

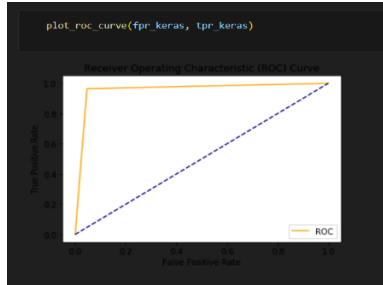

## Model Optimization and Tuning Phase Template

|               |  |
|---------------|--|
| Date          | 15 October 2024  |
| Team ID       | LTVIP2024TMID24772   |
| Project Title | Implementation of Deep Learning Techniques to Detect Malaria |
| Maximum Marks | 10 Marks   |

### Model Optimization and Tuning Phase

The Model Optimization and Tuning Phase involves refining neural network models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency.

### Hyperparameter Tuning Documentation (8 Marks):

| Model | Tuned Hyperparameters   |
|-------|---|
| CNN   | <div> <pre>from numpy import loadmat from keras.models import load_model model = load_model('ml_012.h5')  scores=model.evaluate(x_test,y_test) print(scores)  2755/2755 [=====] - 0s 127us/step [0.1328830818349864, 0.9586206896551724]  accuracy = model.evaluate(x_test, y_test, verbose=1) print('test accuracy: ', accuracy[1])  2755/2755 [=====] - 0s 87us/step Test_Accuracy:- 0.9586206896551724</pre> </div> <div>  <p>Receiver Operating Characteristic (ROC) Curve</p> </div> <div> <pre>print('{}'.format(     precision    recall  f1-score   support  0     0.97    0.95    0.96    1432 1     0.95    0.96    0.96    1323  accuracy          0.96 macro avg         0.96    0.96    0.96    2755 weighted avg      0.96    0.96    0.96    2755</pre> </div> <div> <pre>CM = confusion_matrix(y_true, pred) from mlxtend.plotting import plot_confusion_matrix fig, ax = plot_confusion_matrix(conf_mat=CM, figsize=(5, 5)) plt.show()</pre>  <p>Confusion Matrix values: True Negatives = 1297, False Negatives = 66, False Positives = 30, True Positives = 1362</p> </div> |

### Final Model Selection Justification (2 Marks):

| Final Model   | Reasoning  |
|---------------|--|
| Model 1 (CNN) | <p>The CNN model was chosen as the final optimized model due to its proven effectiveness in image classification tasks, particularly in medical imaging. Its architecture allows for automatic feature extraction, which is crucial for accurately identifying malaria-infected cells in blood smear images. The model's performance was further enhanced through hyperparameter tuning, including adjustments to the learning rate, batch size, dropout rate, and number of epochs. Additionally, the model showed high accuracy during validation, demonstrating its capability to generalize well to unseen data. The use of techniques such as data augmentation and one-hot encoding also contributed to reducing overfitting and improving classification performance. Overall, the CNN's robust architecture and favorable training metrics solidified its selection as the final model for this project.</p> |

### Explanation:

- **Final Model:** Specifies the model that has been selected as the final version for deployment or reporting.

