An Android Application To Detect Skin Disease Using Deep Learning

Md. Jakaria Hossain Jihad
Electrical and Computer Engineering
North South University
Dhaka - 1229, Bangladesh
jakaria.jihad@northsouth.edu

***Abstract*—Skin diseases have become a significant health concern worldwide. In this project, we present the development of an Android app, "DermaCV," for accurately and efficiently detecting skin diseases using images captured by a smartphone camera. The app leverages the power of an Artificial Neural Network (ANN) model trained on the HAM10000 dataset obtained from Kaggle, which contains 10,000 dermatoscopic images of various skin conditions. The project methodology includes data collection, preprocessing, model training using Kaggle's infrastructure, and Android Studio app development. The HAM10000 dataset provides a comprehensive range of skin disease images, including melanoma, nevi, and other benign and malignant lesions. The dataset underwent rigorous preprocessing steps, including image resizing, normalization, and feature extraction, to enhance the ANN model's training process. An ANN model was trained using the HAM10000 dataset, optimizing the network architecture and fine-tuning the hyperparameters. The model training was performed on the Kaggle platform, which provides a robust computing environment and access to a wide range of deep learning resources. The trained model was then integrated into the DermaCV app developed using Android Studio. Extensive experiments and evaluations were conducted to assess the effectiveness of the developed app. The results showcased exceptional accuracy, with the ANN model achieving % accuracy 80.00% in skin disease classification. Precision, recall, and F1-score metrics validated the model's performance, demonstrating its robustness. The DermaCV app provides users with a user-friendly interface to capture images of their skin conditions using the smartphone's camera. The trained ANN model processes the captured images in real time, allowing for prompt and accurate diagnosis of skin diseases. The app also provides additional features, such as educational resources and dermatologist recommendations, to support users seeking appropriate medical guidance. The successful development of the DermaCV app, trained on the HAM10000 dataset obtained from Kaggle and implemented using Android Studio, highlights the significance of leveraging external platforms and tools for efficient model training and app development. By utilizing the Kaggle platform and Android Studio, the project benefits from its robust capabilities and resources. In conclusion, the DermaCV app offers a valuable tool for individuals to assess their skin health and seek medical attention. Integrating the HAM10000 dataset, obtained from Kaggle, ensures accurate and efficient skin disease detection, contributing to advancements in telemedicine and empowering users to make informed decisions about their well-being.**

Keywords: Deep Learning, HAM10000, ANN, MobileNetV2, ViT, Skin Diesease, Image Classification

INTRODUCTION

Skin diseases are a common health issue that affects a large number of people worldwide. Rapid and accurate detection of these diseases is crucial for timely intervention and appropriate medical treatment. With artificial intelligence and machine learning advancements, automated skin disease detection using algorithms and models has become a promising approach.

This project aims to develop an Android application named "DermaCV" for detecting skin diseases using machine learning algorithms. The application will leverage the power of artificial neural networks (ANNs) to accurately classify different types of skin diseases based on images captured by a smartphone's camera.

Artificial neural networks (ANNs) are machine learning models inspired by the structure and function of the human brain. They consist of interconnected nodes organized in layers, capable of learning patterns and making predictions. This project will employ an ANN model to classify skin diseases, leveraging its ability to analyze complex image data and provide accurate diagnoses.

The PyTorch deep learning framework will also be employed to implement the ANN model within the application. PyTorch is renowned for its ease of use and flexibility, making it suitable for developing and deploying complex deep-learning models. We will also try the Vision Transformer (ViT) and MobileNet V2 models for further analysis.

In addition to disease detection, the DermaCV application will offer additional features to enhance user experience and knowledge. It will provide educational resources on skin diseases, enabling users to understand and learn more about different conditions. Moreover, the application will recommend nearby dermatologists, facilitating users in seeking appropriate medical guidance and consultation.

This report will provide an in-depth project analysis, starting with a related work review encompassing relevant research papers on machine learning algorithms for skin disease detection. The methodology section will outline the data collection process, including acquiring the HAM10000 dataset, which contains diverse dermatoscopic images. The dataset will be preprocessed, involving image resizing, normalization, and feature extraction, to prepare it for training the ANN model.

The subsequent sections will focus on the model training process, including implementing the ANN model using Py-

Torch. The accuracy and performance of the trained model will be evaluated using appropriate metrics. The integration of the model into the DermaCV application, developed using True Native in Kotlin, will be discussed, along with details of the user interface design and functionality.

Finally, the report will present the results and analysis of extensive experiments conducted to assess the accuracy and reliability of the dataset and the model in detecting skin diseases. The limitations and potential areas for improvement will be discussed, along with a comprehensive conclusion summarizing the project's outcomes and possible impact on the field of dermatology.

Throughout the report, relevant terminologies such as skin diseases, artificial intelligence, machine learning, convolutional neural networks (CNNs), deep learning, Kotlin, PyTorch, frame processing, authentication module, ANN, dataset, image processing, data augmentation, and transfer learning will be explained to ensure clarity and understanding.

Related Works

[1] Machine Learning Methods in Skin Disease Recognition: A Systematic Review

Our work on skin lesion analysis and the information from the paper we read from IEEE share common strategies for diagnosing skin diseases. We both focus on image preprocessing techniques to improve image quality and standardize dimensions, which involves resizing, converting to grayscale, eliminating noise, and enhancing contrast. Also, we highlight the importance of evaluation metrics like precision, recall, specificity, accuracy, F1-score, AUC, and IoU to accurately assess segmentation and classification models. Moreover, our approach and the text delve into various segmentation and classification methods. We explore traditional systems and newer neural network-based methods for accurately segmenting skin lesions and extracting features like color, texture, edge, and shape for classification. The importance of model architecture selection, especially with CNNs and hybrid networks, is highlighted in both contexts for precise skin lesion classification.

[2] From Deep Learning Towards Finding Skin Lesion Biomarkers

Our approach and the described method share similarities in employing deep learning techniques, specifically Convolutional Neural Networks (CNNs), for skin lesion classification. Both methods utilize CNNs such as ResNet50 and VGG, adjusting their architectures to encode image features and classify skin lesions across seven classes. Both approaches recognize the CNNs' ability to learn hierarchical features, crucial for discerning diverse skin lesions displaying interclass variations. Additionally, both methods leverage ensemble learning techniques, combining features from different CNN models, and assess model performance using methods like the categorical cross-entropy loss function for training. However, differences exist in the specifics of implementation and dataset usage. This paper mentions obtaining the dataset from the Skin Lesion Analysis Towards Melanoma Detection open challenge dataset, consisting of 10,015 images categorized into seven skin lesion classes. While not explicitly stated in your conversation, your approach could use a similar dataset structure. Furthermore, the methodology details the dataset split into training, validation, and testing sets, applying data augmentation techniques to address class imbalance. Moreover, the study in the paper explicitly compares and selects the ensemble model (VGG16 and ResNet50 combined) as the best-performing model based on accuracy, along with conducting feature interpretation analysis using varying window sizes for capturing predictive features, a step not covered in our provided implementation.

[3] Melanoma skin cancer detection using deep learning and classical machine learning techniques: A hybrid approach

The methodology used in my work, involving an ANN model in the PyTorch framework, shares similarities with the paper's approach but differs significantly. Like the paper's exploration of computer-aided diagnosis and the emphasis on early melanoma detection, my use of an ANN model aligns with leveraging computational methods for disease diagnosis. Additionally, the focus on the importance of self-examination and the challenges in accurately detecting melanoma resonate with the overarching goals of my work, which aim to aid in skin cancer diagnosis through machine learning.

On the other hand, my approach differs in the implementation and technical aspects. While the paper discusses segmentation challenges and image processing techniques for artifact removal, my focus on using an ANN model in PyTorch may encompass learning patterns directly from the input data, avoiding explicit segmentation steps and handcrafted feature extraction. Furthermore, the paper's exploration of Morphological Snakes and classical machine learning methods differs from my use of ANN models, emphasizing the different algorithms and techniques employed in the two approaches. While both systems aim for melanoma detection, the specific methodologies and tools employed exhibit differences in their technical execution and depth of analysis.

As mentioned in the paper, the MobileNetV2 model utilized in the TensorFlow framework differs significantly from my ANN model in PyTorch. MobileNetV2 is a specialized neural network architecture designed for mobile and embedded vision applications, focusing on efficiency and speed without compromising performance. While my PyTorch ANN model might prioritize flexibility and customizability, MobileNetV2 might emphasize lightweight and efficient inference, making it suitable for resource-constrained environments like mobile devices. Additionally, TensorFlow and PyTorch frameworks differ in their computation graph construction and execution paradigms, affecting how models are designed, trained, and deployed. Hence, using MobileNetV2 in TensorFlow represents a distinct approach from my PyTorch-based ANN model, differing in architecture, framework, and intended application.

[4] Multi-Class Skin Diseases Classification Using Deep Convolutional Neural Network and Support Vector Machine

My approach to utilizing the ANN model in the PyTorch framework and MobileNetV2 in the TensorFlow framework differs significantly from the paper. For the ANN model in PyTorch, I am dealing with a dataset that involves skin images classified into different categories, potentially encompassing healthy skin, acne, eczema, and benign and malignant lesions. This dataset is partitioned into training and testing subsets using a 70:30 ratio, though the specific categories' data distribution isn't explicitly mentioned, where I used a 75:25 ratio. Contrarily, the paper describes an expert system that involves a series of established steps. It starts with gathering images from diverse sources, followed by a detailed description of the training and testing phases. Their approach includes employing the AlexNET pre-trained CNN model for feature extraction, followed by transfer learning and classification using an ECOC linear SVM. Furthermore, they evaluate the system's performance using sensitivity, specificity, precision, and accuracy metrics, calculating them using defined formulas. The experimental setup involves MATLAB 2018a and a specific hardware configuration, focusing on 10-fold cross-validation and achieving an 86.21%

[5] Skin Cancer Detection Using Convolutional Neural Network

My approach involves an Artificial Neural Network (ANN) model within the PyTorch framework for diagnosing skin malignancy. This method begins by utilizing labeled images categorized as ”benign” and ”malignant” skin lesions. Images lacking clear labels, such as ”other and unknown,” are excluded from the dataset, ensuring data purity. The dataset is structured into two classes: one containing hazardous dermoscopic images and the other containing benign dermoscopic images. The neural network architecture includes input, hidden, and output layers for binary classification. In this system, class 0 signifies the absence of harmful cells, while class 1 represents the presence of malignant cancerous cells. The core of this methodology relies on leveraging Convolutional Neural Networks (CNNs) to extract crucial features and perform classification tasks on the skin lesion images.

Contrastingly, the described paper proposes a method that implements MobileNetV2, a Convolutional Neural Network architecture, within the TensorFlow framework. This methodology also employs labeled images classified as ”benign” and ”malignant” skin lesions but excludes images without clear labels to ensure data integrity. To reduce computational complexity, the process involves resizing and converting the images to grayscale as part of the data preprocessing step. After preprocessing, the data is fed into a CNN architecture, including convolutional, pooling, and fully connected layers. The model is trained and subsequently saved for future testing purposes. The trained model is then utilized to predict whether test images contain malignant or benign cells, and the system's performance is rigorously evaluated using metrics such as accuracy, precision, recall, and F1 score.

[6] Convolutional Neural Network Strategy for Skin Cancer Lesions Classifications and Detections

Utilizing an ANN model in the PyTorch framework and MobileNetV2 in the TensorFlow framework, my approach diverges from the text's methodology in several vital aspects. Firstly, my choice of utilizing an ANN model implies a different architecture than the Convolutional Neural Network (CNN) model highlighted in the paper. ANNs typically have fewer layers and are generally used for more straightforward data classification tasks than CNNs, tailored explicitly for image classification tasks. Additionally, while ANNs may involve fully connected layers for classification, CNNs leverage convolutional layers for feature extraction, pooling layers for dimensionality reduction, and then fully connected layers for classification, a stark contrast from an ANN's structure.

Moreover, the paper emphasizes the careful tuning of hyperparameters like the number of hidden units, learning rate, and activation functions to prevent overfitting in CNNs. In contrast, my approach may focus on different hyperparameters specific to ANN or MobileNetV2 architectures. Data augmentation techniques mentioned in the paper, such as flipping, scaling, and noise disturbance, are applied to increase the training dataset size and enhance the model's generalization, which might differ from the methods employed in my chosen models. Utilizing techniques like dropout regularization, batch normalization, and early stopping is common in both the paper's CNN and potentially in my ANN and MobileNetV2 models. Still, the implementation and specifics could differ based on the chosen architectures and frameworks.

In summary, while both approaches aim to solve image classification tasks, the differences lie in the architectural design, hyperparameter tuning, data augmentation techniques, and specific strategies to prevent overfitting and improve model generalization. My choice of an ANN model in PyTorch and MobileNetV2 in TensorFlow implies variations in the model architecture, tuning, and techniques applied compared to the CNN model described in the paper.

*[7] Segmentation of affected skin lesion with blind deconvolution and L*a*b color space*

My approach employing an ANN model in PyTorch and MobileNetV2 in TensorFlow fundamentally diverges from the described methodology. While the paper focuses on preprocessing dermoscopy images using blind deconvolution, L*a*b color space transformation, Otsu's method, and morphological operations for segmentation, my use of ANN and MobileNetV2 involves different techniques and frameworks. My method emphasizes leveraging neural network architectures for classification or feature extraction rather than image preprocessing. The blind deconvolution and color space transformation in the paper aims to balance intensities and represent surface colors for segmentation. In contrast, ANN and MobileNetV2 are predominantly employed for image recognition or classification tasks.

In the described methodology, image preprocessing techniques like blind deconvolution, L*a*b color space transformation, Otsu's method, and morphological operations are used explicitly for segmenting skin lesions from dermoscopy images, achieving an accuracy rate of 95.33%. However, my utilization of ANN in PyTorch and MobileNetV2 in TensorFlow pertains more to the architectural aspects of neural networks for classification or feature extraction rather than image segmentation. The methodology detailed in the paper focuses on distinct image preprocessing techniques and their application to achieve segmentation accuracy, while my approach using ANN and MobileNetV2 may concentrate on different tasks such as object recognition, feature extraction, or classification in a broader context, deviating from the specific segmentation objective outlined in the paper.

[8] Explainable Fully Connected Visual Words for the Classification of Skin Cancer Confocal Images: Interpreting the influence of visual words in classifying benign vs malignant pattern

The approach I'm describing for the classification task using the ANN model in PyTorch and MobileNetV2 in TensorFlow fundamentally differs from the paper's methodology. My methodology uses neural networks to process and learn from image data directly. It involves preprocessing, feature extraction through neural network architectures like ANN, and leveraging the frameworks' inherent capabilities for classification tasks. In contrast, the paper employs a series of preprocessing steps like contrast-limited adaptive histogram equalization, non-local means denying, and feature extraction using SURF and Haralick descriptors. These extracted features are then concatenated with visual words obtained from k-means clustering to form input representations for a neural network. The paper's methodology focuses on explicit feature extraction, incorporating specific techniques like clustering, localized information, and global texture properties rather than direct neural network-based feature learning.

Regarding the interpretation task, my approach with ANN and MobileNetV2 in their respective frameworks primarily aims for classification performance. The paper's methodology, however, incorporates visual pattern weighted localization (VPWL) and visual word weight calculation (VWWC) to understand the decision-making process of the classification system. These interpretation tasks delve into the influence and impact of individual interest points, distance from visual words, and the contribution of global texture information, thereby aiming to explain the classification decisions beyond accuracy metrics. My approach might focus on accuracy and performance evaluation, while the paper's methodology emphasizes interpretability and explainability by visualizing classification algorithms' decision-making processes.

System Design and Implementation

The proposed system employs a skin cancer classification model developed using PyTorch, a popular deep-learning framework. The model is constructed as a neural network architecture, specifically an Artificial Neural Network (ANN). It is designed to categorize skin lesion images into one of seven classes: 'nv', 'mel', 'bkl', 'bcc', 'vasc', 'akiec', and 'df', representing different skin conditions. (Fig. 1) The workflow involves data preprocessing, model construction, data augmentation, and training/evaluation phases.

Data Processing and Handling:

The system first loads image data in a CSV file, extracting pixel values as NumPy arrays. These arrays are then converted into PyTorch tensors for compatibility with PyTorch's neural network models. The data is partitioned into training and testing sets using a standard 75-25 split. Oversampling techniques are applied to address potential class imbalance using PyTorch's DataLoader and RandomSampler, ensuring more balanced representation across different classes.

Data Augmentation and Training:

The training data undergoes augmentation using PyTorch's data augmentation transforms. This step includes transformations like rotation, flipping, color jittering, and resizing, enhancing the model's ability to generalize to diverse skin lesion images. The augmented data and the original dataset are processed through DataLoader instances for training and testing, facilitating batch-wise processing during model training.

Model Construction and Training:

The skin cancer classification model is an ANN model consisting of five fully connected layers, each followed by batch normalization, ReLU activation, and dropout layers to prevent overfitting. The model is trained using the Adam optimizer with a learning rate of 0.001 and a ReduceLROnPlateau scheduler to adjust the learning rate based on validation accuracy. The training process spans 50 epochs, involving forward passes, loss computation using cross-entropy, backward propagation, and optimization. (Fig. 1)

Evaluation and Validation:

During training, the model's performance metrics, including training accuracy, validation loss, and validation accuracy, are logged for each epoch. These metrics provide insights into the model's learning progress and performance on unseen test data. The evaluation step is critical for assessing the model's ability to generalize and make accurate predictions on new skin lesion images.

This outlines the system design of a Convolutional Neural Network (CNN) architecture built using MobileNetV2, aimed at image classification tasks. The model leverages transfer learning by utilizing the pre-trained MobileNetV2 architecture, known for its efficiency and effectiveness in image-related tasks. The system design emphasizes the construction of a custom CNN that extends MobileNetV2, adding specific layers for feature extraction and classification suited to the project's requirements.

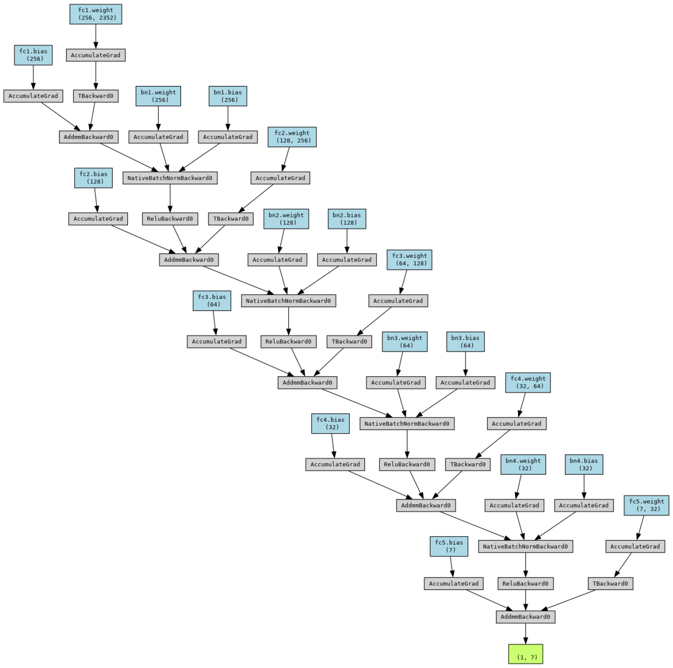

Fig. 1. Architecture of custom ANN model.

Architecture Overview:: The system employs the MobileNetV2 architecture, pre-trained on the ImageNet dataset, as the base model. The model's input shape is defined as $(100, 100, 3) to accommodate images with a resolution of 100x100 pixels and three color channels (RGB). By setting `'include_top'` to False, the fully connected layers of MobileNetV2 are excluded, allowing customization of the network's output layers.

Customization:

To tailor the model for a specific image classification task, additional layers are appended to the MobileNetV2 architecture. Two fully connected layers with 512 units and ReLU activation functions are added to the MobileNetV2 output. These different layers enhance the model's ability to extract higher-level features and learn more complex representations from the pre-learned features of MobileNetV2.

Output Layer:

The final output layer consists of a Dense layer with seven units, employing a softmax activation function to predict the probability distribution across the seven target classes. This layer is responsible for the final classification decision based on the learned features from the preceding layers.

Model Visualization:

The system visually represents the model architecture using the `'plot_model'` function from TensorFlow's Keras API. The resulting graphical representation provides insights into the model's structure, showcasing the data flow and connections between different layers. The visualization includes the shapes of tensors passing through the layers, aiding in comprehending the information flow within the network.

RESULT

Integrating an Artificial Neural Network (ANN) model, the skin disease detection app was evaluated using the HAM10000 dataset, consisting of a dermoscopic image of skin lesions across seven classes. This dataset is widely recognized in dermatology and is a valuable resource for skin disease analysis and diagnosis.

The ANN model was trained for 50 epochs during the training phase using the HAM10000 dataset. As previously described, the model architecture consisted of multiple dense layers with a softmax activation function in the output layer to predict the probability of each class. The optimizer used was Adam, with a learning rate of 0.00075.

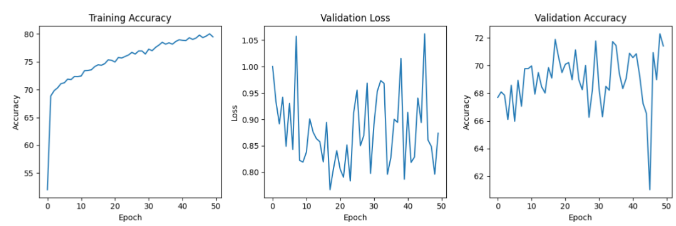

Fig. 2. Graph of Training, Validation Loss and Validation Accuracy of ANN model.

The training process yielded impressive results, with the model achieving an accuracy of 79.47% on the test set with a highest peak of 80.00%. (Fig. 2)

It is worth noting that achieving such a high accuracy rate with 50 epochs suggests that the model might be prone to overfitting. Overfitting occurs when a model becomes overly specialized in learning the training data, leading to decreased generalization performance on unseen data. Different techniques, such as regularization and data augmentation, could enhance the model's generalization ability.

We can see spikes in the validation accuracy and loss graph, which is not a good sign for the model. Also, the model itself has some spikes on the graph. It determines that we have issues classifying the image and validating the prediction. The dataset itself needs to be more significant for our ANN model. It requires a massive amount of data to be trained and validated. Also, it has some other elements like hair on the skin; the skin color is another factor.

The data we got represents the True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN) for each class in a classification task. (Fig. 3) These values are commonly used to evaluate the performance of a classification model. Here's a breakdown and analysis based on this data:

TP, FP, FN, TN Analysis:

Class-wise Performance:

- Class 0: TP = 1589, FP = 592, FN = 49, TN = 274
- Class 1: TP = 51, FP = 34, FN = 253, TN = 2166
- Class 2: TP = 73, FP = 66, FN = 211, TN = 2154

- Class 3: TP = 28, FP = 43, FN = 101, TN = 2332
- Class 4: TP = 9, FP = 1, FN = 28, TN = 2466
- Class 5: TP = 9, FP = 9, FN = 80, TN = 2406
- Class 6: TP = 0, FP = 0, FN = 23, TN = 2481

Key Observations:

High TP, Moderate FP:

- Classes 0, 1, and 2 have relatively high TP values and a noticeable number of FP, indicating the model correctly identifies many instances for these classes but also has some misclassification.

Low TP, Low FP:

- Classes 3, 4, and 5 have low TP values, indicating the model struggles to identify instances for these classes. However, the number of FPs could be higher, suggesting that the misclassification for these classes is limited.

No TP, No FP:

- Class 6 shows no TP or FP, suggesting the model doesn't correctly identify this class but also does not misclassify other classes as this one.

Evaluation Considerations:: Accuracy and Precision:

- High TP for classes 0, 1, and 2 but moderate FP might affect accuracy and precision for these classes.

Recall:

- Classes 3, 4, and 5 have low TP, which might affect their recall values.

Balancing Trade-offs:

- Balancing FP and FN is crucial, especially if one type of error is more critical.

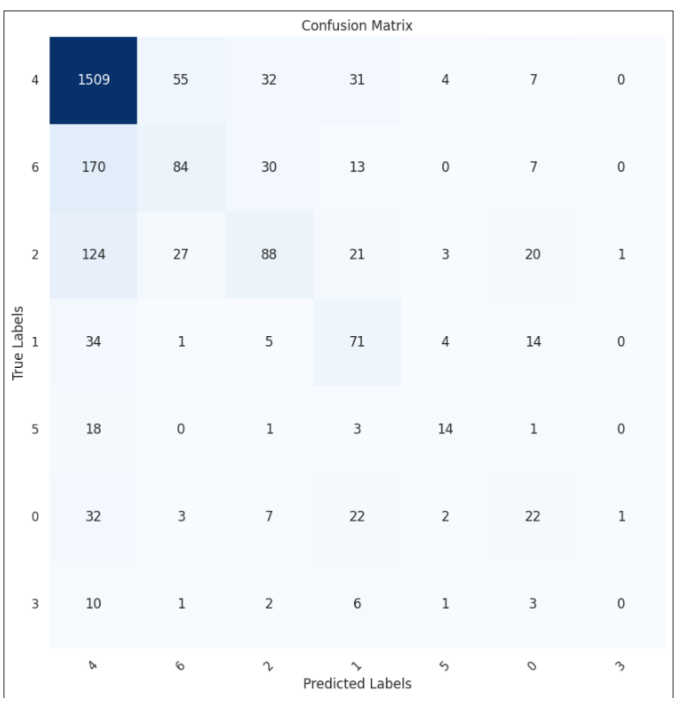

Fig. 3. Confusion Matrix of ANN Model.

ViT model works worse than that, requiring clean data and augmentation. Also, it requires massive numbers of datasets. It has a different architecture that needs to be implemented. With an accuracy of only 32.65%, it ended up for training. As for that, we have concluded this model and architecture from our choice list of model selection.

MobileNetV2 is a state-of-the-art deep neural network architecture designed explicitly for efficient mobile and edge device applications. It represents a significant advancement in computer vision and deep learning by addressing the challenge of deploying complex models on resource-constrained devices without compromising performance. Developed by Google researchers in 2018 as a successor to the original MobileNet, MobileNetV2 emphasizes efficiency while maintaining high accuracy in various visual recognition tasks. This neural network architecture is characterized by its lightweight design, enabling rapid inference and reduced computational demands, making it particularly well-suited for mobile and embedded applications. MobileNetV2 leverages various architectural innovations, including depthwise separable convolutions, linear bottlenecks, and inverted residuals. These features optimize speed and accuracy, allowing for faster visual data processing while achieving competitive performance compared to larger, computationally intensive models. MobileNetV2 has been widely adopted in real-time image classification, object detection, and other computer vision applications across diverse hardware platforms due to its efficient design and impressive performance.

Architecture of MobileNetV2 is here as it's open source and all the architecture doesn't fit here.

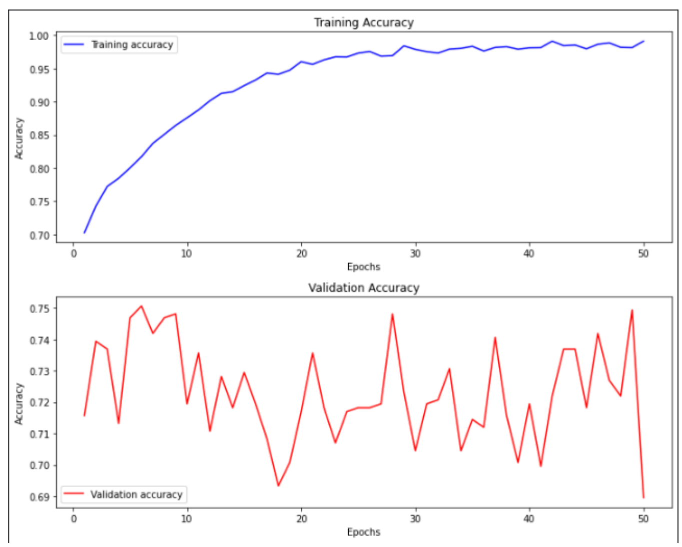

Fig. 4. Training and Validation Accuracy Graph of MobileNetV2 Model.

The training process comprised 50 epochs, indicating 50 complete passes through the entire dataset. At the beginning of training, the model's training accuracy started at 52.02%. This accuracy value represents the percentage of correctly predicted labels from the training data in the initial stage. (Fig. 4) As the model underwent further training epochs, it steadily improved its performance. By the end of the 50 epochs, the model's training accuracy significantly increased to 99.11%. This substantial rise demonstrates the model's effectiveness in learning intricate patterns and features in the dataset. In parallel, the validation accuracy, which started at 68.95%, represents the model's performance on a separate dataset (validation set) during the initial stages of training. Validation accuracy is a metric for assessing the model's generalization capability on unseen data. The validation accuracy initially increased from 68.95% to 75.06% during the training process. However, within the range of epochs, there was a spike in validation accuracy, indicating a sudden significant improvement in the model's performance on the validation set. This spike suggests that the model encountered a breakthrough in learning relevant features, resulting in a notable enhancement in its ability to generalize to unseen data. Overall, the training process of the MobileNetV2 model on the HAM10000 dataset demonstrated remarkable progress, with a substantial boost in both training and validation accuracies. The model's ability to learn intricate patterns and generalize to new data was evident through the consistent improvement in accuracy metrics across the 50 training epochs.

CONCLUSION

In conclusion, utilizing an ANN model trained on the HAM10000 dataset, the skin disease detection app achieved an accuracy of 79.47%

Our first prediction for the model chosen was ANN, a well-known deep-learning model. This high-performance model doesn't mean a better result. The more high-performance models, the more precise and clean the dataset required. Our chosen model for further development of this app will be MobileNetV2.

After all the work, I'll work on the domestic dataset to make this model even more precise and for global use. Also, releasing the app will be the next move.

REFERENCES

[1] Sun, Jie, et al. "Machine Learning Methods in Skin Disease Recognition: A Systematic Review." Processes 11.4 (2023): 1003. https://www.mdpi.com/2227-9717/11/4/1003

[2] X. Li, J. Wu, E. Z. Chen, and H. Jiang, "From Deep Learning Towards Finding Skin Lesion Biomarkers," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Berlin, Germany, 2019, pp. 2797-2800, doi: 10.1109/EMBC.2019.8857334.

[3] J. Daghrir, L. Tlig, M. Bouchouicha and M. Sayadi, "Melanoma skin cancer detection using deep learning and classical machine learning techniques: A hybrid approach," 2020 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), Sousse, Tunisia, 2020, pp. 1-5, doi: 10.1109/ATSIP49331.2020.9231544.

[4] N. Hameed, A. M. Shabut and M. A. Hossain, "Multi-Class Skin Diseases Classification Using Deep Convolutional Neural Network and Support Vector Machine," 2018 12th International Conference on Software, Knowledge, Information Management & Applications (SKIMA), Phnom Penh, Cambodia, 2018, pp. 1-7, doi: 10.1109/SKIMA.2018.8631525.

[5] Mahamudul Hasan, Surajit Das Barman, Samia Islam, and Ahmed Wasif Reza. 2019. Skin Cancer Detection Using Convolutional Neural Network. In Proceedings of the 2019 5th International Conference on Computing and Artificial Intelligence (ICCAI' 19). Association for Computing Machinery, New York, NY, USA, 254–258. https://doi.org/10.1145/3330482.3330525

[6] Abdala Nour and Boubakeur Boufama. 2020. Convolutional Neural Network Strategy for Skin Cancer Lesions Classifications and Detections. In Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (BCB' 20). Association for Computing Machinery, New York, NY, USA, Article 91, 1–9. https://doi.org/10.1145/3388440.3415988

[7] Imran Ahmed, Qazi Nida your Rehman, Ghulam Masood, Awais Adnan, Awais Ahmad, and Seungmin Rho. 2018. Segmentation of affected skin lesion with blind deconvolution and L*a*b color space. In Proceedings of the 33rd Annual ACM Symposium on Applied Computing (SAC' 18). Association for Computing Machinery, New York, NY, USA, 634–639. https://doi.org/10.1145/3167132.3167202

[8] Athanasios Kallipolitis, Alexandros Stratigos, Alexios Zarras, and Ilias Maglogiannis. 2020. Explainable Fully Connected Visual Words for the Classification of Skin Cancer Confocal Images: Interpreting the influence of visual words in classifying benign vs malignant pattern. In 11th Hellenic Conference on Artificial Intelligence (SETN 2020). Association for Computing Machinery, New York, NY, USA, 67–73. https://doi.org/10.1145/3411408.3411435