**CODINGS**

**INDEX.SAMPLE**

import subprocess

import cv2

import numpy as np

import os

import shutil

import argparse

import copy

import json

ap = argparse.ArgumentParser()

ap.add\_argument("-d", "--attach", required=True,help="attach video or image")

args = vars(ap.parse\_args())

nam = args["attach"]

total1 = []

val = nam.split(".")[::-1]

if val[0] == "mp4":

vidcap = cv2.VideoCapture(args["attach"])

vidcap.set(cv2.CAP\_PROP\_POS\_MSEC,50000)

if not os.path.exists('test'):

os.makedirs('test')

success, image = vidcap.read()

count = 0

success = True

while success:

success, image = vidcap.read()

name = './test/frame' + str(count) + '.jpg'

cv2.imwrite(name, image)

cmd = 'python test\_network.py --model dataset/atm.model --image '+name+' --alarm alarm.wav'

p = subprocess.Popen(cmd, stdout=subprocess.PIPE, shell=True)

out, err = p.communicate()

out = out.split('\n')

else:

dst\_dir = "test/"

if not os.path.exists('test'):

os.makedirs('test')

shutil.copy(nam, dst\_dir)

for lin in result:

if not lin.startswith('#'):

print(lin)

total1.append(lin)

**TRAINING.SAMPLE**

import matplotlib

matplotlib.use("Agg")

from keras.preprocessing.image import ImageDataGenerator

from keras.optimizers import Adam

from sklearn.model\_selection import train\_test\_split

from keras.preprocessing.image import img\_to\_array

from keras.utils import to\_categorical

from pyimagesearch.lenet import LeNet

from imutils import paths

import matplotlib.pyplot as plt

import numpy as np

import argparse

import random

import cv2

import os

EPOCHS = 25

INIT\_LR = 1e-3

BS = 32

print("[INFO] loading images...")

data = []

labels = []

imagePaths = sorted(list(paths.list\_images(args["dataset"])))

random.seed(42)

random.shuffle(imagePaths)

data = np.array(data, dtype="float") / 255.0

labels = np.array(labels)

(trainX, testX, trainY, testY) = train\_test\_split(data,

labels, test\_size=0.25, random\_state=42)

trainY = to\_categorical(trainY, num\_classes=2)

testY = to\_categorical(testY, num\_classes=2)

aug = ImageDataGenerator(rotation\_range=30, width\_shift\_range=0.1,

height\_shift\_range=0.1, shear\_range=0.2, zoom\_range=0.2,

horizontal\_flip=True, fill\_mode="nearest")

print("[INFO] compiling model...")

model = LeNet.build(width=28, height=28, depth=3, classes=2)

opt = Adam(lr=INIT\_LR, decay=INIT\_LR / EPOCHS)

model.compile(loss="binary\_crossentropy", optimizer=opt,

metrics=["accuracy"])

print("[INFO] training network...")

H = model.fit\_generator(aug.flow(trainX, trainY, batch\_size=BS),

validation\_data=(testX, testY), steps\_per\_epoch=len(trainX) // BS,

epochs=EPOCHS, verbose=1)

**TESTING.SAMPLE**

import argparse

import cv2

import playsound

from pygame import mixer

import time

import imutils

import numpy as np

from keras.models import load\_model

from keras.preprocessing.image import img\_to\_array

def sound\_alarm(path):

playsound.playsound(path)

k = 0

args = vars(ap.parse\_args())

image = cv2.imread(args["image"])

orig = image.copy()

image = cv2.resize(image, (28, 28))

if k==0:

print("Procession start")

else:

print("stopped due to raw pack")

image = np.expand\_dims(image, axis=0)

model = load\_model(args["model"])

(dont, do) = model.predict(image)[0]

label = "{}: {:.2f}%".format(label, proba \* 100)

output = imutils.resize(orig, width=400)

cv2.putText(output, label, (10, 25), cv2.FONT\_HERSHEY\_SIMPLEX,

0.7, (0, 255, 0), 2)

total.append(label)

total.append(args['image'])

print(total)

cv2.imshow("Output", output)

if total[0] == "dont":

playsound.playsound('alarm.wav', True)

cv2.waitKey(0)