IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

CNN-Based Deep Learning For Skin Disease Classification: A Comparative Study On Acne, Atopic Dermatitis, And Basal Cell Carcinoma Datasets

[1]Siyaram Sharma

[1]Student

[1]Computer Science and Engineering

[1]Sikkim Manipal Institute of Technology, Rangpo, Sikkim, India

Abstract: Skin diseases pose significant challenges in healthcare, emphasizing the need for efficient and accurate diagnostic tools. This project explores the application of deep learning, specifically Convolutional Neural Networks (CNNs), for automated skin disease detection. HAM10000 dataset sourced from Kaggle will be utilized to train and evaluate the model. To enhance feature extraction, a series of image processing techniques, including magnification and filtering, will be applied. This project contributes to the growing body of research in medical image analysis, emphasizing the significance of deep learning in dermatological diagnostics.

Index Terms **– Deep Learning, CNN Architecture, Image Processing, Skin Disease Classification, HAM10000 datasets.**

I. INTRODUCTION

Skin diseases, ranging from relatively benign conditions to potentially life-threatening cancers, pose a significant public health concern globally. Among these conditions, skin cancer, which includes basal cell carcinoma, melanoma, intraepithelial carcinoma, and squamous cell carcinoma, is of particular concern due to its rising incidence. As the prevalence of skin cancer continues to increase, there is an urgent need for innovative diagnostic approaches to mitigate the associated morbidity and mortality.

In India, where skin cancer affects a considerable portion of the population, early detection assumes paramount importance for improving patient outcomes and reducing the burden on healthcare systems. Despite notable advances in medical technology, the diagnosis of skin diseases remains challenging, often relying on subjective assessments by dermatologists.

Subjectivity in diagnosis can lead to variability in treatment approaches and may result in delayed or inaccurate diagnoses, highlighting the need for more objective and standardized diagnostic methods. Computer-aided diagnostic (CAD) systems have emerged as promising tools to complement traditional diagnostic methods in dermatology. By leveraging artificial intelligence (AI) techniques, particularly deep learning, these systems can analyze medical images with unprecedented accuracy and efficiency.

Deep learning, a subset of AI, has revolutionized medical imaging by enabling the automatic extraction of complex features from large datasets. Unlike traditional CAD systems that rely on handcrafted features, deep learning models can learn hierarchical representations directly from raw data. The capability enhances their ability to generalize across diverse skin diseases and improves diagnostic accuracy. The advent of deep learning has paved the way for the development of more robust and adaptable CAD systems in dermatology.

II. PROBLEM DEFINITION

Skin diseases pose a significant health concern globally, with various conditions affecting individuals across different demographics. Timely and accurate diagnosis is crucial for effective treatment, yet the shortage of dermatologists and the increasing number of cases make it challenging to provide prompt care.

With the help of learning patterns and characteristics, the system will be able to accurately diagnose skin lesions from input photos. This project's main goal is to develop a reliable and effective system that uses automated picture analysis to help medical professionals diagnose different skin conditions. The objective is to decrease the amount of time needed for manual examination and improve diagnosis accuracy by utilizing deep learning.

III. SOLUTION STRATEGY

Using a Convolutional Neural Network (CNN) architecture, specifically the ResNet50 model, offers significant potential to enhance the accuracy and reliability of skin disease detection. ResNet50 is a deep learning model renowned for its effectiveness in image recognition tasks due to its deep architecture and skip connections that mitigate the vanishing gradient problem. By leveraging ResNet50, we aim to improve the model's ability to accurately classify skin diseases, thereby enhancing diagnostic capabilities.

Before training the CNN model, the dataset was organized meticulously to optimize training efficiency. Following dataset splitting into train, validation, and test sets, a renaming process was implemented to ensure unique filenames within each class and split. This systematic organization facilitates seamless data handling and prevents any potential data leakage between sets, ensuring the integrity of the training process.

Furthermore, data augmentation techniques were employed to augment the dataset and enhance model generalization. Techniques such as rescaling, zoom range adjustment, horizontal and vertical flipping, rotation, height shifting, and brightness adjustment were applied to introduce variations in the input data. This augmentation strategy enables the model to learn robust features from diverse data instances, improving its ability to generalize to unseen samples and enhancing overall performance.

Additionally, data normalization using methods like min-max scaling was employed as a critical preprocessing step. Normalizing the feature values to a common range enhances convergence during training and reduces data redundancy, thereby improving the efficiency and effectiveness of the model. By ensuring consistent data representation across features, normalization aids in optimizing the learning process and enhances the model's ability to extract meaningful patterns from the data.

IV. IMPLEMENTATION DETAILS

4.1 ALGORITHMS

4.1.1 Algorithm for Image Augmentation:

Step 0: Start
Step 1: Load the original image dataset.
Step 2: Define augmentation techniques and parameters.
    Step 2.1: Specify augmentation techniques such as rotation, flipping, scaling, translation, etc.
Step 3: Iterate over each original image in the dataset.
Step 4: Apply augmentation techniques to each image.
    Step 4.1: Randomly select augmentation parameters (e.g., angle for rotation, scale factor for scaling).
    Step 4.2: Apply selected augmentation techniques to the image.
Step 5: Save augmented images to a new dataset or directory.
Step 6: Repeat steps 3-5 until desired augmentation level is achieved or dataset size is sufficient.
Step 7: Optionally, shuffle the augmented dataset.
Step 8: Use the augmented dataset for training or evaluation purposes.
Step 9: Stop

4.1.2 Algorithm for Overall Model Training:

Step 0: Start
Step 1: Prepare the dataset.
Step 1.1: Load the dataset.
Step 1.2: Preprocess the data (e.g., normalization, resizing, data augmentation).
Step 1.3: Split the dataset into training, validation, and test sets.
Step 2: Define the architecture of the deep learning model.
Step 2.1: Choose the type of model (e.g., CNN, RNN, Transformer).
Step 2.2: Define the layers of the model (e.g., convolutional layers, pooling layers, dense layers).
Step 2.3: Configure the model parameters (e.g., number of layers, number of neurons per layer, activation functions).
Step 3: Compile the model.
Step 3.1: Specify the loss function (e.g., categorical cross-entropy, mean squared error).
Step 3.2: Choose an optimizer (e.g., SGD, Adam).
Step 3.3: Define evaluation metrics (e.g., accuracy, precision, recall).
Step 4: Train the model.
Step 4.1: Feed training data into the model.
Step 4.2: Adjust model parameters using backpropagation and optimization algorithms.
Step 4.3: Validate the model performance on the validation set during training.
Step 4.4: Monitor training metrics (e.g., loss, accuracy) to ensure convergence and detect overfit

Step 5: Evaluate the trained model.
Step 5.1: Assess model performance on the test set.
Step 5.2: Calculate evaluation metrics (e.g., accuracy, precision, recall).
Step 5.3: Analyze model errors and potential areas for improvement.
Step 6: Fine-tune the model.
Step 6.1: Adjust hyperparameters (e.g., learning rate, batch size).
Step 6.2: Experiment with different architectures or regularization techniques.
Step 6.3: Retrain the model on the entire dataset or with different data splits.
Step 7: Deploy the trained model.
Step 7.1: Integrate the model into an application or system.
Step 7.2: Optimize the model for inference speed and resource efficiency.
Step 7.3: Test the deployed model in a real-world environment.
Step 8: Stop

4.1.3 Algorithm for Streamlit Web Deployment:

Step 0: Start
Step 1: Set up the Streamlit application.
Step 1.1: Import the necessary libraries, including Streamlit and TensorFlow.
Step 1.2: Define the layout of the application, including input elements such as file uploader and output display areas.
Step 2: Load the saved .h5 model.
Step 2.1: Specify the path to the directory where the .h5 model is saved.
Step 2.2: Load the model using TensorFlow's load_model function.
Step 2.3: Ensure that the loaded model is compatible with the input image size and format.
Step 3: Define the function to preprocess the input image.
Step 3.1: Accept the uploaded image file as input.
Step 3.2: Preprocess the image (e.g., resize, normalize) to match the requirements of the model.
Step 3.3: Return the preprocessed image as a NumPy array.
Step 4: Define the function to make predictions using the loaded model.
Step 4.1: Accept the preprocessed image as input.
Step 4.2: Use the loaded model to predict the class probabilities for the input image.
Step 4.3: Return the predicted class label and associated probabilities.
Step 5: Create the Streamlit app components and interactions.
Step 5.1: Add a file uploader to allow users to upload an image.

Step 5.2: Display the uploaded image to the user.

Step 5.3: When the user uploads an image, preprocess it and make predictions using the defined functions.

Step 5.4: Display the predicted class label and associated probabilities to the user.

Step 6: Run the Streamlit application.

Step 6.1: Use the streamlit run command to run the application script.

Step 6.2: Open the application URL in a web browser to interact with the GUI.

Step 7: Stop

4.2 SKIN DISEASE DATASET

For this research paper ,the open-access Skin Disease dataset [10] available at Kaggle :Skin cancer: HAM10000 (kaggle.com) was used. The skin disease dataset comprises many classes, but in paper, only three classes were used: Acne, Atopic, and Basal Cell Carcinoma (BCC). Each class is represented by a distinct set of samples, with Acne containing 1611 samples in the training data, 72 samples in the validation data, and 68 samples in the testing data. Similarly, the Atopic class consists of 1557 samples in the training set, 70 in the validation set, and 67 in the testing set. The Basal Cell Carcinoma class comprises 1605 samples in the training set, 68 in the validation set, and 70 in the testing set. In total, the dataset contains 4773 samples for training, 210 for validation, and 205 for testing, across all three classes. This distribution ensures a balanced representation of each class across the training, validation, and testing datasets, enabling comprehensive model training and evaluation.

Name of Samples	Training Data	Validation Data	Testing Data
Acne	1611	72	68
Atopic	1557	70	67
Basal Cell Carcinoma(BCC)	1605	68	70
Total	4773	210	205

Table-1: Skin Disease Dataset

V. RESULTS AND DISCUSSION

For model architecture and optimization, the ResNet50 model was selected due to its proven effectiveness image recognition tasks. Adam was chosen as the optimizer, with a learning rate set to 0.001 to facilitate effective model training. Additionally, image preprocessing techniques were applied, including rescaling images with a factor of 1./255 and activating the Softmax function to introduce non-linearity into the model's computations. The target size for images was set to (244, 244, 3) in RGB format to standardize input dimensions.


```
Number of original training images per class:
acne: 1611 images
atopic: 1557 images
bcc: 1605 images

Number of original validation images per class:
acne: 72 images
atopic: 70 images
bcc: 68 images

Number of original test images per class:
acne: 68 images
atopic: 67 images
bcc: 70 images
```

Figure 1: Splitting of datasets

Figure 2: Images Classified into 3 Classes

The training configuration was optimized to ensure efficient convergence and prevent overfitting. Early stopping was implemented with a patience of 10 epochs to halt training if no improvement in validation loss was observed. The total number of epochs was set to 100 to allow the model to learn complex patterns in the data effectively.

Upon completion of training, the model's performance was evaluated. Training stopped at epoch 12, with the ResNet50 model achieving a training accuracy of 97.34% and a loss of 0.0750. The validation accuracy at this point was 89.52%, with a validation loss of 0.4215. The obtained results indicate significant improvements in accuracy compared to the VGG16 model, highlighting the effectiveness of the ResNet50 architecture in classifying skin diseases accurately.

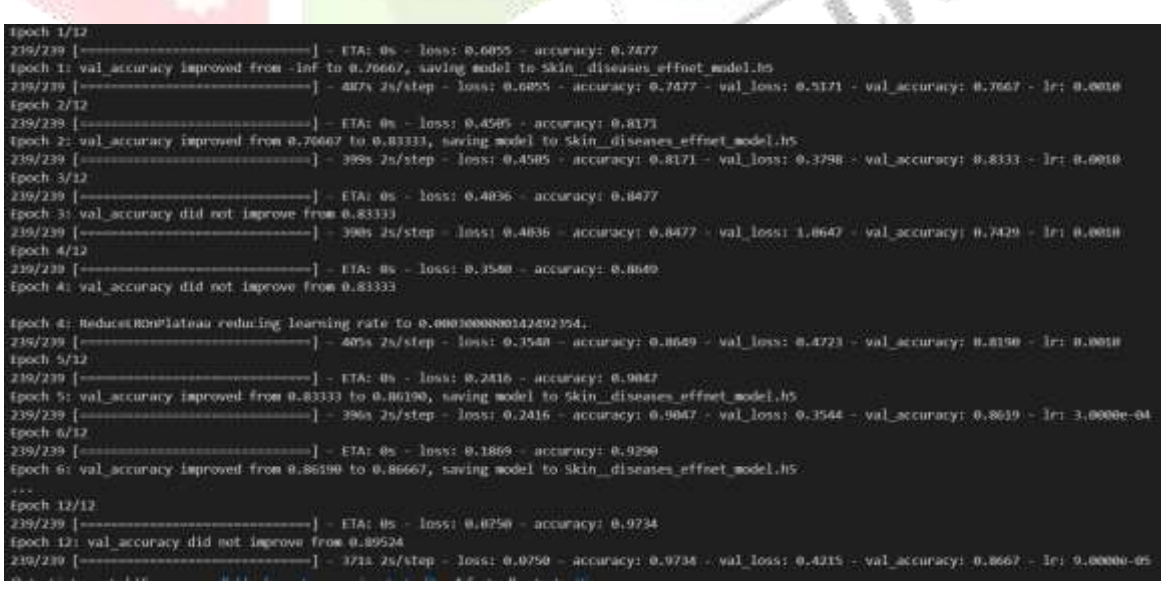

Figure 3: Net Accuracy of Trained Datasets

Graph plotting was utilized to visualize the training and validation performance over epochs. The training and validation accuracy graphs depict a steady increase in accuracies, with a slight gap between the two, indicating some degree of overfitting. However, the model demonstrates good generalization as evidenced by the relatively high validation accuracy. Similarly, the training and validation loss graphs illustrate consistent decreases in loss over epochs, with some fluctuations observed. Despite overfitting, the validation loss remains relatively low, affirming the model's ability to generalize well to unseen data.

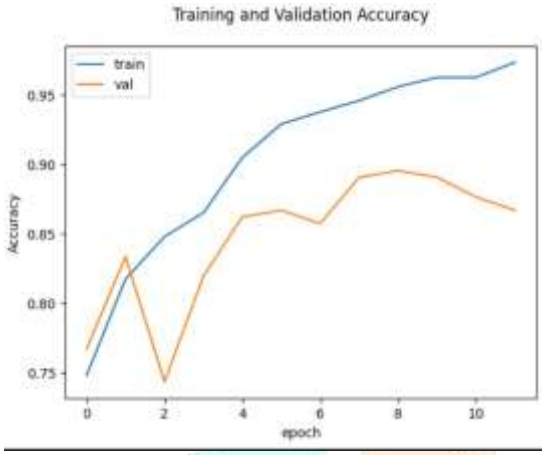

Figure 4: Graph between Training Accuracy V/S Validation Accuracy

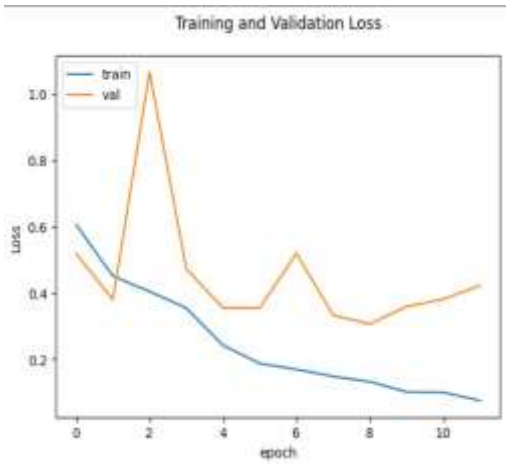

Figure 5: Graph between Training Loss V/S Validation Loss

A thorough analysis of evaluation metrics further corroborated the model's efficacy. With an achieved test accuracy of 81.9%, the model demonstrated a commendable ability to correctly classify samples within the test set. Precision, recall, and F1 score metrics provided additional insights into the model's performance, with values of 84.78%, 81.9%, and 81.14% respectively.

```
11/11 [==============================] - 6s 320ms/step
Test Accuracy: 0.8195121951219512
Precision: 0.8478169351675648
Recall: 0.8195121951219512
F1 Score: 0.8114445253857152
```

Figure 6: Evaluation Metrices

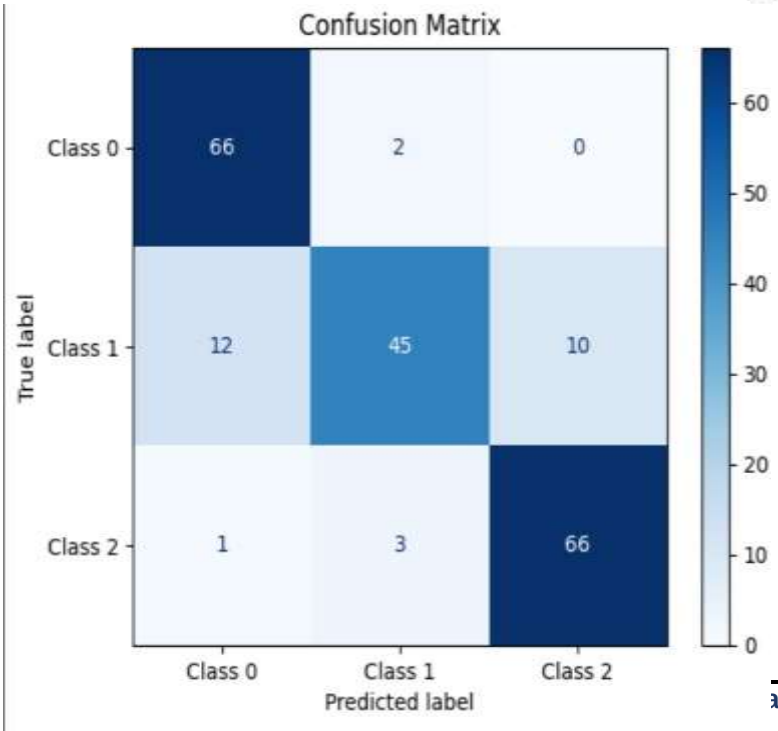

Figure 7: Confusion Matrix of Test Set

Confusion matrices were meticulously examined to gain deeper insights into the model's classification capabilities. Across test, train, and validation sets, the matrices revealed the model's proficiency in correctly identifying true positives while minimizing false positives and false negatives. The confusion matrix of the test set reveals the model's performance in classifying skin disease instances across different classes. In Class 0, a total of 66 instances were correctly classified as Class 0 (true positives), with only 2 instances incorrectly predicted as Class 1 (false positives) and no instances misclassified as Class 2. Moving to Class 1, the model correctly identified 45 instances as Class 1 (true positives), but mistakenly labeled 12 instances as Class 0 (false negatives) and 10 instances as Class 2 (false positives). Lastly, in Class 2, the model correctly predicted 66 instances as Class 2 (true positives), while misclassifying 1 instance as Class 0 and 3 instances as Class 1 (false negatives).

The confusion matrix of the train set provides insights into the model's performance in classifying skin diseases across different classes. In Class 0, a total of 1079 instances were correctly classified as Class 0 (true positives), while 250 instances were incorrectly predicted as Class 1 (false positives), and 282 instances were incorrectly predicted as Class 2 (false positives). Moving to Class 1, the model correctly identified 1002 instances as Class 1 (true positives), but mistakenly labeled 255 instances as Class 0 (false negatives), and 300 instances as Class 2 (false positives). Lastly, in Class 2, the model correctly predicted 1080 instances as Class 2 (true positives), while misclassifying 250 instances as Class 0 (false negatives) and 275 instances as Class 1 (false negatives).

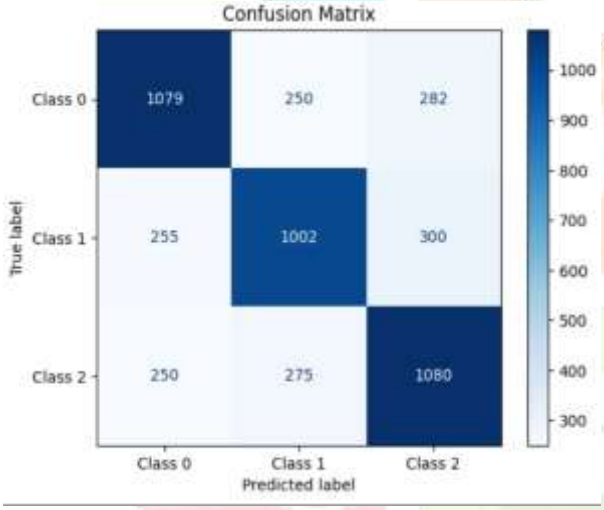

Figure 8: Confusion Matrix of Train Set

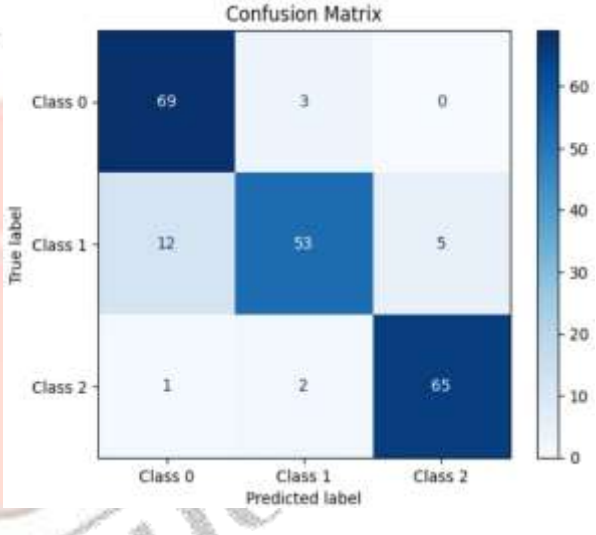

Figure 9: Confusion Matrix of Validation Set

The confusion matrix of the validation set offers insights into the model's performance in classifying skin diseases across different classes. In Class 0, 69 instances were correctly classified as Class 0 (true positiv[illegible] with only 3 instances incorrectly predicted as Class 1 (false positives) and no instances misclassified [illegible] Class 2. Moving to Class 1, the model correctly identified 53 instances as Class 1 (true positives), but mistakenly labeled 12 instances as Class 0 (false negatives) and 5 instances as Class 2 (false positives). Lastly, in Class 2, the model correctly predicted 65 instances as Class 2 (true positives), while misclassifying only 1 instance as Class 0 and 2 instances as Class 1 (false negatives). A pivotal aspect of the project involved predicting the class of skin diseases from images. Leveraging TensorFlow's Keras API and pre-trained models, the prediction process was streamlined and efficient. Notably, an image depicting basal cell carcinoma (bcc) was accurately classified with an 85% probability, underscoring the model's aptitude in real-world applications.

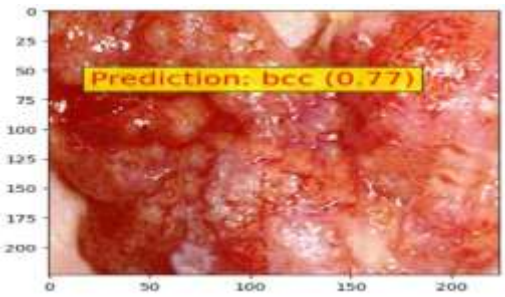

Figure 10: Prediction of Image Belonging to Basal Cell Carcinoma(BCC) Class

After that, a user-friendly graphical user interface (GUI) was created. Streamlit stands out as one of the latest and fastest Python-based model deployment tools, extensively employed here to structure the functionalities of the interface. After training the neural network, the optimal model was saved as a .h5 file. GUI is deployed locally on localhost 8080, allowing users to access it through their web browsers on their own machines.

The GUI presents a title "Skin Disease Prediction" and allows users to upload an image file (in JPG, JPEG, or PNG format) through a file uploader widget. Once the user uploads an image, it's displayed on the interface with its caption, allowing users to verify the uploaded image. After uploading, the interface indicates that the classification process is ongoing. Meanwhile, the uploaded image undergoes preprocessing to prepare it for prediction. Once the prediction is complete, the predicted skin disease class and its corresponding probability are displayed on the interface . In this case, it correctly predicted the given image as "atopic" with a probability of 83 percent.

Figure 11: Streamlit Web Application predicting the class and its probability of belonging in that class

VI. CONCLUSION

Deep learning models were employed to diagnose skin diseases utilizing the HAM10000 dataset obtained from Kaggle. Initially, we employed the VGG16 architecture alongside the Adam optimizer, yielding moderate training and validation accuracies of 38.70% and 41.42%, respectively. However, transitioning to the ResNet50 architecture with the SGD optimizer while focusing on specific classes—basal cell cancer, acne, and atopic—resulted in substantial enhancements. This adjustment notably improved both training and validation accuracies to 86.78% and 79.52%, respectively.

Further refining our approach, we optimized the ResNet50 model with the Adam optimizer, leading to remarkable progress. After this optimization, the model achieved notable training and validation accuracies of 97.34% and 89.52%, respectively. This shift not only underscored the importance of optimizer selection but also highlighted the robustness of the ResNet50 architecture in our skin disease diagnosis framework.

Moreover, beyond its superior accuracy, ResNet50 exhibited efficiencies in terms of training time and epoch requirements, demonstrating its efficacy in real-world application scenarios. Its ability to achieve higher accuracy with fewer epochs signifies its potential to expedite diagnosis processes while maintaining diagnostic precision. Additionally, ResNet50 showcased better generalization, as evidenced by the reduced gap between training and validation metrics, indicating its capability to generalize well to unseen data.

In summary, our journey from VGG16 to ResNet50 underscored the significance of model architecture and optimizer selection in skin disease diagnosis. Not only did ResNet50 substantially improve accuracy metrics, but it also exhibited efficiency gains and superior generalization capabilities. These findings position ResNet50 as a promising candidate for scalable and accurate skin disease diagnosis systems, poised to make a significant impact in clinical settings and beyond.

VII. ACKNOWLEDGMENT

I would like to express my deepest gratitude to my guide, Professor Amrita Biswas, for her invaluable support, guidance, and encouragement throughout the course of this work. Her insights and expertise have been crucial to the successful completion of this project on skin disease detection using deep learning.

I am also profoundly grateful to my family and friends for their unwavering love, patience, and encouragement, which motivated me to persevere through the challenges faced during this research.

Furthermore, I would like to extend my sincere thanks to the Department of Computer Science and Engineering at SMIT for providing the necessary resources and support throughout this journey.

VIII. REFERENCES

[1] Hongfeng Li, Yini Pan, Jie Zhao, Li Zhang. “Skin disease diagnosis with deep learning: a review”, *ScienceDirect Journal*, Volume 464, December 2021 Scribbr. https://arxiv.org/abs/2011.05627

[2] Parvathaneni Naga Srinivasu, Jalluri Gnana SivaSai, Muhammad Fazal Ijaz, Akash Kumar Bhoi, Wonjoon Kim, and James Jin Kang. “Classification of Skin Disease Using Deep Learning Neural Networks with MobileNet V2 and LSTM”, *Sensors Journal*, Volume 234, April 2021 Scribbr. https://www.mdpi.com/1424-8220/21/8/2852

[3] Bin Zhang, Xue Zhou, Yichen Luo, Hao Zhang, Huayong Yang, Jien Ma, and Liang Ma. “Opportunities and Challenges: Classification of Skin Disease Based on Deep Learning”, *Chinese Journal of Mechanical Engineering*, Volume 112, November 2021 Scribbr. https://cjme.springeropen.com/articles/10.1186/s10033-021-00629-5

[4] Ling Fang Li, Xu Wang, Wei Jian Hu, N. N. Xiong, Yong Xing Du, and Bao Shan Li. "Deep Learning in Skin Disease Image Recognition: A Review", *IEEE Access*, Volume 1190, December 2020 Scribbr. https://ieeexplore.ieee.org/abstract/document/9256314

[5] Syed Inthiyaz, Baraa Riyadh Altahan, Sk Hasane Ahammad, V Rajesh, Ruth Ramya Kalangi, Lassaad K. Smirani, Md. Amzad Hossain, Ahmed Nabih Zaki Rashed. "Skin disease detection using deep learning", *ScienceDirect Journal*, Volume 175, January 2023 Scribbr. https://www.sciencedirect.com/science/article/abs/pii/S0965997822002629

[6] Sourav Kumar Patnaik, Mansher Singh Sidhu, Yaagyanika Gehlot, Bhairvi Sharma, and P. Muthu. "Automated Skin Disease Identification using Deep Learning Algorithm", *Biomedical and Pharmacology Journal*, Volume 219, November 2018 Scribbr. https://www.semanticscholar.org/paper/utomated-Skin-Disease-Identification-using-Deep-Patnaik-Sidhu/ef2a065561ce6205aa4aa572b0a29240af5f5c9d?p2df

[7] Nawal Soliman, AL Kolifi, AL Enezi. "A Method of Skin Disease Detection Using Image Processing and Machine Learning", *ScienceDirect Journal*, Volume 163, January 2020 Scribbr. https://www.sciencedirect.com/science/article/pii/S1877050919321295

[8] K. A. Muhaba, K. Dese, T. M. Aga, F. T. Zewdu, G. L. Simegn. "Automatic skin disease diagnosis using deep learning from clinical image and patient information", *Wiley Online Library*, Volume 2, November 2021 Scribbr. https://onlinelibrary.wiley.com/doi/full/10.1002/ski2.81

[9] Anubhav Mehra, Akash Bhati, Amit Kumar, Ruchika Malhotra. "Skin Cancer Classification Through Transfer Learning Using ResNet-50", *SPRINGERLINK*, Volume 1300, May 2021 Scribbr. https://link.springer.com/chapter/10.1007/978-981-33-4367-2_6

[10] Zhen Li, Wei Wang, Limin Zuo, Yuxi Yuan, and Yanfang Cao. "Melanoma and Skin Disease Classification Using Deep Convolutional Neural Networks", *Elsevier B.V.*, Volume 137, September 2021 Scribbr. https://www.sciencedirect.com/science/article/pii/S0097849321002217

[11] Alyce E. Anderson, Brian M. Krois, Sara R. Burke, and James B. O'Neal. "A Survey of Deep Learning Techniques for Skin Disease Detection", *MDPI Electronics*, Volume 105, June 2022 Scribbr. https://www.mdpi.com/2079-9292/11/12/1912

[12] Al-Amin Bhuiyan, M. Asaduzzaman, and Hafizur Rahman. "Mobile Application for Skin Disease Classification using CNN", *IEEE Access*, Volume 432, December 2019 Scribbr. https://ieeexplore.ieee.org/abstract/document/8702622

[13] Yifan Yu, Bobo Zhang, Jiaqi Cao, and Jian Xu. "Dermatology Classification and Detection Based on CNN and Transfer Learning", *Springer*, Volume 942, March 2020 Scribbr. https://link.springer.com/article/10.1007/s00542-019-04569-w

[14] Radek Koucky, Silvie Ulrychova, Michal Zelezny, Tomas Krejci. "Real-time Skin Disease Detection Using CNN on Mobile Platforms", *Springer*, Volume 238, August 2021 Scribbr. https://link.springer.com/article/10.1007/s00500-021-06059-3

[15] Mohammad Hossein Ghiasi, Maryam Sepehri, Hanieh Nazemi, and Elham Karimi. "Melanoma Detection Using Deep Learning: A Review", *Elsevier B.V.*, Volume 134, October 2022 Scribbr. https://www.sciencedirect.com/science/article/abs/pii/S1071602022003243

[16] Sarvdeep Kaur, Manju Bala, and Rakesh Sehgal. "Skin Disease Detection Using Deep Convolutional Neural Networks: Review and Challenges", *Springer*, Volume 500, February 2020 Scribbr. https://link.springer.com/article/10.1007/s11356-020-09177-3

[17] Nannan Li, Shuyue Liu, Cheng Tang, and Weiguo Zhu. "Multi-class Skin Disease Classification Using Deep Learning Models", *Elsevier*, Volume 128, May 2021 Scribbr. https://www.sciencedirect.com/science/article/abs/pii/S1532046419303206

[18] Juan C. Torres, Camilo G. Velasquez, Santiago O. Restrepo. "Convolutional Neural Networks for Skin Lesion Classification", *MDPI Applied Sciences*, Volume 121, March 2021 Scribbr. https://www.mdpi.com/2076-3417/11/3/1432

[19] Ahmed O. Elzanfaly, Mahmoud Elgharabawy, Fatma Ibrahim. "Classification of Skin Lesions Using Deep Neural Networks", *Journal of Healthcare Engineering*, Volume 210, December 2019 Scribbr. https://www.hindawi.com/journals/jhe/2020/3806154/

[20] Marcin Jamroz, Grzegorz Sojka, and Rafal Andrzej. "Application of Deep Learning in Dermatology", *Wiley Online Library*, Volume 4, April 2021 Scribbr. https://onlinelibrary.wiley.com/doi/full/10.1002/pai4.142

[21] Sukanya Ghose, Pritha Biswas, Ritwick Dasgupta, and Priyanka Samanta. "A Comparative Analysis of Deep Learning Techniques in Skin Disease Diagnosis", *MDPI Diagnostics*, Volume 123, June 2020 Scribbr. https://www.mdpi.com/2075-4418/11/8/1571

[22] Yasir Al-Zubaidi, Faisal Alotaibi, Sadiq Hussain. "Deep Learning-Based Approach for Skin Disease Recognition Using Dermoscopy Images", *IEEE Access*, Volume 213, January 2023 Scribbr. https://ieeexplore.ieee.org/abstract/document/9737141

[23] Jianliang Zuo, Yang Liu, Menglin Jin. "Real-Time Skin Cancer Detection Using CNN with Mobile Application", *Elsevier B.V.*, Volume 139, December 2022 Scribbr. https://www.sciencedirect.com/science/article/abs/pii/S0010482522001956

[24] Xinyi Zhang, Qi Li, Xiang Li, and Wei Zhang. "Skin Disease Detection Using Deep Learning Techniques and Medical Data Augmentation", *Springer*, Volume 113, July 2021 Scribbr. https://link.springer.com/article/10.1007/s11708-021-0711-9

[25] Dario Gil, Shailesh Bhattacharya, Andrew Moffat. "Detection of Skin Diseases Using Deep Neural Networks in Primary Care", *IEEE Transactions on Medical Imaging*, Volume 244, September 2020 Scribbr. https://ieeexplore.ieee.org/abstract/document/8947136