

## Model Optimization and Tuning Phase Template

Date	11 July 2024
Team ID	SWTID1720011518
Project Title	WCE curated Colon Disease Classification using Deep Learning
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
VGG16	<pre> tuner.search(train, epochs=20, callbacks=[stop_early])  # Get the optimal hyperparameters best_hps=tuner.get_best_hyperparameters(num_trials=1)[0]  print(f""" The hyperparameter search is complete. The optimal learning rate for the optimizer is {best_hps.get('learning_rate')}). """)  Trial 3 Complete [00h 02m 00s] accuracy: 0.9546874761581421  Best accuracy So Far: 0.9740625023841858 Total elapsed time: 00h 06m 03s  The hyperparameter search is complete. The optimal learning rate for the optimizer is 0.001.  model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=hp_learning_rate),               loss=tf.keras.losses.CategoricalCrossentropy(),               metrics=['accuracy'])  return model </pre>

**Loss Function: loss='categorical\_crossentropy'** - Determines the error between predicted and actual output for multi-class classification tasks.

**Optimizer: optimizer=Adam(learning\_rate=0.01)** - Adam optimizer with a learning rate of 0.01, adjusting the step size during training for better convergence.

**Metrics: metrics=['accuracy']** - Evaluation metric to measure the proportion of correctly classified examples out of the total during training and testing.

No. of epochs = 4

```

▶ model = tuner.hypermodel.build(best_hps)
  history6 = model.fit(train, epochs=4)

  acc_per_epoch = history6.history['accuracy']
  best_epoch = acc_per_epoch.index(max(acc_per_epoch)) + 1
  print('Best epoch: %d' % (best_epoch,))

```

```

Epoch 1/4
160/160 [=====] - 59s 365ms/step - loss: 0.2693 - accuracy: 0.9081
Epoch 2/4
160/160 [=====] - 60s 372ms/step - loss: 0.0733 - accuracy: 0.9741
Epoch 3/4
160/160 [=====] - 58s 359ms/step - loss: 0.0413 - accuracy: 0.9887
Epoch 4/4
160/160 [=====] - 59s 368ms/step - loss: 0.0373 - accuracy: 0.9881
Best epoch: 3

```

But we found best epoch was 3 so went with 3 epochs.

We found the accuracy to be 98.91 % ~ 99%

Resnet50

```

resnet.compile(loss = "categorical_crossentropy", optimizer = "adam", metrics = ["accuracy"])

[ ] history3 = resnet.fit(train , validation_data = test , epochs = 5)

```

```

Epoch 1/5
160/160 [=====] - 91s 528ms/step - loss: 1.6670 - accuracy: 0.5344 - val_loss: 1.5158 - val_accuracy: 0.7084
Epoch 2/5
160/160 [=====] - 84s 526ms/step - loss: 0.8868 - accuracy: 0.6953 - val_loss: 0.2883 - val_accuracy: 0.8909
Epoch 3/5
160/160 [=====] - 100s 627ms/step - loss: 0.5234 - accuracy: 0.7997 - val_loss: 0.7353 - val_accuracy: 0.7619
Epoch 4/5
160/160 [=====] - 86s 535ms/step - loss: 0.5683 - accuracy: 0.7900 - val_loss: 0.2721 - val_accuracy: 0.8956
Epoch 5/5
160/160 [=====] - 84s 526ms/step - loss: 0.6883 - accuracy: 0.7763 - val_loss: 0.3258 - val_accuracy: 0.8647

```

	<p><b>Loss Function: loss='categorical_crossentropy'</b> - Determines the error between predicted and actual output for multi-class classification tasks.</p> <p><b>Optimizer: optimizer=Adam</b> - Adam optimizer with a learning rate of 0.01, adjusting the step size during training for better convergence.</p> <p><b>Metrics: metrics=['accuracy']</b> - Evaluation metric to measure the proportion of correctly classified examples out of the total during training and testing.</p> <p>For 5 epochs we found the best accuracy to be 77.63% ~ 78%</p>
InceptionV3	<pre><code>inception.compile(loss = "categorical_crossentropy", optimizer = "adam", metrics = ["accuracy"])</code></pre> <p><b>Loss Function: loss='categorical_crossentropy'</b> - Determines the error between predicted and actual output for multi-class classification tasks.</p> <p><b>Optimizer: optimizer=Adam</b> - Adam optimizer with a learning rate of 0.01, adjusting the step size during training for better convergence.</p> <p><b>Metrics: metrics=['accuracy']</b> - Evaluation metric to measure the proportion of correctly classified examples out of the total during training and testing.</p> <p>For 5 epochs we found the best accuracy to be 97.63% ~ 98%</p>

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

Final Model	Reasoning
<b>VGG16</b>	<p>The decision to select VGG16 as the final optimized model for colon disease classification was based on its robust capability to extract relevant features from raw image data and its accuracy around 99%.</p> <p>We have built VGG16, INCEPTION V3 and RESNET50 out of which VGG16 had the highest accuracy. To refine its performance and ensure robustness, the VGG16 model underwent further optimization through techniques such as hyperparameter tuning. These enhancements were crucial in enhancing the model's accuracy, reliability, and generalizability for accurately diagnosing colon diseases from image data. Following is the graph which will help visualize the accuracies of the 3 models</p>

