values for accuracy (95.1% for squamous cell carcinoma), sensitivity (98.9% for actinic keratosis), and specificity (94.17% for squamous cell carcinoma). Classification was performed using an SVM classifier, inspired by Andre E's work [2] on skin cancer detection using deep neural networks (DNNs). In that study, a large dataset of 129,450 images was employed, and the model's performance was compared with that of 21 board-certified dermatologists on biopsy-confirmed clinical images, focusing on two critical binary classifications: keratinocyte carcinomas versus benign seborrheic keratoses, and malignant melanomas versus benign nevi. Their model reportedly achieved accuracy on par with the dermatologists.

Another study by Rezvantalab et al. [3] utilized the widely recognized HAM10000 dataset from Kaggle, which contains 10,015 images spanning eight skin lesion classes and serves as the foundation for our work as well. They applied several state-of-the-art pre-trained architectures, including DenseNet201, ResNet152, Inception v3, and InceptionResNet v2. Their findings showed that all deep learning models outperformed dermatologists by at least 11%, with DenseNet201 yielding the highest macro and micro average AUC values across the classification tasks.

Hosny et al. [4] used a combination of custom AlexNet and

Transfer Learning and achieved an accuracy of 98.6%. The original dataset was augmented to 4,400 images. In addition to fine-tuning and data augmentation, transfer learning was applied to AlexNet by replacing last layer by SoftMax to perform classification over three types of lesions (melanoma, common nevus and atypical nevus). The proposed model was trained and tested using the ph2 dataset. *Lee* et al [5] proposed classification of lesion using Fine-Tuned Neural Networks. Ensemble methods was incorporated to classify seven types of skin disease in the ISIC dataset. A 89.9% accuracy was achieved on the training dataset, which dropped to 78.5% on the validation set. DenseNet was combined with other models for segmentation, existing classifiers defined by *Jamil* et al [6] was fine-tuned. This model is built especially to detect Melanoma, they proposed an approach that can automatically preprocess the image and then segment the lesion. The system then filters the unwanted artifacts including hairs, gel, bubbles, and specular reflection. To deal with the preprocessing and segmentation problems, various algorithms have been proposed that can be classified as thresholding, edge-based and region-based. In [7] thresholding-based segmentation approach is presented. They used an ensemble of different thresholding, along with clustering. Their approach did not proved worthy in case of poor contrast of image. Shete et al. [8] proposed an automated pre-processing step to enhance image quality prior to lesion segmentation. For their study, they used the HAM10000 dataset, which contains dermoscopic images. The diagnosis process involved both image recognition and a deep learning algorithm, aimed at improving the accuracy and efficiency of skin lesion classification.

Pham et al. [9] explored the same HAM10000 provided by Kaggle, and ISIC 2016 having 172 images, containing a single class. They compared the classification results of 6 classifiers in combination with 7 feature extraction methods and 4 data preprocessing steps on the two specified datasets. Their system, consisting of Linear Normalization of the input image as data preprocessing step, HSV as feature extraction method and Balanced Random Forest as classifier, yielded best prediction results on the HAM10000 dataset with 81.46% AUC, 74.75% accuracy, 90.09% sensitivit, while the ground truth for the rest of the cases was either follow-up, expert consensus, or confirmation by in-vivo confocal microscopy. Research conducted by *Noel* et al [10] over a dataset including more than 12,500 images across 3 tasks. They took input from real time 900 users. A test for segmentation algorithm, and a test for algorithm ability to generalize were established. Results stated that top segmentation algorithms still fail on over 10% of images on average to generalise well. *Tschandl* et al [11] utilising the HAM10000, collected dermatoscopic images from different populations acquired by different modalities. More than 50% of the predictions made by the model were confirmed by pathologists, while the ground truth for the rest of the cases was either follow-up, expert consensus, or confirmation by in-vivo confocal microscopy. To prevent selection bias for the test images. They have a programmed a random image selector.

Hekler et al [12], considered the combination of HAM10000 and ISIC datasets, and performed classification over 5 classes. The lesion images were classified and using CNN and physicians. The two independently taken decisions were then combined. The combination of physician and model resulted in an accuracy of 82.95%. Another objective of the study being the correct classification of lesions as either benign or malignan. Their finding found that combination of human and artificial intelligence achieves better result than independent results of both of them. *Chaturvedi* el at [13] utilized a pretrained on approximately 1,280,000 images from 2014 ImageNet Challenge, Mobile Net which was further finetuned on 10,015 dermoscopy images of HAM10000 dataset employing transfer learning. This made it possible to achieve an accuracy of 83.1% for the classifying 7 different skin lesions. and top2 accuracy of 91.36% and top 3 accuracy of 95.34%. the weighted average of precision is 89%, weighted avergae of f1 score is recall is 83%, and weighted average of f1 score is 83%. textitEmara et al [14] dealt with HAM10000 dataset and achieved a whooping accuracy of 94.7% using a modified InceptionV4 model. The dataset used in study is inbalanced. Therefore, The proposed model was enhanced by employing feature reuse using long residual connection to increase the model performance. *Chen* et al [15] used a custom dataset with 9 classes and used Res Net 50 pretrained mode and trained the model using transfer learning, and ultimately achieved an accuracy of 83.74%.

Jinnai et al. [16] utilised 5846 clinical images of pigmented skin lesions of 3551 patients, A test dataset was selected randomly of the size 666. They used a faster, region-based CNN (FRCNN) with the training dataset and checked the performance of the model on the test dataset. This resulted in an accuracy of 86.2%. and that of the BCDs and TRNs is 79.5% and 75.1% respectively. where false positive rates were 5.5% and predective postive values were 84.7% *Chaturvedi* et al. [17] performed fine tuning of the HAM10000 dataset and tested over pre-trained CNN model and 4 ensemble methods, accuracy of 93.20% was attained while using a single model, which dropped down to 92.83% in case of ensemble methods. They propsed use of ResNeXt101 to cancer predction due to their ability to reach high accuracy and optimized architecture. *Kondaveeti and Edupuganti* [18] used the HAM10000 dataset to classify 7 types of skin cancers. The base models for classification were ResNet50, MobileNet, and InceptionV3. They used transfer learning utilizing multiple pre-trained models, combined with class-weighted loss and data augmentation techniques for the classification process. With this they managed to achieve a categorical accuracy of 90%. with precision of 0.89 and recall of 0.9.

Maiti et al. [19] used a dataset having 4992 images and performed classification over 3 classes. With the specified color quantization method increased the accuracy of the popular machine learning models it gives an accuracy of 97.08% in random forest. The model preserves a decent accuracy with KNN and gradient boosting as well. *Ashraf* et al. [20] utilised the DQH Hospital Dataset of Faisalabad and Pakistan having 400 images, their proposed method exhibited a classification accuracy of about 93.29%. *Alagu* et al. [21] utilised the ISIC dataset and incorporated DenseNet architecture with transfer learning, classification accuracy of 95% was obtained using CNN with Stochastic Gradient Descent optimization technique. The main objective of proposed research work is to identify the melanoma cells using convolutional neural network architecture incorporated with optimization techniques.

Mijwil [22] used Convolutional neural network and applied 3 architectures including Inception V3, Resnet, and VGG19, and worked upon a dataset containing more than 24000 skin cancer images. The best accuracy of 86.90% was achieved using the Inception V3 architecture. and precision of 87.37%,sensitivity of 86.14%. *Shah* [23] used DCNN and LRNet to train the neural network on

low-resolution images. They used the HAM10000 dataset, and achieved an accuracy of 90.6%,precison of 94.2% and sensitivity of 91%. They claimed that their proposed approach outperforms all the pre-trained models and drastically reduces the training time. *Maron* et al. [24] utilised the SAM dataset containing 319 images, and Implemented CNN over four different architectures AlexNet, VGG16+BN, ResNet50, DenseNet121 although specific accuracy values were not detailed. They have used Skin Archive Munich (SAM), SAM-corrupted (SAM-C) and SAM-perturbed (SAM-P) as benchmark of dataset. To maintain OOD status, ground truth labels are not provided. *Ali* et al. [25] used the HAM10000 dataset, and used a modified DCNN for the accurate classification between benign and malignant skin lesions. To remove the noise and artifacts, kernel or filters were applied. The input images were normalized, the features were extracted that helped for accurate classification. The performance was evaluated by comparing the

Table 1
The architecture of the DenseNet-like model. Conv represents convolutional layers, MaxPool is max pooling, AveragePool is average pooling, BatchNorm is batch normalization, Dropout is dropout, Dense is a fully connected layer, and GlobalAveragePool is global average pooling.

Section	Layer	Type	Filters/Units	Kernel Size	Stride	Input Size	Output Size	Activation
	1	Conv2D	64	(7, 7)	(2, 2)	128x128x3	64x64x64	ReLU
	2	MaxPooling2D	-	(3, 3)	(2, 2)	64x64x64	32x32x64	-
	3	Dense Block 1	Conv2D	(1x1 + 3x3) x 6	(1, 1)	32x32x64	32x32x256	ReLU
	4	Transition Block 1	Conv2D + AveragePooling2D	(1, 1)	(2, 2)	32x32x256	16x16x128	ReLU
Encorder	5	Dense Block 2	Conv2D	(1x1 + 3x3) x 12	(1, 1)	16x16x128	16x16x512	ReLU
	6	Transition Block 2	Conv2D + AveragePooling2D	(1, 1)	(2, 2)	16x16x512	8x8x256	ReLU
	7	Dense Block 3	Conv2D	(1x1 + 3x3) x 24	(1, 1)	8x8x256	8x8x1024	ReLU
	8	Transition Block 3	Conv2D + AveragePooling2D	(1, 1)	(2, 2)	8x8x1024	4x4x512	ReLU
	9	Dense Block 4	Conv2D	(1x1 + 3x3) x 16	(1, 1)	4x4x512	4x4x784	ReLU
	10	BatchNormalization	-	-	-	4x4x784	4x4x784	-
	11	GlobalAveragePooling2D	-	-	-	4x4x784	784	-
	12	Dense	128	-	-	784	128	ReLU
Decoder	13	Dropout	-	-	-	128	128	-
	14	Dense	7	-	-	128	7	-
	15	Activation	-	-	-	7	7	Softmax

Table 2
Architecture of the ResNet-like Model. Conv represents convolutional layers, BatchNorm is batch normalization, Activation is activation functions, MaxPool is max pooling, and Add is addition of residual connections.

Section	Layer	Type	Filters/Units	Kernel Size	Strides	Channel (input)	Channel (output)
	1	Input	-	-	-	128x128x3	128x128x3
	2	Conv2D	64	(7, 7)	(2, 2)	128x128x3	64x64x64
	3	BatchNormalization	-	-	-	64x64x64	64x64x64
	4	Activation	ReLU	-	-	64x64x64	64x64x64
	5	MaxPooling2D	-	(3, 3)	(2, 2)	64x64x64	32x32x64
	6	Conv2D	64	(3, 3)	(1, 1)	32x32x64	32x32x64
	7	BatchNormalization	-	-	-	32x32x64	32x32x64
	8	Activation	ReLU	-	-	32x32x64	32x32x64
	9	Conv2D	64	(3, 3)	(1, 1)	32x32x64	32x32x64
	10	BatchNormalization	-	-	-	32x32x64	32x32x64
	11	Activation	ReLU	-	-	32x32x64	32x32x64
	12	Add (Residual)	-	-	-	32x32x64	32x32x64
	13	Activation	ReLU	-	-	32x32x64	32x32x64
	14	Conv2D	64	(3, 3)	(1, 1)	32x32x64	32x32x64
Encoder	15	BatchNormalization	-	-	-	32x32x64	32x32x64
	16	Activation	ReLU	-	-	32x32x64	32x32x64
	17	Conv2D	64	(3, 3)	(1, 1)	32x32x64	32x32x64
	18	BatchNormalization	-	-	-	32x32x64	32x32x64
	19	Activation	ReLU	-	-	32x32x64	32x32x64
	20	Add (Residual)	-	-	-	32x32x64	32x32x64
	21	Activation	ReLU	-	-	32x32x64	32x32x64
	22	Conv2D	64	(3, 3)	(1, 1)	32x32x64	32x32x64
	23	BatchNormalization	-	-	-	32x32x64	32x32x64
	24	Activation	ReLU	-	-	32x32x64	32x32x64
	25	Conv2D	64	(3, 3)	(1, 1)	32x32x64	32x32x64
	26	BatchNormalization	-	-	-	32x32x64	32x32x64
	27	Activation	ReLU	-	-	32x32x64	32x32x64
	28	Add (Residual)	-	-	-	32x32x64	32x32x64
	29	Activation	ReLU	-	-	32x32x64	32x32x64
	30	GlobalAveragePooling2D	-	-	-	32x32x64	64
	31	Dense	128	-	-	64	128
Decorder	32	Dropout	-	-	-	128	128
	33	Dense	7	-	-	128	7
	34	Activation	Softmax	-	-	7	7

Table 3
Architecture of the InceptionV3-based Model with Additional Layers. InceptionV3 represents the base model, GlobalAveragePooling2D is global average pooling, Dense is fully connected layers, Dropout is dropout layers, and BatchNormalization is batch normalization.

Layer	Type	Filters/Units	Kernel Size	Strides	Channel (input)	Channel (output)
1	Input	-	-	-	128x128x3	128x128x3
2	InceptionV3	-	-	-	128x128x3	Feature Map
3	GlobalAveragePooling2D	-	-	-	Feature Map	128
4	Dense	256	-	-	128	256
5	Dropout	-	-	-	256	256
6	Dense	128	-	-	256	128
7	Dropout	-	-	-	128	128
8	Dense	32	-	-	128	32
9	BatchNormalization	-	-	-	32	32
10	Dense	7	-	-	32	7
11	Activation	Softmax	-	-	7	7

Table 4
Architecture of the Sequential Model with Convolutional and Fully Connected Layers. Conv2D represents convolutional layers, MaxPooling2D is max pooling, BatchNormalization is batch normalization, Flatten is flattening, Dropout is dropout, and Dense is a fully connected layer.

Layer	Type	Filters/Units	Kernel Size	Stride	Channel (input)	Channel (output)	
	1	Conv2D	32	(3, 3)	(1, 1)	128x128x3	128x128x32
	2	MaxPooling2D	-	-	(2, 2)	128x128x32	64x64x32
	3	BatchNormalization	-	-	-	64x64x32	64x64x32
	4	Conv2D	64	(3, 3)	(1, 1)	64x64x32	64x64x64
	5	Conv2D	64	(3, 3)	(1, 1)	64x64x64	64x64x64
Encoder	6	MaxPooling2D	-	-	(2, 2)	64x64x64	32x32x64
	7	BatchNormalization	-	-	-	32x32x64	32x32x64
	8	Conv2D	128	(3, 3)	(1, 1)	32x32x64	32x32x128
	9	Conv2D	128	(3, 3)	(1, 1)	32x32x128	32x32x128
	10	MaxPooling2D	-	-	(2, 2)	32x32x128	16x16x128
	11	BatchNormalization	-	-	-	16x16x128	16x16x128
	12	Flatten	-	-	-	16x16x128	32768
	13	Dropout	-	-	-	32768	32768
	14	Dense	256	-	-	32768	256
	15	Dropout	-	-	-	256	256
	16	Dense	128	-	-	256	128
Decorder	17	Dropout	-	-	-	128	128
	18	Dense	32	-	-	128	32
	19	BatchNormalization	-	-	-	32	32
	20	Dense	7	-	-	32	7
	21	Activation	Softmax	-	-	7	7

model with transfer learning models such as AlexNet, ResNet, VGG-16, and thus obtained 93.16% of training and 91.93% of testing accuracy, and thus claimed that DCNN model define themselves as more reliable and robust when compared with existing transfer models. *Yilmaz* et al [26] used the ISIC 2017 dataset consisting of 2750 images. Images were trained with transfer learning and fine-tuning approach. A total of 9 models (3 different mobile deep learning models and 3 different batch size values) were created using deep learning model and different batch values. NASNetMobile model with 16 batch size, yielding a best accuracy of 82%. *Mazoure* et al [27] used the ISIC 2018 dataset, they presented DUNEScan (Deep Uncertainty Estimation for Skin Cancer), a web server that performs an in-depth analysis of uncertainty in commonly used skin cancer classification models based on convolutional neural networks (CNNs), over it and achieved an accuracy of 83.4%. Grad-CAM and UMAP algorithms were used to visualize the classification manifold for the user's input taken from the frontend. *Bechelli* et al. [28] worked upon the ISIC dataset, and applied algorithms like K-nearest neighbors classifier, Logistic Regression, Linear Discriminant Analysis, Deep Learning, Custom Convolutional Neural Network model, leveraged transfer learning via use of pretrained models such as VGG16, ResNet50 etc. It was concluded that DL models outperformed the ML models, with accuracies up to 88%. Accuracy of ml models was increased using ensemble learning. *Maran* et al. [29] used the Ph2 dataset having three classes and achieved an accuracy of 99.33% using hybrid deep

learning and a sub-band fusion of 3D wavelets. It is a non-invasive and objective method for inspecting skin images. A simple median filtering was used to remove unwanted information such as hair and noise. *Filali* et al. [30] conducted a comparative study between different deep learning approaches for skin cancer classification and analysis, combined both Ph2 and ISIC dataset, and used several models including, ResNet, VGG16, Google Net and AlexNet. Among these models ResNet resulted in best accuracy of 82%. *Tabrizchi* et al. [31] using an enhanced VGG16 model, dealt with ISIC dataset containing nearly 34000 images and achieved an overall accuracy of 86.30%. The proposed model uses a enhanced architecture of VGG-16, which improves the accuracy of skin cancer detection. According to them, proposed model has outperforms alternative technique in respect to accuracy. *Xin* et al. [32] utilised the HAM10000 dataset and performed classification over 7 classes, a VIT network was established to verify effectiveness of Skin-Trans. Multi-scale and overlapping sliding windows were used to serialize the image. The model proposed achieved 94.3% accuracy. *Qasim Gilani* [33] used the ISIC 2019 dataset with around 7000 images and categorising as melanoma and non-melanoma, and achieved an accuracy of 89.57%, f1 score of 90.07% using their spiking VGG-13 model. spiking VGG-13 model is chosen because of its power-efficient behaviour. *Mridha* et al. [34] used an optimised CNN, and worked upon the HAM10000 dataset and classified images into 7 types of skin cancer. Adam and RMSprop were the optimization functions used over Relu, Swich and Tanh, and managed to achieve an accuracy of 82% and 47% loss accuracy. *Guruaj* et al [35] worked upon the HAM10000 dataset, and achieved an accuracy of 91.2%. The f1 score is 91.7% and the accuracy obtained by using undersa,pling technique is 83%. Transfer learning techniques like DenseNet169 and Resnet 50 were used to train the model. The data pre-processing techniques like dull razor and segmentation using autoencoder and decoder were employed.

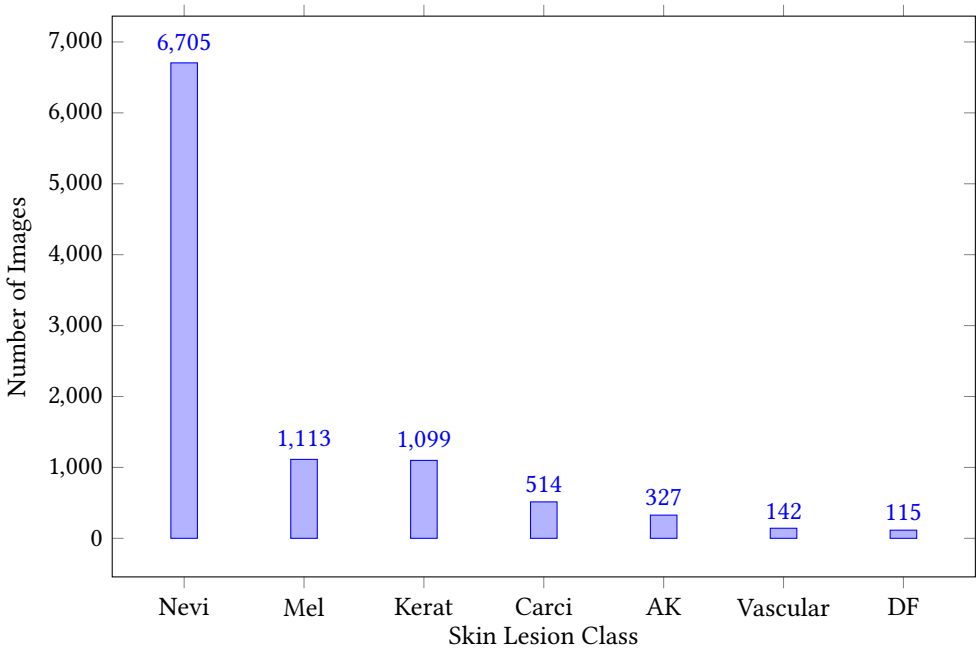


Figure 1. Distribution of Images Across Skin Lesion Classes

3. Materials and Methods

3.1 Dataset

To train the model, we have used Skin Cancer MNIST: HAM10000 dataset developed by Medical University of Vienna. This dataset contains 10,015 dermatoscopic images of common pigmented skin lesions. The dataset contains details of:

1. Disease
2. Patient History
3. Patient age
4. Patient sex
5. Localization of sample image

Distribution of images for each disease:

1. Melanocytic nevi(mv): 6705 images
2. Melanoma(ml): 1113 images
3. Benign keratosis-like lesions(bk): 1099 images
4. Basal cell carcinoma(bc): 514 images
5. Actinic keratoses(ak): 327 images
6. Vascular lesions(vl): 142 images
7. Dermatofibroma(df): 115 images

Link of dataset: https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham10000

3.2 Dataset Distribution

Section 3.1 provides a breakdown of the dataset into various labels. To train and evaluate the model, the dataset was divided into 3 sets: training set, validation set and testing set.

- Training dataset: 60% of the entire dataset
- Testing dataset: 20% of the entire dataset
- Validation dataset: 20% of the entire dataset

Hence, the total count of images used in each dataset is as:

- Training dataset: 6009 images
- Testing dataset: 2003 images
- Validation dataset: 2003 images

3.3 Data Pre-processing

The initial size of the images were 450×600 , since CNN processes each pixel individually through multiple layers of convolutions, pooling, and fully connected layers, and larger images have more pixels to process, ultimately increasing the processing time. To tackle this, we re-scaled the images to a size of 120×120 pixels. Moreover, as the dataset provided us with categorical values, we used a Label Encoder to convert these categorical values to numerical values.

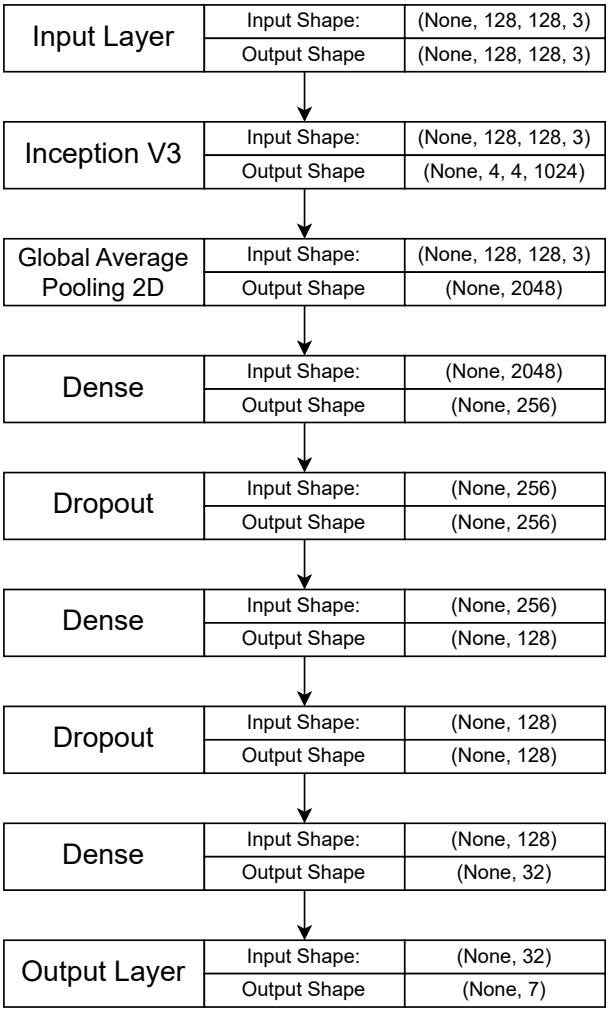
4. Understanding CNN Layers

4.1 Convolutional Layer

Convolutional layer is the building block of CNN. It generally contains a layer which is smaller than the size of the image.

Let us assume that:
H: Height of Input Layer
W: Width of the Input layer
C: Number of channels in input layer
h: Height of filter
w: Width of filter
P: Padding
S: Stride

Figure 2. Model Architecture using InceptionV3.



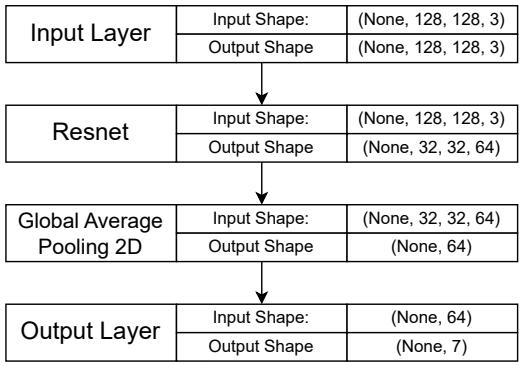


Figure 3. Model Architecture using ResNet.

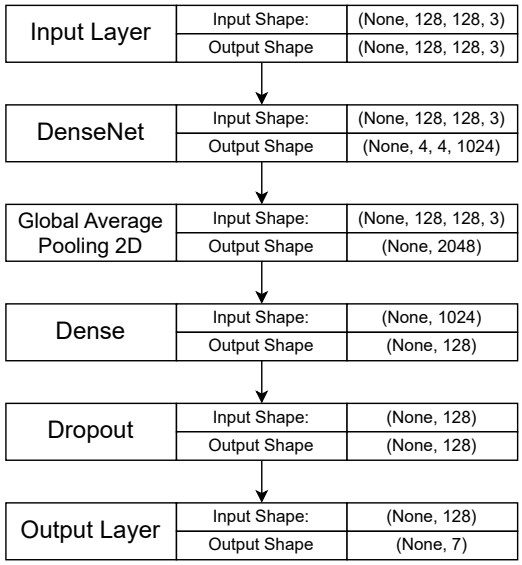


Figure 4. Model Architecture using DenseNet.

Then the output layer will be:

$$H_{\text{out}} = \left\lfloor \frac{H + 2P - h_f}{S} \right\rfloor + 1$$

$$W_{\text{out}} = \left\lfloor \frac{W + 2P - w_f}{S} \right\rfloor + 1$$

4.2 Max Pooling Layer

It is also an important component of a CNN. Generally, the Max Pooling Layer is placed after a Convolutional Layer.
It is used to dimensionally reduce the features of the output layer coming from the Convolutional Layer.

The output layer is calculated as:

$$H_{\text{out}} = \left\lfloor \frac{H - p}{s} \right\rfloor + 1$$

$$W_{\text{out}} = \left\lfloor \frac{W - p}{s} \right\rfloor + 1$$

4.3 Batch Normalization Layer

The Batch Normalization Layer is used to improve the performance and stability of a neural network.
Normalization is done using two operations:

1. **Scaling** (γ)
2. **Shifting** (β)

Batch Normalization Operation:

$$\text{Normalization: } \mu_B = \frac{1}{N}\sum_{i=1}^{N} x_i, \quad \sigma_B^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_B)^2$$

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

$$\text{Scaling and Shifting: } y_i = \gamma \hat{x}_i + \beta$$

4.4 Dense Layer

It is used to connect all neurons of the current layer to the previous layer.
The output vector Y is calculated as:

$$Y = \sigma(X \cdot W + b)$$

4.5 Input Layer

The Input Layer is generally the first layer of a CNN. It is used to take input arrays that are going to be used for prediction.

In our model, the output layer is of size (None, 64, 64, 3).

4.6 Output Layer

The Output Layer is generally the final layer of a CNN. It helps in the classification of the input layer.

The Output Layer is calculated as:

$$Y_i = \frac{e^{X_i}}{\sum_j e^{X_j}}$$

In our model, the output layer is of size (None, 7).

5. Model Development and Training

5.1 Transfer Learning

Transfer learning is a technique in machine learning and deep learning where a model trained on one task is reused as the starting point for a model on a second related task. As shown in Fig. In our work, we have used ResNet50, MobileNet, InceptionV3 as pre-trained models.

5.2 Model Training

For training the model 60% (6009 images) of the dataset were used. The images were resized to 120*120 pixel size before inputting it to the model. The custom 20 layered CNN model used "RELU" as activation function, and Adam as the Loss function. The learning Rate, was varied over various values including 0.1, 0.01, 0.0001, and the later was chosen as the final value. The custom CNN 20 layered model was trained on 41 epoch, while the model using ResNet, MobileNet and Inception were trained on 32, 38 and epochs respectively. Figures 3-6 represent the architecture of the model being used.

6. Result and Performance

6.1 GUI development

To provide an easier access to the developed model, a NodeJs based GUI based upon child processes was also developed. It used cloudinary at the back end to facilitate the process of transmission of image to the model. The child process shift the control to the python script, and any output generated is passed back to the express server. This helps in fast retrieval of the results predicted by the model.h5 file. The model thus predicts the disease, and a mapping of disease with the medicines are made and shown onto the output screen on the GUI.

6.2 Confusion Matrices

6.2.1 20 Layered CNN Model

	mv	ml	bk	bc	ak	vl	df
mv	137	23	4	5	2	3	0
ml	9	153	3	1	4	0	0
bk	7	14	99	3	18	7	0
bc	0	3	0	165	0	0	0
ak	3	9	21	1	110	12	2
vl	2	13	8	2	25	104	4
df	0	0	0	0	0	0	144

The model worked upon an epoch value of 145, and generated a training accuracy of 0.84, which for the testing set dipped to 0.8.

6.2.2 ResNet50

	mv	ml	bk	bc	ak	vl	df
mv	44	10	19	89	6	6	0
ml	3	61	9	86	5	6	0
bk	1	1	88	34	10	14	0
bc	0	2	0	164	0	2	0
ak	1	1	34	14	75	31	2
vl	0	3	7	9	17	120	2
df	0	2	0	5	0	0	137

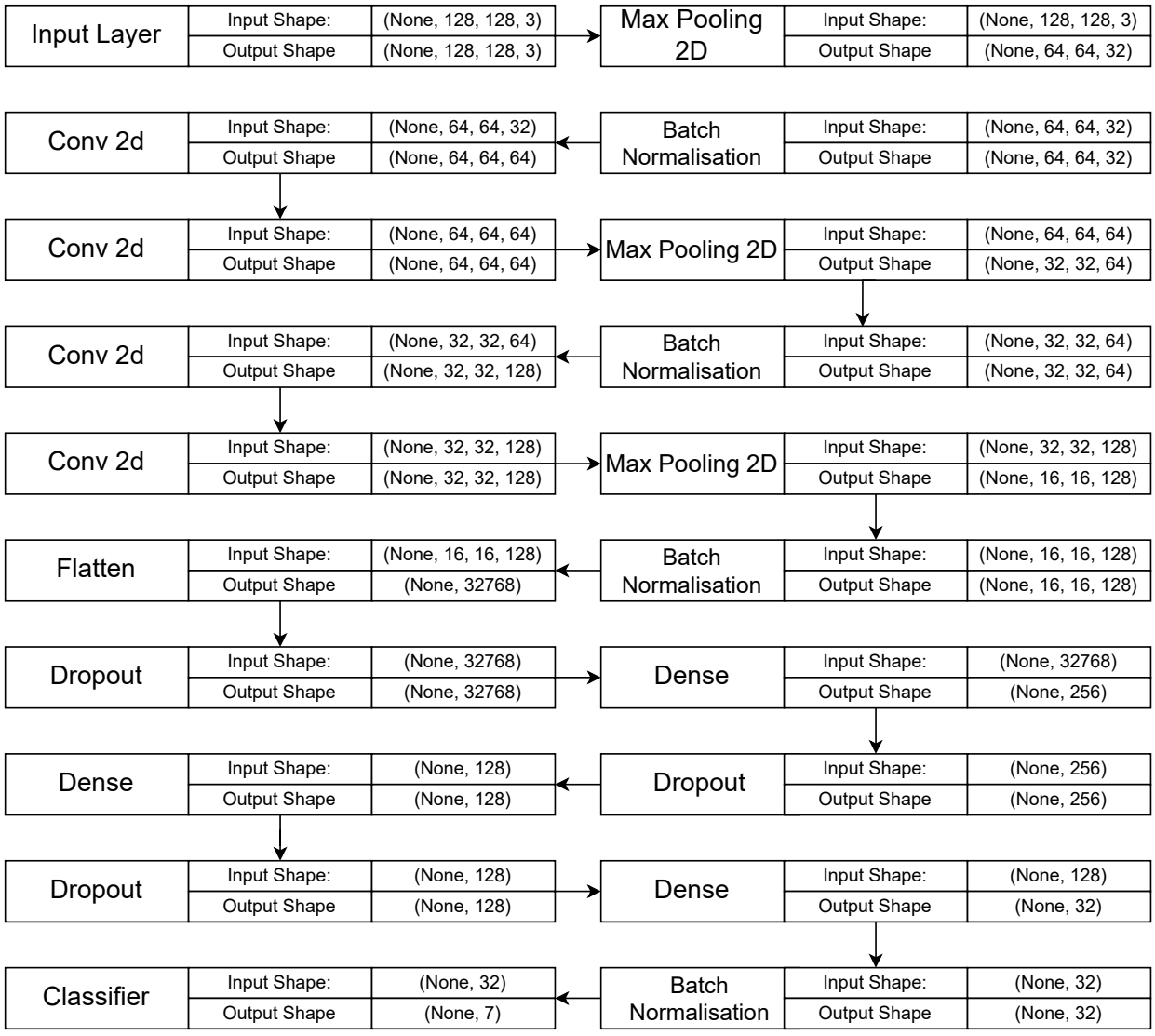


Figure 3 CNN 20 Layered Model Architecture.

The model failed to generalise well on the testing dataset and thus achieved a minimal accuracy of 0.68.

6.2.3 MobileNet

	mv	ml	bk	bc	ak	vl	df
mv	144	15	1	2	9	1	2
ml	14	136	6	1	8	4	1
bk	11	6	87	2	35	7	0
bc	0	0	0	168	0	0	0
ak	3	1	25	3	109	15	2
vl	2	4	9	1	22	116	4
df	1	0	0	0	0	1	142

The model achieved a training accuracy of 0.98 and a testing accuracy of 0.78, when trained for 150 epochs.

6.2.4 DenseNet

	mv	ml	bk	bc	ak	vl	df
mv	153	4	4	2	2	6	1
ml	3	154	5	0	5	2	1
bk	5	6	108	3	13	13	0
bc	0	0	0	168	0	0	0
ak	1	4	20	0	101	31	1
vl	2	4	8	1	15	126	2
df	0	0	0	0	0	0	144

The model achieved a test accuracy of 0.833 when it ran for 30 epochs.

6.3 Result table

The true positive rate, also referred to as sensitivity, measures the proportion of positive examples in the data- set that the model accurately identifed as positive while false positive rate, also given as 1-sensitivity, is the fraction of total negative examples in the dataset the model incorrectly predicted as positive.
The comparision of different models based on various metrices is shown in Table-1.

Model	Precision	Recall	F1-score
CNN	0.8022	0.7955	0.7941
MOBILENET	0.7769	0.7812	0.7778
RESNET	0.6832	0.6872	0.6944
DENSENET	0.8337	0.8339	0.8326

Table-1: Comparing different models.

Disease	TP	FP	Accuracy	Precision	F1-score
Melanocytic nevi	137	21	0.8567	0.8671	0.8614
Melanoma	153	49	0.7306	0.7574	0.7435
Benign keratosis-like lesions	99	47	0.6781	0.6786	0.6784
Basal cell carcinoma	165	49	0.7904	0.7710	0.7798
Actinic keratoses	110	39	0.7831	0.7383	0.7606
Vascular lesions	104	50	0.6742	0.6753	0.6748
Dermatofibroma	144	54	0.7701	0.7273	0.7482

Table-2: Classwise Prediction by CNN model

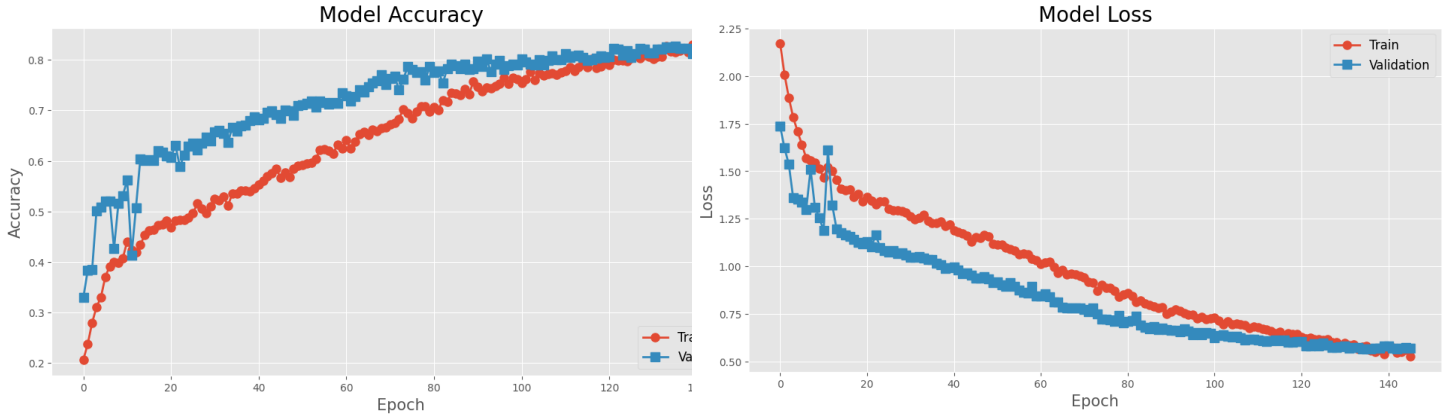


Figure-7. Model Loss and Accuracy of CNN 20 layered model.

Table-3 The minimum estimated prevalence of Benign Keratosis-like lesion disease with a 95% confidence level based on the assumption of a normal distribution for ResNet50

Threshold	TPR	FPR	Accuracy	Precision	F1 Score
0.1	0.7947	0.1878	0.7680	0.3415	0.4799
0.2	0.5929	0.0783	0.8469	0.4677	0.5232
0.3	0.5043	0.0398	0.8744	0.5741	0.5370
0.4	0.3564	0.0130	0.8823	0.7045	0.4749
0.5	0.2637	0.0089	0.8736	0.7025	0.3863
0.6	0.1596	0.0029	0.8659	0.7466	0.2681
0.7	0.0649	0.0009	0.8545	0.6789	0.1246
0.8	0.0067	0.0002	0.8469	0.3710	0.0167

Table-4 The maximum estimated prevalence of Benign Keratosis-like lesion disease with a 95% confidence level based on the assumption of a normal distribution for ResNet50

Threshold	TPR	FPR	Accuracy	Precision	F1 Score
0.1	0.8404	0.2361	0.8160	0.3986	0.5390
0.2	0.6503	0.1131	0.8870	0.5269	0.5819
0.3	0.5633	0.0663	0.9110	0.6320	0.5956
0.4	0.4139	0.0302	0.9177	0.7570	0.5340
0.5	0.3174	0.0240	0.9103	0.7551	0.4446
0.6	0.2053	0.0136	0.9037	0.7963	0.3220
0.7	0.0972	0.0094	0.8937	0.7328	0.1663
0.8	0.0203	0.0064	0.8870	0.4290	0.0356

Table-5 The minimum estimated prevalence of Benign Keratosis-like lesion disease with a 95% confidence level based on the assumption of a normal distribution for Sequential.

Threshold	TPR	FPR	Accuracy	Precision	F1 Score
0.1	0.7947	0.1002	0.8526	0.4810	0.7947
0.2	0.7665	0.0530	0.8968	0.6110	0.7665
0.3	0.6826	0.0321	0.9075	0.6826	0.6826
0.4	0.6204	0.0215	0.9104	0.7305	0.6204
0.5	0.5724	0.0138	0.9124	0.7782	0.5724
0.6	0.5247	0.0105	0.9094	0.7973	0.5247
0.7	0.4704	0.0089	0.9036	0.7996	0.4704
0.8	0.3897	0.0036	0.8987	0.8536	0.3897
0.9	0.2637	0.0003	0.8852	0.8984	0.2637

Table-6 The maximum estimated prevalence of Benign Keratosis-like lesion disease with a 95% confidence level based on the assumption of a normal distribution for Sequential.

Threshold	TPR	FPR	Accuracy	Precision	Recall	F1 Score
0.1	0.8404	0.1385	0.8921	0.5401	0.8404	0.6571
0.2	0.8146	0.0828	0.9300	0.6677	0.8146	0.7339
0.3	0.7363	0.0564	0.9390	0.7363	0.7363	0.7363
0.4	0.6769	0.0423	0.9414	0.7813	0.6769	0.7253
0.5	0.6303	0.0314	0.9430	0.8254	0.6303	0.7147
0.6	0.5834	0.0265	0.9406	0.8427	0.5834	0.6893
0.7	0.5296	0.0240	0.9357	0.8448	0.5296	0.6505
0.8	0.4481	0.0149	0.9317	0.8929	0.4481	0.5955
0.9	0.3174	0.0079	0.9202	0.9314	0.3174	0.4704

Table-7 The minimum estimated prevalence of Benign Keratosis-like lesion disease with a 95% confidence level based on the assumption of a normal distribution for DenseNet.

Threshold	TPR	FPR	Accuracy	Precision	Recall	F1 Score
0.1	0.8303	0.0764	0.8813	0.5515	0.8303	0.6631
0.2	0.7947	0.0503	0.9036	0.6296	0.7947	0.7026
0.3	0.7384	0.0438	0.9026	0.6408	0.7384	0.6862
0.4	0.4442	0.0353	0.9036	0.3894	0.4442	0.4151
0.5	0.7174	0.0294	0.9153	0.7072	0.7174	0.7123
0.6	0.7035	0.0259	0.9173	0.7244	0.7035	0.7138
0.7	0.6826	0.0206	0.9202	0.7532	0.6826	0.7162
0.8	0.6411	0.0164	0.9192	0.7747	0.6411	0.7016
0.9	0.5451	0.0105	0.9124	0.8028	0.5451	0.6497

Table-8 The maximum estimated prevalence of Benign Keratosis-like lesion disease with a 95% confidence level based on the assumption of a normal distribution for DenseNet.

Threshold	TPR	FPR	Accuracy	Precision	Recall	F1 Score
0.1	0.8724	0.1108	0.9169	0.6098	0.8724	0.7177
0.2	0.8404	0.0794	0.9357	0.6857	0.8404	0.7552
0.3	0.7886	0.0714	0.9349	0.6965	0.7886	0.7397
0.4	0.5032	0.0605	0.9357	0.4478	0.5032	0.4738
0.5	0.7691	0.0529	0.9454	0.7595	0.7691	0.7642
0.6	0.7560	0.0482	0.9470	0.7756	0.7560	0.7657
0.7	0.7363	0.0411	0.9494	0.8024	0.7363	0.7679
0.8	0.6967	0.0351	0.9486	0.8221	0.6967	0.7543
0.9	0.6036	0.0265	0.9430	0.8477	0.6036	0.7049

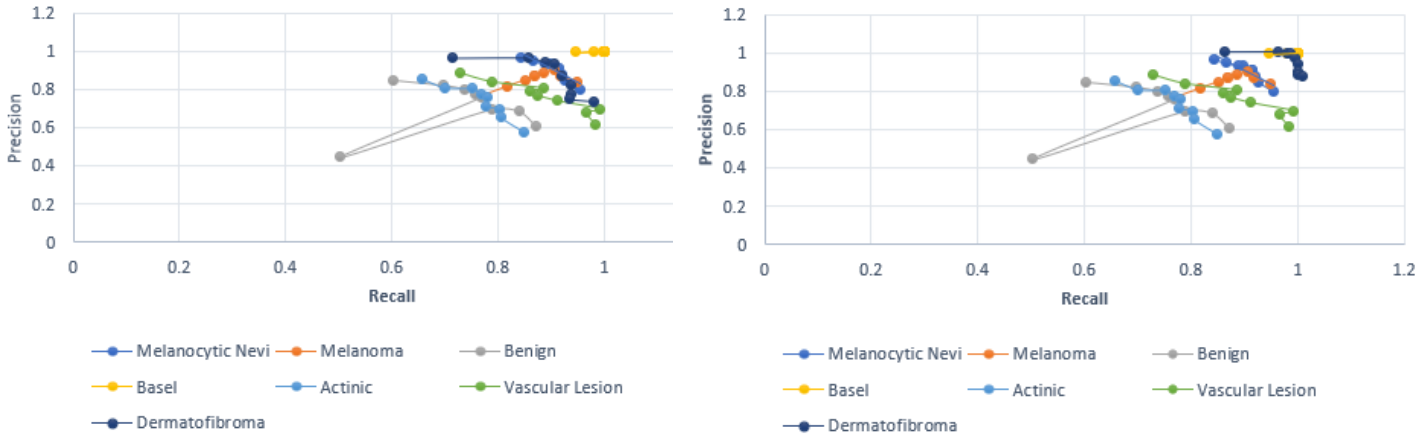


Figure-8. PR curve for lower and upper bound, respectively at 95% confidence interval for ResNet50 model.

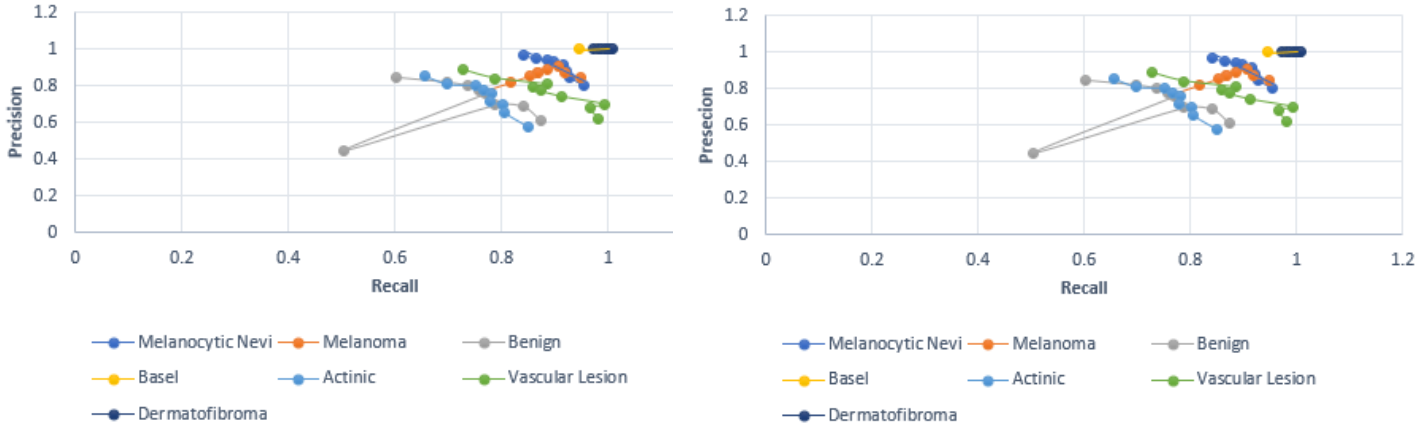


Figure-9. PR curve for lower and upper bound, respectively at 95% confidence interval for Sequential model.

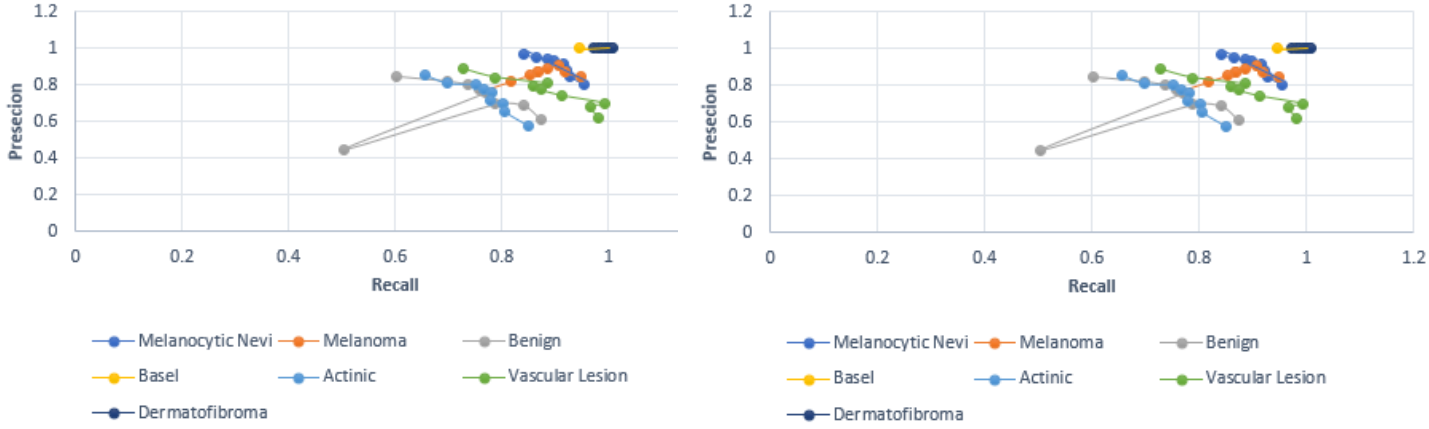


Figure-10. PR curve for lower and upper bound, respectively at 95% confidence interval for DenseNet model.

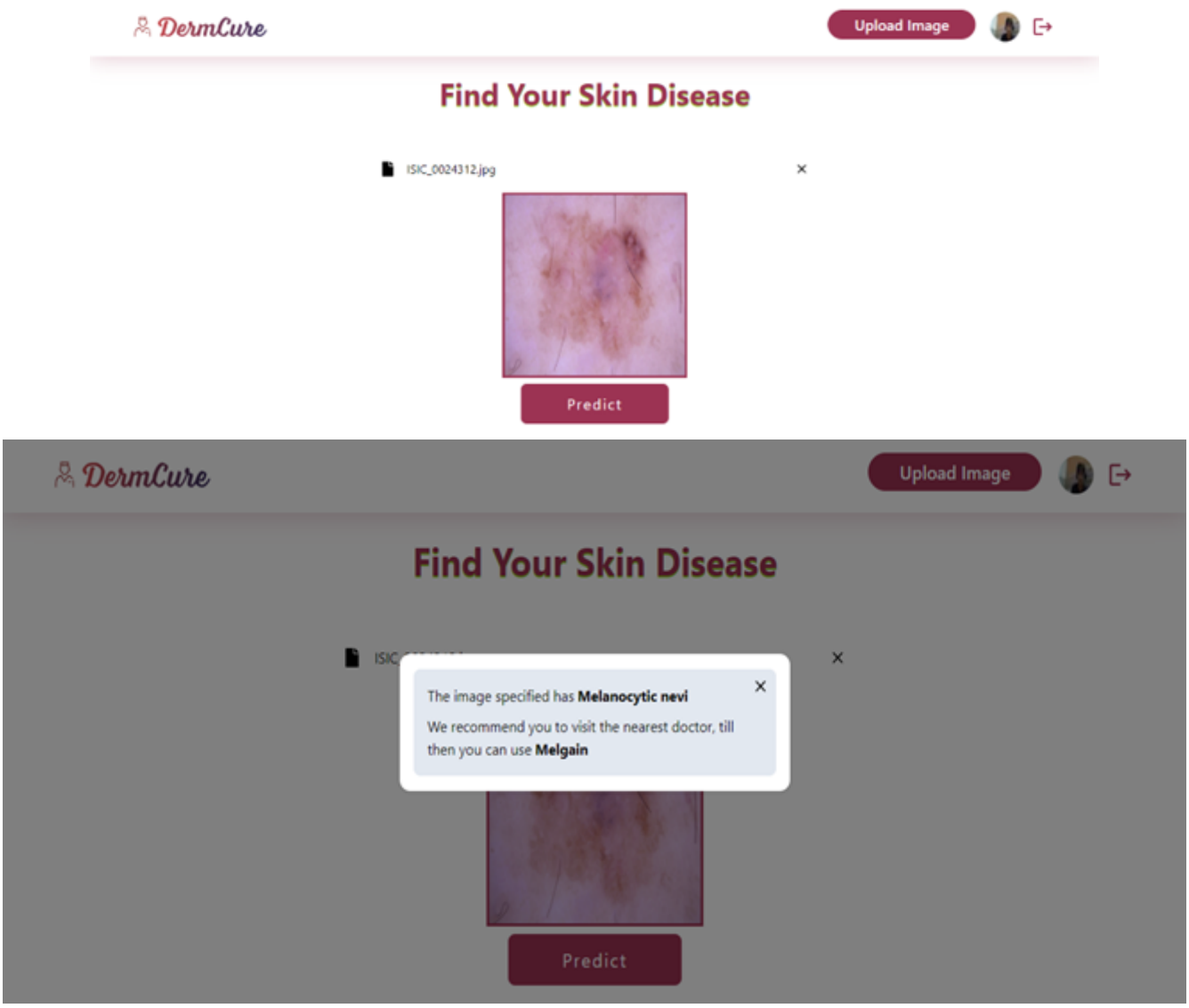


Figure-11. The GUI developed.

7. References

1. Dorj U.-O., Lee K.-K., Choi J.-Y., Lee M. The skin cancer classification using deep convolutional neural network. Multimedia Tools and Applications. 2018. [Link]
2. Andre E, Brett K, Novoa Roberto A, Justin K, Swetter Susan M, Blau Helen M, Sebastian T Dermatologist-level classification of skin cancer with deep neural networks. 2017. [Link]
3. Rezvantalab A., Safigholi H., Karimijeshni S. Dermatologist level dermoscopy skin cancer classification using different deep learning convolutional neural networks algorithms. 2018. [Link]
4. Hosny K. M., Kassem M. A., Foaud M. M. Skin Cancer Classification using Deep Learning and Transfer Learning. 2018. [Link]
5. Lee YC, Jun SH, Won HH, WonDerM: skin lesion classification with fine-tuned neural networks. 2018. [Link]
6. Jamil U, Akram MU, Khalid S, Abbas S, Saleem K Computer based melanocytic and nevus image enhancement and segmentation. BioMed Res Int. 2016. [Link]
7. P. Soille, Morphological Image Analysis: Principles and Applications, Springer, Berline, Germany, 2nd edition, 2003 [Link]
8. Shete AS, Rane AS, Gaikwad PS, Pati MH Detection of skin cancer using CNN algorithm. [Link]
9. Pham T. C., Tran G. S., Nghiem T. P., Doucet A., Luong C. M., Hoang V.-D. A Comparative Study for Classification of Skin Cancer. 2019 [Link]
10. Noel Codella, Veronica Rotemberg, Philipp Tschandl, M. Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, Harald Kittler, Allan Halpern: “Skin Lesion Analysis Toward Melanoma Detection 2018 [Link]
11. Tschandl, P., Rosendahl, C. & Kittler, H. The HAM10000 dataset, a large collection of multi-sources dermatoscopic images of common pigmented skin lesions.2018 [Link]
12. Hekler A., Utikal J. S., Enk A. H., et al. Superior skin cancer

classification by the combination of human and artificial intelligence. European Journal of Cancer .2019 [Link]
13. Chaturvedi S. S., Gupta K., Prasad P. S. Skin Lesion Analyser: An Efficient Seven-Way Multi-class Skin Cancer Classification Using MobileNet. In: Hassanien A., Bhatnagar R., Darwish A., editors. Advanced Machine Learning Technologies and Applications. AMLTA 2020 [Link]
14. Emara T., Afify H. M., Ismail F. H., Hassanien A. E. A Modified Inception-v4 for Imbalanced Skin Cancer Classification Dataset. 2019 [Link]
15. Chen M., Chen W., Chen W., Cai L., Chai G. Skin cancer classification with deep convolutional neural networks. Journal of Medical Imaging and Health Informatics. 2020 [Link]
16. Jinnai S., Yamazaki N., Hirano Y., Sugawara Y., Ohe Y., Hamamoto R. The development of a skin cancer classification system for pigmented skin lesions using deep learning. Biomolecules . 2020. [Link]
17. Chaturvedi S. S., Tembhurne J. V., Diwan T. A multi-class skin cancer classification using deep convolutional neural networks. Multimedia Tools and Applications. 2020 [Link]
18. Kondaveeti H. K., Edupuganti P. Skin Cancer Classification using Transfer Learning. 2020 [Link]
19. Maiti A., Chatterjee B., Santosh K. C. Skin cancer classification through quantized color features and generative adversarial network. 2021. [Link]
20. Ashraf R., Kiran I., Mahmood T., Butt A. U. R., Razzaq N., Farooq Z. An efficient technique for skin cancer classification using deep learning. 2020 [Link]
21. Alagu S., Bhoopathy Bagan K. Skin cancer classification in dermoscopy images using convolutional neural network. Nucleation and Atmospheric Aerosols. 2021 [Link]
22. Mijwil M. M. Skin cancer disease images classification using deep learning solutions. Multimedia Tools and Applications . 2021 [Link]
23. Shah M. LRNet: Skin Cancer Classification Using Low-Resolution Images. 2021 [Link]
24. Maron R. C., Schlager J. G., Haggenmüller S., et al. A benchmark for neural network robustness in skin cancer classification. 2021. [Link]
25. Ali M. S., Miah M. S., Haque J., Rahman M. M., Islam M. K. An enhanced technique of skin cancer classification using deep convolutional neural network with transfer learning models. Machine Learning with Applications. 2021. [Link]
26. Yilmaz A., Kalebasi M., Samoylenko Y., Guvenilir M. E., Uvet H. Benchmarking of lightweight deep learning architectures for skin cancer classification using ISIC 2017 dataset. 2021. [Link]
27. Mazoure B., Mazoure A., Bédard J., Makarenkov V. DUNEScan: a web server for uncertainty estimation in skin cancer detection with deep neural networks. Scientific Reports. 2022 [Link]
28. Bechelli S., Delhommelle J. Machine learning and deep learning algorithms for skin cancer classification from dermoscopic images. Bioengineering. 2022 [Link]
29. Maniraj S. P., Sardar Maran P. A hybrid deep learning approach for skin cancer diagnosis using subband fusion of 3D wavelets. The Journal of Supercomputing. 2022 [Link]
30. Filali Y., El Khoukhi H., Sabri M. A., Aarab A. Analysis, and classification of skin cancer based on deep learning approach. 2022. [Link]
31. Tabrizchi H., Parvizpour S., Razmara J. An improved VGG model for skin cancer detection. Neural Processing Letters . 2023 [Link]
32. Xin C., Liu Z., Zhao K., et al. An improved transformer network for skin cancer classification. Computers in Biology and Medicine. 2022 [Link]
33. Gilani S. Q., Syed T., Umair M., Marques O. Skin cancer classification using deep spiking neural network. Journal of Digital Imaging . 2023 [Link]
34. Mridha K., Uddin M. M., Shin J., Khadka S., Mridha M. F. An interpretable skin cancer classification using optimized convolutional neural network for a smart healthcare system. IEEE Access: Practical Innovations, Open Solutions . 2023 [Link]
35. Gururaj H. L., Manju N., Nagarjun A., Manjunath Aradhya V. N., Flammini F. DeepSkin: a deep learning approach for skin cancer classification. IEEE Access: Practical Innovations, Open Solutions . 2023 [Link]