Recognition of Car License Plates using a neocognitron type of Artificial Neural Network

B.Th.M. Cornet and L.J.M. Rothkrantz

Delft University of Technology
Faculty of Information Technology and Systems
Zuidplantsoen 4, 2628 BZ Delft, the Netherlands
bas.cornet@ect.nl, L.J.M.Rothkrantz@cs.tudelft.nl

Abstract: In this paper, we describe the application of a combined neocognitron type of neural network classifier in a generic Car License Plate Recognition (CLPR) system. The suggested system contains an image processor, a segment processor and five coupled neocognitron network classifiers that act as a character recognizer. The presented model of the system depends neither on specific license plate image features nor on license plates character style and size. Combining neocognitron classifiers was motivated by the fact that manually tuning a training set for a large neocognitron network is tedious. It is shown how training set tuning for a large neocognitron network can be avoided. By connecting small neocognitrons specifically trained on ambiguous character classes, the performance of the recognizer in our CLPR was improved easily. The use of a neocognitron recognizer contributes significantly to the generality of a CLPR system. Besides, character recognition rates of 94% are realized using the proposed neocognitron.

Keywords: *Neocognitron neural network, image processing, license plate recognizer system*

1. Introduction

In the last years, many industrial companies have been involved in system development on automated vehicle identification. The recognition rates achieved by these systems vary from 40% in a commercial application, towards 86% for the VIPUR system developed by Nijhuis[15].
At Delft University there is a project running aimed at automated recognition of registration identification on various objects depicted on digital photographs. As a first application, a CLPR prototype has been developed based on a generic model for character recognition from digital photographs.

Many approaches have been proposed by researchers in the CLPR-field. For numberplate segmentation and character isolation Nijhuis [15] uses grayness and texture features, for character recognition combined feature-based neural networks are used. Barroso [1] searches in the image scan lines for certain grey-level variation 'signatures' that indicate the presence of the numberplate at the scan line under consideration, for character recognition template matchers are used. In this paper, we introduce some generic techniques for segmentation and character recognition that can be used in any character recognition application.

Our CLPR system consists of two major modules: an image-preprocessor and a character recognizer. Basically, only two problems are to be solved to recover the car's license plate from a digital image. First the individual license plates characters are to be isolated from the image and secondly the isolated segments are to be classified into alphanumerical characters.

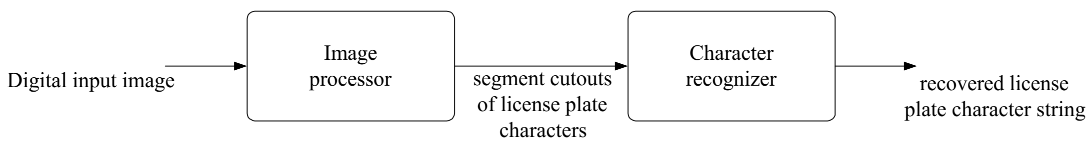

Fig. 1. The car license plate recognition systems block-diagram

License characters can not be simple isolated by searching certain ranges of intensity values of the pixels due to varying illumination conditions. Isolating license plates based on location, size or physical dimension of the license plate is not a practical solution either because, these measures will be typically unknown due to variable camera positions. There are described many solutions for license plate segmentation[1][15], all based on a number of license plates features. We introduce a license plate character isolator based on only two basic features: First, the color of a numberplate is significantly different from the object it is attached to (the car's chassis). Secondly, the background color of license plate and the color of the characters reproduced on it are significantly different.
The recognizer module of the CLPR should classify characters invariant of style, location and size. Remember that no explicit knowledge of the numberplate features is used. Furthermore the number plate characters as isolated from the image, may be deformed highly because of unsharp image recordings, varying camera positions, dirty plates or bad illumination. Most of the CLPR's described in the literature [15][10][16][7] use MLP neural networks with direct or feature input as character recognizers. The neocognitron neural network will act as the character recognizer in our CLPR.

In this paper a model for an image vision system is presented that is capable of recognizing car license plates independent from plate location, size, dimension, color and character style. From this model, a software system has been implemented using C/C++ that runs on a Personal Computer. Based on a license plate model, we have implemented an image processor to isolate individual license plate characters from a digital image. For character recognition, we have implemented the neocognitron network as proposed by Fukisuma [5]. Our motivation to use this type of artificial neural network was because of its capability to read generalized character styles independent from size and location. The characteristics of the neocognitron neural network functional features do fully match the requirements of a character recognizer in a CLPR. Moreover this type of network makes many image processing steps superfluous and therefore makes the system concept more general applicable.

Some years ago CLPR system technology received renewed interest from the Dutch Government because of a political resolution to introduce automated tollgates around the four major cities. Commercial CLPR system applications are widely used for traffic law enforcement however the number of photos to be processed by these systems is low compared to the numerous photos that will be produced by the proposed tollgate cameras. Manual system's recognition verification is surely unwanted in an Electronic Road Pricing system. Therefor the availability of an accurate, robust and reliable CLPR system is mandatory to successfully introduce ERP systems. Our CLPR model and prototype add some techniques towards such required high-performing systems.

2. Neocognitron

2.1 Network structure

The basic function of the neocognitron neural network is to act as a pattern recognition network. The neocognitron is mainly used to classify spatial binary character images into particular characters. The neocognitron is a large and complex artificial neural network. It is not only complex in structure and functionality but the neocognitron is also complex computational wise. Even a simple number recognizer will have millions of connections. The neocognitron has been described comprehensively in many textbooks [13] and many papers [5][12]. Because describing the neocognitron in short but formal matter is beyond the scope of this paper, only a brief description of the main characteristics and functionality of the neocognitron is given. The neocognitron is typically a hierarchical network, composed as cascaded connected layers. Neocognitron layers on their turn are composed of one or more planes. A plane is built as a two-dimensional array of cells. In figure 2, the hierarchical structure of the neocognitron is depicted.

The cells used in neocognitron networks are fundamentally simple objects. In the neocognitron network three types of cells are defined: S-cells, C-cells and V-cells. Each type of cell is assigned to perform a typical function within the neocognitron.

- The S-cells in the neocognitron network are the feature extracting cells. The excitation of a particular cell will be high only if a specific input pattern is presented (a particular feature) to its inputs. Which feature an S-cell should detect is determined by training. During training the input connection weights of the cell are reinforced according to certain rules.
- The V-cells are inhibitory cells; they prevent S-cells to excitate on input features that are combinations of the specific features mentioned in the previous paragraph. Their input connection weights are fixed; their output connection weights are reinforced by training.

- The C-cells have an important task to make the network tolerant for small pattern distortions and deformations and translations of patterns. C-cells blur the excitation patterns of the S-cell.

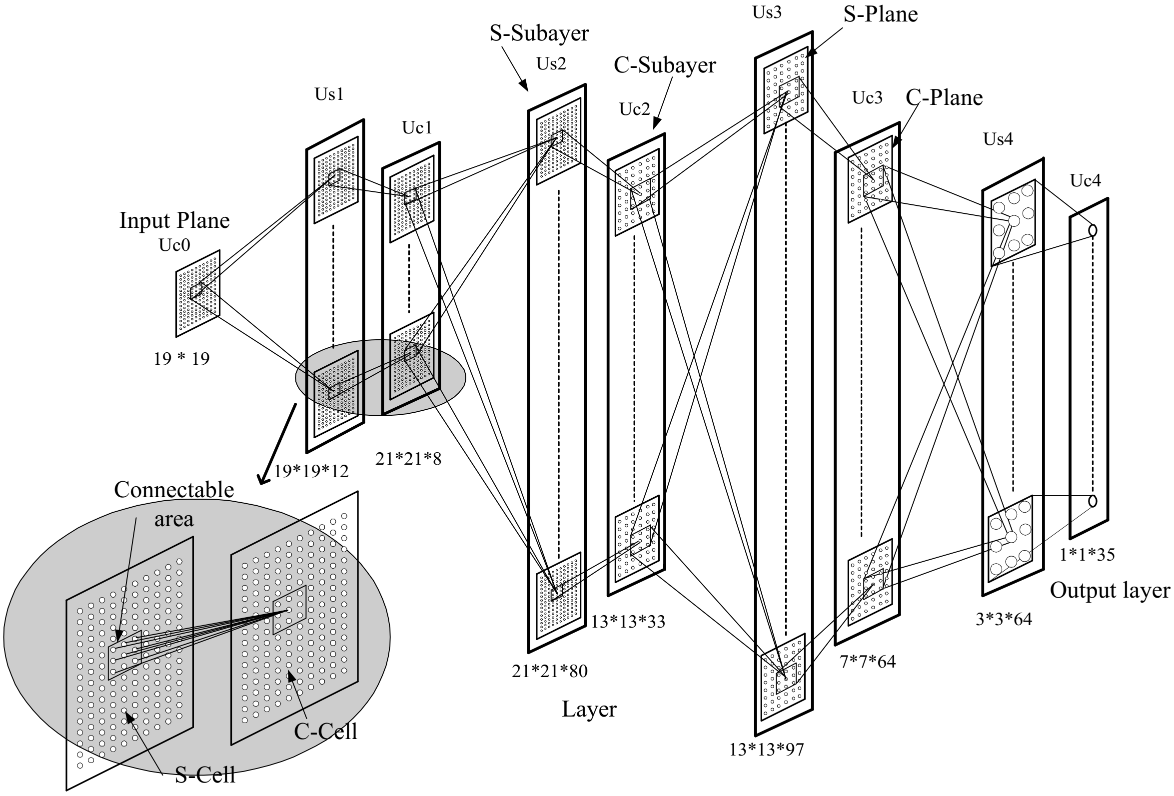

Fig. 2. The hierarchical structure of the neocognitron

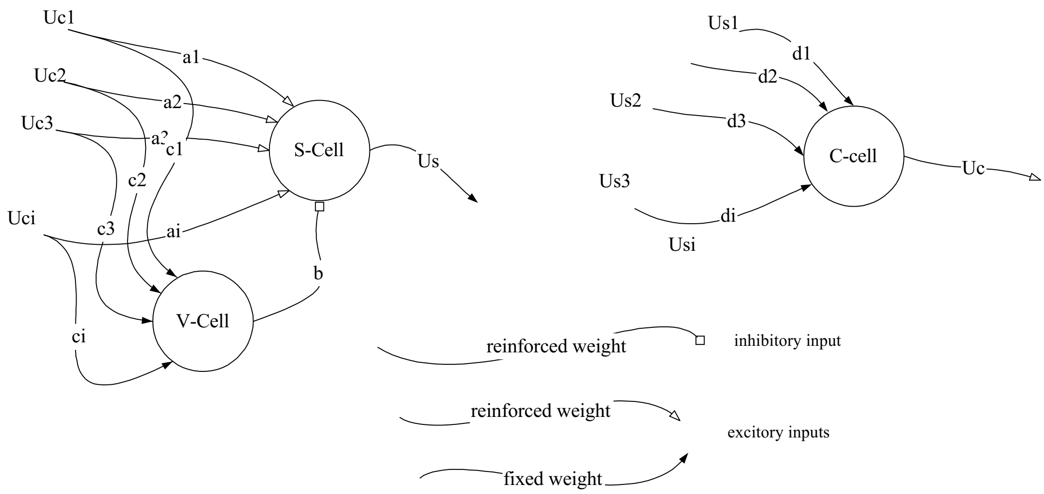

Fig. 3. Neocognitron type cells with cell weights *a,b,c* and *d* and cell excitation values *Us* and *Uc*.

Each cell takes input from several cells of planes in a previous (sub) layer. The source neurons are grouped within a square area. This area is referred to as the connectable area of the cell considered. The size of the connectable area will typically vary per sub-layer. Figure 2 shows the connectable area of 3*3 in the source plane for the center cell in the target plane.

2.2 Network Training

The neocognitron network can be trained to classify its input sample images using a supervised method. The supervised training approach involves the use of direct inputs to the S-plane cells in each sublayer of the network. Typically, the network layers are trained from simple basic input features in the first layer towards typical sample images in the last layer of the network. These typical samples used for training the last layer are the patterns that the neocognitron is designed for to recognize. The layers number one through four in the network depicted in figure 2 are trained respectively on the following characteristic features.

1. Simple and small straight line components.
2. Local features of characters like corners, curvatures, intersections and line endpoints.
3. Global features composed of local features of the previous layer that form complete characters or major characteristic parts of a specific character.
4. Typical samples of alphanumerical characters.

2.3 Neocognitron network used as recognizer in the CLPR

Fukisuma[5] has proposed a network configuration and training set patterns for a neocognitron that classifies handwritten alphanumerical characters. This network exhibits a very poor recognition rate on a typical set of characters with a font in use on Dutch license plates. The poor recognition rate is due to the different style between handwritten characters and license plate characters of Dutch numberplates. Furthermore, the skeletonization operation required has a negative impact on the recognition rate. Remember the neocognitron used is designed to recognize features that have only line segments of one pixel thick, therefore a thinning operation is applied on the segment before the segment image is supplied to the recognizer. We have used the Hillditch algorithm as described by Gonzales in [6] to skeletonise the segments. This algorithm does deform some characters image highly, typically the character symbols 'K' ,'N' and 'X'.

Redesigning the configuration of the network and the training set increased the recognition performance. As stated by Fukisuma, designing a network and finding an appropriate training set is not a trivial task. Specifically finding the training set requires hard labor increasing with the number of characters to be classified. The number of output cells is limited to the 27 characters that are used on Dutch license plates instead of the original 35 output cells as specifified in [5]. Because characters with a fixed font show much more regularities, a three-layer network seemed sufficient instead of the four-layer network originally used. Furthermore, the fact that the neocognitron can classify images independent from size has not been exploited to its full extend. For the neocognitron used in our system, we require only characters with heights in the range from 15 to 19 pixels to be recognized. Character segments isolated from the image with a height not in this range are automatically resized by the segment processor. The reduction in network size and the fact that we do not require the neocognitron to classify a broad range of character image sizes makes the procedure to find a suitable training set easier.

In figure 4 the training set for all three layers of our network configuration is given. In the text below r denotes the selectivity parameter. This parameter determines the measure of inhibition by the V-cells in the network. The γ coefficient determines the strength of the fixed connections between the C-cells and the V-cells. The coefficients δ and δ' determine the strength of the fixed connections between the S-planes and the C-planes. This parameter determines the measure of 'blurring' that makes the network invariant for small positional shifts. The configuration of our 3-layer network is as follows:

- The input layer has 19*19 cells.
- First layer; 12 S-planes each having 19*19 cells. Connectable area of the S-cells in the first layer is a 3*3 area. 8 C-planes each having 21*21 cells and a connectable area of 3*3 cells. The network selectivity parameter r=1.70, γ=0.9, δ=0.9 and δ'=4.0 for all planes. Training set pattern size for the first layer is 3*3.
- The second layer has 86 S-planes with a size of 21*21 connected to 39 C-planes with a size of 13*13. The connectable area of the S-cells in this layer is 5*5, the connectable area of the C-cells in this layer is 7*7. The network selectivity parameter r=3.2 for all planes in this layer. γ=0.9, δ=0.8 and δ'=4.0 for all planes. The neurongap parameter for this layer is 2. Training set pattern size for the second layer is 9*9.
- The third layer of our network has 55 S-planes of size 11*11 and 27 output C-cells. The connectable area for the S-cells in this layer is 11*11. The connectable area of the output cells in the C-plane of this layer is 5*5. The network parameters for this layer are set to the following values: selectivity parameter r=1.4, γ=0.9, δ=0.7 and δ'=1.4 for all planes. The neurongap parameter for this layer is 2. The third layer is trained using samples of 19*19 pixels.

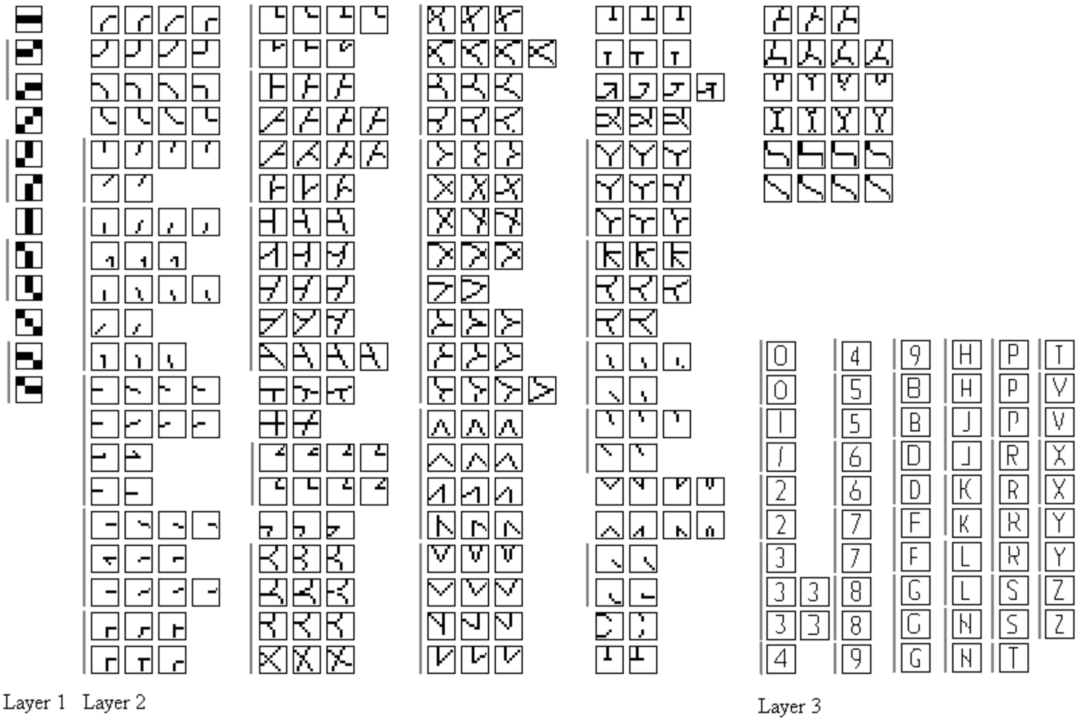

Fig. 4. Training set patterns for the neocognitron used within our CLPR system. The vertical lines on the left of the training set samples indicate that the planes are joined onto one plane in the next sublayer.

In figure 5, the recognition rates are shown of characters with the font in use on Dutch license plates using Fukisuma's original network[5] and the network configuration used in our prototype CLPR. The character samples used in this test are reproduced from 189 original high resolution digital photo images taken from Dutch license plates. The character samples used do not contain much noise and show almost no deformation. Obviously, the recognition rate on alphanumerical characters of Dutch license plates has increased significantly. When verifying the newly configured network on real-life photographs we observe slightly lower recognition rate because the character image segments from these photographs show more irregularities, deformation and noise.

As with many other character image recognizers we observe that some character images are hard to distinct from each other. Typical examples are the numerical '0' and the alpha character 'O' or the numerical '1' and the alpha character 'I'. Neither the character 'I' nor 'O' are used in the Dutch license plate alphabet, but in our case relatively much misclassifications between 0/D, 5/S, 8/B and 2/Z were observed. In the next section, we propose a configuration of combined neocognitron classifiers to eliminate these errors in recognition.

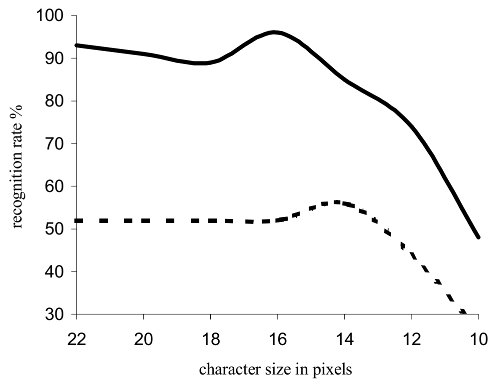

Fig. 5. Recognition rate versus character size observed on skeletonized character images with a font in use on Dutch License Plates. ---- Fukisuma's network designed to classify handwritten characters [5], —— this network approach tuned specifically on Dutch license plate characters

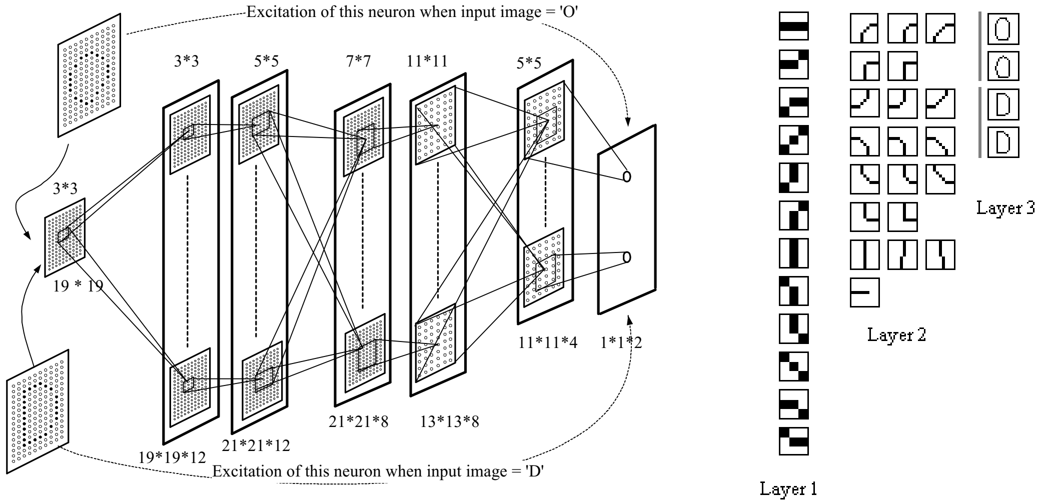

Fig. 6. The O/D discriminator neocognitron configuration layout and it’s training set

2.4 Model for combining neocognitron classifiers

Tuning the training set of the neocognitron such that misclassifications between similar character like 0/D or 2/Z are prevented seemed very hard to model. If we tune the recognition rate of ‘0’ and ‘D’ by adjusting the training set patterns in layer 2, a severe decrease in recognition rate on other alphanumerical characters like for example “8”,”R” or “P” was observed. This is because specific planes in layer 2 do not only determine the recognition of 0/D but also may contribute significantly to the classification of the “8”,”R” or “P”. As stated earlier, finding a good training set requires hard labor. Defining a training set for a neocognitron that only has to classify into two alphanumerical characters is relatively easy. Such a network has typically only three layers and a limited number of planes. With this type of network, the training set can even be tuned by studying individual cell excitations on certain training- or test samples. Manually tuning a training set on the 3-layer network designed to recognize 27 different characters of various styles is a very tedious job.

In figure 6 the neocognitron network layout and training set is depicted which is used within our CLPR to decrease the misclassifications between the ambiguous characters ‘0’ and ‘D’. In figure 7, a block diagram is given how four specific neocognitrons are connected to the “main” neocognitron network in our CLPR prototype setup. The suggested combined neocognitron networks have a recognition rate (correctly classified segments) of 94 % on a set of 726 character images isolated from real-world photographs.

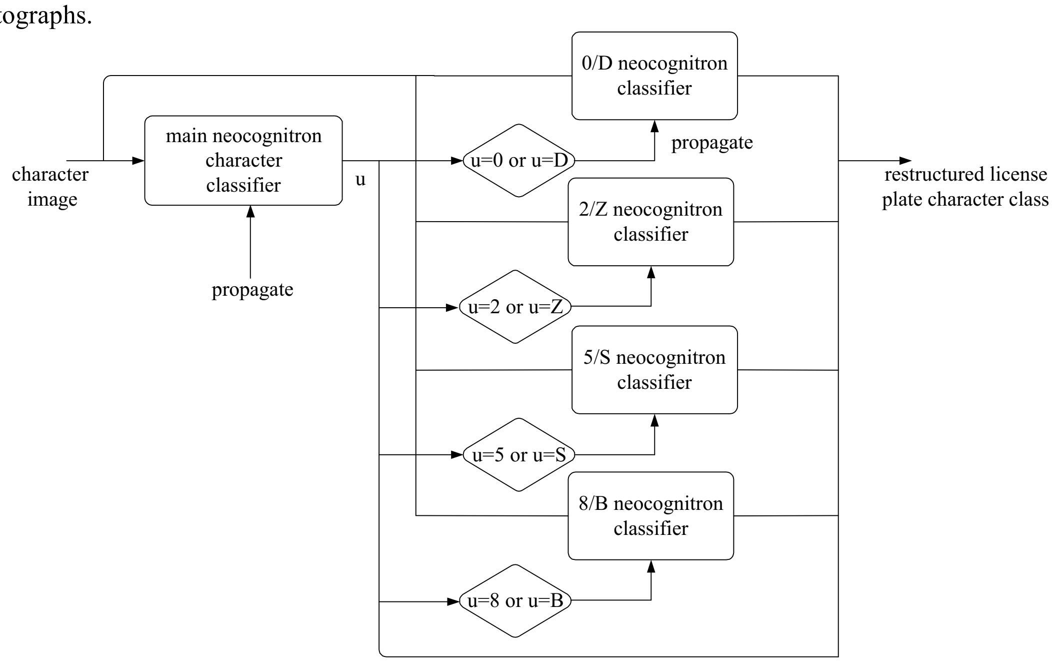

Fig. 7. Combined neocognitron networks in the CLPR application

3. Description of the system

3.1 Architecture

In figure 8 the system architecture diagram is given. The system contains three major software modules. The image processor, the segment processor and the combined neocognitron network. Within the userinterface, an output formatter is included . The output formatter does not really contribute to the image understanding process, it only joins the results of the segment processor and the neocognitron network into the output file of the system.

The *image processor* biniarizes the original images such that all possible character images form connected black pixel regions. This is accomplished by a number of basic image transforms. First the original image is gray-scaled to improve image contrast, and then the image is passed through a low-pass filter to reduce noise. Next, a local biniarization method is applied. The biniarization method, which was found experimentally, whitens the numberplate's background, blackens the numberplate's characters and encloses the numberplate by a connected black boundary region. This method was based on the two basic license plate features mention in section 1. Finally, by applying two flood-filling procedures on the biniarized image the input image for the segment processor is obtained.

The *segment processor* determines the location and size of all connected black pixel regions. It successively cuts out the segments from the original image. The cutouts are either enlarged or reduced if the height less then 15 or greater then 19 pixels. Next the segment processor adaptively biniarizes the resized segments. Finally the segment processor applies a thinning procedure on the segments and pastes the thinned character image sample onto the 19*19 pixel input plane of the combined neocognitron recognizer.

The *combined neocognitron recognizer* classifies the input sample produced by the segment processor by propagating its input plane signals to the output cells of the network. The main neocognitron returns the character class belonging to the output cell that has the highest excitation value of the 27 output cells. If the class happens to be in the set {0, 2, 5, 8, D, Z, S, B}, one of the specifically trained neocognitrons is triggered to re-recognize the input sample. The latest network finally determines to which character class the input sample image belongs. Input samples that do not excitate any output cell of the main neocognitron are marked by a question mark. These character images are considered rejected characters.

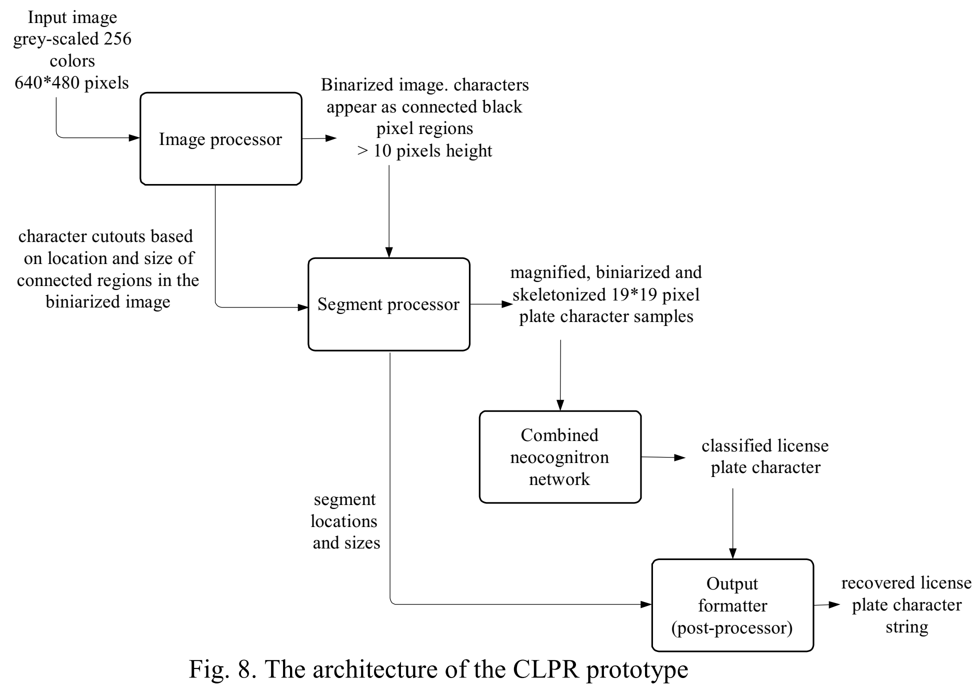

Fig. 8. The architecture of the CLPR prototype

Unfortunately, the image processor erroneously cuts out segments that are not license plates characters. Typically, the image processor isolates objects from the background scenery in the image that have the same characteristics as the license plate characters. In most cases, the recognizer will reject these segments. A future post-processor should delete all these non license plate character segments. Regrettably, this is to be accomplished by using the location and size data of the segment as saved by the segment processor. This introduces the use of some kind of explicit knowledge about plate characters. A post-processor is not implemented in the CLPR prototype, instead we included an output formatter to show the output of the CLPR in the current setup.

3.2 Implementation

The system prototype is modeled and implemented using Object Orientation. The target platform for the CLPR prototype is an Intel PC running WindowsNT. The image processor, the segment processor, and the neocognitron are implemented in ANSI C++. The output formatter module is included in the userinterface of the system and depends on WindowsNT. The neocognitron implementation in this system is derived from earlier work of Steuer [13]. We have used his original software design of a neocognitron simulator that ran on a parallel computer under the UNIX operating system. The runtime for the neocognitron needed to classify an input sample is about 2 seconds. The time needed to train the network using the configuration and training set Fukisuma suggested in [5] is about 20 minutes. The average time for recovering all license plate characters from a 640*480 8-bit grey-scaled image is 30 seconds. All times observed when using a PC equipped with a 1300 Mhz Pentium IV processor. The minimum height of a character that the CLPR system will handle is limited to 10 pixels.

3.3 Processing example

We conclude this section with a serie of figures as an illustration of the processing steps described above. The size of the license plate characters, of the Dutch numberplate on the example image, is 12 pixels. The results of the image processor steps are depicted in figure 9. The results of the segment processor steps are depicted in figure 10. Note that the original character cutouts are magnified from 12 pixels to 19 pixels by the segment processor. In figure 11, the response of the main neocognitron is given when recognizing the first ‘D’ character of the license plate. Obviously, this character is recognized as ‘0’ instead of ‘D’. The 0/D discriminator network however corrects this classification to ‘D’ as shown in figure 12. Finally, in figure 13 the output file produced by the output formatter is shown. In this file the original location and size of all segment cutouts are displayed including the character classes belonging to the 5 most highest output neuron excitation of the main neocognitron network. The asterisk in the line of segment number 2 indicates that the initial character classification by the main neocognitron has been corrected by an associated discriminator network.

Fig. 9. The image processor steps. Respectively the results of the following image transformations are displayed: original image, grey-scaling, filtering, biniarizing and floodfilling twice.

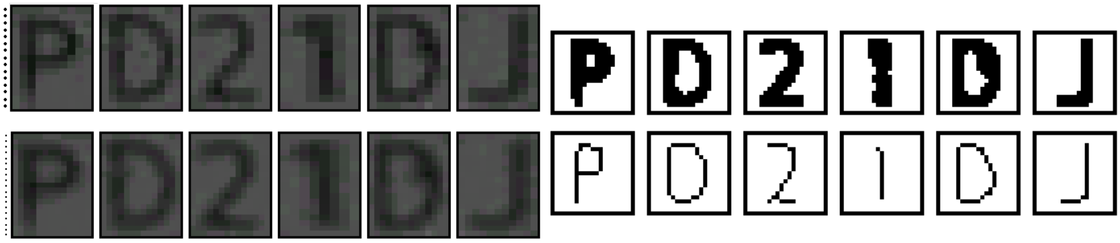

Fig. 10. The segment processor steps. Segment cutouts from the original image, based on the location of black connected pixel regions of the final image produced by the image processor, segment scaling, segment biniarization and segment thinning

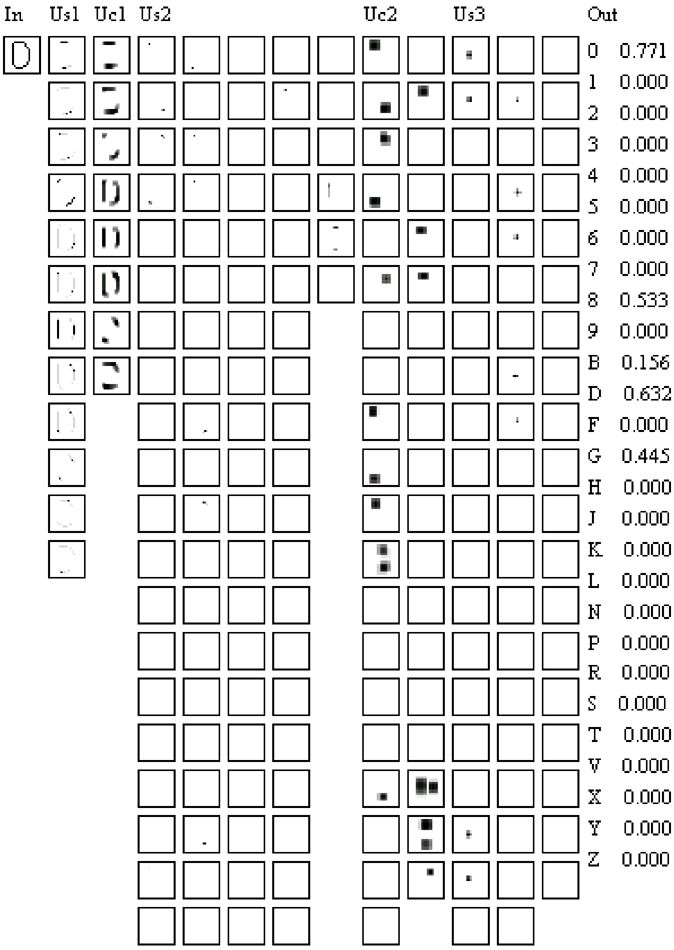

Fig. 11. Main neocognitron cell excitations

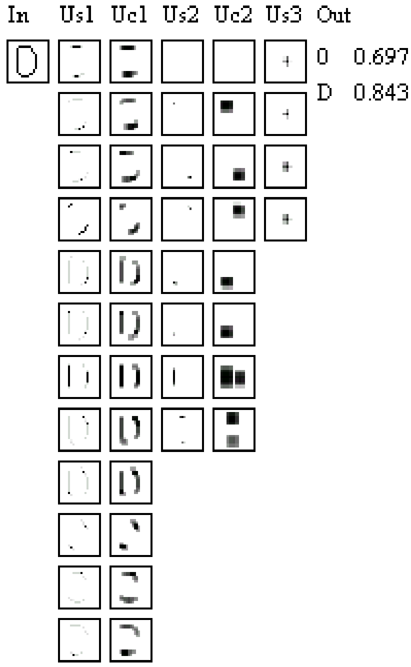

Fig. 12. 0/D neocognitron cell excitations

```
15:38:00  New Bitmap Loaded    File=C:\Afstudeer\pic\256graySM\PIC00028.bmp      Size=640*480
15:38:02  Bitmap Preprocessed using method I
15:38:04  Start recognizing picture segments
15:38:07  analysing segment num=[ 1]    loc=[322,331,113,127]          NeoCognitron Output=[PR000]
15:38:10  analysing segment num=[ 2]    loc=[332,341,113,126]          NeoCognitron Output=[D0G8B]
15:38:12  analysing segment num=[ 3]    loc=[348,357,113,126]          NeoCognitron Output=[2Z000]
15:38:15  analysing segment num=[ 4]    loc=[358,363,113,126]          NeoCognitron Output=[1VY00]
15:38:17  analysing segment num=[ 5]    loc=[370,379,113,126]          NeoCognitron Output=[D0000]
15:38:20  analysing segment num=[ 6]    loc=[380,387,113,126]          NeoCognitron Output=[J0000]
15:38:20  InValid segment num=[ 7]      loc=[194,204,291,301]          skipped
15:38:20  InValid segment num=[ 8]      loc=[187,197,298,311]          skipped
15:38:20  InValid segment num=[ 9]      loc=[182,187,305,315]          skipped
15:38:20  End recognizing picture segments
15:38:20  Processing segment list selecting plate candidate characters
15:38:20  segment num=[ 1]    loc=[323,330,114,126]         size=[ 13,  8]      pos=[  0,323]       recog=[PR000]
15:38:20  segment num=[ 2]    loc=[333,340,114,125]         size=[ 12,  8]      pos=[  0,333]       recog=[D0G8B]
15:38:20  segment num=[ 3]    loc=[349,356,114,125]         size=[ 12,  8]      pos=[  0,349]       recog=[2Z000]
15:38:20  segment num=[ 4]    loc=[359,362,114,125]         size=[ 12,  4]      pos=[  0,359]       recog=[1VY00]
15:38:20  segment num=[ 5]    loc=[371,378,114,125]         size=[ 12,  8]      pos=[  0,371]       recog=[D0000]
15:38:20  segment num=[ 6]    loc=[381,386,114,125]         size=[ 12,  6]      pos=[  0,381]       recog=[J0000]
15:38:20  No More Characters Found
```

Fig. 13. The output log file created by the output formatter

4. Evaluation

Fig. 14. A representative sample of the evaluation set used

For evaluation of our prototype system, the recognition success rate of the prototype has been measured by using a number of photographs of cars parked on the parking lots of the Delft University of Technology. Figure 14 above shows a representative sample of our evaluation set. The images used are 640*480 8-bit grey-scaled bitmaps. The license plate character heights vary from 10 to 17 pixels. The photos were taken under varying illumination conditions and variable camera positions.

In total 140 images were passed through the prototype CLPR. The image processor failed to create a suitable image for the segment processor in 19 cases. Plate characters were lost due to dirty plates, screws, pull-hooks or unsharp images or because the plates color was too similar to its background causing the biniarization operation fail to create a proper plate boundary.

The remaining 121 images could be processed fully by the CLPR prototype. The combined neocognitron network exhibits a recognition rate (correctly classified segments) on character level of 94 %, an error rate (misclassified segments) of 4% and a rejection rate (unclassified segments) of 2%, observed on a sample set with 726 characters derived from the photographs in the evaluation set.

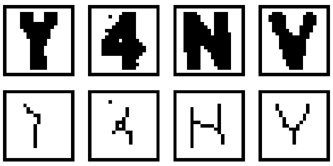

Fig. 15a. Recognizer failure due to thinning

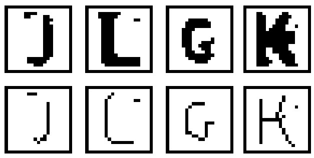

Fig. 15b. Genuine recognition failures

On plate level, we observed rates of 73 %, 16% and 11% respectively. Table 1 gives an overview on the recognition performance per character class. A brief analysis showed the failures of the combined neocognitron network fall in either one of three categories below:

1. Some alphanumerical characters in the set {0, 2, 5, 8, D, Z, S, B} are still wrongly classified despite the specific discriminator neocognitron.
2. The thinning operation on the segments seems not to be able to process certain input patterns correctly. In figure 15a, some typical examples are given of skeletonizer errors.
3. The rest of the failures are due to the neocognitron failure. Seemingly, the training set patterns for the network and/or its configuration were not sufficient to correctly classify the input samples. Figure 15b gives some examples of segments that belong to this category.

About 50 % of all wrongly classified images are due to skeletonizer character deformations. Taken this into account a character recognition rate of about 97% with the proposed neocognitron configuration is achievable.

Table 1. The neocognitron classifier confusion matrix

→ Recognized as
Input character

↓	0	1	2	3	4	5	6	7	8	9	B	D	F	G	H	J	K	L	N	P	R	S	T	V	X	Y	Z	?*	!†	#
0	16	-	-	-	-	-	-	-	-	-	-	**4**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		**4**	20
1	-	26	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			26
2	-	-	23	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**3**		**3**	26
3	1	-	-	25	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		2	27
4	-	-	-	-	17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1		18
5	-	-	-	-	-	25	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**1**	-	-	-	-	-		**1**	26
6	-	-	-	1	-	-	29	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		1	30
7	-	-	-	-	-	-	-	18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			18
8	-	-	-	-	-	-	-	-	24	-	**1**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3	**1**	28
9	-	-	-	-	-	-	-	-	-	22	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1		23
B	-	-	-	-	-	-	-	-	**2**	-	25	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	**2**	28
D	-	-	-	-	-	-	-	-	-	-	1	44	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		1	45
F	-	-	-	-	-	-	-	-	-	-	-	-	38	-	-	-	-	-	-	-	-	-	-	-	-	-	-			38
G	-	-	-	-	-	-	-	-	-	-	-	-	-	34	-	-	-	-	-	-	-	-	-	-	-	-	-	1		35
H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	31	-	-	-	-	-	-	-	-	-	-	-	-			31
J	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	17	-	-	-	-	-	-	-	-	-	-	-	1		18
K	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	8	-	-	-	1	-	-	-	-	-	-	1	1	10
L	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	28	-	-	-	-	-	-	-	-	-	1	1	29
N	-	-	-	-	-	-	-	-	-	-	-	-	-	-	5	-	-	-	28	-	-	-	-	-	-	-	-		5	33
P	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	26	-	-	-	-	-	-	-			26
R	-	-	-	-	-	-	-	-	-	-	-	1	-	-	1	-	-	-	-	-	36	-	-	-	-	-	-		2	38
S	-	-	-	-	-	**2**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	26	-	-	-	-	-		**2**	28
T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	26	-	-	-	-	1		27
V	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-	-	23	-	4	-		5	28
X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	22	2	-	1	2	25
Y	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	17	-	1		18
Z	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-	-	-	-	-	-	26		1	27
#	17	26	23	26	17	27	29	19	26	22	27	49	38	34	37	17	8	29	29	26	37	27	26	23	22	23	29	13	34	726

* ? = rejected characters
† ! = misclassified characters
Bold = ambiguous character misclassifications

5. Conclusion

In this paper, we have proposed an architecture for a generic CLPR system, without using explicit knowledge about character size, -color, -location and -style or plate dimensions. During processing we lose about 13% of the images in the image processor module due to segmentation failures. The character recognizer recovers 73 % of the numberplates of the cars depicted on the set of photographs used for system evaluation which past the image processor succesfully.
As stated in section 3 the system requires a post-processor module. Within this post-processor module, a syntax analyzer should be incorporated. In case of Dutch license plates we will gain much recognition performance on ambiguous characters like {0, 2, 5, 8, D, Z, S, B} by only applying some simple syntax forcing. Additional syntax forcing would yield a license plate recovery of about 82% on successfully segmentated images. The overall performance of the system can be easily pushed towards 90% by both increasing the image processor's performance and the recognizer's performance.

The image processor performance can be increased by:
- Developing a better local biniarization method.
- Tuning the segment validator to decrease the number of erroneously isolated segments.

The recognizer performance can be increased by:
- Modifying the segment thinning algorithm by taken the original gray-scales of the biniarized segment into account.
- Enlarging the neocognitron input plane to 38*38 pixels. Larger character image samples would suffer less from the thinning deformations demonstrated in figure 15a.
- Train the neocognitron in the unsupervised way to make it better classify ambiguous characters.
- Adding a post-processor to include syntax forcing and plate validation to reduced the error rate.

References

[1] Barroso J., Rafael A., Dagless E.L. and Bulas-Cruz J. "Number plate reading using computer vision" on: *http:/www.utad.pt/~jbarroso/html/isie97.html*

[2] Cornet B. (May 2001) "Recognition of car license plates using neocognitron type of artificial neural network", *master's thesis report* Delft University of Technology.

[3] Davies P., Emmott N., and Ayland N. (1990), "License Plate Recognition Technology for Toll Violation Enforcement," in: *Proceedings of IEEE Colloquium on image analysis for transport applications*, Vol. 35, pp. 7/1-7/5.

[4] Fohr R. and Raus M. (1994), "Automatisch lesen amtlicher Kfz-Kennzeichen," Elektronik No. 1, pp. 60-64.

[5] Fukisuma K. and Wake N. "Handwritten alphanumeric character recognition by the neocognitron", (1991) *IEEE Transactions on neural networks VOL 2. No 3. May 1991*, pp.355-365.

[6] Gonzales R.C. and Woods R.E. (1992) *Digital image processing*, Addison-Wesley, ISBN 0-201-5083-6.

[7] Hwang C., Shu S., Chen W., Chen Y. and Wen K. (1992), "A PC-based License Plate Reader." *Proceedings Machine Vision Applications, Architecture and Systems Integration*, Vol. SPIE1823 (Boston, MA), pp.272-283.

[8] Kanayama K., Fujikawa Y., Fujimoto K., and Horino M. (1991) "Development of vehicle-license number recognition using real time image processing and its application to travel time measurement," *Proceedings of the 41st IEEE Vehicular technology Conference (ST. Louis), PP. 798-804.*

[9] Kertesz A. Kertesz V. and Muller T. (1994) "An On-line Image Processing System for registration number identification*," Proceedings of IEEE International Conference on Neural Networks*, Vol. 6 (Orlando, FL), pp. 4145-4147.

[10] Lisa F., Carrabina J., Perez-Vincente C. Avellana N., and Valderrame E. (1993) "Two-bit Weights are enough to solve Vehicle License Plate Recognition Problem." , *Proceedings of the International Conference on Neural Networks*, Vol. 3 (San Francisco, CA), pp.1242-1246.

[11] Lotufo R.A., Morgan A.D., and Johnson A.S. (1994), "Automatic Number-plate Recognition," *Proceedings of the IEEE colloquium on Image Analysis for transport applications*, Vol. 35, pp.6/1-6/6.

[12] Shouno H., Fukisuma K. and Okada M. "Recognition of handwritten digits in the real world by a neocognitron", (1999) *Knowledge-based intelligent techniques in character recognition*, pp.17-47, ISBN 0-8493-9807-X.

[13] Steuer M. (September 1996) "Parallely Implemented Neocognitron-like Neural Network", *master's thesis report* Delft University of Technology.

[14] Tanabe K., Marubayashi E., Kawashima H. Nakanishi, T., and Shio, A.,(1994), "PC-based Car License Plate Reading," *Image and Video Processing*, Vol. SPIE-2182, pp.220-231.

[15] Terbrugge M.H. , Nijhuis J.A.G. , Spaanenburg L. and Stevens J.H. "License plate recognition", (1999) *Knowledge-based intelligent techniques in character recognition*, pp.263-296, ISBN 0-8493-9807-X.

[16] Yoo J.,Chun B., and Shin D. (1994), " A neural network for recognizing characters extracted from moving vehicles," *Proceedings of world congress on Neural networks*, Vol. 3 (San Diego, CA), pp. 162-166.

[17] Williams P.,Kirby H., Montgomery F., and Boyle R. (1989), "Evaluation of video-recognition equipment for number-plate matching", *Proceedings of the 2nd International Conference on Road Traffic Monitoring* (London U.K.), pp. 89-93.