Skin Cancer Detection using Multiclassification CNN Model

Table of Contents

Abstract

The purpose of the project is to build a CNN based model which can accurately detect type of skin cancer.

There are over 200 different forms of cancer. Out of them 9 types of skin cancers are predominant. Those 9 types of skin cancers data set is taken to build a CNN Model. For example, 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.

Background & Business Problem

The current process of examining a skin lesion as part of skin biopsy takes almost a week or more.

The aim of this project is to reduce the current gap to just a couple of days by providing an accurate predictive model.

The method we used here is Convolutional Neural Network (CNN) to classify nine types of skin cancer from skin lesions images.

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.

Technologies Used

- Tensorflow - version 2.12.0
- CUDA Tool Kit - version 12.1.0
- Augmentor Pipeline - version 0.2.12
- Pandas - version 2.0.1

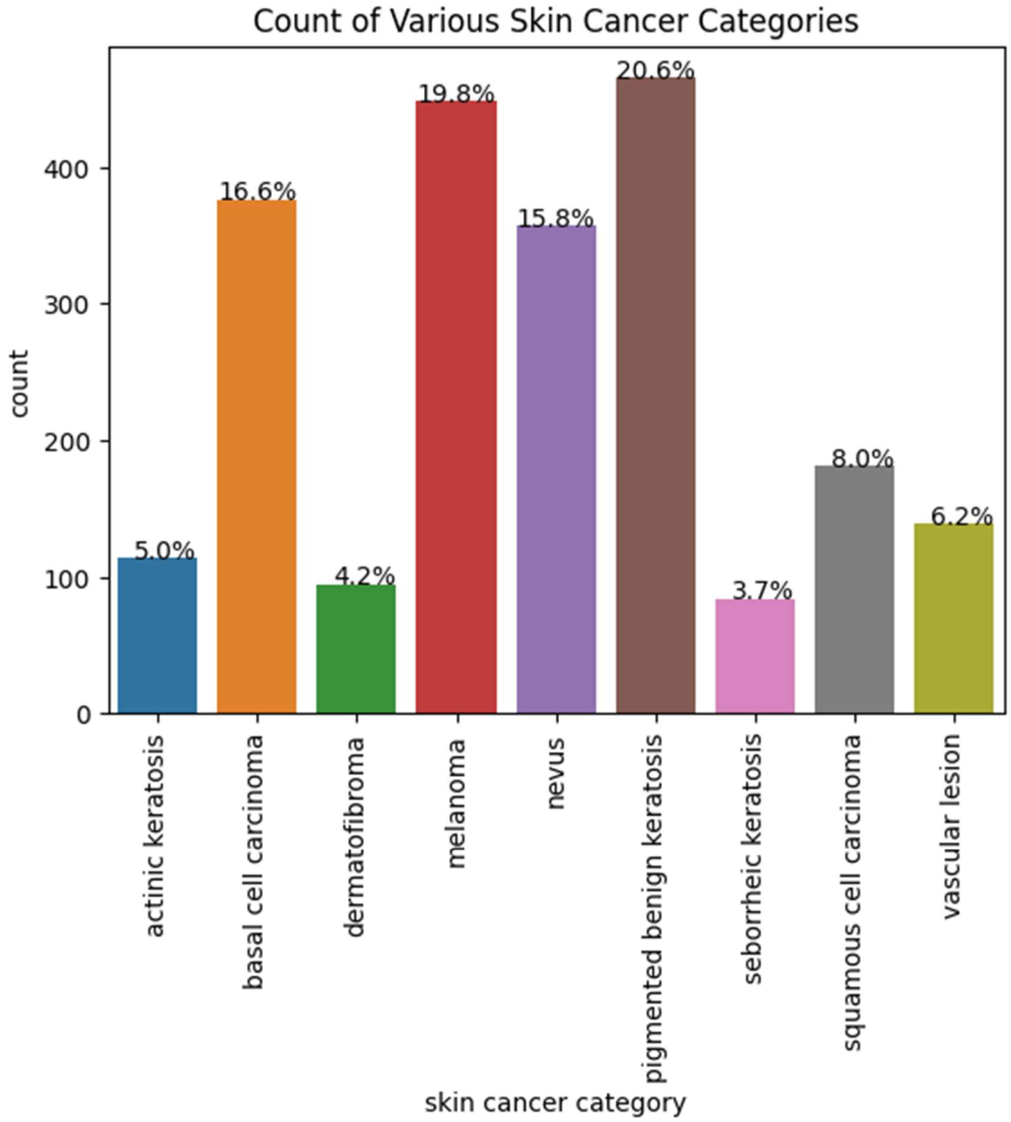
CNN Architecture

The following steps are followed in CNN Model Design:

- Rescalling the Images - For building CNN model it is suggestible to normalize the image input in the [0, 255] range to be in the [0, 1] range.
- Convolution Layer - This Convolutional Layer is made up of 2 nos of 32 filter layer, 1 no of 64 filter layer and 1 no of 128 filter layer, 1 no of dense layer of 512 filter and a softmax output layer.
- Pooling Layer - Total two Pooling layers are used to reduce the dimensions of the feature maps. The pool layer has kernel size of (2,2). A bigger kernel size is not suitable for this problem, since it may result in loss of additional information.
- Dropout Layer - To prevent overfitting drop out layer of 0.2 value has been added. 0.2 represents 20% of neurons to be randomly dropped in the corresponding layer.
- Flatten Layer - Flattening is done to convert a single layer input for softmax to classify.
- Dense Layer - The dense layer is a neural network layer that is connected deeply, which means each neuron in the dense layer receives input from all neurons of its previous layer.
- Activation Function(ReLU) - The hidden layers are given activation function of 'Relu' which gives out only positive value output or zero.
- Activation Function(Softmax) - To predict multiclass, we used 'softmax' as the activation function for output layer.
- Augmentor - To rectify class imbalance in the dataset, we used 'Augmentor' pipeline to randomly generate images of the corresponding category. This helps in increasing training data count for the model to get trained.

Conclusions

Distribution of skin cancer classes in the dataset:



- From the above picture we can see that following classes constitute major part of the dataset and are dominant:

* Pigmented Beningn Keratosis (20.6%)

* Melanoma (19.8%)

* Basal Cell Carcinoma (16.6%)

The following are the skin cancer classes which are under represented:

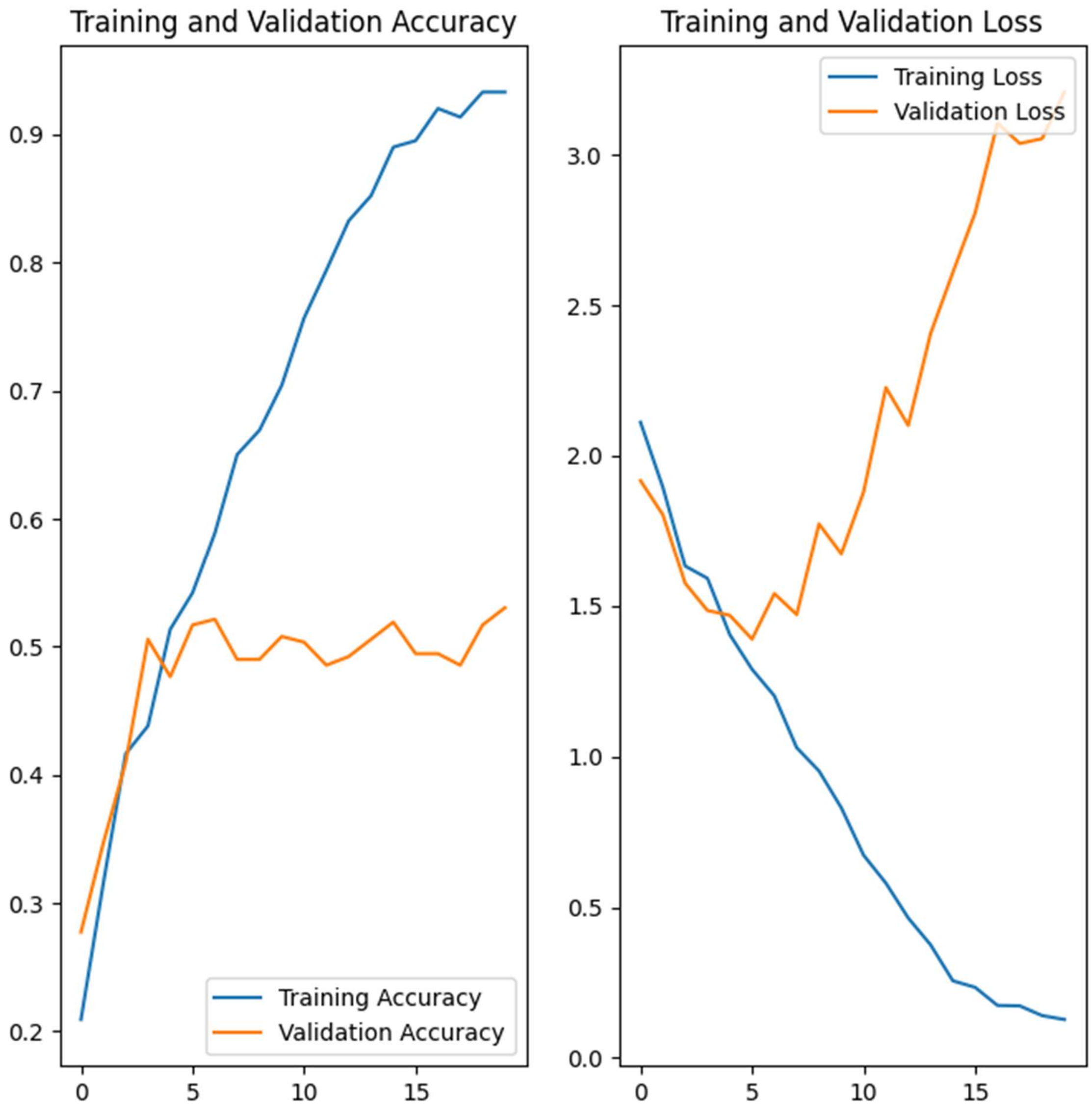
* Seborrheic Keratosis (3.7%)

* Dermatofibroma (4.2%)

* Actinic Keratosis (5.0%)

This class imbalance would result in a model that is biased towards the skin cancer classes which are major in number. Thus we have to rectify the class imbalance using Augmentor Pipeline.

Training and Validation Accuracy on first model:

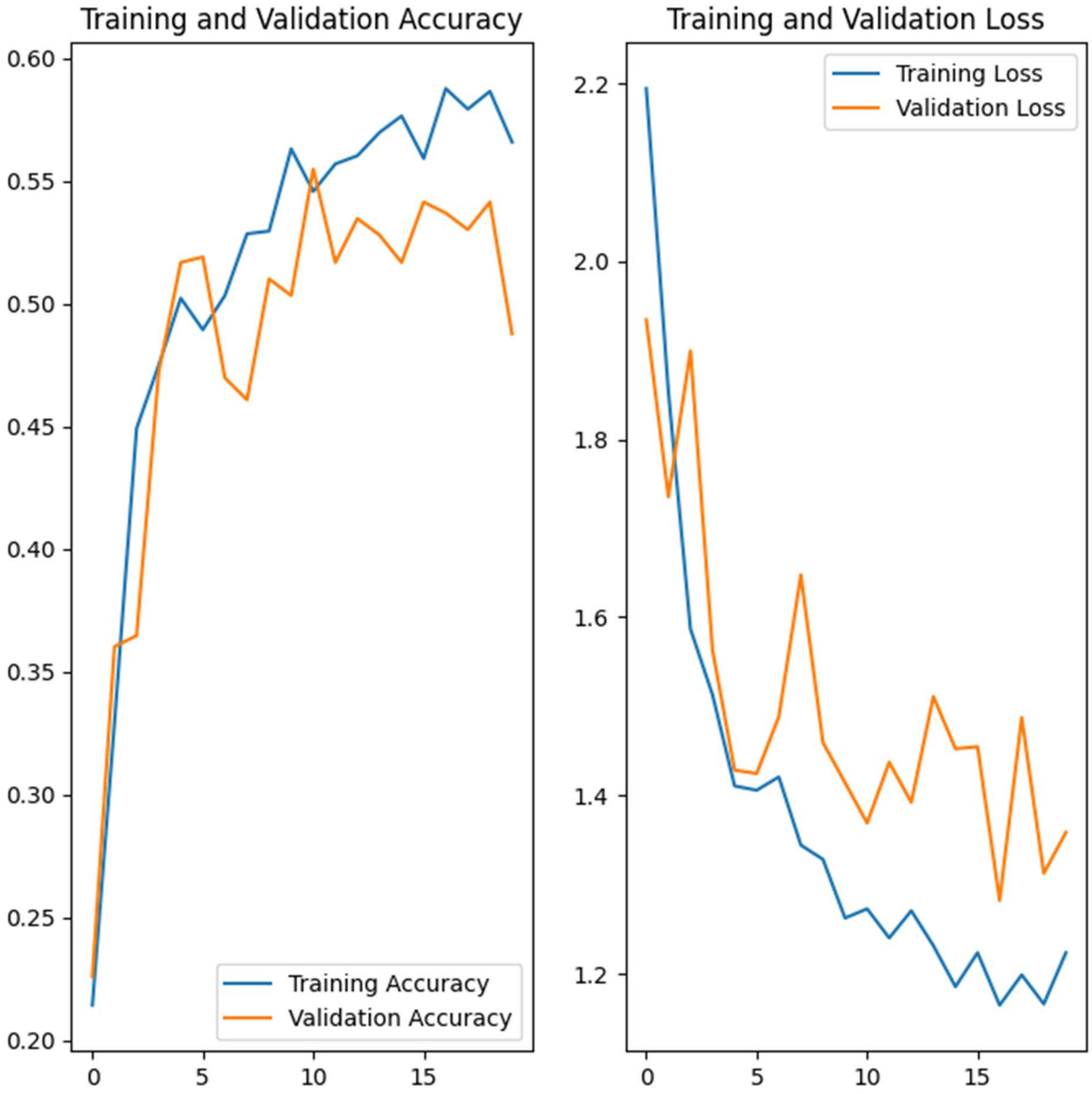


- From the above plot, we can see that there is significant difference between training accuracy (Apprx 93%) and validation accuracy (Apprx 53%) of the last epoch.

- This suggest that there is an overfitting in the model.

- This overfitting can be resolved by using dropout function, adding more training images (i.e., by augmentor pipeline), increasing diversity of images by augmentation (i.e., by random roation, flipping, zooming)

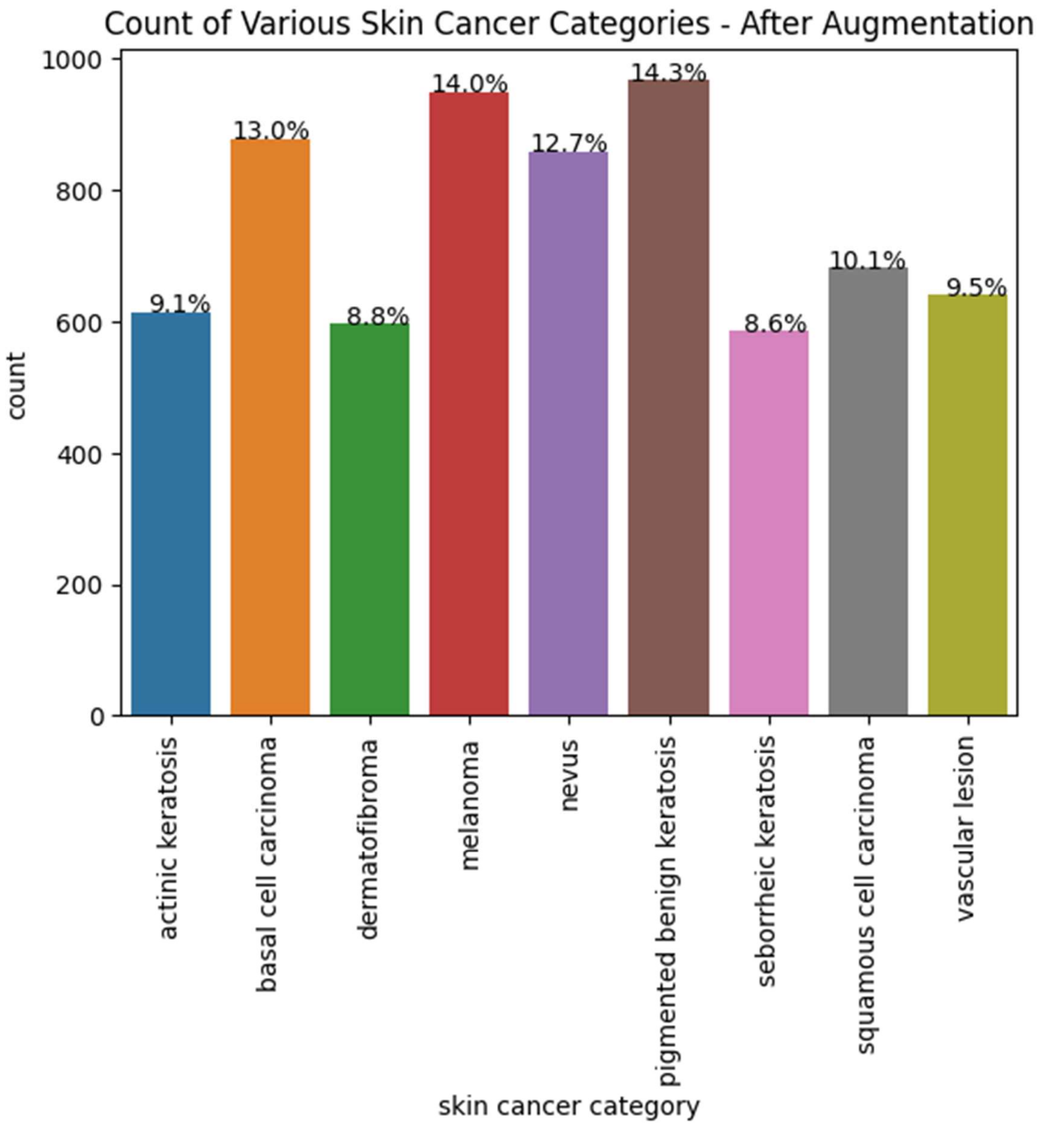
<u>Increasing diversity of data by Augmentation (i.e., by random rotation, flipping and zooming)</u>



- From the above figure, we can see the overfitting problem has been greatly reduced after providing diverse pictures through augmentation and adding a drop out layer at the last hidden layer.

- The training accuracy (apprx 56%) and validation accuracy (apprx 50%) of the last epoch shows that there is no overfitting. But there is still room for improvement.

Augmentor Pipeline:

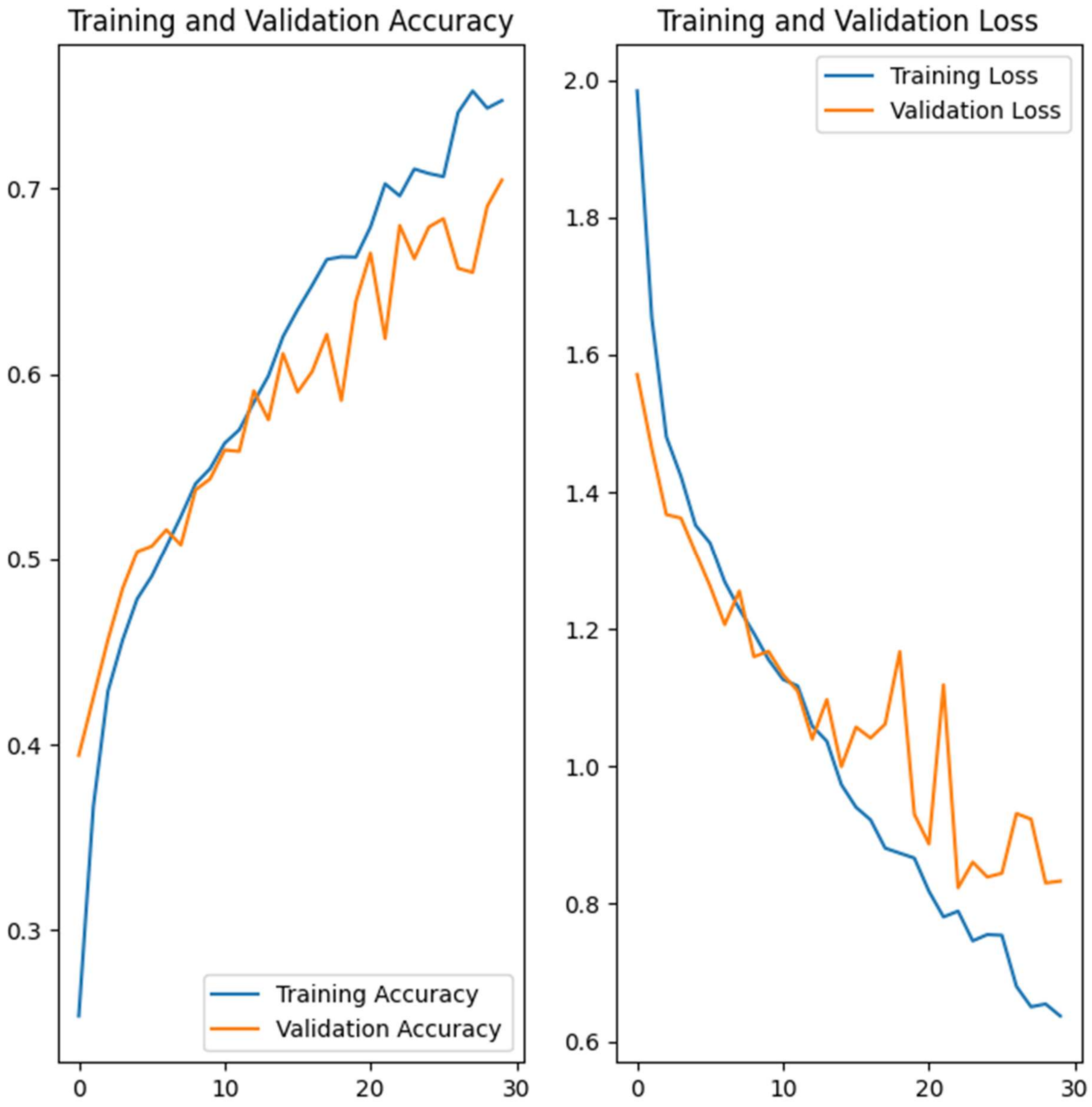


- To rectify the class imbalance, we use 'Augmentor Pipeline'.

- This would add images to the respective skin cancer category, thus making training data more in number.

- From the above percentage plot, we can see that the skin cancer classes which are under represented in previous were increased to 9% approximately.

Final Model:



- After running model for 30 epochs on augmented data, we can see that training accuracy (74 %) and validation accuracy (approximately 70 %) has increased.

- Due to computational constraint, this project is not able to try adding more hidden layers and adding dropouts to improve the accuracy.

- It is observed for this dataset that, Batch Nomralisation is resulting in greater fluctuation of validation accuracies during different epoch runs.

Referrence

Melanoma Skin Cancer from https://www.cancer.org/cancer/melanoma-skin-cancer/about/what-is-melanoma.html

Efficient way to build CNN architecture from https://towardsdatascience.com/a-guide-to-an-efficient-way-to-build-neural-network-architectures-part-ii-hyper-parameter-42efca01e5d7