

Model Development Phase Template

Date	6 th July 2024
Team ID	SWTID1720097611
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):

Model 1 (VGG16):

```
my_callbacks = [EarlyStopping(patience=6)]

vgg16.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
vgg16.fit(train_data, epochs=20, validation_data=test_data, steps_per_epoch=len(train_data)//16,
          validation_steps=len(test_data)//16, callbacks=my_callbacks)
```

Model 2 (ResNet50):

```
resnet50.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])

my_callbacks = [EarlyStopping(patience=6)]

resnet50.fit(train_data,epochs=15,validation_data=test_data,steps_per_epoch=len(train_data)//16,
             validation_steps=len(test_data)//16, callbacks=my_callbacks)
```

Model 3 (InceptionV3):

```
inceptionV3.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])

my_callbacks = [EarlyStopping(patience=6)]
inceptionV3.fit(train_data,epochs=20,validation_data=test_data , steps_per_epoch=len(train_data)//16,
                validation_steps=len(test_data)//16, callbacks=my_callbacks)
```

Model 4 (Xception):

```
xception.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])

my_callbacks = [EarlyStopping(patience=6)]
xception.fit(train_data,epochs=15,validation_data=test_data , steps_per_epoch=len(train_data)//16,
            validation_steps=len(test_data)//16, callbacks=my_callbacks)
```

Model Validation and Evaluation Report (5 marks):

Model	Summary	Training and Validation Performance Metrics
VGG16	<pre>vgg = VGG16(input_shape=(224,224,3),include_top=False) for layers in vgg.layers: layers.trainable = False x = Flatten()(vgg.output) x = Dense(256,activation='relu')(x) output = Dense(4,activation='softmax')(x) vgg16= Model(vgg.input,output) vgg16.summary()</pre>	<pre>Epoch 1/20 12/12 [=====] - 352s 29s/step - loss: 3.6223 - accuracy: 0.4245 - val_loss: 1.8058 - val_accuracy: 0.6458 Epoch 2/20 12/12 [=====] - 345s 29s/step - loss: 0.8430 - accuracy: 0.6562 - val_loss: 0.6532 - val_accuracy: 0.7708 Epoch 3/20 12/12 [=====] - 334s 28s/step - loss: 0.6600 - accuracy: 0.7578 - val_loss: 0.7234 - val_accuracy: 0.7500 Epoch 4/20 12/12 [=====] - 339s 29s/step - loss: 0.5878 - accuracy: 0.7812 - val_loss: 0.5658 - val_accuracy: 0.7708 Epoch 5/20 12/12 [=====] - 328s 27s/step - loss: 0.5933 - accuracy: 0.7552 - val_loss: 0.5688 - val_accuracy: 0.8021 Epoch 6/20 12/12 [=====] - 335s 28s/step - loss: 0.5312 - accuracy: 0.7786 - val_loss: 0.4295 - val_accuracy: 0.8646 Epoch 7/20 12/12 [=====] - 336s 28s/step - loss: 0.5392 - accuracy: 0.7630 - val_loss: 0.3988 - val_accuracy: 0.8542 Epoch 8/20 12/12 [=====] - 315s 28s/step - loss: 0.4608 - accuracy: 0.8287 - val_loss: 0.4658 - val_accuracy: 0.8438 Epoch 9/20 12/12 [=====] - 334s 28s/step - loss: 0.4920 - accuracy: 0.8021 - val_loss: 0.5035 - val_accuracy: 0.8542 Epoch 10/20 12/12 [=====] - 339s 29s/step - loss: 0.4743 - accuracy: 0.8203 - val_loss: 0.4067 - val_accuracy: 0.8333 Epoch 11/20 12/12 [=====] - 334s 28s/step - loss: 0.4811 - accuracy: 0.7891 - val_loss: 0.4145 - val_accuracy: 0.8333 Epoch 12/20 12/12 [=====] - 298s 25s/step - loss: 0.5810 - accuracy: 0.8062 - val_loss: 0.3838 - val_accuracy: 0.8542 Epoch 13/20 12/12 [=====] - 313s 27s/step - loss: 0.5635 - accuracy: 0.7734 - val_loss: 0.4627 - val_accuracy: 0.8125 Epoch 14/20 12/12 [=====] - 333s 28s/step - loss: 0.4892 - accuracy: 0.8568 - val_loss: 0.4268 - val_accuracy: 0.8333 Epoch 15/20 12/12 [=====] - 332s 28s/step - loss: 0.4894 - accuracy: 0.8620 - val_loss: 0.5988 - val_accuracy: 0.7500 Epoch 16/20 12/12 [=====] - 332s 28s/step - loss: 0.3832 - accuracy: 0.8776 - val_loss: 0.3152 - val_accuracy: 0.8854 Epoch 17/20 12/12 [=====] - 332s 26s/step - loss: 0.3281 - accuracy: 0.8958 - val_loss: 0.3658 - val_accuracy: 0.8646 Epoch 18/20 12/12 [=====] - 333s 28s/step - loss: 0.3957 - accuracy: 0.8516 - val_loss: 0.4924 - val_accuracy: 0.8229 Epoch 19/20 12/12 [=====] - 332s 28s/step - loss: 0.4734 - accuracy: 0.8255 - val_loss: 0.4085 - val_accuracy: 0.8854 Epoch 20/20 12/12 [=====] - 337s 28s/step - loss: 0.3481 - accuracy: 0.8620 - val_loss: 0.4055 - val_accuracy: 0.8542 keras.src.callbacks.History at 0x78918d66080</pre>
ResNet50	<pre>resnet= ResNet50(input_shape=(224,224,3),include_top=False) for layers in resnet.layers: layers.trainable = False x=Flatten()(resnet.output) x=Dense(256,activation='relu')(x) output=Dense(4,activation='softmax')(x) resnet50=Model(resnet.input,output) resnet50.summary()</pre>	<pre>Epoch 1/15 12/12 [=====] - 157s 13s/step - loss: 16.5380 - accuracy: 0.2917 - val_loss: 6.5168 - val_accuracy: 0.4583 Epoch 2/15 12/12 [=====] - 138s 11s/step - loss: 3.2902 - accuracy: 0.7568 - val_loss: 1.7918 - val_accuracy: 0.7812 Epoch 3/15 12/12 [=====] - 120s 10s/step - loss: 1.7814 - accuracy: 0.8750 - val_loss: 1.7131 - val_accuracy: 0.8125 Epoch 4/15 12/12 [=====] - 122s 10s/step - loss: 1.4132 - accuracy: 0.9115 - val_loss: 1.6815 - val_accuracy: 0.8042 Epoch 5/15 12/12 [=====] - 187s 11s/step - loss: 1.2343 - accuracy: 0.9583 - val_loss: 1.3384 - val_accuracy: 0.8646 Epoch 6/15 12/12 [=====] - 129s 11s/step - loss: 1.3388 - accuracy: 0.9073 - val_loss: 1.1287 - val_accuracy: 0.8938 Epoch 7/15 12/12 [=====] - 193s 17s/step - loss: 1.2891 - accuracy: 0.9748 - val_loss: 1.3216 - val_accuracy: 0.8479 Epoch 8/15 12/12 [=====] - 117s 10s/step - loss: 1.2501 - accuracy: 0.9453 - val_loss: 1.3484 - val_accuracy: 0.8083 Epoch 9/15 12/12 [=====] - 125s 10s/step - loss: 1.1748 - accuracy: 0.9800 - val_loss: 1.2248 - val_accuracy: 0.8958 Epoch 10/15 12/12 [=====] - 136s 11s/step - loss: 1.1538 - accuracy: 0.9896 - val_loss: 1.2708 - val_accuracy: 0.8271 keras.src.callbacks.History at 0x78918d66080</pre>

<p>Inception</p>	<pre>inception = InceptionV3(input_shape=(224,224,3),include_top=False) for layers in inception.layers: layers.trainable = False x=Flatten()(inception.output) x=Dense(256,activation='relu')(x) output=Dense(4,activation='softmax')(x) inceptionV3=Model(inception.input,output)</pre>	<pre>Epoch 1/20 12/12 [=====] - 82s 6s/step - loss: 22.7226 - accuracy: 0.4792 - val_loss: 6.5945 - val_accuracy: 0.6042 Epoch 2/20 12/12 [=====] - 75s 6s/step - loss: 6.2488 - accuracy: 0.6146 - val_loss: 4.6134 - val_accuracy: 0.7013 Epoch 3/20 12/12 [=====] - 80s 7s/step - loss: 4.0873 - accuracy: 0.6901 - val_loss: 2.7736 - val_accuracy: 0.7917 Epoch 4/20 12/12 [=====] - 82s 7s/step - loss: 4.2119 - accuracy: 0.7109 - val_loss: 3.1315 - val_accuracy: 0.8458 Epoch 5/20 12/12 [=====] - 76s 6s/step - loss: 2.3398 - accuracy: 0.7334 - val_loss: 2.6139 - val_accuracy: 0.7604 Epoch 6/20 12/12 [=====] - 79s 7s/step - loss: 1.8193 - accuracy: 0.7917 - val_loss: 3.8284 - val_accuracy: 0.6842 Epoch 7/20 12/12 [=====] - 79s 7s/step - loss: 1.2389 - accuracy: 0.8281 - val_loss: 0.9795 - val_accuracy: 0.8021 Epoch 8/20 12/12 [=====] - 81s 7s/step - loss: 1.3443 - accuracy: 0.7943 - val_loss: 1.4651 - val_accuracy: 0.8021 Epoch 9/20 12/12 [=====] - 78s 7s/step - loss: 1.6821 - accuracy: 0.7682 - val_loss: 1.5841 - val_accuracy: 0.7604 Epoch 10/20 12/12 [=====] - 80s 7s/step - loss: 1.9994 - accuracy: 0.7109 - val_loss: 1.8056 - val_accuracy: 0.8542 Epoch 11/20 12/12 [=====] - 74s 6s/step - loss: 1.5287 - accuracy: 0.8464 - val_loss: 1.9983 - val_accuracy: 0.7083 Epoch 12/20 12/12 [=====] - 79s 7s/step - loss: 1.5837 - accuracy: 0.7812 - val_loss: 1.8305 - val_accuracy: 0.7708 Epoch 13/20 12/12 [=====] - 81s 7s/step - loss: 1.3807 - accuracy: 0.7839 - val_loss: 2.3536 - val_accuracy: 0.7292 keras.src.callbacks.History at 0x7891808beeb0</pre>
<p>Xception</p>	<pre>xception = Xception(input_shape=(299,299,3),include_top=False) for layers in xception.layers: layers.trainable = False x=Flatten()(xception.output) x=Dense(256,activation='relu')(x) output=Dense(4,activation='softmax')(x) xception=Model(xception.input,output) xception.summary()</pre>	<pre>Epoch 1/15 12/12 [=====] - 346s 28s/step - loss: 10.7327 - accuracy: 0.5729 - val_loss: 6.9207 - val_accuracy: 0.6458 Epoch 2/15 12/12 [=====] - 278s 21s/step - loss: 4.7793 - accuracy: 0.7188 - val_loss: 3.3135 - val_accuracy: 0.7292 Epoch 3/15 12/12 [=====] - 283s 24s/step - loss: 4.8017 - accuracy: 0.6979 - val_loss: 4.9188 - val_accuracy: 0.6979 Epoch 4/15 12/12 [=====] - 251s 21s/step - loss: 3.2561 - accuracy: 0.7448 - val_loss: 2.8701 - val_accuracy: 0.7917 Epoch 5/15 12/12 [=====] - 236s 20s/step - loss: 2.4279 - accuracy: 0.7781 - val_loss: 2.6128 - val_accuracy: 0.7917 Epoch 6/15 12/12 [=====] - 252s 21s/step - loss: 2.3558 - accuracy: 0.7630 - val_loss: 2.4274 - val_accuracy: 0.7917 Epoch 7/15 12/12 [=====] - 284s 24s/step - loss: 1.6635 - accuracy: 0.7865 - val_loss: 1.8533 - val_accuracy: 0.8229 Epoch 8/15 12/12 [=====] - 287s 24s/step - loss: 1.3984 - accuracy: 0.7969 - val_loss: 2.5583 - val_accuracy: 0.7708 Epoch 9/15 12/12 [=====] - 287s 24s/step - loss: 1.3955 - accuracy: 0.8073 - val_loss: 0.7863 - val_accuracy: 0.7812 Epoch 10/15 12/12 [=====] - 251s 21s/step - loss: 1.1529 - accuracy: 0.8229 - val_loss: 1.3036 - val_accuracy: 0.7083 Epoch 11/15 12/12 [=====] - 250s 21s/step - loss: 0.8218 - accuracy: 0.8203 - val_loss: 0.5705 - val_accuracy: 0.7812 Epoch 12/15 12/12 [=====] - 284s 24s/step - loss: 0.8522 - accuracy: 0.8255 - val_loss: 1.1529 - val_accuracy: 0.7917 Epoch 13/15 12/12 [=====] - 249s 21s/step - loss: 0.8832 - accuracy: 0.8464 - val_loss: 0.5329 - val_accuracy: 0.8021 Epoch 14/15 12/12 [=====] - 286s 24s/step - loss: 0.8095 - accuracy: 0.8229 - val_loss: 0.7238 - val_accuracy: 0.8229 Epoch 15/15 12/12 [=====] - 272s 23s/step - loss: 0.6069 - accuracy: 0.8706 - val_loss: 0.8447 - val_accuracy: 0.8021 keras.src.callbacks.History at 0x7891808beeb0</pre>