Proceedings of the 5th WSEAS International Conference on Applied Computer Science, Hangzhou, China, April 16-18, 2006 (pp793-800)

Online Vehicles License Plate Detection and Recognition System using Image Processing Techniques

ASAD ALI SAFI, MUHAMMAD AZAM,
SHAHBAZ KIANI, NADEEM DAUDPOTA
Department of Computer Science,
COMSATS Institute of Information Technology
Abbottabad, NWFP, Pakistan.

Abstract - A growing demand for traffic data concerning traffic flow and automatic vehicle identification, led researchers around the world to adopt advanced electronic and computer vision technologies to monitor and control traffic. Increasing levels of road traffic ask for real time analysis of a moving car in order to extract important data, in this case the vehicle registration number. This paper describes a system based on image processing technology to meet the increasing needs of efficient traffic management. System detects and recognizes a license plate of the car registered in NWFP province of the Pakistan. It performs the recognition in almost real-time, watching cars slowing down in front of video recording device. Beauty of the system lies in a fact that it is using very low cost devices to accomplish this task even in real time.

Key-Words: -Vehicles License Plate Recognition, License Plate Area Location, Edge Detection, Corner Detection.

1. Introduction

The system given the name VLPR is used to identify vehicles by their license plates. VLPR provides automatic detection of car license plates within real view camera scene. It uses a camera to take the image of the front of the vehicle, and then image processing algorithms are applied to analyze the images and extract the plate information. Later on this data can be used for various applications. [5][6] Data can used for (1) Automated entry in parking against prepaid membership, (2) Road-toll calculation between check-in and check-out points, (3) Authentication while crossing countries boarders, (4) Stolen cars tracking by alarming the un-identified vehicles, (5) Detection of charged vehicles, and for various other purposes.[7][8]

The main tasks of VLPR are recognition of location of the license plate area in the image after capturing the image through ordinary video recording device, the segmentation of the characters and their identification. These tasks are strongly inter-related, mainly because the way to check if the license plate has been correctly located is based on the result of the character identification process (it should correspond to a predefined syntax). For the location of the license plate area in the image, the method takes advantage of the "signature" of the license plate area in a horizontal and vertical cross-section of the image, “L” type shape is find out from top right and bottom left by two ratio one, instead of looking for character like shapes in the image.[1][2]

The research is made to make low cost VLPR system, for that ordinary video recording device and Intel P3 System is used to deploy a VLPR system. The working of VLPR is comprised of four modules corresponding to the four main computational phases as shown in the Figure 1, (1) License plate area location module, (2) Preprocessing module, (3) Text recognition module and (4) Authentication module.

2. System Structure

2.1 Phase 1 License Plate Area Location Module

In this section a brief description of the NWFP license plates is given and the technique to locate license plates in the given input video is also proposed.

First of all the issue needed to be handle is when to start the camera to take the video, for this purpose some sensors are placed on road side at a forefront, that send signals to the system to “switch on” the camera for taking video, and it is observed that 8 feet ahead is the good position for sensors placement. As shown in Figure 2. The video of the car from the front is captured now we focus our discussion, to locate the plate from video clip being captured through video recording device. In order to achieve this task, we need the outer body line of the plate that is black thick line (0.5 cm) shown in figure 6.

To find out this rectangle outline that is basically a license plate boundary encompasses the license plate, some steps are needed to be performed on video being captured.

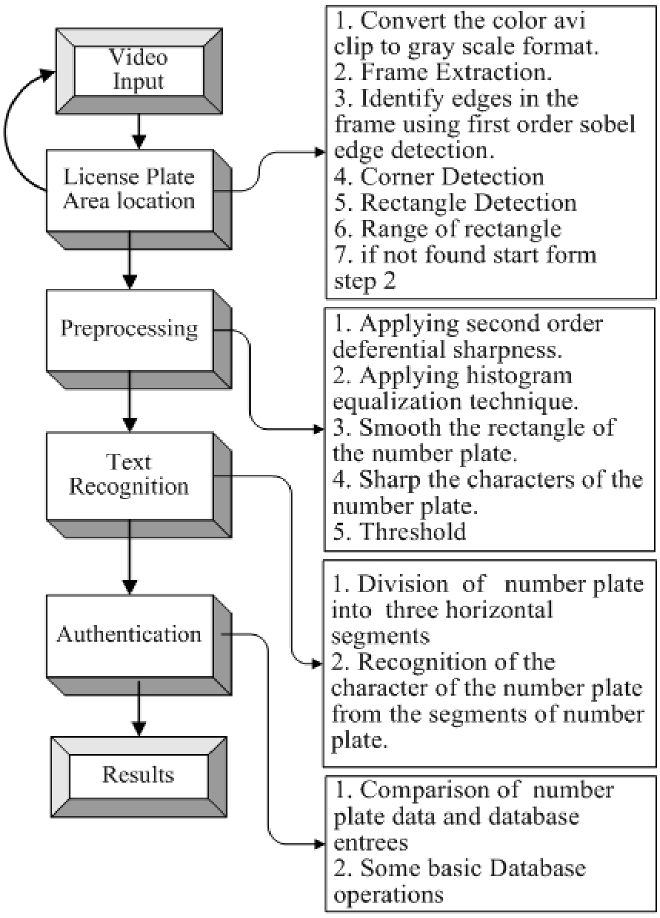

Figure 1: Block diagram

2.1.1. Gray Scale Conversion

The video is converted to gray scale format [9], to get rid from difficulties of colors management.

2.1.2. Frame Extraction

The image processing will be performed on a single image that is why frames are extracted from the video. For efficiency of the system and to have some change in next frame every 10th frame of the video is taken for the processing. Then a particular frame is extracted and

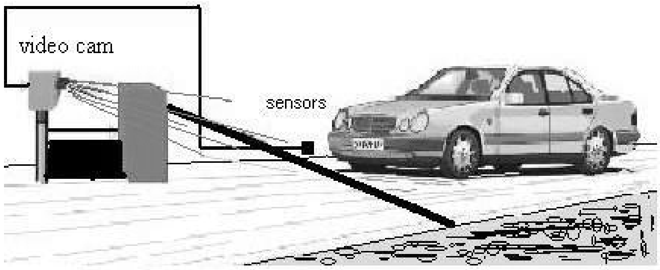

Figure 2 : Sensors placement

copied, that is used to perform image-processing operations to extract the required information.

2.1.3. Edge Detection

For the edge detection in the frames, first order Sobel edge detection filter is applied on the each frame. It is applied in four directions; horizontal, vertical, diagonal edges at – 45 degree and diagonal edges at 45 degree. [9][11] [12]

The Sobel Edge Detector uses a simple convolution kernel to create a series of gradient magnitudes. For those mathematically inclined, applying convolution *K* to pixel group *p* can be represented as:

$$N(x,y) = \sum_{k=-1}^{1} \sum_{j=-1}^{1} K(j,k)p(x-j, y-k)$$

The Sobel Edge Detector uses two convolution kernels, one to detect changes in vertical contrast (h_x) and another to detect horizontal contrast (h_y).

$$h_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad h_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

The data can now be represented as a vector (gradient vector). The two gradients computed using h_x and h_y can be regarded as the x and y components of the vector. Therefore we have a gradient magnitude and direction:

$$\mathbf{g} = \begin{bmatrix} g_x \\ g_y \end{bmatrix}$$

$$g = \sqrt{g_x^2 + g_y^2}$$

$$\theta = \tan^{-1}\left(\frac{g_y}{g_x}\right)$$

Where **g** is the gradient vector, g is the gradient magnitude and θ is the gradient direction.[18] The output of the sobel is shown in Figure 3.

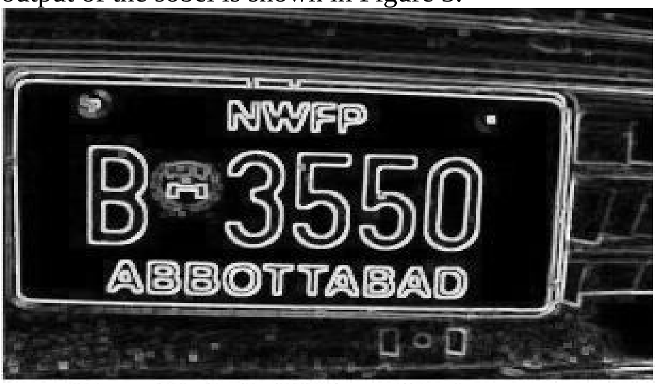

Figure 3: Sobel Filter effect

2.1.4. Plate Corner Detection

The corners of the plate are detected through edges. In this concern an algorithm is designed for detection of license number plate in a frame. In order to find the corners in a frame some preprocessing is also required,

2.1.5. Edge Linking

In practice this set of pixels seldom characterizes an edge completely because of noise, breaks in the edge form nonuniform illumination, and other effects that introduce spurious intensity discontinuities. Thus edge detection algorithms typically are followed by linking procedures to assemble edge pixels into meaningful edges [9]. We used a very simple approach called "local processing" [13] in which the linking edge points are to be analyzed against characteristics of pixels in small neighborhood (5 × 5) about every point (x, y) in an image that has been labeled an edge point, two properties used for establishing similarity of edge pixels, (1) The strength of the response of the gradient operator used to produced the edge pixel, (2) The direction of the gradient vector. The first property is given by the value of ∇f . Thus an edge pixel with coordinates (x_0, y_0) in a predefined neighborhood of (x, y) is similar in magnitude to the pixel at (x, y) if $|\nabla f(x, y) - \nabla f(x_0, y_0)| \leq E$

Where, E is a non-negative (e.g. 25) threshold. The direction of the gradient vector is $\alpha(x, y) = \tan{-1}(G_y / G_x)$, the edge pixel at (x_0, y_0) in the predefined neighborhood of (x, y) has an angle similar to the pixel at (x, y)

$if\ |\alpha(x, y) - \alpha(x_0, y_0)| < A$

Where, "A" is a non-negative angle threshold.

A point in the predefined neighborhood of (x, y) is linked to the pixel at (x, y) if both magnitude and direction criteria are satisfied. The process is repeated at every location in the image. A record must be kept of linked points as the center of the neighborhood is moved from pixel to pixel. [9]

2.1.6. Segmentation

Through edge linking we have determined different edges in image, the techniques that are used to find the objects of interest are usually referred to as segmentation techniques.[9] Our objects of interest are horizontal and vertical lines, all the horizontal edges are selected that lies between $(x_{\min}, y_1)$ & $(x_{\max}, y_2)$.

And the data is stored in array list of horizontal edges, similarly the vertical edges that lies between $(x_1, y_{\min})$ & (x_2, y_{mix}) as shown in Figure 4.

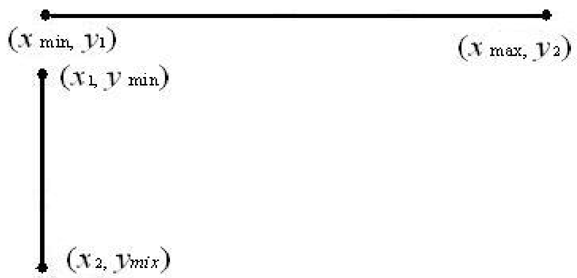

Figure 4 : Objects of interest (horizontal and vertical lines)

2.1.7. Finding point of intersection

Now to find the points of intersections we need some calculations let us consider the general form of the equation for the two points of straight line;

$$\frac{y_2 - y_1}{x_2 - x_1} = \frac{y - y_1}{x - x_1}$$

$$(y_2-y_1)x-(y_2-y_1)x_1=(x_2-x_1)y-(x_2-x_1)y_1$$
$$(y_2-y_1)x+(x_2-x_1)y_1-(y_2-y_1)x_1=(x_2-x_1)y$$
$$Ax+C=By$$

where

$$A=y_2-y_1$$
$$B=x_2-x_1$$
$$C=By-Ax$$

Where A, B and C are numbers which define the line segment. Here we have collection of line segments (horizontal and vertical). We choose a line segment that refers the horizontal line segment as line 1 and vertical line segment as line 2. Each line segment has extreme points. $(x_1, y_1),(x_2, y_2)$

Now if equation for horizontal line segment is
$A_1x+C_1=B_1y$
And for vertical line segment
$A_2x+C_2=B_2y$
By multiplying the equation of horizontal with B_2
$A_1B_2x+C_1B_2=B_1B_2y$
And equation of vertical with B_1
$A_2B_1x+C_2B_1=B_1B_2y$
Then we have value of x
$(A_1B_2-A_2B_1)x=C_2B_1-C_1B_2$

$$x=\frac{C_2B_1-C_1B_2}{A_1B_2-A_2B_1}$$

Similarly by multiplying the equation of horizontal with A_2
$A_1A_2x+C_1A_2=A_2B_1y$
And equation of vertical with A_1
$A_1A_2x+C_2A_1=A_1B_2y$
Then we have value of y
$C_1A_2-C_2A_1=(A_2B_1-A_1B_2)y$

$$y=\frac{C_1A_2-C_2A_1}{A_2B_1-A_1B_2}$$
$$y=\frac{C_2A_1-C_1A_2}{A_1B_2-A_2B_1}$$

Now as we have line segment only so we have to find also whether points (x, y) lie on both line segments or not. So we have to check if both points (x, y) on the line segment.
$if\,(x_1, y_1),(x_2, y_2)$ are edge points of segments then points (x, y) lie on the segments. If

$$\min\ (x_1, x_2) \le x \le \max\ (x_1, x_2)$$
and
$$\min\ (y_1, y_2) \le y \le \max\ (y_1, y_2)$$

Similarly the remaining other points are also checked. If both line segments intersects at point (x, y) then its point of intersection otherwise the point is out of limits.

2.1.8. Corner Detection

Now finding which corner may be possible at that point. We use the simple algorithm for corner detection in which edge points of the horizontal and vertical segments are used.

let
$(x_{h1}, y_{h1}),(x_{h2}, y_{h2})$
where
$x_{h1} < x_{h2}$
are edge points of horizontal segments

let
$(x_{v1}, y_{v1}),(x_{v2}, y_{v2})$
where
$x_{v1} < x_{v2}$
are edge points of vertical segments
and (x, y) is the point of intersection then

```
if (x − x_h1 ≥ 35 )
{
        if (y − y_v1 ≥ 35)
        TopRightCorner
        else
        if (y − y_v2 ≤ −35)
        BottomRightCorner
}
else
{ if (x − x_h2 ≤ −35 )
        if (y − y_v1 ≥ 35)
        TopLeftCorner
        else
```

$if\left(y - y_{v2} \le -35\right)$

BottomLeftCorner

}

2.1.9. Rectangle Detection

The License Plate is detected through the corners by ratio 2:1; the algorithm finds 2:1 rectangles in size that is almost of the size of Number Plate. To finding the rectangle in image as we have four components BottomLeftCorner, BottomRightCorner, TopRightCorner and TopLeftCorner of the rectangle. It is observed that the upper corners of the rectangle will always come, above the mid of the image as shown in figure 6. Let we want to find the rectangle in the image using TopRightCorner which is above mid of the image and BottomLeftCorner then we can achieve this task by proposed algorithm being designed as shown in Figure 5 .

```
Rectangles: Array of rectangle will be identified.
BottomLeftCorner: Array of bottom left corners in any frame.
TopRightCorner: Array of bottom left corners in any frame.

void Rectangle(Array & Rectangles, Array & BottomLeftCorner,
                        Array & TopRightCorners)
Begin
    for (I=0 to I<BottomLeftCorners.Count)
        For (j=0 to j<TopRightCorners.count)
            Begin
             TemRectangle (BottomLeftCorner [I], TopRightCorner [j])
             If (TempRectangle.Width ()>0 &&TempRectangle.Height() >0)
            Begin
               Ratio= TempRectangleWidth()*1.0/TempRectangle.Height( )
               If(Ratio>1.7&&Ratio<2)
                    RectangleAdd(TempRectangle )
            End
        End
     End

IntRectangleWidth( )
Begin
    Return BottomLeftCorner.x –TopRightCorner.x
End

IntHeight( )
Begin
    Return BottomLeftCorner.Y –TopRightCorner.Y
End
```

Figure 5: Algorithm of Rectangle Detection

2.1.10. Rectangle Property Evaluation

Once rectangle is identified by applying the algorithm, following properties against identified rectangle are checked,

- Range of the rectangle, the size of the rectangle and the location of the rectangle in a image
- Color of the area within rectangle if it is white color then it can be a license plate as it is standard of the NWFP.

The rectangle (which is license plate) is copied from the image for further processing. The recognized number plate is shown in Figure 6.

2.2 Phase 2 Preprocessing

The preprocessing phase consists of conventional techniques for example image filtering, improving the visibility of the image and some enhancement in image. Some of the basic filters are applied on the area of number plate that is Histogram Equalization, Median filter and Morphological filters to enhance the particular area for next phase 3.

Figuer 6 : recognized number plate

Histogram equalization is one of the most important part of the software for any image processing. It improves contrast and the goal of histogram equalization is to obtain a uniform histogram. This technique is used on a whole image [3] [9].

The median filter is normally used to reduce noise in an image; it often does a better job than the mean filter of preserving useful detail in the image. Median filtering is a simple and very effective noise removal filtering process. Its performance is particularly good for removing shot noise. Shot noise consists of strong spikelike isolated values. This class of filter belongs to the class of edge preserving smoothing filters which are non-linear filters. This filter smooths the data while keeping the small and sharp details. [9]

The Morphological filter consists mainly on Dilation and Erosion. [14]
Dilation
Grayscale dilation is used to smooth small dark regions. It is defined as the maximum of the sum of a local region of an image and a grayscale mask. The shape of the input mask (known as the structuring element, or SE) is generally chosen to emphasize or de-emphasize elements in the image.
The general effects of performing dilation on a grayscale image are:

1. If all the values in the structuring element are positive, the output image tends to be brighter than the input.
2. Dark elements within the image are reduced or eliminated, depending on how their shapes relate to the structuring element used.

The degree of these effects depends greatly on the shape and values within the structuring element and by the details within the image itself.
Erosion
Grayscale erosion is used to smooth small light regions. It is defined as the minimum of the difference of a local region of an image and a grayscale mask. The shape of the input mask (known as the structuring element, or SE) is generally chosen to emphasize or de-emphasize elements in the image.
The general effects of performing erosion on a grayscale image are:

1. If all the values in the structuring element are positive, the output image tends to be darker than the input.
2. Light elements within the image are reduced or eliminated, depending on how their shapes relate to the structuring element used.

The degree of these effects depends greatly on the shape and values within the structuring element and by the details within the image itself. [9][10]

The image is mostly noise less so it is easy to recognize the text in image.

2.3 Phase 3 Text Recognition

This phase relates to the simple OCR (Optical Character Recognitions) operations [15][16][17]. The information is of different types on license plate and have different size, which creates some difficulties in text recognition for example it is difficult to differentiate between 8 and B or 2 and Z. In order to solve this kind of problems the plate is needed to be divided into segments according to required and provided information. The plate is divided into three horizontal segments and middle segment of this division is again divided into three vertical segments as shown in Figure 7. Hence total divisions on a plate are 5, each part is named with a number (1-5), division and information distribution on the plate is shown in figure 8.

The recognition of the text from the segments of the plate is now an easy job as it is known that there will be always a string of English alphabets in segment-1 (which is province name) of the plate so no difficulties in recognition. The segment-5 of the plate also has English alphabets which are maximum 10 characters (name of the district); the OCR will only find out the 10 capital alphabets from that portion of the plate hence there will also be no conflict of the characters and numeric values

Figure 7: Plate division

Segment-2 contains only one capital later which is the category, segment-3 contained a logo which is not readable by OCR and that is also not a segment of interest and segment-4 contains four numeric digits. Thus OCR has to find only the numeric digits. Finally from all these segmentations, the OCR [19] produces a correct result.

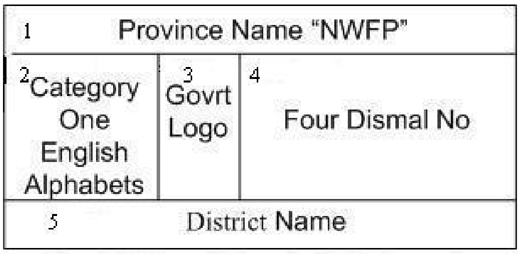

Figure 8: Division and information distribution on plate

2.4 Phase 4 Authentication

The authentication phase is the simple comparison of the values obtained from number plate and the values which are available in database tables. What kind of data should be stored in database, It all depends on the application for which the VLPR is being used. For example if the VLPR system is deployed on the entrance gate of any organization then only the employs vehicle record is entered in database and when ever the vehicles arrive on gate the number plate data is collected through above mentioned phases and compared if the values are same then the gate is opened for the vehicle and if not the non authentic vehicle massage is displayed, and the entry time is saved.

3. Conclusions

In this paper a novel technique is introduced that is very cost effective solution of license plate recognition. We focused on performance factor as well to make this system efficient. This system is capable to recognize the license plates with in few seconds (20-40). [4] The specialty of the system is the use of very low cost devices to achieve this critical and cumbersome task. Instead of using very costly cameras normally recommended by other such application, we used video recording device, easily available in market at very low cost that is capable to perform same task with same quality level. For the extraction of image of interest own designed algorithm is used that is very simple and able to perform the task of image extraction through simple steps that also enrich the performance factor of this system.

This generic license plate recognition software has many applications in the market. It can be used for (1) Automated entry in parking against prepaid membership, (2) Road-toll calculation between check in and check out points, (3) Authentication while crossing countries boarders, (4) Stolen cars tracking by alarming the un-identified vehicles, (5) Detection of charged vehicles as mentioned in section one. The results of License Plate Edge Identification, License Plate Extraction, License Plate Segmentation and License Plate Text Recognition are shown in table 3.1.

Speed	Accuracy	License Plate Edge Identification	License Plate Extraction	License Plate Segmentation	License Plate Text Recognition
Slow < 10km/h	Correct Recognition	89/110	89/110	85/110	86/110
	%age	80.90%	80.90%	77.27%	78.18%
Fast < 20km/h	Correct Recognition	54/110	54/110	50/110	51/110
	%age	49.09%	49.09%	45.45%	46.36%
Still	Correct Recognition	108/110	108/110	107/110	107/110
	%age	98.1%	98.1%	97.27%	97.27%

Table 3.1: Recognition rate for License Plate Edge Identification, License Plate Extraction, License Plate Segmentation and License Plate Text Recognition

References:

[1] Towards Robust Automatic Traffic Scene Analysis in Real-Time, D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russell, IEEE, 1994.

[2] Real time OD Estimation using automatic vehicle identification and traffic count data, Michael P. Dixon, Blackweel Publishers Oxford UK.2002

[3] Adaptive histogram equalization and its variations, Stephen M. Pizer, E. Philip Amburn, John D. Austin, Robert, Geselowitz, Trey, Bart Romeny, John Zimmerman, Academic Press Professional, Inc. San Diego, CA, USA 1987

[4] A High Performance License Plate Recognition System, Hans A. Hegt, Ron J. De la Haye, and Nadeem A. Khan, IEEE 1998.

[5] Automatic recognition of a car license plate using color image processing, Eun Ryung Lee, Pyeoung Kee Kim, and Hang Joon Kim, IEEE 1994.

[6] Automatic car plate detection and recognition through Intelligent vision engineering, Luis Salgado, Jose' M. Mene'ndex, Enrique Renddn and Narciso Garcia, IEEE 1999.

[7] Car plate recognition by neural networks and image processing R.Parisi, E.D.Di Claudio, G.Lucarelli and G.Orlandi, IEEE 1998.

[8] Automatic car license plate extraction using modificatied generalized symmetry transform and image warping, Dong Su Kimm and Sung-ll Chien, IEEE 2001.

[9] Digital Image processing 2end Edition, Rafael C. Gonzalez and Richard E. Woods, Pearson Education Singapore 2004

[10] Cell segmentation with Median filter and Mathematical Morphology Operation, Dwi Anoraganingrum, IEEE 1999.

[11] Processing images and video for an impressionist effect, Peter Litwinowicz, ACM Press New York, USA 1997.

[12] Design of an image edge detection filter using the Sobel operator, Kanopoulos, N. Vasanthavada, and N. Baker, IEEE, 1988.

[13] A Local Approch for fast line detection, Lefever, Dixon, Jeusse and Vincent, IEEE 2002.

[14] Breakdown probabilities of recursive soft morphological filters,Pertti Koivisto and Antti Niemisto, ACM Press New York, USA 2004.

[15] A Generative Probabilistic OCR Model for NLP Applications, Okan Kolak, William Byrne, and Philip Resnik, Association for Computational Linguistics Morristown, NJ, USA 2003

[16] Information Retrieval and OCR From Converting Content to Grasping Meaning, Jamie Callan, Paul Kantor and David Grossman, ACM Press New York, USA 2002.

[17] A Filter Based Post-OCR Accuracy Boost System, Eugene Borovikov, Ilya Zavorin and Mark Turner ACM Press New York, USA 2004.

[18] http://www.generation5.org Artificial Intelligence site on the Internet. Community-orientated, Generation deals with all AI topics.

[19] http://www.pegasusimaging.com