

## Project Development Phase Model Performance Test

Date	19 Feb 2026
Team ID	LTVIP2026TMIDS66121
Project Name	Advanced Blood Cell Classification Using Transfer Learning
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	<b>Classification Model:</b> Accuracy Score- 0.893	<pre>import numpy as np import random as rnd from sklearn.metrics import confusion_matrix  class_labels = ['EOSINOPHIL', 'LYMPHOCYTE', 'MONOCYTE', 'NEUTROPHIL']  cm = confusion_matrix(y_test, pred)  plt.figure(figsize=(10, 7)) sns.heatmap(cm, annot=True, fmt='g', vmin=0, cmap='Blues') plt.xticks(ticks=[0, 1, 2, 3], labels=train_labels) plt.yticks(ticks=[0, 1, 2, 3], labels=test_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>EOSINOPHIL</th><th>LYMPHOCYTE</th><th>MONOCYTE</th><th>NEUTROPHIL</th></tr></thead><tbody><tr><th>EOSINOPHIL</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>27</td><td>729</td><td>12</td></tr><tr><th>NEUTROPHIL</th><td>124</td><td>18</td><td>10</td><td>761</td></tr></tbody></table>	Actual \ Predicted	EOSINOPHIL	LYMPHOCYTE	MONOCYTE	NEUTROPHIL	EOSINOPHIL	100	52	5	78	LYMPHOCYTE	4	757	0	1	MONOCYTE	1	27	729	12	NEUTROPHIL	124	18	10	761
Actual \ Predicted	EOSINOPHIL	LYMPHOCYTE	MONOCYTE	NEUTROPHIL																								
EOSINOPHIL	100	52	5	78																								
LYMPHOCYTE	4	757	0	1																								
MONOCYTE	1	27	729	12																								
NEUTROPHIL	124	18	10	761																								
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 [.....] - 3321 886ms/step  plt.plot(history.history['accuracy'] + history.history['val_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>0.40</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>0.90</td></tr></tbody></table>	epoch	train	val	0	0.40	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	0.90				
epoch	train	val																										
0	0.40	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	0.90																										