

## Model Development Phase - 2

Date	19 June 2025
Team ID	SWTID1749908722
Project Title	CardMaster: Intelligent Playing Card Recognition using Transfer Learning

### Initial Model Training Code:

VGG 16:

```
#creating new model
vgg16 = Model(vgg.input,output)

[ ] #multi-class classification, optimization algorithm, tracking accuracy
vgg16.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])

#training model on 15 epochs
vgg16.fit(train,epochs=15,validation_data=val,steps_per_epoch=len(train),validation_steps=len(val))
```

Inception V3:

```
#Create final model
inception = Model(inputs=base_model.input, outputs=predictions)

#compile the model
inception.compile(optimizer=Adam(learning_rate=0.001), loss='categorical_crossentropy', metrics=['accuracy'])

#training model
inception.fit(train,epochs=15,validation_data=val,steps_per_epoch=len(train),validation_steps=len(val))
```

Xception:

```
#Create final model
xcep = Model(inputs=xception.input, outputs=predictions)

#compile the model
xcep.compile(optimizer=Adam(learning_rate=0.001), loss='categorical_crossentropy', metrics=['accuracy'])

#training model
xcep.fit(train,epochs=15,validation_data=val,steps_per_epoch=len(train),validation_steps=len(val))
```

## Model Validation and Evaluation Report:

Model	Summary	Training and Validation Performance Metrics																																																															
VGG16	<div><div>vgg16.summary()</div><div>Model: "functional"</div><table><thead><tr><th>Layer (type)</th><th>Output Shape</th><th>Param #</th></tr></thead><tbody><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>1,344</td></tr><tr><td>block1_conv2 (conv2D)</td><td>(None, 224, 224, 64)</td><td>96,448</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>71,680</td></tr><tr><td>block2_conv2 (conv2D)</td><td>(None, 112, 112, 128)</td><td>184,384</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>268,144</td></tr><tr><td>block3_conv2 (conv2D)</td><td>(None, 56, 56, 256)</td><td>590,880</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>1,180,160</td></tr><tr><td>block4_conv3 (conv2D)</td><td>(None, 28, 28, 512)</td><td>1,180,160</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>1,180,160</td></tr><tr><td>block5_conv2 (conv2D)</td><td>(None, 14, 14, 512)</td><td>1,180,160</td></tr><tr><td>block5_conv3 (conv2D)</td><td>(None, 14, 14, 512)</td><td>1,180,160</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, 153600)</td><td>0</td></tr><tr><td>dense (Dense)</td><td>(None, 1000)</td><td>1,540,100</td></tr></tbody></table><div>Total params: 17,248,384 (61.20 MB) Trainable params: 1,540,100 (5.07 MB) Non-trainable params: 15,708,284 (56.13 MB)</div></div>	Layer (type)	Output Shape	Param #	input_layer (InputLayer)	(None, 224, 224, 3)	0	block1_conv1 (conv2D)	(None, 224, 224, 64)	1,344	block1_conv2 (conv2D)	(None, 224, 224, 64)	96,448	block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0	block2_conv1 (conv2D)	(None, 112, 112, 128)	71,680	block2_conv2 (conv2D)	(None, 112, 112, 128)	184,384	block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0	block3_conv1 (conv2D)	(None, 56, 56, 256)	268,144	block3_conv2 (conv2D)	(None, 56, 56, 256)	590,880	block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0	block4_conv1 (conv2D)	(None, 28, 28, 512)	1,180,160	block4_conv2 (conv2D)	(None, 28, 28, 512)	1,180,160	block4_conv3 (conv2D)	(None, 28, 28, 512)	1,180,160	block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0	block5_conv1 (conv2D)	(None, 14, 14, 512)	1,180,160	block5_conv2 (conv2D)	(None, 14, 14, 512)	1,180,160	block5_conv3 (conv2D)	(None, 14, 14, 512)	1,180,160	block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0	Flatten (Flatten)	(None, 153600)	0	dense (Dense)	(None, 1000)	1,540,100	<div><div>Training model on 15 epochs</div><div>vgg16.fit(train_epochs=15,validation_data=val_steps_per_epoch=len(train),validation_steps=len(val))</div><div>Epoch 1/15 106s 22ms/step - accuracy: 0.7903 - loss: 0.9205 - val_accuracy: 0.7623 - val_loss: 1.3824</div><div>Epoch 2/15 149s 23ms/step - accuracy: 0.8325 - loss: 0.7116 - val_accuracy: 0.7547 - val_loss: 1.3588</div><div>Epoch 3/15 110s 23ms/step - accuracy: 0.8364 - loss: 0.6980 - val_accuracy: 0.7774 - val_loss: 1.2930</div><div>Epoch 4/15 108s 22ms/step - accuracy: 0.8448 - loss: 0.6873 - val_accuracy: 0.7547 - val_loss: 1.4268</div><div>Epoch 5/15 142s 22ms/step - accuracy: 0.8658 - loss: 0.5898 - val_accuracy: 0.7698 - val_loss: 1.5201</div><div>Epoch 6/15 141s 22ms/step - accuracy: 0.8672 - loss: 0.6510 - val_accuracy: 0.7396 - val_loss: 1.9421</div><div>Epoch 7/15 108s 22ms/step - accuracy: 0.8814 - loss: 0.5517 - val_accuracy: 0.7925 - val_loss: 1.5268</div><div>Epoch 8/15 108s 22ms/step - accuracy: 0.8858 - loss: 0.5564 - val_accuracy: 0.7849 - val_loss: 1.4964</div><div>Epoch 9/15 107s 22ms/step - accuracy: 0.8868 - loss: 0.5159 - val_accuracy: 0.7811 - val_loss: 1.5671</div><div>Epoch 10/15 143s 22ms/step - accuracy: 0.8875 - loss: 0.5149 - val_accuracy: 0.7962 - val_loss: 1.7840</div><div>Epoch 11/15 112s 23ms/step - accuracy: 0.9045 - loss: 0.4704 - val_accuracy: 0.7736 - val_loss: 1.7919</div><div>Epoch 12/15 138s 22ms/step - accuracy: 0.8977 - loss: 0.4955 - val_accuracy: 0.8189 - val_loss: 1.4279</div><div>Epoch 13/15 162s 22ms/step - accuracy: 0.9152 - loss: 0.3875 - val_accuracy: 0.7396 - val_loss: 1.9112</div><div>Epoch 14/15 107s 22ms/step - accuracy: 0.9108 - loss: 0.4597 - val_accuracy: 0.8189 - val_loss: 1.7562</div><div>Epoch 15/15 113s 23ms/step - accuracy: 0.9134 - loss: 0.4300 - val_accuracy: 0.8377 - val_loss: 1.2883</div><div>clara_src.callbacks.history.history at 0x7b2628291580</div></div> <div>Accuracy: 0.9134 Loss: 0.4300 Validation Accuracy: 0.8377 Validation Loss: 1.2883</div>
Layer (type)	Output Shape	Param #																																																															
input_layer (InputLayer)	(None, 224, 224, 3)	0																																																															
block1_conv1 (conv2D)	(None, 224, 224, 64)	1,344																																																															
block1_conv2 (conv2D)	(None, 224, 224, 64)	96,448																																																															
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0																																																															
block2_conv1 (conv2D)	(None, 112, 112, 128)	71,680																																																															
block2_conv2 (conv2D)	(None, 112, 112, 128)	184,384																																																															
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0																																																															
block3_conv1 (conv2D)	(None, 56, 56, 256)	268,144																																																															
block3_conv2 (conv2D)	(None, 56, 56, 256)	590,880																																																															
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0																																																															
block4_conv1 (conv2D)	(None, 28, 28, 512)	1,180,160																																																															
block4_conv2 (conv2D)	(None, 28, 28, 512)	1,180,160																																																															
block4_conv3 (conv2D)	(None, 28, 28, 512)	1,180,160																																																															
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0																																																															
block5_conv1 (conv2D)	(None, 14, 14, 512)	1,180,160																																																															
block5_conv2 (conv2D)	(None, 14, 14, 512)	1,180,160																																																															
block5_conv3 (conv2D)	(None, 14, 14, 512)	1,180,160																																																															
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0																																																															
Flatten (Flatten)	(None, 153600)	0																																																															
dense (Dense)	(None, 1000)	1,540,100																																																															
Xception	<div><div>[10] Summary for Xception Model</div><div>xcep.summary()</div><div>Model: "functional_1"</div><table><thead><tr><th>Layer (type)</th><th>Output Shape</th><th>Param #</th><th>Connected to</th></tr></thead><tbody><tr><td>input_layer_1 (InputLayer)</td><td>(None, 224, 224, 3)</td><td>0</td><td>-</td></tr><tr><td>block1_conv1 (conv2D)</td><td>(None, 112, 112, 32)</td><td>960</td><td>input_layer_1[1-3]</td></tr><tr><td>block1_conv1_bn (BatchNormalizati...</td><td>(None, 112, 112, 32)</td><td>128</td><td>block1_conv1[1-3]</td></tr><tr><td>block1_conv1_act (Activation)</td><td>(None, 112, 112, 32)</td><td>0</td><td>block1_conv1_bn[1-3]</td></tr><tr><td>block1_conv2 (conv2D)</td><td>(None, 112, 112, 32)</td><td>18,432</td><td>block1_conv1_act[1-3]</td></tr><tr><td>block1_conv2_bn (BatchNormalizati...</td><td>(None, 112, 112, 32)</td><td>128</td><td>block1_conv2[1-3]</td></tr><tr><td>block1_conv2_act (Activation)</td><td>(None, 112, 112, 32)</td><td>0</td><td>block1_conv2_bn[1-3]</td></tr><tr><td>block2_sepconv1 (separableconv2D)</td><td>(None, 112, 112, 32)</td><td>8,192</td><td>block1_conv2_act[1-3]</td></tr><tr><td>block2_sepconv1_bn (BatchNormalizati...</td><td>(None, 112, 112, 32)</td><td>128</td><td>block2_sepconv1[1-3]</td></tr><tr><td>block2_sepconv1_act (Activation)</td><td>(None, 112, 112, 32)</td><td>0</td><td>block2_sepconv1_bn[1-3]</td></tr><tr><td>block2_sepconv2 (separableconv2D)</td><td>(None, 112, 112, 32)</td><td>13,312</td><td>block2_sepconv1_act[1-3]</td></tr></tbody></table></div>	Layer (type)	Output Shape	Param #	Connected to	input_layer_1 (InputLayer)	(None, 224, 224, 3)	0	-	block1_conv1 (conv2D)	(None, 112, 112, 32)	960	input_layer_1[1-3]	block1_conv1_bn (BatchNormalizati...	(None, 112, 112, 32)	128	block1_conv1[1-3]	block1_conv1_act (Activation)	(None, 112, 112, 32)	0	block1_conv1_bn[1-3]	block1_conv2 (conv2D)	(None, 112, 112, 32)	18,432	block1_conv1_act[1-3]	block1_conv2_bn (BatchNormalizati...	(None, 112, 112, 32)	128	block1_conv2[1-3]	block1_conv2_act (Activation)	(None, 112, 112, 32)	0	block1_conv2_bn[1-3]	block2_sepconv1 (separableconv2D)	(None, 112, 112, 32)	8,192	block1_conv2_act[1-3]	block2_sepconv1_bn (BatchNormalizati...	(None, 112, 112, 32)	128	block2_sepconv1[1-3]	block2_sepconv1_act (Activation)	(None, 112, 112, 32)	0	block2_sepconv1_bn[1-3]	block2_sepconv2 (separableconv2D)	(None, 112, 112, 32)	13,312	block2_sepconv1_act[1-3]	<div><div>#training model</div><div>xcep.fit(train_epochs=15,validation_data=val_steps_per_epoch=len(train),validation_steps=len(val))</div><div>Epoch 1/15 477/477 211s 17ms/step - accuracy: 0.8772 - loss: 1.4916 - val_accuracy: 0.1308 - val_loss: 1.8761</div><div>Epoch 2/15 132s 27ms/step - accuracy: 0.3810 - loss: 2.0504 - val_accuracy: 0.4302 - val_loss: 1.5391</div><div>Epoch 3/15 132s 27ms/step - accuracy: 0.5153 - loss: 1.4292 - val_accuracy: 0.7840 - val_loss: 0.7120</div><div>Epoch 4/15 149s 29ms/step - accuracy: 0.6904 - loss: 1.0321 - val_accuracy: 0.8491 - val_loss: 0.4894</div><div>Epoch 5/15 140s 29ms/step - accuracy: 0.7689 - loss: 0.7675 - val_accuracy: 0.8653 - val_loss: 0.5284</div><div>Epoch 6/15 137s 28ms/step - accuracy: 0.8876 - loss: 0.6354 - val_accuracy: 0.9178 - val_loss: 0.2592</div><div>Epoch 7/15 137s 27ms/step - accuracy: 0.8343 - loss: 0.5691 - val_accuracy: 0.9434 - val_loss: 0.2315</div><div>Epoch 8/15 143s 27ms/step - accuracy: 0.8377 - loss: 0.5302 - val_accuracy: 0.9434 - val_loss: 0.2863</div><div>Epoch 9/15 141s 27ms/step - accuracy: 0.8637 - loss: 0.4581 - val_accuracy: 0.9409 - val_loss: 0.1497</div><div>Epoch 10/15 142s 27ms/step - accuracy: 0.8868 - loss: 0.3837 - val_accuracy: 0.9623 - val_loss: 0.1080</div><div>Epoch 11/15 143s 27ms/step - accuracy: 0.8982 - loss: 0.3341 - val_accuracy: 0.9132 - val_loss: 0.2757</div><div>Epoch 12/15 142s 27ms/step - accuracy: 0.8906 - loss: 0.3509 - val_accuracy: 0.9608 - val_loss: 0.1409</div><div>Epoch 13/15 129s 27ms/step - accuracy: 0.9819 - loss: 0.1336 - val_accuracy: 0.9623 - val_loss: 0.1191</div><div>Epoch 14/15 142s 27ms/step - accuracy: 0.9188 - loss: 0.3054 - val_accuracy: 0.9326 - val_loss: 0.1619</div><div>Epoch 15/15 141s 26ms/step - accuracy: 0.9160 - loss: 0.2870 - val_accuracy: 0.9472 - val_loss: 0.1979</div><div>clara_src.callbacks.history.history at 0x7a225a4af0</div></div> <div>Accuracy: 0.9160 Loss: 0.2870 Validation Accuracy: 0.9427 Validation Loss: 0.1979</div>															
Layer (type)	Output Shape	Param #	Connected to																																																														
input_layer_1 (InputLayer)	(None, 224, 224, 3)	0	-																																																														
block1_conv1 (conv2D)	(None, 112, 112, 32)	960	input_layer_1[1-3]																																																														
block1_conv1_bn (BatchNormalizati...	(None, 112, 112, 32)	128	block1_conv1[1-3]																																																														
block1_conv1_act (Activation)	(None, 112, 112, 32)	0	block1_conv1_bn[1-3]																																																														
block1_conv2 (conv2D)	(None, 112, 112, 32)	18,432	block1_conv1_act[1-3]																																																														
block1_conv2_bn (BatchNormalizati...	(None, 112, 112, 32)	128	block1_conv2[1-3]																																																														
block1_conv2_act (Activation)	(None, 112, 112, 32)	0	block1_conv2_bn[1-3]																																																														
block2_sepconv1 (separableconv2D)	(None, 112, 112, 32)	8,192	block1_conv2_act[1-3]																																																														
block2_sepconv1_bn (BatchNormalizati...	(None, 112, 112, 32)	128	block2_sepconv1[1-3]																																																														
block2_sepconv1_act (Activation)	(None, 112, 112, 32)	0	block2_sepconv1_bn[1-3]																																																														
block2_sepconv2 (separableconv2D)	(None, 112, 112, 32)	13,312	block2_sepconv1_act[1-3]																																																														

conv2d_3 (Conv2D)	(None, 7, 7, 32)	768,432	add_10[ ][ ]
block13_pool (MaxPooling2D)	(None, 7, 7, 32)	0	block13_sepconv2_
batch_normalization_ (BatchNormalization)	(None, 7, 7, 32)	6,080	conv2d_3[ ][ ]
add_11 (Add)	(None, 7, 7, 32)	0	block13_pool[ ][ ]
block14_sepconv1 (SeparableConv2D)	(None, 7, 7, 32)	1,302,080	add_11[ ][ ]
block14_sepconv1_bn (BatchNormalization)	(None, 7, 7, 32)	6,144	block14_sepconv1_
block14_sepconv1_a_ (Activation)	(None, 7, 7, 32)	0	block14_sepconv1_
block14_sepconv2 (SeparableConv2D)	(None, 7, 7, 32)	1,302,080	block14_sepconv1_
block14_sepconv2_bn (BatchNormalization)	(None, 7, 7, 32)	6,144	block14_sepconv2_
block14_sepconv2_a_ (Activation)	(None, 7, 7, 32)	0	block14_sepconv2_
global_average_pool_ (GlobalAveragePool2D)	(None, 3840)	0	block14_sepconv2_
dense_1 (Dense)	(None, 3840)	1,809,176	global_average_p_
dense_2 (Dense)	(None, 32)	56,128	dense_1[ ][ ]

Total params: 68,932,000 (262.96 MB)  
Trainable params: 32,909,424 (87.58 MB)  
Non-trainable params: 6,144 (213.00 KB)  
Optimizer params: 3,020,000 (175.17 MB)

#summary for InceptionV3 Model  
inception.summary()

Model: "functional\_3"

Layer (type)	Output Shape	Param #	Connected to
input_layer_3 (InputLayer)	(None, 224, 224, 3)	0	-
conv2d_98 (Conv2D)	(None, 112, 112, 32)	640	input_layer_3[ ][ ]
batch_normalization_ (BatchNormalization)	(None, 112, 112, 32)	76	conv2d_98[ ][ ]
activation_96 (Activation)	(None, 112, 112, 32)	0	batch_normalizat_
conv2d_99 (Conv2D)	(None, 112, 112, 32)	9,424	activation_96[ ][ ]
batch_normalization_ (BatchNormalization)	(None, 112, 112, 32)	76	conv2d_99[ ][ ]
activation_95 (Activation)	(None, 112, 112, 32)	0	batch_normalizat_
conv2d_100 (Conv2D)	(None, 112, 112, 32)	10,432	activation_95[ ][ ]
batch_normalization_ (BatchNormalization)	(None, 112, 112, 32)	76	conv2d_100[ ][ ]
activation_96 (Activation)	(None, 112, 112, 32)	0	batch_normalizat_
max_pooling2d_4 (MaxPooling2D)	(None, 56, 56, 32)	0	activation_96[ ][ ]

Inception  
V3

(Activation)	(None, 7, 7, 384)	0	batch_normalizat_
activation_185 (Activation)	(None, 7, 7, 384)	0	batch_normalizat_
activation_186 (Activation)	(None, 7, 7, 384)	0	batch_normalizat_
batch_normalization_ (BatchNormalization)	(None, 7, 7, 384)	76	conv2d_191[ ][ ]
activation_179 (Activation)	(None, 7, 7, 384)	0	batch_normalizat_
mixed0_1 (Concatenate)	(None, 7, 7, 384)	0	activation_181[ ][ ]
concatenate_3 (Concatenate)	(None, 7, 7, 384)	0	activation_185[ ][ ]
activation_187 (Activation)	(None, 7, 7, 384)	0	batch_normalizat_
mixed10 (Concatenate)	(None, 7, 7, 384)	0	activation_179[ ][ ]
global_average_pool_ (GlobalAveragePool2D)	(None, 3840)	0	mixed10[ ][ ]
dense_5 (Dense)	(None, 3840)	1,809,176	global_average_p_
dense_6 (Dense)	(None, 32)	56,128	dense_5[ ][ ]

Total params: 68,932,000 (262.96 MB)  
Trainable params: 32,909,424 (87.58 MB)  
Non-trainable params: 6,144 (213.00 KB)  
Optimizer params: 3,020,000 (175.17 MB)

```
#training model
model.fit(train_epochs=15, validation_data=val_steps_per_epoch=(train, validation_steps=len(val)))
```

Epoch 1/15	218s 312ms/step - accuracy: 0.0864 - loss: 3.5157 - val_accuracy: 0.1547 - val_loss: 3.3446
Epoch 2/15	118s 230ms/step - accuracy: 0.3386 - loss: 2.1110 - val_accuracy: 0.6302 - val_loss: 1.2434
Epoch 3/15	140s 226ms/step - accuracy: 0.5487 - loss: 1.4666 - val_accuracy: 0.7736 - val_loss: 0.7724
Epoch 4/15	141s 224ms/step - accuracy: 0.6380 - loss: 1.2308 - val_accuracy: 0.8189 - val_loss: 0.6450
Epoch 5/15	140s 220ms/step - accuracy: 0.6667 - loss: 1.1921 - val_accuracy: 0.8189 - val_loss: 0.6153
Epoch 6/15	141s 227ms/step - accuracy: 0.6930 - loss: 0.9755 - val_accuracy: 0.8491 - val_loss: 0.6888
Epoch 7/15	140s 228ms/step - accuracy: 0.7252 - loss: 0.9184 - val_accuracy: 0.8415 - val_loss: 0.5453
Epoch 8/15	140s 228ms/step - accuracy: 0.7445 - loss: 0.8522 - val_accuracy: 0.8177 - val_loss: 0.5463
Epoch 9/15	138s 211ms/step - accuracy: 0.7534 - loss: 0.8202 - val_accuracy: 0.8830 - val_loss: 0.5018
Epoch 10/15	143s 213ms/step - accuracy: 0.7426 - loss: 0.8301 - val_accuracy: 0.8528 - val_loss: 0.5639
Epoch 11/15	140s 229ms/step - accuracy: 0.7687 - loss: 0.7741 - val_accuracy: 0.8981 - val_loss: 0.3649
Epoch 12/15	142s 228ms/step - accuracy: 0.7600 - loss: 0.7792 - val_accuracy: 0.8792 - val_loss: 0.4521
Epoch 13/15	143s 231ms/step - accuracy: 0.7819 - loss: 0.7279 - val_accuracy: 0.8415 - val_loss: 0.6906
Epoch 14/15	142s 230ms/step - accuracy: 0.7906 - loss: 0.6921 - val_accuracy: 0.9057 - val_loss: 0.3694
Epoch 15/15	141s 228ms/step - accuracy: 0.8072 - loss: 0.6436 - val_accuracy: 0.9132 - val_loss: 0.3957

clearing call backs.history.History at 0x7a2a423c5180

Accuracy: **0.8072**  
Loss: **0.6436**  
Validation Accuracy: **0.9132**  
Validation Loss: **0.3975**

## Conclusion:

After evaluating all three models — VGG16, Inception V3, and Xception — on the basis of training and validation performance, Xception clearly outperformed the others. It achieved the highest validation accuracy (94.27%) and the lowest validation loss (0.1979), indicating better generalization and learning efficiency. Also, it maintained low training loss with high accuracy, showing stable convergence without overfitting.

VGG16, despite high training accuracy (91.34%), showed a significant drop in validation accuracy (83.77%) and a higher validation loss, which suggests **overfitting** — it learned training data too well but failed to generalize.

Inception V3 had a validation accuracy (91.32%) higher than its training accuracy (80.72%), which is unusual and indicates possible **underfitting** or insufficient training — the model wasn't able to capture enough patterns from the training data.

Hence, Xception was selected as the final model for this project due to its superior performance and computational efficiency.