RECOGNITION OF VEHICLE NUMBER PLATES

Ondrej Martinsky
Brno University of Technology, Faculty of Information Technology
Bozetechova 2, 61266 Brno, Czech Republic

Keywords: Machine vision, artificial intelligence, neural networks, optical character recognition, OCR, ANPR.

Abstract: This work deals with problematic from field of artificial intelligence, machine vision and neural networks in construction of an automatic number plate recognition system (ANPR). This problematic includes mathematical principles and algorithms, which ensure a process of number plate detection, processes of proper characters segmentation, normalization and recognition. Work comparatively deals with methods achieving invariance of systems towards image skew, translations and various light conditions during the capture. Work also contains an implementation of a demonstration model, which is able to proceed these functions over a set of snapshots.

1 INTRODUCTION

Massive integration of information technologies into all aspects of modern life caused demand for processing vehicles as conceptual resources in information systems. Because a standalone information system without any data has no sense, there was also a need to transform information about vehicles between the reality and information systems. This can be achieved by a human agent, or by special intelligent equipment which is be able to recognize vehicles by their number plates in a real environment and reflect it into conceptual resources. Because of this, various recognition techniques have been developed and number plate recognition systems are today used in various traffic and security applications, such as parking, access and border control, or tracking of stolen cars.

In parking, number plates are used to calculate duration of the parking. When a vehicle enters an input gate, number plate is automatically recognized and stored in database. When a vehicle later exits the parking area through an output gate, number plate is recognized again and paired with the first-one stored in the database. The difference in time is used to calculate the parking fee. Automatic number plate recognition systems can be used in access control. For example, this technology is used in many companies to grant access only to vehicles of authorized personnel.

2 MATHEMATICAL ASPECTS OF NUMBER PLATE RECOGNITION SYSTEMS

In most cases, vehicles are identified by their number plates, which are easily readable for humans, but not for machines. For machine, a number plate is 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. Because of this, it is necessary to design robust mathematical machinery, which will be able to extract semantics from spatial domain of the captured image. These functions are implemented in so-called “ANPR systems”, where the acronym “ANPR” stands for an “Automatic Number Plate Recognition”. ANPR system means transformation of data between the real environment and information systems.

The design of ANPR systems is a field of research in artificial intelligence, machine vision, pattern recognition and neural networks. Because of this, the main goal of this project is to study algorithmic and mathematical principles of automatic number plate recognition systems.

The problematic of number plate area detection includes algorithms, which are able to detect a rectangular area of the number plate in original image. Humans define the number plate in a natural language as a “*small plastic or metal plate attached*

Martinsky O. (2008).
RECOGNITION OF VEHICLE NUMBER PLATES.
In *Proceedings of the Tenth International Conference on Enterprise Information Systems - AIDSS*, pages 136-140
DOI: 10.5220/0001673801360140

to a vehicle for official identification purposes”, but machines do not understand this definition. Because of this, there is a need to find an alternative definition of the number plate based on descriptors, which will be comprehensible for machines.

The next step in recognition process is the character segmentation. In most cases, characters are segmented using the horizontal projection of a pre-processed number plate, but sometimes these principles can fail, especially if detected number plates are too warped or skewed. Then, more sophisticated segmentation algorithms must be used.

Detected characters can be normalized in many ways. There are various normalization and detection methods. At first, character dimensions and brightness must be normalized to ensure invariance towards a size and light conditions. Then, a feature extraction algorithm must be applied on a character to filter irrelevant data. It is necessary to extract features, which will be invariant towards character deformations, used font style etc.

Characters can be classified and recognized by the simple nearest neighbor algorithm (1NN) applied to a vector of extracted features, or there is also possibility to use one of the more sophisticated classification methods, such as feed-forward or Hopfield neural networks. Additional heuristic analyses can be used for elimination of non-character elements from the plate.

Sometimes, the recognition process may fail and the detected plate can contain errors. Some of these errors can be detected by a syntactical analysis of the recognized plate. If we have a regular expression, or a rule how to evaluate a country-specific license plate, we can reconstruct defective plates using this rule. For example, a number zero “0” can be automatically repaired to a character “O” on positions, where numbers are not allowed.

3 PRINCIPLE OF NUMBER PLATE AREA DETECTION

The first step in a process of automatic number plate recognition is a detection of a number plate area. This problematic includes algorithms that are able to detect a rectangular area of the number plate in an original image.

Let us 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.

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.

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

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

Consequently, we compute a horizontal and vertical projection for axes x and y:

$$
p_x(x) = \sum_{j=0}^{y_m-1} f'(x,j)\ ;\quad p_y(y) = \sum_{i=0}^{x_m-1} f'(i,y) \quad (2)
$$

We can detect the rectangular area of number plate by an analysis of projection p_x and p_y. In praxis, the analysis is proceeded iteratively in multiple steps. Each step refines the detected area. If there are multiple candidates for number plate, a heuristic analysis is used to determine the correct one. We assume only one step of iterative process:

$$
\begin{aligned}
&x' = \arg\max_{0\le i<x_m}\{p_x(i)\} \quad y' = \arg\max_{0\le j<y_m}\{p_y(j)\} \\
&x_{left} = \max\{i \mid 0 \le i < x' \wedge p_x(i) \le 0.3 p_x(x')\} \\
&x_{right} = \min\{i \mid x' \le i < x_{\max} \wedge p_x(i) \le 0.3 p_x(x')\} \\
&y_{top} = \max\{j \mid 0 \le j < y' \wedge p_y(j) \le 0.3 p_y(y')\} \\
&y_{bottom} = \min\{j \mid y' \le j < y_{\max} \wedge p_y(j) \le 0.3 p_y(y')\}
\end{aligned}
\quad (3)
$$

4 DESKEWING MECHANISM

The captured rectangular plate can be rotated and skewed in many ways due to the positioning of vehicle towards the camera. Since the skew significantly degrades the recognition abilities, it is important to implement additional mechanisms, which are able to detect and correct skewed plates.

The fundamental problem of this mechanism is to determine an angle, under which the plate is skewed. Then, deskewing of so evaluated plate can be realized by a trivial affine transformation.
The amount of skew can be evaluated using the Hough transformation of evaluated image. Hough transform is a special operation, which is used to extract features of a specific shape within a picture. It is important to know, that Hough transform does not distinguish between the concepts such as “rotation” and “shear”. The Hough transform can be used only to compute an approximate angle of image in a two-dimensional domain.

Let $f(x,y)$ be a continuous function. For each point $[a,b]$ in Hough space, there is a line in the “XY” coordinate system. We compute a magnitude of point $[a,b]$ as a summary of all points in the “XY” space, which lie on the line $a \cdot x + b$.

Although the space defined by a coordinate system is always discrete digital computers, we will assume that it is continuous. Generally, we can define the Hough transform $h(a,b)$ of a continuous function $f(x,y)$ in the unified coordinate system as:

$$h(a,b) = \int f(x, a \cdot x + b)\,dx \tag{4}$$

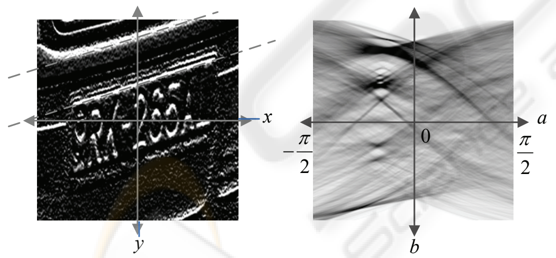

Figure 1: (left) Number plate in the unified “XY” coordinate system after application of the horizontal edge detection filter (right) Hough transform of the number plate in the “θB” coordinate system.

We use the Hough transform of certain image to evaluate its skew angle. You can see the Hough transform on the figure 1b. The pixels with a relatively high value are marked by a black color. Each such pixel corresponds to a long white line in figure 1a. If we assume that the angle of such lines determines the overall angle, we can find the longest line as:

$$(a_m, b_m) = \arg \max_{\substack{0 \le a \le 1 \\ 0 \le b \le 1}} \{h(a,b)\} \tag{5}$$

Then, the overall angle θ of image can be computed as $\theta = \arctan(a_m)$.

5 PRINCIPLE OF CHARACTER SEGMENTATION

After the detection of number plate area, there is a need to segment it into individual characters. Let this area to be described by a discrete function $f_p(x,y)$ before, or $f'_p(x,y)$ after the application of adaptive thresholding.

If characters are not merged together, we can segment them by the horizontal projection p_x of the function f'_p. Let S_x to be a set of all points on the x axis, in which the segmentation is possible: $S_x = \{x \mid p_x(x) > 0.3 \max\{p_x(x)\}\}$. Then, the peaks in horizontal projection p_x represent the spaces between characters. Also, there is necessary to ensure the following condition for the set S_x: The spacing between characters cannot be less than a half of expected character width:

$$\neg\left(\exists x_1, x_2 \in S_x : x_1 \ne x_2 \wedge |x_1 - x_2| < w/2\right) \tag{6}$$

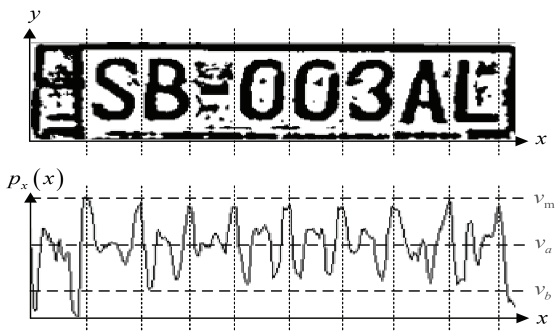

Figure 2: (top) Number plate after application of the adaptive thresholding. (bottom) Horizontal projection of plate with detected peaks. Detected peaks are denoted by dotted vertical lines.

6 FEATURE EXTRACTION AND PROCESSING BY NEURAL NETWORK

There is also a need to extract feature descriptors from segmented characters. As an extraction method significantly affects the quality of whole OCR (optical character recognition) process, it is very important to choose descriptors, which are invariant towards deformations of characters caused by a skew of image. In the first phase of algorithm, we normalize dimensions of character to $m \times n$ pixels by a bilinear interpolation. In the second phase we apply the Skeletonization algorithm to extract the skeleton of resampled character. Then we divide it into several regions (for example consider six regions organized to three rows and two columns, fig 3b). Finally, we count the number of occurrences of individual edge type for each portion of character. Let $m' \times n'$ to be a size of specific portion of character. We denote the number of occurrences of specific type of edge for this portion as a cardinality of the set of coordinates (x, y), in which the pattern H (corresponding to the specific type of edge) occurs:

$$\left| \left\{ \begin{array}{r} (x,y) \,|\, 0 \le x < m'-1 \wedge 0 \le y < n'-1 \wedge \\ \begin{pmatrix} f(x,y) & f(x+1,y) \\ f(x,y+1) & f(x+1,y+1) \end{pmatrix} = H \end{array} \right\} \right| \quad (7)$$

where $H_v = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$, $H_h = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}$, $H_{d1} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $H_{d2} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ are matrices corresponding to patterns for vertical, horizontal and diagonal edges.

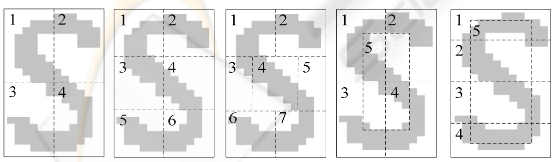

Figure 3: Layouts of regions in the character bitmap. The regions can be disjunctive as well as they can overlap each other.

Then, the vector of so chosen descriptors is twenty-four dimensional ($m = 24$, six regions, each with four edge types). This vector is directly mapped to an input layer of the three-layer backpropagation neural network. The middle layer contains scalable number of neurons (typically $n = 14 \ldots 20$). The output layer of neural network contains $o = 36$ neurons, each neuron for one character from alphabet $(A \ldots Z, 0 \ldots 9)$.

The threshold coefficients of neurons on the middle and output layer are marked as τ_i, or ϑ_i respectively. The weights of synaptical connections for k^{th} layer are marked as $w^k{}_{i,j}$.

The propagation of data in NN is performed according to computational functions of neuron $z_i = g\left(\tau_i + \sum_{j=1}^{m} \left(w_{i,j}^{(1)} \cdot x_j \right) \right)$ and $y_i = g\left(\vartheta_i + \sum_{j=1}^{n} \left(w_{i,j}^{(2)} \cdot z_j \right) \right)$ for the second and third layer, where g is a sigmoidal activation function $g(\xi) = \frac{1-e^{-\xi}}{1+e^{-\xi}}$. The output of network is represented by a vector y, which determines the recognized character i_r of alphabet $i_r = \arg \max_{0 \le i \le Z} \{y_i\}$.

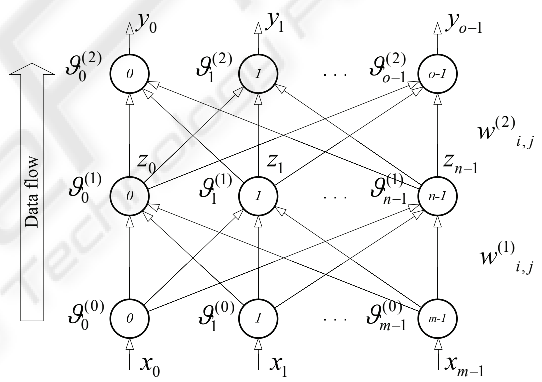

Figure 4: Architecture of the three layer feed-forward neural network.

7 CONCLUSIONS

This project deals with algorithmic and mathematical aspects of the automatic number plate recognition systems, such as problematic of machine vision, pattern recognition, OCR and neural networks. This paper is based on my thesis (http://www.fit.vutbr.cz/study/DP/rpfile.php?id=302), which comparatively describes all implemented principles and contains demonstration ANPR software which demonstrates all described algorithms. This project meets two goals. The first goal is to provide basic know-how for readers and allow them to understand these principles in detail. This work should help readers to implement generally similar system for character recognition. This thesis interconnects theory and praxis, and permits readers to experiment with described principles in the demonstration model written in Java[TM]. This was the second goal of this project. (for more information, see project web site at http://javaanpr.sf.net)

ANPR solution has been tested on static snapshots of vehicles, which has been divided into several sets according to difficultness. Sets of blurry and skewed snapshots give worse recognition rates than a set of snapshots, which has been captured clearly. The objective of the tests was not to find a one hundred percent recognizable set of snapshots, but to test the invariance of the algorithms on random snapshots systematically classified to the sets according to their properties. The average recognition rate was about 80%. The system achieves this recognition rate over a set of static snapshots.

The recognition ability of this system can be rapidly increased by processing data from video sequences. The short video sequence contains several frames of incoming vehicle. The similarity between these frames makes whole video sequence very redundant. This redundancy can be used to increase recognition abilities. If the system has 80% chance to successfully recognize one frame, then the overall recognition rate can be up to 99%, if we process several consequent frames from the video sequence.

Since processing of video sequence is a difficult computational operation, it is necessary to use an appropriate hardware and software platform. The combination of C-language and DSP processor should be suitable.

Currently I am developing a modified version of this algorithm for a practical security application in Brno in cooperation with Faculty of Information Technology.

Table 1: Recognition rates of the ANPR system (static snapshots).

.	Number of plates	Weighted score
Clear plates	68	87.2
Blurred plates	52	46.87
Skewed plates	40	51.64
Average plates	177	73.02

Table 2: Algorithm complexity and system throughput on AMD Opteron[TM] (1 GHz) processor architecture.

Snapshot resolution	Average time
320x240	0.74 s
640x480	1.17 s
720x576	1.40 s
800x600	1.46 s

ACKNOWLEDGEMENTS

JavaANPR is a part of the research plan “Security-oriented research in information technology, MSM 0021630528” at Brno University of Technology.

REFERENCES

Gonzalez, R., Woods R., 2002. *Digital Image Processing*, Prentice Hall, Upper Saddle River, New Jersey.

Shapiro, V., Dimov, D., Bonchev, S., Velichkov, V., Gluhchev, G., 2004. *Adaptive License Plate Image Extraction*, International Conference Computer Systems and Technologies, Rousse, Bulgaria.

Zhang, Y., Zhang, C., 2003. *New Algorithm for Character Segmentation of License Plate*, Intelligent Vehicles Symposium, IEEE.

Fukunaga, K., 1990. *Introduction to statistical pattern recognition*, Academic Press, San Diego, USA.

Kuba M., 1995. *Neural Networks*, Faculty of Informatics, Masaryk University, Brno, Czech Republic.

Kvasnicka, V., Benuskova, L., Pospichal, J., Farkas I., Tino, P., Kral, A. *Introduction to Neural Networks*, Technical University, Kosice, Slovak Republic.

Minsky, M., Papert, S., 1969. *Perceptons. An Introduction to Computational Geometry*, MIT Press:. Cambridge, Massachusetts.

Smagt, P., 1990. *Comparative study of neural network algorithms applied to optical character recognition*, International conference on Industrial and engineering applications of artificial intelligence and expert systems, Charleston, South Carolina, USA.

Srivastava, R., 1991. *Transformations and distortions tolerant recognition of numerals using neural networks*, ACM Annual Computer Science Conference, San Antonio, Texas, USA.