Capstone Project Weekly Progress Report

Semester	Fall 2022
Course Code	AML 2404
Section	Section 2
Project Title	Skin Diseases Classification using Deep Learning
Group Name	G
Student names/Student IDs	Tomson George (C0857730) Praveen Mahaulpatha (C0860583) Thulana Abeywardana (C0861333) Jaskaran Singh Moti (C0860026)
Reporting Week	Week 4 (02 October 2022 to 08 October 2022)
Faculty Supervisor	William Pourmajidi

1. Tasks Outlined in Previous Weekly Progress Report

Task 01 & 02: Improving Model performance of Model 1 and Model 2

Responsible: (Thulana, Praveen)

In these tasks, the key goal was to attain a better performance of a deep learning model we have created in the previous weeks. We have created two separate models (Model 1 and Model2) with different parameters to expedite the process of finding the best-performing model. But the team encountered the model for not performing well on testing data despite it performing well on training data. This was further understood as a cause of a phenomenon named 'overfitting' where the model learns too specific on training data and fails to generalize its predictions on unseen data (in this case the testing data) hence causing a lower performance of the models. In the following sections, the methods that were studied in dealing with overfitting and experimented with to increase the performance will be elaborated along with the performances.

Task 03: Create a server and understand how to implement the model in the cloud and make it live.

Responsible: (Tomson)

Our goal was to find an applicable method to deploy our trained model on a cloud server and to host an already trained model. Once that is done, it is necessary to get the predicted output from the hosted model back to a local device. These steps were targeted for this specific task.

Task 04: Plotting the validation results using Plotly

Responsible: (Jaskaran)

The objective of the task is to get familiarize with the plotly library offered by python which is a library provided for creating interactive visualizations. Even though there will be no dashboard for the end user as planned in the project deliverable, our goal is to test the library by plotting model performance parameters such as the accuracy and losses hence giving us enough exposure to provide visualizations for the end user in the second phase of the project.

2. **Progress Made in Reporting Week**

Task 1.1 : Praveen: Improving Model performance – Model 01

In this section, the performances with respect to the changes done in Model 1 will be discussed. The changes were done with the objective of reducing overfitting and getting a better performance of the model. As per the literature conducted, some of the most used mode generalization (reduction of overfitting) methods are:

- Reducing the complexity of the model (Reducing layers or the number of neurons in a layer)
- Use of Data Augmentation (Increasing the number of images in the same dataset by flipping, translation, rotation, scaling, etc.)
- Use Regularization Techniques
- Use Dropouts
- Adding Early stoppings (To stop model training if a given criteria increases, i.e. validation loss)

The steps taken have not been limited to the above-mentioned generalization techniques and necessary other parameters were experimented with to get an overall better performance of the model.

Step 1:

In the previous week, the implemented model only contained 2 outputs, but as our end goal is to predict multiple skin diseases, we tweaked the model to give 3 outputs hence making it a multi-class classifier from a binary classifier. The corresponding neural network structure and the main parameters are given in below Table 1.1.1 and Table 1.1.2.

Layer	Kernels	Filter Size	Activation Function	Shape / Prob
Input Layer				(244,244,3)
Convo2D	32	3*3	“relu”	
MaxPool2D				
Convo2D	32	3*3	“relu”	
MaxPool2D				
Convo2D	64	3*3	“relu”	
MaxPool2D				
Dropout				0.4
Flatten				
Dense				128
Output Layer				3

Table 1.1.1

Parameter	Value
Learning Rate	0.01
Epochs	20
Loss Function	SparseCategoricalCrossentropy (logits = False)

Table 1.1.2

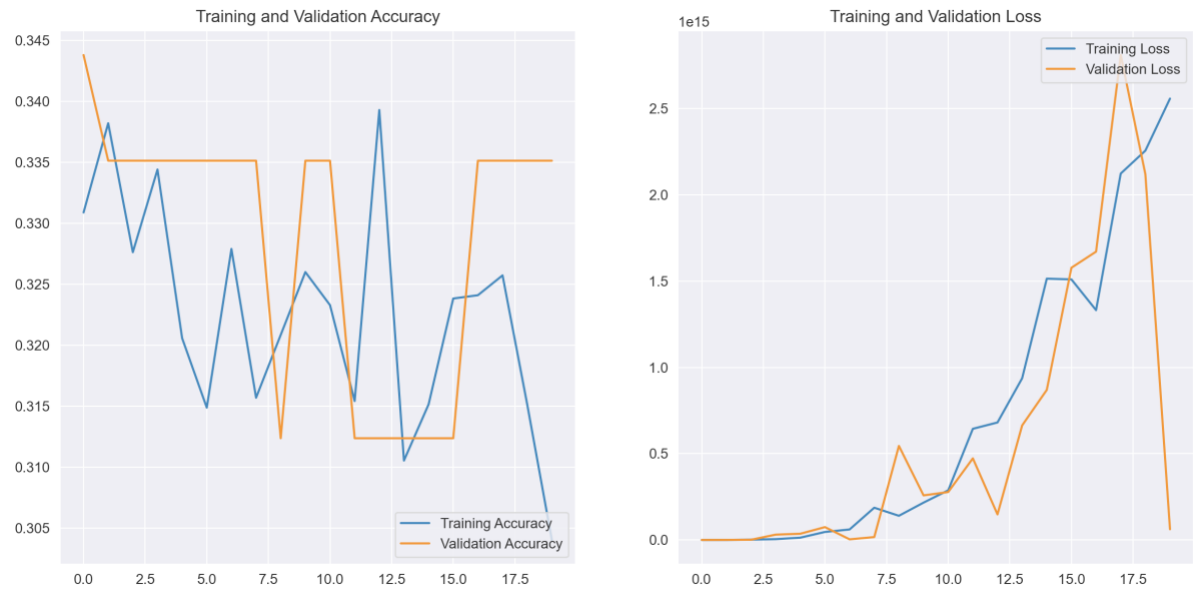

Figure 1.1

Figure 1.1 represents the model for 3 outputs performed relatively poorly to 2 outputs, further the training loss and validation loss seem to have an exponentially increasing trend.

Step 2:

In this step, the input was simplified by decreasing the dimensionality of the input layer thus giving the model less complexity and less chance of overfitting. For this purpose we transformed the input images which had 3(Reg/Green/Blue) color channels to 1 by converting to greyscale. By doing so the input shape of (244,244,3) became (244,244,). The corresponding neural network structure and the main parameters are given in below Table 1.2.1 and Table 1.2.2.

Layer	Kernels	Filter Size	Activation Function	Shape / Prob
Input Layer				(244,244,)
Convo2D	32	3*3	“relu”	
MaxPool2D				
Convo2D	32	3*3	“relu”	
MaxPool2D				
Convo2D	64	3*3	“relu”	
MaxPool2D				
Dropout				0.4
Flatten				
Dense				128
Output Layer				3

Table 1.2.1

Parameter	Value
Learning Rate	0.01
Epochs	20
Loss Function	SparseCategoricalCrossentropy (logits = False)

Table 1.2.2

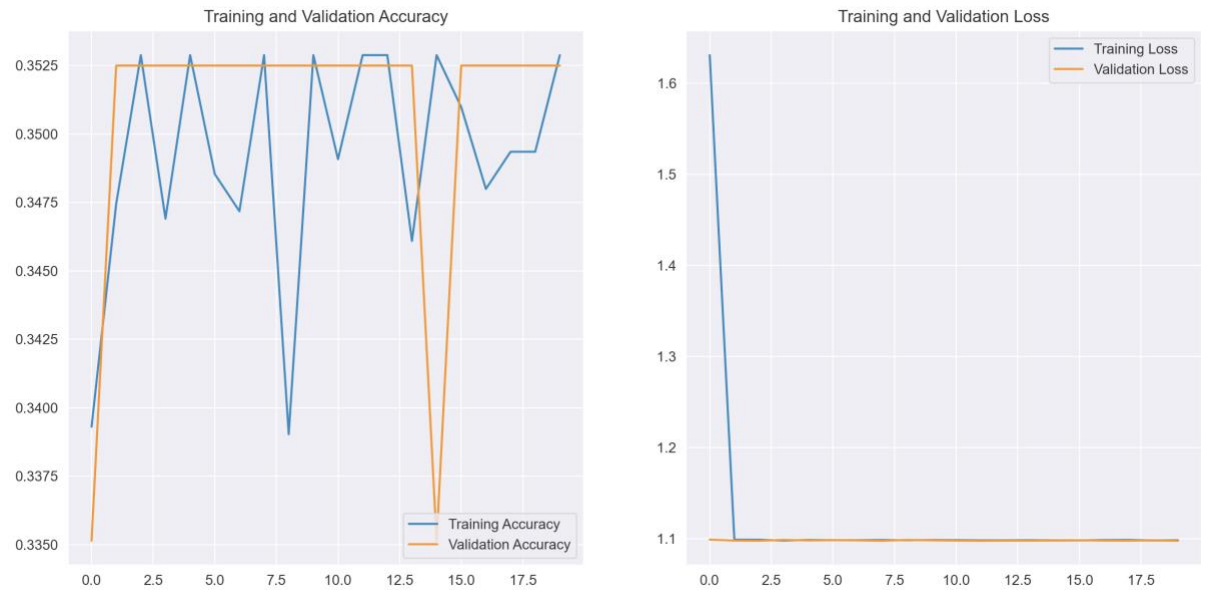

Figure 1.2

Figure 1.2 indicates once the input shape is simplified, the model does not show any improvement in accuracy and a steep drop is visible in the Training and Validation comparison table, which suggests that the model did not train effectively and sudden increases and drops could be accounted due to the learning rate having a higher value.

Step 3:

From step 2 we identified that the model learning rate needed to be reduced in order to obtain a smoother curve of the Graphs hence we decreased the learning rate to 0.001 from 0.01. The corresponding neural network structure and the main parameters are given in below Table 1.3.1 and Table 1.3.2.

Layer	Kernels	Filter Size	Activation Function	Shape / Prob
Input Layer				(244,244,)
Convo2D	32	3*3	“relu”	
MaxPool2D				
Convo2D	32	3*3	“relu”	
MaxPool2D				
Convo2D	64	3*3	“relu”	
MaxPool2D				
Dropout				0.4
Flatten				
Dense				128
Output Layer				3

Table 1.3.1

Parameter	Value
Learning Rate	0.001
Epochs	20
Loss Function	SparseCategoricalCrossentropy (logits = False)

Table 1.3.2

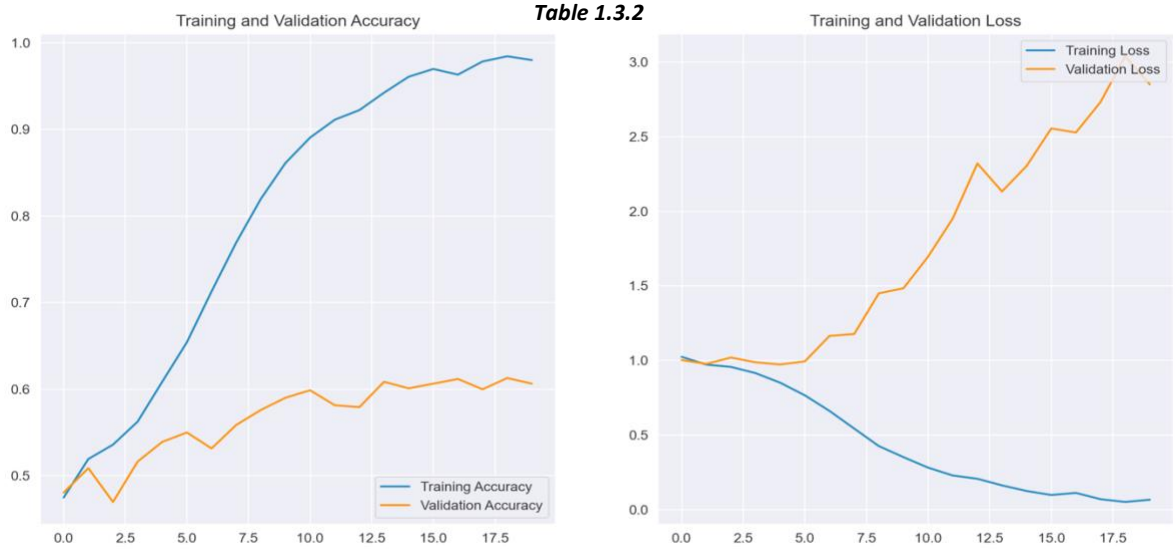

Figure 1.3

The first graph of Figure 1.3 indicates even though the learning accuracy is increasing the relative increase of validation accuracy is low and a slight downward trend can be seen at the end. Also, the second graph shows that even though the training loss is decreasing, the validation loss is increasing thus suggesting still that overfitting exists.

Step 4:

By step 3 we have understood reduction of dimensionality or tweaking the learning rate will not yield desired output hence in this step we tried changing the parameters of the Data Augmentation function. For this purpose we had already used the inbuilt method named 'ImageDataGenerator' from 'keras.preprocessing.image' library. We changed some of the values of the parameters to experiment. The corresponding neural network structure and the main parameters remain same as previous step but the change done to data augmentation function parameters are listed in Table 1.4.1

Parameter	Current Value	Previous Value
Rotation_range	35	40
Width_shift_range	0.3	0.1
Height_shift_range	0.3	0.1

Table 1.4.1

Figure 1.4

The first graph of Figure 1.4 indicates changing the parameters of the data augmentation function does not yield a significant change in performance hence we can assume that the data augmentation is at its optimum level.

Step 05:

In this step, we tried removing the dropout layer to find the significance of that. The corresponding neural network structure and the main parameters are given in below Table 1.3.1 and Table 1.3.2.

Layer	Kernels	Filter Size	Activation Function	Shape / Prob
Input Layer				(244,244,)
Convo2D	32	3*3	“relu”	
MaxPool2D				
Convo2D	32	3*3	“relu”	
MaxPool2D				
Convo2D	64	3*3	“relu”	
MaxPool2D				
Flatten				
Dense				128
Output Layer				3

Table 1.5.1

Parameter	Value
Learning Rate	0.001
Epochs	20
Loss Function	SparseCategoricalCrossentropy (logits = False)

Table 1.5.2

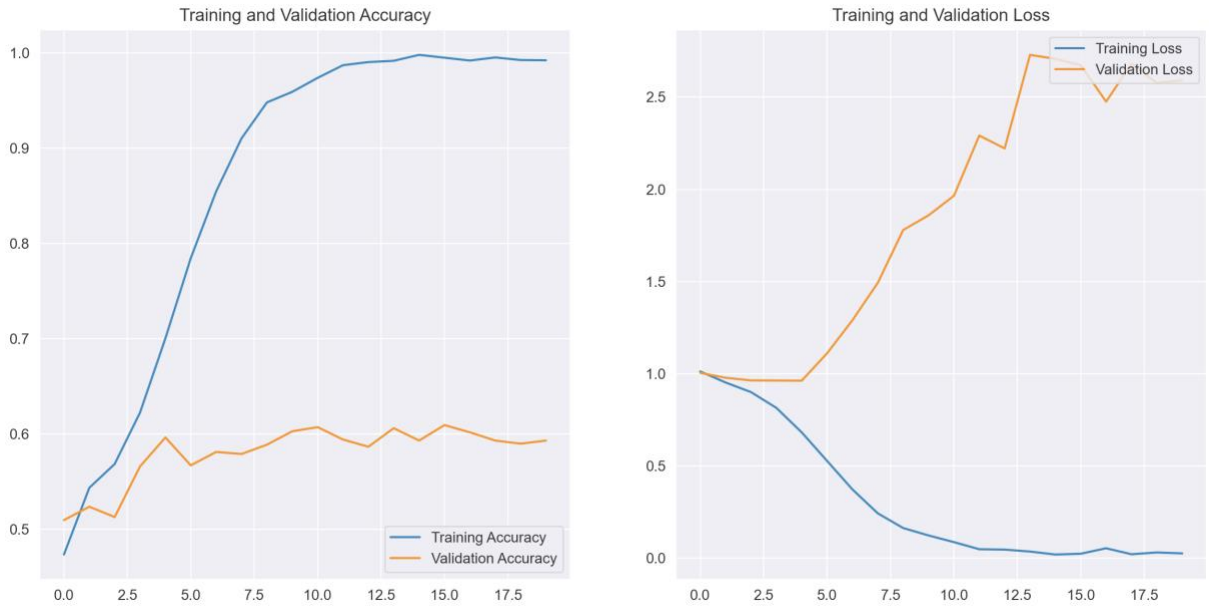

Figure 1.5

According to Figure 1.5, it was evident that we could not find a significant improvement in the validation accuracy and validation loss. For this reason, we have to look deeper for the right solution.

Task 1.2: Thulana: Improving Model performance – Model 02

In this section, the performances with respect to the changes done in Model 2 will be discussed. The changes were done with the objective of reducing overfitting and getting a better performance of the model. Some of the methods applied to counter overfitting are:

- Including dropout layers
- L2 regularization

Step 1:

In the previous week, the validation loss was increasing against the training loss thus concluding overfitting. Dropout layer is one method where randomly dropping neurons in dense layers prevents all neurons in a layer from synchronously optimizing their weights, thus decorrelating the weights and giving less probability to be highly dependent on specific layer structure.

Layer	Kernels	Filter Size	Activation Function	Shape / Prob
Convo2D & Input Layer	32	3*3	"relu"	(255,,255,3)
MaxPool2D				
Flatten				
Dense			"relu"	127
Dropout				0.5
Dense			"relu"	256
Dropout				0.5
Output Layer			"softmax"	4

Table 2.1.1

Parameter	Value
Learning Rate	0.01
Epochs	10
Loss Function	SparseCategloricalCrossentropy (logits = False)

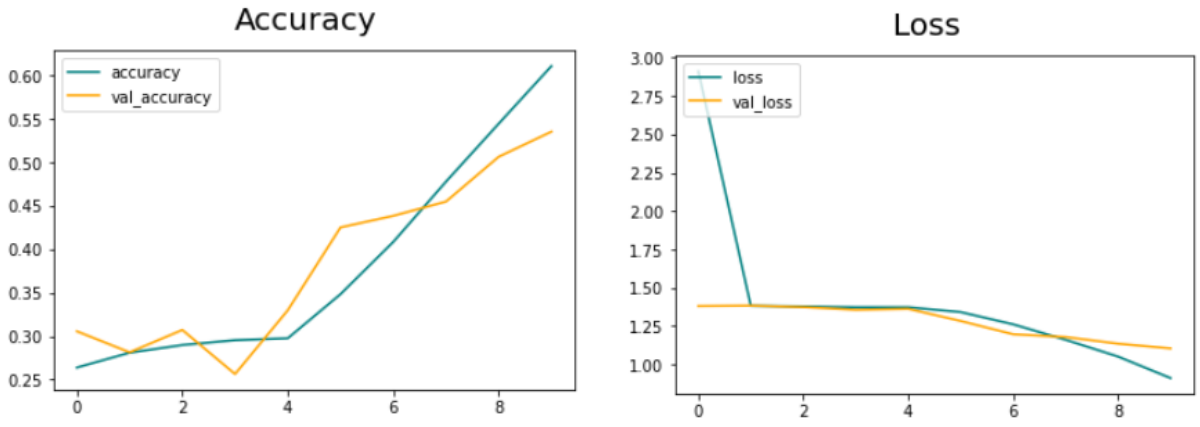

Figure 2.1

Figure 2.1 above shows that, the validation accuracy is increasing along with training accuracy and the losses are decreasing. But, the trend is not as smooth as we expect. For this matter, further generalization methods need to be carried out.

Step 2:

In this step, we have considered L2 regularization method . These methods update the general cost function by adding another term known as the regularization term. Due to the addition of this regularization term, the values of weight matrices decrease because it assumes that a neural network with smaller weight matrices leads to simpler models. Therefore, it will also reduce overfitting to quite an extent. L2 regularization is also known as weight decay as it forces the weights to decay towards zero but not exactly 0. Also, we have increased the number of convolutional layers to increase the accuracy level of the model.

Layer	Kernels	Filter Size	Activation Function	Shape / Prob	Regularizer
Convo2D & Input Layer	32	3*3	"relu"	(255,,255,3)	Kernel_regularizer=0.01 bias_regularizer = 0.01
MaxPool2D					
Convo2D & Input Layer	64	3*3	"relu"	(255,,255,3)	Kernel_regularizer=0.01 bias_regularizer = 0.01
MaxPool2D					
Flatten					
Dense			"relu"	127	
Dropout				0.5	
Dense			"relu"	256	
Dropout				0.5	
Dense			"relu"	127	
Dropout				0.5	
Output Layer			"softmax"	4	

Table 2.2.1

Parameter	Value
Learning Rate	0.01
Epochs	20
Loss Function	SparseCategloricalCrossentropy (logits = False)

Table 2.2.2

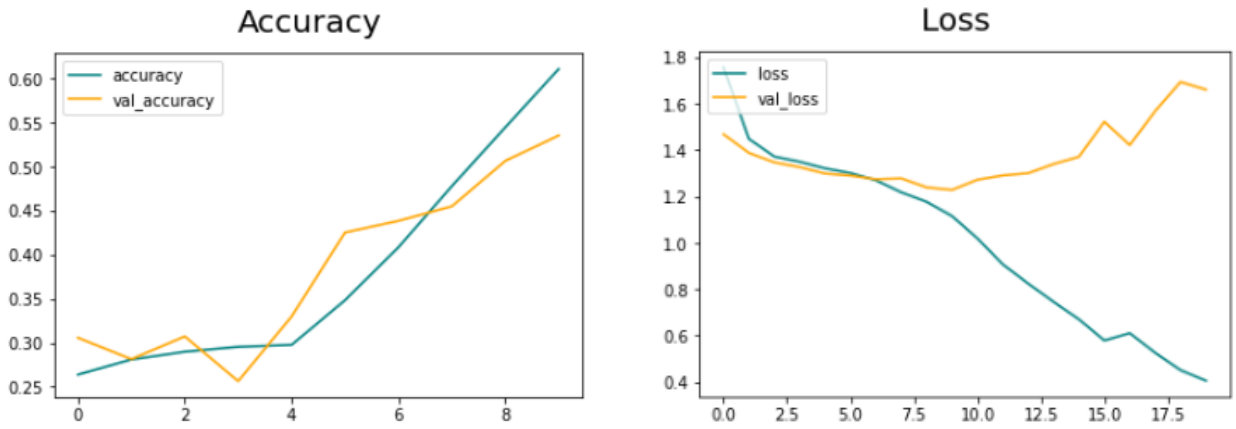

Figure 2.2

Step 3:

In this step, L2 regularization and one convolution layer was removed from the model. Then the model was trained for 15 epochs. However, performance of the model is further decreased after this step.

Layer	Kernels	Filter Size	Activation Function	Shape / Prob
Convo2D & Input Layer	32	3*3	"relu"	(255,,255,3)
MaxPool2D				
Flatten				
Dense			"relu"	128
Dropout				0.5
Dense			"relu"	256
Dropout				0.5
Dense			"relu"	64
Dropout				0.5
Output Layer			"softmax"	4

Table 2.3.1

Parameter	Value
Learning Rate	0.01
Epochs	20
Loss Function	SparseCategloricalCrossentropy (logits = False)

Table 2.3.2

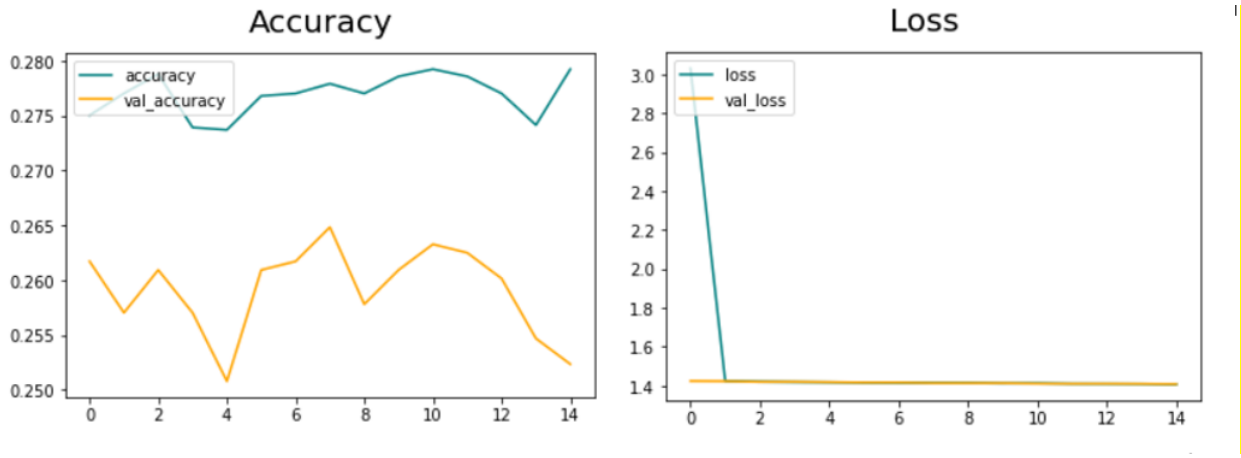

Figure 2.3

Out of figures 2.1, 2.2 and 2.3, only figure 2.2 shows a comparative fare accuracy level, but the validation loss is inclined after 10 epochs. Considering the model performances evaluated last week and this week, significant progress has not been observed in changing the hyperparameters and the layer in the Keras. Therefore, it is concluded that data preprocessing needs to be done again before the training and validation of the model.

Task 3: Tomson: Create a server and understand how to implement the model in the cloud and make it live.

In these tasks, the key goal was to attain a better performance of a deep learning model we have created in the previous weeks. We have created two separate models (Model 1 and Model2) with different parameters to expedite the process of finding the best-performing model. But the team encountered the model for not performing well on testing data despite it performing well on training data. This was further understood as a cause of a phenomenon named 'overfitting' where the model learns too specific on training data and fails to generalize its predictions on unseen data (in this case the testing data) hence causing a lower performance of the models. In the following sections, the methods that were studied in dealing with overfitting and experimented with to increase the performance will be elaborated along with the performances.

Further, we created a code pipeline for the smooth deployment of the application. It will also ensure the redeployment of the code each time it is updated. It consists of five stages, which are

1. Choosing pipeline settings: This stage is mainly for configuring the pipeline with a pipeline name, service role and role name.
2. Add source stage: Here we connected with our GitHub repository so that the server can fetch the code both initially and each time code is updated.
3. Add build stage: We skipped this option since it was not needed.
4. Add deploy stage: In this stage, the server on which the code is to be deployed is selected. Here, the EBS instance previously created was selected.
5. Review: This is the final stage. Here the configurations are reviewed before creating the pipeline.

Using EBS and CodePipeline service provided by Amazon, there was a noticeable reduction in the complexities like manually setting up an EC2 instance and making it a server, uploading code to the created server, and making it live.

Task 4: Jaskaran: Plotting the validation results using Plotly

Basic knowledge to plot different types of charts has been learned. Generally worked on 2D plots and plotly. Plotly consists of three major parts. They are:

- plotly.plotly
  - Serves as the interface between the local computer and Plotly. It has features that call on the Plotly server for a response
- plotly.graph.objects
  - Comprises the objects (Figure, layout, data, and definition of the plots like scatter plot, line chart) that oversee producing the plots.
- plotly.tools
  - Includes a number of tools in the shape of functions that help improve the Plotly user experience.

Below are some of the sample 2D plots that we generated with plotly.

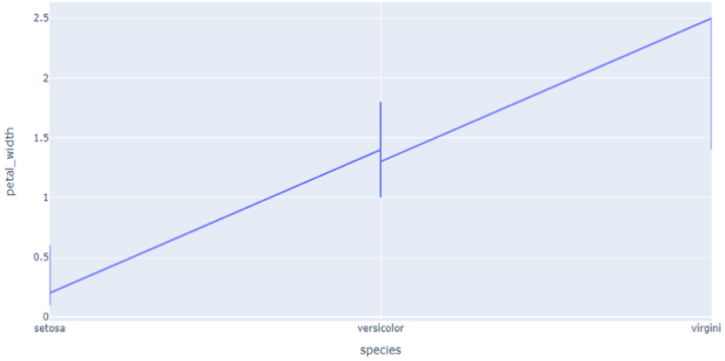

Figure 3.1: Line chart

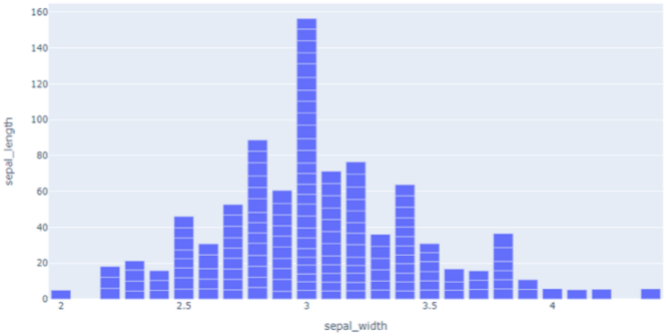

Figure 2.2: Bar chart

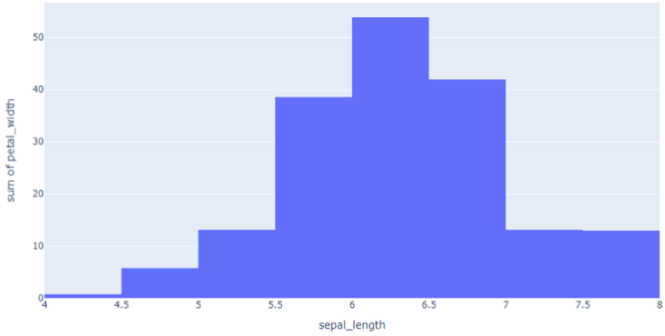

Figure 5.3 : Histogram

Figure 4.4 : Bubble Chart

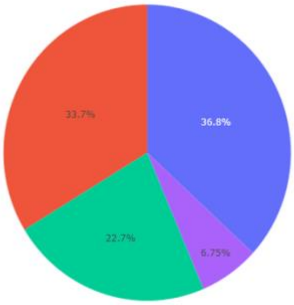

Figure 7.5 : Pie Chart

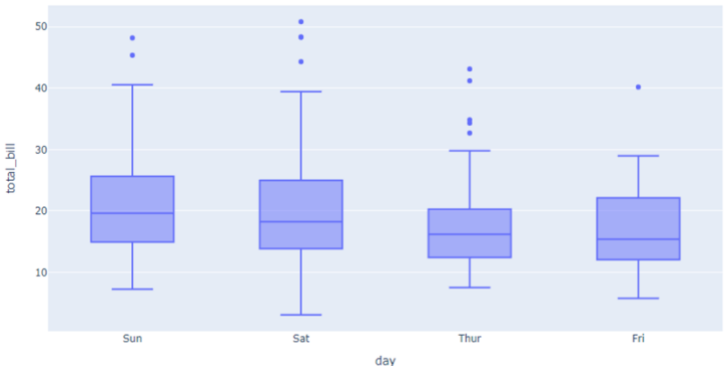

Figure 6.6 : Box plot

It is necessary that we need to experiment and test more on the library to get the maximum output of plotly. This will be carried out in the coming weeks.

3. **Difficulties Encountered in Reporting Week**

   a. 1st model shows a low increase rate of validation accuracy and using different generalization methods did not yield significant progress in model accuracy.
   Solution: Further research and experimentation need to be carried out.
   b. After changing the input shape of the neural network when reducing the dimensionality (Changing the color channels to 1 from 3 explained in step2 of Model1) it was challenging to fit it in the data augmentation function due to shape change.
   Solution: Explicitly define the shape inside the function.
   c. The training and validation take too long due to the size of the dataset(Model 2).
   d. Changing the hyperparameters did not produce considerable progress in the Model 2
   Solution: Data preprocessing
   e. A 502 code error occurred when a sample application was deployed on the server.
   f. Plotting 3D charts are challenging to understand in plotly.

4. **Tasks to Be Completed in Next Week**

Tasks	Responsible
Implementing sample Flask project with and making it live on the server	Tomson
Initialization of MLOps	Praveen
Optimizing Model 2 for Better Validation Accuracy / Validation Loss by data preprocessing	Thulana
Work on complex and 3D charts on plotly.	Jaskaran