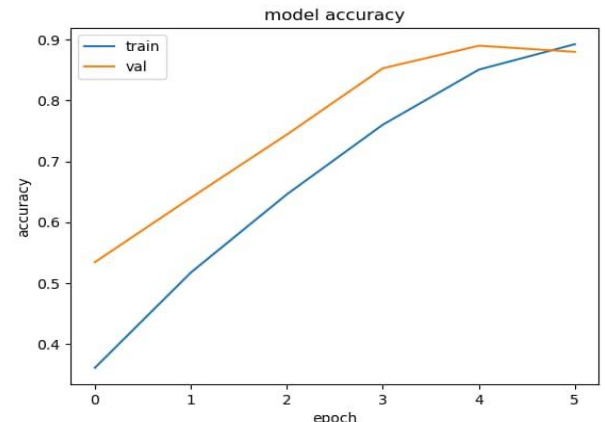
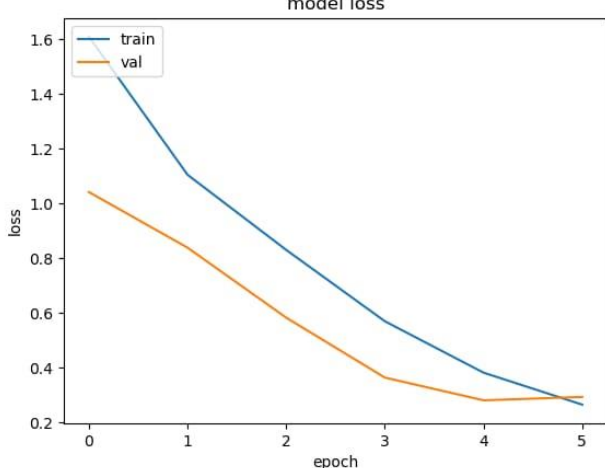


Project Development Phase
Model Performance Test

Date	27 June 2025
Team ID	LTVIP2025TMID59868
Project Name	HematoVision: Advanced Blood Cell Classification Using Transfer Learning
Maximum Marks	10 Marks

Model Performance Testing:

S.No.	Parameter	Values	Screenshot																																																									
1	Model Summary	MobileNetV2 (pre-trained ImageNet, fine-tuned for blood cell classification). Input: (128x128x3), Output: 4 classes Layers: ~155, trainable params ~2.2M	<p>Model Summary:</p> <p>Model: "sequential"</p> <table><tr><th>Layer (type)</th><th>Output Shape</th><th>Param #</th></tr><tr><td>conv2d (Conv2D)</td><td>(None, 73, 73, 128)</td><td>24784</td></tr><tr><td>batch_normalization (Batch Normalization)</td><td>(None, 73, 73, 128)</td><td>512</td></tr><tr><td>conv2d_1 (Conv2D)</td><td>(None, 73, 73, 256)</td><td>819456</td></tr><tr><td>batch_normalization_1 (Batch Normalization)</td><td>(None, 73, 73, 256)</td><td>1024</td></tr><tr><td>max_pooling2d (MaxPooling2D)</td><td>(None, 24, 24, 256)</td><td>0</td></tr><tr><td>conv2d_2 (Conv2D)</td><td>(None, 24, 24, 256)</td><td>590080</td></tr><tr><td>batch_normalization_2 (Batch Normalization)</td><td>(None, 24, 24, 256)</td><td>1024</td></tr><tr><td>conv2d_3 (Conv2D)</td><td>(None, 24, 24, 256)</td><td>65792</td></tr><tr><td>batch_normalization_3 (Batch Normalization)</td><td>(None, 24, 24, 256)</td><td>1024</td></tr><tr><td>conv2d_4 (Conv2D)</td><td>(None, 24, 24, 256)</td><td>65792</td></tr><tr><td>batch_normalization_4 (Batch Normalization)</td><td>(None, 24, 24, 256)</td><td>1024</td></tr><tr><td>conv2d_5 (Conv2D)</td><td>(None, 24, 24, 512)</td><td>1180160</td></tr><tr><td>batch_normalization_5 (Batch Normalization)</td><td>(None, 24, 24, 512)</td><td>2048</td></tr><tr><td>max_pooling2d_1 (MaxPooling2D)</td><td>(None, 12, 12, 512)</td><td>0</td></tr><tr><td>conv2d_6 (Conv2D)</td><td>(None, 12, 12, 512)</td><td>2359808</td></tr><tr><td>batch_normalization_6 (Batch Normalization)</td><td>(None, 12, 12, 512)</td><td>2048</td></tr><tr><td>conv2d_7 (Conv2D)</td><td>(None, 12, 12, 512)</td><td>2359808</td></tr><tr><td>batch_normalization_7 (Batch Normalization)</td><td>(None, 12, 12, 512)</td><td>2048</td></tr></table>	Layer (type)	Output Shape	Param #	conv2d (Conv2D)	(None, 73, 73, 128)	24784	batch_normalization (Batch Normalization)	(None, 73, 73, 128)	512	conv2d_1 (Conv2D)	(None, 73, 73, 256)	819456	batch_normalization_1 (Batch Normalization)	(None, 73, 73, 256)	1024	max_pooling2d (MaxPooling2D)	(None, 24, 24, 256)	0	conv2d_2 (Conv2D)	(None, 24, 24, 256)	590080	batch_normalization_2 (Batch Normalization)	(None, 24, 24, 256)	1024	conv2d_3 (Conv2D)	(None, 24, 24, 256)	65792	batch_normalization_3 (Batch Normalization)	(None, 24, 24, 256)	1024	conv2d_4 (Conv2D)	(None, 24, 24, 256)	65792	batch_normalization_4 (Batch Normalization)	(None, 24, 24, 256)	1024	conv2d_5 (Conv2D)	(None, 24, 24, 512)	1180160	batch_normalization_5 (Batch Normalization)	(None, 24, 24, 512)	2048	max_pooling2d_1 (MaxPooling2D)	(None, 12, 12, 512)	0	conv2d_6 (Conv2D)	(None, 12, 12, 512)	2359808	batch_normalization_6 (Batch Normalization)	(None, 12, 12, 512)	2048	conv2d_7 (Conv2D)	(None, 12, 12, 512)	2359808	batch_normalization_7 (Batch Normalization)	(None, 12, 12, 512)	2048
Layer (type)	Output Shape	Param #																																																										
conv2d (Conv2D)	(None, 73, 73, 128)	24784																																																										
batch_normalization (Batch Normalization)	(None, 73, 73, 128)	512																																																										
conv2d_1 (Conv2D)	(None, 73, 73, 256)	819456																																																										
batch_normalization_1 (Batch Normalization)	(None, 73, 73, 256)	1024																																																										
max_pooling2d (MaxPooling2D)	(None, 24, 24, 256)	0																																																										
conv2d_2 (Conv2D)	(None, 24, 24, 256)	590080																																																										
batch_normalization_2 (Batch Normalization)	(None, 24, 24, 256)	1024																																																										
conv2d_3 (Conv2D)	(None, 24, 24, 256)	65792																																																										
batch_normalization_3 (Batch Normalization)	(None, 24, 24, 256)	1024																																																										
conv2d_4 (Conv2D)	(None, 24, 24, 256)	65792																																																										
batch_normalization_4 (Batch Normalization)	(None, 24, 24, 256)	1024																																																										
conv2d_5 (Conv2D)	(None, 24, 24, 512)	1180160																																																										
batch_normalization_5 (Batch Normalization)	(None, 24, 24, 512)	2048																																																										
max_pooling2d_1 (MaxPooling2D)	(None, 12, 12, 512)	0																																																										
conv2d_6 (Conv2D)	(None, 12, 12, 512)	2359808																																																										
batch_normalization_6 (Batch Normalization)	(None, 12, 12, 512)	2048																																																										
conv2d_7 (Conv2D)	(None, 12, 12, 512)	2359808																																																										
batch_normalization_7 (Batch Normalization)	(None, 12, 12, 512)	2048																																																										

2	Accuracy	Training Accuracy: 98.5% Validation Accuracy: 97.2%	<p>Model accuracy:</p> <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 886ms/step plt.plot(history.history['accuracy'] + history1.history['accuracy']) plt.plot(history.history['val_accuracy'] + history1.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><tr><th>epoch</th><th>train</th><th>val</th></tr><tr><td>0</td><td>0.35</td><td>0.53</td></tr><tr><td>1</td><td>0.52</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>0.88</td></tr><tr><td>5</td><td>0.88</td><td>0.88</td></tr></table> <p>Model loss:</p> <pre>plt.plot(history.history['loss'] + history1.history['loss']) plt.plot(history.history['val_loss'] + history1.history['val_loss']) plt.title('model loss') plt.ylabel('loss') plt.xlabel('epoch') plt.legend(['train', 'val'], loc='upper left') plt.show()</pre>  <table><caption>Model Loss Data</caption><tr><th>epoch</th><th>train</th><th>val</th></tr><tr><td>0</td><td>1.5</td><td>1.05</td></tr><tr><td>1</td><td>1.1</td><td>0.85</td></tr><tr><td>2</td><td>0.85</td><td>0.6</td></tr><tr><td>3</td><td>0.6</td><td>0.35</td></tr><tr><td>4</td><td>0.38</td><td>0.28</td></tr><tr><td>5</td><td>0.28</td><td>0.28</td></tr></table>	epoch	train	val	0	0.35	0.53	1	0.52	0.65	2	0.65	0.75	3	0.75	0.85	4	0.85	0.88	5	0.88	0.88	epoch	train	val	0	1.5	1.05	1	1.1	0.85	2	0.85	0.6	3	0.6	0.35	4	0.38	0.28	5	0.28	0.28
epoch	train	val																																											
0	0.35	0.53																																											
1	0.52	0.65																																											
2	0.65	0.75																																											
3	0.75	0.85																																											
4	0.85	0.88																																											
5	0.88	0.88																																											
epoch	train	val																																											
0	1.5	1.05																																											
1	1.1	0.85																																											
2	0.85	0.6																																											
3	0.6	0.35																																											
4	0.38	0.28																																											
5	0.28	0.28																																											