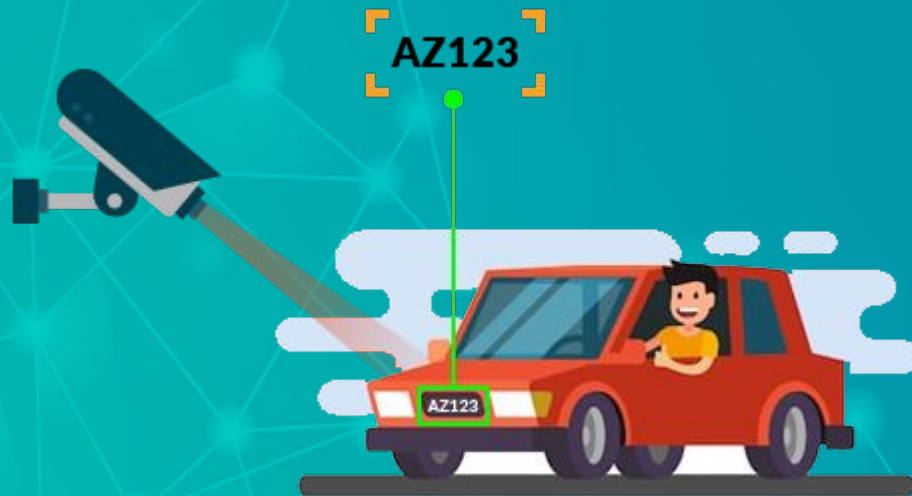


Car Name Plate Detection

User Manual



Author: Sukhleen Singh Virk

Table of Contents

- List of Contents here
- Acknowledgements
- About Me
- About my journey
- About App
- How to use it
- Demo Video
- Topic walkthrough

Acknowledgements

- I would like to express my sincere gratitude to my parents, Mentors, Dr. Ken Khan, Group Members, Friends, Sites etc for helping me in my project, and I would thank the team for giving me opportunity to do this wonderful project of car name plate detection.

About Me..

- **Student's Introduction**

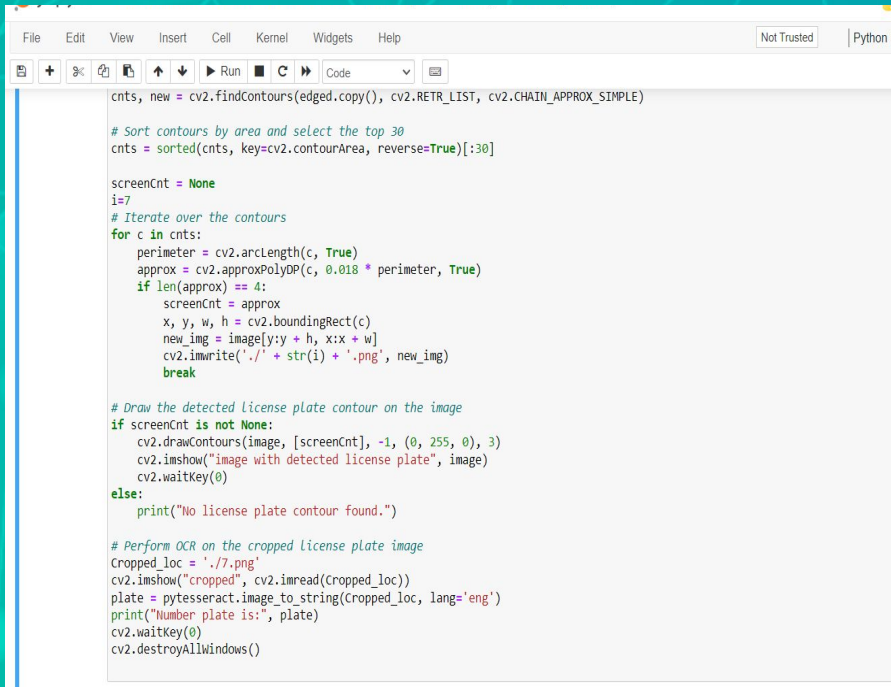
My name is Sukhleen Singh Virk and I am from class 12th, with subject PCM and CS, studying in Khaitan public School, Rajendra Nagar, Ghaziabad.

About My Internship Journey with Clevered..

- Any photographs from sessions etc.

- Your Internship Experience with Clevered

It is a wonderful experience, with the Clevered Internship Course for Artificial Intelligence, I got to learn many new things in this course.

A screenshot of a Jupyter Notebook interface. The top bar shows 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', 'Help', 'Not Trusted', and 'Python 3'. Below the toolbar, there's a code editor with Python code. The code uses OpenCV to find contours, sort them by area, and select the top 30. It then iterates over these contours to find a license plate by checking for a rectangular shape with a perimeter of approximately 4. If found, it crops the license plate and performs OCR using pytesseract. The code is as follows:

```
cnts, new = cv2.findContours(edged.copy(), cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)

# Sort contours by area and select the top 30
cnts = sorted(cnts, key=cv2.contourArea, reverse=True)[:30]

screenCnt = None
i=7
# Iterate over the contours
for c in cnts:
    perimeter = cv2.arcLength(c, True)
    approx = cv2.approxPolyDP(c, 0.018 * perimeter, True)
    if len(approx) == 4:
        screenCnt = approx
        x, y, w, h = cv2.boundingRect(c)
        new_img = image[y:y+h, x:x+w]
        cv2.imwrite('./' + str(i) + '.png', new_img)
        break

# Draw the detected license plate contour on the image
if screenCnt is not None:
    cv2.drawContours(image, [screenCnt], -1, (0, 255, 0), 3)
    cv2.imshow("image with detected license plate", image)
    cv2.waitKey(0)
else:
    print("No license plate contour found.")

# Perform OCR on the cropped license plate image
Cropped_loc = './7.png'
cv2.imshow("cropped", cv2.imread(Cropped_loc))
plate = pytesseract.image_to_string(Cropped_loc, lang='eng')
print("Number plate is:", plate)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

About App..

- App's Main Menu

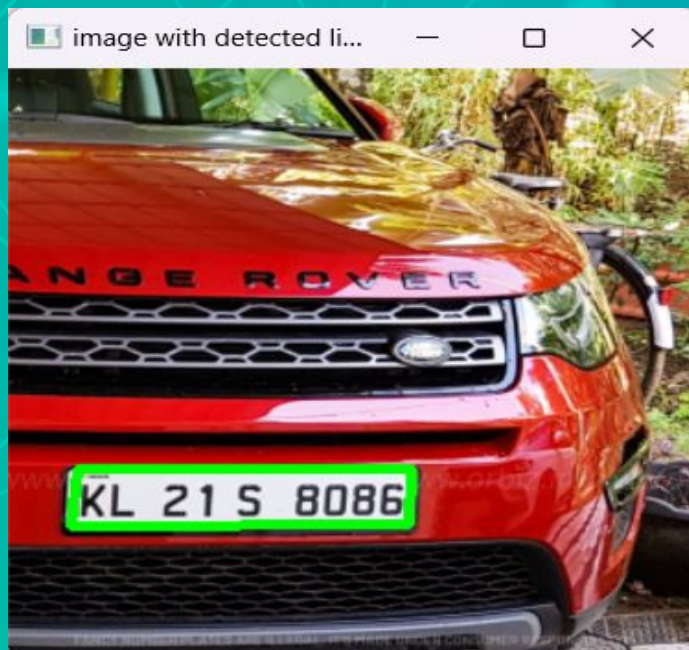
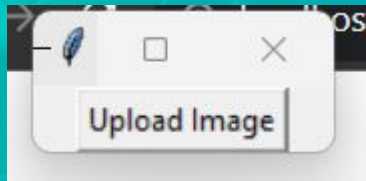
We have to upload the image of the car and then it will detect the number plate and display it on the screen.

- App's Introduction

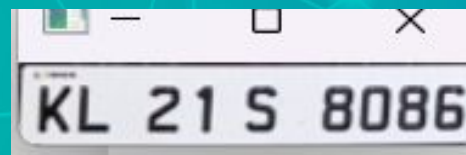
Car name plate detection which helps to read the number plate of vehicle by detecting it, which is helpful in parking management or vehicle information tracking

How do I use the App?

Click on the upload button to select the car image you want.



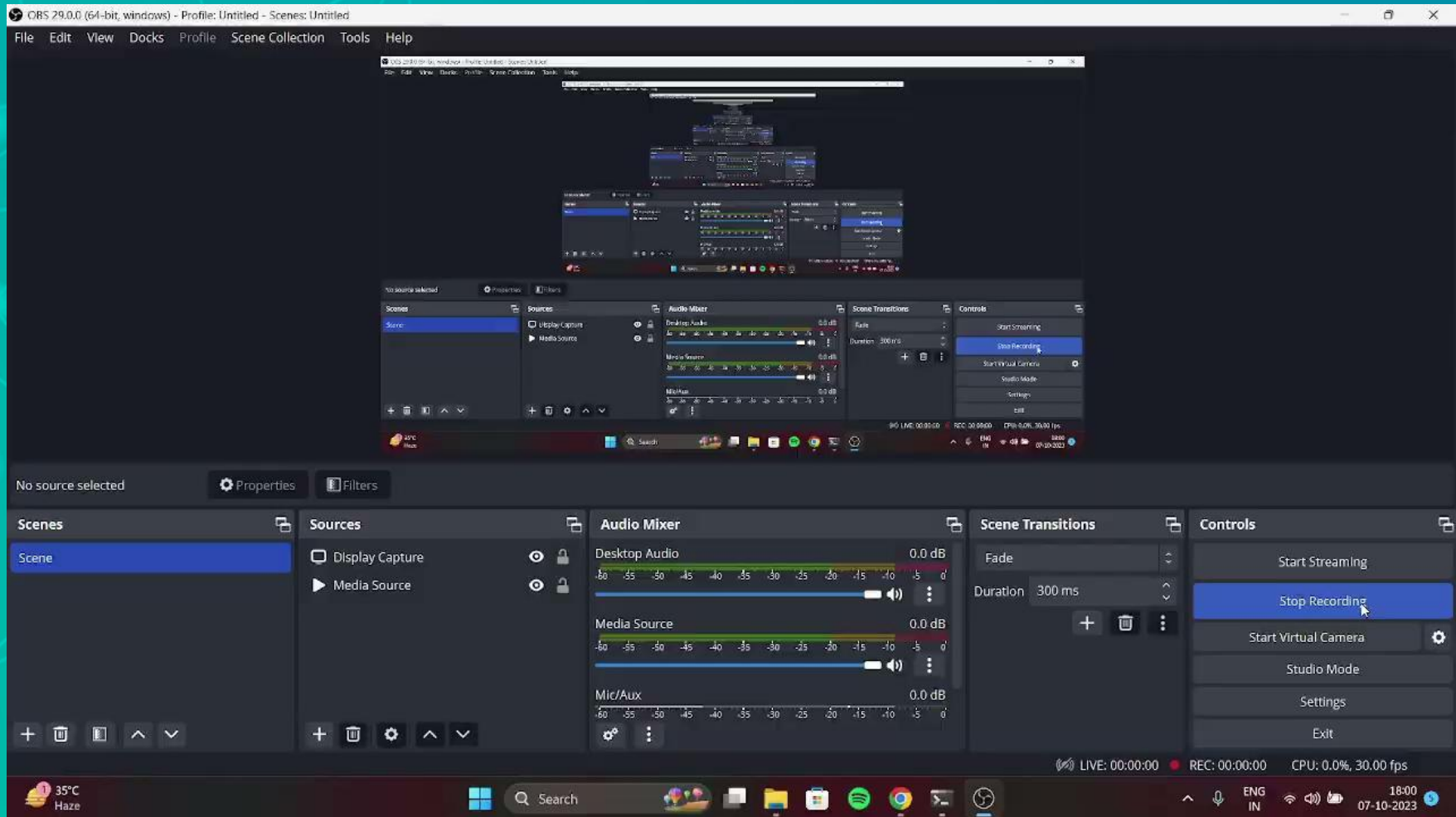
Then after uploading the image it will detect the area of the number plate with the cropped image of the plate and then number will be detected as a result.



Number plate is: KL 21S 8086

Demo Video

This is the video of working of the application.



Toolkit Walkthrough

https://docs.google.com/spreadsheets/d/1fBazmxsEkWnV2ZldFb3nuQF0vR_ZlyMA/edit#gid=1219324219

Event Handlers Matrix

S. No.	Element	Action	What happens?
1	Heading of the App		Shows the project heading
2	Upload image button	"click"	Image is being processed
3	Result		The detected number plate is printed



Thank you!