

Project Development Phase

Model Performance Test

Date	10 November 2022
Team ID	PNT2022TMID13815
Project Name	Emerging methods for Early Detection Of Forests Fires
Maximum Marks	10 Marks

Model Performance Testing:

S.No.	Parameter	Values	Screenshot
1.	Model Summary	-	<pre> Model: "sequential" Layer (type) Output Shape Param # ----- conv2d (Conv2D) (None, 148, 148, 32) 896 activation (Activation) (None, 148, 148, 32) 0 max_pooling2d (MaxPooling2D) (None, 74, 74, 32) 0 flatten (Flatten) (None, 175232) 0 dense (Dense) (None, 150) 26284950 activation_1 (Activation) (None, 150) 0 dropout (Dropout) (None, 150) 0 dense_1 (Dense) (None, 1) 151 activation_2 (Activation) (None, 1) 0 Total params: 26,285,997 Trainable params: 26,285,997 Non-trainable params: 0 </pre>
2.	Accuracy	<p>Training Accuracy – 93.39</p> <p>Validation Accuracy -93.39</p>	<pre> Epoch 12/30 10000/10000 [-----] 100 % 1s/0m - loss: 0.2520 - accuracy: 0.8000 - val_loss: 0.4400 - val_accuracy: 0.8000 Epoch 13/30 10000/10000 [-----] 100 % 1s/0m - loss: 0.2200 - accuracy: 0.8000 - val_loss: 0.4000 - val_accuracy: 0.8000 Epoch 14/30 10000/10000 [-----] 100 % 1s/0m - loss: 0.3400 - accuracy: 0.8000 - val_loss: 0.4000 - val_accuracy: 0.8000 Epoch 15/30 10000/10000 [-----] 100 % 1s/0m - loss: 0.3400 - accuracy: 0.8000 - val_loss: 0.4000 - val_accuracy: 0.8000 Epoch 16/30 10000/10000 [-----] 100 % 1s/0m - loss: 0.2500 - accuracy: 0.8000 - val_loss: 0.4000 - val_accuracy: 0.8000 Epoch 17/30 10000/10000 [-----] 100 % 1s/0m - loss: 0.2500 - accuracy: 0.8000 - val_loss: 0.4000 - val_accuracy: 0.8000 Epoch 18/30 10000/10000 [-----] 100 % 1s/0m - loss: 0.2500 - accuracy: 0.8000 - val_loss: 0.4000 - val_accuracy: 0.8000 Epoch 19/30 10000/10000 [-----] 100 % 1s/0m - loss: 0.2500 - accuracy: 0.8000 - val_loss: 0.4000 - val_accuracy: 0.8000 Epoch 20/30 10000/10000 [-----] 100 % 1s/0m - loss: 0.2500 - accuracy: 0.8000 - val_loss: 0.4000 - val_accuracy: 0.8000 Epoch 21/30 10000/10000 [-----] 100 % 1s/0m - loss: 0.2500 - accuracy: 0.8000 - val_loss: 0.4000 - val_accuracy: 0.8000 Epoch 22/30 10000/10000 [-----] 100 % 1s/0m - loss: 0.2500 - accuracy: 0.8000 - val_loss: 0.4000 - val_accuracy: 0.8000 Epoch 23/30 10000/10000 [-----] 100 % 1s/0m - loss: 0.2500 - accuracy: 0.8000 - val_loss: 0.4000 - val_accuracy: 0.8000 Epoch 24/30 10000/10000 [-----] 100 % 1s/0m - loss: 0.2500 - accuracy: 0.8000 - val_loss: 0.4000 - val_accuracy: 0.8000 Epoch 25/30 10000/10000 [-----] 100 % 1s/0m - loss: 0.2500 - accuracy: 0.8000 - val_loss: 0.4000 - val_accuracy: 0.8000 Epoch 26/30 10000/10000 [-----] 100 % 1s/0m - loss: 0.2500 - accuracy: 0.8000 - val_loss: 0.4000 - val_accuracy: 0.8000 Epoch 27/30 10000/10000 [-----] 100 % 1s/0m - loss: 0.2500 - accuracy: 0.8000 - val_loss: 0.4000 - val_accuracy: 0.8000 Epoch 28/30 10000/10000 [-----] 100 % 1s/0m - loss: 0.2500 - accuracy: 0.8000 - val_loss: 0.4000 - val_accuracy: 0.8000 Epoch 29/30 10000/10000 [-----] 100 % 1s/0m - loss: 0.2500 - accuracy: 0.8000 - val_loss: 0.4000 - val_accuracy: 0.8000 Epoch 30/30 10000/10000 [-----] 100 % 1s/0m - loss: 0.2500 - accuracy: 0.8000 - val_loss: 0.4000 - val_accuracy: 0.8000 Total params: 26,285,997 Trainable params: 26,285,997 Non-trainable params: 0 </pre>
3.	Confidence Score (Only Yolo Projects)	<p>Class Detected – Fire and No Danger</p> <p>Confidence Score – 93.067</p>	<pre> class DetectFire: def __init__(self, model_path, confidence_threshold): self.model_path = model_path self.confidence_threshold = confidence_threshold def detect_fire(self, image_path): # Load the image image = cv.imread(image_path) # Detect fire results = self.model.predict(image) # Get the class names and confidence scores class_names = results.names confidences = results.confidences # Find the index of the class with the highest confidence max_confidence_index = np.argmax(confidences) # Get the class name and confidence score class_name = class_names[max_confidence_index] confidence_score = confidences[max_confidence_index] # Print the results print(f"Class Detected: {class_name}") print(f"Confidence Score: {confidence_score}") # Return the results return class_name, confidence_score # Create an instance of the DetectFire class detect_fire = DetectFire("yolo_weights/best.pt", 0.5) # Detect fire in the image class_name, confidence_score = detect_fire.detect_fire("image.jpg") # Print the results print(f"Class Detected: {class_name}") print(f"Confidence Score: {confidence_score}") </pre> 