Project Development Phase
Model Performance Test
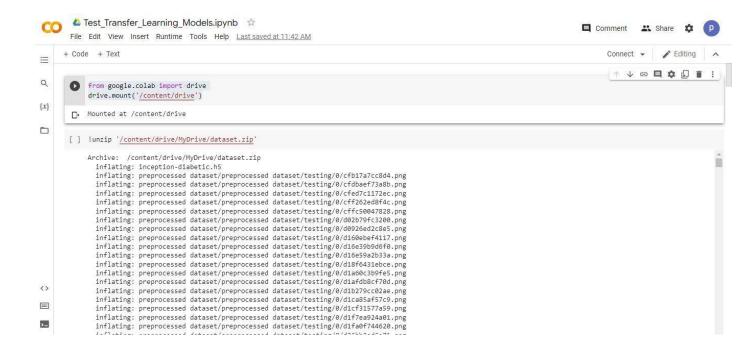
Date	17 November 2022
Team ID	PNT2022TMID26245
Project Name	Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy
Maximum Marks	10 Marks

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Model Summary	Total params: 21,885,485 Trainable params: 1,024,005 Non-trainable params: 20,861,480	Attached below
2.	Accuracy	Training Accuracy - 72% Validation Accuracy - 59%	Attached below
3.	Confidence Score (Only Yolo Projects)	Class Detected - NILL Confidence Score - NILL	NILL

<u>SCREENSHOTS</u> :

```python
from tensorflow.keras.layers import Dense, Flatten, Input
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator, load_img
from tensorflow.keras.applications.xception import Xception, preprocess_input
from glob import glob
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.models import load_model
from tensorflow.keras.models import load_model
#import image class to process the images
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.inception_v3 import preprocess_input
import numpy as np
```

```python
imageSize = [299, 299]
```

```python
xception = Xception(input_shape=imageSize + [3], weights='imagenet',include_top=False)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/xception/xception_weights_tf_dim_ordering_tf_kernels_notop.h5
83683744/83683744 [==============================] - 1s 0us/step
```

```python
# don't train existing weights
for layer in xception.layers:
```

```python
# our layers - you can add more if you want
x = Flatten()(xception.output)
```

```python
prediction = Dense(5, activation='softmax')(x)
```

```python
# create a model object
model = Model(inputs=xception.input, outputs=prediction)
```

```python
# view the structure of the model
model.summary()
```

```
Model: "model"
_____
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)           [(None, 299, 299, 3  0           []
                                )]

 block1_conv1 (Conv2D)          (None, 149, 149, 32  864         ['input_1[0][0]']
                                )

 block1_conv1_bn (BatchNormaliz (None, 149, 149, 32  128         ['block1_conv1[0][0]']
 ation)                         )

 block1_conv1_act (Activation)  (None, 149, 149, 32  0           ['block1_conv1_bn[0][0]']
                                )
```

+ Code   + Text

```python
# tell the model what cost and optimization method to use
model.compile(
    loss='categorical_crossentropy',
    optimizer='adam',
    metrics=['accuracy']
)
```

```python
train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   horizontal_flip = True)

test_datagen = ImageDataGenerator(rescale = 1./255)
```

```python
training_set = train_datagen.flow_from_directory('/content/preprocessed dataset/preprocessed dataset/training',
                                                 target_size = (299, 299),
                                                 batch_size = 32,
                                                 class_mode = 'categorical')

test_set = test_datagen.flow_from_directory('/content/preprocessed dataset/preprocessed dataset/testing',
                                            target_size = (299, 299),
                                            batch_size = 32,
                                            class_mode = 'categorical')
```

```
Found 3662 images belonging to 5 classes.
Found 734 images belonging to 5 classes.
```

+ Code   + Text   Connect ▾   ✎ Editing   ^

```python
r = model.fit_generator(
    training_set,
    validation_data=test_set,
    epochs=30,
    steps_per_epoch=len(training_set)//32,
    validation_steps=len(test_set)//32
)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:6: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please

Epoch 1/30
3/3 [==============================] - 36s 10s/step - loss: 11.1276 - accuracy: 0.3958
Epoch 2/30
3/3 [==============================] - 30s 9s/step - loss: 12.4058 - accuracy: 0.5833
Epoch 3/30
3/3 [==============================] - 30s 9s/step - loss: 9.2441 - accuracy: 0.4583
Epoch 4/30
3/3 [==============================] - 29s 9s/step - loss: 7.6886 - accuracy: 0.5938
Epoch 5/30
3/3 [==============================] - 31s 9s/step - loss: 4.8011 - accuracy: 0.6875
Epoch 6/30
3/3 [==============================] - 29s 9s/step - loss: 3.5280 - accuracy: 0.6562
Epoch 7/30
3/3 [==============================] - 31s 9s/step - loss: 5.4394 - accuracy: 0.6458
Epoch 8/30
3/3 [==============================] - 29s 8s/step - loss: 3.8816 - accuracy: 0.6562
Epoch 9/30
3/3 [==============================] - 29s 8s/step - loss: 5.4999 - accuracy: 0.6562
Epoch 10/30
3/3 [==============================] - 30s 9s/step - loss: 2.3537 - accuracy: 0.7708
```

```
[ ]  model.save('Updated-Xception-diabetic-retinopathy.h5')
```

```
▶  #load saved model file
   model=load_model('Updated-Xception-diabetic-retinopathy.h5')
```

```
[ ]  #load one random image from local system
     img=image.load_img(r'/content/drive/MyDrive/d1ca85af57c9.png',target_size=(299,299))
```

```
[ ]  x=image.img_to_array(img)
```

```
[ ]  import matplotlib.pyplot as plt
```

```
[ ]  x.shape
```

```
(299, 299, 3)
```

```
[ ]  import numpy as np
     x=np.expand_dims(x,axis=0)
     img_data=preprocess_input(x)
     img_data.shape
```

```
(1, 299, 299, 3)
```

```
[ ]  img_data.shape
```

```
(1, 299, 299, 3)
```

```
[ ]  model.predict(img_data)
```

```
1/1 [==============================] - 1s 1s/step
array([[1.0000000e+00, 3.2964899e-14, 1.6276460e-19, 2.8887498e-18,
        1.2188903e-15]], dtype=float32)
```

```
[ ]  output=np.argmax(model.predict(img_data), axis=1)
```

```
1/1 [==============================] - 0s 262ms/step
```

```
[ ]  output==0,output==1,output==2,output==3,output==4
```

```
(array([ True]),
 array([False]),
 array([False]),
 array([False]),
 array([False]))
```

```
[ ]  index=['No Diabetic Retinopathy', 'Mild DR', 'Moderate DR', 'Severe DR', 'Proliferative DR']
     result = str(index[output[0]])
     result
```

```
'No Diabetic Retinopathy'
```