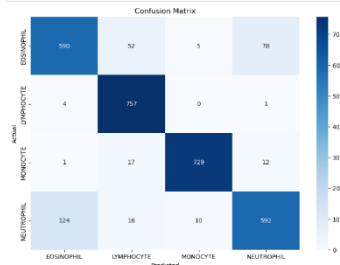
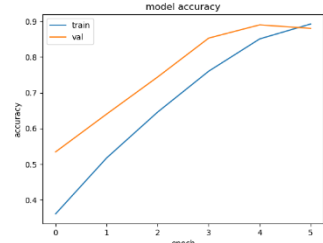


Project Development Phase Model Performance Test

Date	19 January 2026
Team ID	LTVIP2026TMIDS83701
Project Name	Smart Sorting: Transfer learning for identifying Rotten Fruits and Vegetables
Maximum Marks	10 Marks

Model Performance Testing:

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

S.No.	Parameter	Values	Screenshot																									
1.	Metrics	Classification Model: Accuracy Score- 0.893	<pre>import matplotlib.pyplot as plt import seaborn as sns from sklearn.metrics import confusion_matrix class_labels = ['EDSUNORMAL', 'LYMPHOCYTE', 'MONOCYTE', 'NEUTROPHIL'] cm = confusion_matrix(y_test, pred2) plt.figure(figsize=(10, 7)) sns.heatmap(cm, annot=True, fmt='g', vmin=0, vmax=5000) plt.xticks(rotation=45, labels=[0, 1, 2, 3], labels=[class_labels]) plt.yticks(rotation=45, labels=[0, 1, 2, 3], labels=[class_labels]) plt.xlabel('Predicted') plt.ylabel('Actual') plt.title('Confusion Matrix') plt.show()</pre>  <table><caption>Confusion Matrix Data</caption><thead><tr><th>Actual \ Predicted</th><th>EDSUNORMAL</th><th>LYMPHOCYTE</th><th>MONOCYTE</th><th>NEUTROPHIL</th></tr></thead><tbody><tr><th>EDSUNORMAL</th><td>100</td><td>52</td><td>5</td><td>78</td></tr><tr><th>LYMPHOCYTE</th><td>4</td><td>757</td><td>0</td><td>1</td></tr><tr><th>MONOCYTE</th><td>1</td><td>17</td><td>128</td><td>12</td></tr><tr><th>NEUTROPHIL</th><td>124</td><td>16</td><td>10</td><td>301</td></tr></tbody></table>	Actual \ Predicted	EDSUNORMAL	LYMPHOCYTE	MONOCYTE	NEUTROPHIL	EDSUNORMAL	100	52	5	78	LYMPHOCYTE	4	757	0	1	MONOCYTE	1	17	128	12	NEUTROPHIL	124	16	10	301
Actual \ Predicted	EDSUNORMAL	LYMPHOCYTE	MONOCYTE	NEUTROPHIL																								
EDSUNORMAL	100	52	5	78																								
LYMPHOCYTE	4	757	0	1																								
MONOCYTE	1	17	128	12																								
NEUTROPHIL	124	16	10	301																								
2.	Tune the Model	Hyperparameter Tuning - The notebook primarily focuses on training the added dense layers with a pre-trained MobileNetV2 model (frozen base layers). Adam optimizer was used with categorical crossentropy loss. The training ran for 15 epochs, with the best validation accuracy observed around epoch 11. Validation Method - A validation split of 0.2 was used during image data generation (validation_split=0.2).	<pre>pred = model.predict(test) pred = np.argmax(pred, axis=-1) #pick class with highest probability labels = (train.class_indices) labels = dict((v,k) for k,v in labels.items()) pred2 = [labels[k] for k in pred] 374/374 [.....] - 332s 880ms/step plt.plot(history.history['accuracy'], history.history['accuracy']) plt.plot(history.history['val_accuracy'], history.history['val_accuracy']) plt.title('model accuracy') plt.ylabel('accuracy') plt.xlabel('epoch') plt.legend(['train', 'val'], loc='upper left') plt.show()</pre>  <table><caption>Model Accuracy Data</caption><thead><tr><th>epoch</th><th>train</th><th>val</th></tr></thead><tbody><tr><td>0</td><td>0.45</td><td>0.55</td></tr><tr><td>1</td><td>0.55</td><td>0.65</td></tr><tr><td>2</td><td>0.65</td><td>0.75</td></tr><tr><td>3</td><td>0.75</td><td>0.85</td></tr><tr><td>4</td><td>0.85</td><td>0.90</td></tr><tr><td>5</td><td>0.90</td><td>0.95</td></tr></tbody></table>	epoch	train	val	0	0.45	0.55	1	0.55	0.65	2	0.65	0.75	3	0.75	0.85	4	0.85	0.90	5	0.90	0.95				
epoch	train	val																										
0	0.45	0.55																										
1	0.55	0.65																										
2	0.65	0.75																										
3	0.75	0.85																										
4	0.85	0.90																										
5	0.90	0.95																										