

Model Development Phase

Date	01 July 2024
Team ID	SWTID1720084679
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):

Paste the screenshot of the model training code

▼ Inception

```
[ ] #!pip install tensorflow==2.13

train = train_datagen.flow_from_directory(trainPath,target_size=(299,299),batch_size=16)
test = test_datagen.flow_from_directory(testPath,target_size=(299,299),batch_size=16)

Found 3476 images belonging to 4 classes.
Found 1307 images belonging to 4 classes.

[ ] 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=(299 ,299,3))

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/inception_v3/inception_v3_weights_tf_dim_ordering_tf_kernels_notop.h5
87910968/87910968 [=====] - 0s 0us/step
```

+ Code + Text

Connect T4

+ Gemini

```
x = Flatten()(InceptionV3.output)

[ ] output = Dense(4,activation='softmax')(x)

[ ] InceptionV3 = Model(InceptionV3.input,output)

[ ] InceptionV3.summary()

Model: "model_2"
-----
Layer (type)                 Output Shape              Param #   Connected to
-----
input_3 (InputLayer)         [(None, 299, 299, 3)]    0         []
conv2d (Conv2D)              (None, 149, 149, 32)     864       ['input_3[0][0]']
batch_normalization (Batch   (None, 149, 149, 32)     96        ['conv2d[0][0]']
Normalization)
activation (Activation)       (None, 149, 149, 32)     0         ['batch_normalization[0][0]']
conv2d_1 (Conv2D)            (None, 147, 147, 32)     9216      ['activation[0][0]']
batch_normalization_1 (Bat   (None, 147, 147, 32)     96        ['conv2d_1[0][0]']
chNormalization)
activation_1 (Activation)     (None, 147, 147, 32)     0         ['batch_normalization_1[0][0]']
```

Model Validation and Evaluation Report (5 marks):

Model	Summary	Training and Validation Performance Metrics
VGG16 Model	<p>▼ VGG16</p> <pre>[] 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() Model: "vgg16" ----- Layer (type) Output Shape Param # Connected to ----- input_1 (InputLayer) [(None, 224, 224, 3)] 0 [] block1_conv1 (Conv2D) (None, 224, 224, 64) 1792 ['input_1[0][0]']</pre> <pre>[] for layer in vgg.layers: [] layer.trainable=False [] x = Flatten()(vgg.output) [] output = Dense(4,activation='softmax')(x) [] vgg16 = Model(vgg.input,output)</pre>	<pre>[] vgg16.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy']) [] vgg16.fit(train,validation_data=test,epochs=2) Epoch 1/2 218/218 [=====] - 66s 271ms/step - loss: 0.0065 - accuracy: 0.9954 - val_loss: 0.0000e+00 - val_accuracy: 1.0000 Epoch 2/2 218/218 [=====] - 55s 253ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000 keras.src.callbacks.History at 0x7e07002e4160 [] vgg16.save('Vgg19.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_api.save_model'</pre>

ResNet Model

⌵ Resnet

```
[ ] from tensorflow.keras.applications.resnet50 import ResNet50
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.models import Model

[ ] resnet50 = ResNet50(include_top=False, input_shape=(224, 224, 3))

📄 Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resnet50/94765736/94765736 [=====] - 0s 0us/step

[ ] x = Flatten()(resnet50.output)

[ ] output = Dense(4, activation='softmax')(x)

[ ] resnet50 = Model(resnet50.input, output)
```

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

🔴 resnet50.fit(train, validation_data=test, epochs=2)

📄 Epoch 1/2
218/218 [=====] - 61s 259ms/step - loss: 0.0111 - accuracy: 0.9954 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 2/2
218/218 [=====] - 53s 245ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
keras.src.callbacks.History at 0x706f02070eb0

[ ] resnet50.save('ResNet_A5')
```

Inception on

⌵ InceptionV3

```
[ ] train = train_datagen.flow_from_directory(trainPath, target_size=(299, 299), batch_size=16)
test = test_datagen.flow_from_directory(testPath, target_size=(299, 299), batch_size=16)

📄 Found 3476 images belonging to 4 classes.
Found 1307 images belonging to 4 classes.

[ ] 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=(299, 299, 3))

📄 Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/inception_v3/inception_v3_weights_tf_dim_ordering_tf_kernels_notop-87910968.tgz [=====] - 0s 0us/step

[ ] x = Flatten()(InceptionV3.output)

[ ] output = Dense(4, activation='softmax')(x)

[ ] InceptionV3 = Model(InceptionV3.input, output)
```

```
[ ] inceptionv3.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

🔴 InceptionV3.fit(train, validation_data=test, epochs=5)

📄 Epoch 1/5
218/218 [=====] - 146s 47ms/step - loss: 0.0051 - accuracy: 0.9983 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 2/5
218/218 [=====] - 97s 44ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 3/5
218/218 [=====] - 95s 43ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 4/5
218/218 [=====] - 95s 43ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 5/5
218/218 [=====] - 96s 43ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
keras.src.callbacks.History at 0x70680605050b

[ ] InceptionV3.save('InceptionV3_covid_A5')
```

Xception n

⌵ Xception

```
[ ] train = train_datagen.flow_from_directory(trainPath, target_size=(299, 299), batch_size=16)
test = test_datagen.flow_from_directory(testPath, target_size=(299, 299), batch_size=16)

📄 Found 3476 images belonging to 4 classes.
Found 1307 images belonging to 4 classes.

🔴 from tensorflow.keras.applications.xception import Xception
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.models import Model

[ ] Xception = Xception(include_top=False, input_shape=(299, 299, 3))

📄 Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/xception/xception_weights_tf_dim_ordering_tf_kernels_notop-83683744.tgz [=====] - 0s 0us/step

[ ] x = Flatten()(Xception.output)

[ ] output = Dense(4, activation='softmax')(x)

[ ] Xception = Model(Xception.input, output)
```

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

[ ] Xception.fit(train, validation_data=test, epochs=2)

📄 Epoch 1/2
218/218 [=====] - 156s 61ms/step - loss: 0.0070 - accuracy: 0.9960 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 2/2
218/218 [=====] - 119s 54ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 8.4370e-06 - val_accuracy: 1.0000
keras.src.callbacks.History at 0x70687d7f93c0

[ ] Xception.save('Xception_covid_A5')
```