Project Development Phase
Model Performance Test

Date	16 November 2023
Team ID	Team- 591954
Project Name	Weather Classification Using Deep Learning
Maximum Marks	10 Marks

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

<table>
<tr><th>S.No.</th><th>Parameter</th><th>Values</th><th>Screenshot</th></tr>
<tr>
<td>1.</td>
<td>Model Summary</td>
<td>

Model Name: Weather classification using Deep Learning**.**

Description:
Deep learning techniques to classify weather patterns accurately. Through neural network architecture, it transforms diverse meteorological data into actionable insights, enhancing weather forecasting precision.

Architecture:
convolutional neural network (CNN)

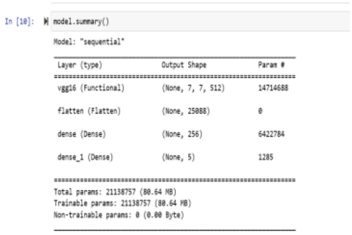
Layers:
VGG16, Sequential, Dense, Flatten.

Hyperparameters:
batch-size = 16

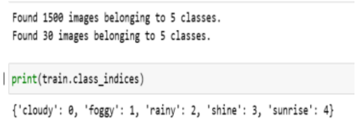
Training Data:
1500 images of five classes(cloudy,sunny,rainy,foggy, sunrise).

Validation Data size: 30 images of five classes.

</td>
<td>

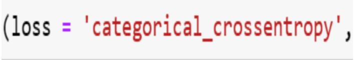
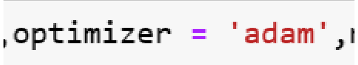
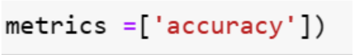


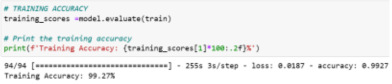
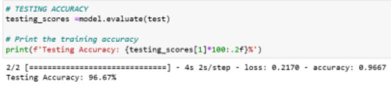
```
In [10]:  model.summary()

          Model: "sequential"
          _________________________________________________________________
           Layer (type)                Output Shape              Param #
          =================================================================
           vgg16 (Functional)          (None, 7, 7, 512)         14714688

           flatten (Flatten)           (None, 25088)             0

           dense (Dense)               (None, 256)               6422784

           dense_1 (Dense)             (None, 5)                 1285

          =================================================================
          Total params: 21138757 (80.64 MB)
          Trainable params: 21138757 (80.64 MB)
          Non-trainable params: 0 (0.00 Byte)
          _________________________________________________________________
```



```
Found 1500 images belonging to 5 classes.
Found 30 images belonging to 5 classes.

print(train.class_indices)

{'cloudy': 0, 'foggy': 1, 'rainy': 2, 'shine': 3, 'sunrise': 4}
```

</td>
</tr>
</table>

		Pre-trained Model: VGG 16
		Framework/Libraries: Tensorflow, Keras, Numpy, Pandas
		Loss Function: categorical cross-entropy
		Optimizer: adam
		Metrics: Accuracy
			
			`(loss = 'categorical_crossentropy',`
			
			`,optimizer = 'adam',`
			
			`metrics =['accuracy'])`

2.	Accuracy	Training Accuracy – 99.27
		Validation Accuracy – 96.67
			
			`# TRAINING ACCURACY`
			`training_scores =model.evaluate(train)`
			`# Print the training accuracy`
			`print(f'Training Accuracy: {training_scores[1]*100:.2f}%')`
			`94/94 [==============================] - 255s 3s/step - loss: 0.0187 - accuracy: 0.9927`
			`Training Accuracy: 99.27%`
			
			`# TESTING ACCURACY`
			`testing_scores =model.evaluate(test)`
			`# Print the training accuracy`
			`print(f'Testing Accuracy: {testing_scores[1]*100:.2f}%')`
			`2/2 [==============================] - 4s 2s/step - loss: 0.2170 - accuracy: 0.9667`
			`Testing Accuracy: 96.67%`