Convolutional Neural Networks

- **Data Reading/Data Understanding** → Defining the path for train and test images

  Complete

- **Dataset Creation**→ Create train & validation dataset from the train directory with a batch size of 32. Also, make sure you resize your images to 180*180.

  Complete

- **Dataset visualisation** → Create a code to visualize one instance of all the nine classes present in the dataset

  Complete

- **Model Building & training :**
  - Create a CNN model, which can accurately detect 9 classes present in the dataset. While building the model, rescale images to normalize pixel values between (0,1).

    Complete

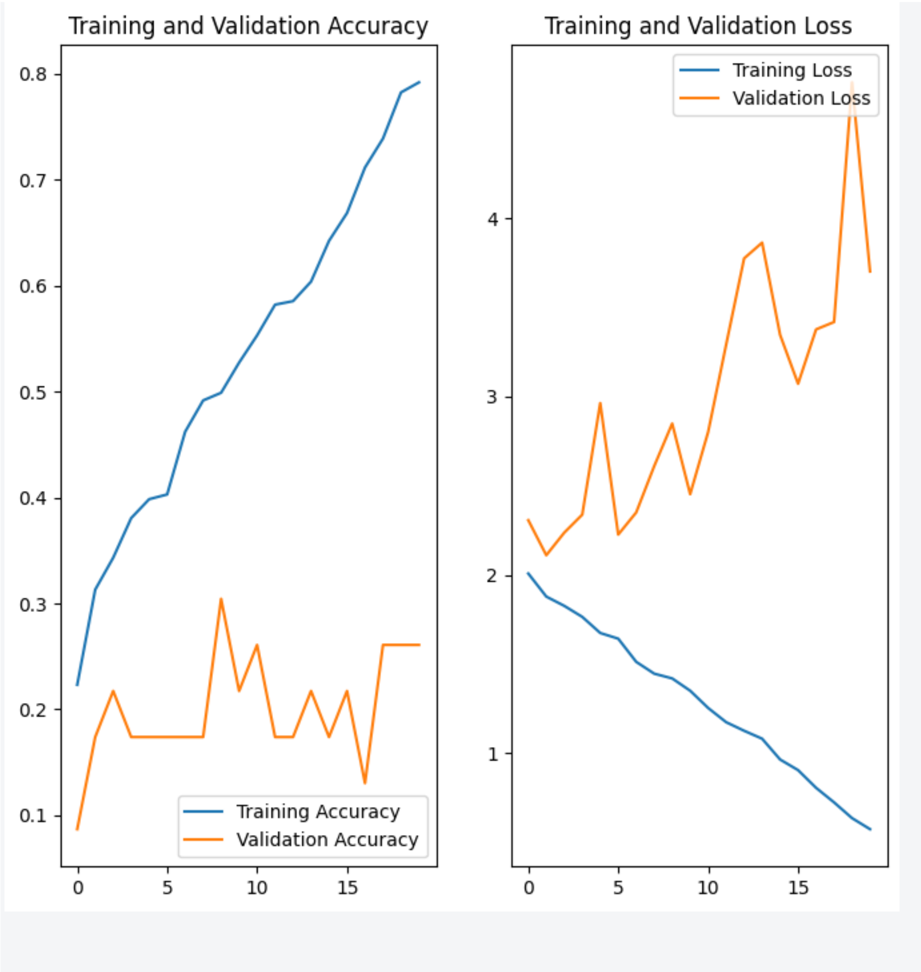
  - Choose an appropriate optimiser and loss function for model training

    Complete

  - Train the model for ~20 epochs

    Complete

  - Write your findings after the model fit. You must check if there is any evidence of model overfit or underfit.



Here's how to interpret these graphs:

Training And Validation Accuracy: Ideal Scenario:

Training Accuracy: Initially, both training and validation accuracies increase as the model learns and improves on the data. Validation Accuracy: Continues to increase along with the training accuracy, showing that the model generalizes well to unseen data. Overfitting:

Training Accuracy: Continues to increase or stays high as the model learns complex patterns within the training data, even noise or irrelevant patterns. Validation Accuracy: Initially increases but starts decreasing or remains stagnant after a point, showing that the model doesn't generalize well to unseen data. Underfitting:

Training Accuracy: Stays low, indicating that the model fails to capture the underlying patterns of the data. Validation Accuracy: Remains low and doesn't improve significantly, showing that the model doesn't generalize well to both the training and validation data.

So, from the above analysis, the graph definitely shows overfitting. The Training accuracy increases along with the Validation accuracy initially. The Validation accuracy tends to remain low while the training accuracy tends to increase along.

Here's how to interpret training loss and validation loss graphs:

Ideal Scenario:

Training Loss: Initially high and gradually decreases as the model learns from the data. Validation Loss: Decreases in tandem with training loss, indicating the model generalizes well to unseen data. Overfitting:

Training Loss: Continues to decrease or remains very low, showing that the model fits the training data very well. Validation Loss: Starts increasing after reaching a minimum, diverging from the training loss, indicating the model fails to generalize to new, unseen data. Underfitting:

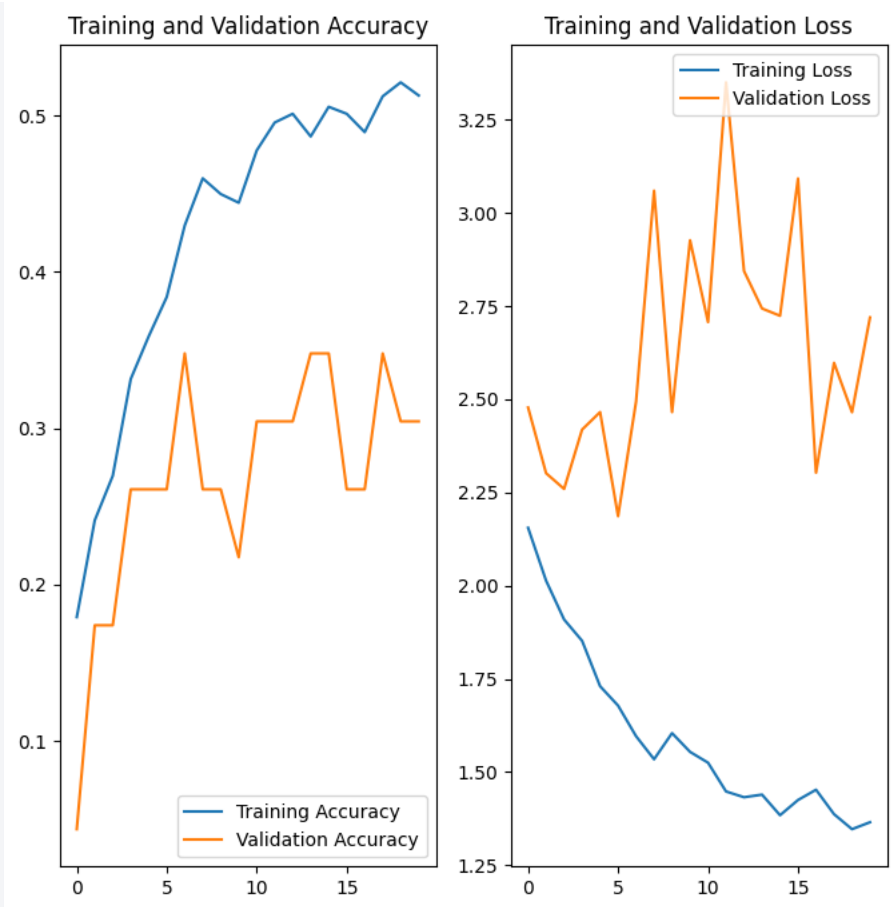
Training Loss: Stays high or decreases very slowly, showing the model fails to capture the patterns in the training data. Validation Loss: Remains high or decreases slightly, indicating the model doesn't generalize well to both the training and validation data.

So, from the graphs, it shows that there is a huge gap between the training loss and validation loss which shows overfitting failing to generalize new or unseen data.

- **Chose an appropriate data augmentation strategy to resolve underfitting/overfitting**

  Complete

- **Model Building & training on the augmented data :**
  - Create a CNN model, which can accurately detect 9 classes present in the dataset. While building the model rescale images to normalize pixel values between (0,1).
  - Choose an appropriate optimiser and loss function for model training
  - Train the model for ~20 epochs
  - Write your findings after the model fit, see if the earlier issue is resolved or not?



We see that there is a decrease in overfitting compared to the previous model. The accuracy of the model has also increased. There is definitely overfitting because of the lack of data in the various categories.

- **Class distribution:** Examine the current class distribution in the training dataset
  - - Which class has the least number of samples?

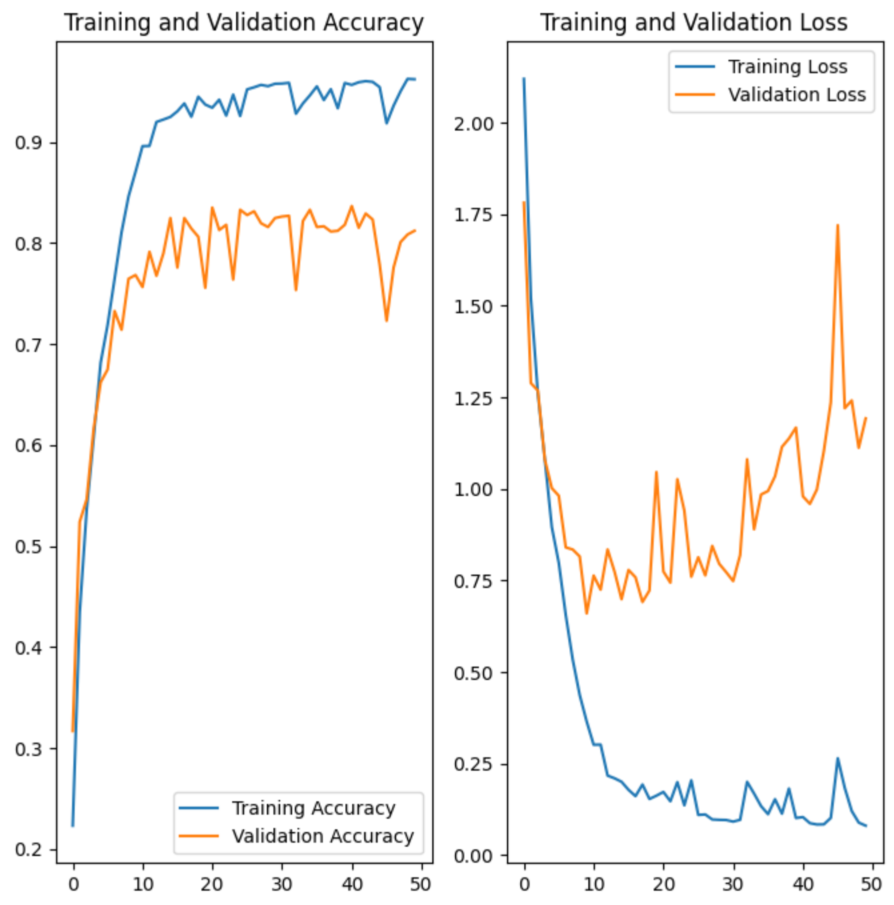
```
Class: melanoma, Number of Images: 352
Class: nevus, Number of Images: 277
Class: seborrheic keratosis, Number of Images: 58
Class: basal cell carcinoma, Number of Images: 309
Class: pigmented benign keratosis, Number of Images: 370
Class: squamous cell carcinoma, Number of Images: 142
Class: dermatofibroma, Number of Images: 77
Class: actinic keratosis, Number of Images: 92
Class: vascular lesion, Number of Images: 115
```

Seborrheic Keratosis has the least number of samples.

  - - Which classes dominate the data in terms of the proportionate number of samples?

Pigmented benign keratosis has the highest number of samples at 370.

- **Handling class imbalances:** Rectify class imbalances present in the training dataset with Augmentor library.
- **Model Building & training on the rectified class imbalance data :**
    - Create a CNN model, which can accurately detect 9 classes present in the dataset. While building the model, rescale images to normalize pixel values between (0,1).
    - Choose an appropriate optimiser and loss function for model training
    - Train the model for ~30 epochs
    - Write your findings after the model fit, see if the issues are resolved or not?




```
We can clearly see the adding Augmentation, drop outs and decreasing
the class imbalance has vastly helped in increasing the accuracy of the
model and decrease overfitting.
```

```
If we see both the graphs, we see that both training and validation
accuracy increases for close to 20 epochs and then the model overfits.
We can also see the same for training and validation loss.
```