

2018

Automatic License Plate Recognition System

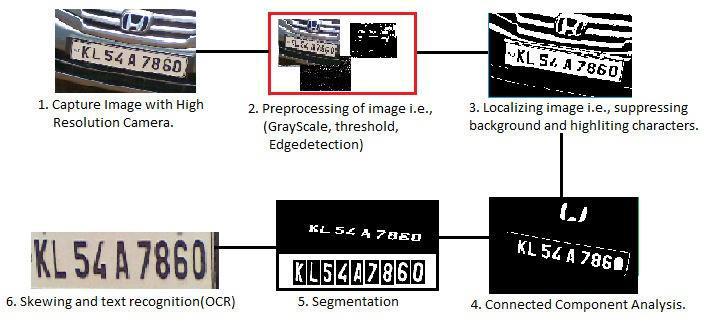
Bala Chander Digumarthi

**Automatic License Plate Recognition System**

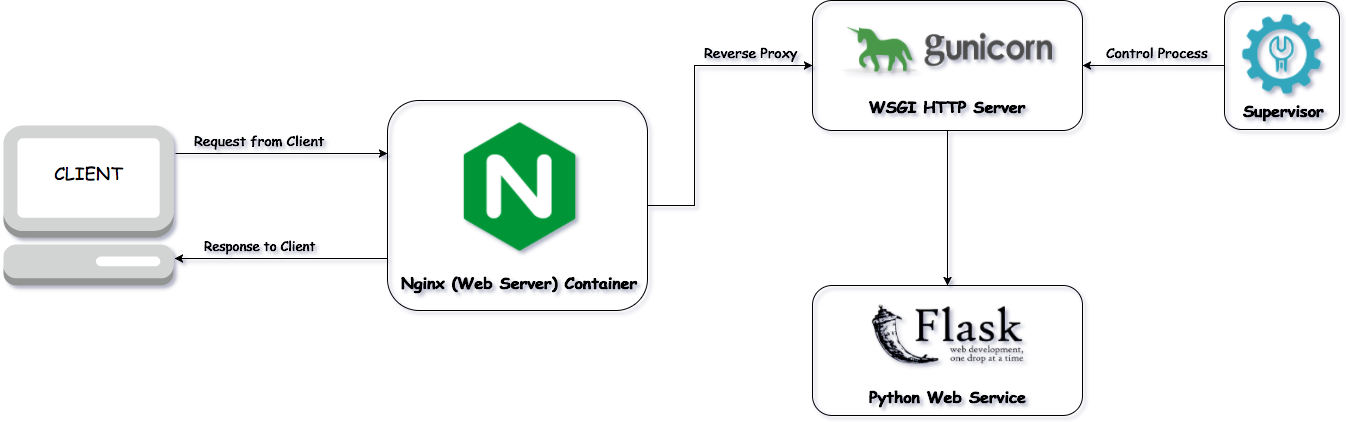
**Introduction:**

Automatic License Plate Recognition system is a real time embedded system which automatically recognizes the license plate of vehicles captured by cameras. There are many applications ranging from complex security systems to common areas and from parking admission to urban traffic control. Automatic license plate recognition (ALPR) has complex characteristics due to diverse effects such as of light and speed. Most of the ALPR systems are built using proprietary tools like Mat lab etc. This approach presents an alternative method of implementing ALPR systems using Free Open Source Software including Python and the Open Computer Vision Library.

**Flow Chart Diagram:**



**System Architecture:-**

****

**Image Processing Steps:**

Step1:

1. Capture The image of the vehicle is captured using a high resolution photographic camera. A better choice is an Infrared (IR) camera. The camera may be rolled and pitched with respect to the license plates.



**Step 2:**

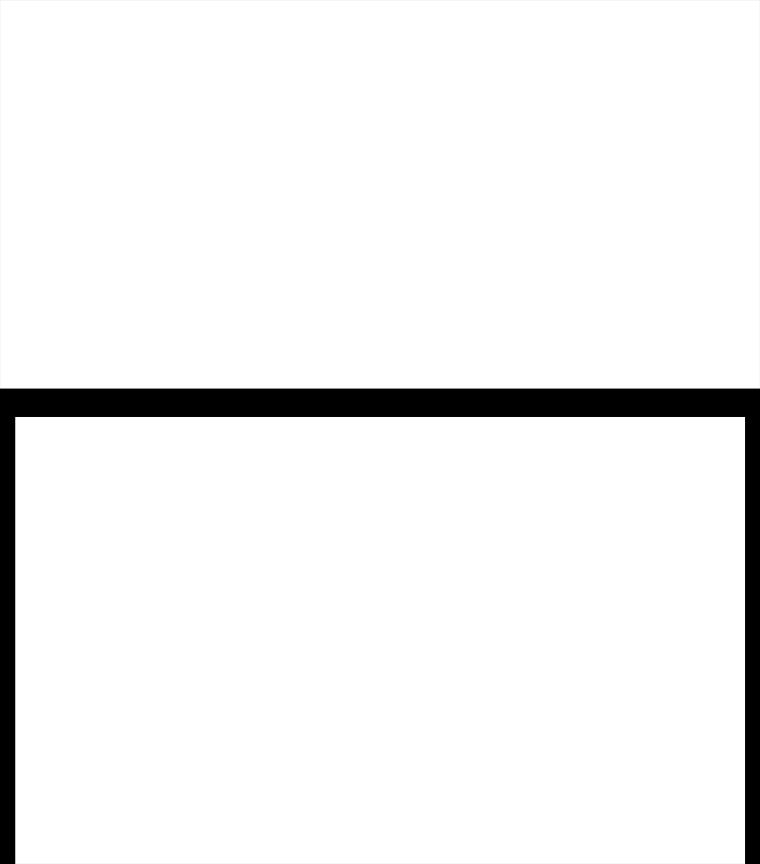
Pre-processing the image by enhancing, resizing and finding connected pixels by using set of image processing algorithms.

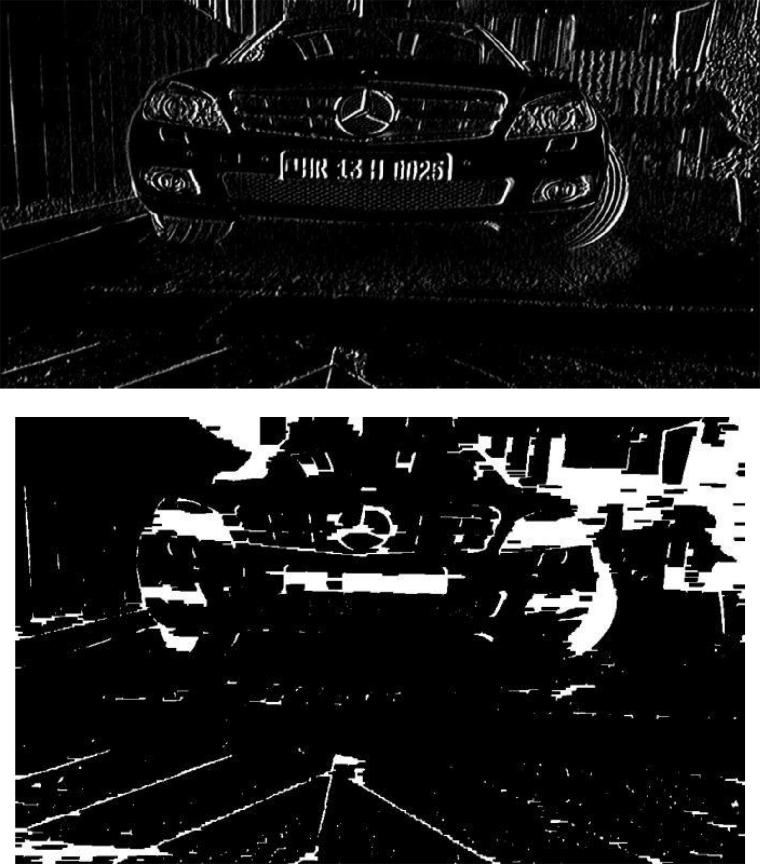


**Step 3:**

Localizing the image i.e., suppressing of background and highlighting the characters this is done by process

called “**Threshold**”.





**Step 4:**

**Connected Component Analysis.**

To eliminate undesired areas on image, a connected component algorithm is first applied to the binarized plate. Connected component analysis is performed to identifythe characters in the image. Basic idea is to traverse through the image and find the connected pixels. Each of the connected components (blobs) are labelled and extracted.

**Step 5:**

**Segmentation.**

Segmentation is the process of cropping out the labelled blobs. These blobs are expected to be the required portion of the license number. A special algorithm called Image Scissoring is introduced here. In this algorithm, the license plate is vertically scanned and scissored at the row on which there is no white pixel and the scissored area is copied into a new matrix.



****



**Step 6:**

**Skewing.**

Skewing is the mechanism of transforming the image by rotation (Automated).

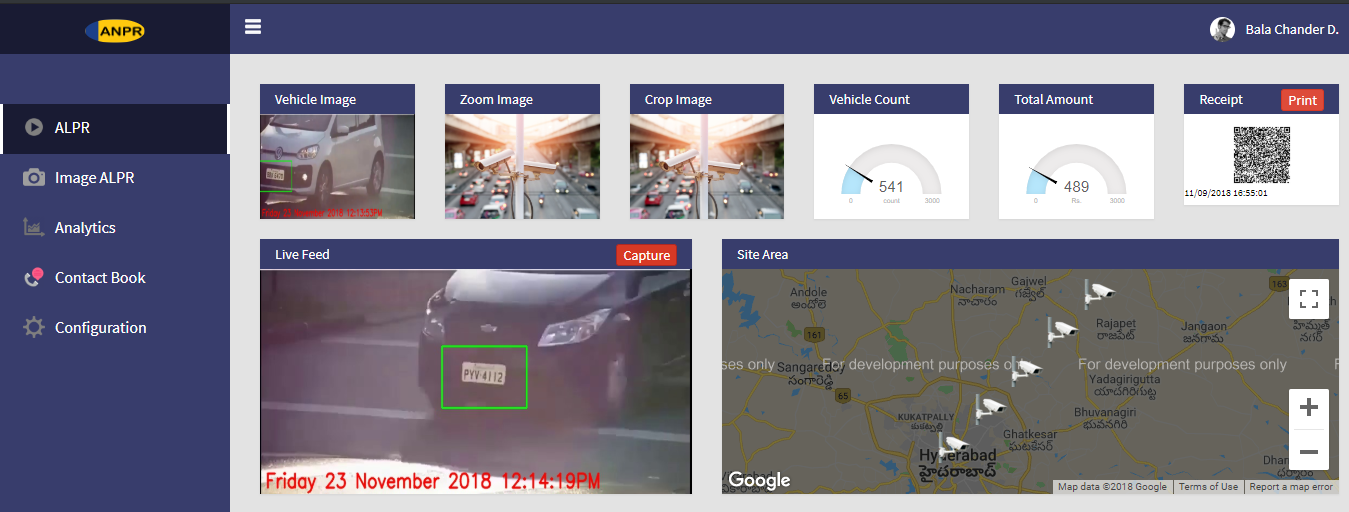
**Step 7:**

**Character Recognition (OCR).**

Skewed image is then sent to Tesseract OCR to get the text on License Plate i.e., License Number.

**Proposed System's.**

1. **Vehicle Hot List**: We could use this ALPR application for vehicle hot listing i.e., if any vehicle is caughtunder hit and run case we could apply ALPR and find out the vehicle through surrounding cameras.
2. **Automatic Toll Collection**: Using ALPR we could do automatic toll collection. When vehicle passesthough camera at toll plaza image will be captured automatically and toll receipt will be generated.
3. **Vehicle Over Speeding**.
4. **Parking Slot Management**.

****

**Fig 1: Working Application Prototype**

When LP is recognized an QRCODE will be generated which will be set with some valuable information related to that LP.

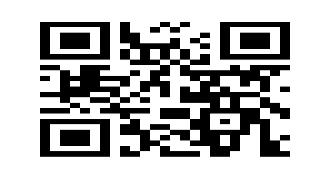
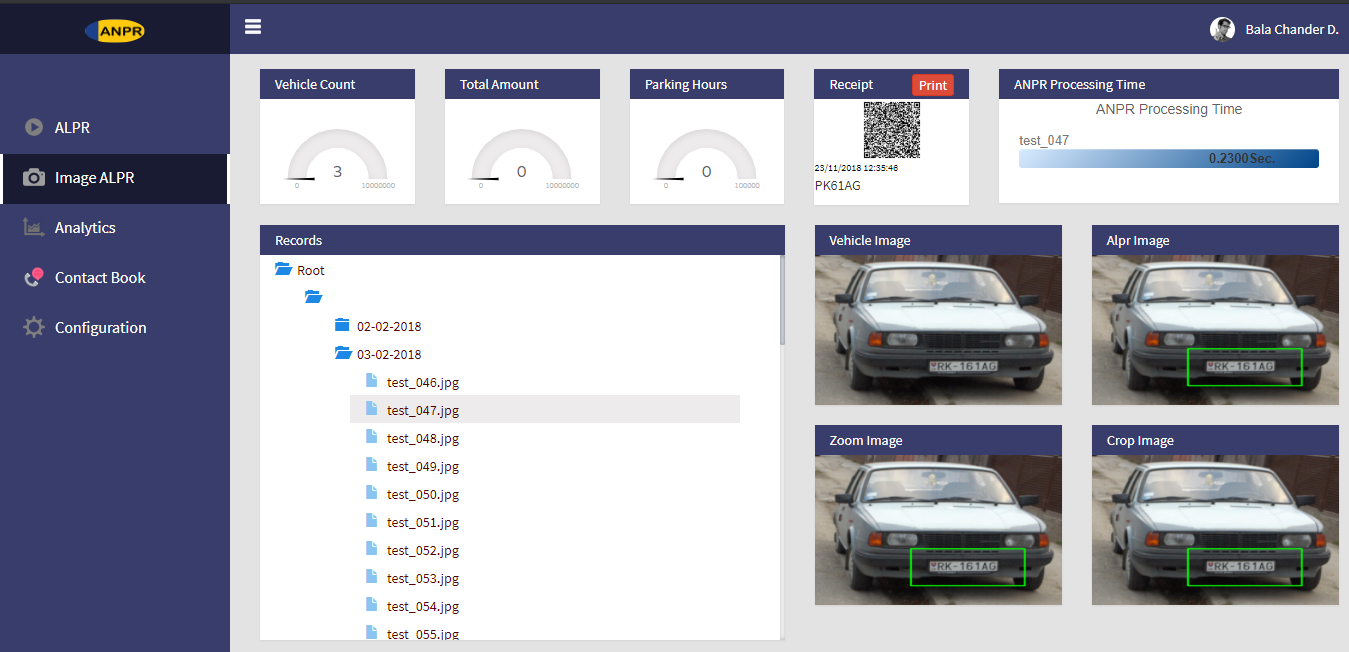


Fig 3. QRCode of LP.

Snippets of the ALPR application which is Developed using Python, flask, Opencv.





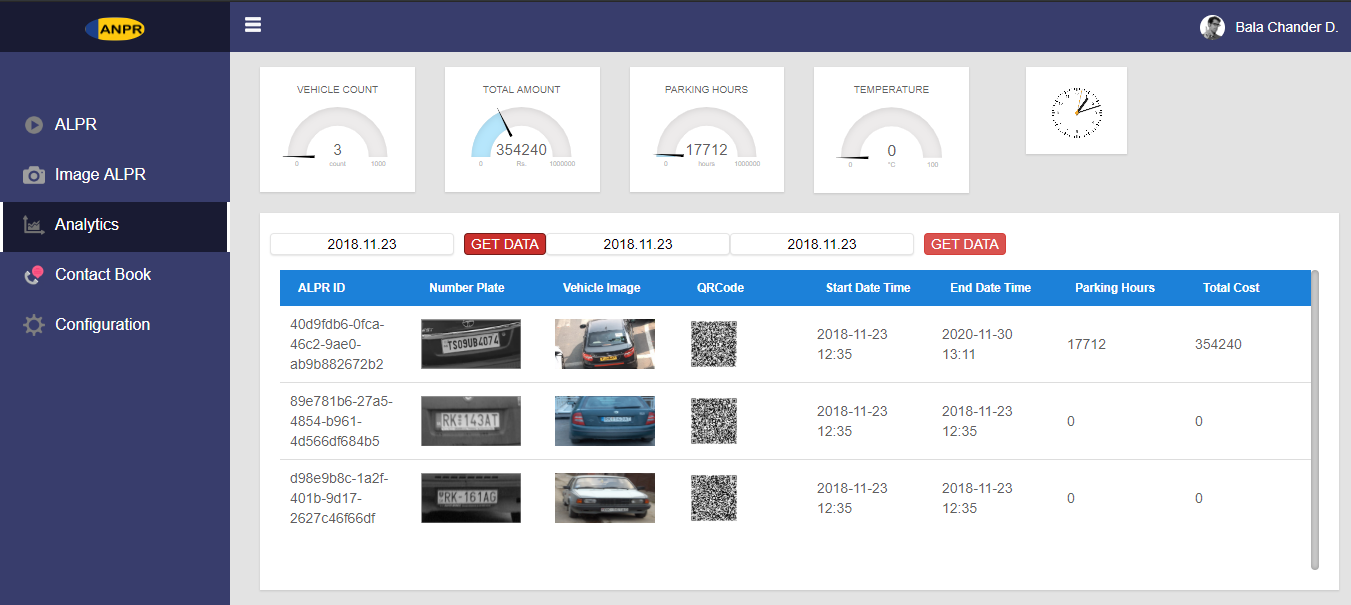
F**ig 2. Snippet illustrating the ALPR application**

While leaving from parking place vehicle owner must submit the receipt at toll collection exit gate.

Where the QRCode will be read from the receipt and will calculate the fare to be Payed. (i.e., difference between start time and end time multiplied with hourly amount).

**Analytics of ALPR:**

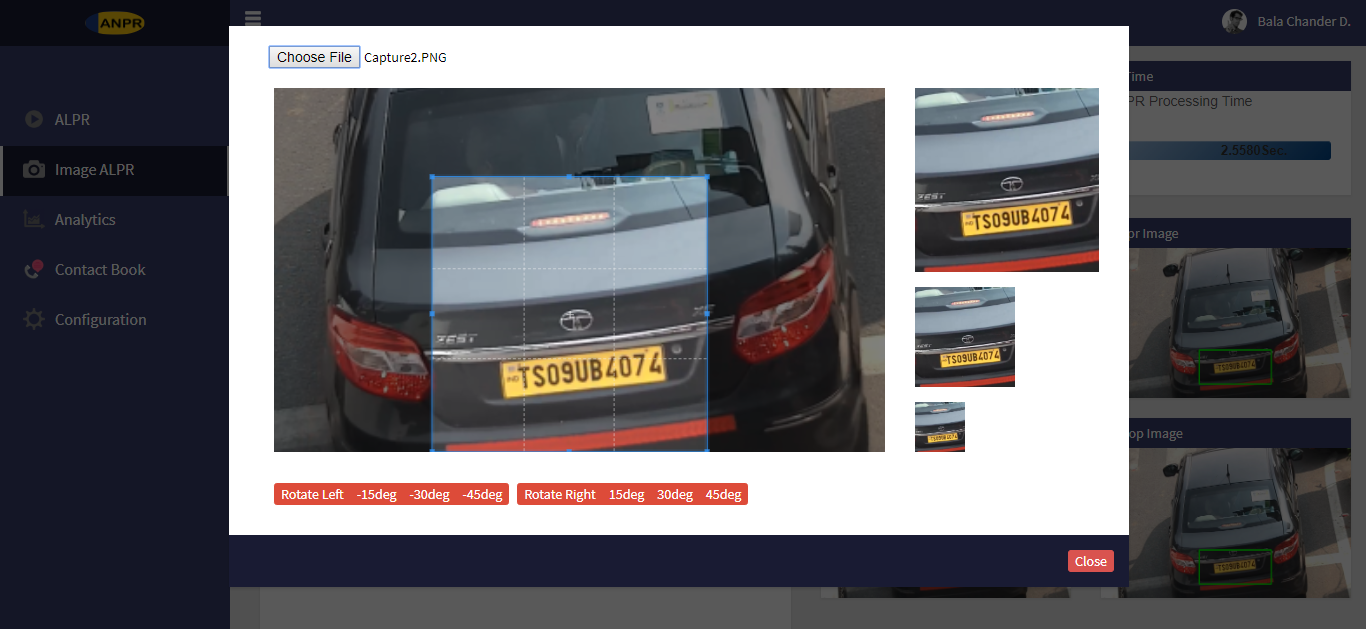
Apart from ALPR functionality we can do some analysis like querying for past data and updating the vehicle exit date time and hours spent in parking spot.

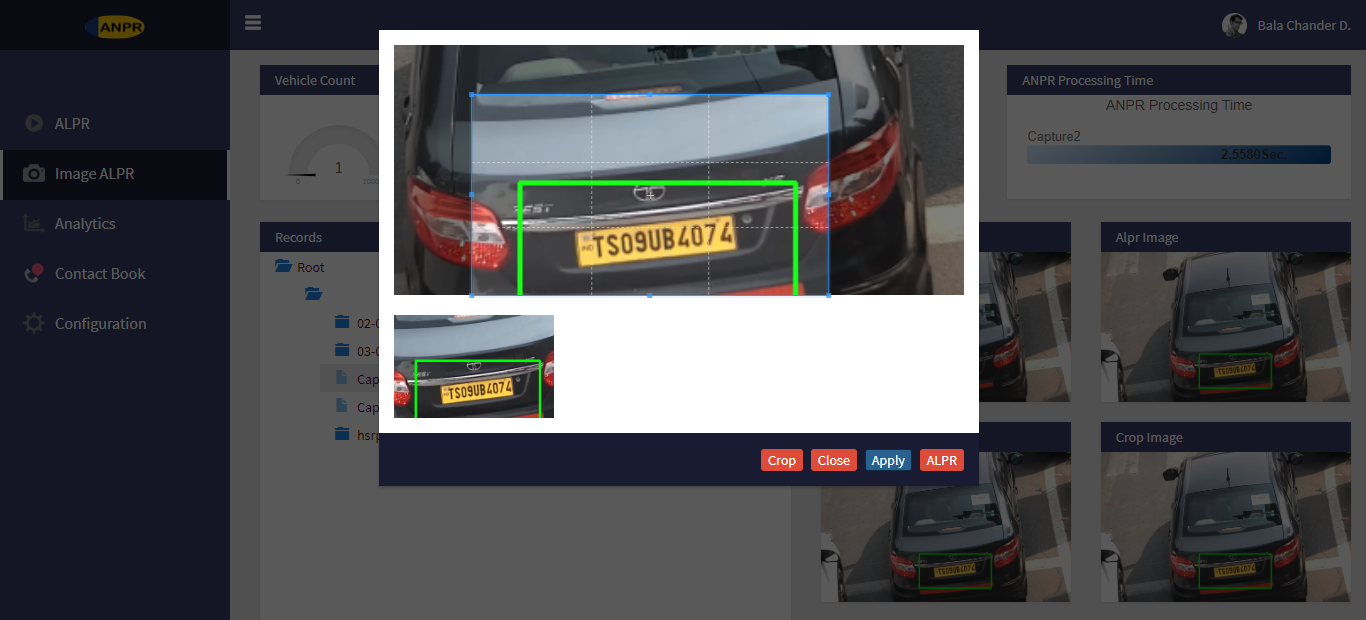


**Fig 4. Analytics on Past data**

****

**Fig 7. Image cropping in ALPR for detailed investigation**





**Fig 6. Deep investigation on vehicle data.**

Video demonstrating ALPR Application.

Link :-

**Tools and Technologies**

**Note:- Tools and Technologies used are open sources.**

* + Python2.7 with all required packages
* OpenCv2.x python build
* Web Cam/Wired/IP Camera (note: for this hackathon I have used pre-recorded video as input)
* RabbitMq
* MySql
* XAMPP server
* OCR – Tesseract (Optical Character Recognition)

**Thank You !!!**