AI- Project

Skin Lesions Pigmentation Classification

Artificial Intelligence

Course Instructor

Shahela Saif

Maliaka Zafar	Talal Habib	Amna Shehzad
21I-1110	21I-1111	21I-1209

Section

SE-P

Date

May 5th , 2024

Spring 2024

Department of Software Engineering

FAST – National University of Computer and Emerging Science

Table of Contents

Table of Figures

1. Introduction

Skin lesions, areas of the skin that differ from the surrounding skin in color, shape, or texture, are a common occurrence. They can range from harmless to potentially fatal, with some being manifestations of serious underlying conditions such as skin cancer. The pigmentation colors of these lesions can provide valuable insights into their causes and types, serving as a critical diagnostic tool.

However, the manual identification and classification of skin lesions can be time-consuming, subjective, and requires expert knowledge. This is where our project comes into play. We aim to leverage the power of machine learning to address this challenge.

2. Objective

Our project's objective is to implement a multi-class classification model that can accurately detect and classify different types of skin lesions based on their pigmentation. This model will be trained to discern whether the lesions are malignant or benign, providing a rapid, objective, and reliable tool for skin lesion analysis.

By analyzing images of skin lesions, our model will not only identify the presence of a lesion but also determine its specific type. This capability will significantly enhance the speed and accuracy of skin lesion diagnosis, potentially leading to earlier interventions and improved patient outcomes.

In essence, our project seeks to harness the power of artificial intelligence to revolutionize the field of dermatology, making skin lesion diagnosis more accessible, accurate, and efficient. We believe that our work will pave the way for more advanced applications of AI in medical imaging and diagnostics, ultimately contributing to better healthcare outcomes.

3. Data Preparation

- ## Dataset:

  The dataset used for training and testing our AI model was sourced from Kaggle, specifically the HAM10000 dataset, which can be found at the following link: https://www.kaggle.com/datasets/volodymyrpivoshenko/skin-cancer-lesions-segmentation

  This dataset contains 10,015 images representing 7 different classes of skin lesions, which are as follows:

  - **AKIEC** - Actinic Keratoses and Intraepithelial Carcinoma / Bowen's Disease
  - **BCC** - Basal Cell Carcinoma
  - **BKL** - Benign Keratosis-like Lesions (Solar Lentigines / Seborrheic **Keratoses** and Lichen-Planus-like Keratoses)
  - **DF** - Dermatofibroma
  - **MEL** - Melanoma
  - **NV** - Melanocytic Nevi
  - **VC** - Vascular Lesions (Angiomas, Angiokeratomas, Pyogenic Granulomas and Hemorrhage)

  Our goal is to train our AI model using this dataset, enabling it to accurately classify any given skin lesion into one of these categories. This will provide a valuable tool in the early detection and treatment of various skin conditions.

- ## Data Preprocessing:

  The data had already been processed, and the only requirement was to map each image to a disease label. For this, we utilized the provided metadata.csv file to map our images to their corresponding labels. The labels were assigned in the following format:

  - **AKIEC** - Actinic Keratoses and Intraepithelial Carcinoma – **Label** - 3
  - **BCC** - Basal Cell Carcinoma – **Label** - 2
  - **BKL** - Benign Keratosis-like Lesions – **Label** - 4
  - **DF** – Dermatofibroma – **Label** - 5
  - **MEL** – Melanoma – **Label** - 0
  - **NV** - Melanocytic Nevi – **Label** - 1
  - **VC** - Vascular Lesions – **Label** – 6

  With these labels, we preprocessed our images for training, providing a clear and organized structure for our machine learning model to learn from. This preprocessing step is crucial for the successful training of our AI model.

4. Techniques Employed:

Some of the techniques employed for the training of our model are listed below:

- **Data Augmentation**:
  Given the limited amount of data, which needed to be further divided for training and testing, we decided to employ data augmentation. This technique artificially increases the size of our training data. We used both manual augmentation methods and the Keras library to augment our data.
- **Transfer Learning**:
  Due to the scarcity of initial data, we knew that building a Convolutional Neural Network (CNN) model from scratch would have led us to resource and training limitations. To overcome this, we employed transfer learning to aid in our model development and training. We utilized the DenseNet201 model pre-trained on the ImageNet dataset as our initial layers. However, these layers were frozen to prevent them from being trained. Instead, they served as the input layer for the additional model layers we added afterwards.
- **Uncertainty Estimation:**
  To prevent overfitting and to handle out-of-distribution samples, we utilized some uncertainty estimation principles. These included using Dropout layers in our model and applying Monte Carlo Dropout during the testing phase to obtain more accurate results.
- **Hyperparameter Tuning:**
  This involves optimizing the model as we proceed with training and retraining. We analyze the shortcomings of previous models and make necessary adjustments to enhance the accuracy and results of subsequent models.
- **Other Techniques**:
  We also employed other techniques such as Max Pooling and Batch Normalization layers in the model.
  - **Max Pooling:**
    This is a down sampling strategy that reduces the spatial dimensions (i.e., width and height) of the input volume. It helps to reduce the computational complexity, memory usage, and number of parameters, thereby reducing overfitting.
  - **Batch Normalization**:
    This is a technique to provide any layer in a neural network with inputs that are zero mean/unit variance. It's used to normalize the input layer by adjusting and scaling the activations. This makes the network faster and more stable.

These techniques collectively contributed to the successful training of our model.

5. Model Training:

The steps and details for developing and training our final model.

- ## Model Development:

Our model utilizes the DenseNet201 model, which is pretrained on the ImageNet dataset, as the initial layers for feature extraction from images. The weights of this model are preserved (i.e., layer. Trainable = False) to retain the knowledge it has already acquired.

Conv2D Layers:

Following this, we have implemented Conv2D layers, which are convolution layers that learn the filters traditionally hand engineered in CNNs. The activation function used here is 'relu', which adds non-linearity to the network and helps to an extent in solving the vanishing gradient problem.

Dropout Layers:

In addition to Conv2D layers, we have also utilized Dropout layers, Dense layers, Batch Normalization, and MaxPooling2D layers. The Dropout layer is a regularization technique used to reduce overfitting in neural networks by preventing complex co-adaptations on training data. It's an efficient way of performing model averaging with neural networks.

MaxPooling2D Layer:

The MaxPooling2D layer reduces the spatial dimensions of the output volume, helping to decrease the computational power required to process data through dimensionality reduction. The Batch Normalization layer normalizes the activations of the previous layer at each batch, applying a transformation that maintains the mean activation close to 0 and the activation standard deviation close to 1. This helps in speeding up the training process and reducing the chance of getting stuck in local optima.

Dense Layers:

The Dense layers are fully connected layers that perform classification on the features extracted by the convolutional layers and down sampled by the pooling layers. They are present towards the end of the network.

Flatten Layer:

The Flatten layer is used when transitioning between convolutional layers and fully connected layers. It flattens the input without affecting the batch size.

Final Dense Layer:

Finally, we have a Dense layer with 7 neurons, using the softmax function for activation. This final layer outputs a probability distribution over the 7 classes. Each

neuron will output a value between 0-1, and the sum of all the output values from the neurons will be 1.

All these layers were identified and utilized after thorough consideration and testing. This forms the final architecture of our model.

- ## Model Training:

For training our model, we first ensured that the training data included 80% of each label. We then used the Densenet201 model's own image preprocessing function to preprocess the images. During this preprocessing phase, we also generated batches of size 16 for our augmented data.

We initially trained our model for 40 epochs. To maintain more control and prevent the model from overfitting, we divided further training into separate batches, each consisting of 10 epochs.

During training, we used the test data as our validation set for the model. This was utilized for Early Stopping. The Early Stopping module monitored the validation loss, and if the loss became stagnant, it would halt the training process.

6. Model Evaluation:

The evaluations are made from training the final model.

- Accuracy Evaluations:

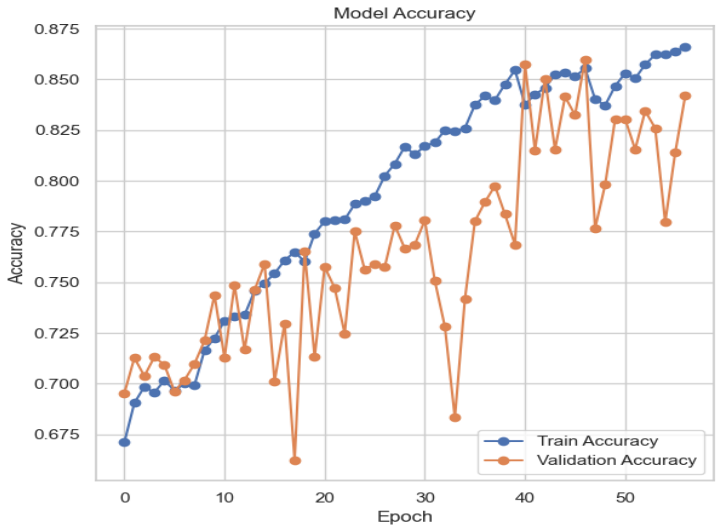

Figure 1 Model Training and Validation Accuracy

During the first training period (Epoch 0 to 39), the validation accuracy started off well, but it began to deteriorate as the epochs progressed. Consequently, we decided to swap the training and testing data again during the second training period (Epochs 40 to 50). This strategy proved beneficial for our model, as we observed an increase in accuracy from 75 to approximately 85. This improvement helped enhance the overall accuracy of our model.

However, during the third training period, we once again swapped the training and test data. This time, our accuracy remained around the same level as before and did not show any significant improvement. Additionally, our Early Stop mechanism abruptly halted the model's training.

In contrast to the validation accuracy, the training accuracy demonstrated a steady increase over the epochs. While there were minor drops in accuracy, these were primarily due to changes in the datasets. Even so, these drops were relatively minor.

The variation in training accuracy and validation accuracy could be attributed to the fact that some labels had more data, so they were learned more easily by the model, while others had less data, so they were not learned as effectively. The augmentation and the larger amount of data being taken by those labels with a higher number could explain the variation between our training and validation accuracy. It's challenging to notice variations in a large overall data set compared to a small one, where minor changes can have significant effects.

- ## Loss Evaluations:

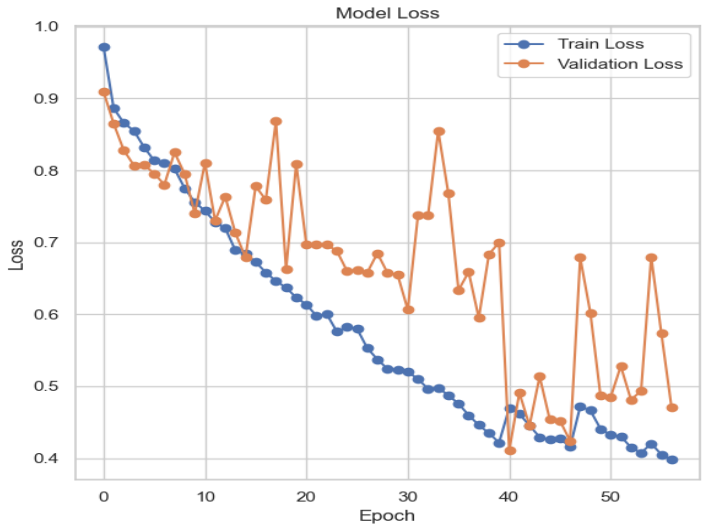

Figure 2 Model Training and Validation Loss

Similar to our validation accuracy, our loss also deteriorated during the first training cycle. However, it started to improve during the second training period and didn't show any significant improvement in the third training period.

Unlike the validation loss, the training loss decreased steadily as the epochs progressed. This could be attributed to the fact that we were augmenting only our training data and not our testing data. This might have resulted in the model learning the features of some labels better than others due to variations in the actual data of each label. Therefore, the optimizer might be working more towards optimizing the model according to the training data.

As discussed earlier, the variation in training accuracy and validation accuracy could be due to the fact that some labels had more data, so they were learned more easily by the model, while others had less data, so they were not learned as effectively. The augmentation and the larger amount of data being taken by those labels with a higher number could explain the variation between our training and validation accuracy. It's challenging to notice variations in a large overall data set compared to a small one, where minor changes can have significant effects.

7. Results:

- ## Model Performance:

We used our original dataset as a benchmark to evaluate the effectiveness of our AI. This allowed us to achieve the highest possible accuracy from all the different models and training methods we utilized.

```
313/313 [==============================] - 59s 169ms/step
313/313 [==============================] - 54s 171ms/step
Accuracy: 0.8843734398402396
```

Figure 3 Model Testing Accuracy

For our original dataset, we were able to achieve an estimated accuracy of 88.4 percent.

- ## Model Visualization:

From our testing we were able to extract the probabilities of how accurately our model was able to predict each of our labels.

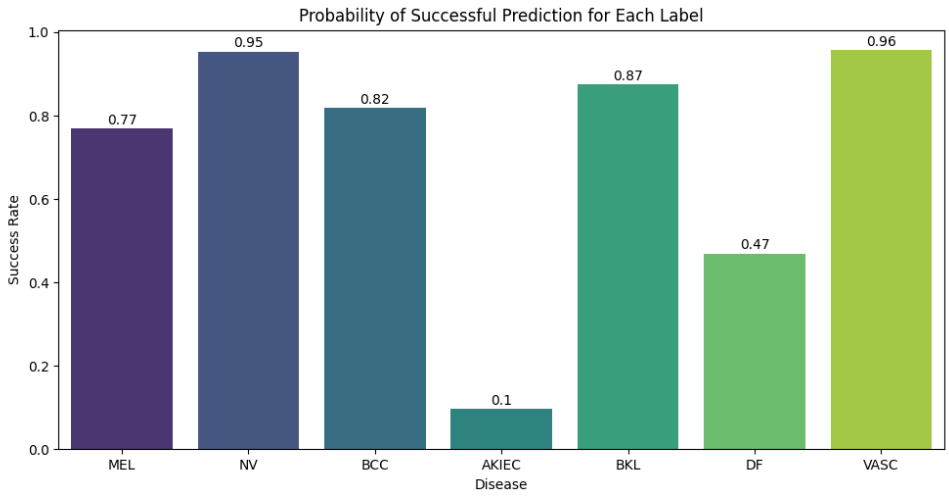

Figure 4 Accuracies of each Labels

Our results indicate that our model performed better on some labels than others. For instance, it achieved the highest accuracy of 0.96 for images labeled as 'VASC'. However, for the same dataset, the model's performance varied significantly for different labels. While the accuracy for 'VASC' was around 0.96, it dropped to around 0.1, or 10%, for 'AIKEC' labeled images. This suggests that the model was able to learn and predict 'VASC' labels more effectively than 'AIKEC' labels from the training images.

However, for the other labels, we were able to achieve an average accuracy of 0.8. This indicates that despite the variations in performance for different labels, the model was generally able to predict the labels accurately for a majority of the images.

- ## Findings:

While we visualized the accuracies for each label, these graphs highlight the actual accuracies of our labels in a more comprehensible format.

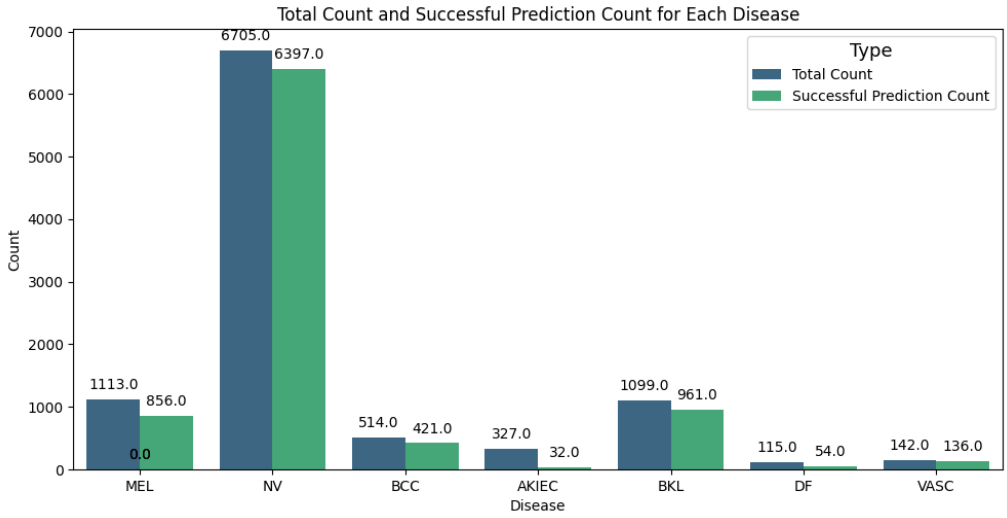

Figure 5 Total, and Accurate Count of labels

This visualization effectively highlights the performance of our model and the data we used for training. The blue bars represent the total number of images for each of our labels, illustrating the variations in our data. The highest count is 6705, while the lowest is 115.

These variations underscore the points we've discussed so far, such as why our loss during training differed from validation. The model accurately predicted about the same number of images for 'AIKEC' and 'DF'. However, our previous prediction accuracy showed a significant variation between these two labels.

This variation is also the reason why we were unable to achieve the same level of accuracy for 'AIKEC' and 'DF', even though we managed to attain higher prediction accuracy for 'MEL', 'NV', 'BCC', and 'VASC'. This suggests that the model's performance can be influenced by the distribution and quantity of data available for each label.

While we were able to achieve a high accuracy, a bit more hyperparameter tuning might have led to an even more accurate model. However, it's certain that even with hyperparameter tuning, we would reach a point where our model would become too complex for the amount of data we were working with. At that point, our option would be to increase the data. Even now, given the variations in our data, if we had a constant amount of data, we might have achieved higher accuracies for each label.

8. Experimentation:

- ## Hyperparameter tuning:

While we were able to find the current model through the process of trial and error, by modifying layer till we find one that worked. For example, using our current model as our benchmark we compare it to some previous generations.

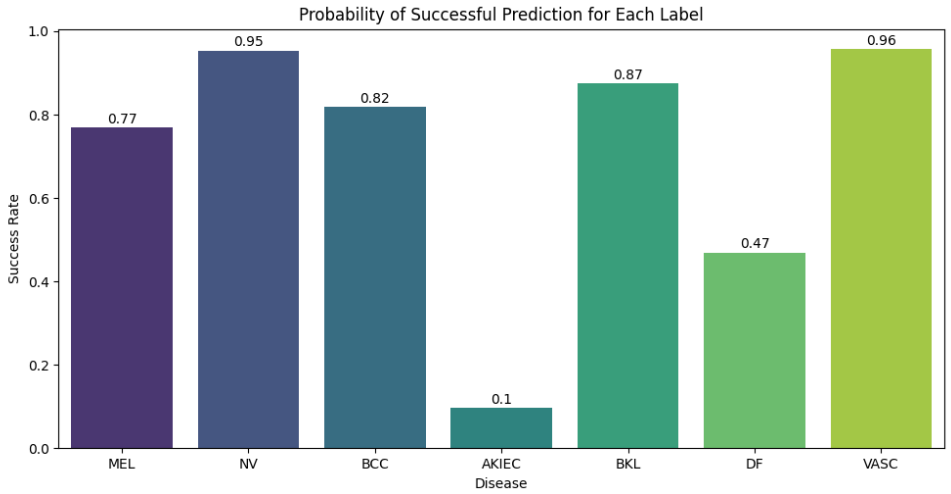

Figure 6 Accuracies of Tuned Model

This is the accuracies of my current model labels

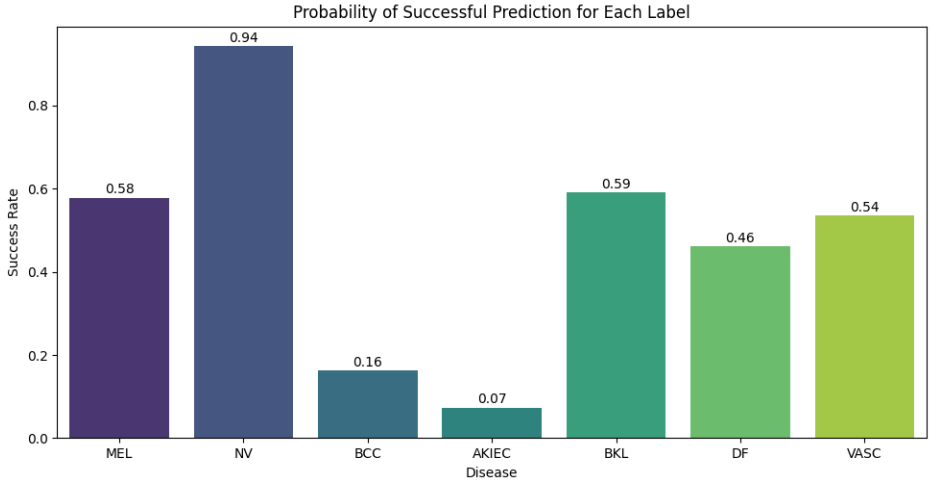

Figure 7 Accuracies of Pre Tuned Model

This refers to the accuracy of the previous generation model, which had fewer Conv2D layers but more filters. Even after 40 epochs, the accuracy of the labels refused to increase. This model achieved a total accuracy of 78.3, while the current one achieved an accuracy of 88.4.

Other models also performed similarly, with accuracies of 76, 74, and 80.3 for even earlier models. These models also had fewer layers but a higher dropout ratio and larger filters.

Compared to my current model these models had more trainable parameters where the average amount of parameter in these models was around 20 million while the current model only has 7 million trainable parameters.

- ## Data Augmentation:

Previously, we mentioned that we used multiple data augmentation techniques. One of these involved increasing all the data to the same amount to address the variations between all the labels. To tackle this issue, we manually used different libraries to modify the images and equalize the number of images for all labels.

However, this approach resulted in worse outcomes than what we were able to achieve with our current model. This was due to the fact that even after augmentation, our images were still close to the original. This only served to cause overfitting in all the models. Even when we were able to address the overfitting issue, we couldn't get the model to a higher accuracy than 80.3. This accuracy was the cut-off point before we had to change our strategy to focus solely on the hyperparameter tuning of our model.

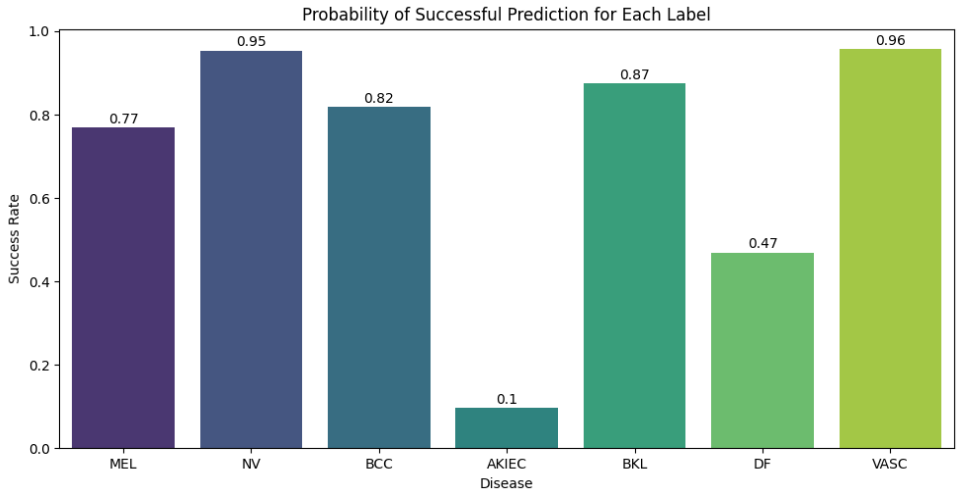

Figure 8 Accuracies of Original Augmented Data

This is the accuracies of my current model labels

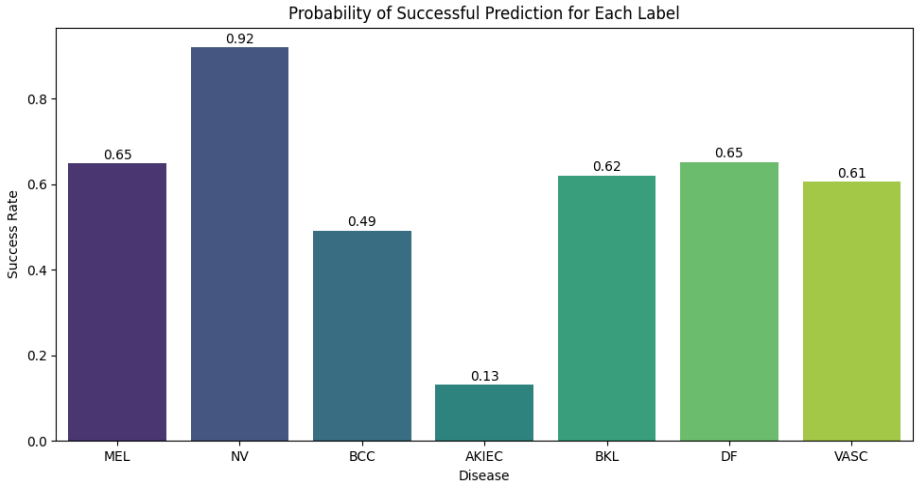

Figure 9 Accuracies of Manual Augmented Data

This is the final model we trained using our manually augmented data. While this model was able to achieve higher accuracy for both 'AIKEC' and 'DF', its overall performance was inferior to our current model, with an accuracy of 80.3.

9. Web Application:

- ## Frontend:

For our AI, we have created a simple webpage where a user can upload a picture of their skin. Our model is then able to predict and display the diagnosis, as well as the probability of how certain the model is that the user has that particular disease.

Figure 10 Frontend Home page

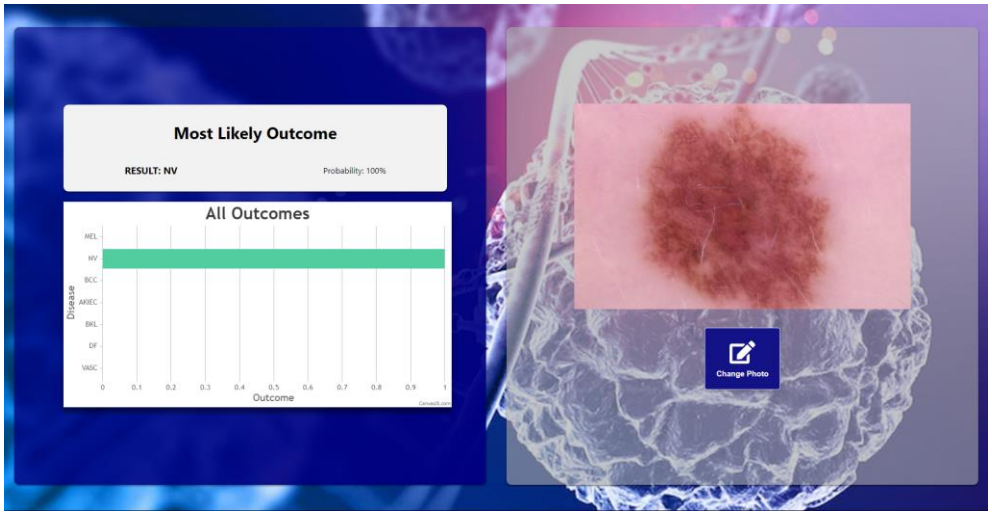

Figure 11 Frontend Result Page

- ## Backend:

For our backend, we opted for a simple approach. We implemented a Flask API where our React application sends a request. The model, loaded on this API, processes the received image, and returns the results to the front end, where these results are displayed to the user. We also opted not to use any database to store these images due to the sensitive nature of our topic.

10. Challenges and Shortcomings:

- ## Challenges:

  Training a skin lesion pigmentation classification model is no simple task, and We've faced numerous challenges throughout the process.

  Firstly, acquiring high-quality and diverse skin lesion images in large quantities is difficult. The images need to be clear and consistent to accurately capture the pigmentation.

  Secondly, we've observed that certain types of lesions are overrepresented in the datasets, while others are underrepresented. This class imbalance can bias the model's learning towards the overrepresented classes.

  Thirdly, there's a significant degree of variability in the diagnosis of skin lesions among expert dermatologists. This variability can infiltrate the training data labels, introducing noise and complicating the model's learning task.

  Lastly, skin lesions can vary greatly in color, size, shape, and texture. Extracting relevant features from these images is a complex task.

- ## Shortcomings:

  Despite these challenges, we've managed to train a model with a maximum accuracy of 88.4%. However, it's not without its shortcomings.

  The model is incorrect about 11.6% of the time, which could potentially lead to incorrect diagnoses. Moreover, it's often unclear why the model made a particular prediction, which is problematic in a medical context where understanding the reasoning behind a diagnosis is crucial.

  The model might not perform well on different populations if the training data is not diverse enough. There's also a risk that the model could be overfitting to the training data, meaning it might not perform as well on unseen data.

  The 'black box' nature of deep learning models is a major issue. It's often unclear why the model made a particular prediction. This lack of interpretability can make it difficult for doctors and patients to trust the model's predictions.

  If the training data is not diverse enough, the model might not perform well when presented with data from different populations. For example, if the model was trained mostly on fair-skinned individuals, it might not perform as well on darker skin tones.

  There's a risk that the model could be overfitting to the training data. This means that while the model performs well on the training data, it might not perform as well on unseen data.

The model might not be robust to variations in image quality, lighting conditions, or the way the image was taken. For instance, an image taken with poor lighting or at an unusual angle might lead to incorrect predictions.

Deep learning models can be computationally intensive to train and require significant amounts of memory. This can be a limitation in resource-constrained settings.

Lastly, when dealing with medical data, there are stringent requirements for data privacy and security. Ensuring that the model complies with all relevant regulations can be a complex task.

These shortcomings underscore the need for ongoing research and development in this field. It's important to continuously evaluate and improve the model to ensure it is safe, effective, and equitable.

These challenges and shortcomings highlight the complexity of this task and underscore the need for careful model design, rigorous validation, and continuous model improvement.

11. Conclusion:

In conclusion, this project has demonstrated the potential of machine learning and artificial intelligence in revolutionizing the field of dermatology, particularly in the diagnosis of skin lesions. The multi-class classification model developed was able to achieve an impressive accuracy of 88.4% on the original dataset, indicating its capability to accurately detect and classify different types of skin lesions based on their pigmentation.

However, the project also highlighted several challenges and limitations inherent in the application of AI in medical imaging. These include the difficulty in acquiring high-quality and diverse skin lesion images, the issue of class imbalance in the datasets, the variability in the diagnosis of skin lesions among expert dermatologists, and the complexity of extracting relevant features from these images.

The model's performance varied significantly for different labels, suggesting that its learning and prediction capabilities can be influenced by the distribution and quantity of data available for each label. Despite achieving high accuracy for some labels, the model was less accurate for others, indicating areas for improvement.

The project also underscored the importance of hyperparameter tuning in enhancing the model's accuracy. However, it also highlighted the risk of overfitting if the model becomes too complex for the amount of data available, suggesting the need for a balance between model complexity and data availability.

The 'black box' nature of deep learning models, the potential for performance variation across different populations, and the stringent requirements for data privacy and security in dealing with medical data were also identified as significant challenges.

Despite these challenges and limitations, the project represents a significant step forward in the application of AI in dermatology. It underscores the need for ongoing research and development, careful model design, rigorous validation, and continuous model improvement to ensure the safe, effective, and equitable use of AI in medical imaging and diagnostics.

The project's success in achieving high accuracy in skin lesion diagnosis using AI provides a promising foundation for future research and development in this field. It paves the way for more advanced applications of AI in medical imaging and diagnostics, with the potential to contribute significantly to improved healthcare outcomes. However, it also serves as a reminder of the complexities and challenges involved in this endeavor, emphasizing the need for careful consideration and handling of these issues in future work.