ATM SECURITY

Team Name:chasers

CT/DT Reference No:

CT20172255364

CT20172255286

**YouTube Video Link :**

Screen Recorded Video with Project explanation

**Github Repository Link :**

Working Exe / Apk / similar file

**1. Abstract:**

In olden days, people used to carry money wherever they are going. But nowadays, people are moving towards ATM for money transactions. At the meantime, the theft in the ATM has also been increased. So, the people are feeling that the ATM is less secured. This problem could be overcome by providing some solutions like, the security camera fixed inside the ATM found if any person inside the ATM is undergoing any illegal activity, then the alarm sound is produced. This solution is based on the concept of Machine Learning using Python. The concept works under the principle of Supervised Learning algorithm. The software used here is Visual Studio Code and the Operating System used is UBUNTU. The input is taken from the surveillance camera inside the ATM and it is compared with the dataset. The dataset are classified into two types as legal dataset and illegal dataset. The dataset first undergo training phase and then testing phase. If the input matches with the illegal dataset, then the alarm is triggered and message is sent to authorized person. If the input matches with the legal dataset then alarm remains quiet.

**2. Introduction:**

Nowadays, in cities or rural areas whenever the ATM thefts are happened there are lots of troubles being faced.People are not much initiative in informing the police and ambulance.In order to provide solution for this serious problem, we have designed a model using machine learning concepts.

In this model, the ATM are continuously monitored through CCTV cameras. The machine is trained with the dataset. The each and every clip of the CCTV camera is compared with the trained datasets. If the clips are of with normal activities of the ATM, then there will be no issues. In case of any clip matches with the trained dataset, immediately the message is passed to the ATM services centre corresponding to the area and alarmed outside the ATM.

**3. Technology Stack:**

**Hardware and Software Requirements:**

1. CCTV Camera
2. Intel core i3 64-bit processor
3. 4 GB ram
4. 1 TB 5400rpm hard drive
5. Visual studio code

**Algorithms involved:**

CNN(Convolutional Neural Networks)

**Domain involved:**

Banking(Machine Learning)

**4. Working Methodology:**

**ARCHITECTURAL DIAGRAM**

CCTV CAMERA

TESTING DATASET

OUTPUT

MODEL CREATED

MACHINE

TRAINING DATA

**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')

5

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":

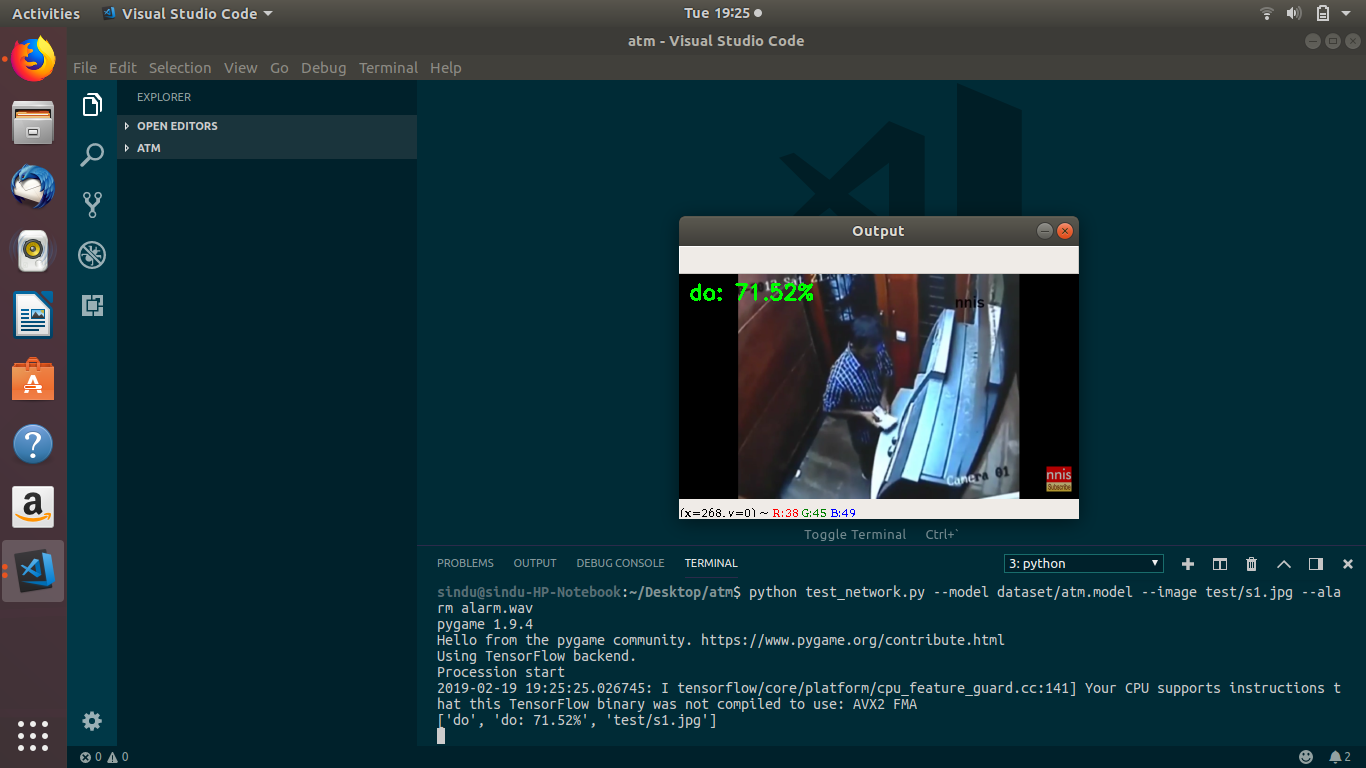
playsound.playsound('alarm.wav', True)

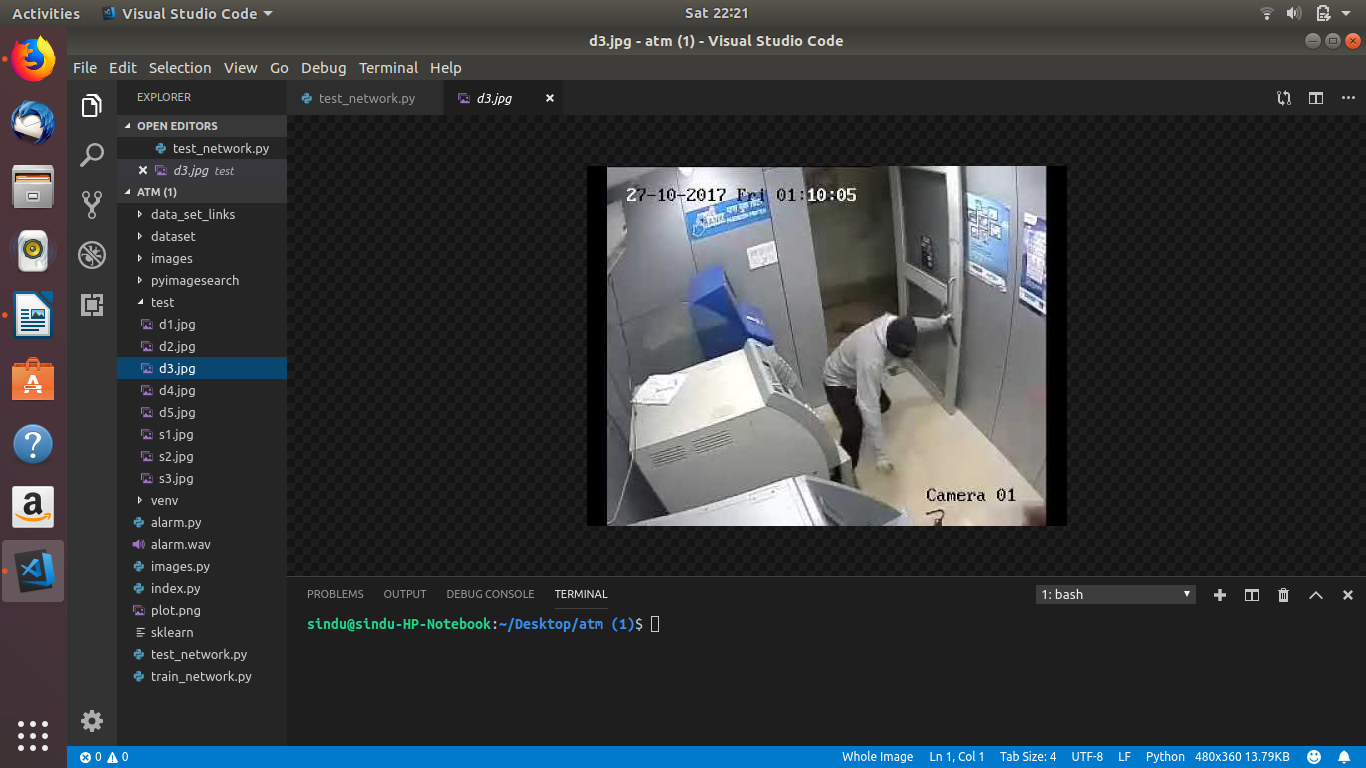
cv2.waitKey(0)

**WORKFLOW**

* Initially we need to collect the dataset of ATM images
* Then we need to rename the dataset in some order(1.jpg,2.jpg,…) thus we can load the dataset is easily
* Now create a python program for loading the image dataset
* If the dataset is not in correct format then the dataset will be automatically deleted
* The dataset must be in correct pixel and correct resolution
* Then the dataset is trained using the cnn algorithm
* The accuracy level can be seen by ploting the graph
* If the number of dataset increases then the accuracy also said to be increased
* After training the dataset the model is created
* Now we need to create the test dataset
* Then we need to check the text dataset with the created model
* If expected output is obtained then our project is correct
* Else we need to train the model again and again until we get the correct output
* Here if the illegal activity image is detected then we need to send message to nearby ambulance and police station and alarm outside the atm
* Thus we can save the time and we can save the life of people

**4. SNAPSHOTS:**

****

****

**5.CONCLUSION:**

Thus, We can reduce the illegal activities in ATMs by detecting the activities automatically with the help of machine learning algorithms and sending the information to their concerned authorities by using our security system.