Analysis of OCR based detection of vehicle license plates with implementation on ARM based embedded model

A Prototype model

Ashwini Ashok Patil
Department of Electronics Engineering
Walchand College of Engineering, Sangli, Maharashtra, India
Email - ashwinipatil.1519@gmail.com

Dr. Sachin D. Ruikar
Department of Electronics Engineering
Walchand College of Engineering, Sangli, Maharashtra, India
Email - ruikarsachin@gmail.com

Abhijit S.Shinde
Electronics Engineering dept.
Walchand College of Engineering, Sangli, Maharashtra, India
Email-abhijeetshinde70@gmail.com

***Abstract*— The primary use of detection of Vehicle license plates is for security purpose. It can be used for monitoring systems in high security areas. Most of these systems are implemented with Matlab and image processing tools already. Now-a-days, Python is becoming more popular in image processing. It has its own advantages over variety of fields. This paper proposes a license plate detection system with a new approach, which includes software stack related to Python and hardware based on ARM architecture.**

***Index Terms*— Python, OpenCV, Neural Networks, machine vision, Python APIs, Feature extraction, ARM architecture, Single Board Computer, Tesseract**

I. INTRODUCTION

This system has boon in the need to covert information about vehicles between the actual scenario and the information systems, which can be able to identify vehicles by their license plates in a real time and transform it into conceptual resources. This paper deals with the broad areas in the field of Electronics and programming. The use of domain knowledge from image processing, automation, Python programming, ARM architecture development, machine vision, neural networks, etc. in designing the automatic license plate recognition system. The proposed system will work in real time environment and based on C++ programming covered with Python APIs. It identifies the number plate, under the normal conditions. With the help of database hosted on single board computer, it can find match for it, so that authorization of that number plate can be performed. This system can be installed at a tollgate for recognizing vehicle license plate and based on the result, the gate can be controlled at tollbooth. It then can be implemented in high security areas like military zones, then also at parking areas, bridges, highways, tunnels, etc. This system involves software tool stack which is compatible to Python programming language, as it is growing and challenging field in image processing area. This paper has combined image processing tool stack in Python with the embedded device for creating prototype model for proposed system. This paper uses Raspberry Pi, which uses ARM architecture and it is used as single board computer here.

II. PROPOSED SYSTEM

A. Proposed Methodology

The reference paper [1], implemented by us is based on software aspects for detection of License plate. Now, this paper will tell you more about actual hardware implementation on ARM architecture and related changes adopted in tool stack of software. You can say that this paper is the actual and real time implementation of work proposed by us in paper [1], with developing enhanced Python software. As shown in figure 1, the proposed system contains a single board computer, a web camera and touchscreen display as the main hardware. The workflow for system is capture license plate image from camera, it is then given to embedded device for processing it by tool stack and software burned into that device and then producing the output based on database hosted on that device. The database is used for comparing output from embedded device and this performs authorization of that license plate. For preparing the prototype model, this paper uses a small database stored on ARM board. According to the tool stack of Python in image processing, compatible hardware is selected, by considering the various design considerations.

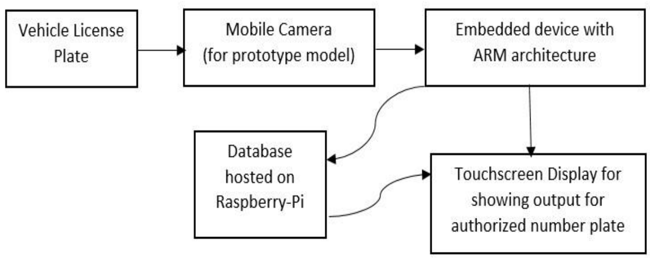

Fig. 1 Block Diagram of proposed application

B. Proposed Algorithm

First step in detection of the license plate area involves 'Area detection'. Once area is found, we proceed to segmentation step. Segmentation step uses binarization, character analysis, plate edge detection and character segmentation. For rotated images, de-skewing algorithm is used. After that, OCR

978-1-5386-4008-1/17/$31.00 ©2017 IEEE

(Optical Character Recognition) phase is there. It involves Feature extraction and classification.

The whole algorithm consists of following technologies:

- Matrix Matching
- Fuzzy Logic
- Feature Extraction
- Structural Analysis
- Neural Networks

III. HARDWARE DESIGN CONSIDERATIONS

Before you begin to format your paper, first write and save the content as a separate text file. Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads—the template will do that for you.

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar.

A. Selection of Single Board Computer (SBC)

SBC should have following features:

- Support for various mainstream operating systems
- Means of input that supports higher frames per second rate for image data
- Sufficient processing power
- Extension capabilities
- Connectivity options for network I/O
- Cost efficient

Comparison among available SBCs:

Board	Price($)	BLE	WIFI	Processor	Architecture Version	Speed	RAM	Flash	SD Card	OS	USB	HDMI	GPU
Beaglebone Blue		Yes	Yes	TI Sitara AM3358	Cortex-A8	1	0.5	4	Yes	Linux	1	Yes	Yes
CHIP	9	Yes	Yes	Allwinner A13	Cortex-A8	1	0.5	4	No	Linux	1	No	Yes
Rapsberry Pi 3	35	Yes	Yes	Broadcom BCM2837	Cortex-A53	1.2	1	0	Yes	Linux/Win	4	Yes	Yes
Beaglebone Black	55	No	No	TI Sitara AM3358	Cortex-A8	1	0.5	4	Yes	Linux	1	Yes	Yes
Intel Edison	50	Yes	Yes	Intel Atom Dual core	x86	0.5	1	4	Yes	Linux/Win	1	No	No

Fig. 2 Comparison among SBCs

Considering above, chosen SBC is - Raspberry Pi 3, with following features:

- Higher Performance due to better processor and ram
- Better support for multiple operating systems
- GPU and DSI port that allows higher frame rates of camera as compared to the standard USB
- Network capabilities, BLE, WIFI, Gigabit Ethernet

B. Selection of Web Camera

Web camera should have following features

- Driver support for Linux and Windows Operating systems
- minimum resolution of 5MP
- Higher frame rate.
- better sensor for images in low light
- compatible with low power configurations, often found in SBC's
- Based on these, chosen camera is - 'Raspi Cam'. It has a good image sensor - Omni vision 5647 CMOS image sensor in a fixed-focus module with integral IR filter.

C. Selection of Touch Screen

After SBC selection, the compatible touch screen is 'Raspberry Pi, 7 inch touchscreen display'.

IV. SOFTWARE DESIGN CONSIDERATIONS

After selecting hardware, the software should have following capabilities:

- run on Linux
- efficient memory processing
- Accuracy should be greater than 70 percent
- should be able to work with images of lower resolution as well as higher resolution
- should be compatible with both arm and x86 architecture

V. BUILDING THE SOFTWARE DESIGN

A. Finding proper tool stack

There are various image processing libraries available in Python. These are – OpenCV, OpenALPR, CNN-ANPR, etc. Based on software considerations, compatible chosen tool stack is OpenALPR.

This paper has used OpenALPR in following manner:

- The following libraries are used along with OpenALPR - OpenCV, Leptonica, Libtesseract
- OpenALPR is used for Area detection and ASCI/Unicode generation.
- OpenCV is used by all the other components like Leptonica and Libtesseract
- Leptonica is used for grayscale conversion, affine transformations and Pixel-wise masking
- Libtesseract is used for segmentation, Feature extraction and classification

The algorithms for each stage in image processing uses combination of Python, C++ and C libraries. Because, this makes the software best in efficient memory handling, garbage collection, performance, speed and accuracy. For using these algorithms, a Python API has developed, to fully access C++ implemented algorithms.

B. Installation Aspects

Flow of installation of chosen tool stack with proper libraries is as follows:

- Installing the dependencies
- Installing OpenCV
- Download and install Leptonica and Tesseract
- Compile Leptonica, Tesseract
- Clone the required OpenALPR repository

Above mentioned each step involves set of Linux commands, which are standard and need to be follow.

VI. SETTING UP HARDWARE

After testing the software code on Linux, let's move towards, implementing it on hardware.

A. Connecting screen to the Pi and Camera set up

- Connect the screen using the DSI port on raspberry
- Connect the ground and 5V power connections.
- Interface Raspi camera to Rpi
- Install corresponding libraries.

For prototype implementation, to reduce cost, this paper uses Webcam application in replacement with Raspi camera. For this,

- Install IP Webcam. Configure system to that IP address.
- Install corresponding libraries.
- Start the broadcast server.

B. Software Setup for raspberry pi

- Install Linux libraries required for our algorithm.
- Cloning them in proper format
- Building Leptonica
- Building tesseract
- Building OpenCV
- Building OpenALPR

C. Final testing on hardware

After this whole setup, this is the phase to build software for ARM architecture. After checking the various requirements, now the entire code is migrated from virtual machine to raspberry pi. For final testing, just open the terminal window on Rpi and run the final script for input images. It shows correct output for standard License plates, under some critical conditions also like low light, rotatory image.

VI. EXPERIMENT AND RESULT

A. Interfacing touch screen display to Raspberry-Pi

Fig 3. Connections for touch screen display

B. Protecting all parts with the help of case for easy and safety handling

Fig 4. Safety case for components

C. Results on hardware model

- Handy Hardware

To compensate the space and make the system handy, power bank is used to supply power to RPi and, whole circuitry is placed at back side of display; so that from front one can see only displayed output, i.e. a very compact hardware.

Fig 5. Handy hardware view from top

- Testing of whether software is correctly installed

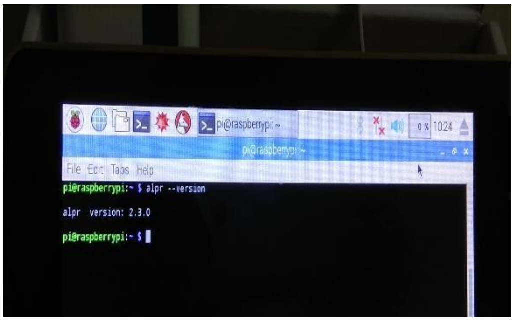

Fig 6. Testing for correctly installed software libraries

As it shows the ALPR version correctly that means it has correctly installed.

- Result of clear image

978-1-5386-4008-1/17/$31.00 ©2017 IEEE

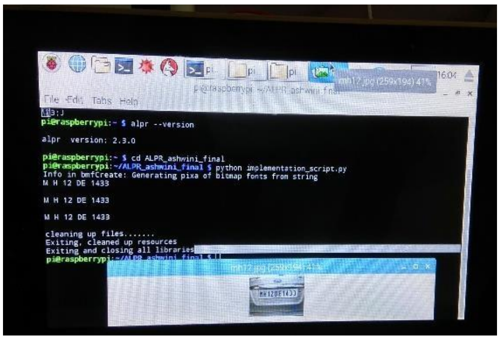

Fig 7. Testing on clear image

- Result of low light image

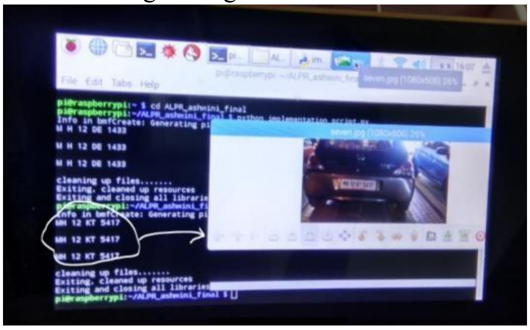

Fig 8. Testing on low-light image

- Result of image with extra flash

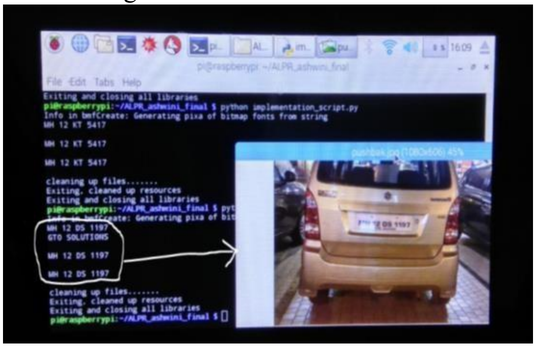

Fig 9. Testing image with extra flash at night time

- Result of blur image:

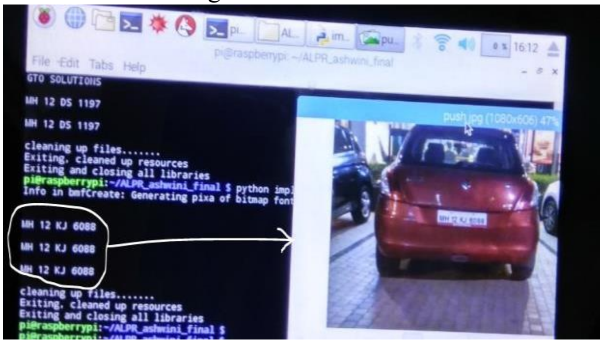

Figure 10. Testing on blur image

VII. CONCLUSION

At first, there were various challenges in algorithm development, e.g. accurate Rectangle detection for license plates, de-skewing of images, support for fonts other than EU and NA, generating dataset for Indian Number plates.

Along with finding proper software libraries and stack, there are various challenges faced while implementing hardware. There are dependencies and differences in ARM architecture as compared to x86, so it was tough to build and compile the software on ARM architecture device, as there is lack of direct executable for ARM based devices. In case of wireless camera, it is working across multiple platforms. And also I have removed the extra hardware needed for webcam by configuring it into your mobile device for demo purpose. The current prototype is a completion of Proof-Of-Concept.

ACKNOWLEDGMENT

I am eternally grateful to my guide, Dr. Sachin D. Ruikar, for presenting me such an interesting thesis topic. He is an expertise in Image processing area and also published his innovative work in various international conferences. Each meeting with him added invaluable aspects to the implementation and broadened my perspective. He has guided me with his invaluable suggestions, lightened up the way in my darkest times and encouraged me a lot. From him I have learned to think critically, to select problems, to solve them and to present their solutions. I am thankful to him for always making time for me through his busy schedule. I feel very proud to work under his guidance. Talking about another author, it is great learning and solving of doubts and helping each other and sharing thoughts is a very great experience. Thanks Abhijit for your help.

REFERENCES

[1] Ashwini Patil, Sachin Ruikar, “Automatic OCR based control and analysis of vehicle license plates”, International Journal of Control Theory and Applications (SCOPUS Indexed, Listed in UGC), volume 10 issue 15.

[2] Pratiksha Jain, Neha Chopra, Vaishali Gupta, “Automatic License Plate Recognition using OpenCV”, International Journal of Computer Applications Technology and Research Volume 3– Issue 12, 756 - 761,
2014, ISSN:- 2319–8656

[3] D.Lavanya, C.V.Keerthi latha, Nirmala,”License Plate Extraction Of Images Using Raspberry Pi”, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 4 Issue 1, January 2015

[4] Kanani, P.; Gupta, A.; Yadav, D.; Bodade, R.; Pachori, R.B.,(April 2013), "Vehicle license plate localization using wavelets", IEEE Conference on Information & Communication Technologies (ICT), 2013, vol., no., pp.1160,1164, 11-12

[5] R.Radha1 and C.P.Sumathi, (August 2012), "A novel approach to extract text from license plate of vehicles", Signal & Image Processing : An International Journal (SIPIJ) Vol.3, No.4

[6] Christos-Nikolaos E. Anagnostopoulos, Member, IEEE, Ioannis E. Anagnostopoulos, Member, IEEE, Ioannis D. Psoroulas, Vassili Loumos, Member, IEEE, and Eleftherios Kayafas, Member, IEEE,” License Plate Recognition From Still Images

and Video Sequences: A Survey", IEEE Transactions On Intelligent Transportation Systems, Vol. 9, No. 3, September 2008

[7] Jaya Gupta, Prof. Supriya Agrawal, "Comparison of two commonly used techniques for Vehicle License Plate Recognition", International Journal of Scientific & Engineering Research, Volume 6, Issue 5, May-2015 739

[8] C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, V. Loumos and E. Kayafas, "A License Plate-Recognition Algorithm for Intelligent Transportation System Applications," in IEEE Transactions on Intelligent Transportation Systems, vol. 7, no. 3, pp. 377-392, Sept. 2006.

[9] Amr Badr, Mohamed M. Abdelwahab, Ahmed M. Thabet, and Ahmed M. Abdelsadek, "Automatic Number Plate Recognition System", Annals of the University of Craiova, Mathematics and Computer Science, Volume 38(1), 2011, Pages 62-71

[10] Ragini Bhat, Bijender Mehandia , "Recognition of vehicle number plate using matlab", International Journal Of Innovative Research In Electrical, Electronics, Instrumentation And Control Engineering Vol. 2, Issue 8, August 2014

[11] Ondrej Martinsky, faculty of information technology department of intelligent systems, "Algorithmic and Mathematical Principles of Automatic Number Plate Recognition Systems", B.Sc. Thesis, BRNO university of technology

[12] K.M.Sajjad , "Automatic License Plate Recognition using Python and OpenCV", Department of Computer Science and Engineering, M.E.S. College of Engineering, Kuttippuram, Kerala

[13] Ian.J.Goodfellow, Yarsolav Bulatov, Julian Ibarz, Sacha Arnoud, Vinay Shet, "Multi-Digit Number recognition from street view imagery using deep convolutional neural networks" , Research at Google, 2014.

[14] Ms. Sonali Sonavane Prof. Alka Khade Prof. V. B. Gaikwad, (August 2013), "Novel Approach for Localization of Indian Car Number Plate Recognition System using Support Vector Machine", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 8

[15] Chirag Patel, Atul Patel, Dipti Shah,(2015), "A Novel Approach for Detecting Number Plate Based on Overlapping Window and Region Clustering for Indian Conditions", IJIGSP, vol.7, no.5, pp.58-65