Automatic Number Plate Recognition System

A Computer Vision GUI application

Amit Kumar Thakur

B.Tech Electronics & Communication Engineering
Department of Electronics & Communication Engineering
School of Engineering & Technology
Jaipur National University

Outlines

Automatic Number Plate Recognition System

1. A computer-vision application
2. Automatic: One time configuration required
3. Detects vehicle licence plate
4. Detects vehicle licence number
5. A fast, light, and multi-platform software application
6. Developed using open-source tools
7. Can be run on embedded systems

Figure 1: Number Plate Recognition

Figure 2: Detected Plate

Figure 3: Detected Characters

Intelligent Transportation System

- Measurement of travel time
- Calculation of traffic volume
- Estimation of traffic pattern

Figure 4: Intelligent Transportation System

Public Transit Security System

- Identifying suspicious vehicles
- Tracking location of vehicles

Figure 5: Security Camera Monitoring

Tolling System

- Detect vehicle licence number
- Detect type of vehicle
- Calculate toll charges

Figure 6: Toll Gate

Vehicle Parking & Entry Access Monitoring

- Detect vehicle licence number
- Allow selected number of vehicles
- Parking charges calculations

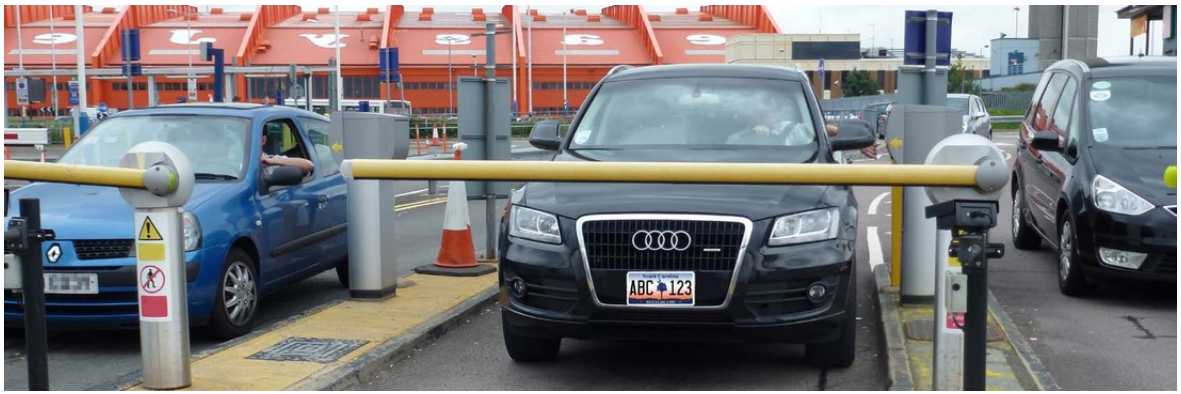

Figure 7: Entry Access Monitoring

Flow Chart

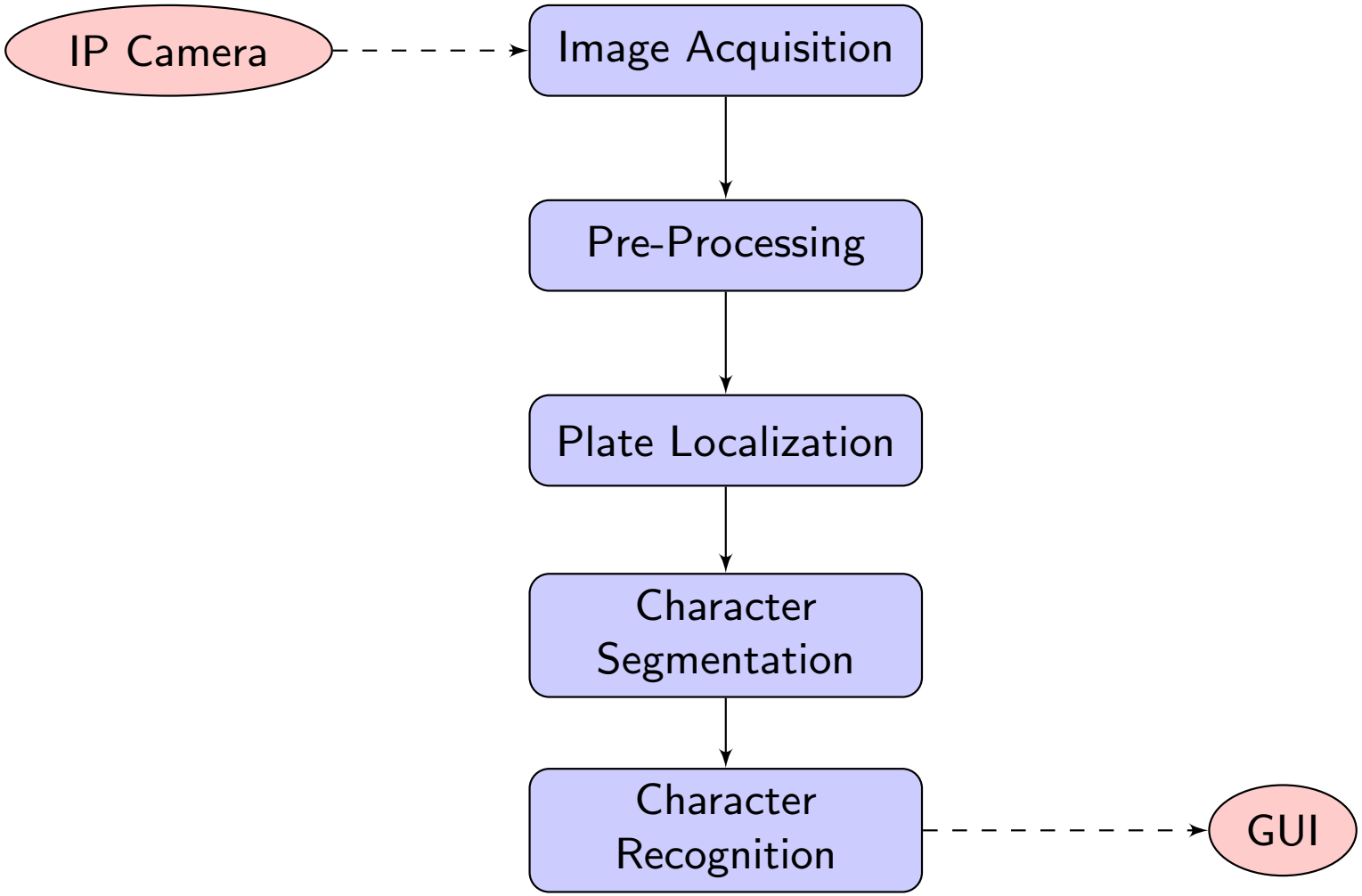

Figure 8: Flow Chart of Techniques used

Tools Used

- **OpenCV C++ Library**
  - Machine Learning & Image Processing Library
  - Open-source
  - Implemented in C++. Bindings in Java, Python, C
- **Qt GUI Framework**
  - Cross-platform application and UI framework
  - Open-source
  - In C++.
- **Tesseract Classifier**
  - Open-source
  - Maintained by Google
  - API in C++.

Figure 9: OpenCV

Figure 10: Qt

Figure 11: Tesseract

Hardware

- 720p IP Camera
- 25-30 fps
- Must have high shutter speed to avoid blur
- Can be connected via Internet

Figure 12: IP Camera

Software: cv::VideoCapture (OpenCV)

- Class for video capturing from video files or cameras
- IP camera video stream URL or video file path can be passed as argument to the constructor

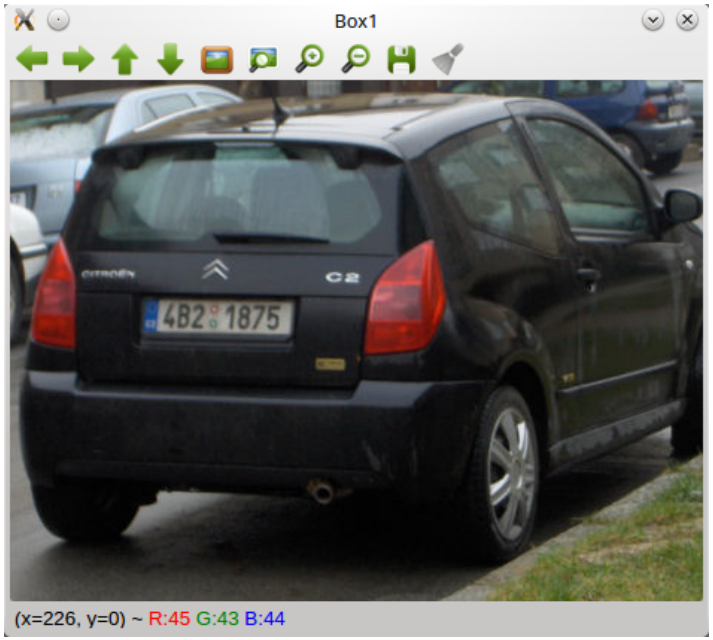

Figure 13: Video Capture

Colourspace conversion:

- **cv::cvtColor()** (Opencv)
- RGB (3 channel) to Grayscale (1 channel)

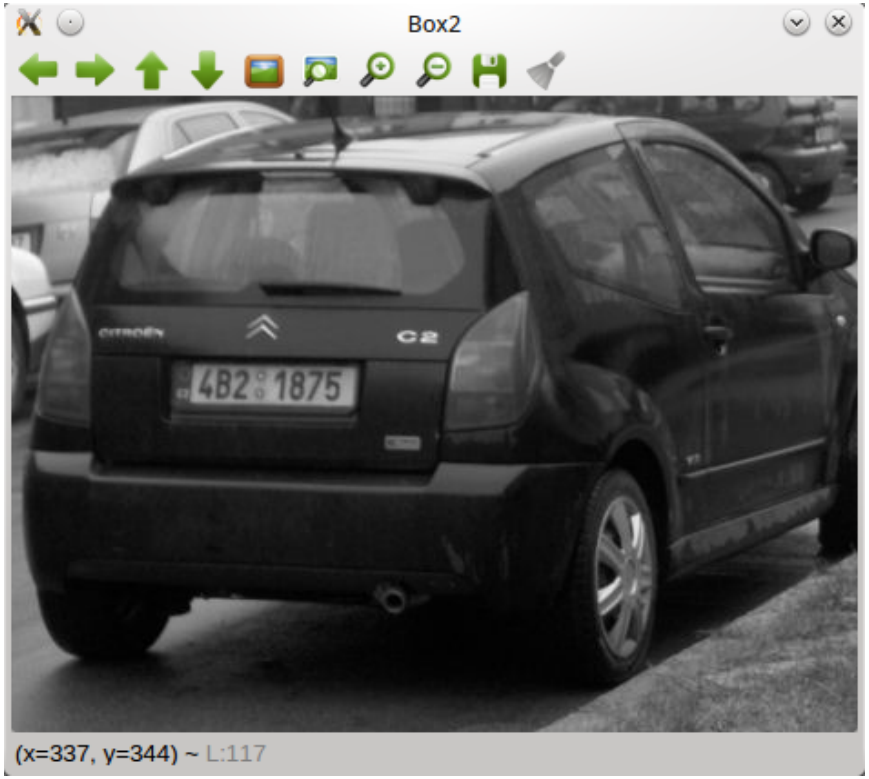

Figure 14: Grayscale Image

Plate Localization

Sobel Filter:

- Plate has high density of vertical edges of characters
- Find the first horizontal derivative
- **cv::Sobel()** (OpenCV)

Figure 15: Sobel Filtered Image

Plate Localization

Thresholding

- Grayscale (0-255) to binary (0-1) conversion
- **cv::threshold()** (OpenCV)

Morphological Closing

- Remove blank spaces between each vertical edge line
- **cv::morphologyEx()** (OpenCV)

Figure 16: Threshold Image

Plate Localization

Contour Analysis

- Find all contours in the binary image
- **cv::findContours()** (OpenCV)
- Apply size constraints: Aspect Ratio, area etc.
- Draw bounding rectangle using **cv::boundingRect()** (OpenCV)

Figure 17: Contours Detection

Equalization & Median Blurring

- **cv::equalizeHist()** (OpenCV): To improve contrast
- **cv::medianBlur()** (OpenCV): To avoid rough edges and small noise

Figure 18: After Equalization and Blurring

Thresholding

- **cv::threshold()** (OpenCV): To get binary image
- Otsu algorithm

Figure 19: Plate Thresholding

Character Segmentation

Contours Detection

- Find all contours in the binary image
- **cv::findContours()** (OpenCV)
- Apply size constraints: Aspect Ratio, area etc.
- Draw bounding rectangle using **cv::boundingRect()** (OpenCV)

Figure 20: Character Segmentation

Tesseract Classifier

- Can be trained using training image data of different characters
- **tesseract::TessBaseAPI** (Tesseract): A class
- **tess.SetPageSegMode()** : Using single character mode

Figure 21: Recognized Characters

Conclusion

ANPRS

1. Implemented using image processing techniques
2. No hassle of installing detecetion hardware on every vehicle like RFID
3. Accuracy can be improved using training with larger variety of image data
4. Can be integrated into any kind of application from small embedded system to large end-frame computers
5. Portable application: For Linux, Windows, Solaris & Mac-OSX

References

Papers

1. **A Real Time Vehicle's License Plate Recognition System**
   http://www.computer.org/csdl/proceedings/avss/2003/1971/00/19710163-abs.html
2. **Algorithmic and Mathematical Principles of ANPRS**
   http://javaanpr.sourceforge.net/anpr.pdf

Project Website:

- http://amitthakur.org/projects/anprs/

Queries & Discussion

Thank You!

Amit Kumar Thakur
B.Tech Electronics & Communication Engineering
8th Semester, Roll No: 7EC-13
Department of Electronics & Communication Engineering
School of Engineering & Technology
Jaipur National University