Automatic Number Plate Recognition with OpenCV and OCR

Abstract

License plate recognition is one of the most exciting and stimulating research topics in recent years. The license plates of each country are shown to be of different shapes and sizes, as well as different colors. In India, utmost mutual vehicle registering plates used yellow/white as the plate color and black as the letter color. This article used for the development of automatic license recognition (ANPR) by using Python, OpenCV and the EasyOCR library. ANPR technology used to identifies and extracts vehicle license from photos or videos. OpenCV used for image pre-processing and control, while EasyOCR stands for optical character recognition (OCR) to recognize license plates. This article includes following steps to create an ANPR system, including image creation, character classification, and OCR. Issues related to strategic implementation and resource development are also discussed. It also shows an easy way to segment all the alphabets and figures used in license plates. The first is to locate the license plate, and the second is to slice all the words and alphabets to recognize each no individually.

Introduction

Today's generation of vehicles is numerous all over the globe. Therefore, it is very necessary to put an eye on your vehicle. In today's era, every vehicle could be tracked without having to manually search for it using a computer, thus increasing correctness. So, the Automobile License Plate Recognition System is a skill used to identify license plates from video captured by cameras. Use methods such as license plate withdrawal, division, and letters/words acknowledgment. The scheme is a mixture of hardware and application that captures license plates, transmits them, and converts them into images. This technology can be used with any gateway. Therefore, the system may be correct if the images taken out from the license plate taken by the camera are clean and noticeable. Images you use must be of very high resolution. The license plate is a design with a great deal of disparity variation. If the number plate is too alike to the background, that will be tough to locate. Illumination and disparity change as the angle of incidence of light changes. Morphologic actions are used to excerpt dissimilarity features inside the plate.

Automatic Number Plate Recognition (ANPR) is a technology that automates the identification of Vehicle licence plates form videos and images. By this technology, finds applications in Various field like traffic management, parking lot access control, automated tolling systems. For ANPR system using Python libraries**: Open CV** for image processing and **EasyOCR** for Optical character Recognition (OCR).

What’s covered:

1. Reading in images to OpenCV.
2. Detecting number plates using Edge Detection.
3. Extracting number plate text using OCR.

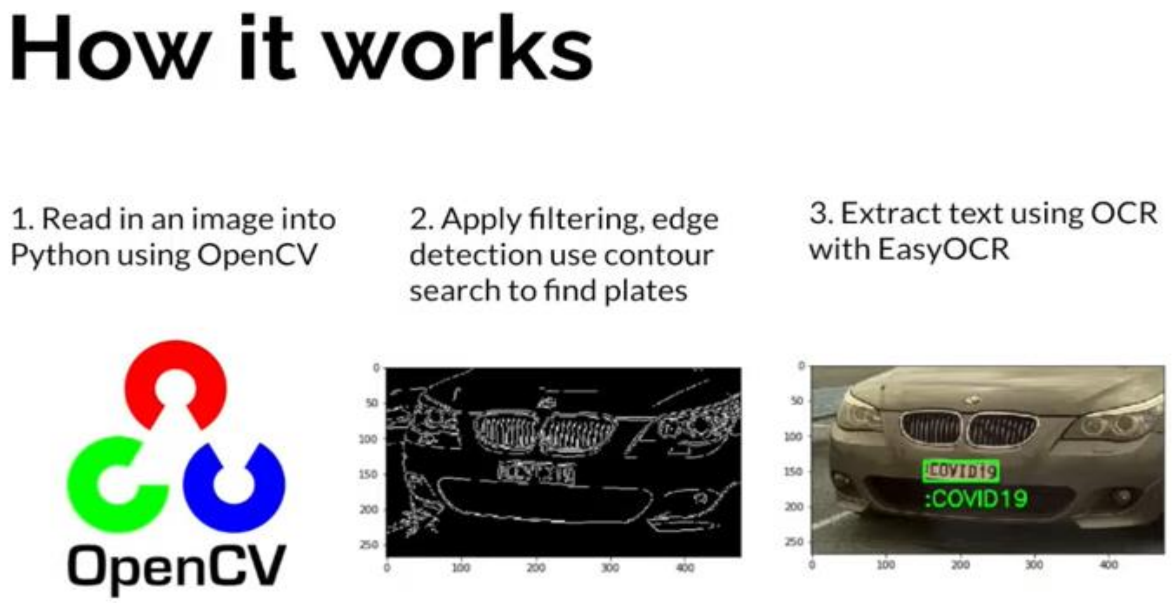
Challenges:

Developing a robust ANPR System comes with its share of challenges:

1. **Image quality variations:** Licence plates can appear in different lighting conditions, with varying resolution, and by dirt or weather elements.
2. **Plate format variations:** Different countries and regions have different licence format, size, colour scheme and character fonts.
3. **Occlusions:** Objects like bumpers or two bars can hinder character recognition.
4. **Real time processing demands:** video stream for real-time ANPR application will requires efficient algorithms for performance.

Solution:

The proposed ANPR system consist of following stages:



How it works

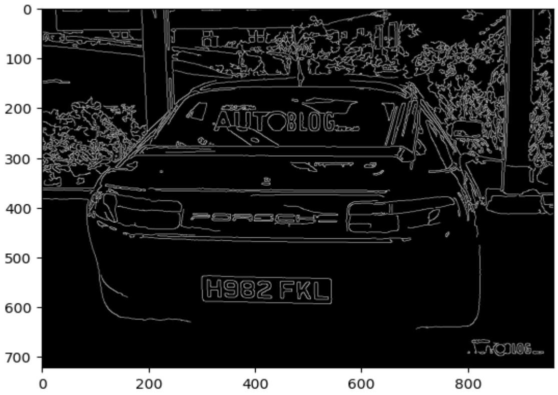
1. Read in an image into Python using OpenCV

2. Apply filtering, edge detection use contour search to find plates

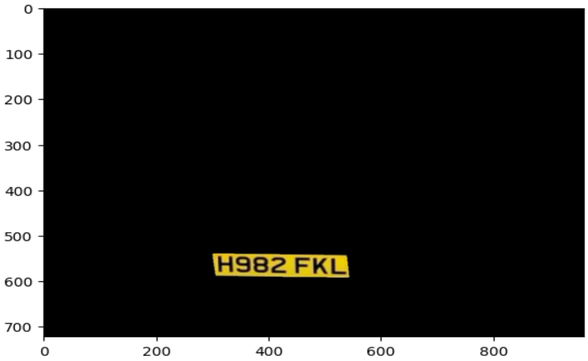
3. Extract text using OCR with EasyOCR

1. **Read in image, Grayscale and Blur:**
   - Read the image containing the vehicle using OpenCV's cv2.imread.
   - Convert the image to grayscale using cv2.cvtColor.

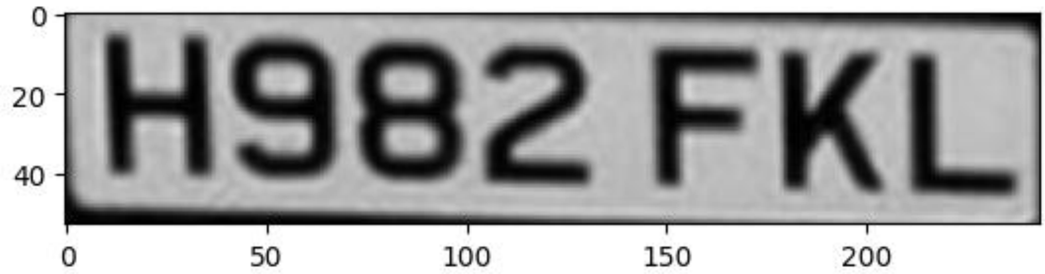
2. **Apply filter and find edges for localization.**
    - Apply noise reduction techniques like blurring or bilateral filtering by using cv2.bilateralFilter.
    - Apply edges detection by cv2.canny to identify edge in grayscale image.
    - Enhance contrast using histogram equalization or adaptive thresholding.



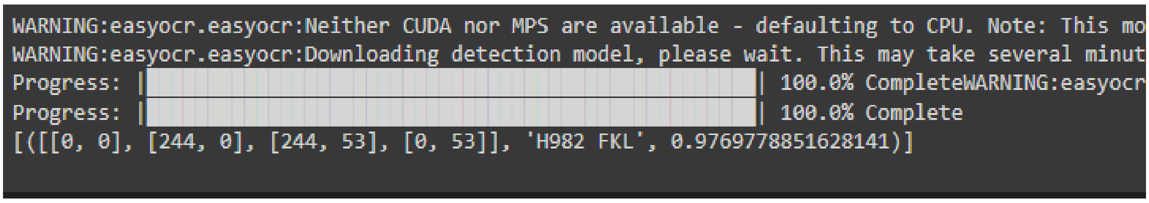
3. **Find Contours and Apply Mask:**
    - Find contours using cv2.findContours.
    - Filter contours based on aspect ratio and area to isolate potential license plates.
    - Draw a bounding box around the most likely license plate contour.



4. **Character Segmentation:**
    - Extract the license plate region from the original image using the bounding box coordinates.
    - Segment the license plate ROI into individual character images using vertical projection profiles or morphological operations.

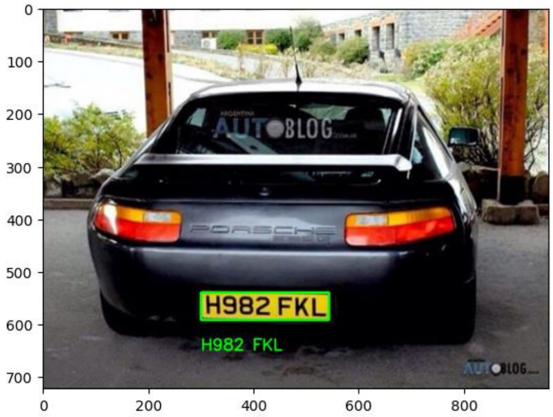


5. **Character Recognition (OCR) to read text:**
   - Use EasyOCR to recognize characters in each segmented image. EasyOCR provides pre-trained models for various languages.
   - Consider confidence scores from EasyOCR to filter out low-confidence detections.



```
WARNING:easyocr.easyocr:Neither CUDA nor MPS are available - defaulting to CPU. Note: This mo
WARNING:easyocr.easyocr:Downloading detection model, please wait. This may take several minut
Progress: |██████████████████████████████████████████████████| 100.0% CompleteWARNING:easyocr
Progress: |██████████████████████████████████████████████████| 100.0% Complete
[([[0, 0], [244, 0], [244, 53], [0, 53]], 'H982 FKL', 0.9769778851628141)]
```

6. **Post-processing for Render Result:**
   - Implement logic to handle potential errors like character substitutions or missing characters.
   - Display the recognized license plate number on the original image or store it in a text file.



Conclusion:

By our evaluation of multiple articles, we concluded that there are various license plate recognition technologies available. Sobel border finding method, reflex number plate appreciation, a new technique to spot edges and seal dumps smaller from eight pixels, function of each stage, car license plate identification, and recognition. So from this article we presented a foundational ANPR system built with Python, OpenCV, and EasyOCR. While this approach offers a basic implementation, several enhancements can elevate performance:

- **Training data:** Train the EasyOCR model with a dataset of license plates from your specific region to improve accuracy for local plate formats.
- **Real-time optimization**: Optimize the code for real-time video processing using techniques like image downscaling or hardware acceleration.
- **Lighting condition handling:** Develop strategies to address variations in lighting conditions by adjusting pre-processing steps.
- **Advanced error handling:** Implement more robust error correction mechanisms to handle complex recognition mistakes.