Multi-Class Skin Cancer Detection: Implementing Various CNN Architectures with Mobile App Integration

Md Zahurul Haque
Department of CSE
Manarat International University

Monoara Sultana Morzina
Department of CSE
Jahangirnagar University

ABSTRACT
The skin, as the largest and one of the most vital organs of the human body, acts as a protective barrier between internal organs and the external environment. It performs several crucial functions, including protection, regulation, and sensation. When skin cells undergo genetic mutations, they can grow and multiply uncontrollably, leading to the formation of malignant tumors and, ultimately, the development of skin cancer. The primary cause of skin cancer is prolonged exposure to ultraviolet (UV) radiation from the sun or artificial sources like tanning beds, which damages the DNA in skin cells and promotes the growth of cancerous cells. Studies indicate that by the age of 70, one in five Americans will develop skin cancer, with more than two individuals succumbing to the disease every hour. However, early detection of skin cancer significantly improves the chances of successful treatment and recovery. Although numerous classification algorithms have been proposed in recent years to detect various stages of skin cancer, many still suffer from limited accuracy and high implementation complexity. In this project, we present a modified CNN-based skin cancer classification model utilizing six distinct architectures: VGG19, DenseNet201, InceptionV3, Xception, ResNet152, and MobileNetV2. These models were trained and evaluated using the HAM10000 dataset (Human Against Machine with 10,000 labeled training images). Our results demonstrate that DenseNet201 outperforms all other CNN architectures, achieving an accuracy of 97% along with the highest precision, recall, and F1-score. A comparative analysis of these models is also provided to highlight their performance differences.

Keywords
Skin Cancer, CNN Architectures, and Classification Algorithms.

1. INTRODUCTION
Skin cancer happens when skin cells develop and duplicate in an undisciplined and chaotic manner. Regularly, a skin cell develops, when the existing cells become old and die or when they become harmed. On the off chance that this interaction doesn't function as it ought to, a gigantic development of cells (some of them can be strange) results. The work of [1] and [2] mentions that this assortment of cells might be harmless (non-cancer cells), which don't spread or inflict any kind of damage, or carcinogenic, which might spread to local tissue or different regions in our body if not identified early and treated. The development of skin cancer initiates in the epidermis, which is primarily composed of three types of cells: squamous cells, basal cells, and melanocyte cells, as shown in Figure 1. The thin, level cells that structure the top layer of the epidermis are called squamous cells. The lowermost round cells are the Basal cells of the epidermis. Melanocytes shield further layers of skin from daylight by creating a brown pigment named melanin. When these cells experience inordinate ultraviolet (UV) beams from the sun, tanning beds, or sunlamps, the DNA changes incite influence the development of skin cells and in the end shape into skin cancer. In [3], three major types of skin cancer are discussed which are commonly engaged with squamous cells, basal cells and melanocytes and they are known as Basal Cell Carcinoma, Squamous Cell Carcinoma, and Melanoma. [3]

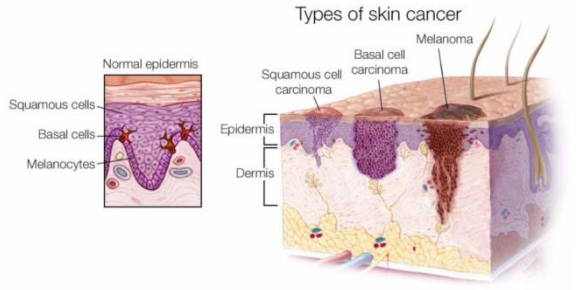

Figure 1: Types of Skin Cancer

Of a wide range of skin cancer growth, melanoma causes the most passing since it tends to spread to different pieces of the body which may principally incorporate fundamental organs. One of the deadliest and most threatening forms of cancer is Malignant Melanoma. Although only 4% of the population is affected by it, Malignant Melanoma is the reason of 75% of deaths caused by malignant skin cancer [4]. Assuming that melanoma is recognized or analyzed in its beginning phases, it tends to be relieved.

If not detected early, Melanoma can expand deeper into the skin and even affect each part of our body. Then, at that point, it turns out to be extremely challenging to treat. Consequently, an early identification framework is required that can work within recognizing kinds of skin cancer or other skin issues, for example, harmless growths on the skin that look basically the same as skin malignant growth. Recently, Convolutional Neural Networks have been widely employed for various classifications, including the accurate identification of different types of skin cancer. In the classification of skin cancer growths, different CNN models have decisively outperformed skilled healthcare experts [6] The aim of this project is to build a model capable of categorizing skin lesion images into one of the seven classes available in the HAM10000 dataset using different deep neural network architectures. We used VGG19 [7], InceptionV3 [8], Xception [9], MobileNetV2 [10], ResNet152 [11], and DenseNet201 [12] as the pre-trained base models. The models in this study were enhanced by adding extra convolutional and dense layers, along with activation functions to improve their performance. These modified models were trained on the HAM10000 dataset. Of all the models tested, DenseNet201 achieved the highest accuracy of 97%. A comparative analysis revealed that our modified

models outperform the original versions. This paper is intended to serve as a valuable resource for future researchers in the field of skin cancer detection.

2. RELATED WORKS

All through ongoing years, scientists have invested significant effort to develop intelligent frameworks for cancer identification. In the medical field, Convolutional Neural Networks (CNN) has performed a crucial role in the detection and classification of images for various applications related to image processing. Here we referenced various research studies that utilize deep learning techniques for the classification of this cancer.

In their study, Filali et al. [13] developed a model for identifying melanoma skin cancer by combining pre-trained and scratch Convolutional Neural Network models. Feature engineering was employed to eliminate irrelevant elements, and to remove artifacts, the Aujol model was used to decompose the image and identify the object's contour. The 'Otsu' algorithm was then used to segment the new object for input into the neural network. The authors reported a high level of accuracy, achieving 87.8% precision using the PH2 dataset.

Saket S. Chaturvedi et al. [3] performed fine-tuning of more than seven classes of the HAM10000 dataset. HAM10000 dataset has a collection of 10,015 pictures isolated into training and validation sets each has 8,912 and 1,103 pictures separately. They conducted a comparative study to analyze the performance of five pre-trained convolutional neural networks (CNN) and ensemble models. To fine-tune the Xception model, a dense layer with the 'Relu initiation capability is added with a seven-class yield softmax layer. They utilized the Adam enhancer with a 0.001 learning rate for quicker model improvement. They reported maximum accuracy of 93.20% among the set of models. They also proposed using the ResNeXt101 model for MCS cancer classification to gain higher accuracy.

Sara Hosseinzadeh Kassani et al. [14] explored the discovery of melanoma utilizing different deep-learning networks using the HAM 10,000 dataset. The accuracy is 84%, 89%, and 90% achieved by AlexNet, VGGNet19, and VGGNet model16 model respectfully. The most noteworthy accuracy in this study was 92% for ResNet50.

Hossin et al. [15] utilize a diverse CNN approach with different regularization procedures named dropout batch normalization system for classifying dermoscopic images. Their proposed model recognized dangerous melanoma with 93.5% exactness utilizing a dataset of 3297 dermoscopic images.

Dr.J. Abdul Jaleel et al. [16] represented an approach in which their system uses Image processing techniques and Artificial Intelligence for diagnosing skin cancer. Their system's cancer identification process includes image filtering, image segmentation, extraction of features, and classification of test data using Artificial Neural Network (ANN). They also used Back-Propagation Neural (BPN) network along with ANN to categorize more accurately.

Another modified method of segmentation has been presented in [17] which is a different approach to implementing the Convolutional Neural Network (CNN) algorithm. In this model CNNs were implemented for extracting features and ANNs were used in order to categorize the extracted features.

3. METHODOLOGY

The generalized model for skin cancer classification is illustrated in Figure 2. The HAM10000 dataset was used to train the models. Initially, dermoscopic skin cancer images were preprocessed to match the input dimensions required by the architectures used in this study. Following this, data augmentation techniques were applied to increase the volume of training data. Six pre-trained models were employed: VGG19, DenseNet201, InceptionV3, Xception, ResNet152, and MobileNetV2, each customized to meet the specific requirements of our study.

The preprocessed and augmented dataset was then fed into these models for training. The training process involved fine-tuning in two phases. In Phase A, only the fully connected layer was trained, while in Phase B, both the convolutional and fully connected layers were trained to further improve accuracy. The final prediction classified the input images into one of seven types of skin cancer: Actinic Keratosis, Melanocytic Nevi, Benign Keratosis, Melanoma, Vascular Lesions, Dermatofibroma, or Basal Cell Carcinoma. This process was repeated across all six modified CNN models.

Figure 2: Proposed model for the skin cancer classification across multi-class.

3.1 Dataset Description

The dataset plays a crucial role in using any deep learning model as it serves as the foundation upon which the entire study is built. The dataset works as the source of information, observations, or measurements. To accomplish the research work, we used the HAM10000 dataset. The dataset encompasses with 10,015 skin lesion images of 7 classes: actinic keratosis, basal cell carcinoma, benign keratosis, dermatofibroma, melanocytic nevus, melanoma, and vascular lesion. The images were assembled over a time of 20 years. The images are accumulated by dermatoscopy instrument. The instrument is a combination of lenses that works as a magnifier that is used to capture the images of skin lesions. [18]

The HAM10000 dataset has become a benchmark dataset for

evaluating the performance of skin cancer detection algorithms and comparing the accuracy and generalization capability of different approaches. From the table given below, it is pretty evident that the dataset is highly imbalanced, as more than two-thirds of the images belong to the Melanocytic Nevi and a few to the Dermatofibroma and Vascular Lesion [19].

Table 1. HAM10000 Dataset Classes.

Diagnostic category	***Abbreviation***	***Number of images***	***Type***
Actinic keratosis	akiec	327	Benign or Malignant
Basal Cell Carcinoma	bcc	514	Malignant
Benign Keratosis Lesions	bkl	1099	Benign
Dermatofibroma	df	115	Benign
Melanocytic Nevi	mv	6705	Benign
Melanoma	mel	1113	Malignant
Vascular Lesion	vasc	142	Benign or Malignant

3.2 Classification Models and Fine-Tuning

Several CNN models have demonstrated great potential in cancer classification, particularly when working with large and complex datasets like medical images. Unlike traditional machine learning methods, which require manual feature extraction, deep neural networks can automatically extract features from images, making them ideal for image classification tasks. These models not only save time but also provide higher accuracy in medical image classification. As a result, we have utilized various CNN models to classify cancer cells.

3.2.1 VGG19

VGG16 consists of 16 layers, while VGG19 includes 19 layers. The basic architecture of VGG19 features 16 convolutional layers and 3 fully connected layers [7]. These convolutional layers are organized into five blocks, each utilizing 3×3 filters with a stride of 1 and padding of 1. These small filters capture intricate details in the images, while 2×2 max-pooling layers with a stride of 2 reduce the spatial dimensions of the feature maps, retaining the most critical information. The fully connected layers contain 4096 neurons each, followed by a final output layer with neurons matching the number of classes in the dataset. The small filters combined with the deeper architecture of VGG19 enable it to learn more complex features from input images. The architecture is designed to process input images of size 224×244 pixels. VGG19 was pre-trained on the ImageNet dataset, which includes over one million labeled images across more than a thousand categories. Due to its additional convolutional layers and a larger number of trainable parameters, VGG19 can handle more complex tasks than VGG16.

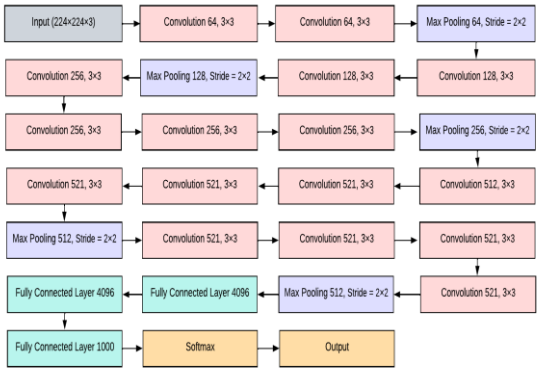

Figure 3: VGG19 model architecture

3.2.2 DenseNet201

DenseNet201 is a deep neural network consisting of 201 layers and over 20 million parameters. The input first goes through a convolutional layer with a 7×7 filter and a stride of 2, followed by a 3×3 max-pooling layer. The architecture is organized into four dense blocks, each containing a different number of convolutional layers: 6, 12, 48, and 32 layers in dense blocks 1, 2, 3, and 4, respectively. Within each dense block, every convolutional layer is connected to all preceding layers, allowing for the reuse of feature maps and more efficient learning. This connectivity results in a rich set of feature maps from the dense blocks. After each dense block, a transition layer is applied to down sample the feature maps, reducing their dimensionality. Once the feature maps pass through dense block 4, they are processed by a global average pooling layer with a 7×7 filter size. The final output is produced by a fully connected layer with a softmax activation function, providing the classification results. This architecture efficiently captures complex features and performs well in tasks involving large-scale image data [12].

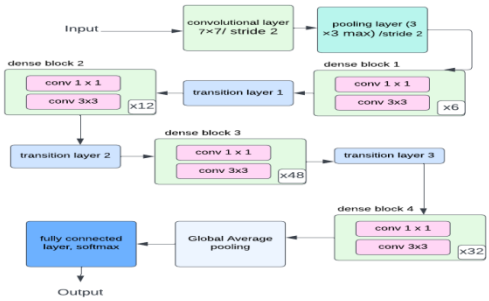

Figure 4: DenseNet201 model architecture

3.2.3 InceptionV3

Several modifications have been introduced to the architecture, such as factorizing convolutions into smaller convolutions and spatial factorization into asymmetric convolutions. For instance, instead of using a standard 3×3 convolution, it can be replaced by a sequence of a 1×3 convolution followed by a 3×1 convolution. This reduces the number of parameters and computational complexity while maintaining the same receptive field. Similarly, if a 3×3 convolution is replaced by a 2×2 convolution, the parameter count will increase on a small scale compared to using asymmetric convolutions.

Another key technique in InceptionV3 is grid reduction, which reduces the grid size to lower computational costs while preserving efficiency and maintaining important features in the network. Pooling operations, such as max-pooling or average-

pooling, are commonly used for grid size reduction, enabling the architecture to process large-scale data more efficiently without compromising performance.

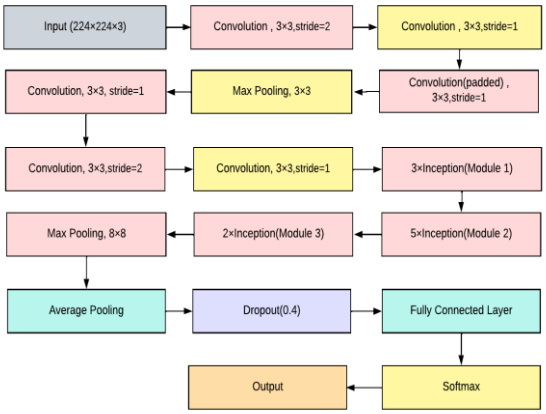

Figure 5: InceptionV3 model architecture

3.2.4 Xception

The architecture is composed of several blocks of convolution layers and separable convolution layers which is followed by batch normalization. The input images go through three different parts of the models: the entry flow, the middle flow, and the exit flow [9].

I. Entry Flow: The Entry Flow handles the initial stages of processing the input images. It begins by passing the images through several standard convolutional layers. This is followed by two sets of depth wise separable convolutional layers, both incorporating residual connections. The first set of separable convolutions is responsible for extracting low-level features, while the second set captures more complex features from the input. Once these processes are completed, the output is fed into the middle flow for further processing.

II. Middle Flow: The Middle Flow is made up of a series of depth wise separable convolutional layers, each connected via residual connections. These layers refine the feature maps received from the entry flow, extracting increasingly higher-level features. Typically, the middle flow repeats eight times, though this number can vary depending on the desired depth of the network. The primary role of the middle flow is to enhance the model's accuracy by learning complex patterns from the input data. After processing, the output is passed to the exit flow.

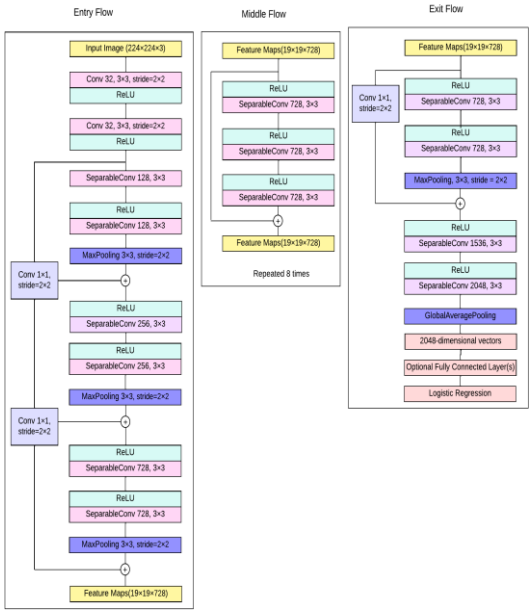

Figure 6: Xception model architecture

III. Exit Flow: The Exit Flow produces the final output of the network. It comprises a series of separable convolutional layers, followed by a global pooling layer and a fully connected layer. The separable convolutions in the exit flow continue to extract high-level features from the middle flow's output, while the global pooling layer reduces the spatial dimensions of the feature maps, transforming them into a 2048-dimensional vector. Finally, the fully connected layer maps this feature vector to the corresponding output class labels.

The Xception architecture is an advanced and efficient CNN model that has demonstrated superior performance compared to models like VGG-16, ResNet, and InceptionV3 in many classification tasks. Its use of depth wise separable convolutions reduces the computational burden while maintaining or improving accuracy. The incorporation of residual (skip) connections further enhances the model's training efficiency. These design choices make Xception a powerful and highly efficient architecture for image classification.

3.2.5 ResNet152

Residual Networks (ResNet) were introduced as a series of CNN models with similar architectures but varying depths. ResNet addresses the issue of degradation in deep neural networks using a residual learning unit, which includes an alternate connection that adds the original input to the network's output, allowing for better gradient flow and avoiding performance degradation in deeper networks.

The ResNet152 architecture consists of 152 convolutional layers. Its first layer applies 64 filters of size 7×7 with a stride of 2 to the input image. Following this, a 3×3 max-pooling layer reduces the spatial dimensions.

The second layer has a central module made up of three sub-layers:

- The first sub-layer uses 1×1 filters with 64 channels,
- The second sub-layer applies 3×3 filters with 64 channels,
- The third sub-layer uses 1×1 filters with 256 channels.

This module is repeated three times in the second layer.

In the third layer, the central module follows the same structure but increases the size of the filters:

- The first sub-layer uses 1×1 filters with 128 channels,
- The second sub-layer uses 3×3 filters with 128 channels,
- The third sub-layer uses 1×1 filters with 512 channels.

This module is repeated eight times in the third layer.
In the fourth layer, the module expands further:

- The first sub-layer applies 1×1 filters with 256 channels,
- The second sub-layer uses 3×3 filters with 256 channels,
- The third sub-layer uses 1×1 filters with 1024 channels.

This module is repeated 36 times, allowing for a deeper learning of features in this layer. These repeated modules with residual connections are key to preventing the vanishing gradient problem in deep networks, making ResNet152 highly efficient for complex tasks [11].
In the **fifth layer** of the ResNet152 architecture, the central module comprises three types of sub-layers:

- The first sub-layer applies 1×1 filters with 512 channels,
- The second sub-layer uses 3×3 filters with 512 channels,
- The third sub-layer applies 1×1 filters with 2048 channels to the feature maps.

This module is repeated three times in the fifth layer, extracting deeper and more complex features from the data. After this, the output from the final convolutional layers undergoes **global average pooling**, reducing the feature maps to a fixed-size vector. Finally, a **fully connected layer** with a **softmax activation function** is used to predict the output class labels.
This deep architecture, with its combination of repeated residual blocks and skip connections, allows ResNet152 to efficiently learn high-level features without suffering from the degradation problem common in deep networks. The architecture has proven highly effective in image recognition tasks.

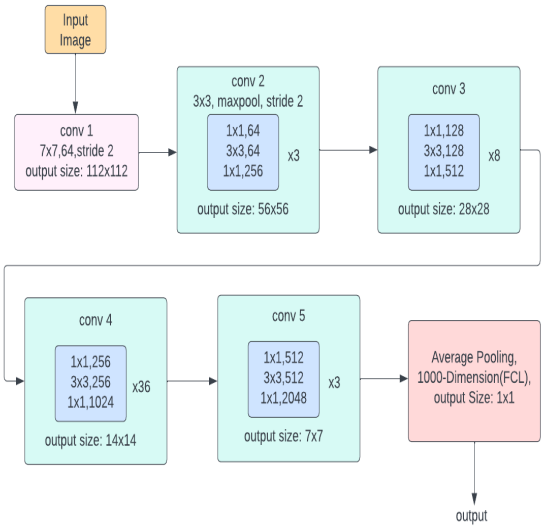

Figure 7: **ResNet152 model architecture**

3.2.6 MobileNetV2

MobileNetV2 is used to build a lightweight deep neural network and is based on a streamlined architecture, developed by Google researchers. MobileNetV2 is built based on the original MobileNet architecture to introduce some new features that enhance the model's accuracy and efficiency [20]. MobileNetV2 has 53 convolution layers, and these convolution layers are divided into two parts: 1x1 Convolution, and 3x3 Depthwise Convolution. The two main components of MobileNetV2 are Inverted Residual Block and Bottleneck Residual Block [10]. Each block in MobileNetV2 is composed of three distinct layers:

- The **first layer** is a convolutional layer with the activation function **ReLU6**, which helps prevent the issue of vanishing gradients while maintaining efficient computation.
- The **second layer** is a depthwise convolution layer, also using **ReLU6** as the activation function, which enables efficient filtering of the input feature maps with fewer parameters.
- The **third layer** is a pointwise convolutional layer, applied without activation (no non-linearity), allowing for the combination of the filtered features.

MobileNetV2 improves upon **MobileNetV1** and **ShuffleNet** by enhancing accuracy while reducing computational cost. The architecture's use of **inverted residuals** and **linear bottlenecks** allows for better efficiency in mobile and resource-constrained environments. Additionally, its customizable nature makes it adaptable to specific requirements, contributing to its reputation as a highly powerful and efficient CNN architecture for a wide range of tasks.

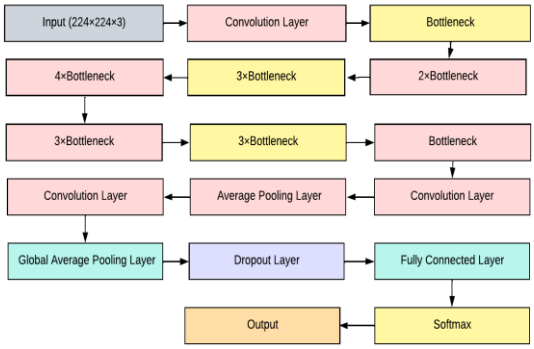

Figure 8: MobileNetV2 basic model architecture

3.2.7 Fine-tuned all the models

Fine-tuning is a transfer learning approach where a pre-trained neural network is used as a foundation for a new model. The weights of the pre-trained network are adjusted to suit a specific task. This method leverages pre-trained weights, which capture essential image features, allowing the model to learn more effectively and achieve higher accuracy with less training data. For our project, we applied pre-trained CNN models trained on the ImageNet dataset and customized architectures including **VGG19**, **DenseNet201**, **InceptionV3**, **Xception**, **ResNet152**, and **MobileNetV2** to enhance performance through fine-tuning.
Customization of Architectures for Fine-Tuning:

- **Pre-trained CNN as the base layer**: We used the pre-trained CNN models as the first layer of our custom model.
- **Flatten Layer**: After the CNN layers, we added a Flatten layer to convert the CNN's output into a one-dimensional feature vector, enabling it to pass

through the dense layers. A Dropout layer with a rate of 0.25 was added to reduce overfitting.

- **Dense Layers**: Two dense hidden layers were added, with 512 and 128 neurons, respectively. Both layers used a **ReLU** activation function and were followed by **BatchNormalization** layers to stabilize the network. Another Dropout layer with a rate of 0.46 was included to prevent overfitting.
- **Output Layer**: Finally, an output layer was added using a Dense layer with a softmax activation function for multi-class classification.

Fine-Tuning Process:

The fine-tuning was executed in two phases:

a) Training the Fully Connected Layers: In the first phase, we trained only the fully connected layers of the model, keeping the pre-trained convolutional layers frozen. The model was compiled using the **sparse categorical cross-entropy** loss function and the **Adam optimizer**. A checkpoint was set to save the best weights based on validation loss during this phase.

b) Fine-tuning the Entire Network: In the second phase, the entire pre-trained network was fine-tuned. The convolutional layers, previously frozen, were made trainable, and the model was recompiled with a lower learning rate to avoid significant weight changes in the pre-trained layers. A new checkpoint was defined to save the best weights during this phase.

Model Evaluation:

We trained the models on the **HAM10000** dataset using these six deep-learning architectures. After training, we analyzed the predictions and plotted the confusion matrix and ROC curve for each model. Key performance metrics such as **Precision**, **Recall**, **F1-score**, and **Accuracy** were computed. A comparative analysis of the accuracy across all models was performed to identify which model achieved the highest accuracy in detecting all types of skin lesions.

This approach enabled us to optimize the CNN models for skin cancer classification, achieving improved performance while maintaining computational efficiency.

4. EXPERIMENTAL RESULT

It is the most difficult task to find the appropriate metrics to assess and analyze the performance of classification systems. This section presents the experimental results and analysis of the models implemented on the HAM10000 dataset. The outcomes of six deep learning models VGG19, DenseNet201, InceptionV3, ResNet152, and MobileNetV2 are compared using some evaluation metrics such as specificity, sensitivity; accuracy, F-measure, and ROC curve [21].

In the context of skin cancer detection, sensitivity is an indicator that gives the aptitude of the model to correctly recognize the category and avoid false negatives, for example, which are cases where actinic keratosis is classified as Melanoma [14].

$$\text{Sensitivity (Recall)} = \frac{TP}{TP+FN} \quad (1)$$

Specificity quantifies the proportion of true negatives that are correctly detected by the model.

$$\text{Specificity} = \frac{TN}{TN+FP} \quad (2)$$

Precision measures the proportion of true positives among all the positive predictions the model has identified.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (3)$$

Accuracy is the overall preciseness of the model's predictions and is defined as the ratio of correctly classified instances to the total number of instances.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

F1-score is a harmonic mean of precision and recall, which measures the equity between the two metrics and expressed as,

$$\text{F1-score} = 2 \times \frac{Precision \times \text{Recall}}{Precision+Recall} \quad (5)$$

4.1 Training and Validation Accuracy

To assess how well the training and validation datasets perform within the deep learning model, we evaluate several key metrics, including training loss, training accuracy, validation loss, and validation accuracy. For this project, we have presented the training loss, training accuracy, validation loss, and validation accuracy across different epochs for the six models under consideration.

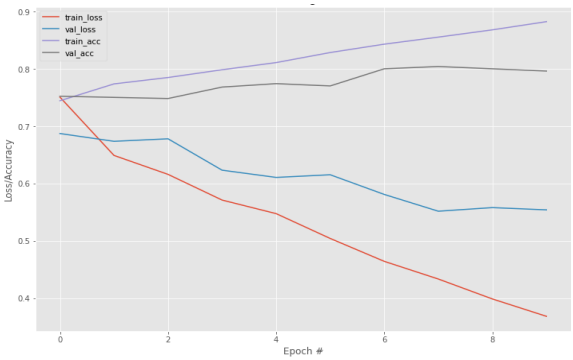

Figure 9: **Visualization of accuracy and loss curve for VGG19.**

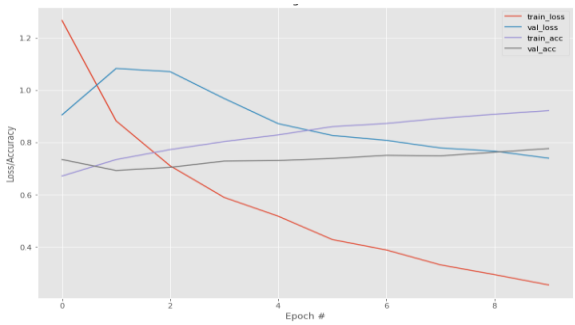

Figure 10: **Visualization of accuracy and loss curve for DenseNet201.**

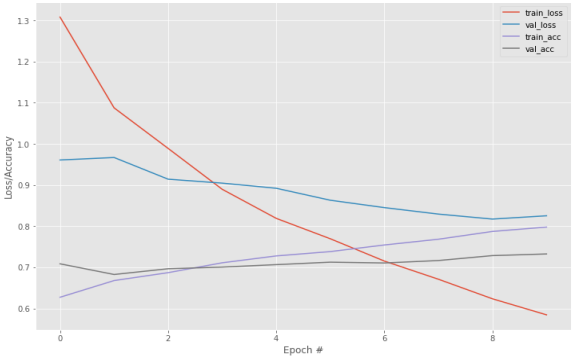

Figure 11: **Visualization of accuracy and loss curve for InceptionV3.**

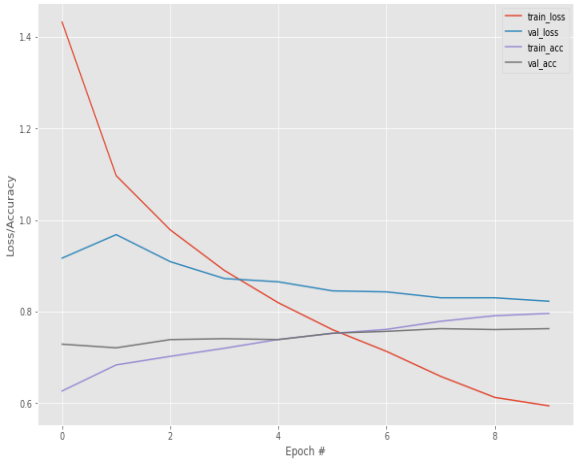

Figure 12: Visualization of accuracy and loss curve for Xception.

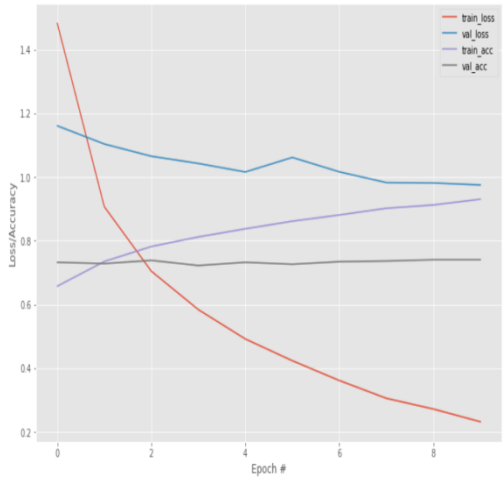

Figure 13: Visualization of accuracy and loss curve for ResNet152.

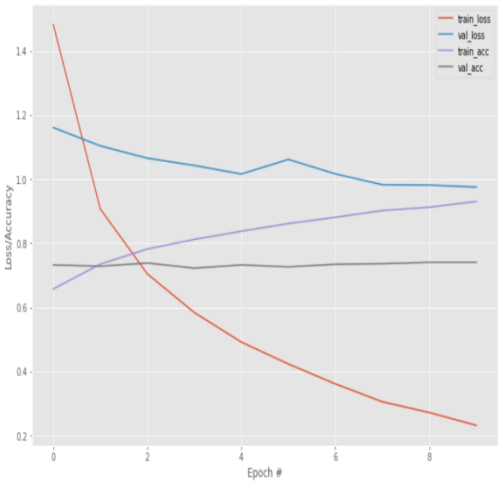

Figure 14: Visualization of accuracy and loss curve for MobileNetV2.

4.2 Evaluate Confusion Metrics

We analyze the performance of all five models by constructing confusion matrices for each model [22]. We can deduce true positive (TP), false positive (FP), true negative (TN), and false negative (FN) rates for each category in the dataset. We can calculate accuracy, precision, recall, and f1 score using these rates.

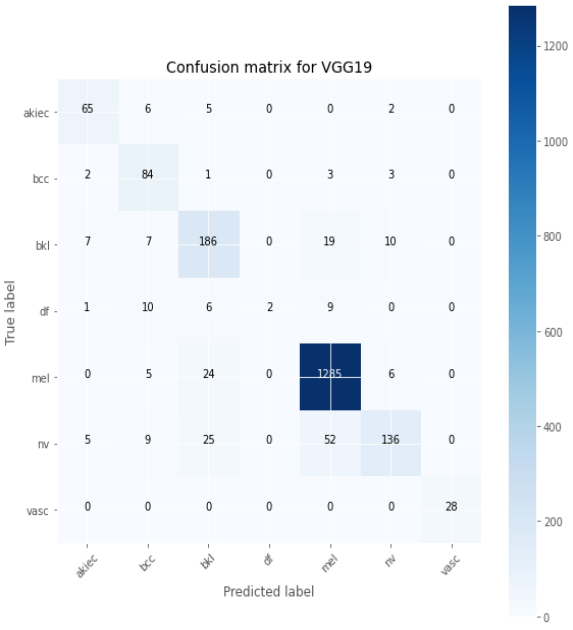

Figure 15: Confusion matrices for VGG19.

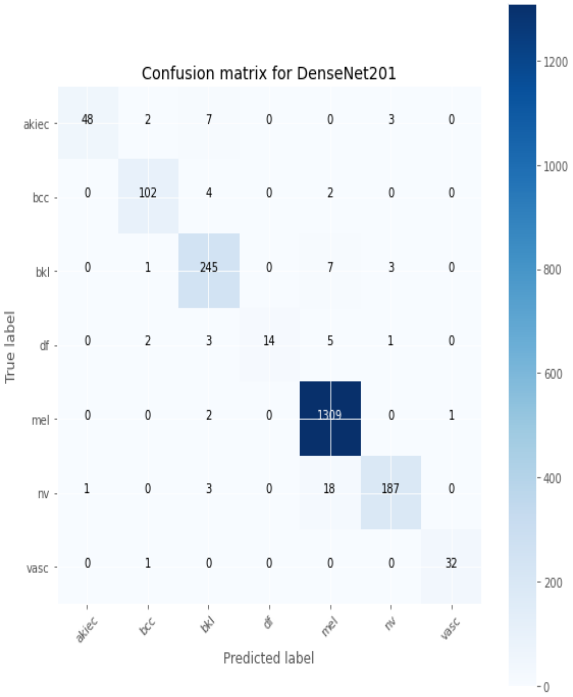

Figure 16: Confusion matrices for DenseNet201.

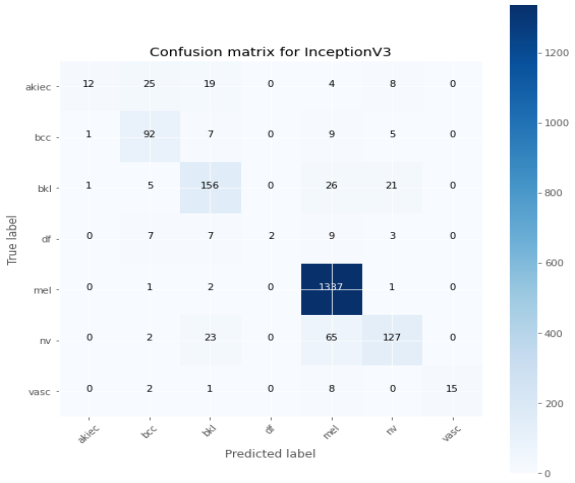

Figure 17: Confusion matrices for InceptionV3.

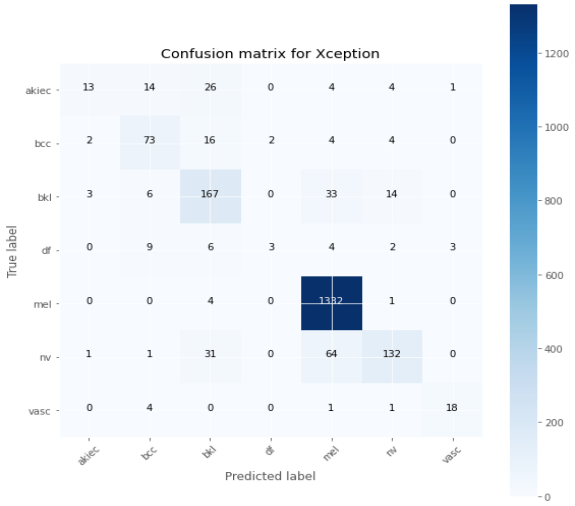

Figure 18: Confusion matrices for Xception.

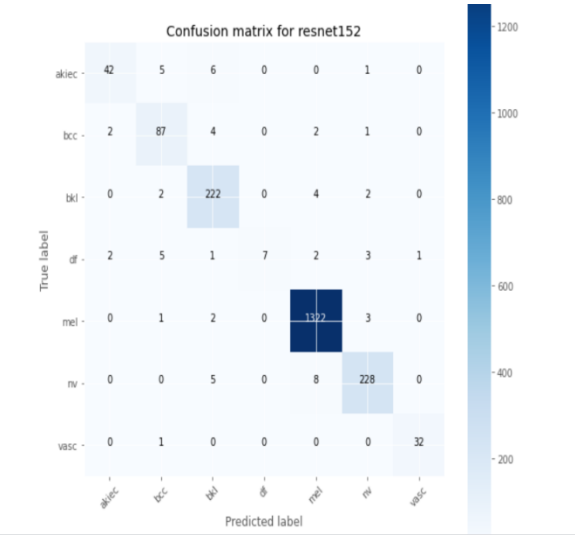

Figure 19: Confusion matrices for ResNet152.

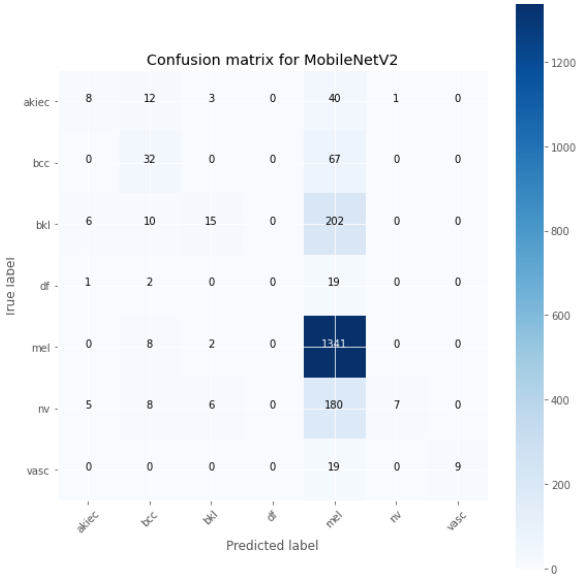

Figure 20: Confusion matrices for MobileNetV2.

We evaluated the performance of six models—VGG19, DenseNet201, InceptionV3, Xception, ResNet152, and MobileNetV2—for skin cancer classification across seven categories, using the performance metrics mentioned earlier. The accuracy achieved by these models is as follows: VGG19 (89%), DenseNet201 (97%), InceptionV3 (87%), Xception (85%), ResNet152 (96%), and MobileNetV2 (70%). Additionally, the weighted averages of precision, recall, and F1-score for each of these models have been assessed and are presented in Table 2.

Table 2: The Performance Analysis of Different Models.

Model	***Accuracy (%)***	***Weighted Average***		
		Precision (%)	***Recall (%)***	***F1-score (%)***
VGG19	89	90	89	88
DenseNet201	**97**	**97**	**97**	**97**
InceptionV3	87	87	87	85
Xception	85	83	85	83
ResNet152	96	96	96	96
MobileNetV2	70	69	70	61

It is evident that the DenseNet201 model has outperformed all other models in terms of accuracy, precision, recall, and F1-score. The weighted averages for DenseNet201 are 97% for precision, 97% for recall, and 97% for the F1-score, making it the best-performing model among the ones evaluated. Table 3 presents the specific values for each category in the DenseNet201 model, providing a clearer understanding of the performance for each class of skin cancer in the dataset.

Table 3: The class-wise performance analysis of the DenseNet201 model.

Categories	***Precision (%)***	***Recall (%)***	***F1-score (%)***
akiec	98	80	88
bcc	94	94	94
bkl	93	96	94
df	100	56	72
nv	98	100	99
vasc	96	89	93
mel	97	97	97
Weighted Average	97	97	97

4.3 AUC-ROC Curves Analytics.

The AUC-ROC curves for the different models provide a visual comparison of their performance. These curves illustrate the trade-off between the true positive rate (TPR) and false positive rate (FPR) across various classification thresholds. A higher AUC value indicates better performance, with the models demonstrating strong discrimination between positive and negative cases.

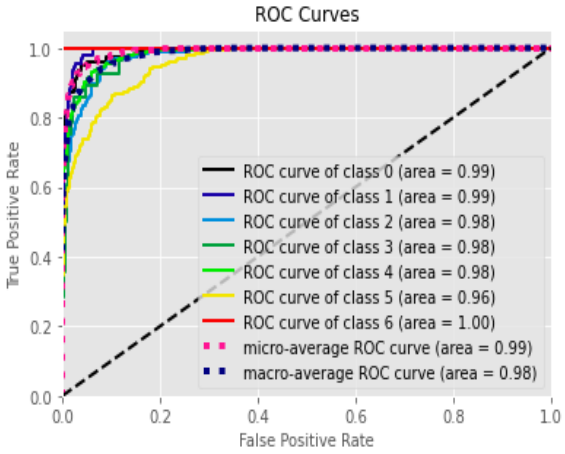

Figure 21: AUC-ROC Curve for VGG19

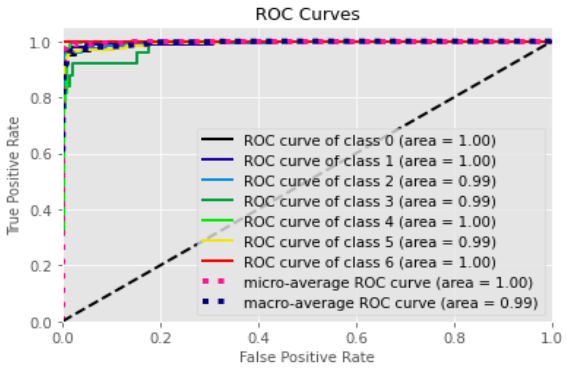

Figure 22: AUC-ROC Curve for DenseNet201.

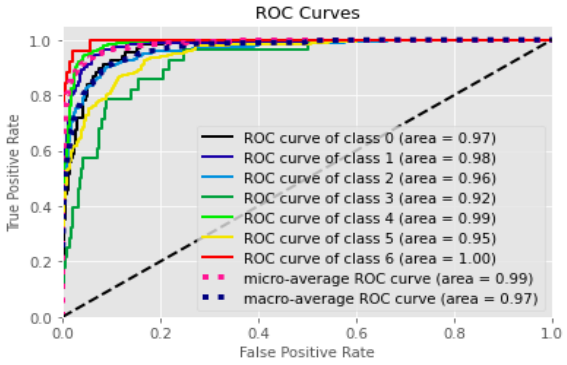

Figure 23: AUC-ROC Curve for InceptionV3.

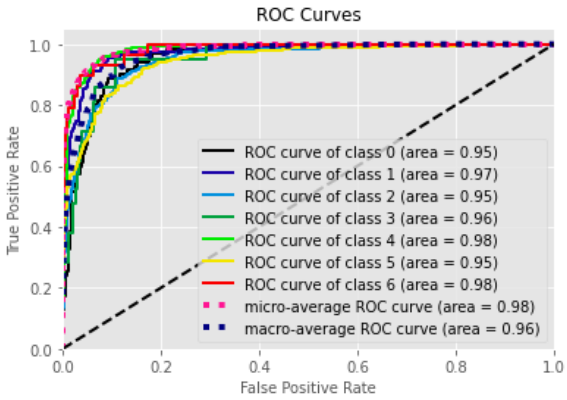

Figure 24: AUC-ROC Curve for Xception.

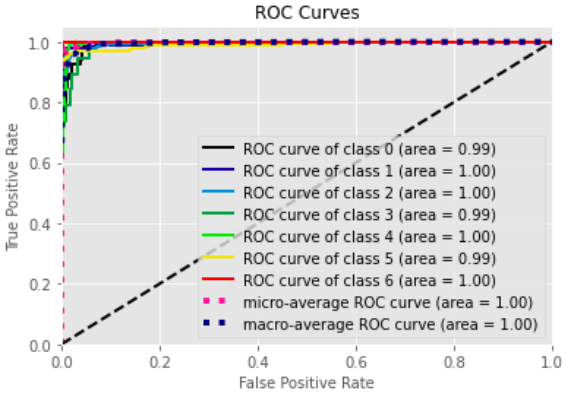

Figure 25: AUC-ROC Curve for ResNet152.

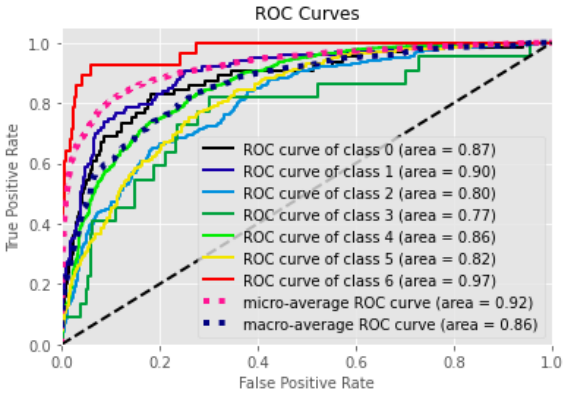

Figure 26: AUC-ROC Curve for MobileNetV2.

4.4 Mobile App Development

The mobile application developed for real-time skin cancer detection was built using Android studio to develop it as an android application which is widely available in most mobile devices, and TensorFlow Lite was utilized for model inference. The app features an intuitive interface where users can capture or upload images of skin lesion, and the integrated model provides immediate disease diagnosis. As the model is already integrated in the app it works completely offline, ensuring seamless recognition and result prediction even in remote areas. The integration of real-time data from the field allows for continuous monitoring and updates, making the tool highly effective for skin lesion [23].

4.4.1 Real-Time Testing with Mobile Application

To validate the model's practical applicability, it was integrated into a mobile app and tested in real-time scenarios. The goal was to develop an intuitive smartphone application that allows users to take or upload pictures and get detection results in real time [23]. The following observations were made:

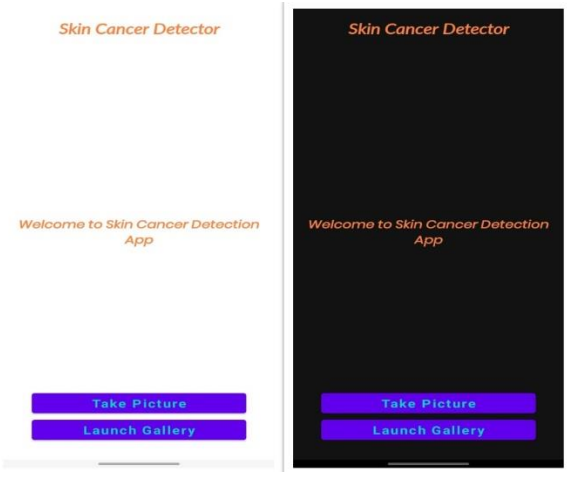

Figure 27: Welcome to the Skin Cancer Detection App.

Figure 28: Skin Cancer Detection Application.

5. CONCLUSION

Skin cancer is one of the most frequently diagnosed cancers worldwide, with an estimated 7,990 deaths from melanoma expected in 2023. Raising awareness about its risk factors and symptoms is vital. Deep learning models, particularly convolutional neural networks (CNNs), have shown exceptional performance on image datasets and can be applied for skin cancer detection in the medical field. In this study, six pre-trained CNN models—VGG19, DenseNet201, InceptionV3, Xception, ResNet152, and MobileNetV2—were evaluated on the HAM10000 dataset to classify seven types of skin lesions. The models were enhanced with additional convolutional and dense layers to improve their sensitivity to subtle features of skin cancer. Performance was assessed using metrics like accuracy, precision, recall, F1-score, and AUC. Among these models, DenseNet201 delivered the best overall performance across all metrics. However, some models, like MobileNetV2 and Xception, did not show notable improvements. These findings highlight the potential of certain CNN architectures, particularly DenseNet201, in the development of automated skin cancer detection tools that can support dermatologists in clinical practice. Future research could explore the use of more complex CNN architectures, such as stacked models like InceptionResNetV2 or ensemble approaches combining models like InceptionV3 and Xception, to further improve detection accuracy. Additionally, to develop automated medical treatment recommendations.

6. REFERENCES

[1] G. I. S. M. H. A. E. Laila Moataz, "Skin Cancer Diseases Classification using Deep Convolutional Neural Network with Transfer Learning Model," in *Journal of Physics*, 2021.

[2] P. E. Hari Kishan Kondaveeti, "Skin Cancer Classification using Transfer Learning," in *IEEE International Conference on Advent Trends inMultidisciplinary Research and Innovation (ICATMRI)*, 2020.

[3] J. V. T. T. D. Saket S. Chaturvedi, "A multi-class skin Cancer classification using deep convolutional neural networks," *Springer Science+Business Media*, pp. 1-2, 2020.

[4] V. M. M, "Melanoma Skin Cancer Detection using

Image Processing and Machine Learning," *International Journal of Trend in Scientific Research and Development - IJTSRD*, vol. 3, no. 4, pp. 780-781, 2019.

[5] B. A. A. N. B. P. G. R. H. A. M. K. S. Mohamed Yacin Sikkandar, "Deep learning based an automated skin lesion segmentation and intelligent classifcation model," *Springer-Verlag GmbH Germany, part of Springer Nature* , 2020.

[6] M. S. H. A. Noortaz Rezaoana, "Detection and Classification of Skin Cancer by Using a Parallel CNN Model," in *IEEE International Women in Engineering (WIE) Conference on Electrical and Computer*, 2020.

[7] X. Z. S. R. J. S. Kaiming He, "Deep Residual Learning for Image Recognition," *IEEE Xplore*, 2015.

[8] V. V. S. I. J. S. Christian Szegedy, "Rethinking the Inception Architecture for Computer Vision," *IEEE Xplore*, pp. 2820-2823.

[9] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," *Google, Inc*, 2017.

[10] A. H. M. Z. A. Z. L.-C. C. Mark Sandler, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *Google Inc.*, 2019.

[11] D. L. Z. L. ,. C. Long D. Nguyen, "Deep CNNs for microscopic image classification by exploiting transfer learning and feature concatenation," in *Research Gate*, 2018.

[12] Z. L. v. d. M. Q. W. Gao Huang, "Densely Connected Convolutional Networks," *IEEE Xpplore*, 2018.

[13] H. E. K. M. A. S. A. A. Youssef Filali, "Efficient fusion of handcrafted and pre-trained CNNs features to classify melanoma skin cancer," *Springer Science+Business Media, LLC, part of Springer Nature*, 2020.

[14] P. H. K. Sara Hosseinzadeh Kassania, "A comparative study of deep learning architectures on melanoma detection," *Elsevier*, 2019.

[15] F. F. R. H. R. M. A. S. F. A. a. S. W. M. A. Hossin, "Melanoma Skin Cancer Detection Using Deep Learning and Advance Regularizer," *International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 2020.

[16] S. S. A. Dr. J. Abdul Jaleel, "Artificial Neural Network Based Detection of Skin Cancer," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 2012.

[17] S. H. K. S. M. D. R. Z. A. Z. Mobeen R., "Classification of Skin Lesion by interference of Segmentation and Convolotion Neural Network," *Research Gate*, 2018.

[18] C. R. K. Philipp Tschandl, "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," *Sci Data*, 2018.

[19] Z. A. S. A. A. K. A. A. L. Karar Ali, "Multiclass skin cancer classification using EfficientNets – a first step towards preventing skin cancer," *Elsevier*, 2021.

[20] M. Z. B. C. D. K. W. T. W. M. A. H. A. Andrew G. Howard, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Application," *Google Inc.*, 2017.

[21] H. S. S. C. Mst Shapna Akter, "Multi-class Skin Cancer Classification Architecture Based on Deep Convolutional Neural Network," in *IEEE International Conference on Big Data*, 2022.

[22] U. S. 2. B. T. ,. E. A. N. ,. M. K. A. ,. A. K. K. Satin Jain 1, "Deep Learning-Based Transfer Learning for Classification of Skin Cancer," *Sensors (Basel)*, 2021.

[23] Md Aziz Hosen Foysal, Foyez Ahmed, Md Zahurul Haque. Multi-Class Plant Leaf Disease Detection: A CNN-based Approach with Mobile App Integration. International Journal of Computer Applications. 186, 41 (Sep 2024), 62-68. DOI=10.5120/ijca2024924026.