Skin Cancer (Melanoma) Detection Using Deep Learning Model

Arnob Chakraborty(0222210005101114)[1*],
Prottoy Dhar(0222210005101117)[2] and
Bristy Deb(0222210005101119)[3]

[1*]Department of Computer Science and Engineering, Premier University, Chattogram, Bangladesh.

*Corresponding author(s). E-mail(s):
chakrabortyarnobpu.cse41@gmail.com;
Contributing authors: prottoydhr@gmail.com;
bristy.puc1119@gmail.com;

Abstract

Skin cancer is still present as one of the most prevalent types of fatal malignancies in every part around the world, and due to long-term UV exposure skin cells grow prematurely and excessively resulting in an uncontrolled state. Early and precise identification is important for reducing the possibility of metastasis and improving survival rates. We propose a deep learning–based diagnosis framework to differentiate between benign and malignant dermoscopic images of skin lesions. Preprocessing The images with Melanoma Skin Cancer Dataset of 10000 images were converted to 1200 images, preprocessed, augmented and balanced (class weighting) since we encountered imbalanced dataset. A custom Convolutional Neural Network (CNN) was trained and compared against the VGG16, and MobileNetV2 architectures using transfer learning. Every model was trained using early stopping to avoid overfitting, and validated by validation metrics including accuracy, precision, recall, and F1-score. The CNN model reached around 84 percent accuracy and the VGG16 and MobileNetV2 based models obtained 87 and 82 percent accuracy.Thus, confirming that transfer learning is useful for dermatological image classification. These findings support the feasibility of deep learning models as tools to help dermatologists diagnose melanoma early, reduce biopsy rates and enable cost-effective tele-dermatology.

Keywords: Skin cancer, Melanoma, Dermoscopy, Deep learning, CNN, Transfer learning

1 Introduction and Problem Statement

Skin cancer is the most common types of cancer with potential lethal consequences worldwide due to the uncontrolled and abnormal megaprofiltration of skin cells. Chronic exposure to ultraviolet (UV) light from the sun or other sources greatly raises the likelihood of skin cancer, including melanoma, basal cell carcinoma and squamous cell carcinoma. Of these, melanoma is the most aggressive and lethal form which can spread quickly to other organs if not caught early. For instance, recent medical findings indicate that when detected and treated early, melanoma patients have a 90 percent cure rate. However, early diagnosis is not straightforward because benign lesions are visually similar to malignancies and due to the high subjectivity of clinical inspection as well as a lack of dermatologists available for consultation in remote or underserved areas.

Thus, the challenge is an automatic precise and low-cost skin cancer diagnosis system which could help doctors in early reporting of melanoma. Manual diagnosis by dermatologists in the traditional way, is time consuming and subjective, susceptible to errors and also heavily dependent on the experience of experts. To overcome this shortfall, deep learning and computer vision methods have gained increasing attention as effective means to analyze medical images with rapid and stable classification when being well-trained.

The goal of this project is to build a deep learning–powered skin lesion detection and classification system with the converted 1200 image dataset. The work uses three models, a customized Convolutional Neural Network (CNN) and two transfer learning models: VGG16 and MobileNetV2 to access performance in terms of accuracy, precision, recall and F1-score. The objectives are to:

- **Dermoscopy Image Preprocessing and Augmentation:** Counter data imbalance and enhance feature learning by applying preprocessing techniques and data augmentation on dermoscopic images.
- **Deep Learning Model Construction and Evaluation:** Construct and train multiple deep learning models for binary classification of skin lesions (*benign* vs. *malignant*).
- **Performance Comparison:** Examine and compare the predictive performance of a conventional Convolutional Neural Network (CNN) with transfer learning architectures such as VGG16 and MobileNetV2.

This project focuses solely on binary classification of Converted 1200 dermoscopic images. It does not do multi-class lesion classification or real-time mobile deployment, but is a basis for scalable diagnostic systems to augment dermatologists and democratize skin cancer screening to broader population.

2 Related Work

Recently, deep learning has emerged as a game-changer in medical image analysis, particularly for the early detection and classification of skin cancer. Traditional hand-crafted feature-based methods, such as those relying on texture, color, and shape, have been overshadowed by deep learning architectures like Convolutional Neural Networks (CNNs), which learn hierarchical features directly from dermoscopic images. Early investigations demonstrated that CNNs were at least comparable, and in many cases superior, to dermatologists in diagnostic accuracy, making the computational classification of large-scale melanoma datasets feasible [1, 2].

Garg *et al.* proposed a decision support system for skin cancer classification using CNNs on the HAM10000 dataset [3], emphasizing the importance of data preprocessing, augmentation, and model robustness by transfer learning [4]. Their proprietary CNN obtained a weighted F1-score of 0.77, and accuracy was upgraded to 90.5% with the ResNet50 through transfer learning. using the ResNet50 architecture. This work established that models pretrained on large-scale datasets such as ImageNet could substantially enhance performance in highly focused medical imaging applications. More recently, Sabir and Mehmood carried out a comprehensive study comparing CNN, ResNet-18, and EfficientNet-B0 models for binary melanoma classification[5]. They reported that EfficientNetB0 outperformed their selected architectures, CNN 80% and ResNet-18 87%, with an accuracy of 97%. They described this gain to compound scaling and squeeze-and-excitation modules which trade off network depth, width, and resolution for computational cost. In addition, sensitivities and specificities were 99% and 93%, respectively, thus placing the model within a clinically acceptable level of diagnostic performance. There are also other works in the same direction. Brinker et al. compared the abilities of CNN models and 157 dermatologists, the results showed that deep learning methods exceeded human abilities for recognizing melanomas [1]. Han *et al.* developed a multi-class CNN classifier trained on over 19,000 clinical images and achieved dermatologist-level performance in benign and malignant lesions[2]. Hosny *et al.*improved classification results by AlexNet transfer learning and more data augmentation, again corroborating the significance of these to enhanced accuracy on dermoscopic datasets [6]. Collectively, these studies reinforce that transfer learning and data augmentation are vital for achieving high classification accuracy when labeled medical data is limited. Despite these advancements, key challenges persist. Many high-performing models, such as ResNet and EfficientNet, are computationally expensive and require substantial resources, making real-time or mobile deployment difficult. Furthermore, there remains a research gap in direct comparisons between conventional CNNs and lightweight transfer learning architectures such as VGG16 and MobileNetV2 under identical preprocessing and evaluation conditions. Such comparative analyses are essential to determine optimal trade-offs between accuracy, inference speed, and hardware efficiency particularly for portable diagnostic tools and low-resource healthcare environments.

In order to reduce these gaps, the current study establishes and assesses a uniform framework for dermoscopic images binary classification (benign versus malignant) HAM10000 dataset [3]. It comprises three models: a hand-crafted CNN, a transfer learning architecture for VGG16 and a light-weight MobileNetV2 network. This work provides a reproducible benchmark with comparable accuracy-complexity-efficiency trade-offs of deep learning on a consistent protocol regarding preprocessing, augmentation, and evaluation protocols. This aids the larger goal of developing scalable, affordable, and automatic skin cancer detection for clinical and telemedicine settings.

3 Dataset

3.1 Source

This study uses the public dataset HAM10000 (Human Against Machine with 10,000 training images)[3], a human-annotated dermoscopic image dataset which has been widely used as a state-of-the-art benchmark skin lesion dataset. This was proposed by Tschandl et al. (2018) and published on Mendeley Data. The dataset contains 10,015 dermoscopic images of pigmented skin lesions taken within three different clinical settings in Austria and Australia. The data consisted of 1 diagnostic classes of skin images melanoma with the dimension of around $224 \times 224 \times 3$ for each image. The HAM10000 dataset is freely available for academic research, and distributed under the Creative Commons Attribution 4.0 International License (CC BY 4.0). In adherence to medical confidentiality and other ethical research protocols, all personal and identifying information have been stripped. An example permanent link to the dataset is at:

- **Mendeley Data:** https://data.mendeley.com/datasets/ggh6g39ps2/3
- **DOI:** https://doi.org/10.17632/ggh6g39ps2.3

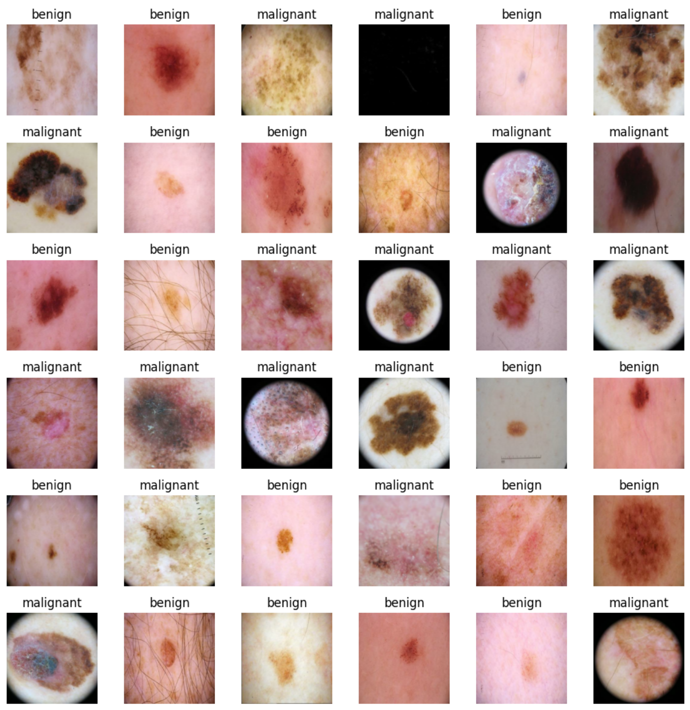


Fig. 1: Visualisation of Train and Test Images

3.2 Preprocessing, Exploratory Data Analysis (EDA), and Subsampling

To reduce computation time and allow to train the deep learning models to be trained quicker, the original size of the HAM10000 dataset (about 10,000 dermoscopic images) was downsampled to the working dataset of 1200 images. Before subsampling, a phase of exploratory data analysis (EDA) was performed in order to study dataset class distribution, visual features, and possible bias factors. The EDA consisted of analysing the distribution of the commonly occuring categories of the lesions, viewing images from the various classes, and validating properties of the images such as dimensions, aspect ratios and possible pixel intensity values. This analysis ascertained the presence of severe class imbalance in the original dataset with positive samples, particularly melanocytic nevi, vastly outnumbering the negative samples such as melanoma. From these observations, a subsampling strategy was implemented with the objective of minimizing imbalance by evenly sampling subpopulations from benign and malignant cases at the same time to keep diagnostic diversity. All the malignant samples were left untouched while the benign samples were downsized at a predetermined rate, to maintain clinically relevant binary distributions for classification (benign: malignant). The class imbalance is lesser in this resulting subset, making the training and evaluation metrics more interpretable and stable. Embeddings were constructed by resizing all selected images to 224×224 pixels(to match the input shape of CNNs, VGG16 and MobileNetv2). If the pair of files was grayscale and could have a non-standard formatting, then images were forced to 3 channel RGB. Pixel intensities were normalized to a common value range to stabilize model training and speed up gradient-based optimization. As the raw dermoscopic images may vary in luminance, colors, or scales, this preprocessing makes certain that the subsequent models leans towards the relevant textual and spatial patterns, excluding the unwarranted differences.

Downsampling and preprocessing the dataset enabled rapid experimentation, faster model convergence, and efficient architectural comparisons using standard GPU resources—an essential consideration for a course-level project. At the same time, the subset retained sufficient variability across lesion types, skin tones, and image acquisition conditions to support statistically meaningful performance evaluation. Future extensions of this work will apply the complete pipeline to the full HAM10000 dataset and integrate more advanced EDA and balancing strategies (e.g., stratified sampling or synthetic oversampling) to further assess model scalability, robustness, and generalization across all lesion categories.

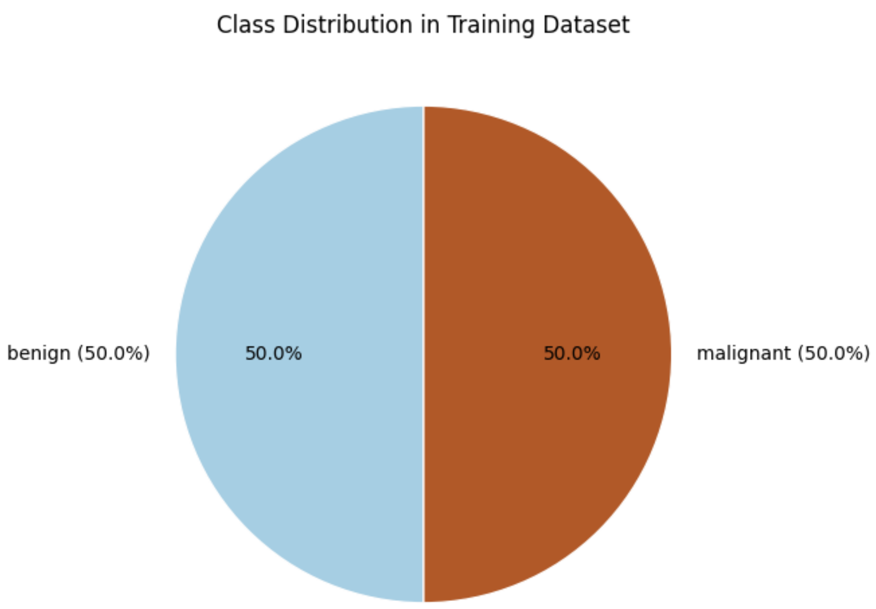


Fig. 2: Visualisation of Train and Test Images

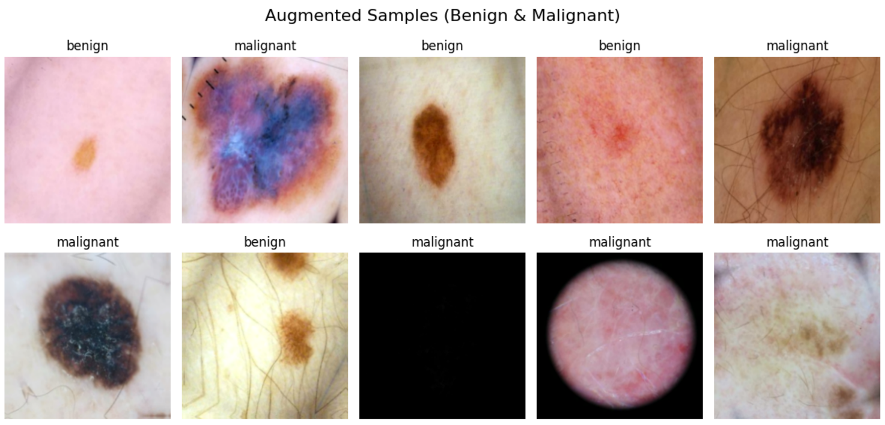


Fig. 3: Images of each Class after Data Augmentation

4 Methodology

This Section will emphasis over the methodology adopted for the detection task. Over all steps of the methodology is shown in figure 4.

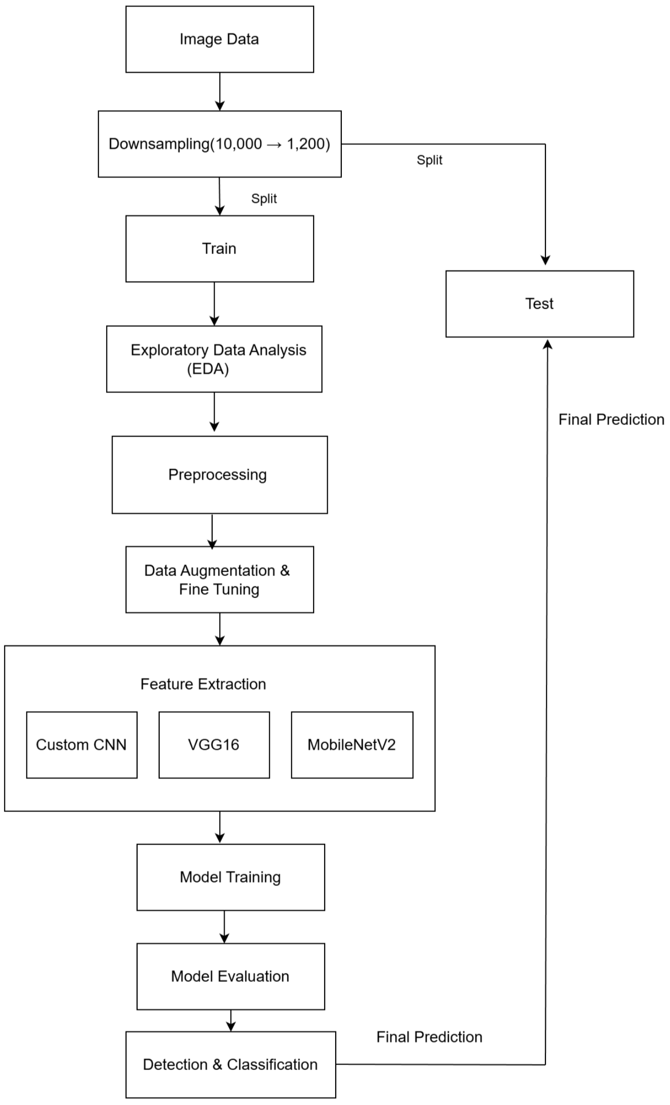


Fig. 4: Workflow of the proposed melanoma detection pipeline.

4.1 Model Architectures

In this work, three deep learning architectures—a custom CNN, VGG16, and MobileNetV2—are adopted for binary melanoma classification. Each model was selected to represent a distinct category of neural network design, ranging from lightweight, scratch-built architectures to heavyweight and mobile-friendly transfer learning approaches. Together, these models provide a comprehensive comparison of representational capacity, computational efficiency, and generalization performance under identical training conditions.

The custom CNN functions as a foundational baseline, learning feature representations solely from the 1,200-image subset without reliance on any external datasets. VGG16 on the other side is a state-of-the-art deep architecture widely applied in large-scale image recognition tasks because it learns numerous features hierarchically from the ImageNet dataset. MobileNetV2, on the other hand, includes modern design principles with efficiency in mind such as inverted residual blocks and depthwise separable convolutions, allowing it to produce a high level of accuracy with an order of magnitude fewer parameters. This work compares three architectures to investigate the effect on model performance related to (1)transfer learning (2) network depth (3) architectural complexity on two medical imaging tasks with small datasets. This contrast highlights the strengths and weaknesses of each model and provides insight into selecting appropriate architectures for practical clinical or mobile-health applications.

4.1.1 Custom CNN Architecture

CNNs are motivated by the physiological structure of the visual cortex in which neurons fire to cover a receptive field on the visual field. Just like in this biological process, CNNs work by learning patterns that lie within small regions of the input data. A CNN typically consists of three main types of layers, Convolutional, Pooling and Fully connected. We also add Batch Normalization and Dropout in our customized CNN structure to enhance training stability and reduce over-fitting.

Convolutional Layers:

The main building blocks of the network are the convolutional layers. In the convolutional layers, learnable filters (kernels) are recursively applied to slide through input images for extracting local patterns including edges, textures and shapes. We form our model with three convolutional layers using increasing filter size—32, 64, and 128 filters respectively. All convolution operations are performed with a kernel size of 3×3 and ReLU activation to enable the network to learn complex nonlinear patterns in dermoscopic images.

Batch Normalization:

At the end of every convolution, a batch normalization is utilized to stabilize learning. This layer normalizes feature maps and accelerates convergence and decreases sensitivity to weight initialization. It also relieves internal covariate shift of the model.

Max-Pooling Layers:

Each convolutional layer is followed by a max-pooling operation which down samples the spatial dimension of the feature map while preserving important information. Max pooling is performed by taking the maximum value in small regions (2×2 blocks in our model). This allows the network to focus on larger features and reduces the computational cost.

Dropout Layers:

Dropout is placed after the deeper convolutional layers and dense layers to minimize overfitting, since it's a super small dataset with only 1,200 images. Through randomly dropping out a portion of neurons during training (0.3, 0.4 and 0.5 dropout rate at different locations in the architecture), this model is generalized and less prone to overfitting on the training data which also avoids memorization of the input samples by some arbitrary hidden neurons with single set of parameters.

Fully Connected Layers:

Output feature maps are flattened to one dimension after the convolutions and pooling. This vector passes through a dense layer of 256 neurons which aims to tell the learnt features and learn useful high level patterns for classifying melanoma. Here the second dropout layer is used for further improvement of generalization. The last dense layer is followed by a softmax activation function giving rise of a probability for assignment in the output classes (benign or malignant).

In general, their CNN architecture model which involves convolution, normalization, pooling dropout and fully connected layers is able to extract discriminative features from the dermoscopic images.

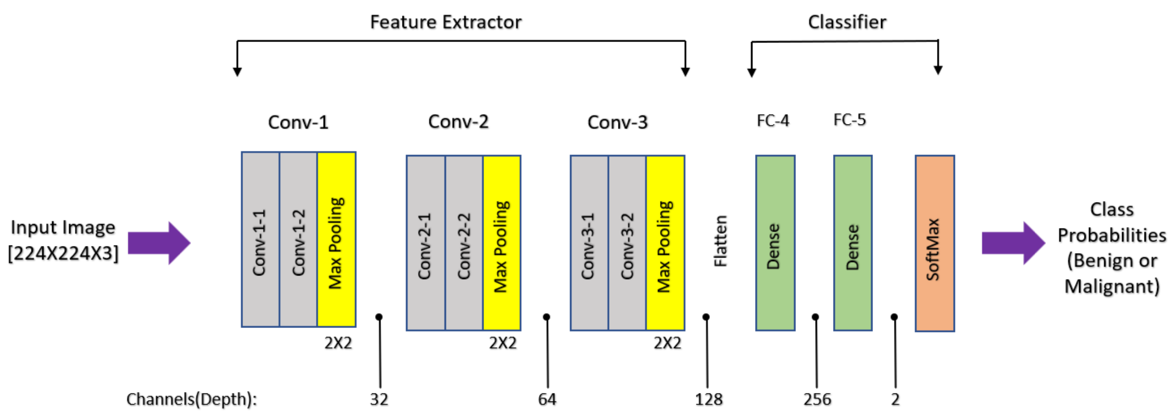


Fig. 5: CNN Architecture

Table 1: Layer-wise output shape and parameter count of the implemented custom CNN model.

Layer	**Type**	**Kernel / Units**	**Output Shape**	**Params**
Input	–	–	$(224, 224, 3)$	0
Conv2D-1	Conv	3×3, 32	$(222, 222, 32)$	896
BatchNorm-1	BatchNorm	–	$(222, 222, 32)$	128
MaxPool-1	MaxPooling	2×2	$(111, 111, 32)$	0
Conv2D-2	Conv	3×3, 64	$(109, 109, 64)$	18,496
BatchNorm-2	BatchNorm	–	$(109, 109, 64)$	256
MaxPool-2	MaxPooling	2×2	$(54, 54, 64)$	0
Dropout-1	Dropout	0.3	$(54, 54, 64)$	0
Conv2D-3	Conv	3×3, 128	$(52, 52, 128)$	73,856
BatchNorm-3	BatchNorm	–	$(52, 52, 128)$	512
MaxPool-3	MaxPooling	2×2	$(26, 26, 128)$	0
Dropout-2	Dropout	0.4	$(26, 26, 128)$	0
Flatten	Flatten	–	(86,528)	0
Dense-1	Dense	256 units	(256)	22,151,424
Dropout-3	Dropout	0.5	(256)	0
Dense-2	Dense	2 units (Softmax)	(2)	514
			Total Trainable Parameters	22,245,634
			Total Non-Trainable Parameters	448
			Total Parameters	**66,737,352**

The table shows the full block-wise architecture of our own CNN based on Convolutional Neural Network (CNN) structure that developed for binary classification of melanoma. Each layer, the type of operation (convolution or pooling), kernel size or number of units, output feature-map dimensions and exact number of trainable parameters are detailed in the table. This decomposition gives detailed insights on how the model processes input dermoscopic image and the growth in parameter-count throughout the network.
It starts from a $224 \times 224 \times 3$ image, which is progressively processed by three convolutional blocks. Every block is a convolutional layer followed by batch normalization and max pooling. The first block is meant to generate low-level feature maps where we extract 32 such feature maps and reduce the spatial resolution to 111×111. The depth is further increased to 64 and 128 in the second and third blocks and resolution is down-sampled by pooling. The use of dropout layers after the second and third pooling stages reduces the problem of overfitting by applying a random deactivation on a certain fraction of neurons during training.
Following the convolutional feature extractor, the output is vectorised to an 86,528 element vector and fed into the fully connected classifier. The use of a dense layer of 256 units is due to the most parameters (which account for nearly 23 million in total) it receives, given the high dimensionality of the flattened feature vector. A last dropout layer also contributes to the generalization success, and the final softmax layer outputs two class probabilities such as non-macres and mass lesions.

The total model has 66,737,352 parameters, 22,245,634 of them are trainable when batch normalization statistics are frozen in the case. This detailed tabular summary calls attention to the computational overhead for each component, and offers insight into how network architecture relates to learning capacity.

4.1.2 VGG16 Architecture

In this work, the VGG16 architecture is employed in transfer learning mode, for the classification of dermoscopic images into benign and melanoma categories. VGG16 is an extensive convolutional neural network pretrained on ImageNet, which includes more than one million natural images in thousand classes. Its underlying power lies in its very simple yet highly effective design that is composed of the whole network being made out of stacks of small 3×3 convolution filters as ordered in depth by five convolutional building blocks.

For our classification model, we will load the VGG16 architecture with weights pre-trained on ImageNet but without the top of the network (i.e., `include_top=False`) so that only features are extracted from VGG16. The base model should not be updated (`base_model.trainable = False`), which virtually causes all the convolutional weights to be non-trainable. It guarantees that the network inherits those generalized visual features learned from ImageNet data such as edges, corners and textures and shapes which are also helpful for analyzing dermoscopic images.

The output of the pre-trained VGG16 base is either passed through a GAP layer (flatten=true) which reduces the spatial extent of each channel in the feature maps to 1 by averaging over all pixels. GAP cuts down trainable parameters compared to fully-connected top layer, still having strong representational power making it apt for modest datasets like the 1,200 images subset employed in this work.

To avoid overfitting, we include a Dropout layer (rate =0.5) after the GAP layer. This randomly deactivates half of the neurons in each training step, which enhances generalization and lessens dependence on any particular activation pathway. The final-layer output is a Dense layer having Softmax activation that is equal to the number of target classes that we have (which in this case, are two). This layer transforms the properties learned from deep layers into predictive probabilities for “benign” and “malignant” skin lesions.

In general, the transfer learning mechanism of VGG16-based architecture can effectively utilize deep pre-trained hierarchical features and learns only a tiny classification head dedicated to melanoma detection. This leads to faster convergence as well as better accuracy and fewer overfitting than training a deep model from scratch.

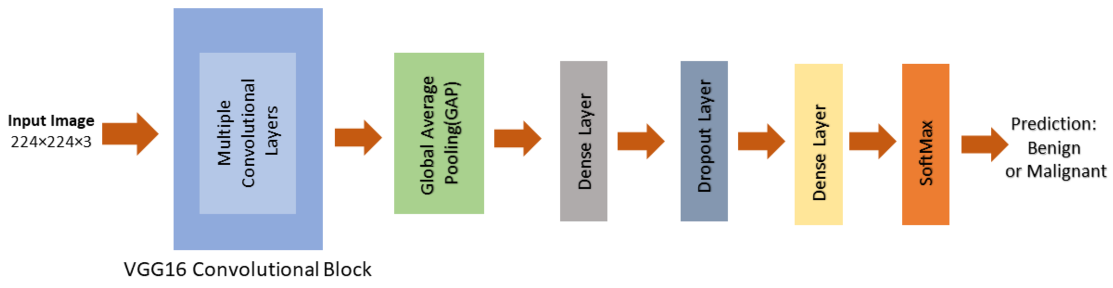

Input Image
224×224×3
Multiple Convolutional Layers
VGG16 Convolutional Block
Global Average Pooling(GAP)
Dense Layer
Dropout Layer
Dense Layer
SoftMax
Prediction: Benign or Malignant

Fig. 6: VGG16 Architecture

- **VGG16-Based Transfer Learning Architecture**
- **Input Layer:**
  - Input: $224 \times 224 \times 3$ (RGB skin image)
- **Feature Extractor:**
  - **VGG16 Convolutional Blocks:**
    - ∗ VGG16: The feature extraction part as described in VGG16 paper with ImageNet train dataset. This consists of:
    - ∗ Several convolutional layers using small filters of size 3×3
    - ∗ ReLU activations after each convolution.
    - ∗ Down-sample the feature maps using MaxPooling layers, which are inserted after every, and a few convolutional layers.
    - ∗ It gives an output in the form of a few feature maps which are the basic structure of VGG16 that has acquired the general features of the picture like edges, textures, and so on.
- **Global Average Pooling (GAP):**
  - GAP is utilized to compress the feature maps to a 1D vector after the VGG16 convolutional base, summarizing feature information into one value per feature map.
- **Dense Layer:**
  - It is a Dense layer with 256 units followed by a ReLU activation. This layer allows the model to take the 1D feature vector outputted from the GAP layer and learn complex patterns relating to melanoma classification.
- **Dropout Layer:**
  - There is an example of adding a Dropout layer with a 50% rate to prevent overfitting by randomly ignoring 50% of the neurons during training.

- **Output Layer:**
  - The last layer provides a Dense layer with 2 units because there are two output classes: "benign" and "malignant".
  - It uses softmax activation to output the class probabilities.

4.1.3 MobileNetV2 Architecture

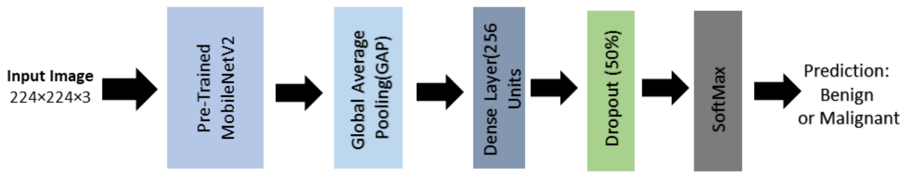


Fig. 7: MobileNetV2 Architecture

This article utilizes the MobileNetV2 architecture within a transfer learning framework to classify dermoscopic images into two categories: benign and malignant.

- MobileNetV2 is a highly new and powerful convolutional neural network. This makes it a smart choice for places where computers and mobile devices don't have a lot of power.
- Depthwise separable convolutions and inverted residuals are the two main parts of MobileNetV2's design. These let you cut down on the number of calculations and parameters needed without hurting performance. This design enables MobileNetV2 achieve a good balance between being very precise and not costing too much to run. This is handy for things like sorting photographs, especially when the datasets are small.
- We use the pretrained MobileNetV2 model without the top classification layers (`include_top=False`) for transfer learning. This lets the model use the great feature extraction skills it learned from ImageNet without having to train the whole network from scratch. We set the convolutional layers' weights to `base_model.trainable=False` so that the broad visual properties gained from the large ImageNet dataset, like edges, textures, and shapes, can still be used to classify melanoma.
- The Global Average Pooling (GAP) layer sends the feature maps from the base model via the convolutional layers. GAP makes each feature map smaller by putting them all together into one value. This step changes the high-dimensional input into a small 1D vector. It preserves the most important information and reduces down on the number of parameters that need to be educated.
- Then, the output from the GAP layer goes to a fully connected (dense) layer with 256 neurons that uses the ReLU activation function. This layer learns complicated

patterns and representations that help tell the difference between good and bad lesions. A Dropout layer with a 50% dropout rate is used to stop overfitting. This randomly turns off half of the neurons while the model is learning so that it can learn how to use what it has learnt in new scenarios.

- There are two neurons in the softmax output layer at the conclusion of the model, one for each class: "benign" and "malignant." This softmax layer informs the model about their probability to belong to each class. Here is how it comes to its final prediction.
- We employ the deep, pretrained convolutional base of MobileNetV2 with a lightweight classification head that has a single dense layer. This is called transfer learning. This approach is faster, more stable and less liable to overfit than building a deep network from scratch. This is particularly an issue when you only have a small number of data points, for example the 1,200 dermoscopic images used in this study.
- The last layer of the model is a softmax output layer with two neurons, one for each class: "benign" and "malignant." This softmax layer shows the chances that the input image belongs to each class. This is how the model makes its final prediction.
- We apply transfer learning with MobileNetV2 to train only the compact, efficient classification head on top of its deep, pretrained convolutional base. This method builds the model faster, more accurately, and with less chance of overfitting than beginning from scratch with a deep network. This is especially true when you just have a small amount of data, as the 1,200 dermoscopic images used in this study.

4.2 Hyperparameters

- **Batch Size**
  - The batch size is the number of training samples used in a single iteration of model learning. Even though it is not specified, the default value for a batch size is typically 32. This size works well in a large number of deep learning cases because it provides an adequate balance between computational effectiveness and speed of the model's convergence.
- **Learning Rate**
  - The learning rate is a vital hyperparameter that controls the size of each step in optimizing the neural network parameters.The Adam optimizer is used with a learning rate set to 0.001. As an adaptive parameter, ships with the Adam optimizer, so it automatically shrinks once the large gradients come in. The concept of learning rate is a methodology that informs how often weights should be updated in the learning process.
- **The Learning Rate Schedule**
  - The learning rate is constant over all epochs in the training. It is not possible to determine whether the learning rate decreases over epochs without this specific information, but an example of schedule would be doing so after each epoch to

gradually refine the already obtained model. Since the code did not have this specific command, we stick with the initial rate throughout the training process.

- **Weight Decay**
  - Weight decay is a regularly used regularization method that is particularly beneficial in large and complicated models to avoid overfitting. It is not yet included, but it can be done easily by adjusting the optimizer's parameters.
- **The Number of Training Epochs**
  - The training process will run for 20 epochs in this case. If before this time, the required level of accuracy is reached, the programmed early stopping functionality will terminate the operation without waste of computational resources.
- **Dropout**
  - The concept of dropout is used with varying rates included at different points of the model:
    - A 0.3 dropout after the second pooling layer,
    - A 0.4 dropout after the third pooling layer,
    - And a 0.5 dropout after a dense layer with 256 units are used in your case.
- **Warm-up**
  - The warm-up is not included because its purpose is specifically to set the learning rate to its first value.
- **Early Stopping**
  - Early stopping is implemented to avoid the waste of computational resources inevitable in trivial model training without termination. The validation loss will be observed with the early stopping callback, and the training will continue for 3 epochs without positive results patience of 3. Finally, the best weights will be reused, which means the training stops the moment validation loss value starts rising.

4.3 Fine-Tuning Details

- **Custom CNN**
- **Initialization:**
  - The conventional CNN model starts with random weights. Since this model isn't transfer learning based (no network beforehand), all of the layers are randomly initialized. The weights are changed during training according to the backpropagated error, instead of using an external pretrained feature set.
- **Frozen Layers:**
  - Zero frozen layers as it is a custom model. All the layers (convolutional and full connected) of the model are initially trainable and their weights will get updated during the training.

- **Adapters/LoRA:**
  - This model does not use adapters or LoRA (Low-Rank Adaptation). This model does not leverage any pretrained components and therefore all the weights are learned from scratch (including the additional adapters if present).
- **Early Stopping:**
  - The training process consists of early stopping in case overfitting occurs. Training stops when the validation loss does not decrease for 3 epochs, and the best weights (the ones that achieved the best validation loss) are restored. It ensures the model is not memorising and saves time and computation.
- **Mixed Precision:**
  - Our custom CNN model does not using mixed precision training. Keras uses standard floating-point precision (32-bit) floating-point numbers to train the model by default.
- **Gradient Accumulation:**
  - Gradient accumulation is not implemented. It calculates the gradients and updates the model weights after each batch.
- **VGG16 (Transfer Learning)**
- **Initialization:**
  - I used the VGG16 model as my convolution base which is initialized with pre-trained ImageNet weights. Specifically, all convolutional layers are initialized with weights trained on a large scale dataset, offering those models generalized feature extraction gained from millions of images. In contrast, the fully connected layers are not included and retrained from scratch.
- **Frozen Layers:**
  - We freeze the convolutional layers of VGG16 (i.e. `base_model.trainable = False`). In other words, the weights in the convolutional layers do not get updated during training. The new layers we added on top (the fully connected layer and the softmax output layer) are the only ones that we can train.
- **Adapters/LoRA:**
  - There is no application of adapters or LoRA techniques in the VGG16 transfer learning setup. This keeps the convolutional layers frozen (using pre-trained weights) but only fine-tunes the top layers.

- **Early Stopping:**
  - Early stopping is utilized for monitoring validation loss during training. If the validation loss does not improve for the last 3 epochs (or patience of 3), training stops prematurely and the model weights are restored to avoid overfitting.
- **Mixed Precision:**
  - The VGG16 model does not use mixed precision training. It trains using standard 32-bit precision, the default for Keras models.
- **Gradient Accumulation:**
  - Gradient accumulation is not applied. The model does not use any gradient accumulation to simulate a larger batch size and updates weights over each batch.
- **MobileNetV2 (Transfer Learning)**
- **Initialization:**
  - Similar to VGG16, MobileNetV2 comes with ImageNet pre-trained weights. For transfer learning, the convolutional base (the feature extractor component) is frozen so that the model can leverage the generalized features learned from ImageNet data. The classification head, Dense layers are initialized from random weights and trained exclusively for this melanoma classification task.
- **Frozen Layers:**
  - We freeze the convolutional layers of MobileNetV2 (`base_model.trainable = False`). This keeps MobileNetV2's feature extraction capabilities because all we will train are the newly added classification layers. The convolutional base shut off for training, which means the weights in the convolutional base is not adjusted.
- **Adapters/LoRA:**
  - The MobileNetV2 model does not use Adapters or LoRA. It trains the new top layers while keeping the weights from the already trained base frozen (the base).
- **Early Stopping:**
  - We use early stopping to stop the training process if the validation loss does not improve for 3 consecutive epochs. This assists in preventing overfitting and we save the best weights, that is when the model performed best on validation set.
- **Mixed Precision:**
  - This model is not using mixed precision training. As with VGG16, MobileNetV2 uses standard 32-bit floating-point precision, sufficient for almost any task unless specially benchmarked for speed.
- **Gradient Accumulation:**
  - MobileNetV2 training process does not use gradient accumulation. A custom Keras model updates the weights after every batch, which is the default set up in Keras for smaller batch size.

5 Training Procedure

5.1 Losses

- **The goal for all the models (Custom CNN, VGG16 and MobileNetV2) in this work is to classify dermoscopic images via binary classification into one of two classes: "benign" and "malignant."**
  - Categorical cross-entropy is used as the loss function in all models for this task, which is a general advocate for multi-class classification problems, even though we only have two classes in this case. The categorical cross-entropy loss computes how well the softmax output class probabilities match the true class labels, which helps tune the model weights to minimize the difference during training.
- **All models use class weighting to deal with class imbalance.**
  - Class weights are used to weight the training on underrepresented class (malignant lesions) heavier. This allows the model to overcome the situation that benign lesions might be the majority cause in the dataset; therefore, it does not bias the prediction towards the majority class. The class weights are computed using the two classes present in the particular dataset and applied within the training of the model.
- **In this setup, we do not implement label smoothing.**
  - Finally, label smoothing is a technique that slightly adjusts the target labels for the model to be less certain about its predictions so that its predictions are potentially more generalizable. Nevertheless for this model labels are hard (0 and 1).
- **The model is not in multi-task learning (though it is in multi-task weights), where the main task is still its binary classification (the second-task multi-class label).**
  - In contrast, in a multi-task learning setting, you would normally sum up further auxiliary losses, weighted by likely different non-zero integer values. Also, as only classification happens here, there are no other tasks to train and no loss weights either.
- **Altogether, categorical cross-entropy loss, class weighting, and a straightforward binary classification task together enable the models to quickly learn useful dividing lines between benign and malignant skin lesions.**

5.2 Optimizers

- **The Adam optimizer is utilized for training for all three models, including Custom CNN, VGG16, and MobileNetV2.**
  - Adam — An efficient adaptive optimizer that has shown to combine the advantages of AdaGrad and RMSProp, and can handle sparse gradients and adapt

learning rate on a per parameter basis. In this case, default parameters for Adam are used which include the following:

- **Momentum (betas):**
  - The Adam optimizer has two momentum terms that it uses, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. These values help increase stability to the moving averages of the gradient and squared gradient and lead to faster convergence.
- **Learning Rate:**
  - The learning rate is 0.001 by default for Adam in many deep learning tasks.
- **Schedulers:**
  - Models do not use any learning rate schedulers. The learning rate is constant in training unless we change it.
- **Gradient Clipping:**
  - In none of the models gradient clipping is adopted. This technique is only used to prevent exploding gradients because of a threshold in backpropagation and further training is conducted without, which is not the case here.

5.3 Seeds and Reproducibility

- **The models used in this work (Custom CNN, VGG16 and MobileNetV2) are trained with a focus on reproducibility.**
  - All random seeds are fixed so that repeated runs give consistent results. Making seed constant will make the model initialization deterministic, data shuffling used for preparation deterministic and any random process in training deterministic. Such randomisation ensures that the randomness of the data allows for consistent results within different experiments, thus reducing variability. We enable the determinism flags to keep the training fixed with respect to randomness.
- **Times shows the models are run only once for the specified number of runs (only one training process for each configuration).**
  - Checkpointing is used to save the best model weights during training to monitor the process and prevent overfitting. This means you can pause the training and resume it later without losing your progress. Also, capture key metrics during training, such as accuracy and loss values via logging to store, for example, a detailed list of the model performance in logs. This guarantees hyperparameter and model structure changes are tracked and can be assessed for the effect they had on performance.

5.4 Environment

- **In this work, training of the models was performed in Google Colab, which provided us a free GPU (T4) for accelerated processing.**
  - This hardware configuration allowed it to process the computational pressure required to train deep learning models effectively. During the training, the batch sizes and model architectures all fitted nicely in the GPU memory with no hassle. As for software, the models were implemented in standard deep learning frameworks (mainly TensorFlow and Keras), which come with most dependencies and appropriate versions already installed in the Colab environment. GPU acceleration with CUDA — the model training with the data is optimized with CUDA, and those data are parallel to process together. All the experiments ran in an environment based on Google Colab, a cloud-based interface for running Jupyter notebooks allowing an easy access to the GPU with the same environment for everybody.

6 Results

Sr. No	Model	Accuracy
1.	Custom CNN	84%
2.	VGG16	**87%**
3.	MobileNetV2	82%

Table 2: Model Accuracy for Custom CNN, VGG16, and MobileNetV2

Classification Reports

MobileNetV2 Classification Report

Precision	Recall	F1-Score	Support	
Benign	0.78	0.90	0.84	120
Malignant	0.88	0.75	0.81	120
Accuracy			0.82	240
Macro avg	0.83	0.82	0.82	240
Weighted avg	0.83	0.82	0.82	240

Table 3: MobileNetV2 Classification Report

CNN Classification Report

Precision	Recall	F1-Score	Support	
Benign	0.77	0.97	0.86	120
Malignant	0.96	0.72	0.82	120
Accuracy			0.84	240
Macro avg	0.86	0.84	0.84	240
Weighted avg	0.86	0.84	0.84	240

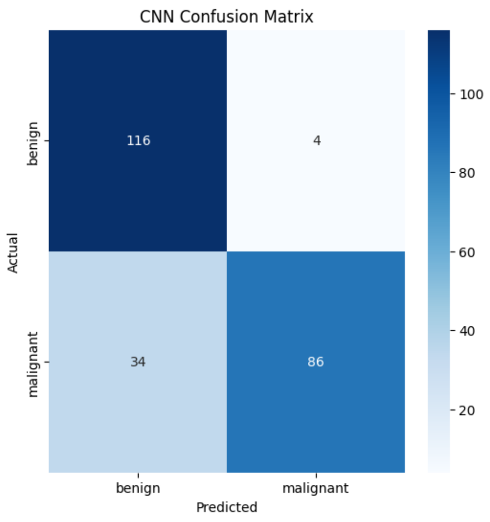
Table 4: CNN Classification Report

VGG16 Classification Report

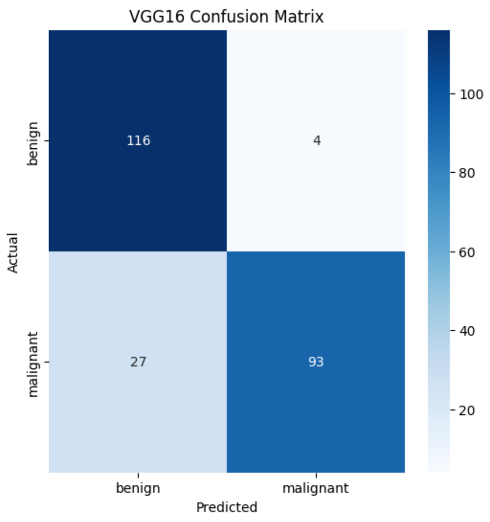
Precision	Recall	F1-Score	Support	
Benign	0.81	0.97	0.88	120
Malignant	0.96	0.78	0.86	120
Accuracy			0.87	240
Macro avg	0.88	0.87	0.87	240
Weighted avg	0.88	0.87	0.87	240

Table 5: VGG16 Classification Report

6.1 Confusion Metrics

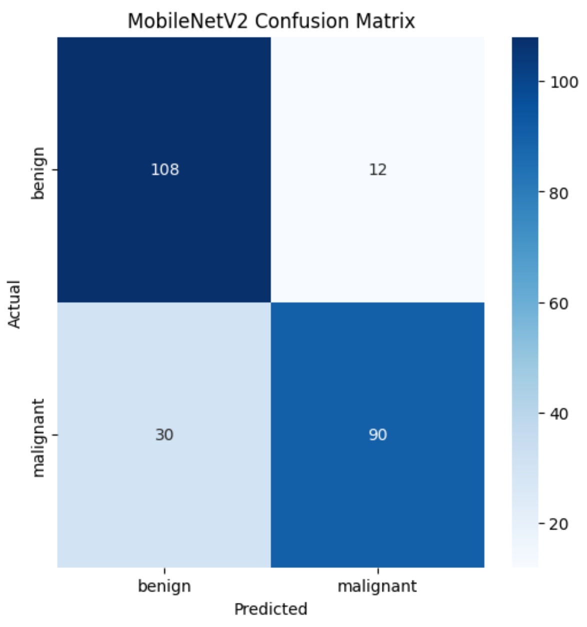


(a) CNN Confusion Matrix



(b) VGG16 Confusion Matrix

Fig. 8: Confusion Matrices for CNN and VGG16



(a) MobileNetV2 Confusion Matrix

Fig. 9: Confusion Matrix for MobileNetV2

6.2 Learning Curves and Additional Figures

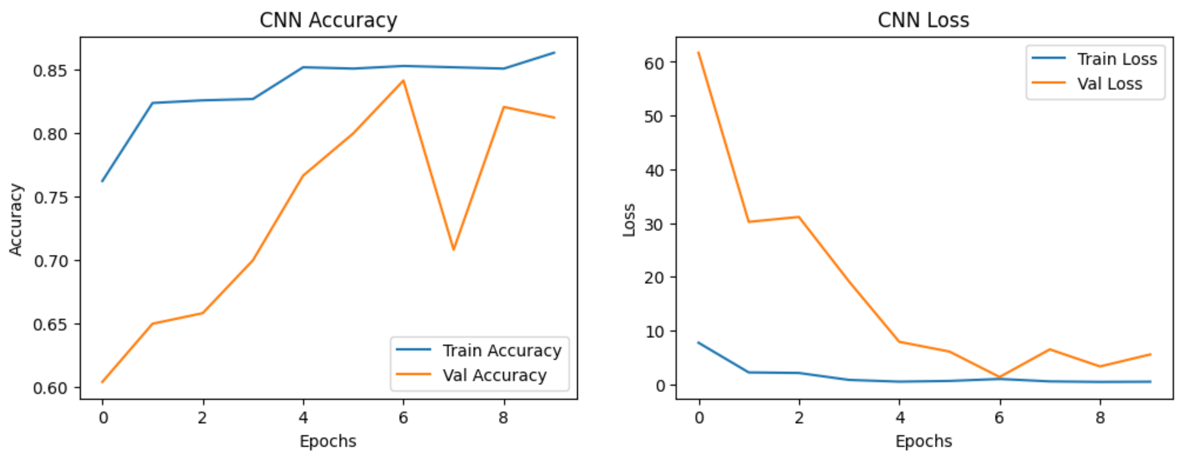


Fig. 10: CNN Accuracy and Loss Curve

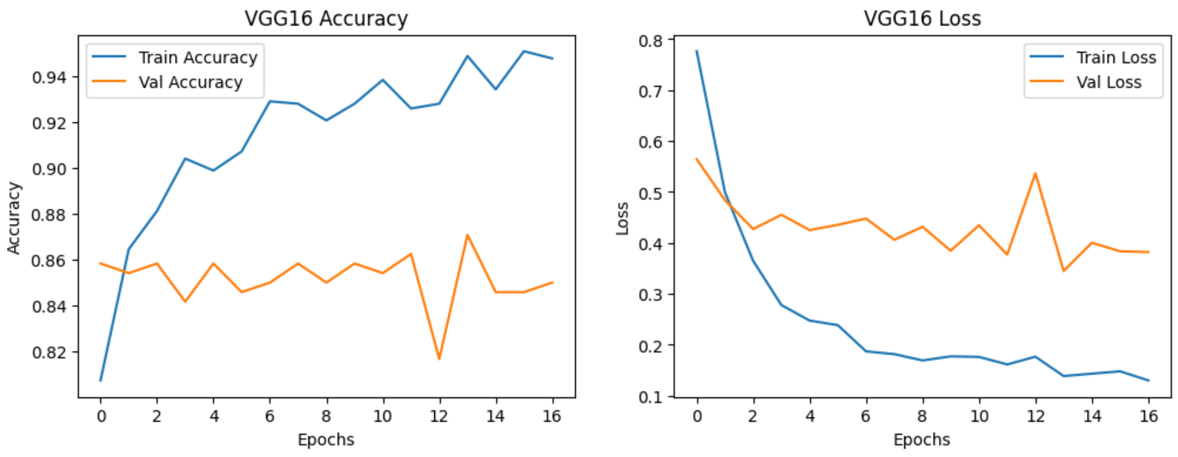


Fig. 11: VGG16 Accuracy and Loss Curve

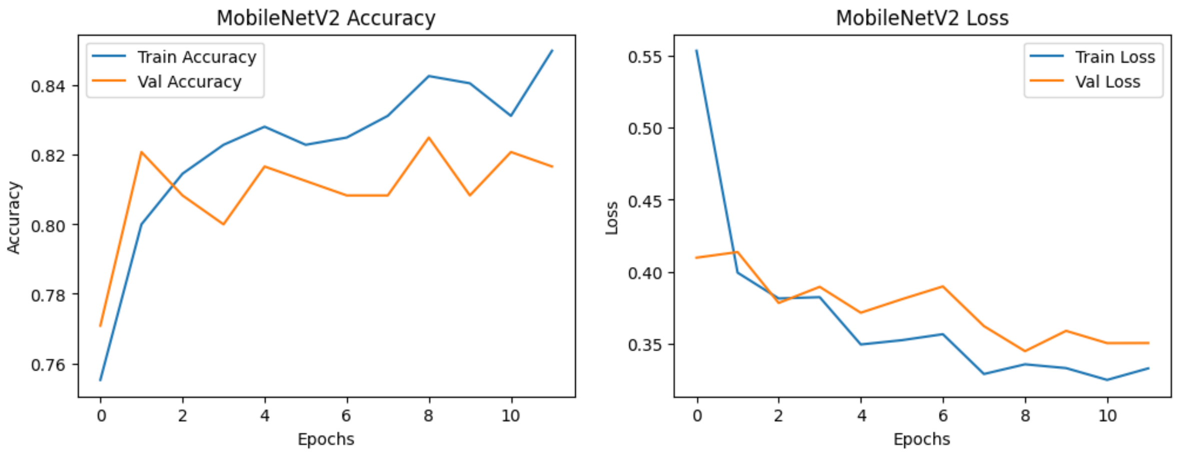
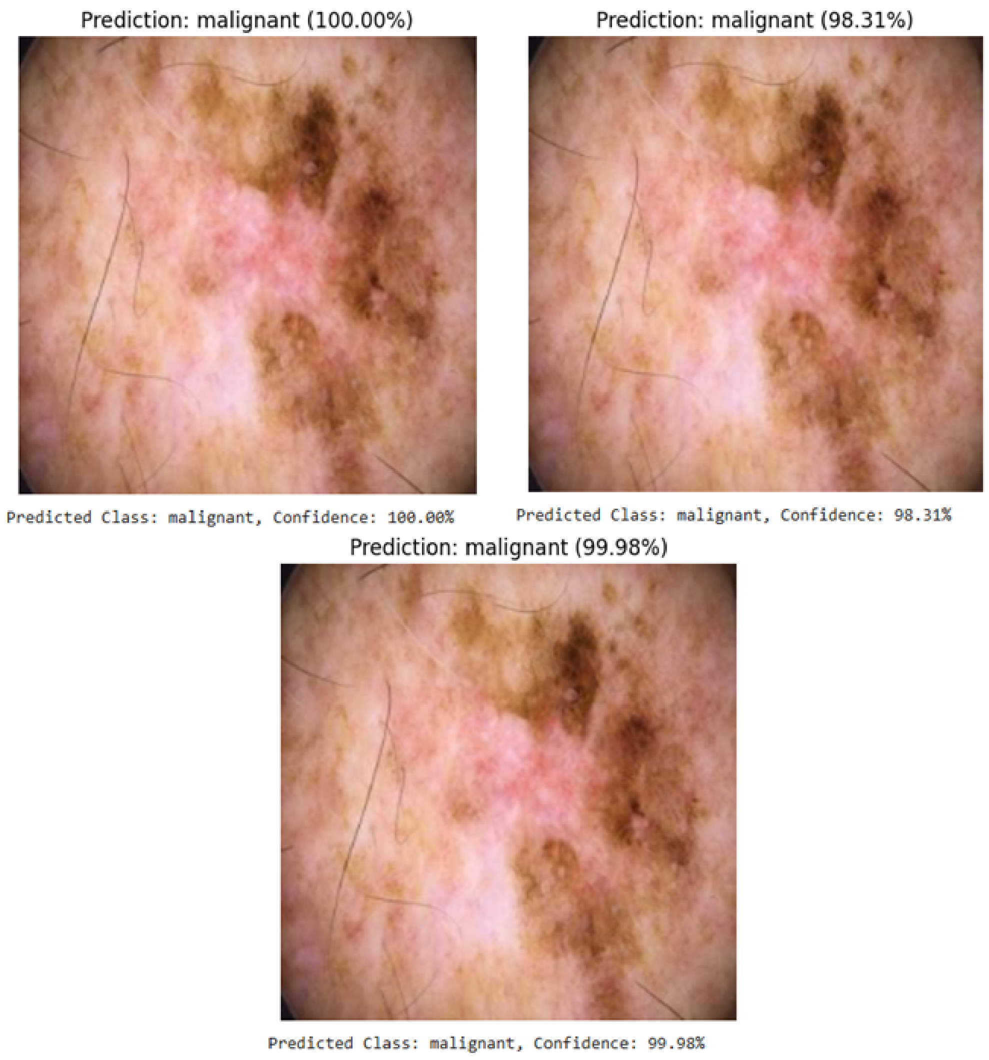


Fig. 12: MobileNetV2 Accuracy and Loss Curve

6.3 Final Prediction


Prediction: malignant (100.00%)
Prediction: malignant (98.31%)
Prediction: malignant (99.98%)

```
Predicted Class: malignant, Confidence: 100.00%
Predicted Class: malignant, Confidence: 98.31%
Predicted Class: malignant, Confidence: 99.98%
```

Fig. 13: Predicted Class Of Each Models

7 Discussion

- **Although the results of this study (Custom CNN, VGG16 and MobileNetV2) can be considered to be a promising method for binary melanoma classification, there are some intrinsic constraints and failure modes that should be recognized.**
- A major limitation is the limited number of updated data. Since we have only 1,200 dermoscopic images to train our models on, the trained model may not be able to generalize well on new unseen data. Overfitting is still a danger though dropout and early stopping are applied for regularization. The performance of the models may degrade when generalizing to more diverse and large datasets or in clinical practice, where image quality and diversity can vary greatly from our controlled experimental settings.
- Another cause for failure is the bias on the data set. If the data of skin types, ages and ethnicities are not reflected, the models might be biased and tend to favor some specific groups. This could lead to a lack of equity in healthcare and disproportionately impact patient populations under-represented in the training data.
- Also, privacy issues do exist in medical data. The language model employed in the current work was trained on anonymous text data, but this processing would be applicable to patient-specific information. It's extremely important to handle data responsibly and to do so in a way that complies with healthcare privacy laws, such as the US's HIPAA or the EU's GDPR.
- Money equity should also be taken into account. Models do not fit all skin tones, ethnicities, genders and we may be perpetuating health disparities. As the models are trained on most images of specific demographic groups, so fairness across different patient populations should be considered.
- Finally, the potential for misuse of the model is real (for example, in clinical care devoid of oversight). The application of deep learning in medical diagnosis must be regulated that the models should only serve as a decision-supporting tool and not diagnostic systems.
- To reduce this risk, data augmentation, fairness-aware algorithms and strong privacy practices need to be employed in addition to ongoing model evaluation and external validation across diverse clinical environments.

8 Conclusion

The models trained in our study such as Custom CNN, VGG16 and MobileNetV2 performed well for the task of binary melanoma classification. The ultimate aim of the study was to differentiate between benign and malignant skin lesions, which we accomplished with all three models. MobileNetV2, VGG16 and Custom CNN had accuracies of 82%, 87% and 84%, respectively, suggesting that all the models were able to extract meaningful features for melanoma discrimination.Though the models worked well with a small amount of data, there were some problems. The sample of the dataset, composed by 1,200 dermoscopic images, can not be considered large enough to ensure an effective generalization: this could have some impact on performances with real world and larger datasets. Even with the use of regularization methods (droput and early stopping), overfitting was still an issue. There was also concern of bias in the data set (non-diversity of skin tones and demographics) affecting fairness/equity of models in disparate patient populations.Key focus areas for the future include the following. It would be beneficial to apply larger and more Universal datasets to increase the generalization and robustness of the employed models. Data augmentation methods could also help to make the models more robust to variations of image quality and other external confounders. Furthermore, the performance of the models can be improved and over-fitting may decrease by adding sophisticated strategies like adversarial training or transfer learning with domain-specific fine-tuning. Ultimately, the models could be validated in clinical practice to evaluate their feasibility and find possible refinements. Future studies should also work on controlling for bias and privacy so that the models are fair and adhere to healthcare regulations.

References

[1] Brinker, T.J., Hekler, A., Enk, A.H., *et al.*: Deep learning outperformed 136 of 157 dermatologists in a head-to-head dermoscopic melanoma image classification task. European Journal of Cancer **113**, 47–54 (2019) https://doi.org/10.1016/j.ejca.2019.04.019

[2] Han, S.S., Kim, M.K., Lim, W.J., Park, G.H., Park, I.S., Chang, S.E.: Classification of the clinical images for benign and malignant cutaneous tumors using a deep learning algorithm. Journal of Investigative Dermatology **138**(7), 1529–1538 (2018) https://doi.org/10.1016/j.jid.2018.01.028

[3] Tschandl, P., Rosendahl, C., Kittler, H.: The HAM10000 Dataset, a Large Collection of Multi-source Dermatoscopic Images of Common Pigmented Skin Lesions. https://doi.org/10.17632/ggh6g39ps2.3 . Accessed on: November 12, 2025. https://data.mendeley.com/datasets/ggh6g39ps2/3

[4] Garg, R., Maheshwari, S., Shukla, A.: Decision support system for detection and classification of skin cancer using cnn. Proceedings of the National Institute of Technology Raipur Conference on Deep Learning Applications, 1–9 (2023)

[5] Sabir, R., Mehmood, T.: Classification of melanoma skin cancer based on image dataset using different neural networks. Scientific Reports **14**(29704), 1–9 (2024) https://doi.org/10.1038/s41598-024-75143-4

[6] Hosny, K.M., Kassem, M.A., Foaud, M.M.: Classification of skin lesions using transfer learning and augmentation with alexnet. PLOS ONE **14**(5), 0217293 (2019) https://doi.org/10.1371/journal.pone.0217293

[7] RedAro: NNFLL Project: Skin Cancer (Melanoma) Detection Using Deep Learning. https://github.com/RedAro/NNFLL_Project_Skin_Cancer-Melanoma-_Detection. Accessed: 2025-11-22 (2025)