TECHNOCOLABS COMPUTER-VISION INTERNSHIP

PROJECT REPORT

Linkedin: https://www.linkedin.com/company/technocolabs/

TEAM:
1.MONDI VAMSI KRISHNA
(https://www.linkedin.com/in/vamsikrishna-mondi-446a601b1)
2. ANOOBA A (https://www.linkedin.com/in/anooba-a-a547571a0/)
3.CHETAN(https://www.linkedin.com/in/chetansarawgi55/)

TITLE :- Text spotting with cameras

AIM:- We are developing fast text spotting algorithms that can reliable detect and localize any text visible in images acquired by a camera.

Overview of project :

1.The dataset
2.Detecting the car plate
3.Cropping the car plate image
4.Extracting the text from image
5. Loading our output car plate number into excel sheet
6.Implimenting our project in web server

Packages required:

1.Opencv-python

2.date-time

3. Flask

Task 1:-Data set

Method 1:-
To create a dataset We have to collected some thousand of images with there txt file (showing the number location) then Now we have train the model in harcascade for doing this you can refer link https://youtu.be/jG3bu0tjFbk

Method 2:-
You can use the pretrained vamsi.xml to so that the detection will we become easy. For this you need not require to train the model directly you can use

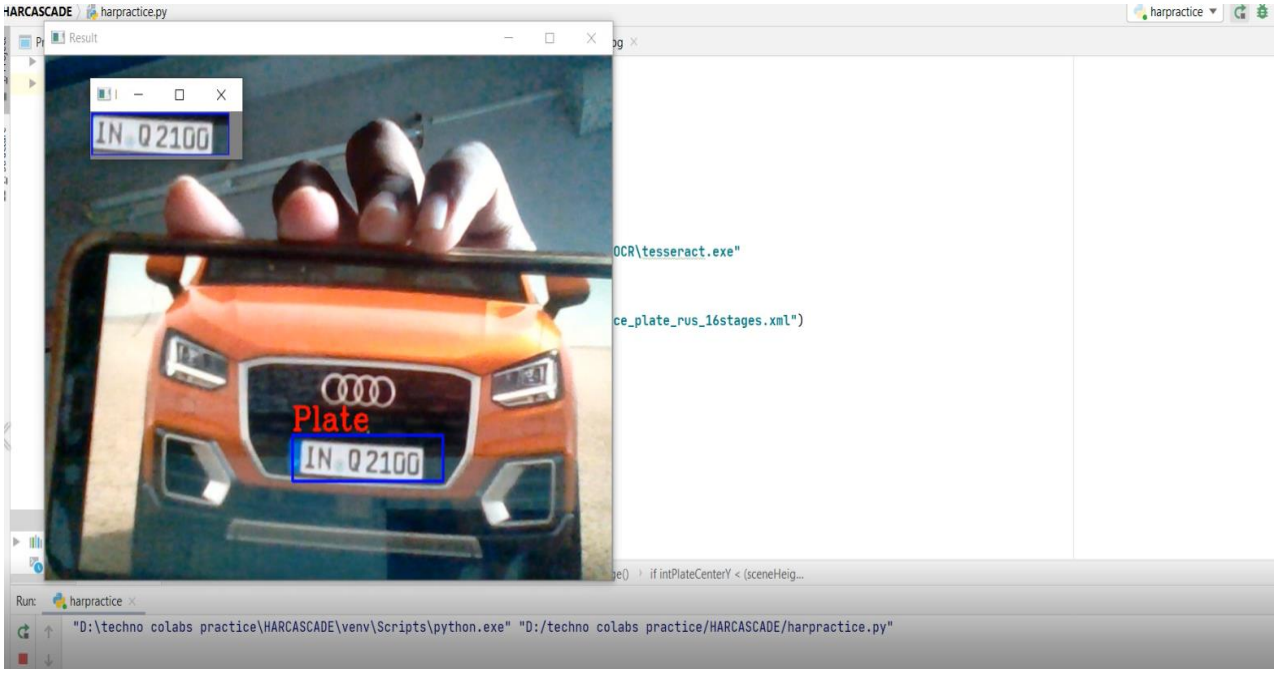
Task 2:- Detecting the car plate

Now after training your CascadeClassifier I am using cv2.CascadeClassifier() to get our trained harcascade which is used for detecting the car plate into a variable named plateCascade so that we use this will your video is running

```
plateCascade = cv2.CascadeClassifier("vamsi.xml")
cap =cv2.VideoCapture(0)
cap.set(3,frameWidth)
cap.set(4,franeHeight)
cap.set(10,150)
while True:
  success , img  = cap.read()
  imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
  numberPlates = plateCascade.detectMultiScale(imgGray, 1.1, 4)
  for (x, y, w, h) in numberPlates:
  area = w*h
  if area > minArea:
    cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)
    cv2.putText(img,"Plate",(x,y5),cv2.FONT_HERSHEY_COMPLEX,1,(0,0,255),2)
    imgRoi = img[y:y+h,x:x+w]
    cv2.imshow("ROI",imgRoi)
```

Here In this code my trained carcade is vamsi.xml . After initializing variable plateCascade we were using the function detectMultiScale to detect all the car plates in the video and pacing it to display as output

OUTPUT:



TASK 3:- Cropping the car plate image

After detecting the in video we have to crop our car plate so that the characters in the car plate will be clear and easily recognized . afte cropping we will also save the the cropped image into Cr.jpg so that this jpg format image can we used for deep learning to detect the characters.

```
if cv2.waitKey(1) & 0xFF ==ord('s'):
    cv2.imwrite("Cr"+".jpg",imgRoi)
    cv2.imshow("Result",img)
```

Output:

Here the Cr.jpg will be looking like this

Task 4:- Extracting the text from image

Extracting text from image I have tried with many thing like pytesserract ,easyocr and many more but I finally used the deep learning algorithm KNN(K-nearest neighbour algorithm to detect the characters more accurately.

To use this algorithm you need packages below:
1.DetectChars
2.DetectPlates
3.Possible Chars
4.Possible plate
5.Preprocess
To train them you also need
1.classifiacations.txt
2.flattened images.txt

You can download all this from my git hub link which I have provided.
So after downloading all these file copy and paste them in the project file where you were working on.

WE have to pass our cropped image Cr.jpg into this algorithm using cv2.imread("Cr.jpg"). After passing out Cr.jpg it will detects the plate and detects all the characters in the number plate exactly and converts that into text .
Here I AM just writing the main code that need to be excuted

Code:

```
SCALAR_BLACK = (0.0, 0.0, 0.0)
SCALAR_WHITE = (255.0, 255.0, 255.0)
SCALAR_YELLOW = (0.0, 255.0, 255.0)
SCALAR_GREEN = (0.0, 255.0, 0.0)
SCALAR_RED = (0.0, 0.0, 255.0)
showSteps = True
def main():

    blnKNNTrainingSuccessful = DetectChars.loadKNNDataAndTrainKNN()

    if blnKNNTrainingSuccessful == False:
        print("\nerror: KNN traning was not successful\n")
        return

    imgOriginalScene = cv2.imread("Cr.jpg")

    if imgOriginalScene is None:
        print("\nerror: image not read from file \n\n")
        os.system("pause")
        return

    listOfPossiblePlates=DetectPlates.detectPlatesInScene(imgOriginalScene)

    listOfPossiblePlates
=DetectChars.detectCharsInPlates(listOfPossiblePlates)

    cv2.imshow("imgOriginalScene", imgOriginalScene)

    if len(listOfPossiblePlates) == 0:
        print("\nno license plates were detected\n")
    else:
        listOfPossiblePlates.sort(key=lambda possiblePlate:
len(possiblePlate.strChars), reverse=True)
        licPlate = listOfPossiblePlates[0]

        cv2.imshow("imgPlate", licPlate.imgPlate)
        cv2.imshow("imgThresh", licPlate.imgThresh)

        if len(licPlate.strChars) == 0:
            print("\nno characters were detected\n\n")
            return


        drawRedRectangleAroundPlate(imgOriginalScene, licPlate)

        print(
            "\nlicense plate read from image = " + licPlate.strChars +
"\n")
```

```
        print("------------")
        mark(licPlate.strChars)

        writeLicensePlateCharsOnImage(imgOriginalScene, licPlate)

        cv2.imshow("imgOriginalScene", imgOriginalScene)

        cv2.imwrite("imgOriginalScene.png", imgOriginalScene)

    cv2.waitKey(0)

    return

def drawRedRectangleAroundPlate(imgOriginalScene, licPlate):

    p2fRectPoints = cv2.boxPoints(licPlate.rrLocationOfPlateInScene)

    cv2.line(imgOriginalScene, tuple(p2fRectPoints[0]),
tuple(p2fRectPoints[1]), SCALAR_RED, 2)
    cv2.line(imgOriginalScene, tuple(p2fRectPoints[1]),
tuple(p2fRectPoints[2]), SCALAR_RED, 2)
    cv2.line(imgOriginalScene, tuple(p2fRectPoints[2]),
tuple(p2fRectPoints[3]), SCALAR_RED, 2)
    cv2.line(imgOriginalScene, tuple(p2fRectPoints[3]),
tuple(p2fRectPoints[0]), SCALAR_RED, 2)

def writeLicensePlateCharsOnImage(imgOriginalScene, licPlate):
    ptCenterOfTextAreaX = 0
    ptCenterOfTextAreaY = 0

    ptLowerLeftTextOriginX = 0
    ptLowerLeftTextOriginY = 0

    sceneHeight, sceneWidth, sceneNumChannels = imgOriginalScene.shape
    plateHeight, plateWidth, plateNumChannels = licPlate.imgPlate.shape

    intFontFace = cv2.FONT_HERSHEY_SIMPLEX
    fltFontScale = float(plateHeight) / 30.0
    intFontThickness = int(round(fltFontScale * 1.5))

    textSize, baseline = cv2.getTextSize(licPlate.strChars, intFontFace,
fltFontScale,
                                         intFontThickness)
    ((intPlateCenterX, intPlateCenterY), (intPlateWidth, intPlateHeight),
     fltCorrectionAngleInDeg) = licPlate.rrLocationOfPlateInScene

    intPlateCenterX = int(intPlateCenterX)
    intPlateCenterY = int(intPlateCenterY)

    ptCenterOfTextAreaX = int(
        intPlateCenterX)

    if intPlateCenterY < (sceneHeight * 0.75):
        ptCenterOfTextAreaY = int(round(intPlateCenterY)) + int(
            round(plateHeight * 1.6))
    else:
        ptCenterOfTextAreaY = int(round(intPlateCenterY)) - int(
            round(plateHeight * 1.6))
    textSizeWidth, textSizeHeight = textSize

    ptLowerLeftTextOriginX = int(
```

```
        ptCenterOfTextAreaX - (textSizeWidth / 2))
    ptLowerLeftTextOriginY = int(
        ptCenterOfTextAreaY + (textSizeHeight / 2))
    cv2.putText(imgOriginalScene, licPlate.strChars,
(ptLowerLeftTextOriginX, ptLowerLeftTextOriginY),
                intFontFace, fltFontScale, SCALAR_YELLOW, intFontThickness)

if __name__ == "__main__":
    main()
```

Output:-

```
1 possible plates found

license plate read from image = 1N02100
```

This the result of the car plate number which we were tring to detect the number plate from the video. Here the number of car plate found are 1 and the number of the plate is displayed.

Task 5:- Loading our output car plate number into excel sheet

Now we have to pass our output into the excel sheet . For doing this we have to create a csv file ,I have creates a file named vamsi.csv where I will store the time and number of the car plate where time is noting but the time at which we are tring to detect the car plate.In the first column time will be there and in the second column number of the plate will be stored

This was the code for storing your number plate text into excel sheet where I am naming it as mark():

Code:

```
def mark(name):
    with open('vamsi.csv','r+')as f:
        mydata=f.readlines()
        namelist=[]
        for line in mydata:
            entry=line.split(',')
            namelist.append(entry[0])
        if name not in namelist:
            now=datetime.now()
            dtstr=now.strftime('%H:%M:%S')
        f.writelines(f'\n{dtstr},{name}')
```

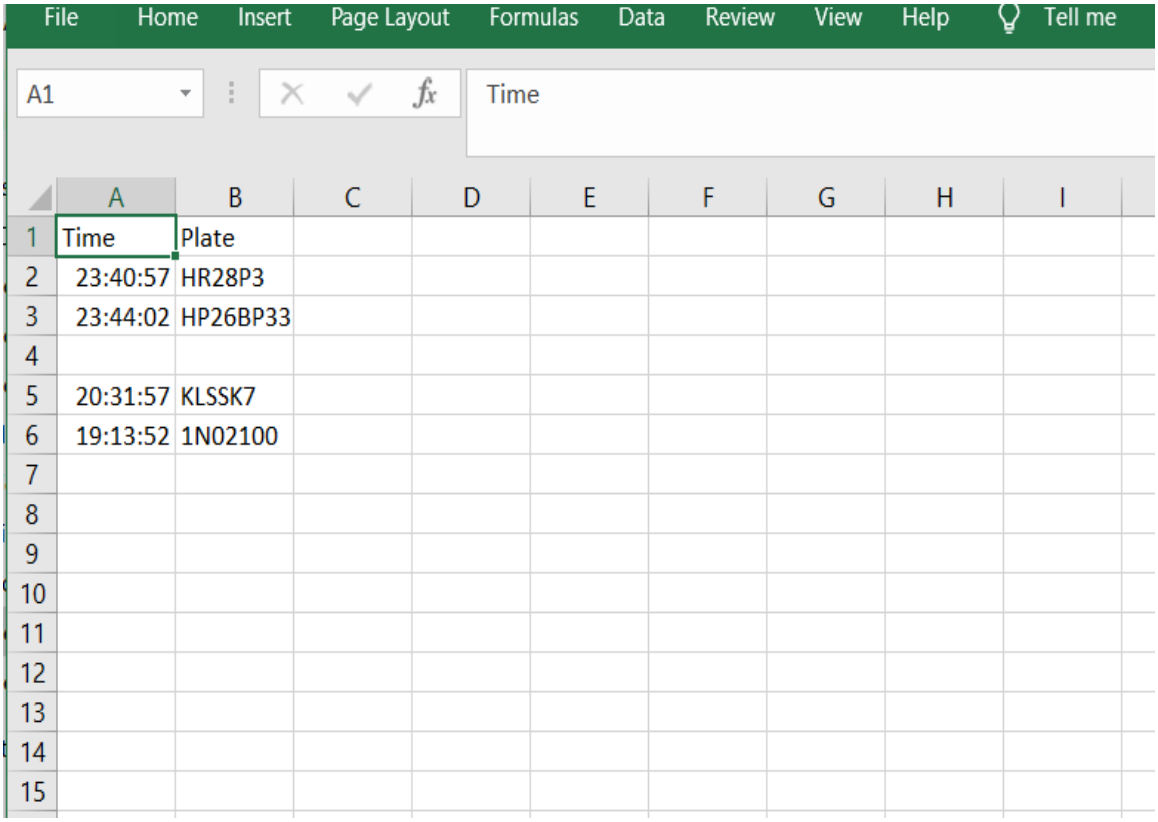
Here in this code I am opening the file vamsi.csv and fetching the data that, the number plate which was present already should not repeat so that the car will plate will be stored only once in excel sheet.

The output which we got after excuting task 4 will me sent to this function `mark(licPlate.strChars)` so here the number in plate will we stored in excel sheet.

CODE:-

```
mark(licPlate.strChars)
```

Output:



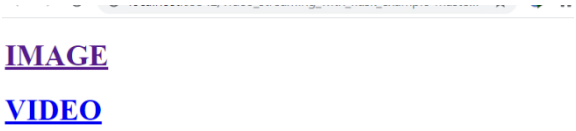
Task-6:-Implimenting our project in web server

For implementing your program in web I am using flask .I am using only flask because it made my work easier to connect the html and program. First to do flask we have to create our web page by using html and css tehn after we will write another program to which connect our html and python file .

MY html page code:-

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>choose your button</title>
</head>
<body>
<a href=""><h1>IMAGE</h1></a>
<a href="video_feed"><h1>VIDEO</h1></a>
</body>
</html>
```

Output of html:-



Now here after creating html page we have to create another html file named video.html

```
<html>
  <head>
    <title>Video Stream</title>
  </head>
  <body>
  <h1>Video Streaming </h1>
  <img id="bg" src="{{ url_for('video_feed') }}">
  </body>
</html>
```

Now comes our main thing that was linking your python file and html .I am saving this file as **app.py**

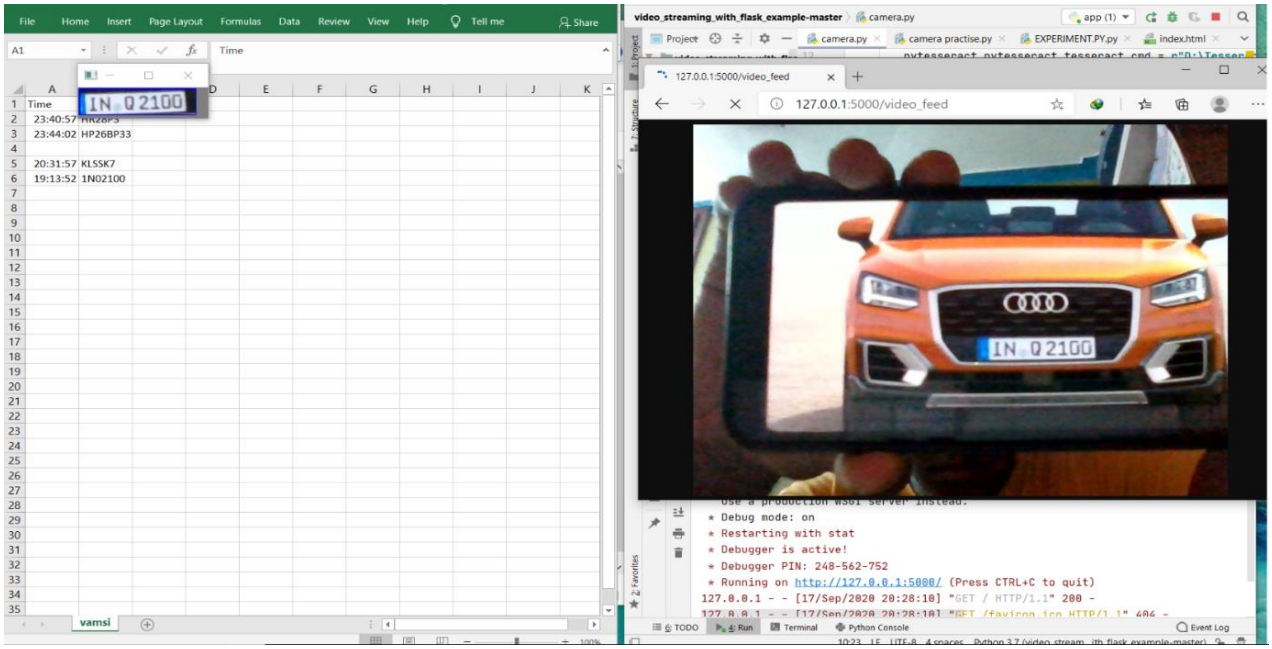
```
from flask import Flask, render_template, Response
from camera import VideoCamera
app = Flask(__name__)
@app.route('/')
def index():
    return render_template('index.html')
def gen(camera):
    while True:
        img= camera.get_frame()
        yield (b'--frame\r\n'b'Content-Type: image/jpeg\r\n\r\n' +img +
b'\r\n\r\n')
@app.route('/video_feed')
def video_feed():
    return Response(gen(VideoCamera()),mimetype='multipart/x-mixed-
replace; boundary=frame')
if __name__ == '__main__':
    app.run(debug=True)
```

Here the camera.py is the file were the main program (our car recognition is there)

Now for detecting characters here I am using pytessract which showed me positive results here and after detecting the carplate number we were also storing the same result in vamsi.csv so that we can see which cars were recognized and passed.

This was our complete project

Final output:



THANK YOU YASIN SHAH SIR FOR GIVING THIS OPPORTUNITY FOR US .