

HO CHI MINH UNIVERSITY OF TECHNOLOGY AND EDUCATION



FINAL PROJECT

Project 3

License Plate Recognition

Lecturer: Huynh Xuan Phung

Members: Nguyen Van Dan 17110010

Doan Viet Thuan 17110080

Nguyen Tan Phat 17110059

Ho Chi Minh City, 1/2021

Score

Criteria	Content	Presentation	Overall
Score			

Observations of Instructor

.....
.....
.....
.....

Evaluation:

Nguyen Van Dan:.....

Doan Viet Thuan:.....

Nguyen Tan Phat:.....

Lecturer

Huynh Xuan Phung

Preface

Along with the development of science and technology and the increasing demand for travel, the number of vehicles on the road appears more and more. With a large number of vehicles, and continuously increasing like that, there are many problems in controlling and managing vehicles. To solve this problem, the need is to adopt automated systems.

Because the main purpose of this study is to learn and build a system "Identify license plates" from images. In order to help detect traffic violations, prevent theft, manage... more easily and quickly. The following are applications of the license plate identification system in general: Traffic toll collection, Vehicle control at borderlines, gate guard stations, anti-theft, automatic parking, etc.

From those facts, while working on Project3 subject, I decided to choose to learn about the topic: "Research to detect license plates on images". To serve the first step in the transport management system in Vietnam. In terms of capacity and limited time, so your report cannot avoid errors. I sincerely hope teachers and friends to contribute ideas so that I can correct and supplement the shortcomings for me to complete my topic.

Content

<i>CHAPTER 01: OVERVIEW</i>	8
<i>1.1 Feature and necessity of identity problems</i>	8
<i>1.1.1 Feature</i>	8
<i>1.1.2 Necessary</i>	8
<i>1.2 Model and feature of the identification process</i>	8
<i>1.2.1 Model</i>	8
<i>1.2.1.1 Parametric model</i>	9
<i>1.2.1.2 Structure model</i>	9
<i>1.2.2 Feature</i>	9
<i>1.2.2.1 Supervised learning</i>	10
<i>1.2.2.2 Non supervised learning</i>	10
 <i>Chapter 2: THEORY</i>	 11
<i>2.1 Image processing</i>	11
<i>2.1.1 Image processing</i>	11
<i>2.1.2 Image representation</i>	11
<i>2.2 ASP.NET MVC3 technology</i>	12
<i>2.2.1 Overview of ASP.NET MVC</i>	12
<i>2.3 OCR technology</i>	13
<i>2.3.1 Concept of OCR</i>	13
<i>2.4 Tesseract Libraries</i>	14
<i>2.4.1 Tesseract Architecture</i>	14
<i>2.5 Summary of how the Tesseract works</i>	15

2.6 Some basic functions in OpenCV library.....	16
2.6.1 Load an image and display.....	16
2.6.1.1 Download an image.....	16
2.6.1.2 Display images.....	16
2.6.2 Convert color image to grayscale image.....	16
2.6.3 Find the object using findContours.....	17
 Chapter 3 IDENTIFICATION OF VEHICLE NUMBER PROBLEM.....	19
3.1 Definition of license plate identification.....	19
3.1.1 Definition	19
3.1.2 Application.....	20
3.2 How to solve the license plate identification problem.....	21
3.2.1 Regional development approach.....	21
3.2.2 Hough edge detection and transformation approach.....	21
3.3 Detecting the number plate container.....	24
3.3.1 Steps to detect vehicle license plate.....	24
 Chapter 4: BUILDING A PROGRAM TO RECOGNIZE VEHICLE NUMBER PLATE.....	28
4.1 Selection of tools to build.....	28
4.1.1 Microsoft Visual Studio 2019.....	28
4.1.2 Microsoft .Net framework.....	28
4.2 License plate identification program.....	29
4.2.1 The main interface of the software.....	29

<i>4.2.2 Steps to identify license plate</i>	<i>30</i>
<i>4.2.3 Pros and cons.....</i>	<i>32</i>
<i>Preference.....</i>	<i>33</i>

Figure

<i>Figure 1: Identity system.....</i>	<i>10</i>
<i>Figure 2: Image processing system.....</i>	<i>11</i>
<i>Figure 3: Model - Controller - View Architecture.....</i>	<i>12</i>
<i>Figure 4: Tesseract Architecture.....</i>	<i>14</i>
<i>Figure 5: Convert color image to grayscale image.....</i>	<i>17</i>
<i>Figure 6: Using findContours find object.....</i>	<i>18</i>
<i>Figure 7: View area.....</i>	<i>20</i>
<i>Figure 8: Steps to detect license plate container.....</i>	<i>24</i>
<i>Figure 9: Transform the gray image and filter the image.....</i>	<i>25</i>
<i>Figure 10: 8-bit image.....</i>	<i>26</i>
<i>Figure 11 Find outlines of objects.....</i>	<i>26</i>
<i>Figure 12: The character separation stages.....</i>	<i>27</i>
<i>Figure 13: The interface of license plate recognition software.....</i>	<i>29</i>
<i>Figure 14: Identification of license plate.....</i>	<i>29</i>
<i>Figure 15: Discover of license plate holder.....</i>	<i>30</i>
<i>Figure 16: Identification of character container.....</i>	<i>30</i>
<i>Figure 17: Image processing.....</i>	<i>31</i>
<i>Figure 18: Identification of characters.....</i>	<i>31</i>
<i>Figure 19: Identification results.....</i>	<i>32</i>

CHAPTER 01: OVERVIEW

1.1 Feature and necessity of identity problems

1.1.1 Feature

- Identification is the process of classifying the objects represented by a certain model and assigning them to a class (assigning the object a name) based on the rules and standard patterns. The process of identifying based on known learning patterns is called supervised learning; in the opposite case, it is called non-supervised learning.
- Identification is an important problem in machine learning subject.

1.1.2 Necessary

- Along with the ongoing development of socio-economic and technical branches. Requires precise management and processing of information that is beyond human power. So we need machines or dynamics to reduce the load or replace heavy, precise, boring human work. Helping machines to identify (collect, classify information) as humans will help machines operate effectively like humans with much higher accuracy.
- Some applications of identity problems.
 - Fingerprint recognition
 - Voice recognition
 - License plate recognition

1.2 Model and feature of the identification process

1.2.1 Model

The choice of an identification process is closely related to the type of description that people use to specify the object. In identity, people are divided into two class families:

- They describe according to parameters.
- They describe according to the structure.

The description chosen determines the model of the object. Thus, they will have two types of models: the parametric model and the structural model.

1.2.1.1 Parametric model

The parametric model uses a vector for relativity. Each element of the vector describes a property of the object. For example, as in the functional features, one uses orthogonal basis functions for representation. And thus the image will be represented by a series of orthogonal functions. Suppose C is the contour of image $C(i, j)$ is the i th point on the contour, $i = 1, 2, \dots, n$ (the line includes n points).

1.2.1.2 Structure model

The approach of this model has based on the description of objects thanks to some concepts that represent basic objects in natural language. To describe an object, we use some primitive forms such as lines, arcs ... For example, a rectangle is defined with four segments perpendicular to each other. In this model, one uses a terminating symbol V_t , a non-terminating symbol set called V_n . There is also a set of production rules that describe how to build suitable objects based on simpler or primitive objects. In this approach, we accept an assertion: the structure is the result of applying the law of production according to certain principles starting from a root form starting. Formally, we can consider this model as equivalent to a grammar $G=(V_t, V_n, P, S)$ with:

- V_t is the terminator symbol.
- V_n is a set of symbols that does not end.
- P is the law of production.
- S is the form (start symbol).

1.2.2 Feature

The identification process consists of three main stages:

- Select a model to represent the object.
- Choosing the decision-making law (identification method) and deducing the learning process.
- Learn to recognize.

When the object representation model has been defined, be it quantitative (parametric model) or qualitative (structural model), the identification process moves to the learning phase. Learning is an important stage. Learning operations aim to improve and adjust the zoning of objects into classes.

Identification is to find rules and algorithms that can assign an object to a class or in other words assign an object a name.

1.2.2.1 Supervised learning

The technique of classifying through prior knowledge is called supervised learning. The basic feature of this technique is that one has a library of standard templates. The identifiable sample is compared with the standard sample to see which category it belongs to. For example, in a remote sensing image, one wants to distinguish a rice field, a forest, or a wasteland that has a description of that object. The main problem is to design a system to be able to match the objects in the image with the standard template and decide to assign them to a class. The comparison is done through decision-making procedures based on a tool called a classifier or decision-making function.

1.2.2.2 Non supervised learning

This technique has to manually define the different classes and define the parameters that are specific to each class. Learning without a teacher is more difficult. On the one hand, because the number of layers is unknown, on the other hand, the class characteristics are also unknown. This technique is to do all possible grouping and choose the best. Starting from the data set, many different processing procedures to classify and upgrade gradually to achieve a classification plan.

In general, an identity system can summarize the following diagram.

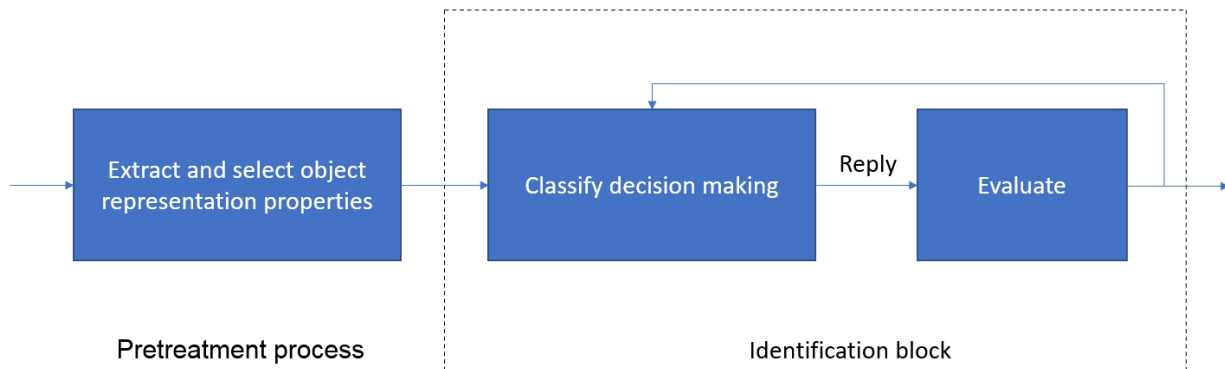


Figure 1: Identity system

Chapter 2: THEORY

2.1 Image processing

2.1.1 Image processing

Humans absorb information through the senses, in which vision plays the most important role. In recent years, with the development of computer hardware, image and graphics processing has grown strongly and there are many applications in life. Image processing and graphics play an important role in robot interaction.

An image can be seen as a collection of pixels and each pixel is seen as a characteristic of light intensity or a sign at a certain position in space and can be seen as a function of n variable $P(c_1, c_2, \dots, c_n)$. Therefore, images in image processing can be viewed as n -dimensional images.

Overview diagram of an image processing system:

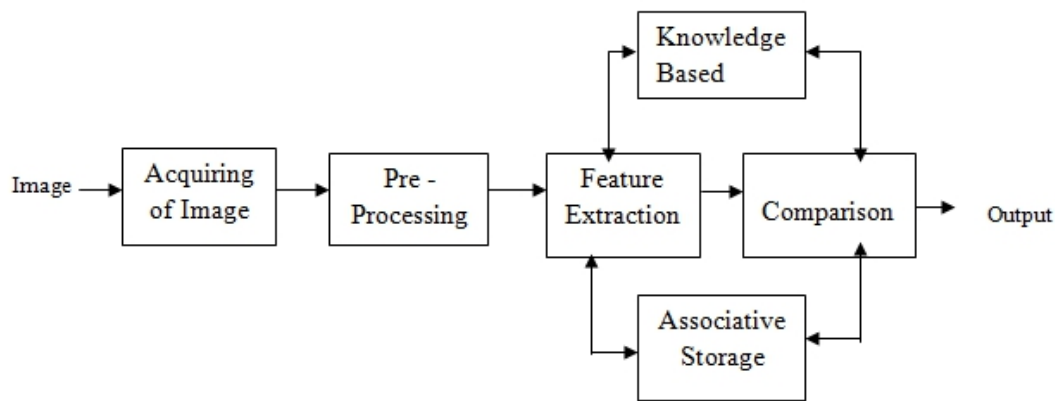


Figure 2: Image processing system

2.1.2 Image representation

Computer images are results obtained by digital methods embedded in different technical devices. The photo storage process has 2 purposes:

-
- + Save memory
 - + Reduced processing time

Storing information in memory has a huge effect on the display, printing, and processing of an image as a collection of dots of the same size, the more pixels you use, the better the picture. The finer the finer and the more detailed the image, the resolution is called this.

Choosing the appropriate resolution depends on the needs and characteristics of each particular image, on that basis, the images are usually represented in two basic models.

2.2 ASP.NET MVC3 technology

2.2.1 Overview of ASP.NET MVC

The Model - View - Controller (MVC) architecture divides a web application into three parts: model, view, and controller. The ASP.NET MVC Framework offers an alternative to the old ASP.NET webform model.

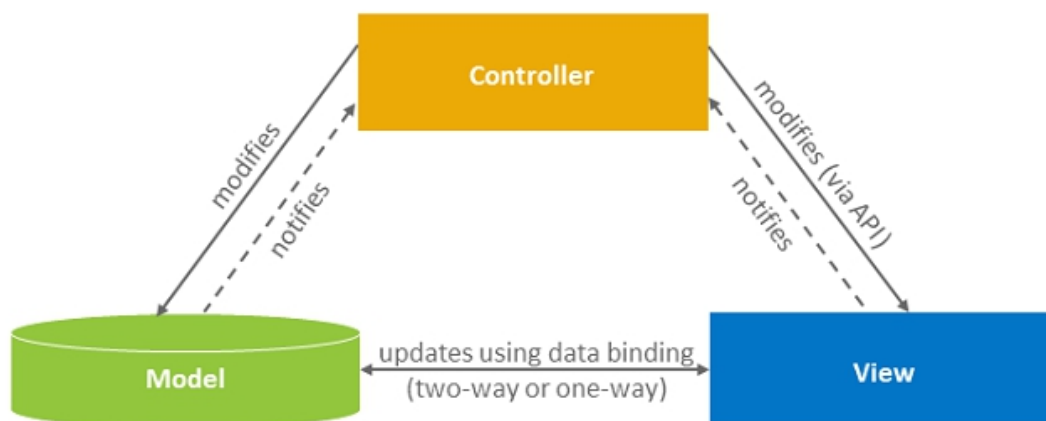


Figure 3: Model - Controller - View Architecture

MVC is a standard design paradigm so many developers can get used to it. MVC Framework includes the following components:

Model: The Model object is a part of the application that implements logical associations for the data layer. Typically, the model object retrieves and stores the model state in a database. For example, a Product object gets information from the database, performs operations on it such as Insert, Update, etc., and then updates the information back to the Products table in the database.

+ **View:** Is a component displaying the user interface.

+ **Controller:** Is the component used to control the user's interaction.

In an MVC application, View is only used to display information, the controller controls and responds to the information for the user. For example, the Controller controls the query, passes it to the Model, from where the Model will use the value to manipulate the database.

2.3 OCR technology

2.3.1 Concept of OCR

Optical Character Recognition (OCR), is a type of computer software created to convert images of handwriting or typewriters (usually scanned with scanners) into documents. document. OCR is made up of a research field in pattern recognition, creation intelligence, and machine vision. Although scholarly research continues, part of OCR's work has shifted to practical application of proven techniques.

Optical character recognition (using optical techniques such as mirrors and lenses) and digital character recognition (using scanners and computer algorithms) were initially considered as two different fields.

Because very few applications exist with true optical techniques, the term Optical Character Recognition is broadened and includes the meaning of digital character recognition.

The recognition system first requires training with patterns of specific characters. "Smart" systems with high recognition accuracy for most fonts are now commonplace. Some systems also have the ability to reproduce document formats close to the original including: images, columns, tables, non-text elements.

Currently, with Vietnamese characters, the software to recognize Vietnamese words in VnDOCR 4.0 is capable of directly recognizing the types of documents scanned through the scanner, without needing to be stored as an intermediate image file. Document pages can be scanned and stored as multipage files. Identification results are stored in formats of Microsoft Word, Excel ... which serve the needs of digitizing data very well.

In addition, there is an OCR Vietnamese project called VietOCR, which is developed on the open source platform tesseract-ocr sponsored by Google. VietOCR is capable of recognizing Vietnamese characters very well. It is an open source Java / .NET program that supports recognition of PDF, TIFF, JPEG, GIF, PNG, and BMP image formats.

2.4 Tesseract Libraries

2.4.1 Tesseract Architecture

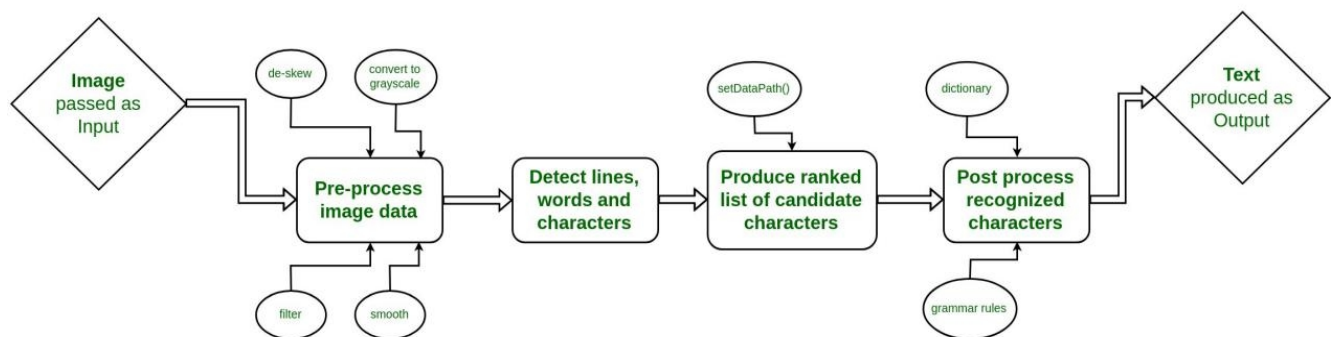


Figure 4: Tesseract Architecture

2.5 Summary of how the Tesseract works

- Outlines are analyzed and stored.
 - Outlines are grouped into Blob.
- Blob is organized into text lines.
- The text line is divided into words.
 - Initially proceed to identify the words above.
 - Satisfactory words will be included in the compatible training set.
 - Lessons learned from the appropriate training set will be developed in step two, in which unsatisfactory words will be identified in the first step.
 - Solve fuzzy space and check for small cap characters.
 - Digital text is output.

In the processing above, Tesseract used:

- Algorithm detects the text flow from skewed pages.
- An algorithm that detects words proportional and non proportional words. The word balance is the word where all of its characters have the same width.
- The algorithm of cutting and importing characters and linking broken characters.

Analyze the language to identify words that are most similar to a group of characters.

- Two character classification classes: a static taxonomy class for implementing training data, and an adaptive classifier to better distinguish upper and lower case characters.

2.6 Some basic functions in OpenCV library

2.6.1 Load an image and display

2.6.1.1 Download an image

```
src mat; // create a src mat
```

```
src = imread (argv [1], 1);
```

- Mat = iplimage.

- Argv [1]: image source (link)

- 1: upload photos in color photo mode

2.6.1.2 Display images

```
// ----- create a window with the name -----
```

```
nameWindow ("source", CV_WINDOW_AUTOSIZE);
```

```
// ----- upload the src image to the source window -----
```

```
imshow ("nguồn", src);
```

2.6.2 Convert color image to grayscale image

This is the function of converting color images to grayscale images of OpenCV library. The step converting color image to grayscale image is an important step necessary for the next steps of image processing.

Convert color photos to grayscale:

```
cvtColor (src, src_gray, CV_BGR2GRAY);
```

- Src: source image

-
- Src_gray: grayscale output.
 - CV_BGR2GRAY: switch from BGR to gray



Figure 5: Convert color image to grayscale image

2.6.3 Find the object using findContours

Here we will find the object through the boundary. It is unorthodox an algorithm so I left findContours in this section as the OpenCV functions useful for finding objects in images.

Here we see how to use the findContours function and see how useful it is when using other tools to highlight the found areas.

`void findContours (InputOutputArray image, OutputArrayOfArrays contours, OutputArray hierarchy, int mode, int method, Point offset = Point ())`

- Image: single channel 8-bit source image. (image via binary processing, threshold splitting, canny)

Contours: contours are stored as a vector of points.

Hierarchy: vector output option that includes information about image topology. It has many factors such as the number of lines.

- Mode: contour retrieval mode: for example CV_RETR_TREE takes all the borders and rebuilds a full hierarchy of nested lines.

-
- Method: method to approximate contour. For example, `CV_CHAIN_APPROX_SIMPLE` compresses the horizontal, vertical and diagonal segments and indicates the position.
 - Offset. Depending on the full compensation.



Figure 6: Using findContours find object

Chapter 3 IDENTIFICATION OF VEHICLE NUMBER PROBLEM

3.1 Definition of license plate identification

3.1.1 Definition

The license plate identification system is a system capable of image analysis and identification of license plates on the vehicle, through videos, video recording equipment, and images. The following is to define information such as owner, ...

Classification of license plate identification applications:

The license plate recognition application is an application that analyzes the image and identifies the license plate from the images captured from the video recording devices. Image sources for the application are numerous. And development, images are directly captured from the camera. In the graduation report, I only stopped at the level of identifying the license plate (cut out the text on the sign) from the photos.

There are many different ways to classify license plate recognition applications. One of the simplest ways is to classify license plate identification applications by the intended use. The license plate recognition applications can be divided into the following two categories:

Type 1: Limiting the view area

- Input: images captured directly from digital image recording devices. Recorded photos are usually limited to the area with license plates.
- Principle of operation: The vehicle must run at a speed slow enough for the image recorder to obtain an image of the license plate area.
- Applications: These license plate identification applications are often used at checkpoints at tollbooths, automatic parking lots, gate guard stations ...



Figure 7: View area

Type 2: Unlimited area view

- Input: input image is obtained from automatically recording the device, regardless of the angle, surrounding objects, the image does not require the capture area only contains license plate, but can be synthesized image as including additional objects such as people, trees, streets ..., as long as the area number plate must be clear enough to be able to recognize characters in that area.
- Principle of operation: due to the unrestricted field of view, the first image can be obtained from a configuration of equipment (camera, camera). And so, the first job is to look in the image, to determine the correct area is the car. Then, perform region splitting and identification. The final attribute on the target uses the identifier result matrix transmitted or stored to serve the needs of the end-user.
- Application: Because it does not depend on the scope of the image, it is possible to use it in many places such as at communication points, at intersections such as junctions and intersections. Control and detect traffic safety violations.

Equipment required:

- + Scan machine
- + Lighting
- + Parts storage - separate images
- + Calculator
- + Some other device use target

3.1.2 Application

- A vehicle license plate identification system is built for the purpose of monitoring and controlling vehicles. Below we cover some of the most common uses for license plate recognition systems:

-
- Traffic toll collection: Install license plate identification system at tollbooths to assist or automate toll collection.
 - Vehicle control at the borderlines: Each country has its own regulations on license plates to serve the management and detection of illegal cross-border vehicles. The installation of license plate identification systems at checkpoints will contribute to national security and inspection.
 - Gate guard stations: The installation of the system will support or automate the opening of gates for vehicles in and out.
 - Handling traffic violations: When installing the system, it will assist in handling road traffic violations and can automatically report the violation. It is currently being piloted on some roads of National Highway 1A to handle speeding errors.
- In addition, the system is also applied to the work of anti-theft of vehicles, automatic parking lots, traffic regulation

3.2 How to solve the license plate identification problem

There are many approaches. Including the following popular approaches.

3.2.1 Regional development approach

The authors Nigel Whyte and Adrien Kiernan are represented for this approach. The idea of this approach is that license plates usually contain a solid color, such as white, and have a certain area. Therefore, it is possible to use the method of developing the area or use the rectangular frame to move in to find the area that satisfies the license plate and carry out identification.

Pros: very simple, and quick to handle images containing only license plate areas.

Cons: when the photo contains more subjects than the license plate area, such as a general shot that includes the outside scenery, this approach becomes ineffective. Therefore, this method is very effective for toll booth systems, gate guard stations, automatic parking.

3.2.2 Hough edge detection and transformation approach

- Team authors Michael Lindenbaum, Rosen Alexander, Vichik Sergey, Sandler Roma are represented for this approach.

The concept of this approach is: The license plate is surrounded by the border. Therefore, you can use the boundary detection method, then use the Hough transformation to extract the vertical and horizontal segments that exist in the

image. The intersection of these lines is the area that contains the license plate. And finally, identify the characters above each sub-region.

- Advantages: high precision. And the identity system has mostly evolved in this approach.

- Disadvantage: The computational complexity is quite high. When the image has more different objects, the calculated volume increases a lot. The purpose is to determine which sub-zones contain license plates.

In addition to the two common approaches above, there are many other approaches to determine exactly which zones contain license plates.

Solution

Size standards:

In each country, there is usually a certain size standard. For our country, the license plate rules are quite uniform for each vehicle, the ratio of length and width for each vehicle is the same.

- For vehicles with a single line, the width/length ratio is $3.5 \leq W / H \leq 4.5$.

- For vehicles with two rows of characters, the ratio length/width is $0.8 \leq W / H \leq 1.4$.

From these features, it can be determined that the sub-regions that satisfy the license plate capacity are very high.

*** Number of characters in license plate:**

Each character usually has the ratio of the width and height to the length and width of the license plate. For example, the height of each letter is always less than 85% of the license plate height and always greater than 33% of the license plate height. The width of the characters is no more than 20% of the length of the license plate. Each letter of the license plate is considered a sub-connected region. Therefore, we can count the child connective region where the product is the character. Note that the number of characters on the license plate is from 6 to 10 characters. In our country, only the number of characters on each line of the car is in the range of 6 to 9 characters. So we can use threshold [6.9] to identify the license plate area.

From the above comments, we can offer a solution to the recognition problem:

Use edge detection and Hough transform. Then use two properties on the license plate to determine the exact sub-zone that contains the license plate. Once you have identified the sub-area containing the number plate, the characters will be recognized. This is the approach of many identification problems using boundary detection combined with Hough.

Use threshold separation and object partitioning. Using threshold splitting eliminates regions that are not number plates (with the same size ratio but with a lower grayscale below 100). Then proceed to find the object area according to the scale of the license plate, we will obtain the number of zoning plates less than. Then use the feature plate finder again with the license plate partitions so that the number of separators is compared with the number of valid characters to create a more accurate license plate partition search. And also proceed to recognize each cut character.

In my report, I will present solution 2, including the following steps:

Step 1: Find the number plate area

- With input grayscale, perform threshold mapping (or canny border detection) and find objects.
- Separate objects with a wide/long scales within the number plate as the source for step 2.

Step 2: Separate each letter on the license plate:

Search for the object on the source provided by step 1, then compare the number of received objects with the number of characters in the name of the license plate. If that is true, those are the number of plates that are most likely to be the highest license plate. Separating each character for optical identification such as OCR or neural network, but in this report, I just stop at separating each letter of the license plate.

3.3 Detecting the number plate container

3.3.1 Steps to detect vehicle license plate

Diagram of detection steps for license plate area.

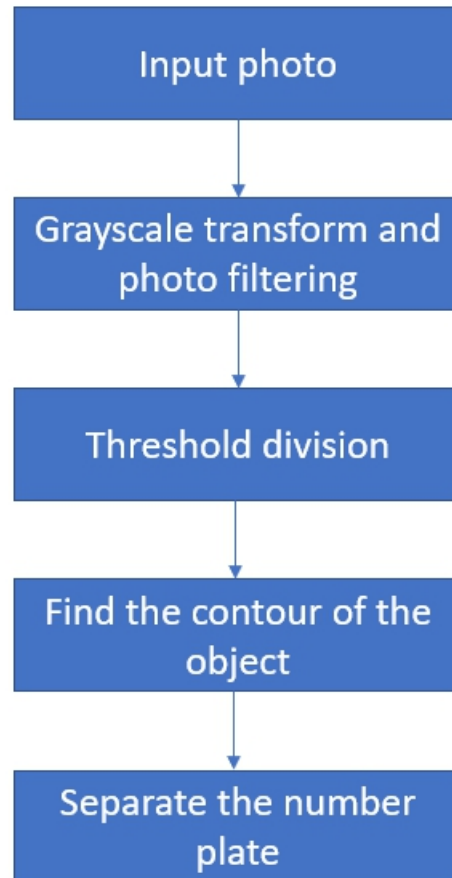


Figure 8: Steps to detect license plate container

Stage 1: Transform the gray image and filter the image

The input image is an image, converted to an image with 256 levels of grayscale and filtered to reduce noise.

To perform a noise reduction filter I used a bilateral filter to reduce the possibility of boundary-blurring.



Figure 9: Transform the gray image and filter the image

Stage 2: Conduct thresholding or boundary detection

There are two different thresholding processes: automatic and threshold division not automatically.

With the number plate partition is quite small compared to the entire image, the automatic thresholding process by taking the Histogram will cause errors and do not clarify the number plate area.

Through experiments, we see that the number plate partition usually has a threshold from 100 - 220, so in this part of me, I try to find the number plate with each threshold by the loop. The loop will start from the threshold 100 to 220 because normally when detecting the sea number of clear signs, the threshold 100 - 120 can stop the loop, saving calculation steps.

At the end of this stage, we obtain a single-channel 8-bit image as the source for stage 3.

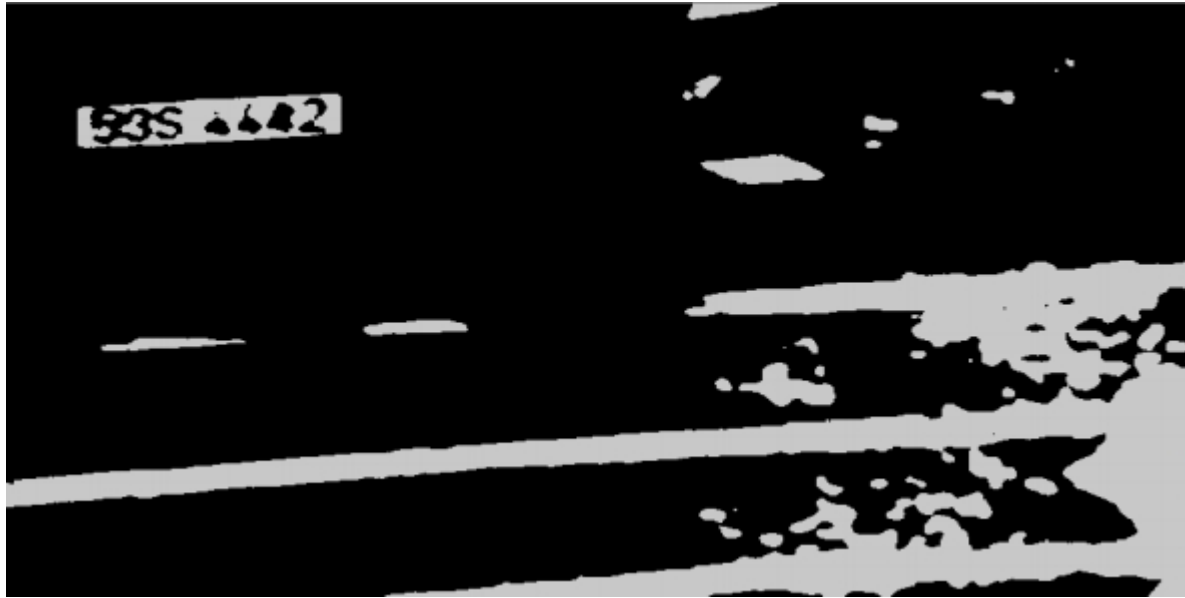


Figure 10: 8-bit image

Stage 3: Find contour of object

When we proceed to the threshold part will make the number plate clear, in this phase, we can use the edge detection methods.

When there is a single-channel 8-bit image obtained in stage 2, we proceed to find the waters as private objects using the findcontours function (in the OpenCV library) to extract the regions and get the parameters of the regions as point coordinates. of the edges, the area of the region.

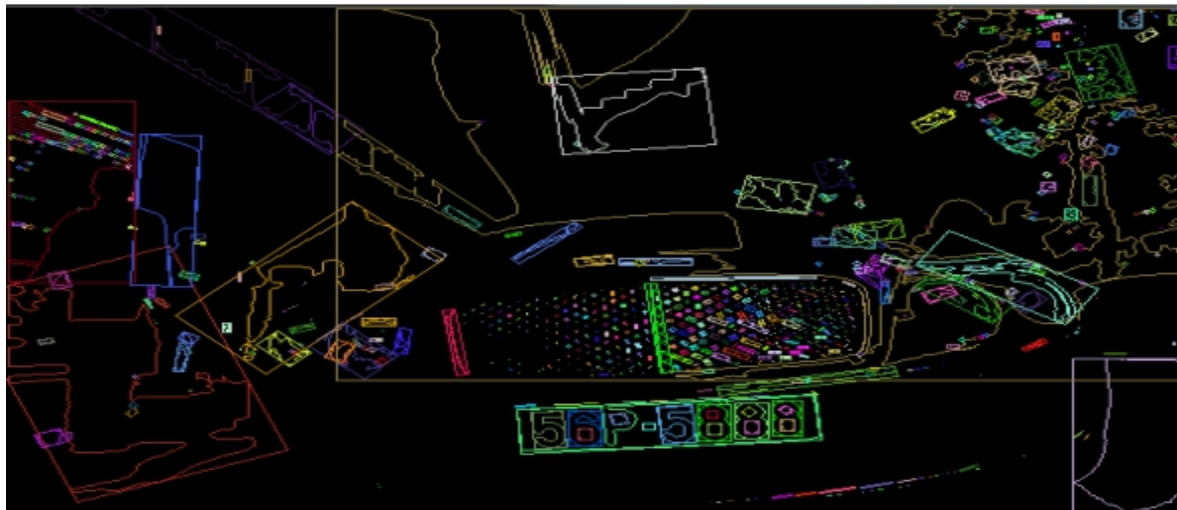


Figure 11 Find outlines of objects

Stage 4: Separation of number plates

After obtaining the parameters of stage 3 found. Based on the characteristics of the license plate, she proceeded as follows:

Find the contour of the object with the criteria of $3.5 \leq W/H \leq 4.5$ or

$0.8 \leq W/H \leq 1.4$ to find out which areas could be the most number of plates.

To limit the number of purple areas, I continue to filter the image according to the area criteria to limit the areas that are too small and too large compared to the number plate. With the input image being resized to the 640X480 size, the number plate area will be in the area from 1500 - 25000.

Proceed to crop the areas that might be the most number plate on the filtered gray image in stage 1 as the source for the character trimming phase to model the character recognition (self-build) processes. At this stage, it is possible to combine character recognition problems to detect license plates.

Separate each character

In this section, the main purpose is to separate the characters to serve as a training model for character recognition on the license plate.

The stages in separating the characters are as follows:

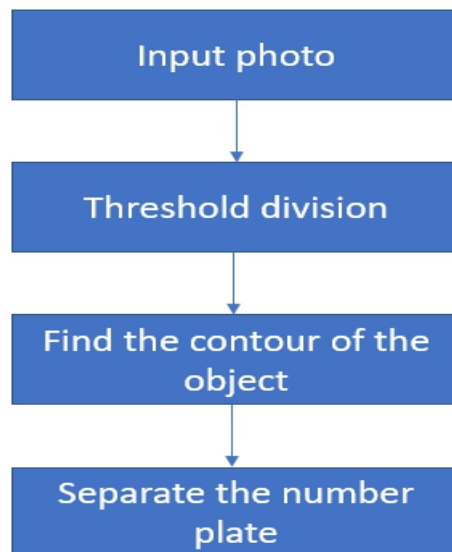


Figure 12: The character separation stages

Chapter 4: BUILDING A PROGRAM TO RECOGNIZE VEHICLE NUMBER PLATE

4.1 Selection of tools to build

4.1.1 Microsoft Visual Studio 2019

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as websites, web applications, and web services.

Visual Studio uses Microsoft software development platform as well

Windows API, Windows Forms, Windows Presentation Foundation, Windows Store, and Microsoft Silverlight. It can produce both machine language and management codes.

Visual Studio supports many different programming languages and allows code editor and debugger to support (to varying degrees) almost any programming language. The integrated languages include C, C++ and C++ / CLI (through Visual C++), VB.NET (through Visual Basic.NET), C# (through Visual C#) and F# (through Visual F#). Support for other languages like Java / J2SE, Python and Ruby is via a separate installation service. It also supports XML / XSLT, HTML / XHTML, JavaScript and CSS.

4.1.2 Microsoft .Net framework

Microsoft .NET Framework is a programming model of Microsoft, it provides a shared library support for applications programmed and run on the .NET platform.

4.2 License plate identification program

4.2.1 The main interface of the software

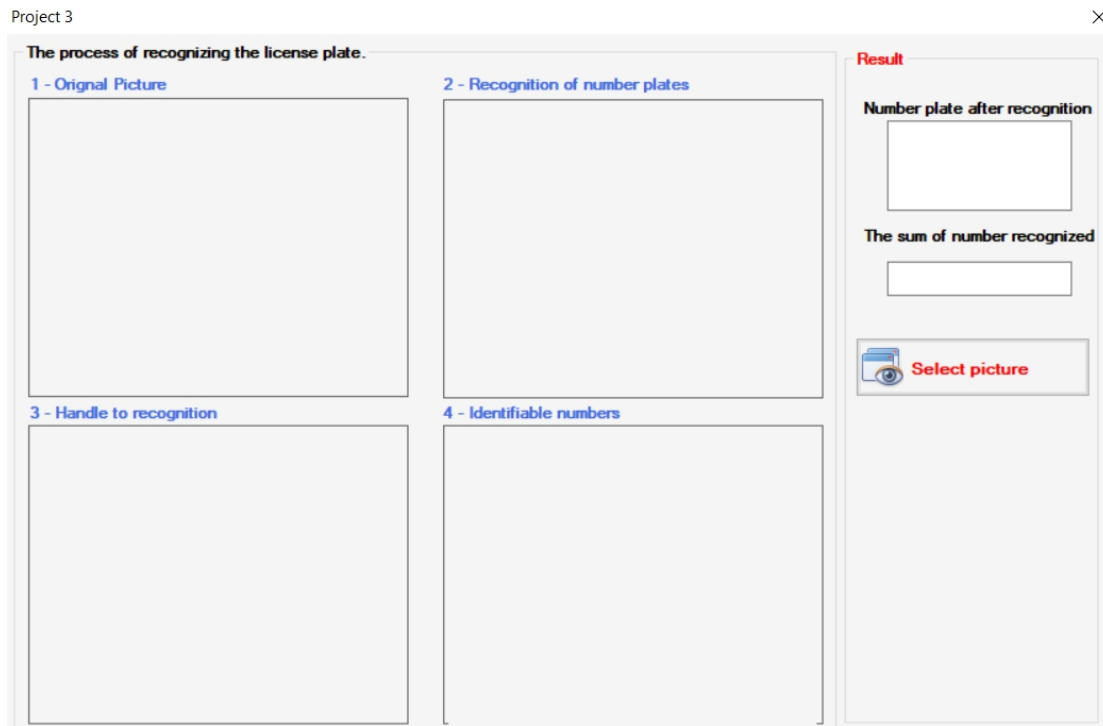


Figure 13: The interface of license plate recognition software



Figure 14: Identification of license plate

4.2.2 Steps to identify license plate

a) Detecting the area containing the license plate



Figure 15: Discover of license plate holder

From the original image to the software, the software determines the position of the license plate.

b.Character container identification

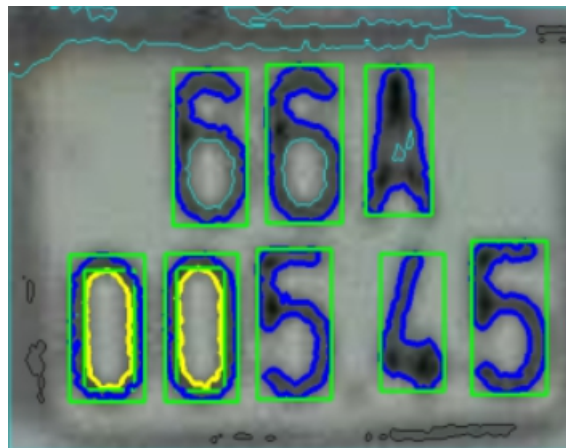


Figure 16: Identification of character container

Find and recognize characters on number plates

c) Image processing



Figure 17: Image processing

Filter out noise and redundant images

d.Character recognition

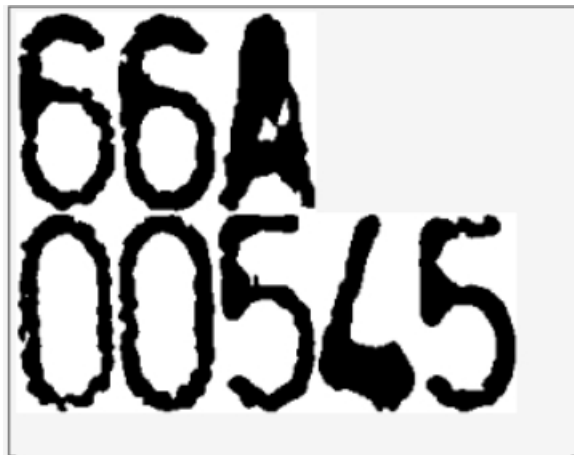


Figure 18: Identification of characters

Remove dot (for 5-digit plates), produce recognizable characters.

e. Results

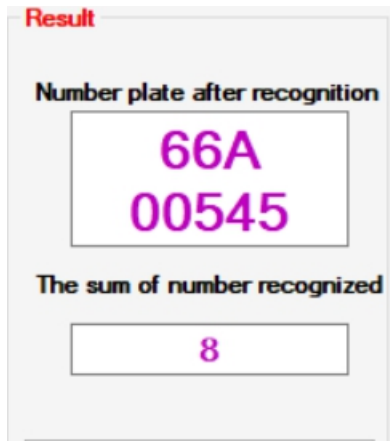


Figure 19: Identification results

The result is the string of characters after recognition.

4.2.3 Pros and cons

a. Advantages

Fast setting algorithm, find number plate and isolate characters with high success rate (in normal number plates), found in natural images, the number plate is tilted.

The result is a string of identifiable characters.

b. Disadvantages

- Poor recognition of number plates with low background contrast such as red background plates, blurred backgrounds, number plates with thin borders, number plates fitted with decorative equipment, plates with glare caused by light , and signs with unclear numerals.

- Only the standard number plates can be recognized as Vietnam's number plates. For foreign license plates (with special colors, letters and images), they cannot be identified.

Preference

<https://stackoverflow.com/questions/981378/how-to-recognize-vehicle-license-number-plate-anpr-from-an-image>

<https://www.freelancer.com/projects/software-architecture/windows-form-webform-alpr-automatic/?ngsw-bypass=&w=f>

https://www.youtube.com/watch?v=UgGLo_QRHJ8&t=233s

<https://www.miai.vn/2019/11/26/nhan-dien-bien-so-xe-chuong-3-phat-hien-bien-so-bang-opencv-thuan/>

https://www.camera-sdk.com/p_6774-how-to-implement-license-plate-recognition-in-c-.html