Multi-class classification model using a custom CNN based model which can accurately detect melanoma. Melanoma is a type of cancer that can be deadly if not detected early

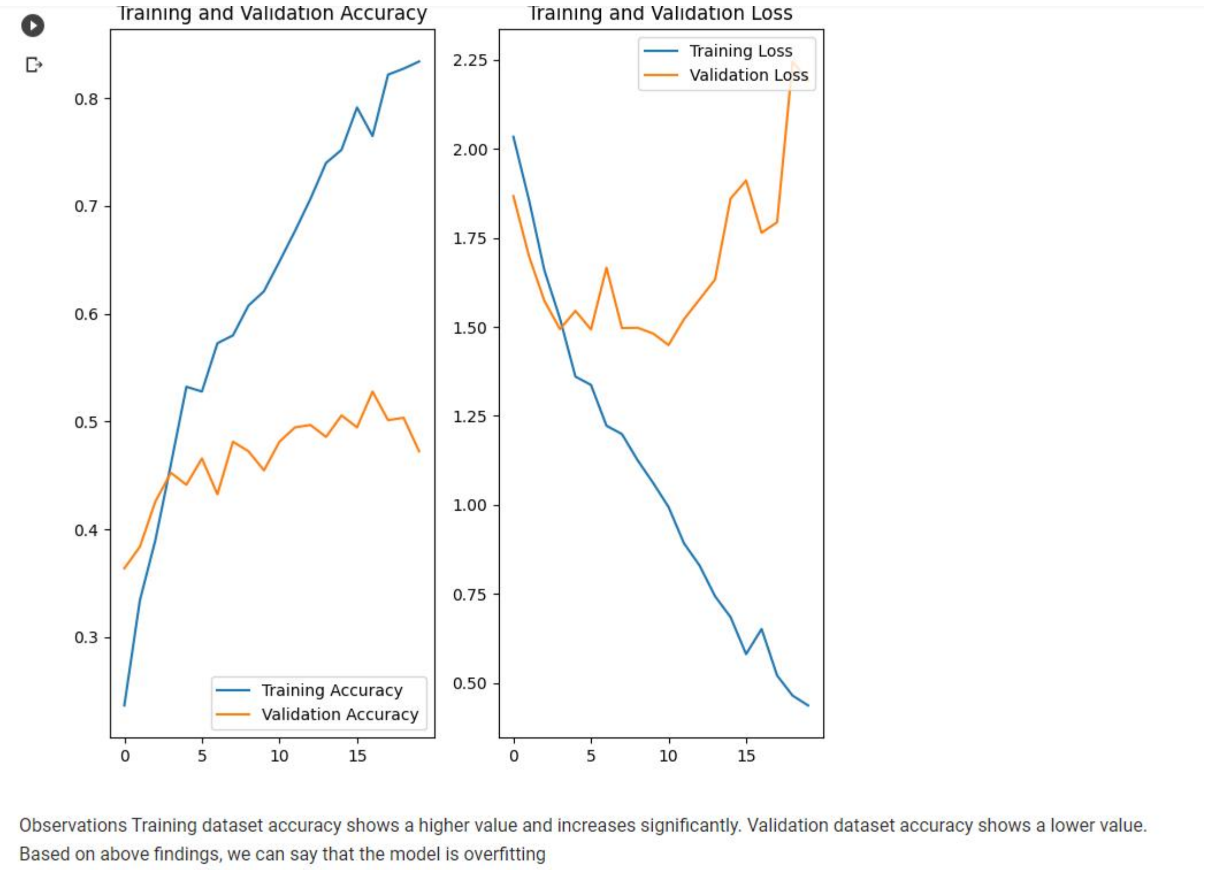
General Information

- Melanoma is a type of cancer that can be deadly if not detected early. It accounts for 75% of skin cancer deaths. A solution that can evaluate images and alert dermatologists about the presence of melanoma has the potential to reduce a lot of manual effort needed in diagnosis.
- Build a multiclass classification model using a custom convolutional neural network in TensorFlow
- Dataset: 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, and all subsets were divided into the same number of images, with the exception of melanomas and moles, whose images are slightly dominant.
- The data set contains the following 9 diseases:
    - Actinic keratosis
    - Basal cell carcinoma
    - Dermatofibroma
    - Melanoma
    - Nevus
    - Seborrheic keratosis
    - Squamous cell carcinoma
    - Pigmented benign keratosis
    - Vascular lesion

Conclusions

- We tried to build three models during the development process
- First model is a overfitting model where the accuracy & loss difference for training and validation data is significantly large

    - ✓ We can observe that,
    - ✓ Training Loss : ~0.44 Validation Loss : 2.19 Training Accuracy : ~0.83 Validation Accuracy : 0.47
    - ✓ We can see that there is a significant increase in accuracy for Training data and accuracy for validation data is very less 47%. This shows Overfitting where the Training accuracy is high 83% and validation data has accuracy of 47%.

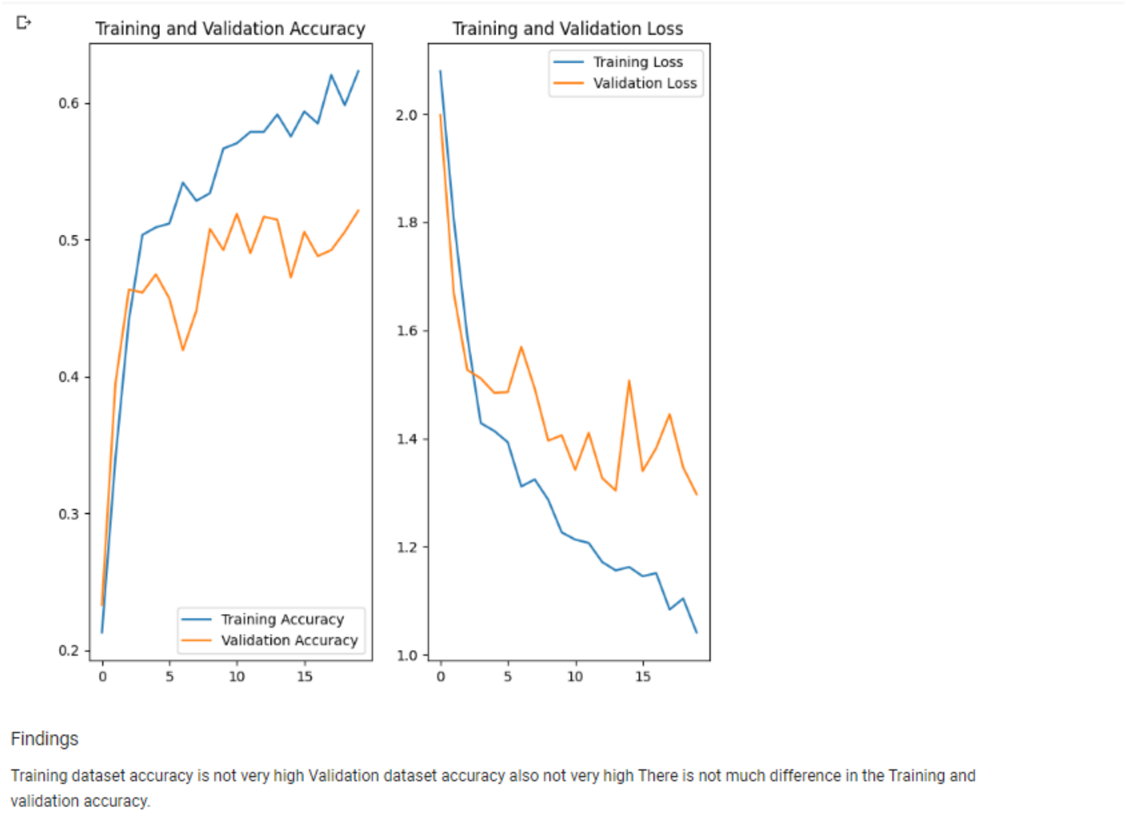


Observations Training dataset accuracy shows a higher value and increases significantly. Validation dataset accuracy shows a lower value. Based on above findings, we can say that the model is overfitting

- Second model is build to handle the overfitting by introducing agumentation, but the model accuracy is low in comparison due to class imbalance.

  ✓ We can observe that,

  ✓ Training Loss : ~1.04 Validation Loss : 1.3 Training Accuracy : ~63% Validation Accuracy : ~52%

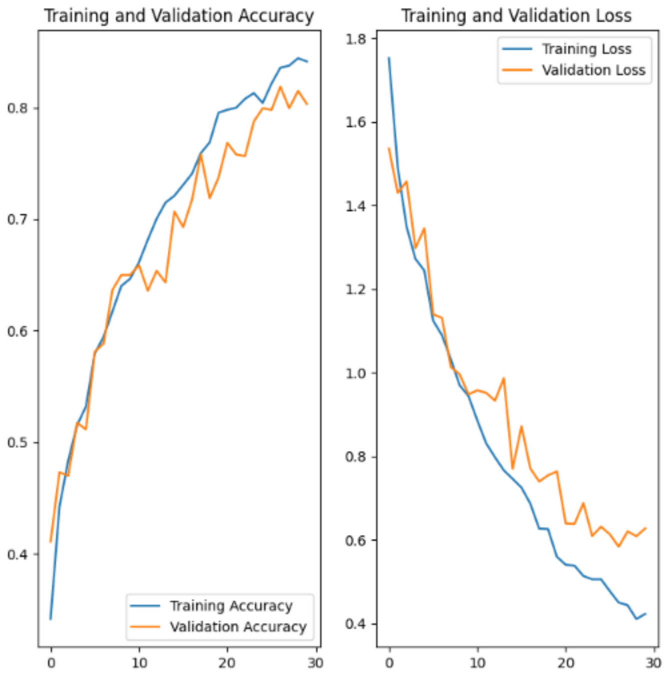


Findings

Training dataset accuracy is not very high Validation dataset accuracy also not very high There is not much difference in the Training and validation accuracy.

-
- Third model is build to handle class imbalance and Overfitting and we observed improvement in accuracy (training accuracy is '84%' & validation accuracy is '80.31%')

    ✓ Training Accuracy - 84% Validation accuracy - 80.31% Training and Validation accuracy have increased significantly and there is not much difference between Training and Validation accuracy, which shows that model is not overfitting and not underfitting also.



Todo: Analyze your results here. Did you get rid of underfitting/overfitting? Did class rebalance help?

We can observe that the training and validation accuracy are following the same trend in both Accuracy & Loss

final model accuracy interms of training accuracy is 84% & validation accuracy is 80.31%

Based on the accuracy numbers, model does not show overfitting as there is not much difference between training and validation dataset. Also model doesnot show underfitting as the accuracy are quite high in both training and validation dataset. Class rebalance using augmentation have definiting helped to achieve better accuracy results for the model.