“Sorting the wood from the trees”

Automatic License Plate Recognition System Based On MATLAB

This report is submitted for the degree of Bsc Computer Science

Third Year Project Report

Author

Weibin Wu

April 2016

Supervised by

Carole Twining
ABSTRACT

This report describes the development of a MATLAB system to perform Automatic Number plate Recognition (ANPR) on real-world, colour still images of vehicles. The final algorithm consists of various computer vision stages (detection of the plate position, adjustment of the plate viewing angle, segmentation of the plate into individual characters), as well as simple Machine Learning (optical character recognition for the standard UK number plate font). The system is tested on a database of UK (rear) number plates, and various failure modes investigated. The robustness of the system under various deformations, and to the addition of image noise, is also investigated.
Acknowledgment

First of all, I would like to thank Dr. Carole Twining for giving me guidance during this year. She is an excellent and patient supervisor. Her sage advice was priceless and her enthusiasm was always keeping me going forward.

I was also like thank my parents for supporting me during the tough times and paying the license of MATLAB for me. A big thank to my flat mates, Joey and Wilson, for staying quiet when I am working on my project. Another thank to my course mates, Wilson Wong and Jinbin Liang, for giving me opinions on various aspects of the project, and helping me to prepare for the seminar and demonstration.

At last, I would like to thank my first year and second year tutor DR. Kung-kiu Lao for giving help and advice to me and all the staff members including lectures, teaching assistants and Student Support Office staff.
Contents

1.1 Vehicle License Plate Recognition System .......................................................... 5

1.2 License Plate and Its Law In the UK ..................................................................... 6

1.3 Image Processing Background ............................................................................. 7
   1.31 HSV ................................................................................................................. 7
   1.32 Radon Transform .............................................................................................. 9
   1.33 Morphological Operation ............................................................................... 10

2.1 The Initial Idea ...................................................................................................... 12

2.2 The Ideal System ................................................................................................. 13

3.1 Programing language .......................................................................................... 14

3.2 Plate Detection .................................................................................................... 15
   3.21 Approximate Approach .................................................................................. 16
   3.22 Further Approach .......................................................................................... 17
   3.23 Cropping the Plate ......................................................................................... 18

3.3 Character Segmentation ....................................................................................... 20

3.4 Character Recognition ........................................................................................ 22

3.5 GUI ....................................................................................................................... 22

3.6 Database .............................................................................................................. 23

4.1 Accuracy ............................................................................................................... 26

4.2 Angle Correction ................................................................................................. 27

4.3 "Salt and Pepper" ............................................................................................. 27

4.4 Compression ........................................................................................................ 29

4.5 Foreign Plates ..................................................................................................... 31

5.1 Key Achievements .............................................................................................. 34
5.2 The Learning Experience ................................................................. 35
Introduction

In this chapter, there are some background information will be talking about and the reasons why I aim to achieve this project and approach I choose.

1.1 Vehicle License Plate Recognition System

With the rapid development of computer technology, the research of automatic processing of information is active and its applications are widely used in many different areas.[1] And the usage of the vehicles is getting more and more essential to people and it is an important research topic in information processing technology about how to improve the traditional traffic manage mode using information auto-collection, dynamic supervise manage and assistant decision-making, etc. [2]

More and more technology were applied for the solving the problem of transportation of city whose compose the Intelligent Transportation System which not only strengthens the bond among vehicles, road and traffic controllers to enhance traffic safety and improve transport efficiency, but also help to achieve automatic road enforcement to reduce environmental pollution, save energy and improve the vigor of the economy.

As an important resource of the vehicles’ information for the Intelligent Transportation System, License Plate Recognition technology is one of the essential topics to improve the Intelligent Transportation System. License Plate Recognition is a technology that be able to read the information provided by the vehicle registration plates by using optical character recognition on images taken from the CCTV or road-rule enforcement cameras. [3]
Starting in the 90s, the first license plate recognition system was introduced at the Police scientific development branch in the UK in 1976. [3] And today license plate recognition technology is common to use in more than 40 countries around the world. However, although there are many methods and algorithms invented to achieve like using Fuzzy Mathematics Theory and the algorithm of neural network to identify the license plate characters. Difficulty are still brought by the change of lighting condition, weather condition and the mud on the plates. [3] And most of methods are concerned based on the morphology. In my opinion, the colour of Automatic plates should be put to excellent use. Thus, the colour approach to read the automatic license plates will be used in this project.

1.2 License Plate and Its Law In the UK

In the UK, there is legislation to ensure all the British license plate are follow the guideline given by the government. The colour and size of the license plate and the format and number of the characters are being used in accordance with The Road Vehicles Regulation 2001. [4]

There are some rules for the license plates are sent out in legislation and are be in used in the project:

- A license plate is yellow at rear while the one is white at the front.
- All character should be written in ‘Charles Wright’ front.
- There are seven characters displayed on a standard plate: the first two are the DVLA memory tag, the next two indicate the age of the vehicle and there are three random letters at the end
- The letter I is not used on the license plate.
In fig, it is example of the UK’s rear automatic license plates. In this project, the software will focus on the rear automatic license plates to process the recognition. In most cases, the rear plates will be shoot to provide the need information due to many reasons. The most important one is that the flash of camera will not blind the drivers when taking the photo from behind which will ensure the road safety. Secondly, the camera will only be triggered when the vehicles are detected by the other equipment. So the camera can only take the photograph from behind. [7] And last only the rear license plate can be easily shoot by the camera setting on the patrol cars when police cars find some suspicious crimes. Above all, the rear automatic license plates will provide the most well researched information to this project.

1.3 Image Processing Background

In this chapter, it will introduce some background about image processing that are essential to understand how the system finds and recognizes a number plate.

1.3.1 HSV

Pixel is the important element to contribute a digital image, and is being coordinated with two channels which represents the x and y pair in the Cartesian coordinates. Mostly, an image is represented by a matrix with more than two dimension. There will be a third dimension that
use to represent the colour of the pixel. And there are usually three channels used to interpret the colour space. RGB (Red, Green and Blue) colour space is the most common and well-known colour space in most colour images as it is widely used in various areas like digital media and internet web. For more details, there are three channels which are red, green and blue channels. Using the combination of these three channel will contribute a specific coloured pixel. [5]

In addition to RGB, HSV (Hue, Saturation and Value) colour space is also other colour space that be able to use in other applications. HSV is a colour space that designed for the computer graphics and is widely used for computer vision applications today. HSV colour space parameterises colours based on their Hue, Saturation and Value. A cylinder shape is designed to represent the HSV colour space. HSV colour space use a circle to represent the colour because the colours are created by mixing the pure colours in HSV colour space, thus the circle can display the small change of colour while mixing. The Hue is the angle around the cylinder’s central vertical axis with the range $0 < x < 360$ and it is the most important measurement as it contributes the shape of the colour. Saturation is the distance from the cylinder’s central vertical axis which denote the intensity of the colour and Value is the distance from the button of the cylinder which represents the brightness of the colour in relation to its saturation. The cylinder can be seen in Figure below.

![Figure 2: HSV colour space cylinder](image)
1.32 Radon Transform

The Radon transform was firstly introduced in 1917 by Johann Radon, who also is the provider of the formula for the inverse transform. [9] In two dimensions, radon transform generally can be understood as: along with different straight lines in a plane (the distance between the origin and the line is \(d\) and the direction angle is \(\alpha\)) for \(f(x, y)\) to do the line integral, the image \(F(d, \alpha)\) is the Radon transform of the function \(f\). In other words, each point of the function \((d, \alpha)\) corresponding to the integrated value of the original function.

More generally speaking, assume your finger is transmitted by a strong parallel light, the image you see facing the light source is 3-D Radon transform's value of the attenuation coefficients when given certain location.

Radon transform is defined mathematically using delta function as: [10]

\[
R(r, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - r) \, dx \, dy
\]

*Where theta is the angle of the line, and r is the perpendicular offset of the line.*

![Figure 3: Radon transform](image)
1.33 Morphological Operation

Morphological operation is a set of non-linear operations which are based on the morphology of features in the image. The relative ordering of pixel will be used in the morphological operations which make it suitable to process the binary images. All the morphological operations are based on two operations: erosion and dilation. And morphological techniques need the structuring element as the shape template to execute the operations. The structuring element will process though all the pixels in the image and it is compared with the neighboring pixels to find out whether it intersects the neighboring pixels or it ‘hits’ the neighboring pixels.

![Figure 4: Morphological structuring element[11]](image)

The different combination of these two operation will bring the different result to the image. By using erosion with small structuring elements will shrink the image by removing the pixels from near the boundaries of regions. The small details can be removed and the gaps between different regions can be larger. On the other hand, it can expands the objects in the
same image by using dilation.

**Figure 4: Erosion and Dilation**[11]
Design

2.1 The Initial Idea

In the chapter, it will introduce about the designing process of the project. This initial idea about the software was to process a set of vehicles images. And this images will be processed by the software to read their automatic license plate. Then the final result will be output and display on the GUI.

However, in the final case, the software will only process one single image and show the process of data processing because it is trivial and inefficiency to collect and process a set of source data into a single result. In addition to that, the image analysis and process steps on the single image should be the major challenge of this project. The design of the software is a pipeline model dividing into three main modules: Plate Detection, Character Segmentation and Character Recognition.

![Diagram](image)

Figure 7: different functional part of the software
2.2 The Ideal System

The ideal software would take a vehicle image as the input. The image is a colour image that contains the vehicle with the license plate, various pedestrian, and some other traffic signs, plus the background of the road. Next it should extract the colour-relevant part from the source image by using the appropriate colour space model like HSV or RGB. The extracting part is an approximate part that contains the license plate and some other interfering signal. Thus the software will enhance the image and using the edge detection to narrow the field. The resulting image that should be the license plate image would be sent to the next module. In addition, the software will also adjust the angle of the photograph to make the license plates are parallel to the horizontal.

In the Character Segmentation module, the software will looking the gap between each character and split the license plate image into several character image. In this step, the license plate image will be converted to binary image. Thus, there will be a thresholding that clearly distinguish the character pixel and the background pixel. The well completeness of the single character is the precondition for the software to execute next step.

In the final step, each single character image will be processed by the Character Recognition module where it will compare the character image with every templates in the database. And the template with the smallest difference will be the final result. The output will be a string of text showing what number is on the license plate in the vehicle image.
Implementation

In this chapter, it will details how the implementation was carried out and some of the
decisions made during this project. It will also talk about the choices between different
algorithm and method and choice of the language used. And this chapter also describes about
the graphical user interface created to show more details about how the result was made and
how the image was process for the users.

3.1 Programming language

As this is an image processing software, there are few choices that allow me to implement the
software. There were two languages that I have used before for other programing project and
I think they are suitable for this image processing project which are C++ and MATLAB.

C++ can call the OpenCV which is a library of computer graphical functions and it is licensed
under an open source library. So it is free to use. And OpenCV can process the image much
faster than MATLAB as OpenCV is a library of function written in C/C++ While MATLAB
is built on Java which is built upon C. Thus, when the code of MATLAB is being executed, it
need to interpret all the code and convert them into Java and then finally execute the code. [12]
Also OpenCV require less system resources needed than MATLAB.

However, MATLAB have a lower learning curve compared to OpenCV. As OpenCV is based
on C so it will require more in memory management or it will easily introduce bugs and due
to the high-level nature of MATLAB, it will automatically manage the memory in the
background to avoid bugs. In addition, there are many high-level function provided by
MATLAB and all of them are well documented. By using the help command will provide
many examples of the code and tutorials of how to use with this code. Furthermore the ‘break point’ function provided in MATLAB can help to test the code parts by parts, and the debugger is quite efficient.

Above all, although OpenCV have a better performance compared with MATLAB in execution of the code. I still think MATLAB is fast enough for this project and it provides an interactive debugging environment that OpenCV cannot provide. In the end, MATLAB is chosen to use in this project for these reason.

3.2 Plate Detection

In this module, the software will process the vehicle image input and output the plate license image to the next module. There are many methods that can achieve this goal and most of them are using edge detection to locate the rectangle with the similar shape compared with the license plate. However the disadvantage of using edge detection is when there are many interfering pixels existing in the image which will cause the software to locate the incorrect field. This is shown in Figure ??, where a greyscale number plate image is shown, along with the edges detected in this image. Thus, it will need a high requirement of the source image to produce correct result. And as the development of the technology and industry, it cost less and less to replace the grayscale camera with the colour camera which allows to use more information to locate the license plate. And the back license plates of the vehicles are reserved to be yellow. Therefore, colour-extraction will be applied to locate the field of licence plate in the source image.
3.21 Approximate Approach

At the beginning, I used the RGB colour space to pick up the pixels that related to yellow. However, it will still keep most of the unwanted pixels or lose some license plate pixels and cannot gives a good enough result image that ensure to locate the correct field of license plate. Thus, I use the HSV colour space as the model to pick up the pixels and it is giving a better result compared with RGB colour space. To compared with RGB colour space, HSV colour space using hue, Saturation and Value to represent the colour of the pixel which is more straight and clear for computer to identify. In addition, the intensity of light like strong light and shadow will easily affect the RGB colour space that confuses the computer to remove the right pixels and pick the wrong ones while the HSV colour space can handle these light really well to stay what should be.
3.22 Further Approach

However, there are still some unwanted yellow-related pixels existing in the image, so there is more step the software will process to select the right field of the license plate.

For this part of function, there are some method will be used to enhance. Firstly, the median filter is used to eliminate the noise on the photograph. For example, there are some dispersed yellow relevant pixels will displayed on the photograph, and the software need to process the median filter to improve the result of later processing[13]. Then, the software will also process morphological open and close operations remove the unwanted parts from the
photograph. The morphological close operation will reduce the gap between the pixels to group them together, the operation is done to keep license plate’s completeness. Likewise, the morphological open operation will eliminate the small regions from the photograph.

3.23 Cropping the Plate

The next step of Plate detection is cropping the plate license image from the source vehicle image. This can be achieved using counting the sum of white pixels (the license plate) by each row and column. In the vertical direction, there is a counter that stores the number of white pixels of each row. Start with the one with the highest value, there are two tags will go upward and downward respectively until there is no white pixels existing in this row. And the values of tags recording are the top and bottom of the field of the license plate. In the same method, the left and right boundary of the license plate can be found for software. Using these values to estimate a rectangle to chop the license plate from the image and the values will be stored in the array for further processing to remove the white frame of the license plate which might confuse the computer to execute the segmentation of characters.
Now, the software has already find the approximate position of the license plate. In some cases, the image are not parallel to the horizontal and need a process to adjust the angle. By using the Radon transform, the sum of the Radon transforms of each pixels will be used to provide the angle that we need to adjust. To compare with the Hough transform, Radon transform is based on the pixels while Hough transform is based on lines. When using Hough transform, the longest line will be assumed to be the scale line to adjust the angle of image. However, if the image with low resolution will not always gives a clear line so the Hough transform is not suitable to be applied here.
3.3 Character Segmentation

In this part, the software will converted the colour license plate image into binary image and applied the histogram for the image to obtain isolated character images.

To implement the function, it is essential to find the best threshold to distinguish the character pixels and the background pixels. Because of the different lighting condition, the static thresholding will faired very well to all the images. Thus the software should have an intelligent method to obtain the best threshold for different images.

Otsu’s method were applied in this part to cover this problem. Otsu’s method is used to compute the threshold that will give the minimum variance of the black and white pixels in the photograph. And this method is already being set in the MATLAB library, so the function ‘graythresh’ is being executed to create the best threshold for individual image.[15]

Now each image will be convert into binary image with its own thresholding, the next step is involved in separating the license plate image into several images that only contains one isolated character. After removing the white frame of the license plate, only the character pixels, background pixels and some noisy pixels leaving in the image. Then we need to count the number of the character pixels of each column and create a histogram.
From the histogram, the peak on the graph means that there is character here and the places without the character pixels are background pixels where should be the gaps between the character. In addition to that, where there is a noisy pixels appearing in the gaps, the software has a threshold that could help to identify whether gap or character is. After that, the values given by the gaps are used to be coordinates to separate the images.
3.4 Character Recognition

As the software already got the isolated image of the characters, the final step is to compare the character image with the template images in the database. The method used in this function to do the optical character recognition is template recognition. As there are standard templates for all the characters and numbers in the building database, the image of the characters obtained by the previous process will send to compare with every template images and finding the differences between them using image subtraction method. The template image with the smallest difference is decide to be the result of the image. After all the character images have been process, the software will output estimated vehicle’s license plate number.

3.5 GUI

A Well-designed user interface is required to observe images being process by the software. In figure 15, there are a bunch of buttons represent different functions so that the users are able to process the image step by step and watch the operations done to the images. At the start, only one image will be shown on the user interface and it will keep changing by executing the operations. However, it will be better to show the entire process to users to observe the change of images. In addition, a button was set to execute all the operations and display the result on the user interface.

Instead of using multiple windows to display all the images, user would like to have a one-piece window that contains multiple images. The design is easy to use by clicking the buttons on the left and seeing the result on the right. A good user interface will not only help to improve the user experience but also help to improve my efficiency for experiment.
3.6 Database

Two database were built in this project:
One is the templates database which have 36 template images. There are 25 letters (A to Z without I) and 10 numbers (0 to 9) and a blank image to deal with the license plate have less than 7 characters. The template image has same as the final result image processed by the software, so it will leave two rows and two columns in white at the boundaries of the image. To ensure the consistency between the template image and character image is the precondition to provide the higher accuracy of reading the license plate number from the image.

The other one is the database with 50 testing vehicle images, which are used to test the accuracy provide by the software. There are different types of cars with British rear license plate and some of them are tilted so it can also test the angle correction function given by the
software. Some of the images are my friends’ own cars and provided by my friends and rest of them are provided by the others.

Figure 16: database of the test examples
Testing and Problems

In this chapter, there are some testing I have done to the software will covered and also some problems I found existing in the software I found by now.

4.1 Accuracy

To run out all vehicles image in the database, there are 39 images are able to give the correct result. And all the images are readable by the human eyes, so the accuracy of the software is:

\[
\frac{39}{50} = 78\%
\]

It is a little lower than average level at 80%, however, all these images are taken by the different cameras with different quality of resolution. It is difficult to adjust the parameter to fit with all the conditions, and I believe that if all the images are taken in the same location with the same camera, the accuracy will significantly improve.

Two main issues are affecting the software accuracy:

The first one is that once there is some yellow objectives appearing in the image, it will highly causing the computer to locate the wrong field of the license plate. The problem might be solved by adding more steps in the first module for location like edge detection and selecting license plates with features (size, ratio).

And the second one is characters matching problem, although the software can process the previous steps well. But in some cases, it will still difficulty recognize some characters like ‘S’ and ‘5’, ‘B’ and ‘8’ and ‘A’ and ‘4’. It should be fixed by adding more template images into the database. It the mean time it will also reduce the process speed.
4.2 Angle Correction

By using the radon transform, it provides an excellent angle correction function for the software. I took five images to make an experiment by changing their angles and see where the software limitation is.

![Figure 18: example of the tilted vehicles](image)

The results showing that even though the angle changing is 80 degree, the image will still be able to convert back that keep the license plate parallel to the ground. This function provide by the radon transform will ensure the characters on the license plate are vertical to the ground. When the character are not vertical may cause the characters overlap together on the histogram then the software will consider these two characters as a single character.

From this experiment, it inspires me that not only the angle correction should be applied but also the perspective transformation should considered to add in the software to make the front side of the license plate is facing toward the user.

4.3 "Salt and Pepper"

In this experiment, the quality of the images was being reduced by adding amount of noisy points. In MATLAB, Using a method called ‘imnoise’ given by the library of MATLAB can achieve this:
J = imnoise(I, 'salt & pepper', D)

Where adding black and white noise to the image I with the density given by D (If D is 0.05 and number of pixels in images is 100, then there are 5 noisy point adding.)

And the functionalities provided by the software are: Plate detection, Character segmentation and Character recognition. The density of the noisy points will be recorded when it cause the failure of the functionalities.

![Origin image](image1)

Origin image 0.05

![Figure 19: images that adding different level of noisy points](image2)

0.17 0.20

Figure 19: images that adding different level of noisy points
When D is 0.05, the software will give a correct result until D reaches 0.17, the software will fail to separate the character but can still locate the license plate. Last, when the D is over than 0.20 the software will fail to locate the field of the license plate.

![Figure 20: result of the test, record the value of D when it cannot implement the functionalities](image.png)

In addition to that, there are other images were processed by this experiment. And they also give a similar result with this example. And I found that the performance of image with high resolution is better to deal with the noisy point. From this experiment, the software has a strong noise resistance to locate the license plates as the algorithm using in the software is based on colour but not edges and lines. I believe that the software can handle the poor weather condition well like foggy and rainy weather. The problem

### 4.4 Compression

In this experiment, the images will be compressed in vertical and horizontal level.
From the result of this experiment, the compression in the vertical direction did not affect the functionalities of software. And as in the Character Recognition module, the software will resize the image into the size of the template image. No matter how the compression be, it will not confuse the recognition function.

However, the compression in the horizontal direction did reduce the performance of the software a lot. Because the loss of pixels during compression will cause the gaps between character hard to be found by the software and the noisy points laying on the license plate cause a negative impact to the software.
4.5 Foreign Plates

Although this project is designed based on the UK Automatic license plate, the experiment is going to apply some other license plates from other countries. Before the experiment, the parameters of the software will be changed to adapt other colour license plates.

![Image of China Automatic license plate processing](image)

*Figure 22: trial of China Automatic license plate*

This is an example when the software is processing the China Automatic license plate. From the images above, the software has a sensitive ability to locate the correct field of the license plate. However, due to the silver frame around the license plate the gaps were not able to be detected. And the first Chinese letter was lost after the morphological operations. Thus, we can see that difference styles ruled by the different laws will reduce software performance. However, the detection based on colour is also workable in other countries.
There is another sample from Israel. In this case, Israeli license plates are familiar with those in the UK. The differences between them are the character format and there are ‘-’ symbol on the license plate to separate the characters.

From the result, the software gave an incorrect result with two problems. The first one is due to the extra dash symbols, the software consider it is a character ‘1’ as there is no template image of dash symbol in the database. And because the limit of number on license plates is seven which is being set by the government. Thus the system ignored the two last character in this case. In addition to that, most of the character being read successfully are recognized to give the correct result. However the formats of some characters will be huge different in some countries.

Above all, the software can process some odd photographs like being compressed, fuzzy and oblique. The colour approach method can locate the field of license plate better than the edge
detection method. However, to process all the steps and give the correct result, it will require the image has enough pixels (high resolution) to being recognized.
Conclusion

In conclusion, the project was successful with achieving most important requirements and goals to solve the problem although it was not completely finished with all the predefined functions at the start. In this chapter, some key achievements and improvements of the project and the gain from the project will be mentioned.

5.1 Key Achievements

Most of the important objectives and requirement has been successful achieved to solve the problem. These were to develop a software that would:

- Be able to read the license plate number from the image.
- Be able to achieve the first objective under different lighting condition
- Be able to achieve the first objective with different angles
- Be able to achieve the first objective with a low quality of image

Given more time, the project could go further and would improve a lot. There are some ideal objectives that could be achieved or improved:

- When there are more cars in the single image, the software should be able to label the different car and read their license plates number respectively.
- Problem about reading the yellow car’s license plate should be fixed by using edge detection after the colour extraction.
If using some neural net algorithm instead of template recognition, maybe a different implementation would give better result.

The software could use the video as the input instead of image.

Be able to read the license plate with two lines in other nations (splitting two lines using the same method as separate the character)

5.2 The Learning Experience

As it is the first time for me to work on a project with this scale. It was very challenging to work out where to start and how to design and implement it. The project was an opportunity to learn about how to make the time management during a big scale project and also provide the experience of searching and learning the knowledge about the project and practice the image process technique in the practical work.

The computer can do so many thing very efficiency for human being. However that’s not mean that computer is clever enough to do all the things easily. Like in this project, reading a number plate is an easy thing for human. But all the information have to be converted to a font that is readable for computer is quite a hard task. I have to think in the computer way about what data I should need to do something. And that’s that fantastic thing that learning about computer science. And the image processing magic also attract me so much.

In conclusion, it was great challenging and rewarding experience to work on this project. And I have really improve myself and enjoyed the project in my final year.
Reference


