Skin Cancer Classification Using Deep Learning

Rajat Mehta
rajatmehta1524@gmail.com

***Abstract*—This paper presents a deep learning-based approach for classifying skin cancer images as benign or malignant. The model is trained on a structured dataset and evaluated using various metrics.**

I. Introduction

Skin cancer is one of the most prevalent forms of cancer. Early detection can significantly improve survival rates. In this work, we leverage deep learning techniques to automate the classification process.

II. Dataset

The dataset consists of labeled images categorized into two classes: Benign (B) and Malignant (M). The training set comprises 6289 benign and 5590 malignant images, while the test set consists of 1000 benign and 1000 malignant images. The test set is balanced, ensuring fair evaluation, whereas the training set has a slight class imbalance.

To improve generalization and model robustness, multiple preprocessing techniques were applied, including data augmentation, noise removal, and contrast enhancement. Details of these preprocessing steps are discussed in the next section.

III. Data Preprocessing

To enhance the quality and diversity of input images, several preprocessing techniques were applied before training. These techniques improve model robustness, reduce noise, and ensure optimal feature extraction.

A. Data Augmentation

Data augmentation was applied to improve generalization and prevent overfitting. The following transformations were used:

- **Random Horizontal Flip:** Images were flipped horizontally with a 50% probability to help the model learn invariance to orientation changes.
- **Brightness Adjustment:** Brightness was randomly adjusted within a range of ±0.1 intensity to account for lighting variations.
- **Contrast Adjustment:** Contrast was randomly scaled between 0.8 and 1.2 to improve robustness against different lighting conditions.
- **Random Cropping:** A 20-pixel padding was added, followed by random cropping to simulate different zoom levels and framing variations.

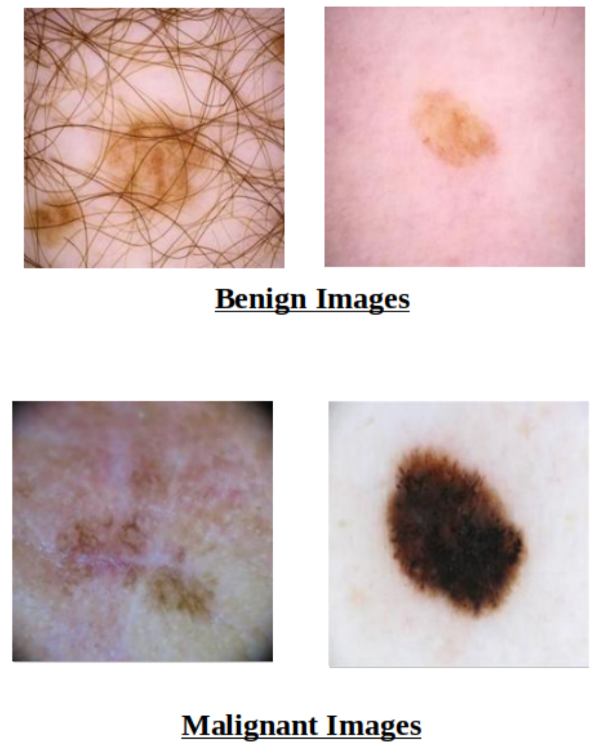


Fig. 1: Sample images from the dataset.

B. Noise Removal

To smoothen the images and remove unnecessary pixel variations, a Gaussian Blur approximation was applied using a 3x3 depthwise convolutional filter:

$$\text{Filter} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \div 16 \tag{1}$$

This operation averages neighboring pixel values, reducing noise and enhancing feature clarity.

C. Contrast Enhancement

To improve feature visibility, contrast adjustment was applied using a scaling factor of 1.5. This enhances pixel intensity differences, making it easier for the model to distinguish skin lesions.

D. Impact on Model Performance

These preprocessing techniques significantly improved model stability and training efficiency:

- **Generalization:** Data augmentation improved robustness to real-world variations.
- **Reduced Overfitting:** The model performed better on unseen data.
- **Feature Enhancement:** Contrast adjustment helped improve detection of lesion boundaries.

IV. Model Architecture

In this experiment, the aim was to train multiple models over the skin image dataset and then compare their performance based on inference results. Three models, namely a **simple CNN** architecture (vanilla pipeline), a **ConvNext Architecture** model and a **Vision Transformer** Model were trained separately on the dataset. The results from the Vision Transformer Model are not included in this report because of the incconsistency and errors (keeping in mind the time constraints), but will be added in the future iterations of this project. The ConvNeXt architecture has demonstrated state-of-the-art performance in image classification, making it a good choice for this task.

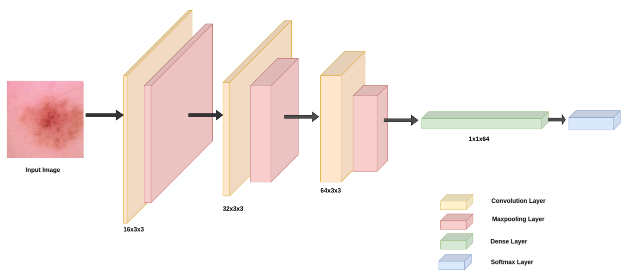


Fig. 2: Architecture of the CNN model used.

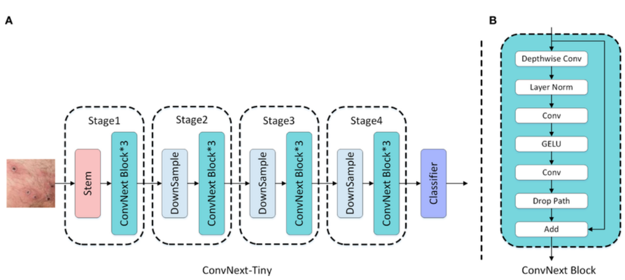


Fig. 3: Architecture of the ConvNeXt model used.

V. Experiments

Training was conducted on a cloud-based GPU environment using **JarvisLabs.AI**, equipped with an **NVIDIA RTX A5000 GPU (24 GB VRAM)**, **32 CPUs**, and **64 GB RAM**. This high-performance setup enabled efficient training with larger batch sizes and faster computation.

The model was trained with the following configurations:

- **Optimizer:** Adam
- **Loss Function:** Cross-Entropy Loss
- **Batch Size:** 8
- **Learning Rate:** 0.001
- **Number of Epochs:** 100 (ConvNext model was also trained on 170 epochs to check for local maxima)

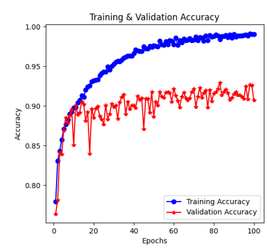
Data preprocessing, particularly augmentation and contrast adjustment, played a crucial role in stabilizing training and improving generalization. Training with preprocessed images resulted in a smoother loss curve and reduced overfitting.

VI. Results and Evaluation

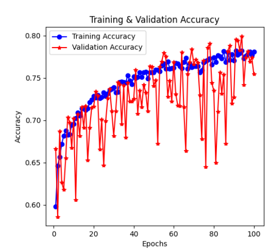
The Accuracy and F1-score for the simple CNN model are shown in Table 1. While the plots for training & validation loss, training & validation accuracy are shown in the subsequent figures. Table 2 shows a comparison between the model performance of CNN and ConvNext Model.

Metric	Value
Accuracy	90.75 (%)
F1-score	0.91

TABLE I: Evaluation Metrics for the Model



(a) Epochs vs Accuracy for CNN Model

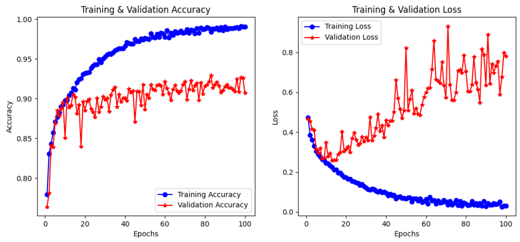


(b) Epochs vs Accuracy for ConvNext Model

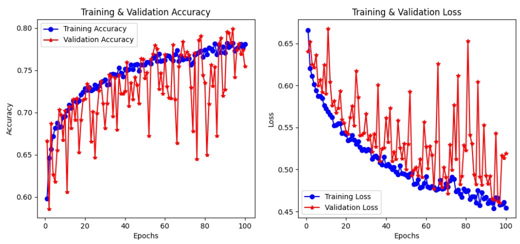
Fig. 4: No. of Epochs vs Accuracy Plot : (a) Simple CNN, (b) ConvNext.

Fig. 5: Inference Result Images

Training on the A5000 GPU significantly reduced the training time, averaging 10.66 minutes for the simple CNN Model



(a) Train vs Val Curves for CNN Model



(b) Train vs Val Curves for ConvNext Model

Fig. 6: Train vs Val Curves: (a) Simple CNN, (b) ConvNext.

Model	Training Time (mins)	Model Size (MB)	Accuracy (%)
CNN	10.66	32.01	90.75
ConvNext	78.13	108.67	75.45

TABLE II: Comparison of Inference Time, Model Size, and Accuracy

and 78.13 minutes for the ConvNext model for 100 epochs. The model inference time was benchmarked on both GPU and CPU, showing that GPU inference was significantly faster.

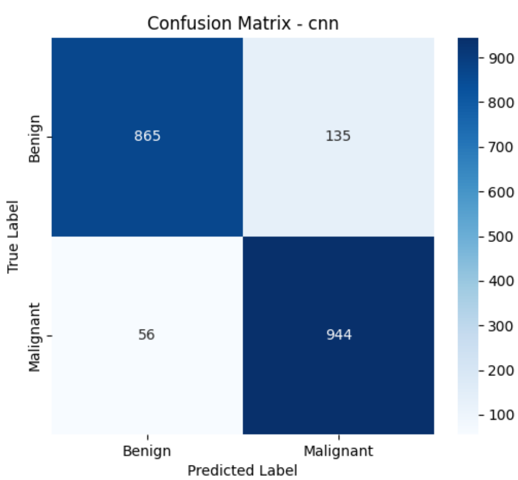


Fig. 7: Confusion Matrix for the Simple CNN Model

The plots for the simple CNN Model shows that the model is getting overfitted to the dataset. Although the test dataset being a small one, the model performance is not affected much, but this can cost us when the data is changed or increased. While on the other hand, for the ConvNext Model, even though the training is a little slow (evident from the accuracy plots), the model does not overfit to the data. To check for the efficiency of ConvNext model, its training was continued with the same parameters for 170 epochs and the resulting graphs are plotted as follows.

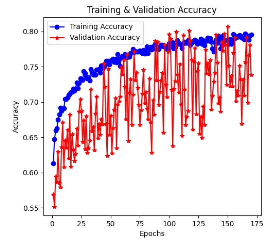


Fig. 8: Epochs vs Accuracy for ConvNext Model (170 Epochs)

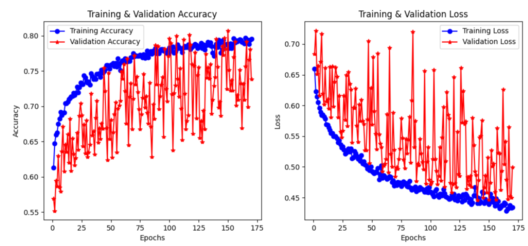


Fig. 9: Train vs Val Curves for ConvNext Model(170 epochs)

The graphs above reflects the very slow learning process for ConvNext Model. The Validation plots fluctuates a lot bringing inconsistency to the model results. Hence, further training and changes to the model loss and other parameters are required to keep the val plots in control and expedite the training process.

VII. Future Work and Conclusion

There are a lot of possible solutions to make the model more stable and accurate. The model training can use class weights to reduce the effect of imbalance in the dataset. Techniques like Regularization, Dropout, Batch Normalization, etc. can be added to model architecture and training process to reduce overfitting and increase accuracy. Vision Transformer Model can be trained with adequate data preprocessing for feature extraction to further improve effeciency. Future improvements could involve using more diverse datasets, incorporating ensemble learning, and deploying the model for real-world use.

In conclusion, this work demonstrates the feasibility of using deep learning based CNN architectures for automated skin cancer classification, with promising initial results, but further scope of improvement.