

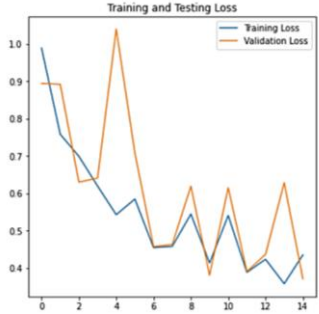
Project Development Phase

Model Performance Test

Date	10 November 2022
Team ID	PNT2022TMID27942
Project Name	Project - Natural Disasters Intensity Analysis and Classification using Artificial Intelligence
Maximum Marks	10 Marks

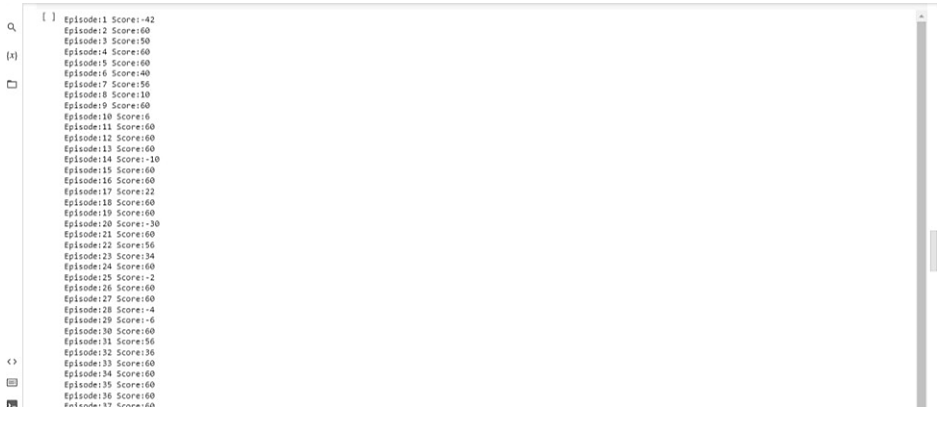
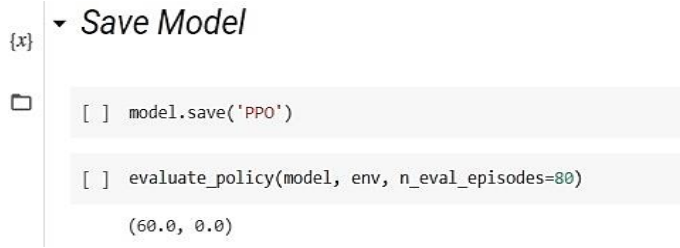
1) Feature-1 (CNN Model)

S.No	Parameter	Screenshot																																							
1	Model Summary	<div> <div>Model Summary</div> <pre>[] model.summary()</pre> <table border="1"> <thead> <tr> <th>Layer (type)</th><th>Output Shape</th><th>Param #</th></tr> </thead> <tbody> <tr> <td>conv2d_16 (Conv2D)</td><td>(None, 74, 74, 32)</td><td>896</td></tr> <tr> <td>max_pooling2d_12 (MaxPooling2D)</td><td>(None, 37, 37, 32)</td><td>0</td></tr> <tr> <td>conv2d_17 (Conv2D)</td><td>(None, 35, 35, 32)</td><td>9248</td></tr> <tr> <td>max_pooling2d_13 (MaxPooling2D)</td><td>(None, 17, 17, 32)</td><td>0</td></tr> <tr> <td>conv2d_18 (Conv2D)</td><td>(None, 15, 15, 64)</td><td>18496</td></tr> <tr> <td>max_pooling2d_14 (MaxPooling2D)</td><td>(None, 7, 7, 64)</td><td>0</td></tr> <tr> <td>conv2d_19 (Conv2D)</td><td>(None, 5, 5, 64)</td><td>36928</td></tr> <tr> <td>max_pooling2d_15 (MaxPooling2D)</td><td>(None, 2, 2, 64)</td><td>0</td></tr> <tr> <td>flatten_3 (Flatten)</td><td>(None, 256)</td><td>0</td></tr> <tr> <td>dense_7 (Dense)</td><td>(None, 300)</td><td>77100</td></tr> <tr> <td>dense_8 (Dense)</td><td>(None, 150)</td><td>45150</td></tr> <tr> <td>dense_9 (Dense)</td><td>(None, 3)</td><td>453</td></tr> </tbody> </table> <p> Total params: 188,271 Trainable params: 188,271 Non-trainable params: 0 </p> </div>	Layer (type)	Output Shape	Param #	conv2d_16 (Conv2D)	(None, 74, 74, 32)	896	max_pooling2d_12 (MaxPooling2D)	(None, 37, 37, 32)	0	conv2d_17 (Conv2D)	(None, 35, 35, 32)	9248	max_pooling2d_13 (MaxPooling2D)	(None, 17, 17, 32)	0	conv2d_18 (Conv2D)	(None, 15, 15, 64)	18496	max_pooling2d_14 (MaxPooling2D)	(None, 7, 7, 64)	0	conv2d_19 (Conv2D)	(None, 5, 5, 64)	36928	max_pooling2d_15 (MaxPooling2D)	(None, 2, 2, 64)	0	flatten_3 (Flatten)	(None, 256)	0	dense_7 (Dense)	(None, 300)	77100	dense_8 (Dense)	(None, 150)	45150	dense_9 (Dense)	(None, 3)	453
Layer (type)	Output Shape	Param #																																							
conv2d_16 (Conv2D)	(None, 74, 74, 32)	896																																							
max_pooling2d_12 (MaxPooling2D)	(None, 37, 37, 32)	0																																							
conv2d_17 (Conv2D)	(None, 35, 35, 32)	9248																																							
max_pooling2d_13 (MaxPooling2D)	(None, 17, 17, 32)	0																																							
conv2d_18 (Conv2D)	(None, 15, 15, 64)	18496																																							
max_pooling2d_14 (MaxPooling2D)	(None, 7, 7, 64)	0																																							
conv2d_19 (Conv2D)	(None, 5, 5, 64)	36928																																							
max_pooling2d_15 (MaxPooling2D)	(None, 2, 2, 64)	0																																							
flatten_3 (Flatten)	(None, 256)	0																																							
dense_7 (Dense)	(None, 300)	77100																																							
dense_8 (Dense)	(None, 150)	45150																																							
dense_9 (Dense)	(None, 3)	453																																							
2	Accuracy	<div> <pre>[] #Training and Validation Accuracy Plots epochs_range = range(15) plt.figure(figsize=(6,6)) plt.plot(epochs_range, history.history['accuracy'], label='Training Accuracy') plt.plot(epochs_range, history.history['val_accuracy'], label='Validation Accuracy') plt.legend() plt.title('Training and Testing Accuracy') plt.show()</pre> </div>																																							

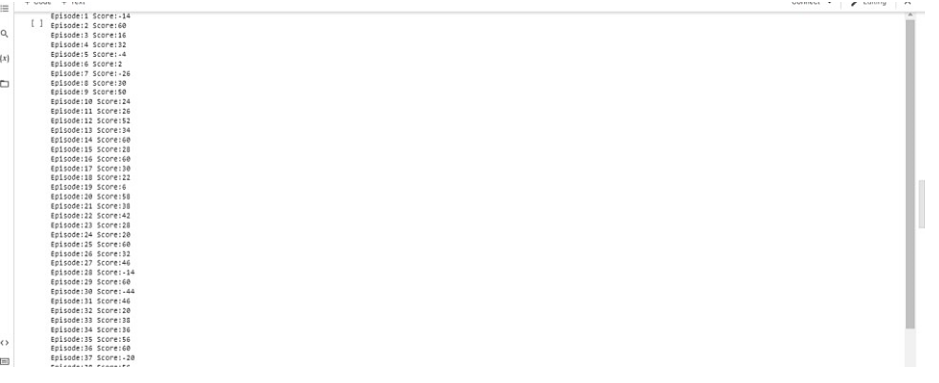
3.	Loss	<div><div>Plotting Loss Graph</div><div><pre>[] #Training and Validation Loss Plot plt.figure(figsize=(6,6)) plt.plot(epochs_range, history.history['loss'], label='Training Loss') plt.plot(epochs_range, history.history['val_loss'], label='Validation Loss') plt.legend() plt.title('Training and Testing Loss') plt.show()</pre></div><div></div></div>																																			
4.	Confusion Matrix	<div><pre>#Confution Matrix and Classification Report Y_pred = model.predict_generator(Xtest,500 // 100) y_pred = np.argmax(Y_pred, axis=1) print('Confusion Matrix') print(confusion_matrix(Xtest.classes, y_pred))</pre></div> <div>WARNING:tensorflow:Your input ran out of data; interrupting training. Make sure that your dataset</div> <div>Confusion Matrix</div> <div><pre>[[20 22 13] [16 22 12] [16 14 20]]</pre></div>																																			
5.	Classification Report	<div><div>Classification Report</div><div><pre>{x} [] print('Classification Report') target_names = ['Cyclone', 'Earthquake', 'Flood'] print(classification_report(Xtest.classes, y_pred, target_names=target_names))</pre></div><div><table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>Cyclone</td><td>0.38</td><td>0.36</td><td>0.37</td><td>55</td></tr><tr><td>Earthquake</td><td>0.38</td><td>0.44</td><td>0.41</td><td>50</td></tr><tr><td>Flood</td><td>0.44</td><td>0.40</td><td>0.42</td><td>50</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.40</td><td>155</td></tr><tr><td>macro avg</td><td>0.40</td><td>0.40</td><td>0.40</td><td>155</td></tr><tr><td>weighted avg</td><td>0.40</td><td>0.40</td><td>0.40</td><td>155</td></tr></tbody></table></div></div>		precision	recall	f1-score	support	Cyclone	0.38	0.36	0.37	55	Earthquake	0.38	0.44	0.41	50	Flood	0.44	0.40	0.42	50	accuracy			0.40	155	macro avg	0.40	0.40	0.40	155	weighted avg	0.40	0.40	0.40	155
	precision	recall	f1-score	support																																	
Cyclone	0.38	0.36	0.37	55																																	
Earthquake	0.38	0.44	0.41	50																																	
Flood	0.44	0.40	0.42	50																																	
accuracy			0.40	155																																	
macro avg	0.40	0.40	0.40	155																																	
weighted avg	0.40	0.40	0.40	155																																	

2) Feature-2 (Reinforcement Learning Model)

- For Earthquake

S.No.	Parameter	Screenshot
1.	Episode and Score	
2.	Evaluation Policy	

- For Cyclone and Flood

S.No.	Parameter	Screenshot
1.	Episode and Score	
2.	Evaluation Policy	