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 **7** (**23** October 2022 to **29** October 2022)
Faculty Supervisor	William Pourmajidi

1. Tasks Outlined in Previous Weekly Progress Report

Task 01: Creating a separate python file to Predict a given image
Responsible: (Praveen)
In this task, the objective was to create a separate python application to load a pre-trained model and predict the skin disease given an image. For this, we have saved an instance of model2 which is currently being tested and further improved and pass the loaded model an image located in the local drive. The goal is to implement the same procedure on the cloud upon completion of the cloud architecture.

Task 02: Improving Model performance of Model 02
Responsible: (Thulana)
Image augmentation methods were applied to the model in order to improve model performance, Basically Random Flip and Random Rotation methods were applied to the model. Even though the model shows an improvement in performance compared to the previous steps, the accuracy level has not reached the accepted level.

Task 03: Create a dummy python file on the cloud and access the image uploaded by the user through it
Responsible: (Tomson)
The functionality for uploading the image and saving it on the cloud under the folder 'static' was written the last week. This week's target was to implement the functionality to read the uploaded image data so that it can be passed to the model. But all the attempts that have been done for reading and storing images have been failing this week

Task 04: Learn advances and interactive graph representation using Plotly
Responsible: (Jaskaran)
It was necessary that we experiment and test more on the Plotly library to get the maximum output of plotly. This task helps to identify how to use plotly to better visualization of our model and the dataset(Images). Instead of using Matplotlib, Plotly enables us to easily generate dynamic web charts using Python. With the help of plotly, we can also turn geographic, scientific, statistical, and financial data into animations and interactive graphs. Even there are many complex graphs and visualizations that can be plotted using Plotly which helps us understand the data.

2. Progress Made in Reporting Week

Task 1 : Create a separate python file to Predict a given image

In this task, a separate python file was created to load the pre-trained Model2 and make predictions for a given image. Model2 is currently undergoing improvements, but we have selected the latest to proceed with this implementation. Below shows the simple architecture of the task.

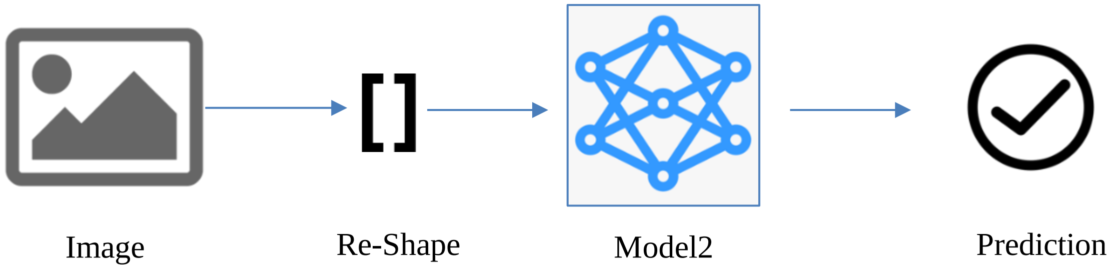


Figure 2.1

We have used an inbuilt function in the Keras library to load the saved model along with its metadata and weights. Once that was done, the image was reshaped in order to feed the model. It was successfully implemented after having to deal with correct reshaping of the image.

Task 2: Thulana: Improving Model performance of Model 02

Step 1:

In addition to the rotation and the flip of the images, the images were zoomed as an image augmentation method. The corresponding parameters of the neural network model are as follows:

Layer	Kernels	Filter Size	Activation Function	Shape / Other parameters
Input Layer				(224,224,3)
Random flip				horizontal_and_vertical
Random rotation				0.2

RandomZoom				height_factor= 0.5,
				width_factor = 0.2

Convo2D	32	3*3	"relu"	
MaxPool2D				
Convo2D	32	3*3	"relu"	
MaxPool2D				
Convo2D	32	3*3	"relu"	
MaxPool2D				
Flatten				
Dense				128
Dropout				0.5
Dense				256
Dropout				0.5
Dense				64
Dropout				0.5
Output Layer				4

Table: 2.1

Parameter	Value
Learning Rate	0.01
Epochs	5
Loss Function	SparseCategoricalCrossentropy (logits = False)

Table: 2.2

Following results were observed:

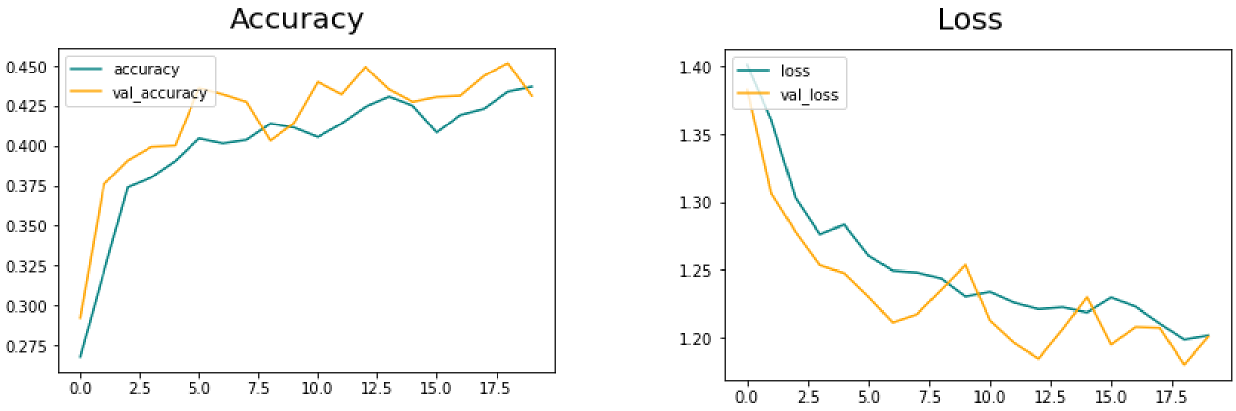


Figure: 2.2

<u>Step 2</u>:
As the model did not achieve desired results, the optimizer changed to 'RMSProp' without changing other parameters:
The corresponding parameters of the neural network model are as follows:

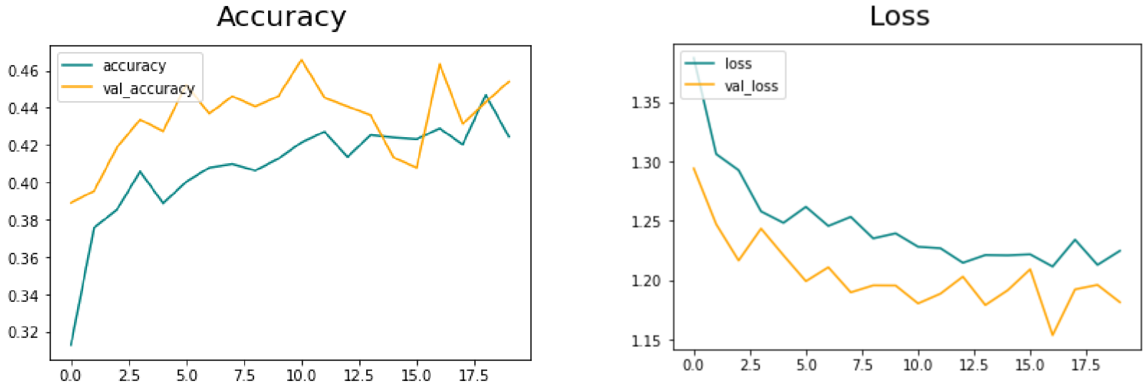


Figure: 2.3

Step 3:
Then learning rate was taken down to 0.007 with Adam optimizer as follows:

Layer	Kernels	Filter Size	Activation Function	Shape / Other parameters
Input Layer				(224,224,3)
Random flip				horizontal_and_vertical
Random rotation				0.2
RandomZoom				height_factor= 0.5, width_factor = 0.2
Convo2D	32	3*3	“relu”	
MaxPool2D				
Convo2D	32	3*3	“relu”	
MaxPool2D				
Convo2D	32	3*3	“relu”	
MaxPool2D				
Flatten				
Dense				128
Dropout				0.5
Dense				256
Dropout				0.5
Dense				64
Dropout				0.5
Output Layer				4

Table: 2.3

Parameter	Value
Learning Rate	0.007
Epochs	5
Loss Function	SparseCategoricalCrossentropy (logits = False)

Table: 2.4

Following results were observed:

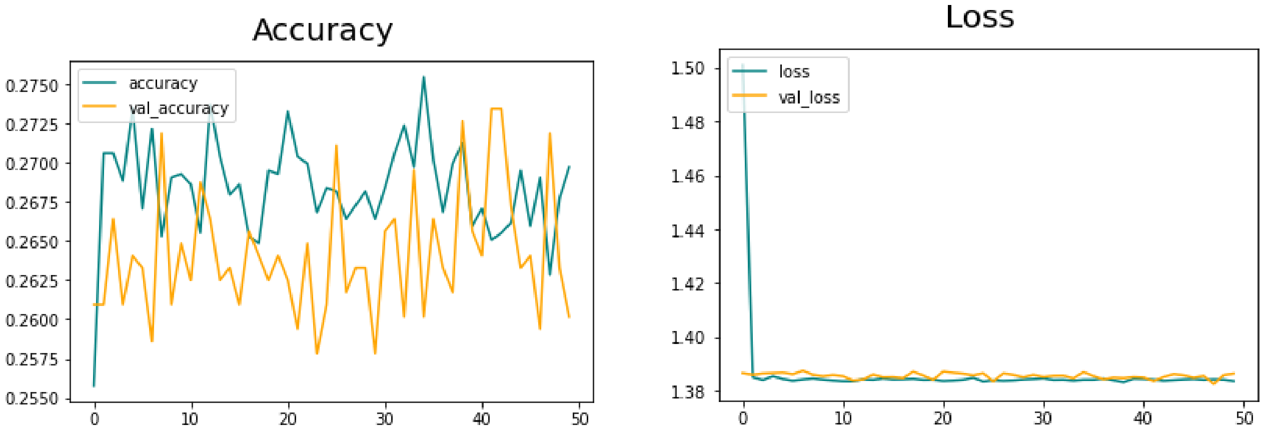


Figure: 2.4

Observations:

Even though data were preprocessed using different types of techniques, desired performances did not achieve. Therefore, need to deal with a different dataset.

Task 3: Create a dummy python file on the cloud and access the image uploaded by the user through it.

Our main usage of this function is to pass the uploaded image to the ML model. The application was successfully launched in an elastic beanstalk server. When the function for saving/reading the image was executed a server error occurred. Further going deep into the logs of the server, it was a file not found error. But the code is perfectly executing in our local windows machine. First, we concluded, it should have been due to the underlying OS dependency. We tested this hypothesis by executing the code into another machine but found out that's not the case. So to get more insights, we decided to manually set up an EC2 instance and execute the code instead of depending on the elastic beanstalk. But we were getting other problems when trying to bring the server live.

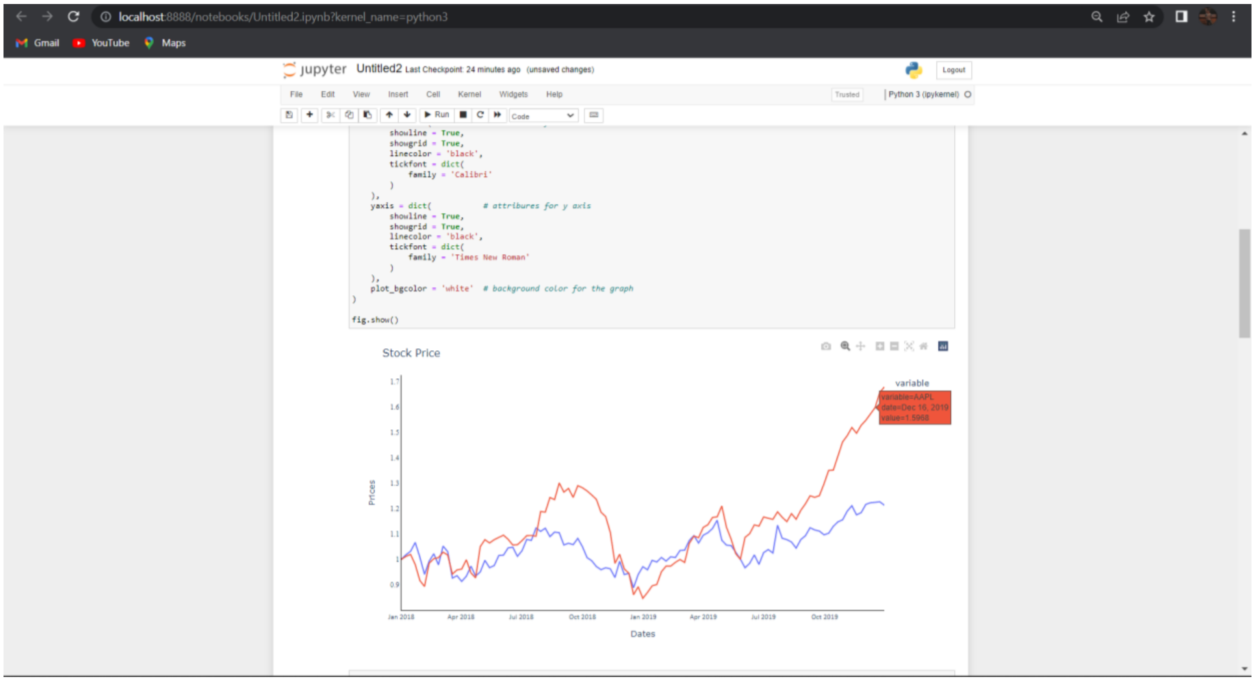
Task 4: Jaskaran: Learn advances and interactive graph representation using plotly.

Learned how to plot images in plotly along with that how we can plot multiple graphs in visualization. Images can be downloaded, zoomed, panned, box selected, lasso selected, auto scaled and reset axes. Also learned how to use images in different plots so that it's easier to understand which data is representing which image.

Experimented how to plot images in plotly along with that how we can plot multiple graphs in visualization.

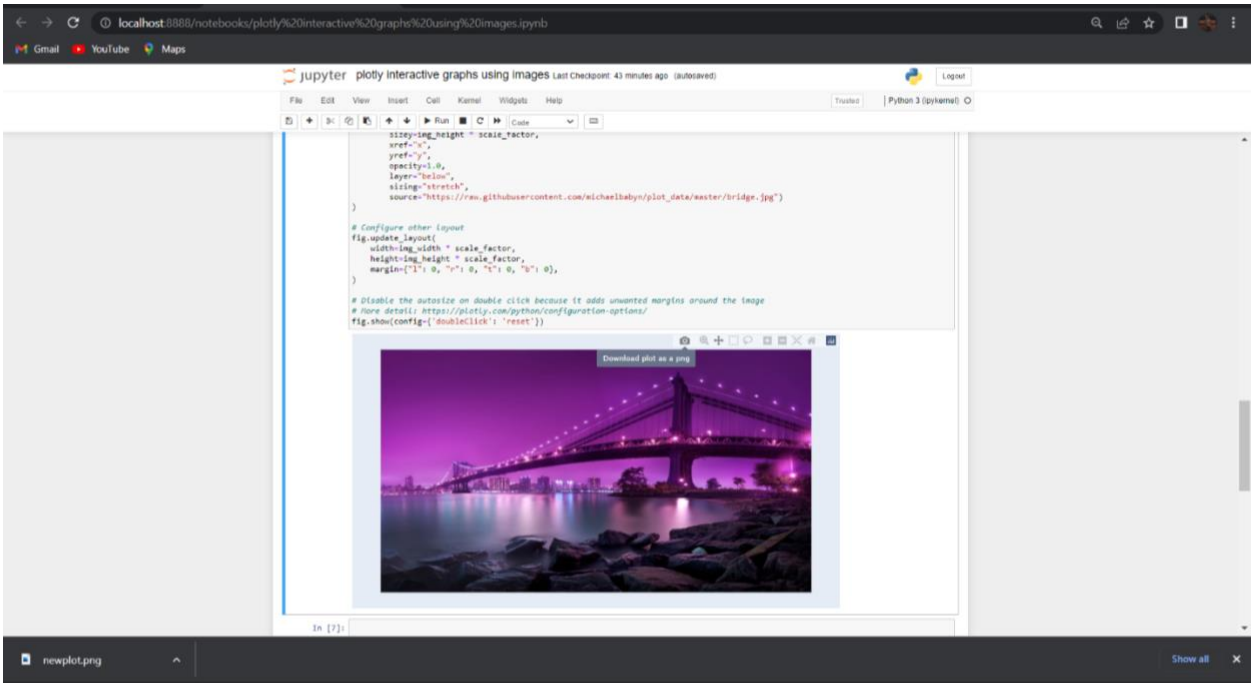
- How to customize the figure.
- How to add traces and update the layouts of the figure with graph objects.

A background image can be added to the layout of a figure with fig.add_layout_image or by setting the images parameter of go.Layout. The source attribute of a go.layout.Image can be the URL of an image, or a PIL Image object (from PIL import Image; img = Image.open('filename.png'))


localhost:8888/notebooks/Untitled2.ipynb?kernel_name=python3
Gmail
YouTube
Maps
jupyter
Untitled2 Last Checkpoint: 24 minutes ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 (ipykernel)
Run
Code

```
        showline = True,
        showgrid = True,
        linecolor = 'black',
        tickfont = dict(
            family = 'Calibri'
        )
    ),
    yaxis = dict(           # attribures for y axis
        showline = True,
        showgrid = True,
        linecolor = 'black',
        tickfont = dict(
            family = 'Times New Roman'
        )
    ),
    plot_bgcolor = 'white'  # background color for the graph
)

fig.show()
```

Figure 2.5

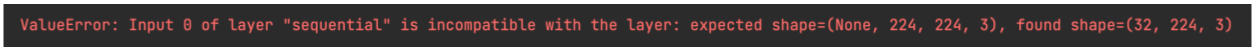

localhost:8888/notebooks/plotly%20interactive%20graphs%20using%20images.ipynb
Gmail
YouTube
Maps
jupyter
plotly interactive graphs using images Last Checkpoint: 43 minutes ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 (ipykernel)
Run
Code

```
        sizey=img_height * scale_factor,
        xref="x",
        yref="y",
        opacity=1.0,
        layer="below",
        sizing="stretch",
        source="https://raw.githubusercontent.com/michaelbabyn/plot_data/master/bridge.jpg")
)

# Configure other layout
fig.update_layout(
    width=img_width * scale_factor,
    height=img_height * scale_factor,
    margin={"l": 0, "r": 0, "t": 0, "b": 0},
)

# Disable the autosize on double click because it adds unwanted margins around the image
# More detail: https://plotly.com/python/configuration-options/
fig.show(config={'doubleClick': 'reset'})
```

Figure 2.6

3. Difficulties Encountered in Reporting Week

a. Making the image fit the pre-trained model threw errors with regard to the input shape. One example is given in Figure 3.1.



```
ValueError: Input 0 of layer "sequential" is incompatible with the layer: expected shape=(None, 224, 224, 3), found shape=(32, 224, 3)
```

Figure: 3.1

b. As the dataset has many images, it takes more time to run the model.

c. Since the main functionalities were not working as expected most of the time is spent on debugging the code.

4. Tasks to Be Completed in Next Week

Tasks	Responsible
Implement the read/write functionality with the help of Boto3	Tomson
Modify Model 01 for better performance	Praveen
Test with a different dataset for Model02	Thulana
Use plotly to read images and give representations related.	Jaskaran