



Number Plate Detection of Vehicles

Overview

This Python project implements a real-time car number plate detection system using OpenCV. It utilizes Harcascade classifiers for detecting license plates in live video streams captured through a webcam.

Aim:

To develop an automated system that can detect and recognize vehicle license plate numbers from images or video feeds using image processing and optical character recognition techniques.

Procedure:

- Install the required libraries:

```
pip install opencv-python
```

- Download the pre-trained Harcascade XML file (haarcascade_russian_plate_number.xml) and place it in the model directory.

- Run the Script:

```
python number_plate.py
```

- The webcam window will open, displaying the live video stream.
- As the video runs, the system will detect license plates in the frames and draw rectangles around them. The detected plate regions will be highlighted as “Number Plate”.
- Press the ‘s’ key to save the cropped image of the detected plate. The saved images will be stored in the plates directory.

Project Structure:

car-number-plate-detection/

```
|
|— model/
|   └─ haarcascade_russian_plate_number.xml # Pre-trained model file for plate
detection
|
|— plates/
|   └─ scanned_img_0.jpg # Sample detected plate images
|   └─ scanned_img_1.jpg
|   └─ ... (additional detected plate images)
|
|— LICENSE          # File specifying the project's license (e.g., MIT License)
|— README.md        # Project README file
|— easy_OCR.ipynb    # Jupyter Notebook for Optical Character Recognition (OCR)
|— number_plate.py   # Python script for car number plate detection
└─ requirements.txt  # File listing project dependencies
```

The minimum area threshold for plate detection can be adjusted in the script (`min_area` variable). Experiment with different cascade models or tuning parameters for improved plate detection accuracy. Consider adding post-processing steps for character segmentation and optical character recognition (OCR) to extract text from detected plates.

Program:

```
import cv2

harcascade = "model/haarcascade_russian_plate_number.xml"

cap= cv2.VideoCapture(0)

cap.set(3, 640) #width
cap.set(4, 480) #height

min_area = 500
count=0

while True:
    success, img = cap.read()

    plate_cascade = cv2.CascadeClassifier(harcascade)
    img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    plates = plate_cascade.detectMultiScale(img_gray, 1.1, 4)

    for (x, y, w,h) in plates:
```

```

area = w * h

if area > min_area:
    cv2.rectangle(img, (x,y), (x+w, y+h), (0,255,0), 2)
    #(0,255,0)is RGB(red,green, blue)
    cv2.putText(img, "Number Plate", (x,y-5),
cv2.FONT_HERSHEY_COMPLEX_SMALL, 1, (255, 0, 255), 2)
    #txtfont, skill, color, 2=thcknss of txt

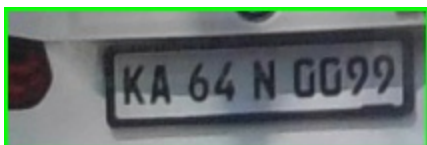
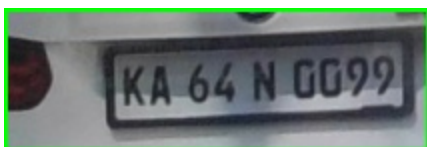
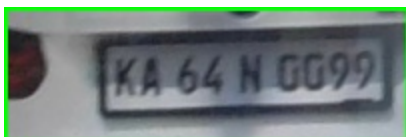
    img_roi = img[y : y+h, x:x+w] #to crop only no.plate prtn
    #roi(region of interest)
    cv2.imshow("Roi", img_roi)

cv2.imshow("Result", img)

if cv2.waitKey(1) & 0xFF == ord('s'):
    cv2.imwrite("plates/scanned_img_" + str(count) + ".jpg",
img_roi)
    cv2.rectangle(img, (0,200), (640,300), (0,255,0), cv2.FILLED)
    cv2.putText(img, "Plate Saved", (150, 265),
cv2.FONT_HERSHEY_COMPLEX_SMALL, 2, (0, 0, 255), 2)
    cv2.imshow("Results",img)
    cv2.waitKey(500)
    count += 1

```

Output:



Recognized Number Plate: KA 64 N 0099



Result:

Thus the detect and recognize vehicle number plates from images using **OpenCV** for image processing and output was verified successfully.