DeepMelaNet: Advancing Melanoma Stage Classification in Skin Cancer Diagnosis

Md Sadi Al Huda
Department of Computer Science, American International University-Bangladesh, 408/1, Kuratoli, Khilkhet, Dhaka 1229, Bangladesh
20-43129-1@student.aiub.edu

Tahmid Enam Shrestha
Department of Computer Science, American International University-Bangladesh, 408/1, Kuratoli, Khilkhet, Dhaka 1229, Bangladesh
20-43108-1@student.aiub.edu

Asmaul Hossain
Department of Computer Science, American International University-Bangladesh, 408/1, Kuratoli, Khilkhet, Dhaka 1229, Bangladesh
20-44209-3@student.aiub.edu

Nissan Bin Sharif
Department of Computer Science, American International University-Bangladesh, 408/1, Kuratoli, Khilkhet, Dhaka 1229, Bangladesh
20-44351-3@student.aiub.edu

Md Asraf Ali
Department of Computer Science, American International University-Bangladesh, 408/1, Kuratoli, Khilkhet, Dhaka 1229, Bangladesh
asrafali@aiub.edu (corresponding author)

Timotei Istvan Erdei
Department of Vehicles Engineering, Faculty of Engineering, University of Debrecen, Ótemető Str. 2–4, Debrecen 4028, Hungary
timoteierdei@eng.unideb.hu

Received: 8 July 2024 | Revised: 19 November 2024 | Accepted: 5 December 2024

 DOI: https://doi.org/10.48084/etasr.8336

ABSTRACT

Melanoma skin cancer is a global public health threat due to its increasing rates and the possibility of severe outcomes if not adequately addressed. Melanoma is caused by ultraviolet radiation and, among its two stages, malignant is more dangerous than benign. The diagnosis of melanoma is typically based on visual inspection and manual methods carried out by experienced physicians. However, this method is usually slow and has a high probability of error. Deep-learning-based models can offer better and low-cost treatments for people with melanoma. This study aimed to develop a deep-learning model to classify melanoma skin cancer in its early stages. This study presents a modified deep-learning model, named DeepMelaNet, to correctly classify skin cancer images as benign or malignant. The proposed classifier achieved an accuracy of 93.40%, a precision of 98%, a recall of 94%, and an F1 score of 93% on a dataset of 10,000 melanoma skin cancer images, offering a practical solution that can help healthcare professionals in early skin cancer prediction.

Keywords-melanoma; malignant; benign; deep learning; DeepMelaNet; early detection; skin cancer

I. INTRODUCTION

The skin is the largest and most vital organ, covering the entire body. Cancer is a condition in which certain cells in the body grow uncontrollably and spread to other parts of it [1]. According to the World Health Organization (WHO), cancer is one of the leading health concerns in the world [2]. Among cancers, melanoma is particularly harmful and ranks ninth among the most common [3]. Ultraviolet (UV) light from the sun damages the DNA of unprotected skin, leading to mutations that cause uncontrolled cell growth and cancer [4]. There are two types of melanoma cancer: Malignant is dangerous because it can spread to other parts of the body, leading to metastasis, and benign. If melanoma skin cancer is diagnosed in its early stages, there is a high chance of recovery through minor surgery [5]. Dermoscopy is considered a popular imaging technique for examining the skin's surface but is partially effective because of its heavy reliance on the practitioner's visual acuity.

To provide more effective and smart treatments, deep-learning approaches [6-9] have recently been applied to detect melanoma skin cancer. For example, in [10], a deep-learning approach was applied to classify melanoma skin cancer, demonstrating significant advances in accuracy and efficiency. In [11], a multi-layered Convolutional Neural Network (CNN) with advanced regularization achieved 93.58% accuracy but suggested trimming for further improvement. In [12], a modified U-net network was employed for segmentation, attaining 98.3% accuracy. However, this study used a small dataset that was not enough to generalize and make robust decisions. In [13], various image preprocessing techniques were used, such as level-encoder function, image-size interpolation, normalization, data-reduction, and data augmentation, along with the ReLU function in a Deep-CNN model to enhance its performance. This study suggested that future efforts should focus on the CNN architecture reformation, recent augmentation techniques, and the development of smartphone applications for this model. In [14], an enhanced deep CNN was used to detect melanoma cancer. This approach used a three-layer network consisting of two 2D convolutional layers with 3×3 convolutions, outperforming other models. This study suggested merging multiple datasets and training the algorithm in a broader set of images. In [15], ESR-GAN was used for image generation and segmentation to improve accuracy with small datasets. This study used a modified ResNet50 deep-learning model for classification on a small dataset. In [16], a novel deep learning model within a transfer learning framework for melanoma skin cancer classification was proposed. This method involved obtaining optical skin images and preprocessing them by resizing, normalizing, extracting ROI, and augmenting with random transformations. This model achieved an accuracy of 97% and a loss of 0.09 for the optical dataset. Table I shows a summary of some other related studies.

According to these studies, deep-learning models demonstrated superior performance in cancer detection. However, limitations such as small or imbalanced datasets and limited classification techniques hindered their overall effectiveness. This study aimed to develop a highly accurate deep-learning model for melanoma skin cancer classification and test it on a large dataset of 10,000 images.

TABLE I. SUMMARY OF RELATED STUDIES

Ref.	Method	Limitation	Accuracy
[17]	ANN, CNN, KNN	Limited scope of classification techniques and lack of detailed guidance on technique selection	91%
[18]	SVM, MobileNetV2	Imbalanced dataset, not classified accurately	85%
[19]	CNN with SMTP	Classifies melanoma cancer stages solely based on tumor thickness	N/A
[20]	Transfer learning	Limited to binary skin cancer classification	90.96%
[21]	Mask RCNN, ResNet152	Dataset variety and limited to melanoma detection	90.4%
[22]	CNN, VGG19, ResNet50, and SqueezeNe	Limited model evaluation and questions about generalization	80%
[23]	SVM, KNN, and DT	Limited to binary skin cancer classification and lack of preprocessing exploration	88.02%

This study introduces a robust deep-learning model, named DeepMelaNet, designed to detect early-stage melanoma skin cancer. The initial phase involved preprocessing the melanoma skin cancer dataset to train the model under standardized conditions. The performance of the proposed model was evaluated using accuracy, precision, recall, F1-score, and confusion matrices, ensuring its effectiveness and reliability for clinical applications. Additionally, a comparative analysis with existing melanoma detection methods was carried out to determine the performance of the proposed model and identify potential areas for improvement. The key contributions of this research are as follows:

- The design of a robust Deep Neural Network (DNN) model architecture. This study introduces DeepMelaNet, a new DNN model specifically engineered to improve the diagnostic accuracy of melanoma skin cancer detection.
- Demonstrates an advanced DNN algorithm for precise melanoma skin cancer classification in the early stages.
- Compares the proposed model's performance with recent models to confirm its effectiveness, demonstrating its practicality and superiority over existing dermatological diagnostic methods.

II. METHODOLOGY

A. Dataset

This study utilized the Melanoma Skin Cancer Dataset, which comprises 10,000 images [24]. This dataset includes two stages of melanoma skin cancer: malignant and benign. By employing this dataset, this study obtains an advantage from a substantial amount of data, therefore improving the accuracy and dependability of the model's results. Moreover, the dataset's focus on melanoma underscores the potential impact and significance of this study in the realm of medical imaging and diagnostics. Table II provides a dataset description.

TABLE II. DATASET OVERVIEW

Class	Training	Validation	Test
Benign	5000	500	500
Malignant	4605	500	500
Total	**9605**	**1000**	**1000**

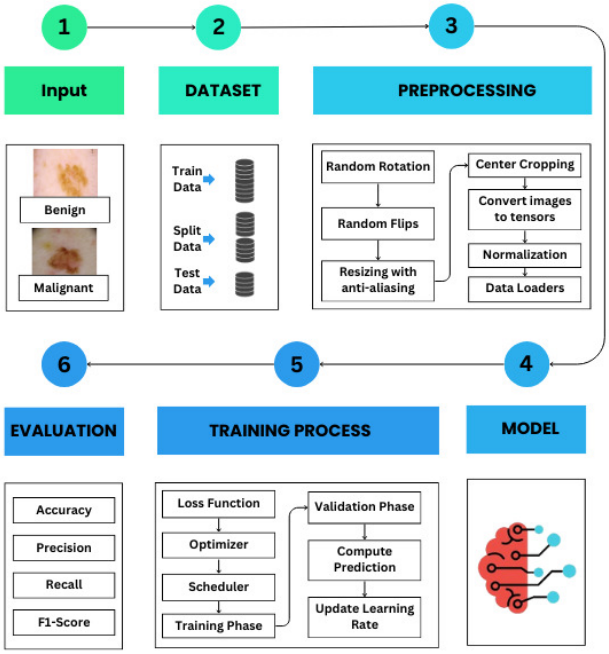


Fig. 1. Block diagram of the proposed method.

B. Experimental Setup

The experimental framework was established on Kaggle, a renowned cloud-based platform that provides computational resources tailored for data science and machine learning initiatives. This study used Kaggle's T4 GPU instances, powered by NVIDIA's Tesla T4 GPUs, to balance computational efficiency with performance, catering to the deep learning requirements. The software environment was orchestrated around PyTorch, an open-source machine learning library favored for its flexibility and dynamic computation capabilities, essential for developing deep learning models such as EfficientNet. The specific version of PyTorch used was 2.2.1, which ensured full compatibility with the model requirements. Auxiliary libraries such as NumPy, for handling numerical operations, and Matplotlib, for visualization purposes, were also integral to the experimental setup.

C. Data Preprocessing

Several preprocessing techniques were applied to increase the dataset quality and prepare it for the proposed model. The dataset was split into training and validation sets using PyTorch's ImageFolder class. This division ensures that the model learns on a representative subset of the data and is tested on unseen data to evaluate its generalizability. Data augmentation methods were used on the training set, which included applying random rotations of up to 20° and randomly flipping images horizontally and vertically with a probability of 30% for each image. These changes add artificial variances in the training data, assisting the model in learning robust features and boosting its capacity to generalize to unseen images. The images in both the training and validation sets were shrunk to a fixed size of 112×112 pixels using antialiasing for better quality, and a center crop of the same size was extracted. The images were then transformed into tensors and normalized using the mean and standard deviation values particular to ImageNet. Finally, PyTorch's DataLoaders were used to efficiently load the training and validation data in batches throughout the training process. The batch size was chosen to balance training speed and memory utilization. Training data were shuffled in each epoch to ensure that the model was exposed to a random sequence of images, preventing it from storing the order of the data and further boosting its capacity to generalize. These preprocessing steps provide practical model training and dependable research outcomes. Figure 2 illustrates the preprocessing techniques applied.

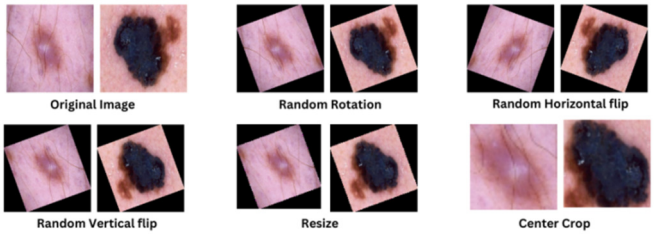


Fig. 2. Overview of the preprocessing steps.

D. Proposed DeepMelaNet Architecture

The DeepMelaNet architecture was developed from EfficientNet, incorporating several modifications to optimize the memory access costs. One of the key modifications is the utilization of a reduced expansion ratio for MBConv. Smaller expansion ratios are preferred, as they tend to incur fewer memory access costs. Additionally, the DeepMelaNet architecture favors smaller (3×3) kernel sizes. However, to compensate for the resulting decrease in the receptive field caused by the smaller kernel size, extra layers are added to the network. EfficientNet is a robust CNN architecture that has been pre-trained on a large-scale image dataset (ImageNet). By modifying a pre-trained model, DeepMelaNet can harness the significant features learned from the images in the dataset. These features are then fine-tuned specifically for the task of skin cancer classification. Figure 3 illustrates the proposed DeepMelaNet model architecture.

1) Fused-MB Conv2

The initial block consists of a convolutional layer with a 3×3 filter, batch normalization, and SiLU activation, represented as "ConV (3×3), BN, SiLU". This block produces an output of 32 channels with a spatial dimension of 128. The second block consists of a convolutional layer with an 1×1 filter and batch normalization, denoted "ConV (1×1), BN". This layer delivers an output of 128 channels with a spatial dimension of 64.

2) Fused-MB Conv3

The first process consists of a 3×3 convolution with batch normalization and SiLU activation, resulting in 64 channels of size 256 each. The second operation is a 3×3 convolution with batch normalization, which increases the number of channels to 256 while keeping the size at 64.

3) Fused-MB Conv4

This layer consists of a 3×3 convolution operation with normalization batches and SiLU activation, which generates a 64×256 output. Then, it is followed by an 1×1 convolution operation with batch normalization, resulting in a 256×96 output.

4) Fused-MB Conv5

The layer consists of a 3×3 convolution operation with batch normalization and SiLU activation function, resulting in an output size of 96×384. This is followed by an 1×1 convolution operation with batch normalization, resulting in an output size of 384×96.

5) Fused-MB Conv6

This layer begins with an 1×1 convolution operation followed by batch normalization and SiLU activation, resulting in a size of 96×384. It then proceeds with a 3×3 convolution operation with batch normalization, resulting in a size of 384×384. Next, it incorporates a Squeeze Excitation block with a sigmoid activation function, which modifies the channel depth to 24 and then restores it to 384. Finally, it concludes with an 1×1 convolution operation with batch normalization, resulting in a size of 384×192.

6) Fused-MB Conv7

This layer starts with an 1×1 convolution operation, including batch normalization and SiLU activation. This operation produces an output of size 192×768. Next, a 3×3 convolution operation with batch normalization is applied, resulting in an output of 768×768. Following this, a Squeeze Excitation block is used, which involves reducing the number of channels to 48 and then expanding them back to 768. Finally, it performs an 1×1 convolution operation with batch normalization, resulting in an output size of 768×192.

7) Fused-MB Conv8

The "Fused-MB Conv8" layer begins with an 1×1 convolution operation that includes batch normalization and SiLU activation. This operation produces an output of size 192×1152. Next, a 3×3 convolution operation with batch normalization is performed, resulting in an output of 1152×1152. A Squeeze Excitation block is then applied, which adjusts the number of channels to 48 and then back to 1152 using a sigmoid activation function. Finally, an 1×1 convolution operation with batch normalization is performed, resulting in an output of size 1152×224.

8) Fused-MB Conv9

This sequence begins with a 1×1 convolution that includes batch normalization and SiLU activation. This produces an output of size 224×1344. It then proceeds with a 3×3 convolution that includes batch normalization, resulting in an output of size 1344×1344. The sequence also incorporates a Squeeze Excitation block, which uses a sigmoid activation to modulate the channels. This compression reduces the size to 56 and then expands it back to 1344. Finally, the sequence concludes with an 1×1 convolution that includes batch normalization, resulting in a final output size of 1344×224.

9) Fused-MB Conv10

This layer starts an 1×1 convolution combined with batch normalization and SiLU activation. This produces an output with dimensions of 224×1344. The layer then transitions to a 3×3 convolution with batch normalization, resulting in an output of 1344×1344. A Squeeze Excitation mechanism is incorporated, which employs a sigmoid function to refine the channel dynamics. This mechanism narrows the channels to 56 and then expands them back to 1344. Finally, the layer concludes with a 1×1 convolution with batch normalization, resulting in a final output of 1344×384.

10)Fused-MB Conv11

The first step of this layer is an 1×1 convolution with batch normalization and SiLU activation, producing an output of 384×2304. Then, a 3×3 convolution with batch normalization produces an output of 2304×2304. Finally, a Squeeze Excitation block with sigmoid activation adjusts the channel dimensions to 96, then back up to 2304, and finally, an 1×1 convolution with batch normalization produces a 2304×384 output.

11)Fused-MB Conv12

The first step in this layer is a 1×1 convolution with batch normalization and SiLU activation, which yields a 384×2304 output. Next, a 3×3 convolution with batch normalization produces an output of 2304×2304. Finally, a Squeeze Excitation block with sigmoid activation modulates the channel depth to 96 and then re-expands to 2304. Finally, an 1×1 convolution with batch normalization culminates in a 2304×640 final output.

12)Fused-MB Conv13

Beginning with a 1×1 convolution with batch normalization and SiLU activation, the "Fused-MB Conv13" layer generates an output of 640×3840. Next, a 3×3 convolution with batch normalization is applied, resulting in an output of 3840×3840. Next, a Squeeze Excitation block with sigmoid activation is used, manipulating the channel values down to 160 and then right back up to 3840. Finally, an 1×1 convolution with batch normalization is applied, yielding an output of 3840×640.

13)Fused-MB Conv14

The final classification layer of the pre-trained model is modified to have a single output neuron with sigmoid activation. Figure 3 shows the layer architectures. This modification adapts the model for binary classification, where the output represents the probability that an image belongs to the malignant and benign classes.

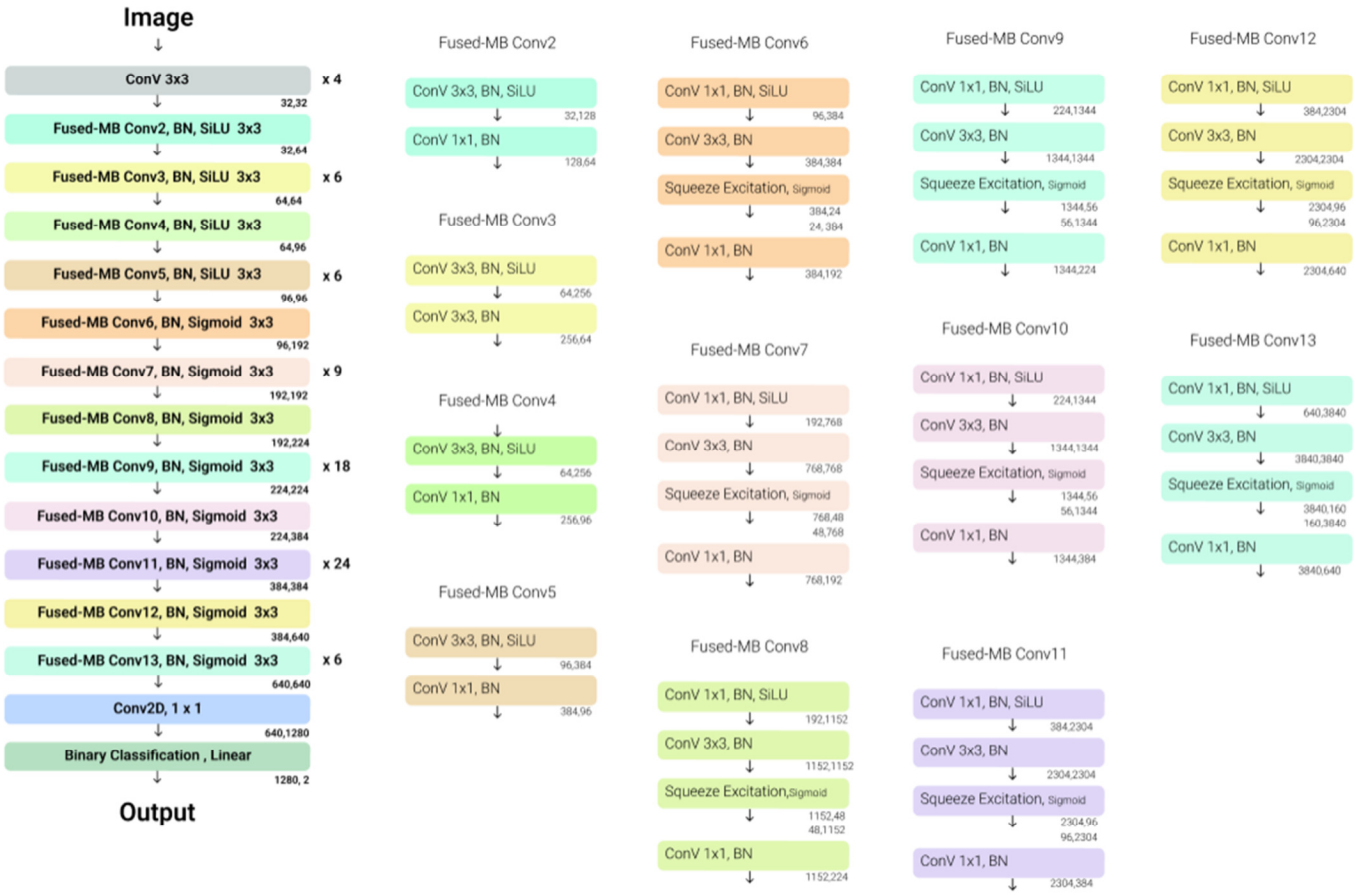


Fig. 3. Proposed DeepMelaNet model architecture.

E. Data Transformation

During the training data transformations, dataset images experience a sequence of alterations. At first, a random rotation was performed within a range of -20 to 20 degrees. Subsequently, the images can be mirrored horizontally and vertically, with a chance of 0.3 for each. The images were then enlarged to dimensions of 112×112 using antialiasing, and a center crop was implemented to preserve the original size. Afterward, the images were initially in the PIL Image format and transformed into PyTorch tensors. Finally, the image sensor was normalized using the given mean [0.485, 0.456, 0.406] and standard deviation [0.229, 0.224, 0.225] for each RGB channel.

F. Training And Validation Loop

Training proceeded in epochs, each denoting a complete pass through the dataset. In each epoch, the model learns by processing data batches for parameter updates. Loss and accuracy metrics are recorded. When the training session is completed, the model goes into evaluation mode to assess its performance against the validation set and calculate the validation metrics. To enhance the performance during training and its efficiency, the validation loss helps in obtaining the learning rate scheduler adjustments. Algorithm 1 illustrates the training and validation loops of the proposed model.

```
ALGORITHM 1: TRAINING AND VALIDATION LOOP WITH PERFORMANCE METRICS
Input: model, trainLoader, valLoader, optimizer, criterion, scheduler, epochs
Output: trainLosses, trainAccs, valLosses, valAccs, valPrecisions, valRecalls, valF1s
Initialize scaler for mixed-precision training
Initialize metric lists: trainLosses, trainAccs, valLosses, etc.
for epoch = 1 to epochs do
  Set model to training mode
  Initialize running Loss←0, correct_train←0
  for each input, labels in trainLoader do
    Move inputs, labels to device
    Zero gradients: optimizer.zero_grad()
    With autocast enabled:
    Outputs ← model(inputs)
    loss ← criterion(outputs, labels)
    Backpropagate and update model:
```

```
        scaler.scale(loss).backward(),
        scaler.step(optimizer)
15:     Update running loss and correct
        predictions
16:   end for
17:   Calculate and store trainLoss and
      trainAcc
18:   Print training metrics
19:   Set model to evaluation mode
20:   Initialize validation metrics
21:   With gradient disabled:
22:   for each input, labels in valLoader
23:     Move inputs, labels to device
24:     outputs ← model(inputs), compute
        loss
25:     Update validation loss and
        store predictions
26:   end for
27:   Calculate and store validation
      metrics: avgLoss, accuracy,
      precision, etc.
28:   Print validation metrics
29:   Update scheduler based on validation
      loss
30: end for
```

G. Epoch

This is the number of times the training process undergoes the whole training set, and it consists of one or more batches. The number of epochs was set to 30. After each epoch, the training loss and accuracy are calculated and recorded, and the training metrics are reported. The model is subsequently set to evaluation mode. With gradient computation disabled, for each batch of inputs and labels in the validation loader, the inputs and labels are transported to the device, outputs are calculated, and the loss is calculated.

H. Learning Rate

The learning rate determines the pace at which training weights are adjusted. A lower rate results in slower updates to the weights, necessitating more epochs for training. Conversely, a higher learning rate accelerates model convergence but risks leading to less-than-optimal results.

I. Training Function

The binary cross-entropy with logit loss function was employed, which is appropriate for binary classification problems with sigmoid activation. Adam optimizer was used, which changes the model weights during training at a learning rate of 0.001. A scheduler, especially the ReduceLROnPlateau scheduler, was created to change the learning rate during training. This scheduler helps avoid overfitting by lowering the learning rate if the validation loss stagnates for a set number of epochs. The training incorporated an iterative training loop that runs for a preset number of epochs. The model is set to train mode throughout each epoch and iterates over the training data stream in batches. For each batch, the model performs a forward pass to compute predictions, calculates the loss using the loss function, and performs a reverse pass to compute gradients. Finally, the scheduler modifies the learning rate depending on the validation loss. This repeated training procedure refines the model's capacity to identify between benign and malignant skin cancer images, boosting the results' dependability. After training, the model's performance was assessed on a separate test set not utilized during training or validation. This review objectively assesses the model's generalization to real-world data. Metrics such as accuracy, precision, recall, and F1-score were calculated using the test set to evaluate the model classification performance.

The validation procedure aimed to assess the effectiveness of the DeepMelaNet model in classifying medical images as "Benign" or "Malignant," indicating the presence of benign or malignant skin cancers, respectively. Model weights were retrieved from a saved checkpoint, and the model was set to evaluation mode to ensure consistent results. Each image was classified by the model, and the prediction along with its associated probability was recorded. These predictions were compared to the actual ground truth class of each image (benign or malignant) to assess the model's classification accuracy.

J. Model Evaluation Parameters

The model was evaluated using accuracy, precision, recall, and F1-score [25-28] to validate its performance.

1) Accuracy

Accuracy represents the ratio of correctly predicted instances to the total instances in the dataset. It is a measure of how often the model makes the correct predictions. Mathematically, accuracy is given by:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

2) Precision

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. It quantifies the accuracy of positive predictions. It indicates how many of the predicted positive instances are truly positive. Mathematically, precision is given by:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

3) Recall

Recall, also known as sensitivity or actual positive rate, measures the ratio of correctly predicted positive observations to all actual positive observations. It quantifies the ability of the model to find all the relevant cases within a dataset. Mathematically, recall is given by:

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

4) F1-score

F1-score is the harmonic mean of precision and recall, providing a balance between them. It is calculated as the weighted average of precision and recall, where the F1-score reaches its best value at 1 and worst at 0. It is useful when there is an uneven class distribution, as it considers both false

positives and false negatives. Mathematically, F1-score is given by:

$$F1 - score = \frac{2\times Recall \times Precision}{Recall + Precision} \quad (4)$$

III. RESULTS AND DISCUSSION

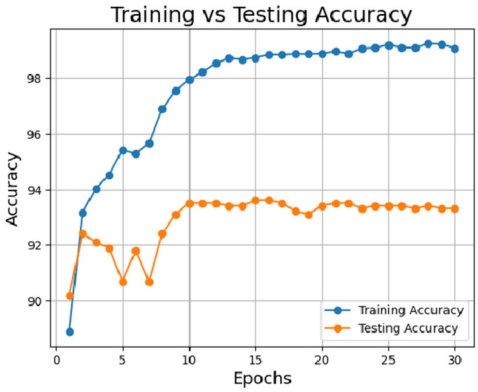
A. Performance Evaluation Matrix Analysis

The proposed approach excelled in predictive performance, achieving a high accuracy of 93.4%, indicating that the model predominantly makes accurate classifications. Moreover, it demonstrated a precision of 98%, effectively minimizing false positives. The model's recall reached 94%, critical for disease detection, where missing positive cases could be detrimental. Alongside these metrics, the model attained an F1-score of 93%, reflecting the model's robustness and reliability. These metrics highlight the efficacy of the proposed approach in handling complex and sensitive tasks, ensuring comprehensive and dependable performance across various scenarios.

B. Model Performance Graph

The accuracy graph depicts the accuracy of a classification model during its training and validation phases over epochs. Figure 4(a) shows the training and testing accuracies through epochs. Each accuracy improves rapidly at the beginning of training. The training accuracy increased until it plateaued near 100%, indicating that the model almost perfectly matches the training data. In contrast, the testing accuracy initially increased but then dropped much earlier and remained constant at 92-94%. Training loss reflects the model's learning process on the training data while testing loss measures its performance on unseen data. Figure 4(b) shows the training and testing losses across training epochs. Initially, training loss started high but rapidly decreased, indicating swift learning and accurate classification of training data. As epochs progress, the training loss curve flattens, suggesting that the model has primarily stabilized and is not significantly improving on the training data. In contrast, testing loss begins high and decreases, albeit less sharply than training loss. This indicates that the model is learning from the testing data but is less efficient than the training data. However, the validation loss begins to increase slowly after an initial decline, implying potential overfitting of the training data. Overfitting occurs when a model excessively learns the training data, including its noise and outliers, compromising its performance on new data.

(a)



(b)

Fig. 4. Model accuracy (a) and loss (b) graphs.

C. Confusion Matrix Analysis

Figure 5 shows the confusion matrix of the proposed model in identifying melanoma skin cancer. The matrix shows four quadrants: the top left (TP) with 477, indicating that 477 benign cases were correctly identified, the top right (FN) with 23, showing that 23 malignant cases were misclassified as benign, the bottom left (FP) with 47, where 47 benign cases were incorrectly labeled as malignant; and the bottom right (TN) with 453, meaning 453 malignant cases were correctly identified. The high numbers along the diagonal (true positives and negatives) suggest the model performs well. However, FN and FP indicate room for improvement, mainly because FN in a medical context can be very dangerous, as they represent missed diagnoses of potentially life-threatening conditions. The model balances sensitivity (actual positive rate) and specificity (valid negative rate). However, further efforts should be made to reduce the number of FN due to the critical nature of cancer diagnosis.

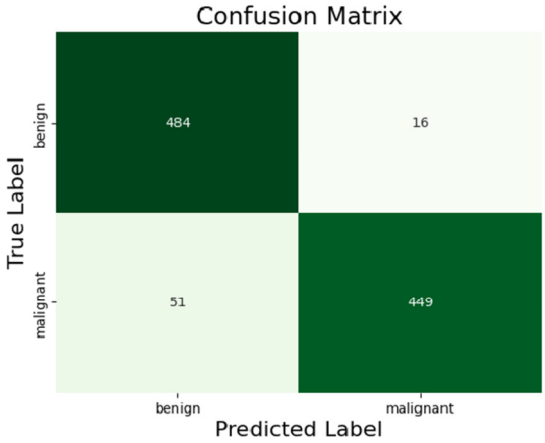


Fig. 5. Confusion matrix of the proposed model.

D. Discussion

This study developed a modified deep-learning architecture, called DeepMelaNet, to address the crucial task of classifying melanoma skin cancer. This method has advanced feature extraction and generalization capabilities in image classification, making it a pivotal component of this research framework. Its scalability and the delicate balance between

classification accuracy and computational efficiency made it an indispensable tool for this study. The distinctive design of the proposed method includes depth-wise separable convolutions, facilitating more efficient computation without compromising model performance. Its versatility extends beyond performance metrics to include considerations of computational resource utilization. Its efficient parameters and computation utilization allow seamless deployment across diverse computational platforms, from resource-constrained edge devices to high-performance computing clusters. This adaptability underscores its suitability for real-world deployment scenarios, where computational constraints often dictate the feasibility of model implementation.

The experimental results of the proposed deep learning model in classifying images of melanoma skin cancer cells were promising. The model was trained using the Kaggle T4 GPU, with hyperparameter fine-tuning to achieve the best results, achieving an accuracy of 93.40%. Training was carried out over 30 epochs. In particular, the best validation accuracy of 93.40% was achieved in the 24th epoch. The evaluation metrics, including accuracy, precision, recall, and F1-score, demonstrate the effectiveness of the proposed model in detecting melanoma skin cancer.

E. Comparison with Related Works

The proposed approach shows superior performance compared to recent works in melanoma skin cancer detection. Traditional methods such as KNN, SVM, NB, and NN achieved accuracies ranging from 88.2% to 90.9%, whereas the proposed DeepMelaNet model achieved a significantly higher accuracy of 93.40%. Additionally, compared to CNN-based approaches, the proposed method surpassed their accuracy of 91.95% and 90.9%, respectively. While another approach utilizing RCC, BCC, and GCC methods requires more accurate information for direct comparison, the proposed approach outperformed it on the same dataset. Table III shows a detailed comparison analysis.

TABLE III. COMPARISON WITH RECENT RESEARCH

Ref.	Dataset	Method	Accuracy (%)
[15]	Melanoma Skin Cancer Dataset of 10000 Images	CNN, Modified ResNet50	86
[29]	Melanoma Skin Cancer Dataset of 10000 Images	CNN	91.95
[30]	Melanoma Skin Cancer Dataset of 10000 Images	RCC, BCC, GCC	88
[31]	Melanoma Skin Cancer Dataset of 10000 Images	CNN	90.9
Proposed method	Melanoma Skin Cancer Dataset of 10000 Images	DeepMelaNet	93.40

DeepMelaNet showed a high accuracy rate of 93.40% and strong performance measures such as precision, recall, and F1-score. Due to its depth-wise convolutions, this model can work well on a range of computing systems, from edge devices to high-performance clusters. This versatility can help in finding and treating diseases earlier and better.

IV. CONCLUSION AND FUTURE WORK

This study introduced DeepMelaNet, a robust deep neural network tailored for accurate melanoma skin cancer classification in its early stages. Using a publicly available dataset comprising 10,000 skin images, the model effectively discerned crucial details necessary for precise classification. This model demonstrated strong performance, achieving 93.40% accuracy, showcasing its potential as a valuable tool for skin cancer classification. DeepMelaNet represents a significant advance in the detection of skin diseases, particularly in the distinction between malignant and benign skin melanomas.

In the future, the CNN model should be trained and tested on more extensive and varied sets of skin cancer images to improve generalization. This will ensure that the model learns a broader range of features and patterns, making it more robust in real-world scenarios. Cross-validation methods can provide a more accurate picture of how well the model performs. This is done by splitting the dataset into several smaller sets and training the model on various combinations of these smaller sets. Averaging the results can reduce the effect of data bias and make the review more reliable. Better model performance can come from different hyperparameter optimization methods. Ensemble methods, such as model averaging or stacking, can make the model work even better. Putting together predictions from several CNN models or different designs can help reduce the flaws in each model and produce more accurate results. Before the model can be used in clinical settings, it must be thoroughly tested following all legal rules.

REFERENCES

[1] C. Carlberg and E. Velleuer, *Cancer Biology: How Science Works*. Cham, Switzerland: Springer International Publishing, 2021.

[2] "Radiation: Ultraviolet (UV) radiation and skin cancer," World Health Organization (WHO), Geneva, Switzerland, 2017.

[3] T. Saba, "Computer vision for microscopic skin cancer diagnosis using handcrafted and non-handcrafted features," *Microscopy Research and Technique*, vol. 84, no. 6, pp. 1272–1283, 2021, https://doi.org/10.1002/jemt.23686.

[4] S. Wirunchit, C. Apivitcholchat, T. Chodjarusawad, and W. Koetniyom, "The study of UV protection materials," *AIP Conference Proceedings*, vol. 2010, no. 1, Sep. 2018, Art. no. 020024, https://doi.org/10.1063/1.5053200.

[5] S. Z. Shalhout, H. L. Kaufman, K. S. Emerick, and D. M. Miller, "Immunotherapy for Nonmelanoma Skin Cancer: Facts and Hopes," *Clinical Cancer Research*, vol. 28, no. 11, pp. 2211–2220, Jun. 2022, https://doi.org/10.1158/1078-0432.CCR-21-2971.

[6] D. Adla, G. V. R. Reddy, P. Nayak, and G. Karuna, "Deep learning-based computer aided diagnosis model for skin cancer detection and classification," *Distributed and Parallel Databases*, vol. 40, no. 4, pp. 717–736, Dec. 2022, https://doi.org/10.1007/s10619-021-07360-z.

[7] M. S. Ali, M. S. Miah, J. Haque, M. M. Rahman, and M. K. Islam, "An enhanced technique of skin cancer classification using deep convolutional neural network with transfer learning models," *Machine Learning with Applications*, vol. 5, Sep. 2021, Art. no. 100036, https://doi.org/10.1016/j.mlwa.2021.100036.

[8] W. Gouda, N. U. Sama, G. Al-Waakid, M. Humayun, and N. Z. Jhanjhi, "Detection of Skin Cancer Based on Skin Lesion Images Using Deep Learning," *Healthcare*, vol. 10, no. 7, Jul. 2022, Art. no. 1183, https://doi.org/10.3390/healthcare10071183.

[9] S. M. Jaisakthi, M. P, C. Aravindan, and R. Appavu, "Classification of skin cancer from dermoscopic images using deep neural network

architectures," *Multimedia Tools and Applications*, vol. 82, no. 10, pp. 15763–15778, Apr. 2023, https://doi.org/10.1007/s11042-022-13847-3.

[10] T. Imran, A. S. Alghamdi, and M. S. Alkatheiri, "Enhanced Skin Cancer Classification using Deep Learning and Nature-based Feature Optimization," *Engineering, Technology & Applied Science Research*, vol. 14, no. 1, pp. 12702–12710, Feb. 2024, https://doi.org/10.48084/etasr.6604.

[11] Md. A. Hossin, F. F. Rupom, H. R. Mahi, A. Sarker, F. Ahsan, and S. Warech, "Melanoma Skin Cancer Detection Using Deep Learning and Advanced Regularizer," in *2020 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Depok, Indonesia, Oct. 2020, pp. 89–94, https://doi.org/10.1109/ICACSIS51025.2020.9263118.

[12] R. L. Araújo, F. H. D. de Araújo, and R. R. V. e. Silva, "Automatic segmentation of melanoma skin cancer using transfer learning and fine-tuning," *Multimedia Systems*, vol. 28, no. 4, pp. 1239–1250, Aug. 2022, https://doi.org/10.1007/s00530-021-00840-3.

[13] Md. K. Islam *et al.*, "Melanoma Skin Lesions Classification using Deep Convolutional Neural Network with Transfer Learning," in *2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA)*, Riyadh, Saudi Arabia, Apr. 2021, pp. 48–53, https://doi.org/10.1109/CAIDA51941.2021.9425117.

[14] N. Girdhar, A. Sinha, and S. Gupta, "DenseNet-II: an improved deep convolutional neural network for melanoma cancer detection," *Soft Computing*, vol. 27, no. 18, pp. 13285–13304, Sep. 2023, https://doi.org/10.1007/s00500-022-07406-z.

[15] G. Alwakid, W. Gouda, M. Humayun, and N. U. Sama, "Melanoma Detection Using Deep Learning-Based Classifications," *Healthcare*, vol. 10, no. 12, Dec. 2022, Art. no. 2481, https://doi.org/10.3390/healthcare10122481.

[16] W. Abbes and D. Sellami, "Deep Neural Networks for Melanoma Detection from Optical Standard Images using Transfer Learning," *Procedia Computer Science*, vol. 192, pp. 1304–1312, Jan. 2021, https://doi.org/10.1016/j.procs.2021.08.134.

[17] M. Dildar *et al.*, "Skin Cancer Detection: A Review Using Deep Learning Techniques," *International Journal of Environmental Research and Public Health*, vol. 18, no. 10, Jan. 2021, Art. no. 5479, https://doi.org/10.3390/ijerph18105479.

[18] R. Indraswari, R. Rokhana, and W. Herulambang, "Melanoma image classification based on MobileNetV2 network," *Procedia Computer Science*, vol. 197, pp. 198–207, Jan. 2022, https://doi.org/10.1016/j.procs.2021.12.132.

[19] R. Patil and S. Bellary, "Machine learning approach in melanoma cancer stage detection," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, Part B, pp. 3285–3293, Jun. 2022, https://doi.org/10.1016/j.jksuci.2020.09.002.

[20] R. Dandu, M. V. Murthy, and Y. B. R. Kumar, "Transfer learning for segmentation with hybrid classification to Detect Melanoma Skin Cancer," *Heliyon*, vol. 9, no. 4, Apr. 2023, https://doi.org/10.1016/j.heliyon.2023.e15416.

[21] M. F. Jojoa Acosta, L. Y. Caballero Tovar, M. B. Garcia-Zapirain, and W. S. Percybrooks, "Melanoma diagnosis using deep learning techniques on dermatoscopic images," *BMC Medical Imaging*, vol. 21, no. 1, Jan. 2021, Art. no. 6, https://doi.org/10.1186/s12880-020-00534-8.

[22] A. Khamparia, P. K. Singh, P. Rani, D. Samanta, A. Khanna, and B. Bhushan, "An internet of health things-driven deep learning framework for detection and classification of skin cancer using transfer learning," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 7, 2021, Art. no. e3963, https://doi.org/10.1002/ett.3963.

[23] D. Keerthana, V. Venugopal, M. K. Nath, and M. Mishra, "Hybrid convolutional neural networks with SVM classifier for classification of skin cancer," *Biomedical Engineering Advances*, vol. 5, Jun. 2023, Art. no. 100069, https://doi.org/10.1016/j.bea.2022.100069.

[24] M. H. Javid, "Melanoma Skin Cancer Dataset of 10000 Images." Kaggle, https://doi.org/10.34740/kaggle/dsv/3376422.

[25] K. Tanvir *et al.*, "Enhancing Early-Stage Detection of Melanoma using a Hybrid BiTDense," *TWIST*, vol. 19, no. 2, pp. 298–305, May 2024.

[26] Md. N. Hossain, N. Fahad, R. Ahmed, A. Sen, Md. S. Al Huda, and Md. I. Hossen, "Preventing Student's Mental Health Problems with the Help of Data Mining," *International Journal of Computing*, pp. 101–108, Apr. 2024, https://doi.org/10.47839/ijc.23.1.3441.

[27] Md. A. Ali, Md. K. Morol, M. F. Mridha, N. Fahad, M. S. A. Huda, and N. Ahmed, "Exploring a Novel Machine Learning Approach for Evaluating Parkinson's Disease, Duration, and Vitamin D Level," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 12, 2023, https://doi.org/10.14569/IJACSA.2023.0141265.

[28] T. E. Shrestha, A. R. Aurnob, S. A. Tanim, M. Islam, and K. Nur, "Revolutionizing Cucumber Agriculture: AI for Precision Classification of Leaf Diseases," in *2024 6th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT)*, Dhaka, Bangladesh, May 2024, pp. 776–781, https://doi.org/10.1109/ICEEICT62016.2024.10534530.

[29] H. M. Balaha, A. E. S. Hassan, E. M. El-Gendy, H. ZainEldin, and M. M. Saafan, "An aseptic approach towards skin lesion localization and grading using deep learning and harris hawks optimization," *Multimedia Tools and Applications*, vol. 83, no. 7, pp. 19787–19815, Feb. 2024, https://doi.org/10.1007/s11042-023-16201-3.

[30] M. M. M. Al-Hatab, A. S. I. Al-Obaidi, and M. A. Al-Hashim, "Exploring CIE Lab Color Characteristics for Skin Lesion Images Detection: A Novel Image Analysis Methodology Incorporating Color-based Segmentation and Luminosity Analysis," *Fusion: Practice and Applications*, vol. 15, no. 1, pp. 88–97, 2024, https://doi.org/10.54216/FPA.150108.

[31] S. R. Waheed *et al.*, "Melanoma Skin Cancer Classification based on CNN Deep Learning Algorithms," *Malaysian Journal of Fundamental and Applied Sciences*, vol. 19, no. 3, pp. 299–305, May 2023, https://doi.org/10.11113/mjfas.v19n3.2900.