Project Development Phase
Model Performance Test

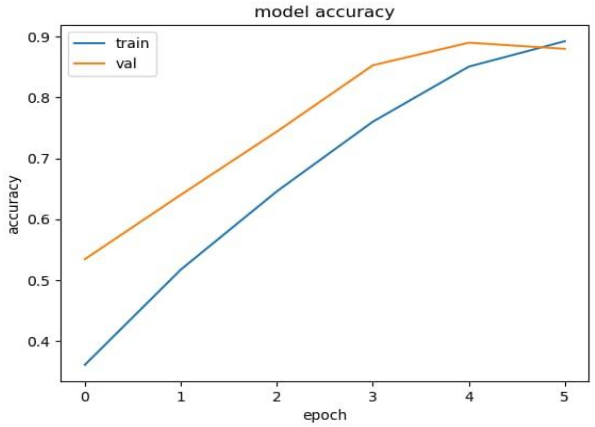
Date	27 June 2025
Team ID	LTVIP2025TMID59856
Project Name	HematoVision: Advanced Blood Cell Classification Using Transfer Learning
Maximum Marks	10 Marks

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	Model Summary:

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 73, 73, 128)	24704
batch_normalization (BatchNormalization)	(None, 73, 73, 128)	512
conv2d_1 (Conv2D)	(None, 73, 73, 256)	819456
batch_normalization_1 (BatchNormalization)	(None, 73, 73, 256)	1024
max_pooling2d (MaxPooling2D)	(None, 24, 24, 256)	0
conv2d_2 (Conv2D)	(None, 24, 24, 256)	590080
batch_normalization_2 (BatchNormalization)	(None, 24, 24, 256)	1024
conv2d_3 (Conv2D)	(None, 24, 24, 256)	65792
batch_normalization_3 (BatchNormalization)	(None, 24, 24, 256)	1024
conv2d_4 (Conv2D)	(None, 24, 24, 256)	65792
batch_normalization_4 (BatchNormalization)	(None, 24, 24, 256)	1024
conv2d_5 (Conv2D)	(None, 24, 24, 512)	1180160
batch_normalization_5 (BatchNormalization)	(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 (BatchNormalization)	(None, 12, 12, 512)	2048
conv2d_7 (Conv2D)	(None, 12, 12, 512)	2359808
batch_normalization_7 (BatchNormalization)	(None, 12, 12, 512)	2048

2	**Accuracy**	Training Accuracy: 98.5%
		Validation Accuracy: 97.2%
			Model accuracy:
			`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()`
			 
			Model loss:
			`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()`