**DAY 04**

**22.03.2023**

**Dataset provided:**

An examination dataset was provided in google drive.

It contained three folders namely train, validation and validation2.

Each folder further had 5 sub folders(multi-class) which were normal, cheat, phone, paperseeing and paperexchange.

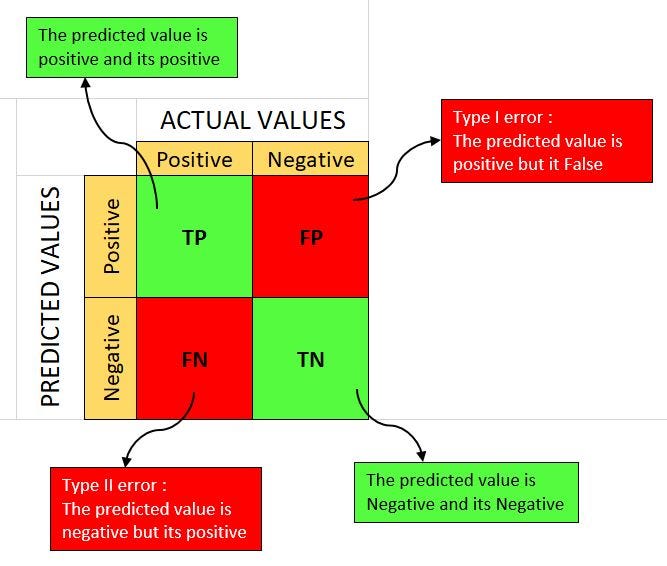
**Google Drive:**

https://drive.google.com/drive/folders/1v0nUtgz8wMnn7JJWqw4MHPaVJZZ1GqOa?usp=drive\_link

**Explanation to Various topics**:

**Confusion Matrix**

The confusion matrix is a matrix used to determine the performance of the classification models for a given set of test data. It can only be determined if the true values for test data are known. The matrix itself can be easily understood, but the related terminologies may be confusing. Since it shows the errors in the model performance in the form of a matrix, hence also known as an **error matrix**.



**Batch size**

The batch size is a hyperparameter that defines the number of samples to work through before updating the internal model parameters.

**Epoch**

An epoch is a complete iteration through the entire training dataset in one cycle for training the machine learning model. During an epoch, Every training sample in the dataset is processed by the model, and its weights and biases are updated in accordance with the computed loss or error.

**Optimizer**

**Optimizers** are algorithms or methods used to minimize an error function(loss function)or to maximize the efficiency of production. Optimizers are mathematical functions which are dependent on model’s learnable parameters i.e Weights & Biases. Optimizers help to know how to change weights and learning rate of neural network to reduce the losses.

**Jupyter Notebook:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import plotly.express as px

import os

import tensorflow as tf

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from sklearn.preprocessing import LabelBinarizer

from sklearn.metrics import roc\_curve, auc, roc\_auc\_score

# from IPython.display import clear\_output

import warnings

warnings.filterwarnings('ignore')

train\_dir = "/content/drive/MyDrive/EXAMDATASET/train"

test\_dir = "/content/drive/MyDrive/EXAMDATASET/validation"

SEED = 12

IMG\_HEIGHT = 64

IMG\_WIDTH = 64

BATCH\_SIZE = 64      #Was 46

EPOCHS = 1

LR =  0.001

NUM\_CLASSES = 14

CLASS\_LABELS = ['cheat','normal','peperexchange','paperseeing','phone']

train\_dir = "/content/drive/MyDrive/EXAMDATASET/train"

test\_dir = "/content/drive/MyDrive/EXAMDATASET/validation"

SEED = 12

IMG\_HEIGHT = 64

IMG\_WIDTH = 64

BATCH\_SIZE = 64      #Was 46

EPOCHS = 1

LR =  0.001

NUM\_CLASSES = 14

CLASS\_LABELS = ['cheat','normal','peperexchange','paperseeing','phone']

from tensorflow.keras.callbacks import ModelCheckpoint

filepath="weights-improvementResnet50-{epoch:02d}-{val\_accuracy:.2f}.hdf5"

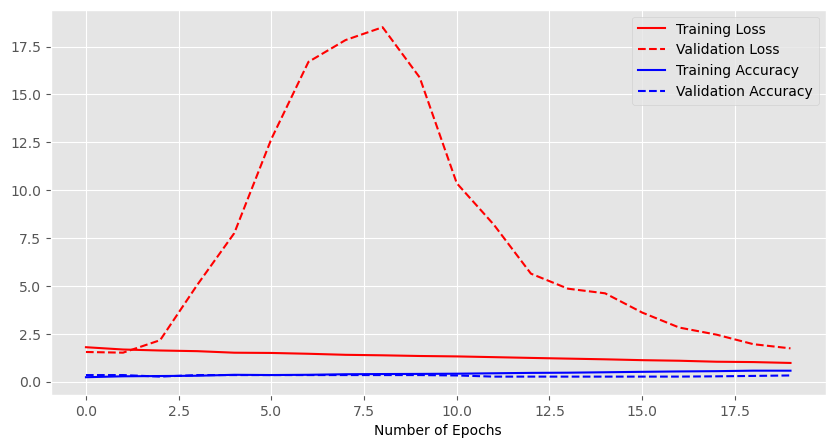
checkpoint = ModelCheckpoint(filepath, monitor='val\_accuracy', verbose=1, save\_best\_only=True, mode='max')

callbacks\_list = [checkpoint]

history = model.fit(x = train\_generator,validation\_data=validation\_generator,epochs = 20,callbacks=callbacks\_list)

#history = model.fit(x = train\_generator,validation\_split=0.2,epochs = EPOCHS)

**Graphs:**

****

****