Model Building

Pre-trained CNN Model as feature extractor

```
[ ]  1  xception = Xception ( input_shape = img_size + [ 3 ] , weights = 'imagenet' , include_top = False )
     2  # don't train existing weights
     3  for layer in xception.layers :
     4      layer.trainable = False
     5
     6  x = Flatten ( ) ( xception.output )
```

Adding Dense Layers

```
[ ]  1  prediction = Dense ( 5 , activation = 'softmax' ) ( x )
     2  model = Model ( inputs = xception.input , outputs = prediction )
```

```
1  model.summary()
```

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

block1_conv1 (Conv2D)        (None, 149, 149, 32)      864

block1_conv1_bn (BatchNormalization)  (None, 149, 149, 32)  128

block1_conv1_act (Activation)  (None, 149, 149, 32)  0

block1_conv2 (Conv2D)        (None, 147, 147, 64)      18432

block1_conv2_bn (BatchNormalization)  (None, 147, 147, 64)  256
```

Configure the Learning Process

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

Train the model

```
# fit the model
r = model.fit(
 training_set ,
 validation_data = test_set ,
 epochs = 30 ,
 steps_per_epoch = len ( training_set ) // 32 ,
 validation_steps = len ( test_set ) // 32)
```

```
Epoch 1/30
3/3 [==============================] - 38s 11s/step - loss: 11.2760 - accuracy: 0.4062
Epoch 2/30
3/3 [==============================] - 30s 9s/step - loss: 14.2826 - accuracy: 0.6354
Epoch 3/30
3/3 [==============================] - 30s 9s/step - loss: 16.8780 - accuracy: 0.5208
Epoch 4/30
3/3 [==============================] - 30s 9s/step - loss: 9.6413 - accuracy: 0.5104
```

Save the Model

```
model.save ( "Updated-xception-diabetic-retinopathy.h5")
```