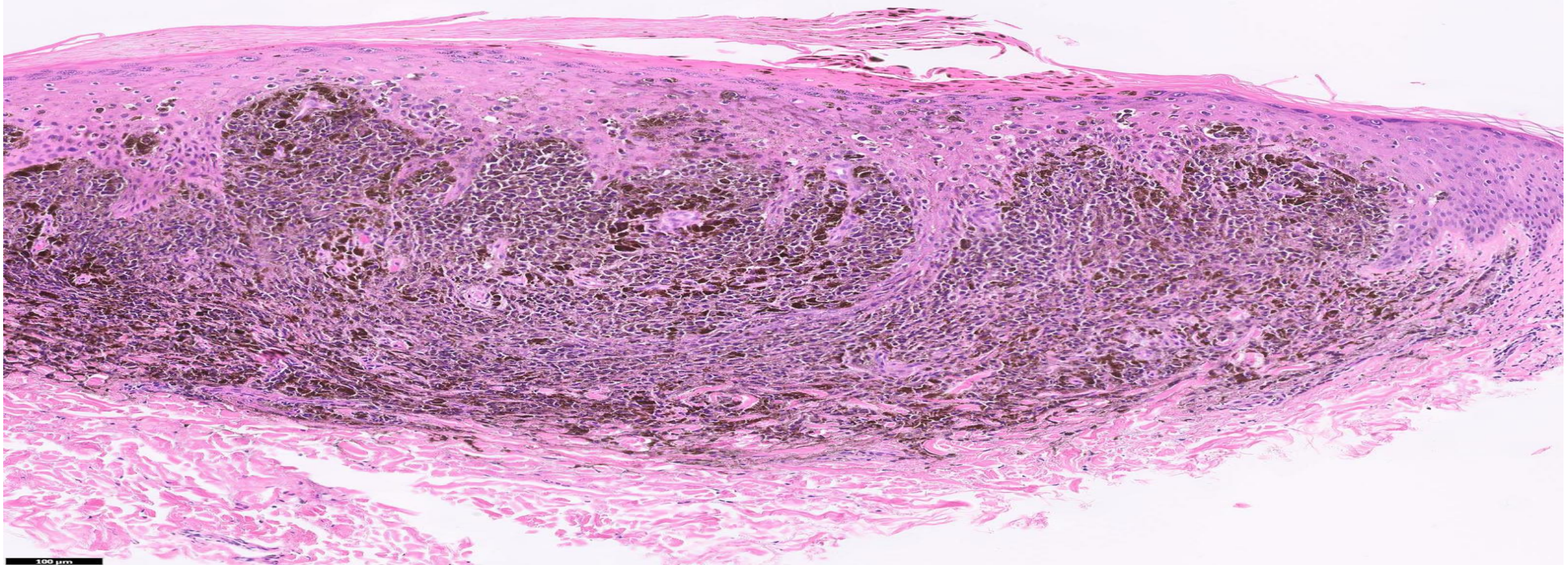


MELANOMA DETECTION



SUBMITTED BY :

PRINCE FRANCIS

DIBYA RANJAN MOHANTY

PRATIMA PANDEY

PROBLEM STATEMENT

- To build a CNN based model which can accurately detect melanoma. Melanoma is a type of cancer that can be deadly if not detected early. It accounts for 75% of skin cancer deaths. A solution which can evaluate images and alert the dermatologists about the presence of melanoma has the potential to reduce a lot of manual effort needed in diagnosis.

TABLE OF CONTENTS

- GENERAL INFORMATION
- PROJECT DESCRIPTION
- CONCLUSIONS
- LIBRARIES USED

GENERAL INFORMATION

- Melanoma is the most dangerous type of skin cancer. Globally, in 2012, it newly occurred in 232,000 people. In 2015, 3.1 million people had active disease, which resulted in 59,800 deaths. Australia and New Zealand have the highest rates of melanoma in the world. High rates also occur in Northern Europe and North America, while it is less common in Asia, Africa, and Latin America. In the United States, melanoma occurs about 1.6 times more often in men than women. Melanoma has become more common since the 1960s in areas mostly populated by people of European descent.
- For those in whom melanoma has spread, immunotherapy, biological therapy, radiation therapy, or chemotherapy may improve survival. With treatment, the five-year survival rates in the United States are 99% among those with localized disease, 65% when the disease has spread to lymph nodes, and 25% among those with distant spread.
- The dataset consists of 2357 images of malignant and benign oncological diseases, which were formed from the **International Skin Imaging Collaboration (ISIC)**. All images were sorted according to the classification taken with ISIC

PROJECT DESCRIPTION

1. Data Folder Hierarchy

- Skin cancer ISIC The International Skin Imaging Collaboration

- Train

- actinic keratosis
- basal cell carcinoma
- Dermatofibroma
- Melanoma
- Nevus
- pigmented benign keratosis
- seborrheic keratosis
- squamous cell carcinoma
- vascular lesion

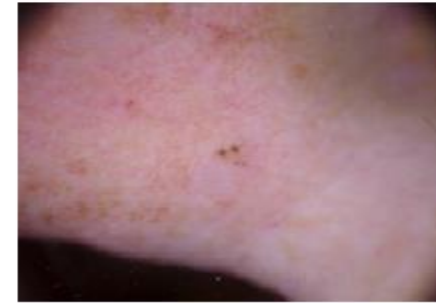
- Test

- actinic keratosis
- basal cell carcinoma
- Dermatofibroma
- Melanoma
- Nevus
- pigmented benign keratosis
- seborrheic keratosis
- squamous cell carcinoma
- vascular lesion

actinic keratosis



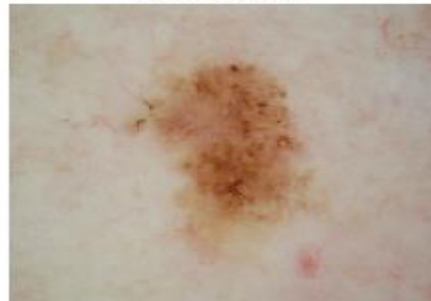
basal cell carcinoma



dermatofibroma



melanoma



nevus



pigmented benign keratosis



seborrheic keratosis



squamous cell carcinoma



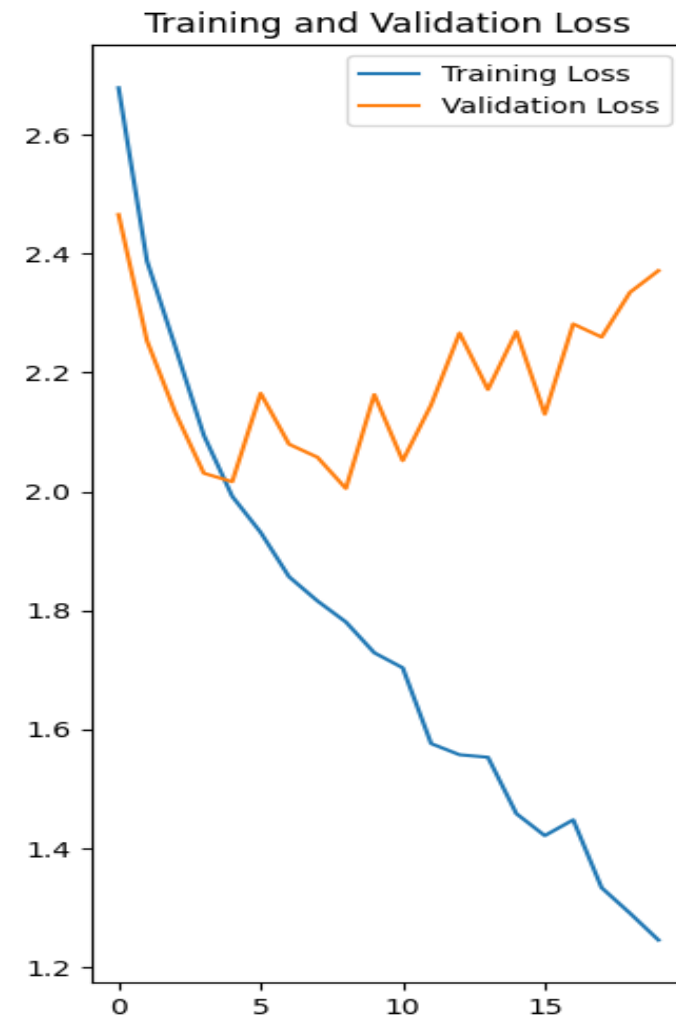
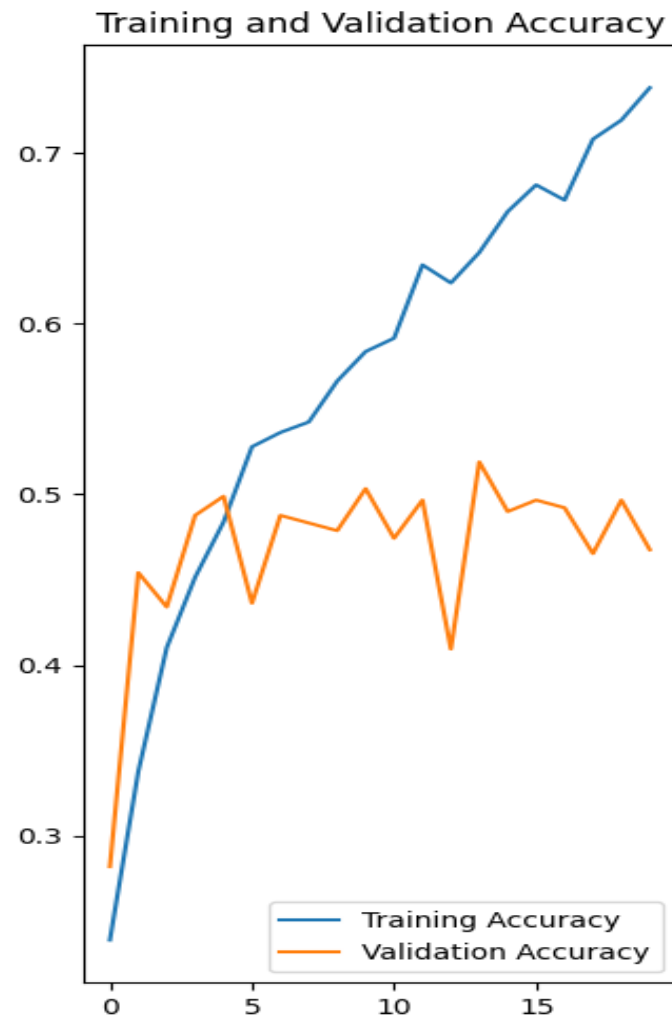
vascular lesion



2. Building a simple CNN

Model: "sequential"

Layer (type)	Output Shape	Param #
rescaling (Rescaling)	(None, 180, 180, 3)	0
conv2d (Conv2D)	(None, 180, 180, 16)	448
max_pooling2d (MaxPooling2D)	(None, 90, 90, 16)	0
conv2d_1 (Conv2D)	(None, 90, 90, 32)	4640
max_pooling2d_1 (MaxPooling2D)	(None, 45, 45, 32)	0
conv2d_2 (Conv2D)	(None, 45, 45, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 22, 22, 64)	0
dropout (Dropout)	(None, 22, 22, 64)	0
flatten (Flatten)	(None, 30976)	0
...		
Total params: 3988640 (15.22 MB)		
Trainable params: 3988640 (15.22 MB)		
Non-trainable params: 0 (0.00 Byte)		



- Clearly the CNN is overfitting as per the observation.

3. Data Augmentation to reduce overfitting

- Flip → Image is flipped horizontally



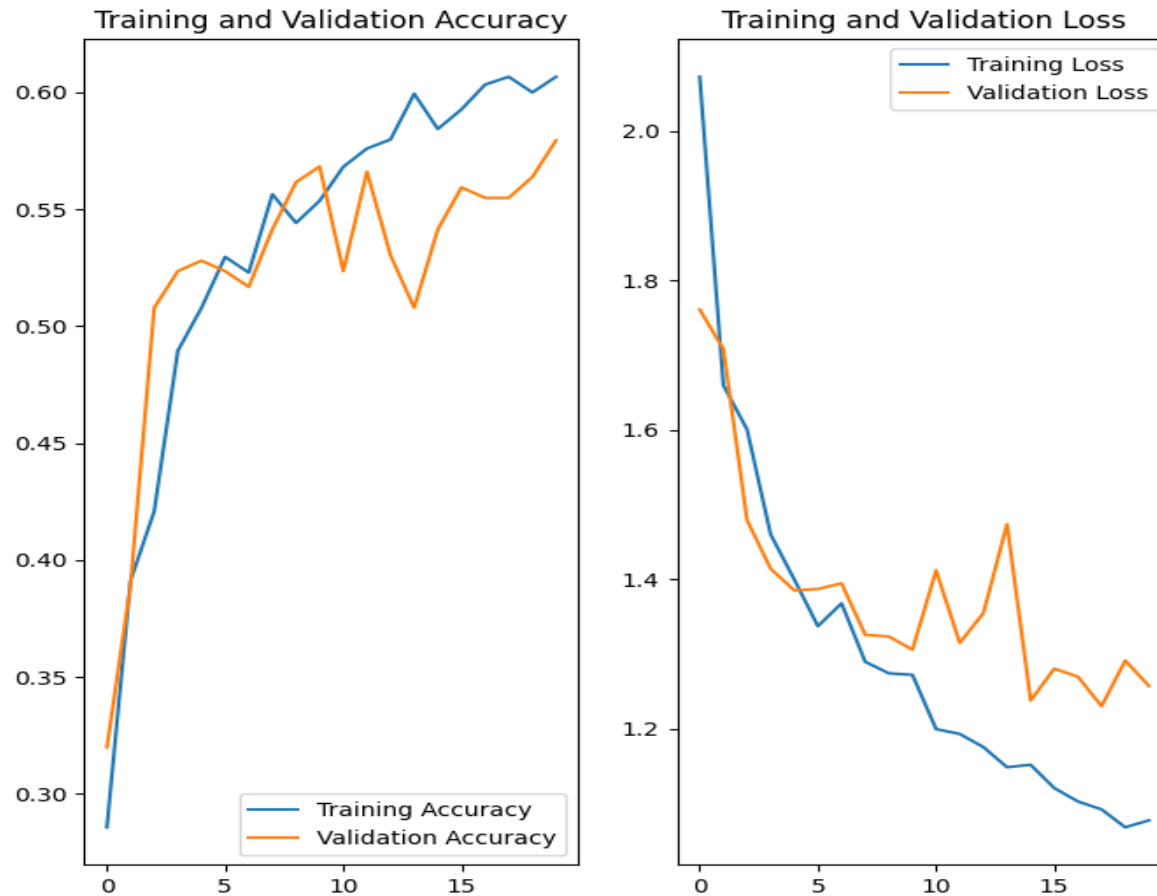
- Rotation → Image is rotated



- Zoom → Image is zoomed



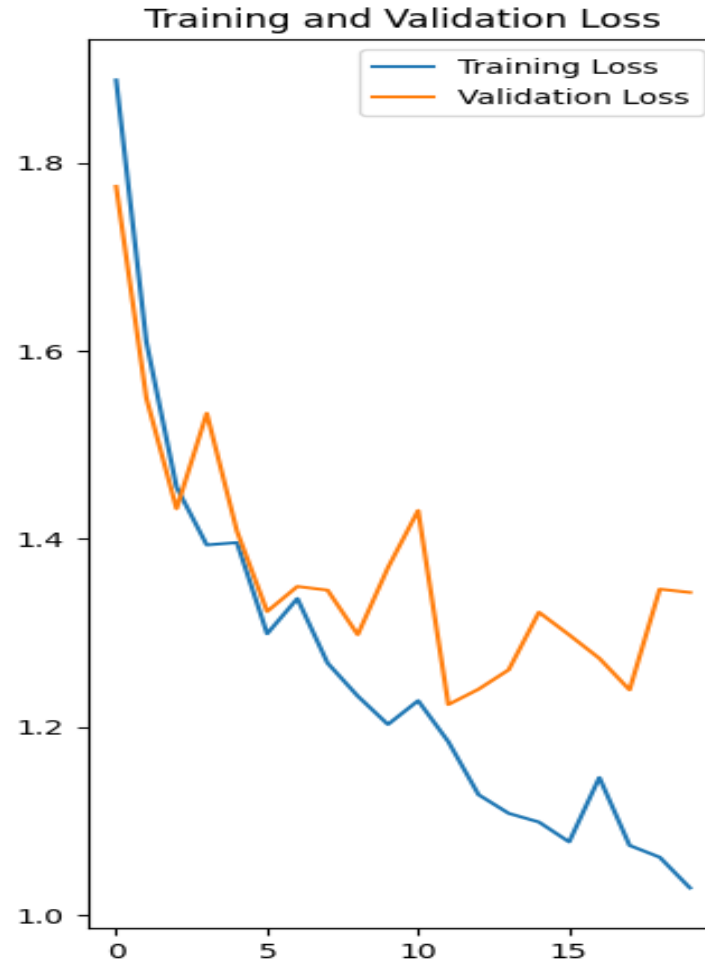
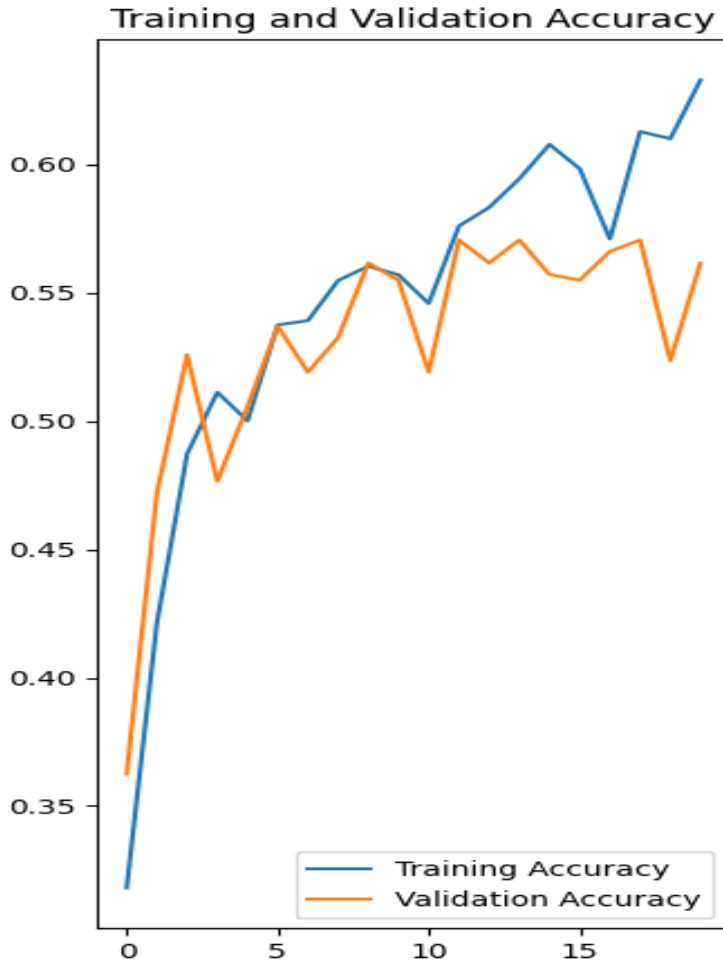
Data Augmentation(Contd..)



Inference

- We can clearly see that the model overfitting has reduced significantly since the training and validation accuracy is very similar.
- This simply means the model is not mugging up all the information.

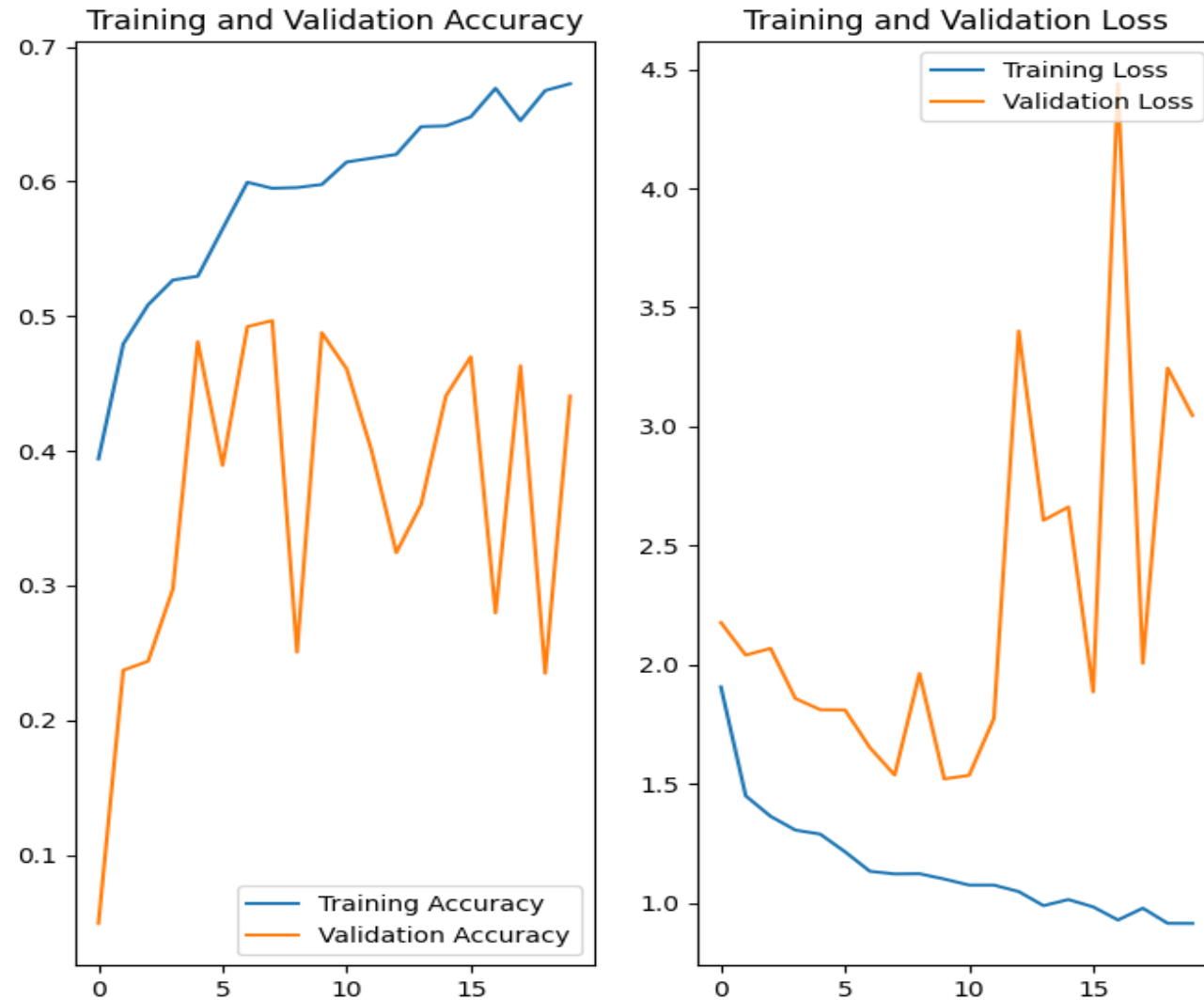
4. Adding Dropout layer



Inference

- Adding a Dropout layer and making 25% percent of the weights to reduce overfitting.
- Dropout is a regularization technique.
- Clearly the training and validation accuracy has increased slightly and we have also resolved overfitting.

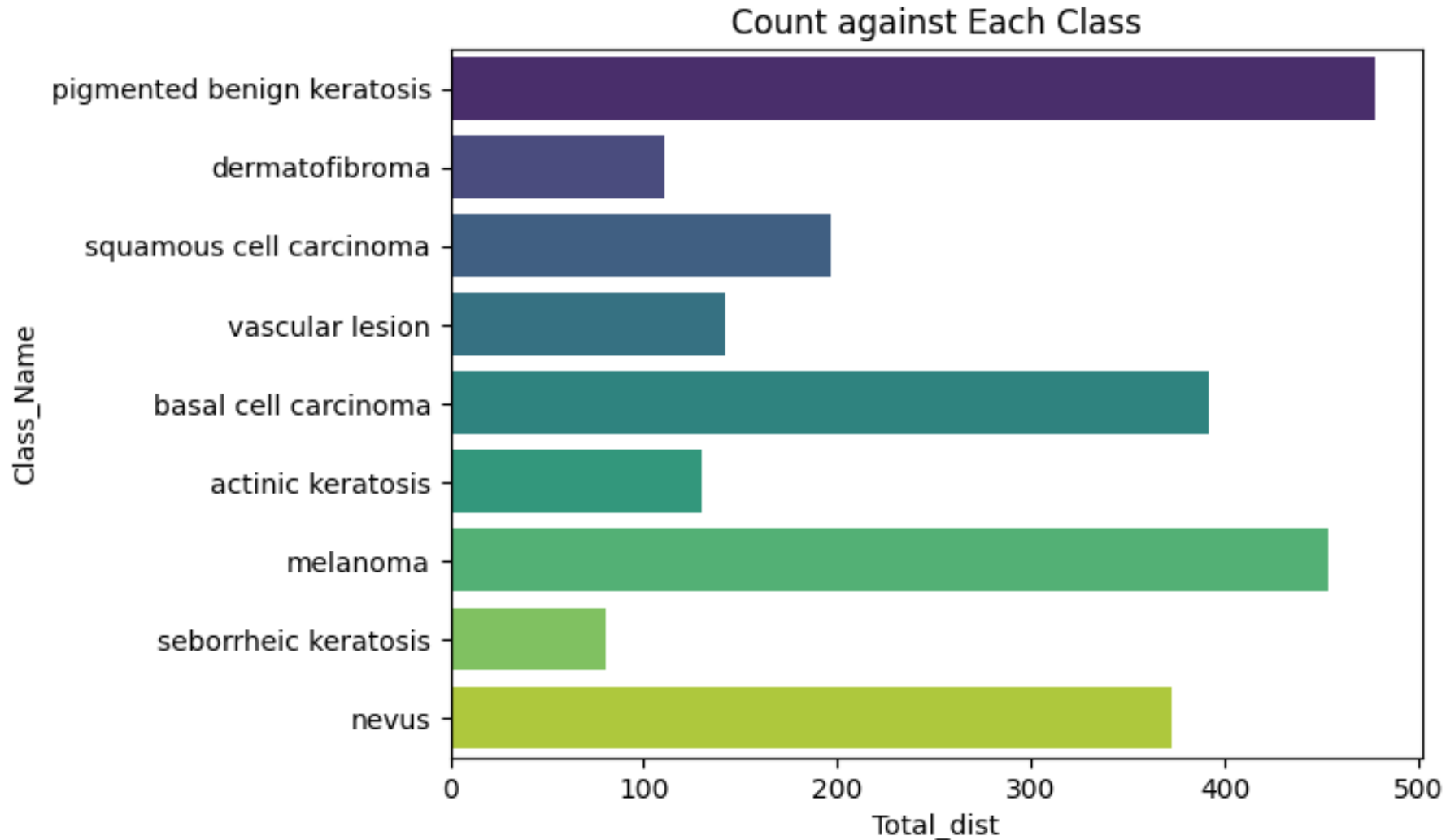
5. Adding Batch Normalization



Inference

- Adding a Batch Normalization on top of the augmented and Dropout layer has not improved the model in any sense.
- There is significant overfitting introduced in the model.
- Model is not able to generalize on unseen data.

6. Class Imbalance



Handling Class Imbalance(Class imbalance contd...)

- We used the Augmentor library to make 500 samples each each of the classes.
- Each of the 500 samples are stored in an output folder in each of the subdirectories.

7.Final Model

Model: "sequential_11"

Layer (type)	Output Shape	Param #
sequential_1 (Sequential)	(None, 180, 180, 3)	0
rescaling (Rescaling)	(None, 180, 180, 3)	0
conv2d_30 (Conv2D)	(None, 180, 180, 16)	448
max_pooling2d_30 (MaxPooling2D)	(None, 90, 90, 16)	0
conv2d_31 (Conv2D)	(None, 90, 90, 32)	4640
max_pooling2d_31 (MaxPooling2D)	(None, 45, 45, 32)	0
conv2d_32 (Conv2D)	(None, 45, 45, 64)	18496
max_pooling2d_32 (MaxPooling2D)	(None, 22, 22, 64)	0
dropout_10 (Dropout)	(None, 22, 22, 64)	0

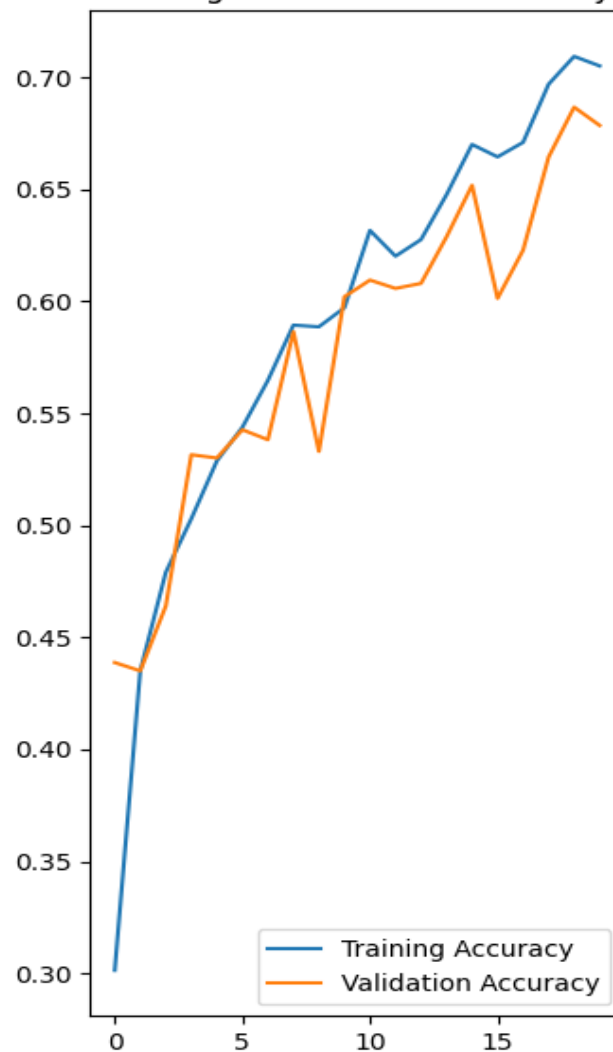
...

Total params: 3989801 (15.22 MB)

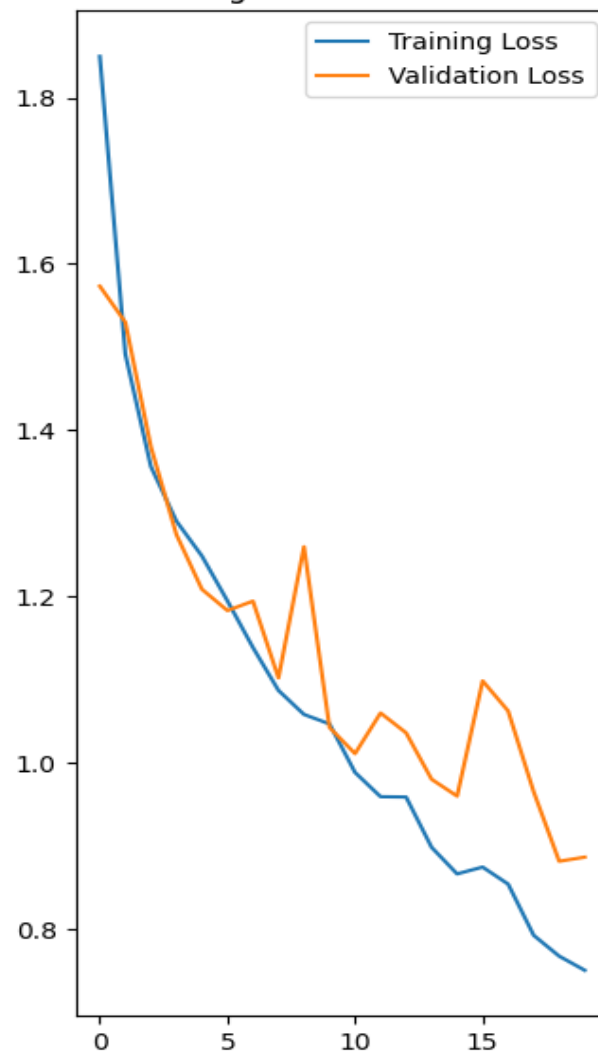
Trainable params: 3989801 (15.22 MB)

Non-trainable params: 0 (0.00 Byte)

Training and Validation Accuracy



Training and Validation Loss



8. Conclusion

- Trained Model Output

- Train Accuracy- 70.5%
- Validation Accuracy – 67.85%
- Train Loss- 0.77
- Validation Loss – 0.84

- Inference

- Comparing the test and validation accuracies we can say that our initial problem of overfitting has been reduced.
- Data Augmentation, Dropout, and handling class imbalance have significantly helped to improve model accuracy and reduced overfitting.

Conclusion (Contd....)

- Model evaluation on test set
 - Loss on test set : 0.83
 - Accuracy on test set : 63.8%
- Inference
 - Model is performing significantly well on the test set since our validation accuracy was 67.85%.
 - Our model is performing well on unseen data well.

```
56/56 [=====] - 0s 9ms/step - loss: 0.8313 - accuracy: 0.6830  
Loss on the test set:0.83  
Acc on the test set:68.3%
```

- Model prediction on a random subset of class

```
1/1 [=====] - 0s 17ms/step  
Predicted Class:vascular lesion  
Actual Class:vascular lesion  
This image most likely belongs to vascular lesion with a 100.00 percent confidence.
```

LIBRARIES USED

Below mentioned are all the libraries and modules used in the project

```
● ## Importing important libraries
  import pathlib
  import matplotlib.pyplot as plt
  import numpy as np
  import pandas as pd
  import seaborn as sns
  import requests
  import zipfile
  import os

  import tensorflow as tf
  import PIL
  from PIL import Image
  import keras
  # import Augmentor
  from tensorflow import keras
  from tensorflow.keras import layers
  from tensorflow.keras.models import Sequential
  from keras.layers import Conv2D,MaxPool2D,Dropout,Activation,Flatten,Dense,Rescaling,BatchNormalization
  import tensorflow.keras.preprocessing.image as load_img
  from keras.utils import image_dataset_from_directory
  from keras.layers import RandomFlip,RandomRotation,RandomZoom
```

Other mentions: Notebook has been trained on Google Collab for accelerated computation using GPU.