3. Adding Flatten Layer

```
for layer in vgg16.layers:
    layer.trainable = False

folders = glob('/content/drive/MyDrive/Intelligent Vehicle Damage Assessment
& Cost Estimator For Insurance Companies/Dataset/body/training/*')


folders


['/content/drive/MyDrive/Intelligent Vehicle Damage Assessment & Cost Estimat
or For Insurance Companies/Dataset/body/training/02-side',
 '/content/drive/MyDrive/Intelligent Vehicle Damage Assessment & Cost Estimat
or For Insurance Companies/Dataset/body/training/00-front',
 '/content/drive/MyDrive/Intelligent Vehicle Damage Assessment & Cost Estimat
or For Insurance Companies/Dataset/body/training/01-rear']


x = Flatten()(vgg16.output)


len(folders)
```

Out[ ]:

```
3
```

#==========Build a Model
model = tf.keras.models.Sequential()

model.add(keras.layers.Flatten(input_shape=(28, 28, 3)))#reshapes to (2352)=28x28x3
model.add(layers.experimental.preprocessing.Rescaling(1./255))#normalize
model.add(keras.layers.Dense(128,activation=tf.nn.relu))
model.add(keras.layers.Dense(2,activation=tf.nn.softmax))

model.build()
model.summary()# summary of the model

#==========Build a Model
tensor = tf.keras.backend.placeholder(dtype=tf.float32, shape=(None, 28, 28, 3))

model = tf.keras.models.Sequential()

```
    model.add(keras.layers.InputLayer(input_tensor=tensor))
    model.add(keras.layers.Reshape([2352]))
model.add(layers.experimental.preprocessing.Rescaling(1./255))#normalize
    model.add(keras.layers.Dense(128,activation=tf.nn.relu))
    model.add(keras.layers.Dense(2,activation=tf.nn.softmax))

    model.build()
    model.summary()# summary of the model
```

Flattening is converting the data into a 1-dimensional array for inputting it to the next layer. We flatten the output of the convolutional layers to create a single long feature vector. In some architectures, e.g. CNN an image is better processed by a neural network if it is in 1D form rather than 2D.