APPROVAL PAGE FOR GRADUATE THESIS OR PROJECT

SUBMITTED IN PARTIAL FULFILLMENT OF REQUIREMENTS FOR
DEGREE OF MASTER OF SCIENCE AT CALIFORNIA STATE
UNIVERSITY, LOS ANGELES BY

ZhengHui Hu
Candidate
Computer Science
Field of Concentration

TITLE: AN AUTOMATIC LICENSE PLATE RECOGNITION SYSTEM
USING IMAGE PROCESSING AND NEURAL NETWORK

APPROVED: Eun-Young Kang, Ph.D.
Faculty Member                   Signature

Jiang Guo, Ph. D.
Faculty Member                   Signature

Raj S Pamula, Ph.D.
Department Chairperson           Signature

DATE: 03/19/2007
AN AUTOMATIC LICENSE PLATE RECOGNITION SYSTEM
USING IMAGE PROCESSING AND NEURAL NETWORK

A Thesis
Presented to
The Faculty of the Department of Computer Science
California State University, Los Angeles

In Partial Fulfillment
Of the Requirements for the Degree
Master of Science

By
ZhengHui Hu
March 2007
Acknowledgment

I would like to thank my lovely wife Jessie for all the love and support that she has provided me during the period of working on this thesis. In the many arduous and sleepless nights, she was always there by my side. No matter how difficult the situation became, I know I always had someone.

I would like to also thank God for a lovely gift that we received during this time. My first son Jonah Hu was born last year, and he has been a constant source of joy in our family.

I need to specially thank my advisor Dr. Eun-Young Elaine Kang for her constant guidance and supervision over my work. She has always lead me into the right direction when I was confused by my own thoughts. Without her assistance, I would not have been able to accomplish my goals for this project.

I sincerely thank Dr. Raj. Pamula (Chair) for helping me in every aspect during these years at CSULA computer science program. Thanks to Tricia for all the kindness and attention. All the CS faculties that taught me about every aspect of the discipline.

Last but not least, thanks to my parents for their understanding and encouragement at all time.
Abstract

A surveillance system that can automatically identify vehicles by reading their license plate number can be extremely useful for any publicly accessible infrastructure by providing a cheap and efficient way of monitoring all passing vehicle. Such surveillance system would work automatically with minimal human intervention, and can interconnect to other system once suspect vehicles are located.

However, there are many challenges in developing a reliable and efficient automatic License Plate Recognition System (LPRS). Being able to cope with images taken under different conditions, where lighting, size and location of the plate change has been a topic of many researches in computer vision and pattern recognition.

In this Thesis, innovative methods are proposed for license plate recognition that are targeted to solve the inherited issues. For plate localization, a heuristic combining several traditional image processing technique is used. Techniques such as histogram equalization, edge detection, filtering and component analysis each plays a role in the extraction process. For character segmentation, a interpolation algorithm based on information obtained by filtering and adaptive thresholding is used to separate each character. Artificial neural networks (ANN) are in charge of the Optical Character
Recognition. Traditional training method for ANN are extremely time consuming and often result in sub-optimal configurations. A hybrid training method is introduced by combining traditional gradient descent based Back-propagation and random selection based Simulated Annealing process to overcome such shortcomings.
# Table of Contents

Acknowledgment ............................................................................................................ ii  
Abstract ........................................................................................................................ iii  
List of Tables .................................................................................................................. vii  
List of Figures ................................................................................................................. viii  
1. Introduction ................................................................................................................ 1  
2. Previous Work ............................................................................................................. 6  
   2.1. Plate Localization .................................................................................................. 6  
   2.2. Character Segmentation ....................................................................................... 7  
   2.3. Character Recognition ......................................................................................... 8  
3. System Architecture .................................................................................................... 10  
   3.1. Plate Localization ............................................................................................... 11  
      3.1.1. Noise Filtering .............................................................................................. 13  
      3.1.2. Brightness Normalization .............................................................................. 14  
      3.1.3. Extract Edge Information ............................................................................ 15  
      3.1.4. Filtering ........................................................................................................ 16  
      3.1.5. Connected Component Analysis .................................................................. 17  
   3.2. Character Segmentation ....................................................................................... 18  
      3.2.1. Apply Color Filter ......................................................................................... 20  
      3.2.2. Thresholding ................................................................................................. 20
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.3. Characters Extraction</td>
<td>21</td>
</tr>
<tr>
<td>3.3. Character Recognition</td>
<td>22</td>
</tr>
<tr>
<td>3.3.1. Artificial Neural Networks</td>
<td>22</td>
</tr>
<tr>
<td>3.3.2. Simulated Annealing</td>
<td>27</td>
</tr>
<tr>
<td>3.3.3. Neural Network Classifier Implementation</td>
<td>28</td>
</tr>
<tr>
<td>4. Experimental Results</td>
<td>32</td>
</tr>
<tr>
<td>4.1. Settings</td>
<td>32</td>
</tr>
<tr>
<td>4.2. Plate Localization</td>
<td>32</td>
</tr>
<tr>
<td>4.3. Character Segmentation</td>
<td>33</td>
</tr>
<tr>
<td>4.4. Optical Character Recognition</td>
<td>34</td>
</tr>
<tr>
<td>4.5. Processing Time</td>
<td>35</td>
</tr>
<tr>
<td>5. Conclusion</td>
<td>36</td>
</tr>
<tr>
<td>References</td>
<td>38</td>
</tr>
<tr>
<td>Appendix A - Successful Images</td>
<td>41</td>
</tr>
<tr>
<td>Appendix B - Unsuccessful Images</td>
<td>45</td>
</tr>
</tbody>
</table>
List of Tables

Table 1: Plate Localization Success Rate ................................................................. 32
Table 2: Character Segmentation Success Rate ............................................................ 33
Table 3: OCR Recognition Rate ................................................................................... 34
List of Figures

Figure 1: Recognition Steps..............................................................6
Figure 2: Framed Plate .................................................................8
Figure 3: California License Plate....................................................11
Figure 4: Sample Input Image..........................................................12
Figure 5: Plate Location.................................................................13
Figure 6: Intensity Histogram Before Equalization..........................14
Figure 7: Intensity Histogram After Equalization..............................14
Figure 8: Image Revealing Edges....................................................15
Figure 9: Filtered Image.................................................................16
Figure 10: Connected Component Analysis......................................18
Figure 11: Character Segmentation................................................19
Figure 12: Filtered Plate.................................................................20
Figure 13: Character Extraction......................................................22
Figure 14: Multi-layer Feed-forward Network..................................24
Figure 15: Neural Network training space......................................27
Figure 16: Hybrid Back Propagation-Simulated Annealing Training....30
1. Introduction

In recent years, with the increase of terrorist activities around the world, security has become a major concern. The demand for security related services has been higher than ever before, and there is a great need to find new ways to protect ourselves or improve the existing methods by using information technology. One area of interest has been automated surveillance systems controlled by computers that could work independently with minimal human intervention. An automated system that could identify suspect vehicles passing through can issue alerts or report such incidents to corresponding authorities immediately. This will speed up response time and save lives.

Traditionally, electronic surveillance systems have relied on specially installed devices, in order to automatically identify a vehicle. A pre-installed device will transmit a special identification code upon request and the code is used for identification of the vehicle, granting access if authorized. This approach works fine in a closed community where the vehicles to be watched are known in advance and the aforementioned devices are installed. However, these systems will fail to track any vehicle without these devices, which makes it inappropriate for open access areas.

Other type of systems only work as a monitoring system without identification capabilities. These systems might capture videos or pictures of passing through vehicles
by installing video or high speed cameras in specific gateways, but these videos or images are only stored without any processing. If there is a need to search for a specific vehicle later on, the stored videos/images must be pulled out the database and visually inspected one by one, most likely by a human operator. For an area of high traffic volume, this approach will present considerable work to find any particular car, its cost-effectiveness is extremely low and there is no immediate response at all.

An automatic License Plate Recognition System (LPRS) is the perfect solution to replace these legacy system. A LPRS will take as input images of the passing automobiles, captured using a high-speed camera at specified gateways. Then the captured images will go through the system that will identify the license plate number of the vehicle without human intervention. The retrieved identity and the original image taken can be stored for review. Since vehicle information has already been detected at the time of storing, the information of interest can be indexed for fast retrieval and easily searched. The system can be completely automated by including motion sensor to trigger the image capturing device and a database system for storage.

A LPRS is desirable by many areas. A LPRS can be installed at the entrance of a gated community, it can automatically open the gates for the returning tenants by identifying their vehicles stopping at the doorway. It can be installed at the entrance of a parking structure and monitor the passing cars. It can be installed at the toll-ways and
print the vehicle information on the receipt. Or it can be used by law-enforcement authorities to track and find suspect vehicles.

Many challenges are involved in building such a LPRS to produce a recognition rate comparable to human vision. Human can easily distinguish and recognize different pattern and contents by visual examination of a digital image. Our eyes and brain are trained to work together to quickly locate the target area on an image by filtering out backgrounds, and process only the desired information. Even if we are presented with an image with high level of noise or some type of distortion, we can still adjust with little effort.

Computer programs hardly know how to adjust to distortions, or imperfections in the input image, as it will directly affect the outcome of the recognition process. Some issues are related to the external environment and settings that are related to the process of input images capturing. The quality of the input image depends on the some external settings in which the images are acquired. Camera location is an important factor, since narrow angle will result into distorted license plate with content all clustered and difficult to distinguish. Appropriate lighting condition is needed in order to obtain clear images, poor lightening conditions can result into inadequate exposure distorting the color and creating shades obfuscating the real object. Also, if a LPRS must recognize plate number out of moving vehicles, then camera speed must be high enough to avoid unnecessary
blurring. Image resolution is another factor to consider, lower resolution images can be difficult to recognize, and even though higher resolution images will provide much more information helping recognition rate, the resolution increase will translate directly into longer processing times. These are mostly external issues that require a hardware solution, but they do need some considerations during software design.

Even after acceptable input images are obtained, there is no guarantee of success. Computers are excellent at performing millions of calculations at high speed, but they fail to establish simple relationship between object. The task of identifying a car out of a background might be simple enough that a little kid can do, but it is hardly trivial for a computer. Digital images are represented just as a big numerical matrix to the computer. The LPRS must attempt to “visually” inspect an image by analyzing the number to find some kind of pattern, determine whether these corresponds to a vehicle license plate, and extract the plate identity. If image capturing process is also automated, there is no guarantee on the exact location where the plate will appear within the image, nor its size, or whether constant lighting condition will be kept. The LPRS system must be able to cope with all these conditions on the fly and produce results at real-time.

This thesis explore the architecture of a LPRS software system and proposes innovative methods for implementing one such system that can quickly and accurately identify vehicles license plate numbers by examining digital images of the cars. For plate
localization, a heuristic combining several traditional image processing techniques is used in combination to filter out the background and extract the candidate region enclosing the plate. Techniques such as histogram equalization, edge detection, filtering and component analysis each plays a role in the extraction process. For character segmentation, an algorithm based on information obtained by filtering, adaptive thresholding and clustering is used to extract each individual character. Then, Artificial Neural Networks (ANN) are in charge of the final Optical Character Recognition (OCR). Traditional training method for ANN are extremely time consuming and often resulting in sub-optimal configurations. A hybrid training method is introduced, this new training method combines the strength of traditional gradient descent based Back-propagation algorithm and random selection based Simulated Annealing process to overcome such shortcomings.
2. Previous Work

The license plate recognition process can be roughly divided into three steps as shown in Figure 1, Plate Localization, Character Segmentation and Character Recognition. Each step will be carried out by an independent module. An input image submitted to the system is first examined and processed to obtain the vehicle license plate region, then the plate region is processed to locate each individual digit and character, these are then submitted to the final Optical Character Recognition (OCR) process to determine the identification.

![Figure 1: Recognition Steps](image)

2.1. Plate Localization

This is the process of extracting the license plate region out of an image. It
involves basic image processing techniques combined with some decision making based on deterministic threshold. Without any prior knowledge on how large the plate is, or where it is located, the entire image must be inspected and analyzed in order to extract candidate regions.

There are many different approaches on how to accomplish this, some algorithm assume that the plate region location of the image should not vary by much [1], and it should be adjusted by using sensors [5], thus limiting the search range for fast results. Some technique make use of only edge information for plate location [3][8], and there are also very complex algorithms such as vector quantization [6], fuzzy clustering [7] and fuzzy logic[9].

2.2. Character Segmentation

Once the candidate region is detected from the input image, the next stage is to segment the plate to extract individual digit/character for recognition. This process is highly dependent on the format of the plate being processed. Because different country and regions have different plate shapes and sizes, the color used as plate background and foreground are totally different and their content varies both in length and combination of digit and characters. For example, the Chinese license plates analyzed in [8] have dark background with character in lighter color, the algorithm used can not be directed applied to the plates in Alberta, Canada [4], since those are completely opposite.
Some techniques do exist for specific cases and can be adapted to other cases, such as combination of vertical and horizontal projection to determined glyph location [4], or by using an adaptive clustering technique by finding white spaces between columns of higher density of dark pixels [1]. However, these techniques do not take into account that sometimes there could be frames that are partially connected to the plate content. Take as an example Figure 2, which features a plate that has a frame that is connected to the lower part of the middle five glyphs. The use of simple projection or clustering techniques will not yield adequate results.

Figure 2: Framed Plate

2.3. Character Recognition

Once each individual digit and/or character is extracted, it must be identified in some way. This process is called Optical Character Recognition, and there are several different solution to this problem. Two approaches are particularly popular among many different researches on license plate recognition.

One of the methods is template matching [5][7]. In this method, a series of slightly different templates of all glyphs are kept in a database. Once a image is submitted
for recognition, existing templates are compared to the new image, and the best fit will determine its identity. This method requires that the template database be large enough to cover most glyph variations, and it must also have an efficient algorithm to process large set of templates.

The other way is by using Artificial Neural Networks as classifier [1]. The ANN classifier must be trained before use to recognize all the different digits, characters and symbols that require identification.

Several other approaches are also being used, feature analysis is the method used in [9], which is an analysis of the characters to distinguish it. And for recognizing a totally different set of symbols in those plates in Thailand, Hausdorff Distance technique was the choice [10].
3. **System Architecture**

Each country and region has its own specifications for their legal vehicle license plates, and they differ from each other in several aspects. To begin with, the license plate itself can have different shape/size, have totally opposite set of colors for background and foreground [4][8] and it can be located at different location on the vehicle. Also, the number of digits or characters will differ, A region with only a couple of thousands cars only need to use a plate with 6 or 7 numeric digit, but for a region with several million cars that configuration is not nearly enough. This can change the location of each digit/character on the plate, they can be in a specific font, on some specific background colors or not. For a country that does not use Latin characters as their primary written language, even totally different set or symbols are needed[10]. These substantial difference among them make it almost impossible to create a single algorithm that can be universally used for all plates.

The algorithm defined in this chapter are designed to recognize California State license plates as shown in Figure 3. The regular California license plate has the following configuration

- The plate background is a single uniform color with no graphics.
- The word “California” appears at the top-center on the plate in red and italic.
The plate number starts with a digit(0-9), followed by three English characters (A-Z) and three more digits (0-9).

- All seven digit/characters are sans type font.
- The plate background is a light shade of gray while its characters are of dark blueish color.

![California License Plate](image)

**Figure 3: California License Plate**

### 3.1. Plate Localization

The algorithm proposed does not make any assumption regarding possible location and/or size of the plate region (See Appendix A for some sample images of different cars).
It only relies on both edge and color information for extracting the license plate region out of random images. It makes use of some constants that provide helpful information on the plate location in any image.

- No matter the lighting condition, plate background color are all lighter and plate foreground are darker.

- Plate region are all rectangular shape with same specific proportion between width and height.

- Plate region will be high in edge concentration.

The following a flow diagram of the algorithm used (Figure 5)
3.1.1. Noise Filtering

Images acquired through a digital camera may be contaminated by a variety of noise sources. In image processing terms, noise is refer to stochastic variations as opposed to deterministic distortions such as shading. While most current cameras will reduce the noise levels to a very low level, noise can never be eliminated completely.

Noise can be further reduced by application of the smoothing filter. In this case, a 3x3 Gaussian convolution filter with $\sigma = 0.5$ is used.
3.1.2. Brightness Normalization

Images can be taken at various lighting conditions. In each case, the difference in brightness will greatly influence the color pixel value of the plate and the characters. This process is an attempt to compensate for the different lighting conditions under which the image is taken.

This is achieved by applying the following:

1. Convert the image into HSV color space
2. Equalize its intensity component.
3. Convert back into RGB color space

Figure 6: Intensity Histogram Before Equalization

Figure 7: Intensity Histogram After Equalization
3.1.3. Extract Edge Information

None of the traditional edge detection operations such as derivative based detection or canny edge detection is used. Instead, the edges are revealed by creating a difference image $img$ as

$$img = \text{threshold} \left( \text{close} \left( src \right) - src \right)$$

where $src$ is the grayscaled source image, $\text{threshold}$ is a binary threshold function and $\text{close}(src)$ is the morphological close operation.

This process enhances high frequency features in the image, thus highlighting the license plate content. This image provide a good starting point on where to locate the plate.

Figure 8: Image Revealing Edges
3.1.4. Filtering

The next step is to filter the image eliminating backgrounds and unnecessary elements. This is accomplished by a combination of using the edges information obtained from previous step and some assumption about California license plates.

1. From previous image, the glyphs in the plate are clearly presented. And next to these pixels, the plate background can also be extracted.

2. As mentioned, California state license plates has a light background which will be captured as different shades of gray depending on lighting conditions.

An color is picked according to above two conditions, and it is assumed to be the plate background color and used as filter.

![Image](image.png)

*Figure 9: Filtered Image*
3.1.5. Connected Component Analysis

These patches are converted into binary images, and grouped as connected components. Components of different size and shapes will be scattered around the image at this point and one of them contains the license plate being searched.

The analysis process will find the candidate region out of the collection of connected components by using some previous knowledge about the plate. A component is considered to be a candidate if

1. the area covered by it also covers an area of high concentration of non-zero pixels from Figure 8, an indication of possible presence of many character/digits.

2. and its width/height proportion is similar to the CA license plate.

Any connected component matching these two criteria is considered to be a candidate region worthy of further examination. This process can produce results ranging from zero to several candidate regions. If many candidate regions are found, they are all considered to be valid candidates and submitted to the next stage of processing.

As shown in Figure 10, out of all the different connected components created on the filtered image, two of are identified as candidates. Both are considered valid regions.
and pass along to the character segmentation stage.

Figure 10: Connected Component Analysis

If no candidate region is found, the process is repeated from 3.1.4 by picking a different color and using the new “plate background” color for filtering. New connected components of different size and shape will be obtained and analyzed.

If the system still fail to find any region after multiple color has been used, the image is rejected by the system as lacking a plate region.

3.2. Character Segmentation

The segmentation algorithm is a combination of several techniques that are
applied as necessary. First, a simple method like adaptive binary thresholding is used along with some clustering. If results are negative, then some other methods are added progressively increasing its complexity.

The high level flowchart of the algorithm is shown in Figure 11. The following steps are repeated for each possible candidate region until valid set of character/digit regions are found in one of them.

![Figure 11: Character Segmentation](image)

Figure 11: Character Segmentation
3.2.1. Apply Color Filter

The candidate region is filtered by eliminating the colors that do not correspond to neither the background nor foreground. This filter is applied again to eliminate the red “California” text at the top of the plate. It also eliminates other foreign objects on the plate, such as stickers.

3.2.2. Thresholding

Use an adaptive algorithm to find a threshold for this image and use an inverse thresholding function to create a binary image.

\[
img_{ij} = \begin{cases} 
255, & \text{if } img_{ij} < \text{threshold} \\
0, & \text{otherwise}
\end{cases}
\]

In the best case scenario, the only pixels left after the binarization would be the pixels forming the license number (Figure 12).

Figure 12: Filtered Plate
3.2.3. **Characters Extraction**

Create connected component out of the filtered image and use boxes to fit the components. Try to extract the glyphs by finding a series of boxes with very similar size and shape. For a successful image, seven boxes should be found (Figure 13), each one will contain a group of pixels forming a digit or a character.

If the number of boxes found is less than seven, then one of two possible route is selected.

1. If more than two boxes has been found, then an interpolation is performed to reconstruct the required seven boxes based on existing boxes.

2. If less than two boxes are found within the plate region and there is a large cluster of pixels, then this could indicate the presence of a frame attached to the plate content in some ways. An attempt for frame removal is performed, and the image is re-evaluated starting from 3.1.1.

3. If above steps are applied several times without success, the candidate region is discarded and the next region from the collection is evaluated.

After individual characters are extracted, each character is converted into a 20x10 matrices of one's and zero's. This matrix will be submitted to the final stage of
recognition by neural network.

![Character Extraction](image)

**Figure 13: Character Extraction**

3.3. **Character Recognition**

The Optical Character Recognition method chosen as part of the LPRS system is an Artificial Neural Network.

3.3.1. **Artificial Neural Networks**

Artificial neural network (ANN) is an area of research in Artificial intelligence (AI) that attempts to model the human brain. An ANN is composed of a large network of interconnected processing components (Neurons). Just as our brain process information by having neuron cells interact with each other, an ANN try to solve problems in the same way. ANN differs form conventional AI problem solving, as it does not depend on particular algorithms, but it relies on all its neurons working as independent agents.

Neuron interact with each other by receiving and firing signals according to its
internal structure. The firing of the neurons is determined by all the input signal it receives, adjustment weights and a firing rule. Weights are used to alter the signals traveling from neuron to neuron. By adjusting these weights in the learning process, a network can be taught to produce the desired outcome given an input. The Firing rule used by neuron can be a threshold-based step function

\[ f = \begin{cases} 1, & \text{if } \sum w_{ij} > \text{threshold} \\ 0, & \text{otherwise} \end{cases} \]

or a sigmoid function

\[ f = \frac{1}{1 + e^{-t}} \]

ANN can solve classification problems. Input data act as external stimulus to the network, and all neurons interact with each other in parallel as the it receives signal from some neurons and fire signals to others. The final output of the ANN will be a pattern that identifies the class/group in which input data belongs to.

Neural Network Architecture

ANN can be divided into categories based on their different topologies

- Single-layer neural network: neuron receive input and process to produce output.
o Multi-layer Network (Figure 14): One input layer, one output layer with or without hidden layer(s). This architecture is also called Feedforward as information flow is strictly one way. Each neuron receive signal from layer above and fire signal to layer below.

o Self-Organized: Network connections are dynamically established or broken to produce the desired outcome.

![Multi-layer Feed-forward Network](image)

**Figure 14: Multi-layer Feed-forward Network**

**Neural Networks Training**

ANN has the ability of learning by sample. And it can learn through supervised or unsupervised manner.

- Supervised training: Both the inputs and the outputs are provided. The network processes the inputs and compares the results against the desired
outputs. Errors are then calculated, and back-propagated to the weights which control the network. This process occurs over and over as the weights are continually tweaked.

- Unsupervised training: the network is only provided with inputs. The network then adjust itself by reorganizing or adapting to classify the inputs.

**Back-Propagation Algorithm**

The term is an abbreviation for "backwards propagation of errors". It is one of the most used methods of supervised training for ANN. It is most useful for training multi-layer feed-forward neural networks [14].

The algorithm can be summarized as follows

1. Introduce input training data to the neural network.
2. Compare the network's output to the desired output from that input. Calculate the error in each output neuron by
   \[ E = \frac{1}{2} \sum (correct - output)^2 \]
3. Calculate the error at each weight and adjust
4. propagate error values to upper levels
Limitations of Back-Propagation Algorithm

Back-Propagation algorithms work as a form gradient descent method to gradually adjust the weights and reducing the error produced. At each weight adjustment, the system takes a step toward the gradient, by doing so, it guarantees that new result produced after the adjustment will be no worse than previous. As the number of iterations increase, the system gradually advance toward the valley.

The Error Back-Propagation method of training suffers from the limitations of the general gradient descent method (Figure 15):

- Lengthy training process: The process can be length depending on where the starting point is.
- Traveling through large flat plateau: during the gradually ascend/descend process, it system might spend lot of time trapped in a large flat plateau with no improvement at all
- Trapped in a local minima: this is a worst problem. Once trapped in a local minima, the algorithm does not offer anyway of getting out. So the system will actually never reach the globally best solution.
A possible way to overcome these limitation is by including a simulated annealing process to the training algorithm

3.3.2. Simulated Annealing

Simulated annealing (SA) is a generic probabilistic heuristic algorithm for locating the global optimum of a given function in a large search space. Instead of performing gradient descent, it performs a random search method. It was invented by Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P. in 1983 [15].

The origin of the word annealing comes from metallurgy, annealing is a technique involving heating and controlled cooling of a material reduce their defects and obtain
material in their optimum state. If a metal is heated and cooled down quickly, sometimes it might be stuck in some sub-optimal state of higher energy. By introducing some heat to the metal causes the atoms to wander randomly through different states; and the slower cooling gives them more chances of finding configurations with the lowest internal energy (optimum state).

**Simulated Annealing Algorithm**

Similar to the metallurgical process, each step of a Simulated Annealing algorithm replaces the current solution by a random "nearby" solution, chosen with a probability that depends on a global parameter $T$ (called the temperature). $T$ is gradually decreased during each step of the process. Initially, when $T$ is high, the system will basically change almost randomly around the entire search space. But as $T$ decreases, the system will come into an configuration which is the best of all the configurations tested. The random wandering saves the system from becoming stuck at local minima. Thus the temperature decrease takes long enough, the system will eventually come down to a globally optimum solution.

3.3.3. **Neural Network Classifier Implementation**

Two different neural networks are used in LPRS for recognizing the glyphs extracted in the segmentation stage. One is used only to recognize numbers, and the other
is trained to recognize exclusively characters. The choice of separate neural networks was made specifically to target California license plates, which has predetermined placement for digits and characters. Separated neural network means that each network have to learn to recognize less symbols, this expedited the training process and reduced errors rate.

Both neural network are Feed-forward networks, and use Sigmoid function to control neuron firing. The character neural network has one input layer, one hidden layer and one output layer, the input layer is a 20x10 matrix of neurons, the hidden layer has 5 neurons, finally the output layer size is 26, each representing a character in the English alphabet. The digit neural network also has three layers, and has the same input layer size. Its hidden layer has 4 neurons and its output layer size is 10, one of each digit.

The training on the neural networks is carried out by using a combination of classical error back-propagation and simulated annealing process. Each Training block in the diagram is a shortened back-propagation training process in which the ANN will gradually and steadily converge “downhill”.

After the ANN has received some training, it is submitted to the annealing process to avoid local minima. Some random neighboring samples are chosen by altering neuron weights within a normal range based on global temperature $T$. The Energy of each neural
network is obtained by calculating the overall error ($\sigma$) as

$$\sigma = \text{rand}(T) \sum E_i,$$

Where $E_i$ is the results of the error function evaluated for the $i^{th}$ training data set and $\text{rand}(T)$ is a random value based on $T$. By adjusting the sum of error, “uphill” movement is made possible. The network with the lowest energy (lowest $\sigma$) is considered to be the best configuration at that point and will received more back-propagation training. The process is repeated until $T$ reaches zero or $\sigma$ reaches zero.

The following is the flow chart of the ANN training algorithm implemented

![Flow Chart](image)

Figure 16: Hybrid Back Propagation-Simulated Annealing Training
Simulated annealing are proven to have the ability of not getting stuck of any local minima, but the random nature of its sampling heuristic does not always guarantee that system is moving toward best possible direction. In the other hand, back-propagation almost guarantees a fine granular descent into a nearly best, but that nearby best is not always the global optimal. By combining the two methods, this training heuristic produces better and faster results than using either method separately, the system will be altered to the placed around a configuration near the globally best solution, and quickly descend into it.
4. Experimental Results

4.1. Settings

The computer used for running the experiment is a notebook, with a Pentium 4 1.0 GHz processor and 512 MB of RAM.

Due to the limited availability of data at the time of experiment, the LPRS system was tested by using a reduced set of data. A total of 50 images with vehicle license plates taken under different lighting conditions, different distance, different exposure were used as input to the system. The images were originally taken using a 2 Mega-pixel digital camera and resized to 800x600 for quicker processing.

4.2. Plate Localization

All images were submitted to the plate localization process, candidate regions are marked with a red rectangle after processing (See Appendix A for sample image). The images are visually inspected to verify its correctness.

Table 1: Plate Localization Success Rate

<table>
<thead>
<tr>
<th>Total Images</th>
<th>Plate located</th>
<th>Failed to locate</th>
<th>Success rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>48</td>
<td>2</td>
<td>96%</td>
</tr>
</tbody>
</table>
The reason of two of the images failed were exclusively because of the quality of the image (See Appendix B for sample image). Once picture was blurred and the other picture was taken from a distance and as a result, the plate area was very small.

4.3. Character Segmentation

Only those images with candidate regions identified are submitted to Character Segmentation. After processing, the plate area is extracted from the input image and all seven boxes enclosing the characters are drawn in red (See Appendix A for sample success image).

Table 2: Character Segmentation Success Rate

<table>
<thead>
<tr>
<th>Total Images</th>
<th>Character segmented</th>
<th>Failed</th>
<th>Success rate</th>
<th>Cumulative success rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>48</td>
<td>45</td>
<td>3</td>
<td>93.75%</td>
<td>90%</td>
</tr>
</tbody>
</table>

Out of the three failed detections, two of them were of the same vehicle which had a frame covering the lower part of the plate (See Appendix B for image). And the other one did interpolate seven boxes, and five of the boxes were on target, but the other two boxes failed to completely enclose the glyph it should be associated with.
4.4. Optical Character Recognition

Both networks were trained by using a set of training data consisting of many variations of the digit/character to identity.

For the digit ANN, the training data set was created as follows

1. five different images of a digit were used as base images, these images all have slight differences like tilt, height/width proportion, etc...

2. Nine synthetic images are created from the base images by distorting its proportions, by adding Gaussian noise, by adding or removing entire row/column of pixels.

3. A total of fifty training image for each digit.

The data set are submitted to the ANN and trained by using the hybrid Back-propagation Simulated Annealing method.

The preliminary results are as follow

Table 3: OCR Recognition Rate

<table>
<thead>
<tr>
<th>Recognition on training data</th>
<th>80%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recognition on actual images extracted from LPRS</td>
<td>60%</td>
</tr>
</tbody>
</table>
4.5. Processing Time

The processing time was an average of 300ms per image from input submit to result display which should allow LPRS to run in real-time. The processing time is mostly spent on Plate Localization and Character Segmentation, the ANN recognition takes very little time to process for each segmented character.
5. Conclusion

A LPRS was proposed with a color-edge filtering process to locate vehicle plate out of an image, a filter-interpolated approach for character segmentation and an hybrid Back-propagation Simulated Annealing process for neural network training.

The preliminary experimental results confirm that the hybrid approaches' performance are quite satisfactory and with some minor adjustments they should be able to consistently produce better results. The accuracy of the system can still be improved by experimenting with slightly different parameters during image pre-processing stage of plate localization and character segmentation. Also, a better component analysis should identify the single region instead of multiple candidate thus speeding up the processing time.

As future work, current LPRS can be extended to recognize unusual California license plates with different configuration. This will require the system to adapt its algorithm so it can process plates with less than seven digit/characters, it also requires the LPRS to recognize symbols, such as “♥”.

Other possible methods can be tried for character recognition. One possible approach will be modifying the current neural network configuration. Instead of using one neural network for characters and one neural network for digits, multiple neural
networks can be used, each specialized for certain symbols. This could greatly shorten the training time and improve the accuracy of the classification.
References


39


Appendix A - Successful Images

Input Image

Image showing candidate regions

segmented plate
Input Image

Image showing candidate regions

segmented plate
Input Image

Image showing candidate regions

segmented plate

Valley Sierra
Input Image

Image showing candidate regions

segmented plate
Appendix B - Unsuccessful Images

Failed to detect license plate region.

Caused by image blurring.

Failed to segment the characters.

A Frame is attached to it.