Comparative Analysis of Multiple Models for Skin Disease Detection

Priya Singh, Abhishek Basu, Hanisha Anil Mohinani, Utkarsh Farkya
The University Of Texas at Dallas
{priya.singh, abhishek.basu, hanishaanil.mohinani, utkarsh.farkya}@utdallas.edu

1. Abstract

Skin diseases are one of the most pressing global health concern which impacts millions of individuals worldwide. Prompt and accurate detection of these conditions is very vital for effective treatment and management. However, significant challenges are faced in the diagnosis phase due to the diverse nature of skin diseases.

Also, their symptoms being very similar in the early stage, and the lesion samples being extremely unbalanced, their classification is challenging. [1]

In response to these challenges, this study endeavors to conduct a comparative analysis of multiple models for skin disease detection.

The main objective of this study is to discern the most effective and reliable approach for automated skin disease diagnosis through the evaluation of various models. This study aspires to contribute to improved healthcare outcomes and enhanced patient well-being.

2. Introduction

Skin diseases pose a significant global health challenge, affecting millions of people worldwide. Early detection and accurate diagnosis are crucial for effective treatment and management of these conditions. However, the path to accurate diagnosis is often complicated by several factors, including the diverse range of skin diseases and the similarity of their symptoms in the initial stages. Moreover, the classification of these diseases is further complicated by the highly imbalanced distribution of lesion samples. This study aims to address these challenges by conducting a comparative analysis of multiple models for skin disease detection.

There have been many endeavors to implement traditional medicine across the different parts of the globe especially in the countries which are not technologically advanced, but the efforts have been met with challenges such as huge cost of medical tools and equipments and also lack of medical expertise [3].

To understand the background and scope of this study, Section 3 discusses related work in the field of skin disease detection, outlining existing approaches and identifying their limitations. Section 4 details our methodology, including information about the dataset, data preprocessing, and the model frameworks used in this study. Section 5 presents the results of our experiments and evaluations, showcasing the performance of different models in detecting various skin diseases. Finally, Section 6 concludes the study with a summary of findings and potential directions for future research.

3. Related Work

Recent advancements in skin disease classification and diagnosis have been propelled by innovative approaches leveraging convolutional neural networks (CNNs). For instance, recent research by [1] introduced a model fusion technique that enhances feature extraction capacity and classification accuracy by combining deep and shallow features and integrating an attention module. This approach surpassed baseline models like DenseNet201 and ConvNeXt_L, showcasing superior performance in classifying skin diseases. Moreover, similar work by [2] also explores challenges in skin disease classification, proposing a model fusion approach that significantly improves classification accuracy, particularly on datasets dominated by acne-like skin diseases. These studies collectively underscore the importance of model fusion and feature extraction techniques in advancing dermatological healthcare.

Furthermore, parallel advancements in skin disease diagnosis have extended to automated image-based systems, as highlighted by [3], which proposes an automated system for skin disease recognition utilizing machine learning classification techniques. This system, based on Convolutional Neural Networks (CNNs), offers enhanced accuracy and efficiency compared to traditional diagnostic methods. Additionally, the integration of machine learning algorithms with mobile technology, exemplified by the work of [4], showcases the development of mobile applications for skin disease diagnosis, providing users with timely and accurate information about their skin conditions. These advancements collectively underscore the progress in dermatological healthcare, emphasizing the potential for more accurate,

efficient, and accessible diagnostic solutions.

4. Method

The dataset that was used for the study consists of 878 labeled images and 9 different classes and is curated from the International Skin Imaging Collaboration (ISIC) gallery, accessible at ISIC Archive. Given below are the frameworks for each model analyzed in this study:

4.1. CNN

Prior to model training, the dataset underwent preprocessing to prepare the images for classification. Each image was resized to a fixed dimension of 240×240 pixels using the OpenCV library. Additionally, the pixel values of the images were normalized to the range [0, 1]. During this process, images were grouped into respective classes based on the provided ground truth labels using a dictionary mapping each class label to an integer representation.

Convolutional Layers: The CNN architecture comprises several sets of convolutional layers. Each convolutional layer applies a set of learnable filters to the input image to extract spatial features. Rectified Linear Unit (ReLU) activation functions are applied after each convolutional operation to introduce non-linearity into the model.

- **Conv2D (32 filters, 3×3 kernel)**: This initial layer employs 32 filters with a 3×3 kernel size to detect low-level features like edges and textures in the input image.
- **Conv2D (64 filters, 3×3 kernel)**: Following the first layer, this layer uses 64 filters with a 3×3 kernel to extract higher-level features from the image.
- **Conv2D (128 filters, 3×3 kernel)**: Two subsequent layers each use 128 filters with a 3×3 kernel size to capture increasingly complex features.
- **Conv2D (256 filters, 3×3 kernel)**: Finally, two more layers with 256 filters each and a 3×3 kernel further enhance the network's ability to recognize intricate patterns in the input.

The convolutional neural network (CNN) architecture comprises several key components. MaxPooling layers reduce spatial dimensions and extract pivotal features from feature maps between convolutional layers, aiding in feature extraction. Dropout layers, positioned after each max-pooling layer, mitigate overfitting by randomly deactivating neurons, preventing the network from relying excessively on specific features during training. Following the final convolutional layer, a Flatten layer transforms two-dimensional feature maps into a one-dimensional vector, ensuring compatibility with subsequent fully connected layers. These dense layers learn non-linear relationships between features and class labels, facilitated by ReLU activation functions applied to all but the last layer. The output layer, consisting of 9 neurons corresponding to classification classes, utilizes a softmax activation function to convert raw predictions into class probabilities, enabling effective multi-class classification.

4.2. DenseNet

DenseNet, a variant of convolutional neural networks, features dense connections between layers through "Dense Blocks." These blocks link layers directly, provided their feature-map sizes are compatible. Each layer in this framework receives inputs from all preceding layers and transmits its own feature-maps to subsequent layers in a feed-forward manner. Prior to model training, data preprocessing was conducted similarly to CNN, as detailed in Section 4.1. The dataset comprises 897 images spanning nine classes of skin disease images, split into a training and test set at an 80:20 ratio. Within the training set, a further 80:20 split was performed to create a validation set comprising 20% of the training data. Categorical encoding was employed to represent each image class uniquely within a dictionary data structure.

Dense Layer Block: A dense block architecture is very simple in nature. A node in each of the intermediate layer receives a neural connection from each of the nodes present in the previous layer. This is why dense layers are also called as feed-forward fully connected layers. Mathematically, the output for a dense connection is a linear operation on the input feature vector which is followed by an activation function for inducing non-linearity. Figure 1 shows a pictorial representation of a dense block. Each subsequent layer receives a concatenated connection from the previous layer which results in subsequent layers having a collective knowledge base from the previous layers. The feature dimensions remain same across all dense layers.

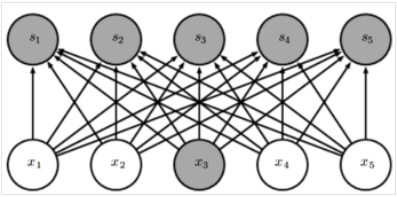

Figure 1. Dense Connections

Dropout: Dropout is a regularizing technique which randomly sets a few input features to a layer as zero for ensuring that these layers don't becomes co-dependent on each other. It is an important tool that plays a major role in avoiding over-fitting when size of dataset is small.

Feature Propagation: A basic CNN can resemble a DenseNet architecture by repetitively using fully connected layers, but such a framework would be prone to the vanishing gradient problem whereby certain features get suppressed due to near-zero gradient during the model training. DenseNet is optimized to alleviate the vanishing gradient problem and ensure a stronger feature propagation. The dense blocks help in reusing the features while having narrower layers that ultimately ends up in having fewer parameters to learn during the model training phase.

4.3. ResNet

Another convolutional model that is used in this comparative study is the Residual Network i.e. ResNet. ResNet is a state-of-the-art image classification model that can be considered as an upgraded version of the VGG architecture, the difference being that ResNet consists of skip connections. Skip connections is a clever way of solving the vanishing gradient problem which occurs in deep neural network models. In earlier CNN architectures as more and more layers were added to the neural network, it was observed that the performance of the model started to drop because the layers failed to retain the trends of images captured as we go deeper through the layers. Skip connections help in overcoming this drawback by skipping training from a few layers and connecting it to the output. This helps the network skip the layers, which are hurting the performance of the model, allowing the use of deep networks without loosing the important features.

Architecture: ResNet is made up of two types of skip connections, Identity block and the Bottleneck/Convolutional block. The only difference between these two block is that the convolutional block perform a convolution followed by batch normalisation on the residue before adding it to the output while identity adds the residue directly to the output. Figure 2 shows the skip connection blocks, left block in the figure is the identity block while the right block shows the architecture of the convolutional block.

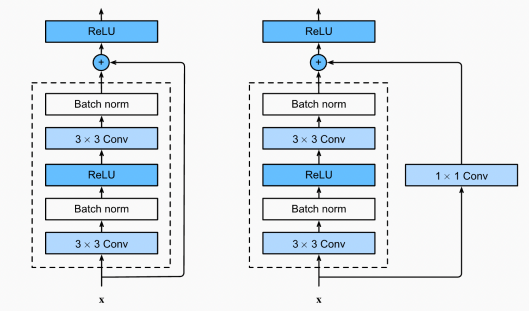

Figure 2. Skip Connections

These two blocks make up most of the ResNet models. Different types of ResNet models can be can be develoepd based on the depth of the network like ResNet-50 or ResNet-152. The number at the end signifies the number of layers in the network. Detailed studies have shown that emperically ResNet are easier to optmize and its accuracy increase with considerable depth increase. For this study, we have limited ourselves to ResNet-18 and ResNet-34.

4.4. MobileNet

Prior to model training, each image in the dataset was resized to a standard 240×240 pixel format using OpenCV. To prepare for neural network input, the pixel values were scaled to the $[0, 1]$ range. The labels were then converted to a one-hot encoded format for use in multi-class classification tasks.The MobileNet-based framework for the skin disease detection task starts with a pre-trained MobileNet model and includes additional layers to customize it for the specific application.

1. **Base Model- MobileNet:** The MobileNet model, pretrained on the ImageNet dataset, serves as the base of this framework. It uses depthwise separable convolutions to reduce computational complexity while maintaining a high level of accuracy.

2. **Freezing Layers:** All layers in the base MobileNet model are frozen, ensuring that pre-trained weights remain unchanged during the initial training.

3. **Dropout Layer:** After the base model, a dropout layer with a rate of 30% is added. This helps prevent overfitting by randomly deactivating a portion of the neurons during training.

4. **Convolutional Layer:** A Convolution2D layer with 32 filters and a 3×3 kernel is included with padding set to 'same'. This layer applies L2 regularization (weight decay) to reduce overfitting risk and extracts additional features from the MobileNet output.

5. **Global Average Pooling Layer:** The GlobalAveragePooling2D layer flattens the spatial dimensions into a single feature vector, minimizing model parameters while retaining significant information.

6. **Fully Connected Layers:** Following the Global Average Pooling layer, a Dense layer with 128 units and ReLU activation is used for further feature extraction and applying L2 regularization to reduce overfitting.

7. **Output Layer:** The final output layer has 9 neurons, corresponding to the number of target classes in the classification task. A softmax activation function converts raw outputs into class probabilities for multi-class classification. The Adam optimizer is used with

a learning rate of 0.00001 and the model is compiled with categorical cross-entropy loss and accuracy as the metric. Overall, this MobileNet-based framework balances pre-trained feature extraction with customized layers for fine-tuning and adapts to the specific task of multi-class skin disease classification.

5. Experiments

5.1. CNN

The performance of the CNN model on the test dataset is summarized in Table 1:

Metric	Value (%)
Train Accuracy	70.94
Test Accuracy	53.04
Precision	45.46
Recall	53.03

Table 1. Evaluation metrics of the CNN model

The model achieved a train accuracy of 70.94% and a test accuracy of 53.04%. Precision is calculated as 45.46%, while recall is 53.03%.

The Receiver Operating Characteristic (ROC) curve (Figure 3) illustrates the performance of the CNN model across various threshold settings. The ROC curve plots the true positive rate against false positive rate, providing insights into model's ability to discriminate between different classes.

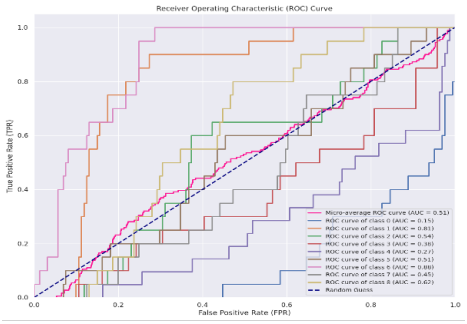

Figure 3. Receiver Operating Characteristic (ROC) curve of the CNN model

The confusion matrix (Figure 4) provides a detailed breakdown of the model's predictions across different classes. From the confusion matrix, it can be observed that classes 6, 8, and 5 are frequently misclassified, indicating areas where the model may require further improvement.

5.2. DenseNet

The performance of DenseNet is summarized in Table 2. The model achieved a train accuracy of 73.97% and a test accuracy of 61.33%. These values do look decent to an extent, but it is important to note the low validation accuracy of 22.86%. The evaluation metrics show a low value for precision and recall indicating a considerable amount of false positives and false negatives in the predictions.

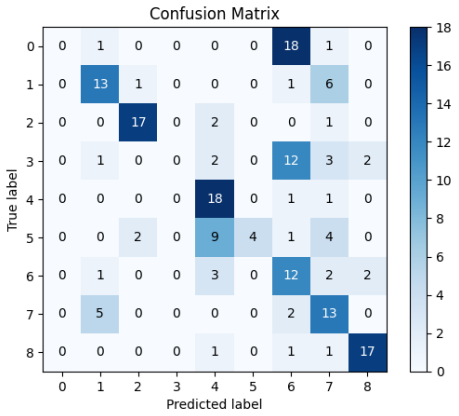

Figure 4. Confusion Matrix of the CNN model

Metric	Value (%)
Train Accuracy	73.97
Validation Accuracy	22.86
Test Accuracy	61.33
Precision	32.66
Recall	33.86

Table 2. Evaluation metrics of DenseNet model

The Receiver Operating Characteristic (ROC) curve shown in Figure 5 illustrates the performance of the DenseNet model. The ROC curve plots the true positive rate against the false positive rate, providing insights into the model's ability to discriminate between different classes and this clearly shows that there are certain classes of images that explains why DenseNet achieves a low precision and recall despite having a decent training and testing accuracy.

Analysis of the confusion matrix is important to determine the quality of this DenseNet model. Figure 6 shows the confusion matrix based on predictions made by DenseNet during the testing phase. It is evident from Figure 6 that there are three classes of images that are contributing to the poor performance metrics. Class 5 has got significant number of misclassifications as DenseNet predicts it as Class 4 majority of the times. Similarly, Class 0 and Class 3 images are confused as some other class of images as per predictions made by DenseNet.

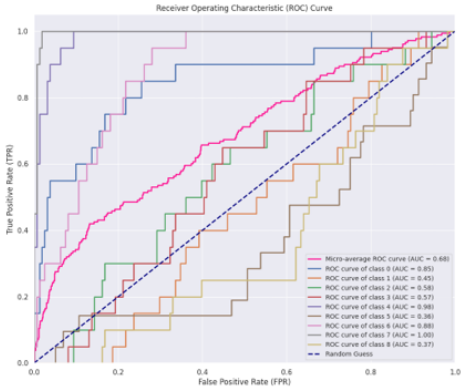

Figure 5. Receiver Operating Characteristic (ROC) curve of the DenseNet model

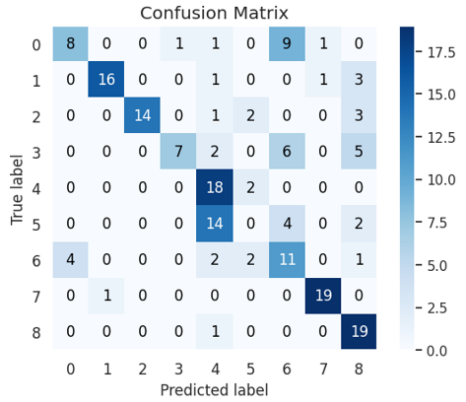

Figure 6. Confusion Matrix of the DenseNet model

5.3. ResNet

The performance of ResNet is summarized in Table 3. The models (ResNet-18 and ResNet-34) achieved train accuracy's of 91.12% and 91.12% and test accuracy's of 61.57% and 58.17%. These values do look decent to an extent, but it is important to note the low validation accuracy's for both of the models. The evaluation metrics show a fairly mid-range value for precision and recall indicating a considerable amount of false positives and false negatives in the predictions.

Metric	ResNet 18 (%)	ResNet 34 (%)
Train Accuracy	91.12	88.37
Test Accuracy	61.57	58.17
Precision	56.32	45.78
Recall	62.26	50.01

Table 3. Evaluation metrics of ResNet model

Analysis of the confusion matrix is important to determine the performance of the ResNet models. Figure 7 and 8 shows the confusion matrices based on predictions made by ResNet models during the testing phase. It is evident from Figure 7 that there are three classes of images that are contributing to the poor performance metrics of Resnet-18 model. Class 3 has got significant number of misclassifications. On the other hand, it can be seen from 8 that ResNet-34 is performing poorly as compared to ResNet-18, exactly opposite of what the theories suggest. Classes 0, 3, 4, 5, 6 has got significant number of misclassifications that degrades the performance of the model.

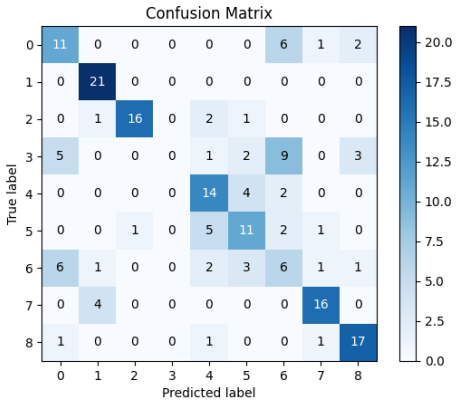

Figure 7. Confusion Matrix of the Resnet-18 model

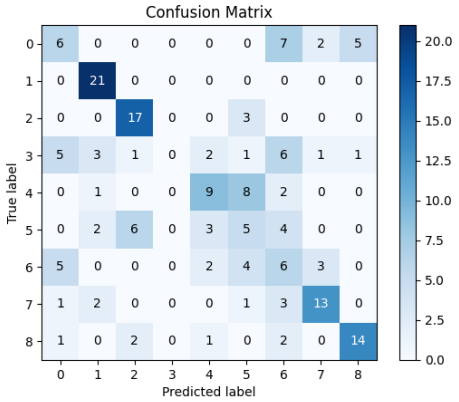

Figure 8. Confusion Matrix of the Resnet-18 model

5.4. MobileNet

The performance of the MobileNet model on the dataset is summarized in Table 4. The model achieved a training accuracy of 96.84% and a test accuracy of 80.11%. Precision was 80.58%, indicating accuracy of positive predictions, while recall was 80.01%, reflecting detection of actual positives.

The Receiver Operating Characteristic (ROC) curve (Figure 9) above demonstrates the performance of the Mo-

Metric	Value (%)
Train Accuracy	96.84
Test Accuracy	80.11
Precision	80.58
Recall	80.01

Table 4. Evaluation metrics of the MobileNet model

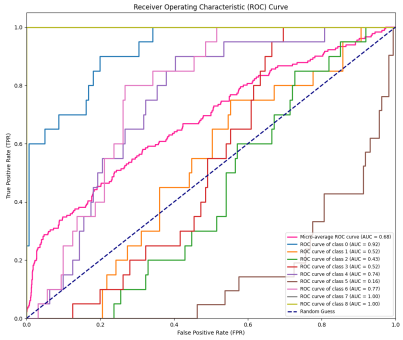

Figure 9. Receiver Operating Characteristic (ROC) curve of the MobileNet model

bileNet model across all nine classes. It can be clearly seen from the ROC Curve that for most classes, the true positives and false positves are being predicted correctly and only for a few classes, the model does not perform well.

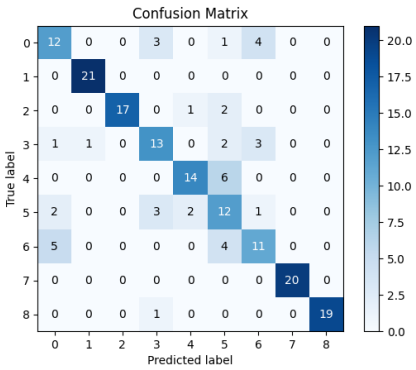

Figure 10. Confusion Matrix of the MobileNet model

The confusion matrix in Figure 10 provides insights into the accuracy of predictions across the nine classes in the MobileNet model. It shows that most classes are predicted correctly, but classes 0, 5, and 6 tend to be misclassified and show poor performance. It is also evident that for class 1 and 7, all the images have been predicted correctly, while for other few classes, the model can be tuned so as to provide significant improvement in prediction.

6. Conclusion

Convolution Neural Network (CNN): The CNN, the most basic neural network in this project, performs decently but exhibits numerous misclassifications across the image dataset. This performance likely stems from varied convolution blocks in the framework. Further improvement might result from testing different block combinations and fine-tuning parameters.

DenseNet: DenseNet turns out to be a very powerful tool that has been employed on a very small dataset which makes it prone to over-fitting. One solution to improve DenseNet performance on small dataset is to gradually increase the DropOut which nullifies a few neural networks and tries to avoid over-fitting. The model in this report strikes the best balance between Dropout and DenseNet's feature reuse. Despite outperforming CNN, there are still a significant number of misclassifications as seen in Figure 5 and Figure 6. Hence, DenseNet isn't optimal for this task.

ResNet: ResNet, like DenseNet, faces overfitting challenges on small datasets, leading to imperfect classification. Introducing dropout blocks could alleviate this issue. Limited training data and involvement of validation set contributes to ResNet's underperformance and hinders generalization to the test set. Utilizing transfer learning with a pre-trained CIFAR model could enhance the performance.

MobileNet: As visible from Figure 10, MobileNet comes out to be the best model trained under our experiments. The performance can be further improved using transfer learning and regularization techniques. Also, the dataset was very small, so we can use technique like data augmentation that can also help improving the performance of the model.

Final Verdict: MobileNet emerges as the top performer for this image classification task based on the listed model performances. The confusion matrix for MobileNet (Figure 10) demonstrates its superior prediction accuracy with minimal misclassifications. Further enhancements can be achieved through techniques like transfer learning, regularization, and additional image augmentation. These strategies promise improved results not only for MobileNet but also for the other neural network frameworks in this project.

References

[1] Mingjun Wei, Qiwei Wu, Hongyu Ji, Jingkun Wang, Tao Lyu, Jinyun Liu, and Li Zhao. 2023. "A Skin Disease Classification Model Based on DenseNet and ConvNeXt Fusion." *Electronics* 12, no. 2: 438. https://doi.org/10.3390/electronics12020438 1

[2] A. Ajith, V. Goel, P. Vazirani and M. M. Roja, "Digital dermatology: Skin disease detection model using image processing" *ICICCS*, 2017, vol. 1, pp. 1–10, 2022. https://doi.org/10.3390/electronics12020438 1

[3] J. Rathod, V. Waghmode, A. Sodha and P. Bhavathankar, "Diagnosis of skin diseases using Convolutional Neural Networks" *ICECA*, 2018, pp. 1048-1051, doi: 10.1109/ICECA.2018.8474593. https://ieeexplore.ieee.org/document/8474593 1

[4] P. Sunil Kumar, P. Sundaresan, R. Logith and N. Mathivanan, "Skin Disease Detection based on Image Processing Technique" *ICICCS*, 2023, pp. 1258-1263, doi: 10.1109/ICICCS56967.2023.10142281. https://ieeexplore.ieee.org/document/10142281 1

[5] R. G., S. G. A., K. S. and R. R, "Skin Disease Detection based on Machine Learning Techniques" *SMART*, 2022, pp. 1249-1252, doi: 10.1109/SMART55829.2022.10047634. https://ieeexplore.ieee.org/document/10047634

[6] K. Roy, S. S. Chaudhuri, S. Ghosh, S. K. Dutta, P. Chakraborty and R. Sarkar, "Skin Disease detection based on different Segmentation Techniques" *Optronix*, 2019, pp. 1-5, doi: 10.1109/OPTRONIX.2019.8862403. https://ieeexplore.ieee.org/document/8862403