Melanoma Detection Assignment

Problem Statement: In this assignment, ask is to build a CNN (Convolutional Neural Network) model which can evaluate the images and help the dermatologists to detect the presence of melanoma.

The dataset consists of 2357 images of malignant and benign oncological diseases, which were formed from the International Skin Imaging Collaboration (ISIC) The dataset consists of 9 categories of diseases. We need to create a CNN model which can accurately detect 9 classes present n the dataset.

Need to implement the concept the Data augmentation and handle class imbalance(if any) to sort out the model Overfit/Underfit issue

We followed the below steps to build CNN model:

Step1. We imported the regular python libraries and additionally imported the keras library as we will be handling big size images. With Keras library, we will be using image_dataset_from_directory utility, sequential function for easier data transformation on these images

Step2: As a next set up, we defined the path for the train and test data set and set up the parameters to define batch size and image size for the training data

Step3: We took the next step to set up the train data and validation data using the image size defined in parameter.

Step4: We will visualize one instance of each category of data using matplotlib library

Step5: Then we created a CNN model using Sequetial utility and **layers.experimental.preprocessing.Rescaling**

Step6: Then we compiled the model using 'adam' optimizer. Adam optimizer is the most preferred optimizer as it can automatically tune the learning rate

Step7: Thereafter, we passed the train model passing multiple time using epoch=20

Step7: Then we visualized the results by checking Training accuracy vs validation accuracy and validate the model fit. We found that there is huge difference in the accuracy between the two. Hence it is a Overfitting model.

Data Augmentation:

Step8: To address the model overfit, we used data augmentation. For this we used data transformation techniques like randonflip, randomrotation, randomzoom.

Step9: Then we created the CNN model again and trained this model for epoch=20

Step 10: We visualized the model again using matplotlib and found that , the model this time after data augmentation, came out to be a better fit.

Class Imbalance:

Step 11: As a next step, We analyzed the distribution of class and its samples to identify if there is any class imbalance. We used sklearn for this task

Step12: We found that Class 0,2 and 6 have very less samples and class 6 being the least with 77 samples. Model may become biased with this class imbalance, hence we will have to address this

Step13: To address this class imbalance, we randomly added 500 samples in each class using Augmenter

Step14: We then built the CNN model again using this augmented data where class imbalance was addressed and trained the model with epoch=30

Step15: We then checked the accuracy of training data and validation data set and found that the model is a better fit where Train data set is showing accuracy of ~94% and validation data set is showing frequency of ~82%.