

## Model Development Phase Template

Date	20 April 2024
Team ID	738184
Project Title	Eye Disease Detection using 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):

#### CNN Model:

```
# Define the CNN model architecture
def create_cnn_model(input_shape, num_classes):
    model = models.Sequential()

    # Add convolutional layers
    model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=input_shape))
    model.add(layers.MaxPooling2D((2, 2)))
    model.add(layers.Conv2D(64, (3, 3), activation='relu'))
    model.add(layers.MaxPooling2D((2, 2)))
    model.add(layers.Conv2D(64, (3, 3), activation='relu'))

    # Flatten the output and add dense layers
    model.add(layers.Flatten())
    model.add(layers.Dense(64, activation='relu', kernel_initializer='random_uniform'))
    model.add(layers.Dense(num_classes, activation='softmax',)) # Output layer

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

## XCEPTION :

```
# Load pre-trained Xception model (excluding top layers)
base_model = Xception(weights='imagenet', include_top=False, input_shape=input_shape)

# Freeze the pre-trained layers
for layer in base_model.layers:
    layer.trainable = False

# Add custom classification head on top of the base Xception model
x = GlobalAveragePooling2D()(base_model.output)
x = Dense(256, activation='relu')(x)
predictions = Dense(num_classes, activation='softmax')(x)

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

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

## VGG19:

```
vgg = VGG19(include_top=False, input_shape=(64, 64, 3), weights='imagenet')
```

Downloading data from <https://storage.googleapis.com/tensorflow/keras-apps/80134624/80134624> [=====] - 4s 0us/step

```
for layer in vgg.layers:
    layer.trainable=False
```

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

```
output = Dense(4, activation='softmax')(x)
```

```
vgg19 = Model(vgg.input, output)
```

## INCEPTION:

```
# Load pre-trained Xception model (excluding top layers)
base_model = InceptionV3(weights='imagenet', include_top=False, input_shape=input_shape)

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/inception_v3/inception_v3_weights_tf_dim_ordering_tf_kernels_notop.h5
87910968/87910968 [=====] - 5s 0us/step

# Freeze the pre-trained layers
for layer in base_model.layers:
    layer.trainable = False

# Add custom classification head on top of the base Xception model
x = GlobalAveragePooling2D()(base_model.output)
x = Dense(256, activation='relu')(x)
predictions = Dense(num_classes, activation='softmax')(x)

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

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

## Model Validation and Evaluation Report (5 marks):

Model	Summary	Training and Validation Performance Metrics																											
CNN	<p>Model: "sequential"</p> <table> <thead> <tr> <th>Layer (type)</th><th>Output Shape</th><th>Param #</th></tr> </thead> <tbody> <tr> <td>conv2d (Conv2D)</td><td>(None, 62, 62, 32)</td><td>896</td></tr> <tr> <td>max_pooling2d (MaxPooling2D)</td><td>(None, 31, 31, 32)</td><td>0</td></tr> <tr> <td>conv2d_1 (Conv2D)</td><td>(None, 29, 29, 64)</td><td>18496</td></tr> <tr> <td>max_pooling2d_1 (MaxPooling2D)</td><td>(None, 14, 14, 64)</td><td>0</td></tr> <tr> <td>conv2d_2 (Conv2D)</td><td>(None, 12, 12, 64)</td><td>36928</td></tr> <tr> <td>flatten (Flatten)</td><td>(None, 9216)</td><td>0</td></tr> <tr> <td>dense (Dense)</td><td>(None, 64)</td><td>589888</td></tr> <tr> <td>dense_1 (Dense)</td><td>(None, 4)</td><td>260</td></tr> </tbody> </table> <p>                     Total params: 646468 (2.47 MB)                      Trainable params: 646468 (2.47 MB)                      Non-trainable params: 0 (0.00 Byte)                 </p>	Layer (type)	Output Shape	Param #	conv2d (Conv2D)	(None, 62, 62, 32)	896	max_pooling2d (MaxPooling2D)	(None, 31, 31, 32)	0	conv2d_1 (Conv2D)	(None, 29, 29, 64)	18496	max_pooling2d_1 (MaxPooling2D)	(None, 14, 14, 64)	0	conv2d_2 (Conv2D)	(None, 12, 12, 64)	36928	flatten (Flatten)	(None, 9216)	0	dense (Dense)	(None, 64)	589888	dense_1 (Dense)	(None, 4)	260	<pre>eye_detection = cnn_model.fit_generator(     x_train,     steps_per_epoch=len(x_train),     epochs=80,     validation_data=x_test,     validation_steps=len(x_test) )  Epoch 45/80 180/180 [=====] - 28s 267ms/step - loss: 0.2415 - accuracy: 0.9885 - val_loss: 0.4971 - val_accuracy: 0.8228 Epoch 46/80 180/180 [=====] - 27s 253ms/step - loss: 0.2142 - accuracy: 0.9177 - val_loss: 0.5679 - val_accuracy: 0.8145 Epoch 47/80 180/180 [=====] - 27s 254ms/step - loss: 0.2185 - accuracy: 0.9159 - val_loss: 0.5472 - val_accuracy: 0.8240 Epoch 48/80 180/180 [=====] - 28s 261ms/step - loss: 0.2840 - accuracy: 0.9224 - val_loss: 0.5208 - val_accuracy: 0.8121 Epoch 49/80 180/180 [=====] - 27s 254ms/step - loss: 0.2893 - accuracy: 0.9182 - val_loss: 0.5556 - val_accuracy: 0.8157 Epoch 50/80 180/180 [=====] - 28s 267ms/step - loss: 0.1935 - accuracy: 0.9259 - val_loss: 0.5644 - val_accuracy: 0.8098 Epoch 51/80 180/180 [=====] - 28s 258ms/step - loss: 0.2858 - accuracy: 0.9159 - val_loss: 0.4586 - val_accuracy: 0.8371 Epoch 52/80 180/180 [=====] - 27s 253ms/step - loss: 0.1740 - accuracy: 0.9322 - val_loss: 0.6077 - val_accuracy: 0.8285</pre>
Layer (type)	Output Shape	Param #																											
conv2d (Conv2D)	(None, 62, 62, 32)	896																											
max_pooling2d (MaxPooling2D)	(None, 31, 31, 32)	0																											
conv2d_1 (Conv2D)	(None, 29, 29, 64)	18496																											
max_pooling2d_1 (MaxPooling2D)	(None, 14, 14, 64)	0																											
conv2d_2 (Conv2D)	(None, 12, 12, 64)	36928																											
flatten (Flatten)	(None, 9216)	0																											
dense (Dense)	(None, 64)	589888																											
dense_1 (Dense)	(None, 4)	260																											

## VGG19

Model: "model"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 64, 64, 3)]	0
block1_conv1 (Conv2D)	(None, 64, 64, 64)	1792
block1_conv2 (Conv2D)	(None, 64, 64, 64)	36928
block1_pool (MaxPooling2D)	(None, 32, 32, 64)	0
block2_conv1 (Conv2D)	(None, 32, 32, 128)	73856
block2_conv2 (Conv2D)	(None, 32, 32, 128)	147584
block2_pool (MaxPooling2D)	(None, 16, 16, 128)	0
block3_conv1 (Conv2D)	(None, 16, 16, 256)	295168
block3_conv2 (Conv2D)	(None, 16, 16, 256)	590080
block3_conv3 (Conv2D)	(None, 16, 16, 256)	590080
block3_conv4 (Conv2D)	(None, 16, 16, 256)	590080
block3_pool (MaxPooling2D)	(None, 8, 8, 256)	0

block4_conv1 (Conv2D)	(None, 8, 8, 512)	1180160
block4_conv2 (Conv2D)	(None, 8, 8, 512)	2359808
block4_conv3 (Conv2D)	(None, 8, 8, 512)	2359808
block4_conv4 (Conv2D)	(None, 8, 8, 512)	2359808
block4_pool (MaxPooling2D)	(None, 4, 4, 512)	0
block5_conv1 (Conv2D)	(None, 4, 4, 512)	2359808
block5_conv2 (Conv2D)	(None, 4, 4, 512)	2359808
block5_conv3 (Conv2D)	(None, 4, 4, 512)	2359808
block5_conv4 (Conv2D)	(None, 4, 4, 512)	2359808
block5_pool (MaxPooling2D)	(None, 2, 2, 512)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 4)	8196

=====  
Total params: 20032580 (76.42 MB)  
Trainable params: 8196 (32.02 KB)  
Non-trainable params: 20024384 (76.39 MB)  
=====

```
vgg_detection=vgg19.fit(x_train,validation_data=x_test,epochs=50,steps_per_epoch=len(x_train),validation_steps=len(x_test))
```

```
100/100 [====...] - 9s 83ms/step - loss: 0.5733 - accuracy: 0.7687 - val_loss: 0.7085 - val_accuracy: 0.6968  
Epoch 42/50  
100/100 [====...] - 8s 79ms/step - loss: 0.5660 - accuracy: 0.7737 - val_loss: 0.7087 - val_accuracy: 0.7241  
Epoch 43/50  
100/100 [====...] - 8s 78ms/step - loss: 0.5610 - accuracy: 0.7778 - val_loss: 0.6443 - val_accuracy: 0.7515  
Epoch 44/50  
100/100 [====...] - 9s 80ms/step - loss: 0.5603 - accuracy: 0.7716 - val_loss: 0.6670 - val_accuracy: 0.7467  
Epoch 45/50  
100/100 [====...] - 9s 84ms/step - loss: 0.5595 - accuracy: 0.7796 - val_loss: 0.6887 - val_accuracy: 0.7325  
Epoch 46/50  
100/100 [====...] - 8s 79ms/step - loss: 0.5468 - accuracy: 0.7796 - val_loss: 0.6679 - val_accuracy: 0.7337  
Epoch 47/50  
100/100 [====...] - 8s 79ms/step - loss: 0.5581 - accuracy: 0.7755 - val_loss: 0.7248 - val_accuracy: 0.7158  
Epoch 48/50  
100/100 [====...] - 9s 83ms/step - loss: 0.5516 - accuracy: 0.7775 - val_loss: 0.6047 - val_accuracy: 0.7325  
Epoch 49/50  
100/100 [====...] - 9s 88ms/step - loss: 0.5494 - accuracy: 0.7814 - val_loss: 0.6127 - val_accuracy: 0.7693  
Epoch 50/50  
100/100 [====...] - 9s 88ms/step - loss: 0.5538 - accuracy: 0.7882 - val_loss: 0.6700 - val_accuracy: 0.7360
```

## XCEPTION

Model: "model\_1"

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	[(None, 299, 299, 3)]	0	[]
block1_conv1 (Conv2D)	(None, 149, 149, 32)	864	['input_2[0][0]']
block1_conv1_bn (Batch Normalization)	(None, 149, 149, 32)	128	['block1_conv1[0][0]']
block1_conv1_act (Activation)	(None, 149, 149, 32)	0	['block1_conv1_bn[0][0]']
block1_conv2 (Conv2D)	(None, 147, 147, 64)	18432	['block1_conv1_act[0][0]']
block1_conv2_bn (Batch Normalization)	(None, 147, 147, 64)	256	['block1_conv2[0][0]']
block1_conv2_act (Activation)	(None, 147, 147, 64)	0	['block1_conv2_bn[0][0]']
block2_sepconv1 (Separable Conv2D)	(None, 147, 147, 128)	8768	['block1_conv2_act[0][0]']
block2_sepconv1_bn (Batch Normalization)	(None, 147, 147, 128)	512	['block2_sepconv1[0][0]']

conv3d_4 (Conv2D)	(None, 74, 74, 128)	8192	['block2_sepconv1_act[0][0]']
block2_pool (MaxPooling2D)	(None, 74, 74, 128)	0	['block2_sepconv2_bn[0][0]']
batch_normalization_4 (Batch Normalization)	(None, 74, 74, 128)	512	['conv3d_4[0][0]']
add_12 (Add)	(None, 74, 74, 128)	0	['block2_pool[0][0]'] ['batch_normalization_4[0][0]']
block3_sepconv1_act (Activation)	(None, 74, 74, 128)	0	['add_12[0][0]']
block3_sepconv1 (Separable Conv2D)	(None, 74, 74, 256)	33920	['block3_sepconv1_act[0][0]']
block3_sepconv1_bn (Batch Normalization)	(None, 74, 74, 256)	1024	['block3_sepconv1[0][0]']
block3_sepconv2_act (Activation)	(None, 74, 74, 256)	0	['block3_sepconv1_bn[0][0]']
block3_sepconv2 (Separable Conv2D)	(None, 74, 74, 256)	67840	['block3_sepconv2_act[0][0]']

```
# Train the model  
xception_model = model.fit(  
    x_train,  
    steps_per_epoch=50,  
    epochs=epochs,  
    validation_data=x_test,  
    validation_steps=len(x_test)  
)
```

```
Epoch 43/50  
50/50 [====...] - 45s 899ms/step - loss: 0.2854 - accuracy: 0.8881 - val_loss: 0.5200 - val_accuracy: 0.7979  
Epoch 44/50  
50/50 [====...] - 46s 922ms/step - loss: 0.3805 - accuracy: 0.8831 - val_loss: 0.5697 - val_accuracy: 0.7824  
Epoch 45/50  
50/50 [====...] - 45s 895ms/step - loss: 0.2791 - accuracy: 0.8925 - val_loss: 0.5734 - val_accuracy: 0.7776  
Epoch 46/50  
50/50 [====...] - 45s 905ms/step - loss: 0.2974 - accuracy: 0.8881 - val_loss: 0.6129 - val_accuracy: 0.7717  
Epoch 47/50  
50/50 [====...] - 48s 970ms/step - loss: 0.2858 - accuracy: 0.8927 - val_loss: 0.5250 - val_accuracy: 0.8038  
Epoch 48/50  
50/50 [====...] - 49s 980ms/step - loss: 0.2578 - accuracy: 0.9025 - val_loss: 0.6442 - val_accuracy: 0.7788  
Epoch 49/50  
50/50 [====...] - 52s 1s/step - loss: 0.2850 - accuracy: 0.8906 - val_loss: 0.7857 - val_accuracy: 0.7827  
Epoch 50/50  
50/50 [====...] - 45s 900ms/step - loss: 0.3116 - accuracy: 0.8786 - val_loss: 0.7661 - val_accuracy: 0.7899
```

	<pre> block14_sepconv1 (Separabl (None, 10, 10, 1536) 1582080 ['add_23[0][0]'] eConv2D) block14_sepconv1_bn (Batch (None, 10, 10, 1536) 6144 ['block14_sepconv1[0][0]'] Normalization) block14_sepconv1_act (Acti (None, 10, 10, 1536) 0 ['block14_sepconv1_bn[0][0]'] vation) block14_sepconv2 (Separabl (None, 10, 10, 2048) 3159552 ['block14_sepconv1_act[0][0]'] eConv2D) block14_sepconv2_bn (Batch (None, 10, 10, 2048) 8192 ['block14_sepconv2[0][0]'] Normalization) block14_sepconv2_act (Acti (None, 10, 10, 2048) 0 ['block14_sepconv2_bn[0][0]'] vation) global_average_pooling2d_1 (None, 2048) 0 ['block14_sepconv2_act[0][0]'] (GlobalAveragePooling2D) dense_2 (Dense) (None, 256) 524544 ['global_average_pooling2d_1[0][0]'] dense_3 (Dense) (None, 4) 1028 ['dense_2[0][0]'] ----- Total params: 21307862 (81.59 MB) Trainable params: 525572 (2.00 MB) Non-trainable params: 20881488 (79.58 MB) </pre>	
INCEPTION	<pre> Model: "model" Layer (type) Output Shape Param # Connected to ----- input_1 (InputLayer) [(None, 299, 299, 3)] 0 [] conv2d (Conv2D) (None, 149, 149, 32) 864 ['input_1[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]'] conv2d_2 (Conv2D) (None, 147, 147, 64) 18432 ['activation_1[0][0]'] batch_normalization_2 (Bat (None, 147, 147, 64) 192 ['conv2d_2[0][0]'] chNormalization) activation_2 (Activation) (None, 147, 147, 64) 0 ['batch_normalization_2[0][0]']  max_pooling2d (MaxPooling2 (None, 73, 73, 64) 0 ['activation_2[0][0]'] D) conv2d_3 (Conv2D) (None, 73, 73, 80) 5120 ['max_pooling2d[0][0]'] batch_normalization_3 (Bat (None, 73, 73, 80) 240 ['conv2d_3[0][0]'] chNormalization) activation_3 (Activation) (None, 73, 73, 80) 0 ['batch_normalization_3[0][0]'] conv2d_4 (Conv2D) (None, 71, 71, 192) 138240 ['activation_3[0][0]'] batch_normalization_4 (Bat (None, 71, 71, 192) 576 ['conv2d_4[0][0]'] chNormalization) activation_4 (Activation) (None, 71, 71, 192) 0 ['batch_normalization_4[0][0]']  max_pooling2d_1 (MaxPoolin (None, 35, 35, 192) 0 ['activation_4[0][0]'] gD) conv2d_5 (Conv2D) (None, 35, 35, 64) 12288 ['max_pooling2d_1[0][0]'] batch_normalization_5 (Bat (None, 35, 35, 64) 192 ['conv2d_5[0][0]'] chNormalization) activation_5 (Activation) (None, 35, 35, 64) 0 ['batch_normalization_5[0][0]']  conv2d_6 (Conv2D) (None, 35, 35, 48) 9216 ['max_pooling2d_1[0][0]'] conv2d_9 (Conv2D) (None, 35, 35, 96) 55296 ['activation_5[0][0]'] batch_normalization_6 (Bat (None, 35, 35, 48) 144 ['conv2d_6[0][0]'] chNormalization) batch_normalization_9 (Bat (None, 35, 35, 96) 288 ['conv2d_9[0][0]'] chNormalization) activation_6 (Activation) (None, 35, 35, 48) 0 ['batch_normalization_6[0]'] activation_9 (Activation) (None, 35, 35, 96) 0 ['batch_normalization_9[0]']  average_pooling2d (Average (None, 35, 35, 192) 0 ['max_pooling2d_1[0][0]'] Pooling2D) conv2d_5 (Conv2D) (None, 35, 35, 64) 12288 ['max_pooling2d_1[0][0]'] conv2d_7 (Conv2D) (None, 35, 35, 64) 76800 ['activation_6[0][0]'] conv2d_10 (Conv2D) (None, 35, 35, 96) 82944 ['activation_9[0][0]'] conv2d_11 (Conv2D) (None, 35, 35, 32) 6144 ['average_pooling2d[0][0]'] batch_normalization_5 (Bat (None, 35, 35, 64) 192 ['conv2d_5[0][0]'] chNormalization) </pre>	<pre> # Train the model detection_inception = model.fit(     train_generator,     steps_per_epoch=len(x_train),     epochs=epochs,     validation_data=validation_generator,     validation_steps=len(validation_generator) ) </pre> <pre> Epoch 18/50 180/180 [=====] - 80s 758ms/step - loss: 0.4246 - accuracy: 0.8344 - val_loss: 0.7216 - val_accuracy: 0.7158 Epoch 19/50 180/180 [=====] - 81s 767ms/step - loss: 0.3880 - accuracy: 0.8403 - val_loss: 0.6388 - val_accuracy: 0.7776 Epoch 20/50 180/180 [=====] - 81s 766ms/step - loss: 0.4111 - accuracy: 0.8395 - val_loss: 0.5755 - val_accuracy: 0.7919 Epoch 21/50 180/180 [=====] - 81s 761ms/step - loss: 0.4059 - accuracy: 0.8409 - val_loss: 0.6152 - val_accuracy: 0.7586 Epoch 22/50 180/180 [=====] - 81s 761ms/step - loss: 0.3722 - accuracy: 0.8516 - val_loss: 0.6190 - val_accuracy: 0.7384 Epoch 23/50 180/180 [=====] - 80s 758ms/step - loss: 0.3923 - accuracy: 0.8418 - val_loss: 0.5547 - val_accuracy: 0.7812 Epoch 24/50 180/180 [=====] - 80s 758ms/step - loss: 0.3980 - accuracy: 0.8451 - val_loss: 0.5936 - val_accuracy: 0.7776 Epoch 25/50 180/180 [=====] - 80s 756ms/step - loss: 0.3543 - accuracy: 0.8643 - val_loss: 0.6121 - val_accuracy: 0.7515 Epoch 26/50 180/180 [=====] - 81s 765ms/step - loss: 0.3726 - accuracy: 0.8516 - val_loss: 0.6277 - val_accuracy: 0.7551 Epoch 27/50 180/180 [=====] - 81s 768ms/step - loss: 0.3653 - accuracy: 0.8587 - val_loss: 0.5854 - val_accuracy: 0.7574 Epoch 28/50 180/180 [=====] - 82s 772ms/step - loss: 0.3589 - accuracy: 0.8540 - val_loss: 0.5980 - val_accuracy: 0.7693 Epoch 29/50 180/180 [=====] - 82s 778ms/step - loss: 0.3416 - accuracy: 0.8682 - val_loss: 0.5887 - val_accuracy: 0.7785 </pre>

	<pre> activation_85 (Activation) (None, 8, 8, 320)      0      ['batch_normalization_85[0][0]'] mixed9_3 (Concatenate)      (None, 8, 8, 768)      0      ['activation_87[0][0]',  'activation_88[0][0]'] concatenate_1 (Concatenate) (None, 8, 8, 768)      0      ['activation_91[0][0]',  'activation_92[0][0]'] activation_93 (Activation) (None, 8, 8, 192)      0      ['batch_normalization_93[0][0]'] mixed10 (Concatenate)      (None, 8, 8, 2048)     0      ['activation_85[0][0]',  'mixed9_3[0][0]',  'concatenate_1[0][0]',  'activation_93[0][0]'] global_average_pooling2d ( (None, 2048)      0      ['mixed10[0][0]'] GlobalAveragePooling2D) dense (Dense)      (None, 256)      524544      ['global_average_pooling2d[0][0]'] dense_1 (Dense)      (None, 4)      1028      ['dense[0][0]'] ===== Total params: 22328556 (85.18 MB) Trainable params: 525572 (2.00 MB) Non-trainable params: 21802784 (83.17 MB) </pre>	
--	--	--