Python Script – ocr.py

#!usr/bin/python

#

# IOT Based Traffic Management System Project :

# By Srujan Deshpande, 19183291

# M.E. (Information and Network Security FT)

# Academic Year : 2019-2020

#

#

# This file contains python script to perform digial signal processing

# on vehicle number plate image to extract registration number from

# registration number plate image. In the process it alsow identifies

# emergency vehicle (police, ambulance and fire brigade).

#

import cv2

import imutils

import numpy as np

import pytesseract

import sys

import time

from PIL import Image

from collections import namedtuple

from datetime import datetime

from random import seed

from random import randint

seed(1)

57 | P a g emyVehicleList = [ {"Reg Number":"KA51 EB 6905", "Color":"GOLDEN", "Type":"CAR",

"Brand":"CHEVROLET OPTRA", "Fuel":"PETROL", "Capacity":"1.8L", "Dues":0},

{"Reg Number":"KA04 MK 1809", "Color":"WHITE", "Type":"SUV",

"Brand":"MAHINDRA XUV 5OO", "Fuel":"DIESEL", "Capacity":"2.5L", "Dues":0},

{"Reg Number":"KA05 ML 1513", "Color":"GREY", "Type":"2-W", "Brand":"HONDA

ACTIVA", "Fuel":"PETROL", "Capacity":"1.5L", "Dues":500},

{"Reg Number":"KAO5 ML 1513", "Color":"GREY", "Type":"2-W", "Brand":"HONDA

ACTIVA", "Fuel":"PETROL", "Capacity":"1.5L", "Dues":500},

{"Reg Number":"EMERGENCY VEHICLE :: POLICE", "Color":"BLUE", "Type":"JEEP",

"Brand":"SWARAZ MAZDA", "Fuel":"Diesel", "Capacity":"2.5L", "Dues":0},

{"Reg Number":"EMERGENCY VEHICLE :: AMBULANCE", "Color":"WHITE",

"Type":"VAN", "Brand":"SWARAZ MAZDA", "Fuel":"Diesel", "Capacity":"2.5L",

"Dues":0},

{"Reg Number":"EMERGENCY VEHICLE :: FIRE BRIGADE", "Color":"RED",

"Type":"TRUCK", "Brand":"TATA", "Fuel":"Diesel", "Capacity":"2.5L", "Dues":0},

] # Sort of database sample look up table for few vehicle types

vehIndex = 7

vehicleType = "NORMAL VEHICLE"

sigStatusFile = "./log/sigop.txt" # File that contains traffic signal status

vehTypeFile = "./log/veh.txt" # This file is used to detect presence of emergency vehicle.

Detection is done by this python script "ocr.py" that performs DSP on vehicle image to extract

vehicle registration number

LEAVY\_NEW\_PENALTY = 1000 # Penalty to levy in local currency

scriptName = ""

Month = ["JAN", "FEB", "MAR", "APR", "MAY", "JUN", "JUL", "AUG", "SEP", "OCT",

"NOV", "DEC"]

58 | P a g e59 | P a g e

dateStr = ""

def getDateTime():

#

# This funcion retruns a string composed of script name,

# current date and time. This general purpose function

# is useful to keep track of simulaeted events of traffic

# on the road which faces real time signal light

# implementation using raspberry pi #4 which is handled by

# another script <trlight.py>

#

scrName = str(sys.argv[0])

dateTimeObj = datetime.now()

year = str(dateTimeObj.year)

month = dateTimeObj.month

day = str(dateTimeObj.day)

hour = str(dateTimeObj.hour)

minute = str(dateTimeObj.minute)

second = dateTimeObj.second

usecond = dateTimeObj.microsecond

float\_sec = round((second\*1000000 + usecond)/1000000,3)

y = str(float\_sec)

nfsecStr = y[3:len(y)]

scrName = scrName.rjust(10)

dateStr = '[' + scrName + '/' + day + '-' + Month[month-1] + '-' + year + ' :: ' + hour + ':'

+ minute + ':' + str(second).zfill(2) + '.' + nfsecStr.zfill(2) + '' + '] ->'

return (dateStr)

def vehicleLookup(regNumber):

# 60 | P a g e

# This function determines whether arriving vehicle at the signal

# is an emergency vehicle (Police, Ambulance or Fire Brigade) or not.

# The detection is done based on number plate in front of vehicle.

#

global vehIndex

found = 0

vehicleType = "EMERGENCY VEHICLE"

if(regNumber == "POLICE"):

vehicleType = "EMERGENCY VEHICLE"

print("%s Emergency Vehicle :: %s is %s" %

(getDateTime(),regNumber,"Detected"))

return vehicleType

if(regNumber == "AMBULANCE"):

vehicleType = "EMERGENCY VEHICLE"

print("%s Emergency Vehicle :: %s is %s" % (getDateTime(),

regNumber,"Detected"))

return vehicleType

if(regNumber == "FIRE BRIGADE"):

vehicleType = "EMERGENCY VEHICLE"

return vehicleType

else:

vehicleType = "NORMAL VEHICLE"

for n in range(0,vehIndex):

if(regNumber == myVehicleList[n]["Reg Number"]):

found = found + 1

if(found == 0):

print("%s Vehicle Not Found" % (getDateTime()))

return vehicleType61 | P a g e

def levyPenalty(regNumber,newPenalty):

#

# This function levy's penalty against a registerred vehicle.

# If vehicle has alread due penalty, it adds new penatly to it.

#

global scriptName

for n in range(0,vehIndex):

if(regNumber == myVehicleList[n]["Reg Number"]):

myVehicleList[n]["Dues"] = myVehicleList[n]["Dues"] + newPenalty

print("%s New Fault : Signal Jump" % (getDateTime()))

print("%s New Penalty levied for Vehicle Regn No. %s is %d" %

(getDateTime(), regNumber,newPenalty))

print("%s Total Penalty Due for Vehicle Regn No. %s is %d" %

(getDateTime(), myVehicleList[n]["Reg Number"], myVehicleList[n]["Dues"]))

def processImage(myFile):

#

# This function performns Image Processing to extract

# text from number plate in front of vehicle

#

global scriptName

# myImage =Image.open (myFile)

#display image

myImage = np.array(cv2.imread(myFile))

#cv2.imshow("Image", myImage)

#print("Press 'c/C' to continue")

#while(1):

#

if cv2.waitKey(25) & 0xFF == ord('c'):

#

break;

#cv2.destroyAllWindows()

text = pytesseract.image\_to\_string(myImage, lang='eng', config='-l eng --oem 3 --psm

6 outputbase digits

-c

tessedit\_char\_whitelist=ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789')

return(text)

def main():

#

# Main program

#

global scriptName

totalArg = len(sys.argv)

scriptName = str(sys.argv[0])

for n in range(1,totalArg):

myFile = str(sys.argv[n])

myVeh = processImage(myFile)

vehicleType = vehicleLookup(myVeh)

myFile = open(sigStatusFile,"r")

sigStatus = str(myFile.read())

print("%s Current Signal is %s, while %s is crossing" % (getDateTime(),

sigStatus, vehicleType))

myFile = open(vehTypeFile,"w")

myFile.write(vehicleType)

myFile.close()63 | P a g e

if(vehicleType == "NORMAL VEHICLE"):

if(sigStatus == "RED"):

levyPenalty(myVeh,LEAVY\_NEW\_PENALTY)

#myFile.close()

scaledValue = randint(1,10)

time.sleep(scaledValue)

main() # call main script to run