Model Development Phase Template

Date	9 July 2024
Team ID	SWTID1720043892
Project Title	WCE Curated Colon Disease Using Deep Learning
Maximum Marks	10 Marks

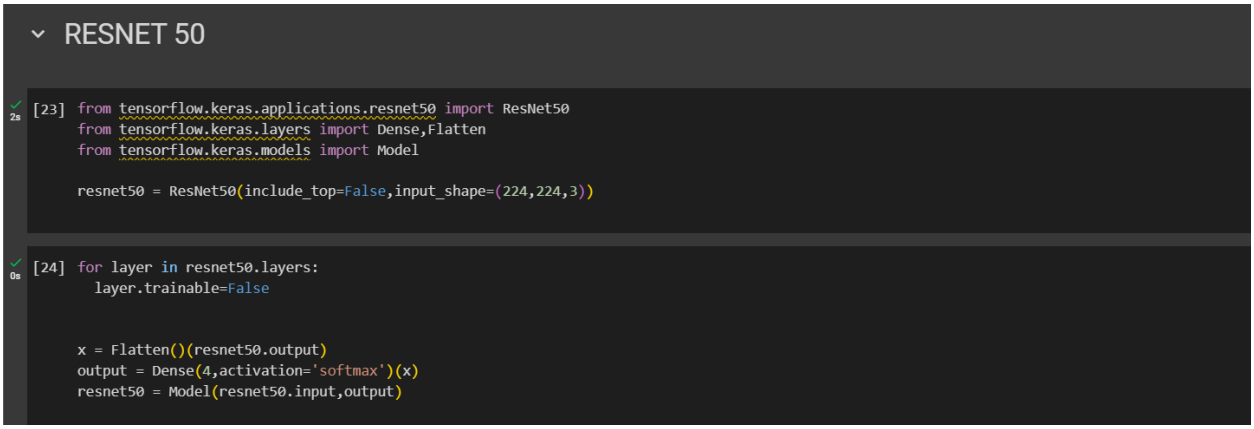
Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include a summary and training and validation performance metrics for multiple models, presented through respective screenshots.

Initial Model Training Code (5 marks):

Paste the screenshot of the model training code

RESNET50 –



```
import os

from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
base_path = '/content'
model_save_path = os.path.join(base_path, "model/resnet50_model.h5")
##model.fit(train_generator, validation_data=validation_generator, epochs=2)
# Define callbacks
early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)
checkpoint = ModelCheckpoint(model_save_path, monitor='val_loss', save_best_only=True, verbose=1)

# Compile the model
resnet50.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Train the model with callbacks
history = resnet50.fit(
    train,
    validation_data=test,
    epochs=50,
    callbacks=[early_stopping, checkpoint]
)

# Print the model save path
print(f"Model saved at: {model_save_path}")

# Save the model
os.makedirs(os.path.dirname(model_save_path), exist_ok=True)
resnet50.save(model_save_path)
```

VGG16 –

```
from tensorflow.keras.applications.vgg16 import VGG16
from tensorflow.keras.layers import Dense,Flatten
from tensorflow.keras.models import Model

vgg = VGG16(include_top=False,input_shape=(224,224,3))
vgg.summary()
```

```
for layer in vgg.layers:
  print(layer)
```

```
[18] for layer in vgg.layers:
       layer.trainable = False
     x = Flatten()(vgg.output)
     output = Dense(4,activation='softmax')(x)
     vgg16 = Model(vgg.input,output)
     vgg16.summary()
```

```
import os

from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
base_path = '/content'
model_save_path = os.path.join(base_path, "model/vgg16_model.h5")
##model.fit(train_generator, validation_data=validation_generator, epochs=2)
# Define callbacks
early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)
checkpoint = ModelCheckpoint(model_save_path, monitor='val_loss', save_best_only=True, verbose=1)

# Compile the model
vgg16.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Train the model with callbacks
history = vgg16.fit(
    train,
    validation_data=test,
    epochs=50,
    callbacks=[early_stopping, checkpoint]
)

# Print the model save path
print(f"Model saved at: {model_save_path}")

# Save the model
os.makedirs(os.path.dirname(model_save_path), exist_ok=True)
vgg16.save(model_save_path)
```

InceptionV3 Model –

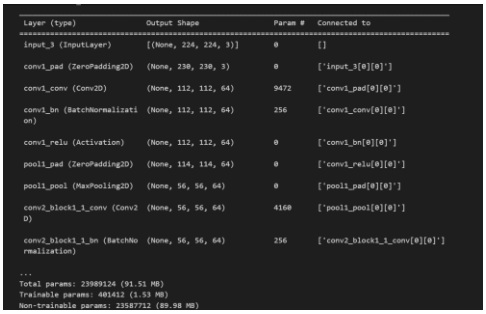
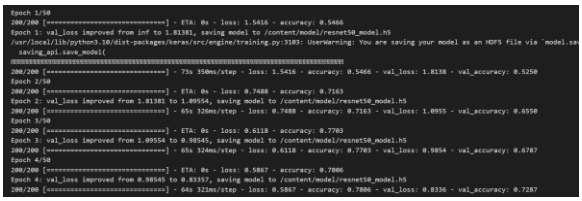
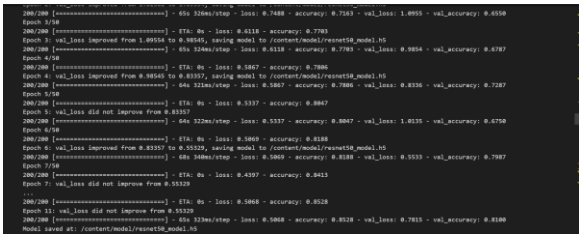
```
from tensorflow.keras.applications.inception_v3 import InceptionV3
from tensorflow.keras.layers import Dense,Flatten
from tensorflow.keras.models import Model

InceptionV3 = InceptionV3(include_top=False,input_shape=(224 ,224,3))
```

```
for layer in InceptionV3.layers:
  print(layer)
```

```
x = Flatten()(InceptionV3.output)
output = Dense(4,activation='softmax')(x)
InceptionV3 = Model(InceptionV3.input,output)
InceptionV3.summary()
```

```
InceptionV3.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
InceptionV3.fit(train,validation_data=test,epochs=5)

import os

from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
base_path = '/content'
model_save_path = os.path.join(base_path, "model/InceptionV3_model.h5")
##model.fit(train_generator, validation_data=validation_generator, epochs=2)
# Define callbacks
early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)
checkpoint = ModelCheckpoint(model_save_path, monitor='val_loss', save_best_only=True, verbose=1)

# Compile the model
InceptionV3.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Train the model with callbacks
history = InceptionV3.fit(
    train,
    validation_data=test,
    epochs=50,
    callbacks=[early_stopping, checkpoint]
)

# Print the model save path
print(f"Model saved at: {model_save_path}")

# Save the model
os.makedirs(os.path.dirname(model_save_path), exist_ok=True)
InceptionV3.save(model_save_path)
```

Model Validation and Evaluation Report (5 marks):

Model	Summary	Training and Validation Performance Metrics
RESNET 50 –		 

<table>
<tr>
<td>VGG16</td>
<td>

```
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 224, 224, 3)]     0

block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792

block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928

block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0

block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856

block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584

block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0

block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168

block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080

block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080

block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0
...
Total params: 14815044 (56.51 MB)
Trainable params: 100356 (392.02 KB)
Non-trainable params: 14714688 (56.13 MB)
```

</td>
<td>

```
Epoch 1/50
200/200 [==============================] - ETA: 0s - loss: 0.2686 - accuracy: 0.9016
Epoch 1: val_loss improved from inf to 0.32143, saving model to /content/model/vgg16_model.h5
200/200 [==============================] - 78s 323ms/step - loss: 0.2686 - accuracy: 0.9016 - val_loss: 0.3214 - val_accuracy: 0.8737
Epoch 2/50
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103: UserWarning: You are saving your model as an HDF5 file via `model.save()`. T
  saving_api.save_model(
200/200 [==============================] - ETA: 0s - loss: 0.0740 - accuracy: 0.9734
Epoch 2: val_loss did not improve from 0.32143
200/200 [==============================] - 66s 329ms/step - loss: 0.0740 - accuracy: 0.9734 - val_loss: 0.7599 - val_accuracy: 0.7862
Epoch 3/50
200/200 [==============================] - ETA: 0s - loss: 0.0443 - accuracy: 0.9872
Epoch 3: val_loss improved from 0.32143 to 0.20602, saving model to /content/model/vgg16_model.h5
200/200 [==============================] - 64s 321ms/step - loss: 0.0443 - accuracy: 0.9872 - val_loss: 0.2060 - val_accuracy: 0.9150
Epoch 4/50
200/200 [==============================] - ETA: 0s - loss: 0.0622 - accuracy: 0.9747
Epoch 4: val_loss did not improve from 0.20602
200/200 [==============================] - 64s 321ms/step - loss: 0.0622 - accuracy: 0.9747 - val_loss: 0.7836 - val_accuracy: 0.7950
Epoch 5/50
200/200 [==============================] - ETA: 0s - loss: 0.0296 - accuracy: 0.9900
Epoch 5: val_loss did not improve from 0.20602
200/200 [==============================] - 65s 325ms/step - loss: 0.0296 - accuracy: 0.9900 - val_loss: 1.2455 - val_accuracy: 0.7700
Epoch 6/50
200/200 [==============================] - ETA: 0s - loss: 0.0275 - accuracy: 0.9897
Epoch 6: val_loss did not improve from 0.20602
200/200 [==============================] - 64s 320ms/step - loss: 0.0275 - accuracy: 0.9897 - val_loss: 0.3648 - val_accuracy: 0.8712
Epoch 7/50
200/200 [==============================] - ETA: 0s - loss: 0.0292 - accuracy: 0.9891
Epoch 7: val_loss did not improve from 0.20602
200/200 [==============================] - 64s 318ms/step - loss: 0.0292 - accuracy: 0.9891 - val_loss: 1.0586 - val_accuracy: 0.7788
Epoch 8/50
200/200 [==============================] - ETA: 0s - loss: 0.0220 - accuracy: 0.9925
Epoch 8: val_loss did not improve from 0.20602
200/200 [==============================] - 69s 345ms/step - loss: 0.0220 - accuracy: 0.9925 - val_loss: 1.0403 - val_accuracy: 0.7912
Model saved at: /content/model/vgg16_model.h5
```

</td>
</tr>
<tr>
<td>Inception V3 Model</td>
<td>

```
Model: "model_4"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_4 (InputLayer)            [(None, 224, 224, 3)]  0         []

conv2d (Conv2D)                 (None, 111, 111, 32)  864        ['input_4[0][0]']

batch_normalization (Batch      (None, 111, 111, 32)  96         ['conv2d[0][0]']
Normalization)

activation (Activation)         (None, 111, 111, 32)  0          ['batch_normalization[0][0]']

conv2d_1 (Conv2D)               (None, 109, 109, 32)  9216       ['activation[0][0]']

batch_normalization_1 (Bat      (None, 109, 109, 32)  96         ['conv2d_1[0][0]']
chNormalization)

activation_1 (Activation)       (None, 109, 109, 32)  0          ['batch_normalization_1[0][0]'
                                                                 ]

conv2d_2 (Conv2D)               (None, 109, 109, 64)  18432      ['activation_1[0][0]']

batch_normalization_2 (Bat      (None, 109, 109, 64)  192        ['conv2d_2[0][0]']
chNormalization)
...
Total params: 22007580 (83.95 MB)
Trainable params: 21973156 (83.82 MB)
```

</td>
<td>

```
Epoch 1/5
200/200 [==============================] - 111s 345ms/step - loss: 0.9464 - accuracy: 0.8166 - val_loss: 808537.1250 - val_accuracy: 0.2500
Epoch 2/5
200/200 [==============================] - 69s 345ms/step - loss: 1.4173 - accuracy: 0.7303 - val_loss: 0.9928 - val_accuracy: 0.7538
Epoch 3/5
200/200 [==============================] - 67s 337ms/step - loss: 0.3797 - accuracy: 0.8672 - val_loss: 0.9083 - val_accuracy: 0.7387
Epoch 4/5
200/200 [==============================] - 68s 337ms/step - loss: 0.3335 - accuracy: 0.8975 - val_loss: 5.1727 - val_accuracy: 0.6237
Epoch 5/5
200/200 [==============================] - 68s 341ms/step - loss: 0.4610 - accuracy: 0.8813 - val_loss: 0.5140 - val_accuracy: 0.7412
Epoch 1/50
200/200 [==============================] - ETA: 0s - loss: 0.4691 - accuracy: 0.8831
Epoch 1: val_loss improved from inf to 0.59138, saving model to /content/model/Inception3_model.h5
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103: UserWarning: You are saving your model as an HDF5 file via `model.save()`. T
  saving_api.save_model(
==================================================================================================
200/200 [==============================] - 98s 352ms/step - loss: 0.4691 - accuracy: 0.8831 - val_loss: 0.5914 - val_accuracy: 0.7650
Epoch 2/50
200/200 [==============================] - ETA: 0s - loss: 0.6425 - accuracy: 0.8684
Epoch 2: val_loss did not improve from 0.59138
200/200 [==============================] - 69s 345ms/step - loss: 0.6425 - accuracy: 0.8684 - val_loss: 3.7434 - val_accuracy: 0.5587
Epoch 3/50
200/200 [==============================] - ETA: 0s - loss: 0.2654 - accuracy: 0.9050
Epoch 3: val_loss did not improve from 0.59138
200/200 [==============================] - 68s 341ms/step - loss: 0.2654 - accuracy: 0.9050 - val_loss: 0.8989 - val_accuracy: 0.7337
```

</td>
<td>

```
Epoch 4/50
200/200 [==============================] - ETA: 0s - loss: 0.2561 - accuracy: 0.9181
Epoch 4: val_loss did not improve from 0.59138
200/200 [==============================] - 69s 342ms/step - loss: 0.2561 - accuracy: 0.9181 - val_loss: 0.6276 - val_accuracy: 0.7412
Epoch 5/50
200/200 [==============================] - ETA: 0s - loss: 0.5895 - accuracy: 0.8800
Epoch 5: val_loss did not improve from 0.59138
200/200 [==============================] - 72s 362ms/step - loss: 0.5895 - accuracy: 0.8800 - val_loss: 1.0470 - val_accuracy: 0.7500
Epoch 6/50
200/200 [==============================] - ETA: 0s - loss: 0.5622 - accuracy: 0.8800
Epoch 6: val_loss did not improve from 0.59138
200/200 [==============================] - 68s 341ms/step - loss: 0.5622 - accuracy: 0.8800 - val_loss: 1.4380 - val_accuracy: 0.7400
Model saved at: /content/model/Inception3_model.h5
```

</td>
</tr>
</table>