BORDEAUX UNIVERSITY

Class: MINF18

**Instructor: Pascal**

**Professional Project**

# License Plate Number Recognition System (LPNRS)

**Members:**

Phan Thanh Son: phanthanhsont@gmail.com

Tran Quang Nhat: tranquangnhat2211@gmail.com

GitHub For Demo Project and report document:

https://github.com/nhat2211/license-plate-regconizing-minf18

## Contents

# 1. Introduction

Nowaday, The License plate number of cars be very common and be applied very many in the life. Such as: toll fee stations on the highway, parking, etc... The purpose of this project be research the way which how to do that we are able to detect the content on images be captured. And also be make a simple demo for detecting the license plate from images.

### 1.1 LPNRS systems as a practical application of artificial intelligence

A standalone information system without data will be not able work and has sense for detecting and analysis for the exactly result. We must have a standard data set for the purpose detecting the content in The license plate.This data such as images in a standard for the size and can be collected by human or by some modern equipments which can capture the images of license plate of cars, for example that is the camera which be installed in toll fee station on the highway or in the parking system also will have the cameras for the purpose be capture and collect the images after that The LPNRS will use these images for standard data.

In toll-fees collection station on the highway, number plates be captured and calculate the fees when the car go over the station.The owner of cars will have to register a user which link with their car's number plate. On the system will have data of users link with the number plate of the car therefore after that camera capture the image of number plate that the system will know be who be owner of the car and will charge toll fee exactly.

In the parking, numberplate be captured and calculate the time in duration of the parking.When a car enters an input gate , number plate will be stored in database.When this car exists a output gate, the number plate will also captured one again and will compare with the last number plate in database if be matched then will calculate the duration and allowing the car exists.

For the other example be LPNRS system be used for control cars of the staff in a company. The only cars of staff have to grant to access the gate.

In some developed countries, LPNRS system will store all number license plate of the vehicles therefore will be very easy to checking some cars has be in black list or not, also helping the management and checking violation of the vehicle easier and fastly.

### 1.2 Mathematical aspects of number plate recognition systems

In most cases, number plate be used to identify the vehicle. Every vehicle will have only one the number plate and this number is unique. Human can be easily to read the number plate in their eyes but The computer be not.With the machine, a number plate be only a grey picture defined as a two-dimensional function f(x,y), where x and y are spatial coordinates, and f is a light intensity at that point.Therefore, we need to design a robust mathematic machine which will be able to extract semantics from spatial domain of the captured images. The function which we will implement in professional project be called is **LPNRS (Full name is License Plate Number Recognition System).**

The Design and research in professional project for LPNRS is a field of artificial intelligence, machine learning, pattern recognition and netural network

### 1.3 Physical aspects of number plate recognition systems

License Plate Number Recognition System is a set of hardware and software components.Hardware such as camera ,sensors to detect the cars when it go over station by the sensors and the camera will capture the images which has number plate of the cars.

The hardwards necessary when build a LPNRS includes:

- Cameras: Cameras be installed at station to capture number plate images of cars go over the station
- Sensors: Detecting the cars go over the station and trigger Camera capture
- Computers at stations: Computer has software to control and collect,detecting images and after that sending to SERVER
- Servers: SERVER be used to store number plate and information of user in Database.
- The other equipments: necessary to setup the system.

## 2. Principles of number plate area detection

The first step in a process of automatic number plate recognition is a detection of a number plate area. We have to find and build a algorithm be able to detect number range in the images. This range is a rectangle.The machine do not understand what “Road” and “vehicle” or whatever else is.Because of this, we need to find a alternative definition of a number plate based on descriptors that will be comprehensible for machines.

We will define the number plate as a “rectangular area with increased occurrence of horizontal and vertical edges”.

The high density of horizontal and vertical edges on a small area
is in many cases caused by contrast characters of a number plate, but not in every case. This process can sometimes detect a wrong area that does not correspond to a number plate. Because of this, we often detect several candidates for the plate by this algorithm, and then we choose the best one by a further heuristic analysis

Let a input image be defined by function f(x,y), where x and y are spatial coordinates, and f is an intensity of light at that points.This function is always discrete on computers such as $x \in N^{\circ} \cap y \in N^{\circ}$ , $where\ N^{\circ}$ denotes the set of natural numbers including zero.

We define operations such as edge detection or rank filtering as mathematical transformations of function f.

The detection of a number plate area consists of a series of convolve operations. Modified snapshot is then projected into axes x and y . These projections are used to determine an area of a number plate

### 2.1 Edge detection and rank filtering

We can use a periodical convolution of the function f with specific types of matrices m to detect various types of edges in an image:

$$f'(x,y) = f(x,y) \tilde{*} \mathbf{m}[x,y] = \sum_{i=0}^{w-1} \sum_{j=0}^{h-1} f(x,y) \cdot \mathbf{m}\left[\operatorname{mod}_w(x-i), \operatorname{mod}_h(y-j)\right]$$

where $w$ and $h$ are dimensions of the image represented by the function $f$

*Note: The expression* $\mathbf{m}[x,y]$ *represents the element in* $x^{th}$ *column and* $y^{th}$ *row of matrix* $\mathbf{m}$.

(Source: http://javaanpr.sourceforge.net/)

### 2.1.1 Convolution matrices

Each image operation (or filter) is defined by a convolution matrix. The convolution matrix
defines how the specific pixel is affected by neighboring pixels in the process of convolution.

Individual cells in the matrix represent the neighbors related to the pixel situated in the centre of
the matrix. The pixel represented by the cell y in the destination image (fig. 2.1) is affected by
the pixels x0…..x8 according to the formula:

$$y = x_0 \times m_0 + x_1 \times m_1 + x_2 \times m_2 + x_3 \times m_3 + x_4 \times m_4 + x_5 \times m_5 + x_6 \times m_6 + x_7 \times m_7 + x_8 \times m_8$$

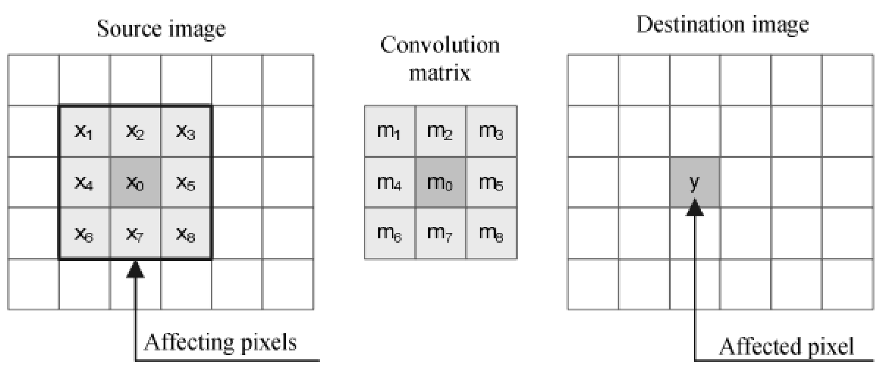



**Figure 2.1:** The pixel is affected by its neighbors according to the convolution matrix.

(Source: http://javaanpr.sourceforge.net/)

**Horizontal and vertical edge detection**

To detect horizontal and vertical edges, we convolve source image with matrices mhe and mve .
The convolution matrices are usually much smaller than the actual image. Also, we can use
bigger matrices to detect rougher edges

$$\mathbf{m}_{he} = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} ; \mathbf{m}_{ve} = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

**Sobel edge detector**

The Sobel edge detector uses a pair of 3x3 convolution matrices. The first is dedicated for evaluation of vertical edges, and the second for evaluation of horizontal edges .

$$\mathbf{G}_x = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} ; \mathbf{G}_y = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

The magnitude of the affected pixel is then calculated using the formula $|\mathbf{G}| = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2}$ . In praxis, it is faster to calculate only an approximate magnitude as $|\mathbf{G}| = |\mathbf{G}_x| + |\mathbf{G}_y|$ .

**Horizontal and vertical rank filtering**

Horizontally and vertically oriented rank filters are often used to detect clusters of high density
of bright edges in the area of the number plate. The width of the horizontally oriented rank filter
matrix is much larger than the height of the matrix ( w h $\gg$ ), and vice versa for the vertical rank
filter ( w h $\ll$ ).

To preserve the global intensity of an image, it is necessary to each pixel be replaced with an average pixel intensity in the area covered by the rank filter matrix. In general, the convolution matrix should meet the following condition:

$$\sum_{i=0}^{w-1}\sum_{j=0}^{h-1}\mathbf{m}_{hr}\left[i,j\right]=1.0$$

where $w$ and $h$ are dimensions of the matrix.

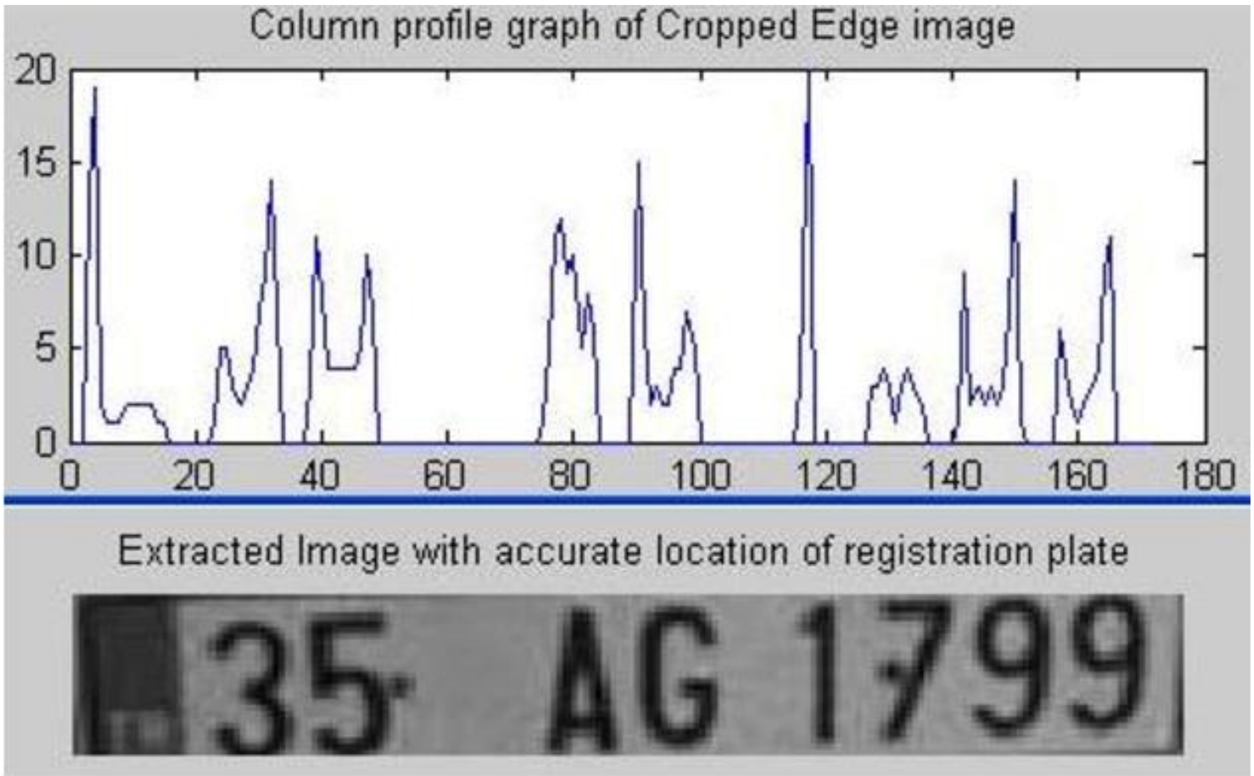

51H
619.39


خصوصي
لبنان
B123456

### 2.2 Horizontal and vertical image projection

We can detect an area of the number plate according to a statistics of the image. We will use horizontal and vertical projection of an image into the axes *x* and *y*.

The vertical projection of the image is a graph




Source of image: [www.semanticscholar.org](http://www.semanticscholar.org)

### 2.3 Vertical Detection

This principle is applied iteratively to detect several possible bands. The yb0 and yb1 coordinates are computed in each step of iterative process. After the detection, values of projection py in interval y y b b 0 1 , are zeroized. This idea is illustrated by the following pseudo-code:

Let L is a list of detected points

For i:=0 to number of band to be detected do

Begin

Detect yb0 and yb1 by analysis of projection py

Save yb0 and yb1 to a list L

Zeroize interval (yb0,yb1)

End

### 2.4 Horizontal Detection

**First phase:** At first, the band must be processed by a vertical detection filter. If w is a width of the band (or a width of the analyzed image), the corresponding horizontal projection $p_x^r(x)$ contains w values:

$$p_x(x) = \sum_{j=y_{b0}}^{y_{b1}} f(x, j)$$

**Second phase:**

In the second phase of detection, the horizontal position of a number plate is detected in another way. Due to the skew correction between the first and second phase of analysis, the wider plate area must be duplicated into a new bitmap

### 2.5 Heuristic analysis and priority selection of number plate candidates

The basic following steps to detect number license plate is below:

1. Detect possible number plate candidates

2. Sort them according to their cost (determined by a basic heuristics).

3. Cut the first plate from the list with the best cost.

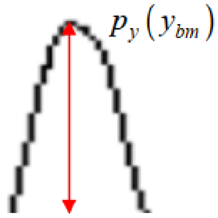
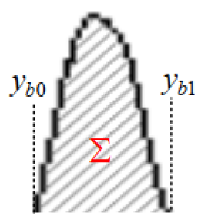
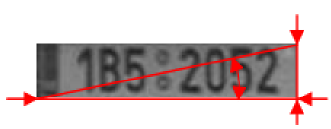
4. Segment and analyze it by a deeper analysis (time consuming).

5. If the deeper analysis refuses the plate, return to the step 3

### 2.5.1 Priority selection and basic heuristic analysis of bands

The basic analysis is used to evaluate the cost of candidates, and to sort them according to this cost. There are several independent heuristics, which can be used to evaluate the cost $\alpha i$ . The heuristics can be used separately, or they can be combined together to compute an overall cost of candidate by a weighted sum:

$$\alpha = 0.15 \cdot \alpha_1 + 0.25 \cdot \alpha_2 + 0.4 \cdot \alpha_3 + 0.4 \cdot \alpha_4$$

| Heuristics | Illustration | Description |
|---|---|---|
| $\alpha_1 = \lvert y_{b0} - y_{b1} \rvert$ |  | The height of band in pixels. Bands with a lower height will be preferred. |
| $\alpha_2 = \frac{1}{p_y(y_{bm})}$ |  | The "$p_y(y_{bm})$" is a maximum value of peak of vertical projection of snapshot, which corresponds to the processed band. Bands with a higher amount of vertical edges will be preferred. |
| $\alpha_3 = \frac{1}{\sum_{y=y_{b0}}^{y_{b1}} p_y(y)}$ |  | This heuristics is similar to the previous one, but it considers not only the value of the greatest peak, but a value of area under the graph between points $y_{b0}$ and $y_{b1}$. These points define a vertical position of the evaluated band. |
| $\alpha_4 = \left\lvert \frac{\lvert x_{p0} - x_{p1} \rvert}{\lvert y_{b0} - y_{b1} \rvert} - 5 \right\rvert$ |  | The proportions of the one-row number plates are similar in the most countries. If we assume that width/height ratio of the plate is about five, we can compare the measured ratio with the estimated one to evaluate the cost of the number plate. |

### 2.5.2 Deeper Analysis

The deeper analysis determines the validity of a candidate for the number plate. Number plate candidates must be segmented into the individual characters to extract substantial features. The list of candidates is iteratively processed until the first valid number plate is found. The candidate is considered as a valid number plate, if it meets the requirements for validity.

Assume that plate p is segmented into several characters p p 0 1 … n- , where n is a number of characters. Let wi be a width of ith character . Since all segmented characters have roughly uniform width, we can use a standard deviation of these values as a heuristics:

$$\beta_1 = \sqrt{\frac{1}{n}\sum_{i=0}^{n-1}\left(w_i - \overline{w}\right)^2}$$

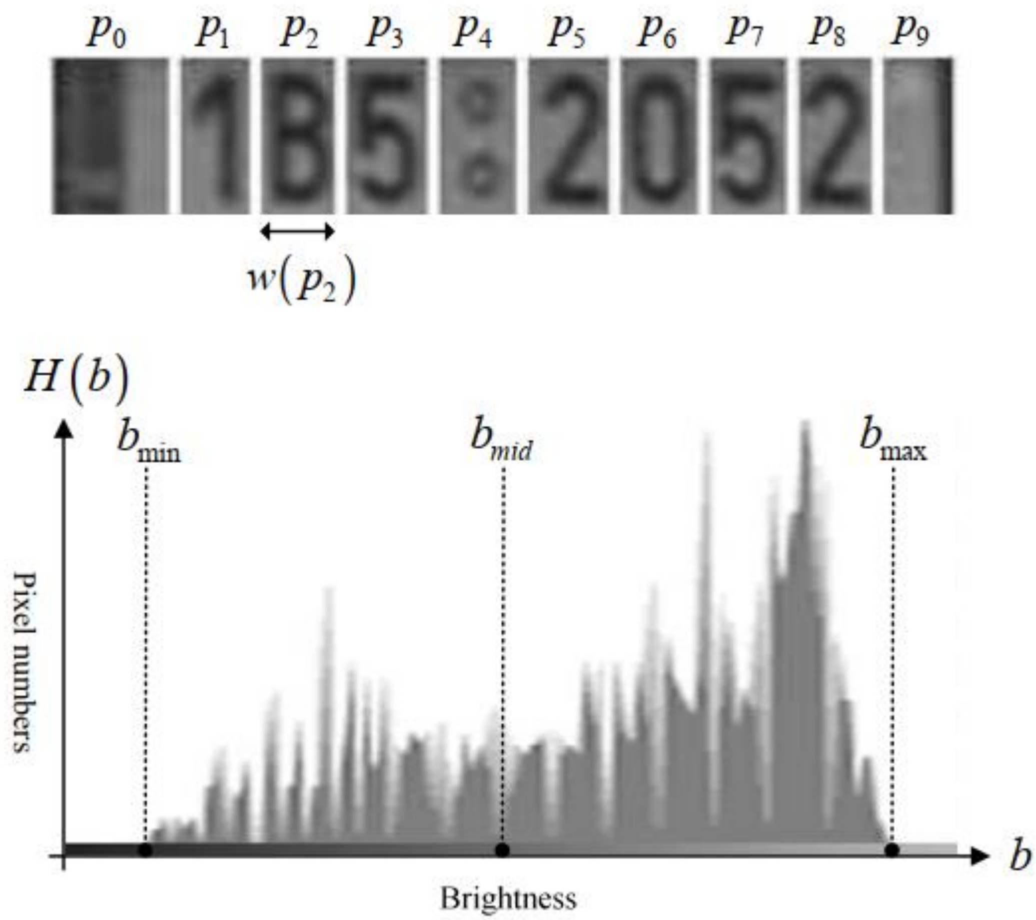
where $\overline{w}$ is an arithmetic average of character widths $\overline{w} = \frac{1}{n}\sum_{i=0}^{n-1} w_i$ .

If we assume that the number plate consists of dark characters on a light background, we can use a brightness histogram to determine if the candidate meets this condition. Because some country-specific plates are negative, we can use the histogram to deal with this type of plates.

Let $H(b)$ be a brightness histogram, where $b$ is a certain brightness value. Let $b_{\min}$ and $b_{\max}$ be a value of a darkest and lightest point. Then, $H(b)$ is a count of pixels, whose values are equal to $b$. The plate is negative when the heuristics $\beta_2$ is negative:

$$\beta_2 = \left(\sum_{b=b_{mid}}^{b_{\max}} H(b)\right) - \left(\sum_{b=b_{\min}}^{b_{mid}} H(b)\right)$$

where $b_{mid}$ is a middle point in the histogram, such as $b_{mid} = \frac{b_{\max} - b_{\min}}{2}$ .



$p_0$ $p_1$ $p_2$ $p_3$ $p_4$ $p_5$ $p_6$ $p_7$ $p_8$ $p_9$
$w(p_2)$
$H(b)$
$b_{\min}$
$b_{mid}$
$b_{\max}$
Pixel numbers
Brightness
$b$

3. **Principles of plate segmentation**

The next step after the detection of the number plate area is a segmentation of the plate. The segmentation is one of the most important processes in the automatic number plate recognition, because all further steps rely on it. If the segmentation fails, a character can be improperly divided into two pieces, or two characters can be improperly merged together.We can use a horizontal projection of a number plate for the segmentation, or one of the more sophisticated methods, such as segmentation using the neural networks. If we assume only one-row plates, the segmentation is a process of finding horizontal boundaries between characters. Section 3.2 deals with this problematic.
The second phase of the segmentation is an enhancement of segments. The segment of a plate contains besides the character also undesirable elements such as dots and stretches as well as redundant space on the sides of character. There is a need to eliminate these elements and extract only the character. Section 3.3 deals with these problems.

**3.1 Segmentation of plate using a horizontal projection**

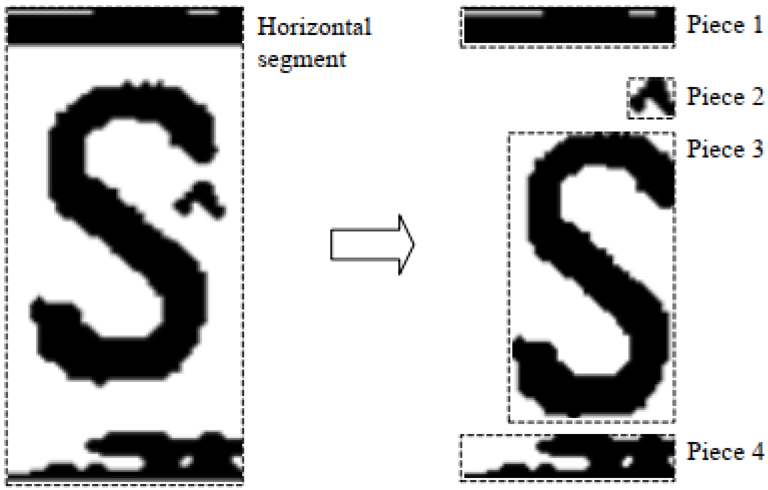
The algorithm of segmentation iteratively finds the maximum peak in the graph of vertical projection.following step below:

1. Determine the index of the maximum value of horizontal projection:
   $$x_m = \arg \max_{0 \le x < w} \{ p_x(x) \}$$
2. Detect the left and right foot of the peak as:
   $$x_l = \max_{0 \le x \le x_m} \{ x \mid p_x(x) \le c_x \cdot p_x(x_m) \}$$
   $$x_r = \min_{x_m \le x < w} \{ x \mid p_x(x) \le c_x \cdot p_x(x_m) \}$$
3. Zeroize the horizontal projection $p_x(x)$ on interval $\langle x_l, x_r \rangle$
4. If $p_x(x_m) < c_w \cdot v_m$, go to step 7.
5. Divide the plate horizontally in the point $x_m$.
6. Go to step 1.
7. End.

**3.2 Extraction of characters from horizontal segments**

The segment of plate contains besides the character also redundant space and other undesirable
elements. We understand under the term “segment” the part of a number plate determined by a
horizontal segmentation algorithm. Since the segment has been processed by an adaptive thresholding filter, it contains only black and white pixels. The neighboring pixels are

grouped
together into larger pieces, and one of them is a character. Our goal is to divide the segment into
the several pieces, and keep only one piece representing the regular character.



## 4. Feature extraction and normalization of characters

To recognize a character from a bitmap representation, there is a need to extract feature descriptors of such bitmap. As an extraction method significantly affects the quality of whole OCR process, it is very important to extract features, which will be invariant towards the various light conditions, used font type and deformations of characters caused by a skew of the image.
The first step is a normalization of a brightness and contrast of processed image segments. The characters contained in the image segments must be then resized to uniform dimensions (second step). After that, the feature extraction algorithm extracts appropriate descriptors from the normalized characters (third step). This chapter deals with various methods used in the process of normalization.

### 4.1 Normalization of brightness and contrast

The brightness and contrast characteristics of segmented characters are varying due to different light conditions during the capture. Because of this, it is necessary to normalize them. There are many different ways, but this section describes the three most used: histogram normalization, global and adaptive thresholding.
Through the histogram normalization, the intensities of character segments are redistributed on the histogram to obtain the normalized statistics.
Techniques of the global and adaptive thresholding are used to obtain monochrome representations of processed character segments. The monochrome (or black & white) representation of image is more appropriate for analysis, because it defines clear boundaries of contained characters.

### 4.2 Normalization of dimensions and resampling

Before extracting feature descriptors from a bitmap representation of a character, it is necessary to normalize it into unified dimensions. We understand under the term "resampling" the process of changing dimensions of the character. As original dimensions of unnormalized characters are usually higher than the normalized ones, the characters are in most cases downsampled. When we downsample, we reduce information contained in the processed image.
There are several methods of resampling, such as the pixel-resize, bilinear interpolation or the weighted-average resampling. We cannot determine which method is the best in general, because the successfulness of particular method depends on many factors. For example, usage of the weighed-average downsampling in combination with a detection of character edges is not a good solution, because this type of downsampling does not preserve sharp edges (discussed later). Because of this, the problematic of character resampling is closely associated with the problematic of feature extraction.
We will assume that $m \times n$ are dimensions of the original image, and $m' \times n'$ are

dimensions of the image after resampling. The horizontal and vertical aspect ratio is defined as $r_x = m'/m$ and $r_y = n'/n$, respectively.

## 4.3 Feature extraction

Information contained in a bitmap representation of an image is not suitable for processing by computers. Because of this, there is need to describe a character in another way. The description of the character should be invariant towards the used font type, or deformations caused by a skew. In addition, all instances of the same character should have a similar description. A description of the character is a vector of numeral values, so-called “descriptors”, or “patterns”:

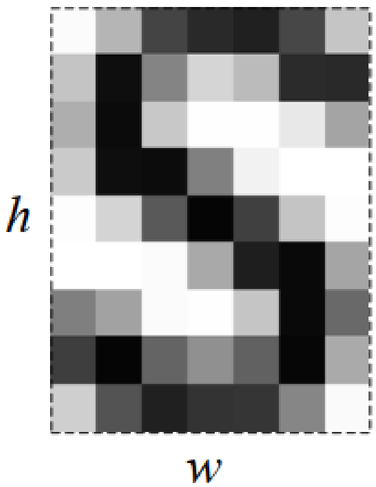
$$\mathbf{x} = \left(x_0, \ldots, x_{n-1}\right)$$

Generally, the description of an image region is based on its internal and external representation. The internal representation of an image is based on its regional properties, such as color or texture. The external representation is chosen when the primary focus is on shape characteristics. The description of normalized characters is based on its external characteristics because we deal only with properties such as character shape. Then, the vector of descriptors includes characteristics such as number of lines, bays, lakes, the amount of horizontal, vertical and diagonal or diagonal edges, and etc. The feature extraction is a process of transformation of data from a bitmap representation into a form of descriptors, which are more suitable for computers. If we associate similar instances of the same character into the classes, then the descriptors of characters from the same class should be geometrically closed to each other in the vector space. This is a basic assumption for successfulness of the pattern recognition process.
This section deals with various methods of feature extraction, and explains which method is the most suitable for a specific type of character bitmap. For example, the “edge detection” method should not be used in combination with a blurred bitmap.

### 4.3.1 Pixel Matrix

The simplest way to extract descriptors from a bitmap image is to assign a brightness of each pixel with a corresponding value in the vector of descriptors. Then, the length of such vector is equal to a square ( w x h ) of the transformed bitmap:

Bigger bitmaps produce extremely long vector of descriptors, which is not suitable for recognition. Because of this, size of such processed bitmap is very limited. In addition, this method does not consider geometrical closeness of pixels, as well as its neighboring relations.Two slightly biased instances of the same character in many cases produce very different description vectors. Even though, this method is suitable if the character bitmaps are too blurry or too small for edge detection.




$$\mathbf{x} = \begin{pmatrix} 251, 181, 068, 041, 032, 071, 197, \\ 196, 014, 132, 213, 187, 043, 041, \\ 174, 011, 200, 254, 254, 232, 164, \\ 202, 014, 012, 128, 242, 255, 255, \\ 253, 212, 089, 005, 064, 196, 253, \\ 255, 255, 251, 196, 030, 009, 165, \\ 127, 162, 251, 254, 197, 009, 105, \\ 062, 005, 100, 144, 097, 006, 170, \\ 207, 083, 032, 051, 053, 134, 250 \end{pmatrix}$$

# 5. Recognition Characters

## 5.1 The following pseudo-code demonstrates the active phase of feed-forward neural network.

procedure activePhase (input: $\mathbf{w}$, // vector of thresholds and weights
$\mathbf{x}$; // input pattern to be classified
output: $\mathbf{z}$, // vector of activities of neurons in hidden layer
$\mathbf{y}$ // vector of activities of neurons in output layer (neural network response)
)
begin
// first step: evaluate activities of neurons in the hidden layer
for each neuron in hidden layer with index $i \in 0,\ldots,n-1$ do
begin
let $\xi = \mathbf{w}\left[\vartheta_i^{(1)}\right]$
for each input with index $j \in 0,\ldots,m-1$ do
let $\xi = \xi + \mathbf{w}\left[w_{i,j}^{(1)}\right] \cdot x_j$
let $z_i = g(\xi)$
end

// second step: evaluate activities of neurons in the output layer
for each neuron in output layer with index $i \in 0,\ldots,o-1$ do
begin
let $\xi = \mathbf{w}\left[\vartheta_i^{(2)}\right]$
for each input with index $j \in 0,\ldots,n-1$ do
let $\xi = \xi + \mathbf{w}\left[w_{i,j}^{(2)}\right] \cdot z_j$
let $y_i = g(\xi)$
end
end

## 5.2 Heuristic analysis of characters

Step by step:

1. Segment the plate
2. Analyse the brightness and contrast of segments and exclude faulty ones.
3. Apply the piece extraction algorithm on segments
4. Analyse the dimensions of segments and exclude faulty ones.

## SUMMARY

The objective of this thesis was to study and resolve algorithmic and mathematical aspects of the automatic number plate recognition systems, such as problematic of machine vision, pattern recognition, OCR and neural networks. The problematic has been divided into several chapters,according to a logical sequence of the individual recognition steps. Even though there is a strong succession of algorithms applied during the recognition process, chapters can be studied independently.

## References:

**http://javaanpr.sourceforge.net/**