In [1]:
```
from google.colab import drive
drive.mount('/gdrive')
%cd /gdrive
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g
3pfee6491hc0brc4i.apps.googleusercontent.com&redirect_uri=urn%3aietf%3awg%3aoauth%3a2.0%3aoob&response_type=c
ode&scope=email%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdocs.test%20https%3a%2f%2fwww.googleapis.com%2fau
th%2fdrive%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive.photos.readonly%20https%3a%2f%2fwww.googleapis.c
om%2fauth%2fpeopleapi.readonly

Enter your authorization code:
··········
Mounted at /gdrive
/gdrive
```

In [2]:
```
pip install tensorflow-gpu==2.0.0
```

```
Collecting tensorflow-gpu==2.0.0
  Downloading https://files.pythonhosted.org/packages/25/44/47f0722aea081697143fbcf5d2aa60d1aee4aaacb5869aee2
b568974777b/tensorflow_gpu-2.0.0-cp36-cp36m-manylinux2010_x86_64.whl (380.8MB)
     |████████████████████████████████| 380.8MB 37kB/s
Collecting tensorflow-estimator<2.1.0,>=2.0.0
  Downloading https://files.pythonhosted.org/packages/fc/08/8b927337b7019c374719145d1dceba21a8bb909b93b1ad6f8
fb7d22c1ca1/tensorflow_estimator-2.0.1-py2.py3-none-any.whl (449kB)
     |████████████████████████████████| 450kB 45.2MB/s
Requirement already satisfied: google-pasta>=0.1.6 in /usr/local/lib/python3.6/dist-packages (from tensorflow
-gpu==2.0.0) (0.2.0)
Requirement already satisfied: six>=1.10.0 in /usr/local/lib/python3.6/dist-packages (from tensorflow-gpu==2.
0.0) (1.12.0)
Requirement already satisfied: wrapt>=1.11.1 in /usr/local/lib/python3.6/dist-packages (from tensorflow-gpu==
2.0.0) (1.12.1)
Requirement already satisfied: numpy<2.0,>=1.16.0 in /usr/local/lib/python3.6/dist-packages (from tensorflow-
gpu==2.0.0) (1.18.3)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.6/dist-packages (from tensorflow-gp
u==2.0.0) (1.1.0)
Requirement already satisfied: wheel>=0.26 in /usr/local/lib/python3.6/dist-packages (from tensorflow-gpu==2.
0.0) (0.34.2)
Collecting tensorboard<2.1.0,>=2.0.0
  Downloading https://files.pythonhosted.org/packages/76/54/99b9d5d52d5cb732f099baaaf7740403e83fe6b0cedde940f
abd2b13d75a/tensorboard-2.0.2-py3-none-any.whl (3.8MB)
     |████████████████████████████████| 3.8MB 41.7MB/s
Requirement already satisfied: keras-applications>=1.0.8 in /usr/local/lib/python3.6/dist-packages (from tens
orflow-gpu==2.0.0) (1.0.8)
Requirement already satisfied: protobuf>=3.6.1 in /usr/local/lib/python3.6/dist-packages (from tensorflow-gpu
==2.0.0) (3.10.0)
Requirement already satisfied: grpcio>=1.8.6 in /usr/local/lib/python3.6/dist-packages (from tensorflow-gpu==
2.0.0) (1.28.1)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.6/dist-packages (from tensorflow-g
pu==2.0.0) (3.2.1)
Collecting gast==0.2.2
  Downloading https://files.pythonhosted.org/packages/4e/35/11749bf99b2d4e3cceb4d55ca22590b0d7c2c62b9de38ac4a
4a7f4687421/gast-0.2.2.tar.gz
Requirement already satisfied: absl-py>=0.7.0 in /usr/local/lib/python3.6/dist-packages (from tensorflow-gpu=
=2.0.0) (0.9.0)
Requirement already satisfied: astor>=0.6.0 in /usr/local/lib/python3.6/dist-packages (from tensorflow-gpu==
2.0.0) (0.8.1)
Requirement already satisfied: keras-preprocessing>=1.0.5 in /usr/local/lib/python3.6/dist-packages (from ten
sorflow-gpu==2.0.0) (1.1.0)
Requirement already satisfied: setuptools>=41.0.0 in /usr/local/lib/python3.6/dist-packages (from tensorboard
<2.1.0,>=2.0.0->tensorflow-gpu==2.0.0) (46.1.3)
```

```
Requirement already satisfied: werkzeug>=0.11.15 in /usr/local/lib/python3.6/dist-packages (from tensorboard<
2.1.0,>=2.0.0->tensorflow-gpu==2.0.0) (1.0.1)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/local/lib/python3.6/dist-packages (fr
om tensorboard<2.1.0,>=2.0.0->tensorflow-gpu==2.0.0) (0.4.1)
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.6/dist-packages (from tensorboar
d<2.1.0,>=2.0.0->tensorflow-gpu==2.0.0) (2.21.0)
Requirement already satisfied: google-auth<2,>=1.6.3 in /usr/local/lib/python3.6/dist-packages (from tensorbo
ard<2.1.0,>=2.0.0->tensorflow-gpu==2.0.0) (1.7.2)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.6/dist-packages (from tensorboard<2.
1.0,>=2.0.0->tensorflow-gpu==2.0.0) (3.2.1)
Requirement already satisfied: h5py in /usr/local/lib/python3.6/dist-packages (from keras-applications>=1.0.8
->tensorflow-gpu==2.0.0) (2.10.0)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.6/dist-packages (from googl
e-auth-oauthlib<0.5,>=0.4.1->tensorboard<2.1.0,>=2.0.0->tensorflow-gpu==2.0.0) (1.3.0)
Requirement already satisfied: idna<2.9,>=2.5 in /usr/local/lib/python3.6/dist-packages (from requests<3,>=2.
21.0->tensorboard<2.1.0,>=2.0.0->tensorflow-gpu==2.0.0) (2.8)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.6/dist-packages (from requests<3,
>=2.21.0->tensorboard<2.1.0,>=2.0.0->tensorflow-gpu==2.0.0) (2020.4.5.1)
Requirement already satisfied: urllib3<1.25,>=1.21.1 in /usr/local/lib/python3.6/dist-packages (from requests
<3,>=2.21.0->tensorboard<2.1.0,>=2.0.0->tensorflow-gpu==2.0.0) (1.24.3)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in /usr/local/lib/python3.6/dist-packages (from requests
<3,>=2.21.0->tensorboard<2.1.0,>=2.0.0->tensorflow-gpu==2.0.0) (3.0.4)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.6/dist-packages (from google-a
uth<2,>=1.6.3->tensorboard<2.1.0,>=2.0.0->tensorflow-gpu==2.0.0) (0.2.8)
Requirement already satisfied: rsa<4.1,>=3.1.4 in /usr/local/lib/python3.6/dist-packages (from google-auth<2,
>=1.6.3->tensorboard<2.1.0,>=2.0.0->tensorflow-gpu==2.0.0) (4.0)
Requirement already satisfied: cachetools<3.2,>=2.0.0 in /usr/local/lib/python3.6/dist-packages (from google-
auth<2,>=1.6.3->tensorboard<2.1.0,>=2.0.0->tensorflow-gpu==2.0.0) (3.1.1)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.6/dist-packages (from requests-oauth
lib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tensorboard<2.1.0,>=2.0.0->tensorflow-gpu==2.0.0) (3.1.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/python3.6/dist-packages (from pyasn1-mo
dules>=0.2.1->google-auth<2,>=1.6.3->tensorboard<2.1.0,>=2.0.0->tensorflow-gpu==2.0.0) (0.4.8)
Building wheels for collected packages: gast
  Building wheel for gast (setup.py) ... done
  Created wheel for gast: filename=gast-0.2.2-cp36-none-any.whl size=7540 sha256=acebd4a5577b451073b798626965
4533b2c226a536ad180e264d6bbd8134b8f4
  Stored in directory: /root/.cache/pip/wheels/5c/2e/7e/a1d4d4fcebe6c381f378ce7743a3ced3699feb89bcfbdadadd
Successfully built gast
ERROR: tensorflow 2.2.0rc3 has requirement gast==0.3.3, but you'll have gast 0.2.2 which is incompatible.
ERROR: tensorflow 2.2.0rc3 has requirement tensorboard<2.3.0,>=2.2.0, but you'll have tensorboard 2.0.2 which
is incompatible.
ERROR: tensorflow 2.2.0rc3 has requirement tensorflow-estimator<2.3.0,>=2.2.0rc0, but you'll have tensorflow-
estimator 2.0.1 which is incompatible.
```

```
ERROR: tensorflow-probability 0.10.0rc0 has requirement gast>=0.3.2, but you'll have gast 0.2.2 which is inco
mpatible.
Installing collected packages: tensorflow-estimator, tensorboard, gast, tensorflow-gpu
  Found existing installation: tensorflow-estimator 2.2.0
    Uninstalling tensorflow-estimator-2.2.0:
      Successfully uninstalled tensorflow-estimator-2.2.0
  Found existing installation: tensorboard 2.2.1
    Uninstalling tensorboard-2.2.1:
      Successfully uninstalled tensorboard-2.2.1
  Found existing installation: gast 0.3.3
    Uninstalling gast-0.3.3:
      Successfully uninstalled gast-0.3.3
Successfully installed gast-0.2.2 tensorboard-2.0.2 tensorflow-estimator-2.0.1 tensorflow-gpu-2.0.0
```

In [3]:
```
pip install keras==2.2.4
```

```
Collecting keras==2.2.4
  Downloading https://files.pythonhosted.org/packages/5e/10/aa32dad071ce52b5502266b5c659451cfd6ffcbf14e6c8c4f
16c0ff5aaab/Keras-2.2.4-py2.py3-none-any.whl (312kB)
     |████████████████████████████████| 317kB 3.5MB/s eta 0:00:01
Requirement already satisfied: numpy>=1.9.1 in /usr/local/lib/python3.6/dist-packages (from keras==2.2.4) (1.
18.3)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.6/dist-packages (from keras==2.2.4) (3.13)
Requirement already satisfied: h5py in /usr/local/lib/python3.6/dist-packages (from keras==2.2.4) (2.10.0)
Requirement already satisfied: scipy>=0.14 in /usr/local/lib/python3.6/dist-packages (from keras==2.2.4) (1.
4.1)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.6/dist-packages (from keras==2.2.4) (1.1
2.0)
Requirement already satisfied: keras-preprocessing>=1.0.5 in /usr/local/lib/python3.6/dist-packages (from ker
as==2.2.4) (1.1.0)
Requirement already satisfied: keras-applications>=1.0.6 in /usr/local/lib/python3.6/dist-packages (from kera
s==2.2.4) (1.0.8)
Installing collected packages: keras
  Found existing installation: Keras 2.3.1
    Uninstalling Keras-2.3.1:
      Successfully uninstalled Keras-2.3.1
Successfully installed keras-2.2.4
```

In [ ]:
```

```

In [4]:
```
gpu_info = !nvidia-smi
gpu_info = '\n'.join(gpu_info)
if gpu_info.find('failed') >= 0:
  print('Select the Runtime → "Change runtime type" menu to enable a GPU accelerator, ')
  print('and then re-execute this cell.')
else:
  print(gpu_info)
```

```
Fri Apr 24 19:34:45 2020
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 440.64.00    Driver Version: 418.67       CUDA Version: 10.1     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla P100-PCIE...  Off  | 00000000:00:04.0 Off |                    0 |
| N/A   42C    P0    30W / 250W |      0MiB / 16280MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

In [5]:
```
from keras.preprocessing.image import ImageDataGenerator
from keras.preprocessing.image import img_to_array
from keras.preprocessing.image import load_img
import numpy as np
import argparse
import cv2
import os
from imutils import paths
```

```
Using TensorFlow backend.
```

In [ ]:
```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import VGG16
from tensorflow.keras.applications.resnet_v2 import ResNet152V2
from tensorflow.keras.layers import AveragePooling2D
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Input
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.utils import to_categorical
from sklearn.preprocessing import LabelBinarizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from imutils import paths
import matplotlib.pyplot as plt
import numpy as np
import argparse
import cv2
import os
```

In [ ]:
```
x_train=np.load('My Drive/Colab Notebooks/215-FP_Dataset/numpy5/x_train.npy')
```

In [ ]:
```
y_train=np.load('My Drive/Colab Notebooks/215-FP_Dataset/numpy5/y_train.npy')
```

In [103]:
```

```

Out[103]:
```
array([[0.     , 0.00392],
       [0.     , 0.00392],
       [0.     , 0.00392],
       ...,
       [0.     , 0.00392],
       [0.     , 0.00392],
       [0.     , 0.00392]], dtype=float16)
```

In [ ]:
```

```

In [ ]:

In [24]:
```
y_train
```

Out[24]:
```
array([[1., 0.],
       [1., 0.],
       [1., 0.],
       ...,
       [1., 0.],
       [1., 0.],
       [1., 0.]])
```

In [ ]:
```
x_valid=np.load('My Drive/Colab Notebooks/215-FP_Dataset/numpy5/x_valid.npy')
```

In [ ]:
```
y_valid=np.load('My Drive/Colab Notebooks/215-FP_Dataset/numpy5/y_valid.npy')
```

In [ ]:

In [ ]:

In [27]:
```
y_valid
```

Out[27]:
```
array([[1., 0.],
       [1., 0.],
       [1., 0.],
       ...,
       [1., 0.],
       [1., 0.],
       [1., 0.]])
```

In [ ]:
```
y_train2=y_train
y_valid2=y_valid
```

In [ ]:
```
y_train2=np.argmax(y_train,axis=1)
y_valid2=np.argmax(y_valid,axis=1)
```

In [30]:
```
y_train2[:50]
```

Out[30]:
```
array([0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0,
       1, 1, 0, 1, 1, 1])
```

In [31]:
```
y_valid2[:50]
```

Out[31]:
```
array([0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0,
       0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 1, 0, 1, 1])
```

In [34]:
```
summation1=0
for i in range(len(x_valid)):
    if y_valid2[i]==0:
        summation1=summation1+1
summation1
```

Out[34]:
```
488
```

In [35]:
```
summation2=0
for i in range(len(x_train)):
    if y_train2[i]==0:
        summation2=summation2+1
summation2
```

Out[35]:
```
1951
```

In [36]:
```
len(x_train)
```

Out[36]:
```
6025
```

In [ ]:
```
#y_train=y_train2
y_train=((y_train)/255.).astype('float16')
```

In [ ]:
```
#y_valid=y_valid2
y_valid=((y_valid)/255.).astype('float16')
```

In [ ]:
```

```

In [ ]:

In [ ]:

In [44]:
```
from __future__ import print_function
from tensorflow.keras import backend as K
from tensorflow.keras.layers import Layer
from tensorflow.keras import activations
from tensorflow.keras import utils
from tensorflow.keras.models import Model
from tensorflow.keras.layers import *
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
import keras
from tensorflow.keras.callbacks import ModelCheckpoint
from tensorflow.keras import optimizers

K.set_image_data_format('channels_last')

def squash(x, axis=-1):
    s_squared_norm = K.sum(K.square(x), axis, keepdims=True) + K.epsilon()
    scale = K.sqrt(s_squared_norm) / (1 + s_squared_norm)
    return scale * x


def softmax(x, axis=-1):

    ex = K.exp(x - K.max(x, axis=axis, keepdims=True))
    return ex / K.sum(ex, axis=axis, keepdims=True)


def margin_loss(y_true, y_pred):

    lamb, margin = 0.5, 0.1
    return K.sum((y_true * K.square(K.relu(1 - margin - y_pred)) + lamb * (
        1 - y_true) * K.square(K.relu(y_pred - margin))), axis=-1)


class Capsule(Layer):


    def __init__(self,
                 num_capsule,
                 dim_capsule,
                 routings=3,
                 share_weights=True,
```

```
                 activation='squash',
                 **kwargs):
        super(Capsule, self).__init__(**kwargs)
        self.num_capsule = num_capsule
        self.dim_capsule = dim_capsule
        self.routings = routings
        self.share_weights = share_weights
        if activation == 'squash':
            self.activation = squash
        else:
            self.activation = activations.get(activation)

    def get_config(self):
        config = super().get_config().copy()
        config.update({
        'num_capsule':  self.num_capsule,
        'dim_capsule' : self.dim_capsule,
        'routings':  self.routings,
        'share_weight':self.share_weights,



        })
        return config

    def build(self, input_shape):
        input_dim_capsule = input_shape[-1]
        if self.share_weights:
            self.kernel = self.add_weight(
                name='capsule_kernel',
                shape=(1, input_dim_capsule,
                       self.num_capsule * self.dim_capsule),
                initializer='glorot_uniform',
                trainable=True)
        else:
            input_num_capsule = input_shape[-2]
            self.kernel = self.add_weight(
                name='capsule_kernel',
                shape=(input_num_capsule, input_dim_capsule,
                       self.num_capsule * self.dim_capsule),
                initializer='glorot_uniform',
                trainable=True)
```

```
    def call(self, inputs):


        if self.share_weights:
            hat_inputs = K.conv1d(inputs, self.kernel)
        else:
            hat_inputs = K.local_conv1d(inputs, self.kernel, [1], [1])

        batch_size = K.shape(inputs)[0]
        input_num_capsule = K.shape(inputs)[1]
        hat_inputs = K.reshape(hat_inputs,
                               (batch_size, input_num_capsule,
                                self.num_capsule, self.dim_capsule))
        hat_inputs = K.permute_dimensions(hat_inputs, (0, 2, 1, 3))

        b = K.zeros_like(hat_inputs[:, :, :, 0])
        for i in range(self.routings):
            c = softmax(b, 1)
            o = self.activation(keras.backend.batch_dot(c, hat_inputs, [2, 2]))
            if i < self.routings - 1:
                b = keras.backend.batch_dot(o, hat_inputs, [2, 3])
                if K.backend() == 'theano':
                    o = K.sum(o, axis=1)

        return o

    def compute_output_shape(self, input_shape):
        return (None, self.num_capsule, self.dim_capsule)



batch_size = 16
num_classes = 2
epochs = 100



#class weights to handle class imbalance

#class_weights = {0: 1-np.count_nonzero(y_train==0)/len(y_train),
            #    1: 1-np.count_nonzero(y_train==1)/len(y_train)}
```

```
input_image = Input(shape=(None, None, 3))
x = Conv2D(64, (3, 3), activation='relu',trainable = False)(input_image)
x=BatchNormalization(axis=-1, momentum=0.99, epsilon=0.001, center=True, scale=True, beta_initializer='zeros'
, gamma_initializer='ones', moving_mean_initializer='zeros', moving_variance_initializer='ones', beta_regular
izer=None, gamma_regularizer=None, beta_constraint=None, gamma_constraint=None)(x)
x = Conv2D(64, (3, 3), activation='relu',trainable = False)(x)
x = AveragePooling2D((2, 2),trainable = False)(x)
x = Conv2D(128, (3, 3), activation='relu',trainable = False)(x)
x = Conv2D(128, (3, 3), activation='relu',trainable = False)(x)
```

```
x = Reshape((-1, 128))(x)
x = Capsule(32, 8, 3, True)(x)
x = Capsule(32, 8, 3, True)(x)
capsule = Capsule(5, 16, 3, True)(x)
output = Lambda(lambda z: K.sqrt(K.sum(K.square(z), 2)))(capsule)
```

```
model = Model(inputs=[input_image], outputs=[output])

model.load_weights('My Drive/Colab Notebooks/215-FP_Dataset/pre-train.h5')

capsule2 = Capsule(2, 16, 3, True)(model.layers[-3].output)
output2 = Lambda(lambda z: K.sqrt(K.sum(K.square(z), 2)))(capsule2)


model2 = Model(inputs=[input_image], outputs=[output2])


adam = optimizers.Adam(lr=0.001)
model.compile(loss=margin_loss, optimizer=adam, metrics=['accuracy'])
model2.compile(loss=margin_loss, optimizer=adam, metrics=['accuracy'])
model2.summary()

data_augmentation = False
# The best model is selected based on the loss value on the validation set
```

```
filepath='My Drive/Colab Notebooks/215-FP_Dataset/after-train.h5'
checkpoint = ModelCheckpoint(filepath, monitor='val_loss', verbose=1, save_best_only=True, mode='min')
callbacks_list = [checkpoint]
```

```
if not data_augmentation:
    print('Not using data augmentation.')
    model2.fit(
        [x_train], [y_train],
        batch_size=batch_size,
        epochs=epochs,
        validation_data=[[x_valid], [y_valid]],
        shuffle=True,callbacks=callbacks_list)

else:
    print('Using real-time data augmentation.')
    # This will do preprocessing and realtime data augmentation:
    datagen = ImageDataGenerator(
        featurewise_center=False,  # set input mean to 0 over the dataset
        samplewise_center=False,  # set each sample mean to 0
        featurewise_std_normalization=False,  # divide inputs by dataset std
        samplewise_std_normalization=False,  # divide each input by its std
        zca_whitening=False,  # apply ZCA whitening
        zca_epsilon=1e-06,  # epsilon for ZCA whitening
        rotation_range=0.1,  # randomly rotate images in 0 to 180 degrees
        width_shift_range=0.1,  # randomly shift images horizontally
        height_shift_range=0.1,  # randomly shift images vertically
        brightness_range=[0.5,1.5],
        shear_range=0.1,  # set range for random shear
        zoom_range=0.1,  # set range for random zoom
        channel_shift_range=0.,  # set range for random chann el shifts
        # set mode for filling points outside the input boundaries
        fill_mode='nearest',
        cval=0.,  # value used for fill_mode = "constant"
        horizontal_flip=True,  # randomly flip images
        vertical_flip=True,  # randomly flip images
        # set rescaling factor (applied before any other transformation)
        rescale=None,
```

```
        # set function that will be applied on each input
        preprocessing_function=None,
        # image data format, either "channels_first" or "channels_last"
        data_format=None,
        # fraction of images reserved for validation (strictly between 0 and 1)
        validation_split=0.0)

    # Compute quantities required for feature-wise normalization
    # (std, mean, and principal components if ZCA whitening is applied).
    datagen.fit(x_train)

    # Fit the model on the batches generated by datagen.flow().
    model.fit(
        datagen.flow(x_train, y_train, batch_size=batch_size),
        epochs=epochs,
        validation_data=(x_valid, y_valid),shuffle=True)
```

```
Model: "model_19"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_10 (InputLayer)        [(None, None, None, 3)]   0
_________________________________________________________________
conv2d_36 (Conv2D)           (None, None, None, 64)    1792
_________________________________________________________________
batch_normalization_9 (Batch (None, None, None, 64)    256
_________________________________________________________________
conv2d_37 (Conv2D)           (None, None, None, 64)    36928
_________________________________________________________________
average_pooling2d_9 (Average (None, None, None, 64)    0
_________________________________________________________________
conv2d_38 (Conv2D)           (None, None, None, 128)   73856
_________________________________________________________________
conv2d_39 (Conv2D)           (None, None, None, 128)   147584
_________________________________________________________________
reshape_9 (Reshape)          (None, None, 128)         0
_________________________________________________________________
capsule_36 (Capsule)         (None, 32, 8)             32768
_________________________________________________________________
capsule_37 (Capsule)         (None, 32, 8)             2048
_________________________________________________________________
capsule_39 (Capsule)         (None, 2, 16)             256
_________________________________________________________________
lambda_19 (Lambda)           (None, 2)                 0
=================================================================
Total params: 295,488
Trainable params: 35,200
Non-trainable params: 260,288
_________________________________________________________________
Not using data augmentation.
Train on 6025 samples, validate on 1507 samples
Epoch 1/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0100 - accuracy: 0.5620
Epoch 00001: val_loss improved from inf to 0.00287, saving model to My Drive/Colab Notebooks/215-FP_Dataset/a
fter-train.h5
6025/6025 [==============================] - 22s 4ms/sample - loss: 0.0100 - accuracy: 0.5625 - val_loss: 0.0
029 - val_accuracy: 0.6105
Epoch 2/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0027 - accuracy: 0.6420
Epoch 00002: val_loss improved from 0.00287 to 0.00269, saving model to My Drive/Colab Notebooks/215-FP_Datas
```

```
et/after-train.h5
6025/6025 [==============================] - 25s 4ms/sample - loss: 0.0027 - accuracy: 0.6415 - val_loss: 0.0
027 - val_accuracy: 0.6470
Epoch 3/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0027 - accuracy: 0.6595
Epoch 00003: val_loss improved from 0.00269 to 0.00263, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 24s 4ms/sample - loss: 0.0027 - accuracy: 0.6596 - val_loss: 0.0
026 - val_accuracy: 0.6563
Epoch 4/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0026 - accuracy: 0.6750
Epoch 00004: val_loss improved from 0.00263 to 0.00259, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 25s 4ms/sample - loss: 0.0026 - accuracy: 0.6745 - val_loss: 0.0
026 - val_accuracy: 0.6735
Epoch 5/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0026 - accuracy: 0.6972
Epoch 00005: val_loss improved from 0.00259 to 0.00257, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 23s 4ms/sample - loss: 0.0026 - accuracy: 0.6978 - val_loss: 0.0
026 - val_accuracy: 0.6981
Epoch 6/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0026 - accuracy: 0.7067
Epoch 00006: val_loss improved from 0.00257 to 0.00255, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 24s 4ms/sample - loss: 0.0026 - accuracy: 0.7069 - val_loss: 0.0
026 - val_accuracy: 0.7067
Epoch 7/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.7177
Epoch 00007: val_loss improved from 0.00255 to 0.00254, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 22s 4ms/sample - loss: 0.0025 - accuracy: 0.7180 - val_loss: 0.0
025 - val_accuracy: 0.7259
Epoch 8/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.7197
Epoch 00008: val_loss improved from 0.00254 to 0.00253, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 21s 4ms/sample - loss: 0.0025 - accuracy: 0.7195 - val_loss: 0.0
025 - val_accuracy: 0.7372
Epoch 9/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.7295
Epoch 00009: val_loss improved from 0.00253 to 0.00252, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
```

```
6025/6025 [==============================] - 24s 4ms/sample - loss: 0.0025 - accuracy: 0.7288 - val_loss: 0.0
025 - val_accuracy: 0.7478
Epoch 10/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.7415
Epoch 00010: val_loss improved from 0.00252 to 0.00252, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 23s 4ms/sample - loss: 0.0025 - accuracy: 0.7416 - val_loss: 0.0
025 - val_accuracy: 0.7611
Epoch 11/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.7572
Epoch 00011: val_loss improved from 0.00252 to 0.00251, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 22s 4ms/sample - loss: 0.0025 - accuracy: 0.7570 - val_loss: 0.0
025 - val_accuracy: 0.7585
Epoch 12/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.7610
Epoch 00012: val_loss improved from 0.00251 to 0.00251, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 24s 4ms/sample - loss: 0.0025 - accuracy: 0.7608 - val_loss: 0.0
025 - val_accuracy: 0.7777
Epoch 13/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.7688
Epoch 00013: val_loss improved from 0.00251 to 0.00251, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 22s 4ms/sample - loss: 0.0025 - accuracy: 0.7685 - val_loss: 0.0
025 - val_accuracy: 0.7903
Epoch 14/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.7813
Epoch 00014: val_loss improved from 0.00251 to 0.00251, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 22s 4ms/sample - loss: 0.0025 - accuracy: 0.7806 - val_loss: 0.0
025 - val_accuracy: 0.8069
Epoch 15/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.7965
Epoch 00015: val_loss improved from 0.00251 to 0.00250, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 22s 4ms/sample - loss: 0.0025 - accuracy: 0.7967 - val_loss: 0.0
025 - val_accuracy: 0.8129
Epoch 16/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.8013
Epoch 00016: val_loss improved from 0.00250 to 0.00250, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 24s 4ms/sample - loss: 0.0025 - accuracy: 0.8013 - val_loss: 0.0
```

```
025 - val_accuracy: 0.8228
Epoch 17/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.8145
Epoch 00017: val_loss improved from 0.00250 to 0.00250, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 26s 4ms/sample - loss: 0.0025 - accuracy: 0.8149 - val_loss: 0.0
025 - val_accuracy: 0.8109
Epoch 18/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.8348
Epoch 00018: val_loss improved from 0.00250 to 0.00250, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 24s 4ms/sample - loss: 0.0025 - accuracy: 0.8350 - val_loss: 0.0
025 - val_accuracy: 0.8646
Epoch 19/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.8357
Epoch 00019: val_loss improved from 0.00250 to 0.00250, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 22s 4ms/sample - loss: 0.0025 - accuracy: 0.8357 - val_loss: 0.0
025 - val_accuracy: 0.8368
Epoch 20/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.8493
Epoch 00020: val_loss improved from 0.00250 to 0.00250, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 25s 4ms/sample - loss: 0.0025 - accuracy: 0.8490 - val_loss: 0.0
025 - val_accuracy: 0.8825
Epoch 21/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.8678
Epoch 00021: val_loss improved from 0.00250 to 0.00250, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 23s 4ms/sample - loss: 0.0025 - accuracy: 0.8684 - val_loss: 0.0
025 - val_accuracy: 0.8832
Epoch 22/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.8697
Epoch 00022: val_loss improved from 0.00250 to 0.00250, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 24s 4ms/sample - loss: 0.0025 - accuracy: 0.8700 - val_loss: 0.0
025 - val_accuracy: 0.9011
Epoch 23/100
6016/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.8815
Epoch 00023: val_loss improved from 0.00250 to 0.00250, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 22s 4ms/sample - loss: 0.0025 - accuracy: 0.8813 - val_loss: 0.0
025 - val_accuracy: 0.8945
```

```
Epoch 24/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.8938
Epoch 00024: val_loss improved from 0.00250 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 21s 4ms/sample - loss: 0.0025 - accuracy: 0.8939 - val_loss: 0.0
025 - val_accuracy: 0.9111
Epoch 25/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9027
Epoch 00025: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9026 - val_loss: 0.0
025 - val_accuracy: 0.9204
Epoch 26/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9112
Epoch 00026: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9109 - val_loss: 0.0
025 - val_accuracy: 0.9164
Epoch 27/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9165
Epoch 00027: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 18s 3ms/sample - loss: 0.0025 - accuracy: 0.9167 - val_loss: 0.0
025 - val_accuracy: 0.9250
Epoch 28/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9248
Epoch 00028: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9246 - val_loss: 0.0
025 - val_accuracy: 0.9131
Epoch 29/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9358
Epoch 00029: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 18s 3ms/sample - loss: 0.0025 - accuracy: 0.9358 - val_loss: 0.0
025 - val_accuracy: 0.9463
Epoch 30/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9393
Epoch 00030: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9391 - val_loss: 0.0
025 - val_accuracy: 0.9536
Epoch 31/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9473
```

```
Epoch 00031: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9476 - val_loss: 0.0
025 - val_accuracy: 0.9575
Epoch 32/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9570
Epoch 00032: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9570 - val_loss: 0.0
025 - val_accuracy: 0.9529
Epoch 33/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9563
Epoch 00033: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9563 - val_loss: 0.0
025 - val_accuracy: 0.9635
Epoch 34/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9667
Epoch 00034: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9668 - val_loss: 0.0
025 - val_accuracy: 0.9688
Epoch 35/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9703
Epoch 00035: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9705 - val_loss: 0.0
025 - val_accuracy: 0.9701
Epoch 36/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9725
Epoch 00036: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9724 - val_loss: 0.0
025 - val_accuracy: 0.9741
Epoch 37/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9770
Epoch 00037: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9769 - val_loss: 0.0
025 - val_accuracy: 0.9754
Epoch 38/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9803
Epoch 00038: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
```

```
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9804 - val_loss: 0.0
025 - val_accuracy: 0.9768
Epoch 39/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9818
Epoch 00039: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9819 - val_loss: 0.0
025 - val_accuracy: 0.9794
Epoch 40/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9843
Epoch 00040: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9844 - val_loss: 0.0
025 - val_accuracy: 0.9794
Epoch 41/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9882
Epoch 00041: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9882 - val_loss: 0.0
025 - val_accuracy: 0.9821
Epoch 42/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9887
Epoch 00042: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9887 - val_loss: 0.0
025 - val_accuracy: 0.9814
Epoch 43/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9907
Epoch 00043: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9907 - val_loss: 0.0
025 - val_accuracy: 0.9854
Epoch 44/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9915
Epoch 00044: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9915 - val_loss: 0.0
025 - val_accuracy: 0.9900
Epoch 45/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9942
Epoch 00045: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9942 - val_loss: 0.0
025 - val_accuracy: 0.9861
Epoch 46/100
```

```
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9928
Epoch 00046: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9929 - val_loss: 0.0
025 - val_accuracy: 0.9881
Epoch 47/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9935
Epoch 00047: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9935 - val_loss: 0.0
025 - val_accuracy: 0.9881
Epoch 48/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9943
Epoch 00048: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9944 - val_loss: 0.0
025 - val_accuracy: 0.9861
Epoch 49/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9957
Epoch 00049: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9957 - val_loss: 0.0
025 - val_accuracy: 0.9907
Epoch 50/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9950
Epoch 00050: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9950 - val_loss: 0.0
025 - val_accuracy: 0.9914
Epoch 51/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9953
Epoch 00051: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 18s 3ms/sample - loss: 0.0025 - accuracy: 0.9954 - val_loss: 0.0
025 - val_accuracy: 0.9900
Epoch 52/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9972
Epoch 00052: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9972 - val_loss: 0.0
025 - val_accuracy: 0.9927
Epoch 53/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9968
Epoch 00053: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9968 - val_loss: 0.0
025 - val_accuracy: 0.9934
Epoch 54/100
```

```
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9965
Epoch 00054: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9963 - val_loss: 0.0
025 - val_accuracy: 0.9920
Epoch 55/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9975
Epoch 00055: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9975 - val_loss: 0.0
025 - val_accuracy: 0.9927
Epoch 56/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9963
Epoch 00056: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9962 - val_loss: 0.0
025 - val_accuracy: 0.9914
Epoch 57/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9963
Epoch 00057: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9963 - val_loss: 0.0
025 - val_accuracy: 0.9940
Epoch 58/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9983
Epoch 00058: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9983 - val_loss: 0.0
025 - val_accuracy: 0.9927
Epoch 59/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9977
Epoch 00059: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9977 - val_loss: 0.0
025 - val_accuracy: 0.9940
Epoch 60/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9972
Epoch 00060: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9972 - val_loss: 0.0
025 - val_accuracy: 0.9940
Epoch 61/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9983
Epoch 00061: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9983 - val_loss: 0.0
025 - val_accuracy: 0.9947
Epoch 62/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9968
```

```
Epoch 00062: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9968 - val_loss: 0.0
025 - val_accuracy: 0.9907
Epoch 63/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9975
Epoch 00063: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9975 - val_loss: 0.0
025 - val_accuracy: 0.9940
Epoch 64/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9987
Epoch 00064: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9987 - val_loss: 0.0
025 - val_accuracy: 0.9920
Epoch 65/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9983
Epoch 00065: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9983 - val_loss: 0.0
025 - val_accuracy: 0.9947
Epoch 66/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9982
Epoch 00066: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9982 - val_loss: 0.0
025 - val_accuracy: 0.9927
Epoch 67/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9988
Epoch 00067: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9988 - val_loss: 0.0
025 - val_accuracy: 0.9940
Epoch 68/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9997
Epoch 00068: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9997 - val_loss: 0.0
025 - val_accuracy: 0.9914
Epoch 69/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9990
Epoch 00069: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 18s 3ms/sample - loss: 0.0025 - accuracy: 0.9990 - val_loss: 0.0
025 - val_accuracy: 0.9954
Epoch 70/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9987
Epoch 00070: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
```

```
et/after-train.h5
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9987 - val_loss: 0.0
025 - val_accuracy: 0.9967
Epoch 71/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9998
Epoch 00071: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9998 - val_loss: 0.0
025 - val_accuracy: 0.9954
Epoch 72/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9980
Epoch 00072: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 18s 3ms/sample - loss: 0.0025 - accuracy: 0.9980 - val_loss: 0.0
025 - val_accuracy: 0.9947
Epoch 73/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9988
Epoch 00073: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9988 - val_loss: 0.0
025 - val_accuracy: 0.9940
Epoch 74/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9993
Epoch 00074: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9993 - val_loss: 0.0
025 - val_accuracy: 0.9920
Epoch 75/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9980
Epoch 00075: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9980 - val_loss: 0.0
025 - val_accuracy: 0.9934
Epoch 76/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9988
Epoch 00076: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9988 - val_loss: 0.0
025 - val_accuracy: 0.9947
Epoch 77/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9993
Epoch 00077: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9993 - val_loss: 0.0
025 - val_accuracy: 0.9914
Epoch 78/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9993
```

```
Epoch 00078: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9993 - val_loss: 0.0
025 - val_accuracy: 0.9960
Epoch 79/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9990
Epoch 00079: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9990 - val_loss: 0.0
025 - val_accuracy: 0.9934
Epoch 80/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9990
Epoch 00080: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9990 - val_loss: 0.0
025 - val_accuracy: 0.9954
Epoch 81/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9988
Epoch 00081: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9988 - val_loss: 0.0
025 - val_accuracy: 0.9954
Epoch 82/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9992
Epoch 00082: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9992 - val_loss: 0.0
025 - val_accuracy: 0.9947
Epoch 83/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9987
Epoch 00083: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9985 - val_loss: 0.0
025 - val_accuracy: 0.9927
Epoch 84/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9990
Epoch 00084: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9990 - val_loss: 0.0
025 - val_accuracy: 0.9940
Epoch 85/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9982
Epoch 00085: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9982 - val_loss: 0.0
025 - val_accuracy: 0.9934
Epoch 86/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9992
Epoch 00086: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
```

```
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9992 - val_loss: 0.0
025 - val_accuracy: 0.9947
Epoch 87/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9997
Epoch 00087: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9997 - val_loss: 0.0
025 - val_accuracy: 0.9954
Epoch 88/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9993
Epoch 00088: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9993 - val_loss: 0.0
025 - val_accuracy: 0.9954
Epoch 89/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9995
Epoch 00089: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9995 - val_loss: 0.0
025 - val_accuracy: 0.9934
Epoch 90/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9992
Epoch 00090: val_loss improved from 0.00249 to 0.00249, saving model to My Drive/Colab Notebooks/215-FP_Datas
et/after-train.h5
6025/6025 [==============================] - 18s 3ms/sample - loss: 0.0025 - accuracy: 0.9992 - val_loss: 0.0
025 - val_accuracy: 0.9940
Epoch 91/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9992
Epoch 00091: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9992 - val_loss: 0.0
025 - val_accuracy: 0.9947
Epoch 92/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9988
Epoch 00092: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9988 - val_loss: 0.0
025 - val_accuracy: 0.9960
Epoch 93/100
6000/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9992
Epoch 00093: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9992 - val_loss: 0.0
025 - val_accuracy: 0.9954
Epoch 94/100
6016/6025 [============================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9998
Epoch 00094: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9998 - val_loss: 0.0
025 - val_accuracy: 0.9960
```

```
Epoch 95/100
6000/6025 [===========================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9995
Epoch 00095: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9995 - val_loss: 0.0
025 - val_accuracy: 0.9954
Epoch 96/100
6000/6025 [===========================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9998
Epoch 00096: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9998 - val_loss: 0.0
025 - val_accuracy: 0.9954
Epoch 97/100
6000/6025 [===========================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9992
Epoch 00097: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9990 - val_loss: 0.0
025 - val_accuracy: 0.9954
Epoch 98/100
6000/6025 [===========================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9988
Epoch 00098: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9988 - val_loss: 0.0
025 - val_accuracy: 0.9960
Epoch 99/100
6000/6025 [===========================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9990
Epoch 00099: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9990 - val_loss: 0.0
025 - val_accuracy: 0.9940
Epoch 100/100
6000/6025 [===========================>.] - ETA: 0s - loss: 0.0025 - accuracy: 0.9993
Epoch 00100: val_loss did not improve from 0.00249
6025/6025 [==============================] - 17s 3ms/sample - loss: 0.0025 - accuracy: 0.9993 - val_loss: 0.0
025 - val_accuracy: 0.9960
```

In [ ]:

In [ ]:

In [ ]:
```
batch_size = 16
num_classes = 2
epochs = 100



x_test=  x_valid
y_test=  y_valid









#model: model without pre-training

input_image = Input(shape=(None, None, 3))
x = Conv2D(64, (3, 3), activation='relu')(input_image)
x=BatchNormalization(axis=-1, momentum=0.99, epsilon=0.001, center=True, scale=True, beta_initializer='zeros'
, gamma_initializer='ones', moving_mean_initializer='zeros', moving_variance_initializer='ones', beta_regular
izer=None, gamma_regularizer=None, beta_constraint=None, gamma_constraint=None)(x)
x = Conv2D(64, (3, 3), activation='relu')(x)
x = AveragePooling2D((2, 2))(x)
x = Conv2D(128, (3, 3), activation='relu')(x)
x = Conv2D(128, (3, 3), activation='relu')(x)




x = Reshape((-1, 128))(x)
x = Capsule(32, 8, 3, True)(x)
x = Capsule(32, 8, 3, True)(x)
capsule = Capsule(2, 16, 3, True)(x)
output = Lambda(lambda z: K.sqrt(K.sum(K.square(z), 2)))(capsule)


#model2: model with pre-training
input_image2 = Input(shape=(None, None, 3))
x2 = Conv2D(64, (3, 3), activation='relu')(input_image2)
x2=BatchNormalization(axis=-1, momentum=0.99, epsilon=0.001, center=True, scale=True, beta_initializer='zero
s', gamma_initializer='ones', moving_mean_initializer='zeros', moving_variance_initializer='ones', beta_regul
```

```
arizer=None, gamma_regularizer=None, beta_constraint=None, gamma_constraint=None)(x2)
x2 = Conv2D(64, (3, 3), activation='relu')(x2)
x2 = AveragePooling2D((2, 2))(x2)
x2 = Conv2D(128, (3, 3), activation='relu')(x2)
x2 = Conv2D(128, (3, 3), activation='relu')(x2)




x2 = Reshape((-1, 128))(x2)
x2 = Capsule(32, 8, 3, True)(x2)
x2 = Capsule(32, 8, 3, True)(x2)
capsule2 = Capsule(2, 16, 3, True)(x2)
output2 = Lambda(lambda z: K.sqrt(K.sum(K.square(z), 2)))(capsule2)




model = Model(inputs=[input_image], outputs=[output])
model2 = Model(inputs=[input_image2], outputs=[output2])



model.load_weights('My Drive/Colab Notebooks/215-FP_Dataset/after-train.h5')
model2.load_weights('My Drive/Colab Notebooks/215-FP_Dataset/weights-improvement-binary-after-44.h5')

predict=model.predict([x_test])
predict=np.argmax(predict,axis=1)

predict2=model2.predict([x_test])
predict2=np.argmax(predict2,axis=1)


summation1=0
```

In [47]:
```
y_test=np.argmax(y_test,axis=1)
y_test[0:100]
```

Out[47]:
```
array([0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0,
       0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0,
       1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1,
       0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1])
```

In [48]:
```
predict[0:100]
```

Out[48]:
```
array([0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0,
       0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1,
       0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1])
```

In [ ]:
```
'''from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import OneHotEncoder
predict=np.array(predict)
label_encoder = LabelEncoder()
predict = label_encoder.fit_transform(predict)
predict



onehot_encoder = OneHotEncoder(sparse=False)
predict = predict.reshape(len(predict), 1)
predict = onehot_encoder.fit_transform(predict)
predict

predict2=np.array(predict2)
label_encoder = LabelEncoder()
predict2 = label_encoder.fit_transform(predict2)
predict2
onehot_encoder = OneHotEncoder(sparse=False)
predict2 = predict2.reshape(len(predict2), 1)
predict2 = onehot_encoder.fit_transform(predict2)
predict2'''
```

In [ ]:
```

```

In [63]:
```
accuracy_before=0
specificity_before=0
sensitivity_before=0
precision_before=0
f1_before=0
newaccuracy=0
from __future__ import print_function
from tensorflow.keras import backend as K
from tensorflow.keras.layers import Layer
from tensorflow.keras import activations
from tensorflow.keras import utils
from tensorflow.keras.models import Model
from tensorflow.keras.layers import *
import numpy as np
import keras
from sklearn.metrics import roc_curve, auc
from matplotlib import pyplot as plt

summation1=0

for i in range(len(x_test)):
    if predict[i]==y_test[i]:
        summation1=summation1+1
print('accurate predictions: ',summation1)
print('Total predictions to be made in testing :', len(x_test))
accuracy_before=summation1/len(x_test)



summation1=0
summation2=0
summation3=0
summation4=0

for i in range(len(x_test)):
    if predict[i]==y_test[i] and y_test[i]==0:
        summation1=summation1+1

specificity_before=summation1/np.count_nonzero(y_test==0)

for i in range(len(x_test)):
    if predict[i]==y_test[i] and y_test[i]==1:
```

```
        summation2=summation2+1

sensitivity_before=summation2/np.count_nonzero(y_test==1)

for i in range(len(x_test)):
    if predict[i]!=y_test[i] and y_test[i]==1:
        summation3=summation3+1

for i in range(len(x_test)):
    if predict2[i]!=y_test[i] and y_test[i]==0:
        summation4=summation4+1


precision_before=summation1/(summation1+summation3)
newaccuracy=((summation1+summation2)/(summation1+summation2+summation3+summation4))
f1_before=2*((sensitivity_before*precision_before)/(sensitivity_before + precision_before))
#print("TN: ",summation1,", TP : ",summation2,", FP:",summation3, ", FN :",summation4)
print('accuracy : ',accuracy_before)
print('specificity  : ',specificity_before)
print('sensitivity or precision  : ',sensitivity_before)
print('precision  : ', precision_before)
print('f1-score  : ', f1_before)
#print('newaccuracy : ',newaccuracy)
```

```
accurate predictions:  1498
Total predictions to be made in testing : 1507
accuracy :  0.9940278699402787
specificity  :  0.9960745829244357
sensitivity or precision  :  0.9897540983606558
precision  :  0.9950980392156863
f1-score  :  0.9924188748760097
```

In [ ]:

In [ ]: