Model Development Phase Template

Date	15 March 2024
Team ID	SWTID1720351492
Project Title	CovidVision: Advanced COVID-19 Detection from Lung X-rays with 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):

```python
def build_model():
    base_model = VGG19(weights='imagenet', include_top=False, input_shape=(224, 224, 3))

    for layer in base_model.layers:
        layer.trainable = False

    flatten_layer = Flatten()(base_model.output)
    dense_layer1 = Dense(1024, activation='relu')(flatten_layer)
    dense_layer2 = Dense(512, activation='relu')(dense_layer1)
    output_layer = Dense(1, activation='sigmoid')(dense_layer2)

    model = Model(inputs=base_model.input, outputs=output_layer)
    return model
```
[23] Python

```python
model = build_model()
```
Python

```python
model.summary()
```
Python

```python
plot_model(model, to_file='/kaggle/working/model_plot.png', show_shapes=True, show_layer_names=True)
```
Python

```python
EPOCHS = 40
initial_lr = 1e-5
decay_steps = 10
decay_rate = 1.0

lr_scheduler = ExponentialDecay(initial_lr, decay_steps, decay_rate)
training_optimizer = Adam(learning_rate=lr_scheduler)
early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)

model.compile(optimizer=training_optimizer, loss='binary_crossentropy', metrics=['accuracy','precision','recall','auc','binary_accuracy'])
```
[27] Python

```python
history = model.fit(x_train, y_train, batch_size=32, epochs=EPOCHS, validation_data=(x_val, y_val), verbose=0, callbacks=[early_stopping, TqdmCallback(verbose=2)])
```
Python

```python
save_path = '/kaggle/working/covid-model.h5'
model.save(save_path)
```
[30] Python

Model Validation and Evaluation Report (5 marks):

<table>
<tr><th>Model</th><th>Summary</th><th>Training and Validation Performance Metrics</th></tr>
<tr><td>Model 1</td><td>
<table>
<tr><th>Layer (type)</th><th>Output Shape</th><th>Param #</th></tr>
<tr><td>input_layer (InputLayer)</td><td>(None, 224, 224, 3)</td><td>0</td></tr>
<tr><td>block1_conv1 (Conv2D)</td><td>(None, 224, 224, 64)</td><td>1,792</td></tr>
<tr><td>block1_conv2 (Conv2D)</td><td>(None, 224, 224, 64)</td><td>36,928</td></tr>
<tr><td>block1_pool (MaxPooling2D)</td><td>(None, 112, 112, 64)</td><td>0</td></tr>
<tr><td>block2_conv1 (Conv2D)</td><td>(None, 112, 112, 128)</td><td>73,856</td></tr>
<tr><td>block2_conv2 (Conv2D)</td><td>(None, 112, 112, 128)</td><td>147,584</td></tr>
<tr><td>block2_pool (MaxPooling2D)</td><td>(None, 56, 56, 128)</td><td>0</td></tr>
<tr><td>block3_conv1 (Conv2D)</td><td>(None, 56, 56, 256)</td><td>295,168</td></tr>
<tr><td>block3_conv2 (Conv2D)</td><td>(None, 56, 56, 256)</td><td>590,080</td></tr>
<tr><td>block3_conv3 (Conv2D)</td><td>(None, 56, 56, 256)</td><td>590,080</td></tr>
<tr><td>block3_conv4 (Conv2D)</td><td>(None, 56, 56, 256)</td><td>590,080</td></tr>
<tr><td>block3_pool (MaxPooling2D)</td><td>(None, 28, 28, 256)</td><td>0</td></tr>
<tr><td>block4_conv1 (Conv2D)</td><td>(None, 28, 28, 512)</td><td>1,180,160</td></tr>
<tr><td>block4_conv2 (Conv2D)</td><td>(None, 28, 28, 512)</td><td>2,359,808</td></tr>
<tr><td>block4_conv3 (Conv2D)</td><td>(None, 28, 28, 512)</td><td>2,359,808</td></tr>
<tr><td>block4_conv4 (Conv2D)</td><td>(None, 28, 28, 512)</td><td>2,359,808</td></tr>
<tr><td>block4_pool (MaxPooling2D)</td><td>(None, 14, 14, 512)</td><td>0</td></tr>
<tr><td>block5_conv1 (Conv2D)</td><td>(None, 14, 14, 512)</td><td>2,359,808</td></tr>
<tr><td>block5_conv2 (Conv2D)</td><td>(None, 14, 14, 512)</td><td>2,359,808</td></tr>
<tr><td>block5_conv3 (Conv2D)</td><td>(None, 14, 14, 512)</td><td>2,359,808</td></tr>
<tr><td>block5_conv4 (Conv2D)</td><td>(None, 14, 14, 512)</td><td>2,359,808</td></tr>
<tr><td>block5_pool (MaxPooling2D)</td><td>(None, 7, 7, 512)</td><td>0</td></tr>
<tr><td>flatten (Flatten)</td><td>(None, 25088)</td><td>0</td></tr>
<tr><td>dense (Dense)</td><td>(None, 1024)</td><td>25,691,136</td></tr>
<tr><td>dense_1 (Dense)</td><td>(None, 512)</td><td>524,800</td></tr>
<tr><td>dense_2 (Dense)</td><td>(None, 1)</td><td>513</td></tr>
</table>

Total params: 46,240,833 (176.39 MB)
Trainable params: 26,216,449 (100.01 MB)
Non-trainable params: 20,024,384 (76.39 MB)
</td><td>

```
0epoch [00:00, ?epoch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
I0000 00:00:1720385220.728965     133 device_compiler.h:186] Compiled cluster using XLA!  This line is logged at most once for the lifetime of the process.
W0000 00:00:1720385220.755687     133 graph_launch.cc:671] Fallback to op-by-op mode because memset node breaks graph update
W0000 00:00:1720385264.454326     133 graph_launch.cc:671] Fallback to op-by-op mode because memset node breaks graph update
W0000 00:00:1720385266.172561     135 graph_launch.cc:671] Fallback to op-by-op mode because memset node breaks graph update
W0000 00:00:1720385284.676634     135 graph_launch.cc:671] Fallback to op-by-op mode because memset node breaks graph update
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
  0%|          | 0.00/204 [00:00<?, ?batch/s]
```

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