Model Development Phase Template

Date	15 March 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	`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()`
		Epoch 1/20
		12/12 [==============================] - 352s 29s/step - loss: 3.6223 - accuracy: 0.4245 - val_loss: 1.0858 - val_accuracy: 0.6458
		Epoch 2/20
		12/12 [==============================] - 345s 29s/step - loss: 0.8410 - accuracy: 0.6562 - val_loss: 0.6532 - val_accuracy: 0.7708
		Epoch 3/20
		12/12 [==============================] - 334s 28s/step - loss: 0.6660 - 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: 0.5618 - val_accuracy: 0.7708
		Epoch 5/20
		12/12 [==============================] - 320s 27s/step - loss: 0.5933 - accuracy: 0.7552 - val_loss: 0.5698 - val_accuracy: 0.8021
		Epoch 6/20
		12/12 [==============================] - 335s 28s/step - loss: 0.5112 - 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: 0.4660 - accuracy: 0.8287 - val_loss: 0.4610 - 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: 0.5010 - 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: 0.4092 - accuracy: 0.8568 - val_loss: 0.4268 - val_accuracy: 0.8333
		Epoch 15/20
		12/12 [==============================] - 332s 28s/step - loss: 0.4094 - accuracy: 0.8620 - val_loss: 0.5908 - 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 [==============================] - 312s 26s/step - loss: 0.3281 - accuracy: 0.8958 - val_loss: 0.3650 - 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: 0.4065 - 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 0x7891868a6600>

ResNet50	`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()`
		Epoch 1/15
		12/12 [==============================] - 157s 13s/step - loss: 16.5380 - accuracy: 0.2917 - val_loss: 6.5160 - val_accuracy: 0.4583
		Epoch 2/15
		12/12 [==============================] - 138s 11s/step - loss: 3.2902 - accuracy: 0.3568 - val_loss: 1.7918 - val_accuracy: 0.2812
		Epoch 3/15
		12/12 [==============================] - 120s 10s/step - loss: 1.7814 - accuracy: 0.3750 - val_loss: 1.7131 - val_accuracy: 0.3125
		Epoch 4/15
		12/12 [==============================] - 122s 10s/step - loss: 1.4112 - accuracy: 0.4315 - val_loss: 1.0815 - val_accuracy: 0.6042
		Epoch 5/15
		12/12 [==============================] - 187s 11s/step - loss: 1.2343 - accuracy: 0.4583 - val_loss: 1.3304 - val_accuracy: 0.3646
		Epoch 6/15
		12/12 [==============================] - 129s 11s/step - loss: 1.3388 - accuracy: 0.4073 - val_loss: 1.1267 - val_accuracy: 0.5938
		Epoch 7/15
		12/12 [==============================] - 193s 17s/step - loss: 1.2891 - accuracy: 0.4740 - val_loss: 1.3216 - val_accuracy: 0.4479
		Epoch 8/15
		12/12 [==============================] - 117s 10s/step - loss: 1.2501 - accuracy: 0.4453 - val_loss: 1.3494 - val_accuracy: 0.2083
		Epoch 9/15
		12/12 [==============================] - 125s 10s/step - loss: 1.1740 - accuracy: 0.5000 - val_loss: 1.2248 - val_accuracy: 0.3958
		Epoch 10/15
		12/12 [==============================] - 136s 11s/step - loss: 1.1538 - accuracy: 0.4896 - val_loss: 1.2708 - val_accuracy: 0.4271
		<keras.src.callbacks.History at 0x7891a8e54ca0>

Inception	`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)`
		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: 0.7083
		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: 0.6458
		Epoch 5/20
		12/12 [==========] - 76s 6s/step - loss: 2.3398 - accuracy: 0.7734 - val_loss: 2.6139 - val_accuracy: 0.7604
		Epoch 6/20
		12/12 [==========] - 79s 7s/step - loss: 1.8193 - accuracy: 0.7917 - val_loss: 3.0264 - val_accuracy: 0.6042
		Epoch 7/20
		12/12 [==========] - 79s 7s/step - loss: 1.2109 - accuracy: 0.8281 - val_loss: 0.9765 - 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: 1.0656 - val_accuracy: 0.8542
		Epoch 11/20
		12/12 [==========] - 74s 6s/step - loss: 1.3287 - 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 0x7891868c0ee0>

Xception	`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()`
		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 [==========] - 270s 23s/step - loss: 4.7793 - accuracy: 0.7388 - 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 [==========] - 253s 21s/step - loss: 3.2561 - accuracy: 0.7448 - val_loss: 2.0761 - 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: 2.5503 - 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: 0.7230 - val_accuracy: 0.8229
		Epoch 15/15
		12/12 [==========] - 272s 23s/step - loss: 0.6069 - accuracy: 0.8708 - val_loss: 0.9447 - val_accuracy: 0.8021
		<keras.src.callbacks.History at 0x7891bfa53eb0>