License Plate Detection: Anatomy of Image Based Techniques

[1]Akhila L, [2]Varunlal R

[1]Assistant Professor, [2] Assistant Professor
[1]Electronics and Communication Engineering
[2]Electronics and Communication Engineering
[1]Cochin University College of Engineering Kuttanadu, Alappuzha, India
[2]Muslim Association College of Engineering, Venjarammodu, Trivandrum, India

[1] akhilamailid@gmail.com , [2]varunlalr@gmail.com

***Abstract*— License plate or vehicle registration plate is a metal or plastic plate attached to a motor vehicle for identification purposes. The registration identifier is an alphanumeric or numeric code printed on the license plate which uniquely identifies the vehicle within the issuing territory. The size, color and material used for the plates are decided by the licensing authority. The number plate tracking is done by the authorities to identify traffic violation, analysis of theft, parking management, collection of tolls, judicial judgements etc. In the manual inspection, it will be very difficult. So, image-based edge detection is proposed as a replacement. Various methods in image processing to identify the license plate and to extract the unique identification number is discussed, compared and performance analysis is performed using image processing algorithms. The edge detection is such an image processing technique, which is possible with segmentation. Segmentation is a continuously growing area in the field of image processing. It involves various spatial edge detection methods such as first order and second order derivative methods. In this paper, spatial edge detection methods and fuzzy based edge detection method have described and analyzed.**

***Index Terms*— License plate detection, image processing, segmentation, fuzzy based edge detection.**

I. INTRODUCTION

License plate detection is very important in the area of traffic management in almost all countries. The increasing number of vehicles makes the proper identification of license plate a tedious task. The efficient detection and storage of license number plate helps the traffic authorities to understand the traffic condition as well as detection of law violating vehicles. The edge detection in license plate image can be determined by segmentation technique. By detecting the edges it is easy to identify the number belongs to the number plate. An image segmentation method has two types of categories to achieve accuracy. They are based on discontinuity and similarity properties of intensity values of image. The first category is describing a partition of images based on sharp changes in intensity known as edge and the second category is based on a set of predefined criteria known as thresholding, region growing and region splitting and merging. Throughout the length of an edge, it will be creating connected pixels and causes sharp changes in the intensity value between two regions. Since edge detection can be used for detecting the shape of numbers and characters, edge detection in image is very important. Edge detection includes a variety of mathematical methods that aim at identifying points in a digital image at which the image brightness changes sharply. Edge detection is a fundamental tool in image processing, machine vision and computer vision, particularly in the areas of feature detection and feature extraction. Applying an edge detection algorithm to an image may significantly reduce the amount of data to be processed and may therefore filter out information that may be regarded as less relevant, while preserving the important structural properties of an image.

II. EDGE DETECTION TECHNIQUES

Edge detection can be grouped into two categories, search-based and zero-crossing based. The search-based methods detect edges by first computing a measure of edge strength, usually a first-order derivative expression such as the gradient magnitude, and then searching for local directional maxima of the gradient magnitude using a computed estimate of the local orientation of the edge, usually the gradient direction. The zero-crossing based methods search for zero crossings in a second-order derivative expression computed from the image in order to find edges, usually the zero-crossings of the Laplacian or the zero-crossings of a non-linear differential expression. First order derivatives are worked out by using gradient and second order derivatives are found by the Laplacian.

In mathematics, the term gradient of a function means how a function is changing with respect to its arguments or independent variables For a single variable function, we refer to this as the slope. For a 3-variable function f(x,y,z), the gradient, if it exists, is given by

$$\nabla f = \frac{\partial f}{\partial x}\mathbf{i} + \frac{\partial f}{\partial y}\mathbf{j} + \frac{\partial f}{\partial z}\mathbf{k}$$

Thus, the gradient provides two pieces of information – magnitude and direction. The direction of the gradient tells us the direction of greatest increase while the magnitude represents the rate of increase in that direction. At each image point, the gradient vector points in the direction of largest possible intensity increase, and the magnitude corresponds to the rate of change in that direction. Thus for an image f(x,y), the gradient direction and magnitude is given by

Magnitude : $$||\nabla f|| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Direction : $$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$

Image gradient in x-direction measures the horizontal change in intensity while the gradient in y measures the vertical change in intensity. Since edges are an abrupt change in the intensity values thus the largest gradient values will occur across an edge in an image. Thus, the x-gradient will find the vertical edges while y-gradient will highlight the horizontal edges as shown in **Fig 1**.

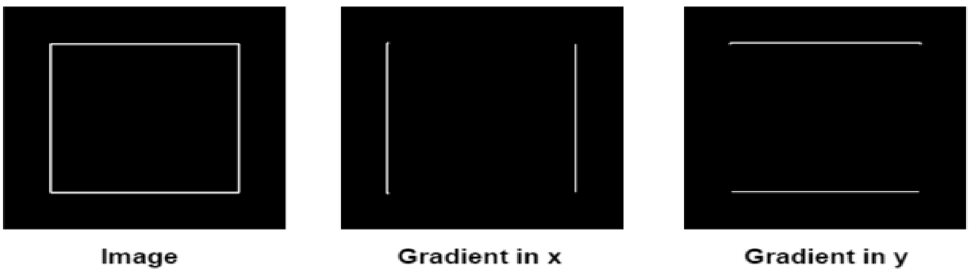


Figure 1

This implies that edges are perpendicular to the gradient direction. That is why gradients are used in edge detection.

Computation of Gradient Components

For the calculation of $\frac{\delta f}{\delta x}$ and $\frac{\partial f}{\partial y}$ at all pixel locations, consider a 3 X 3 image area which will define the neighborhood as below. Here z denotes gray level values.

Z_1	Z_2	Z_3
Z_4	Z_5	Z_6
Z_7	Z_8	Z_9

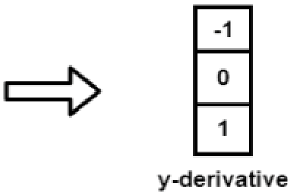
A 3X3 area.

A. First Order Derivative Kernels for Edge Detection

The image is a discrete 2D function and the derivatives exists only for continuous functions, we approximated the image gradients using central finite approximation for one variable as $f'(x) \approx \frac{f(x+0.5\,h)-f(x-0.5h)}{h}$. Using this central difference, we can obtain the derivative filter in x and y directions as shown below

$$f(x,y) = \frac{f(x+h,y) - f(x-h,y)}{2h}$$

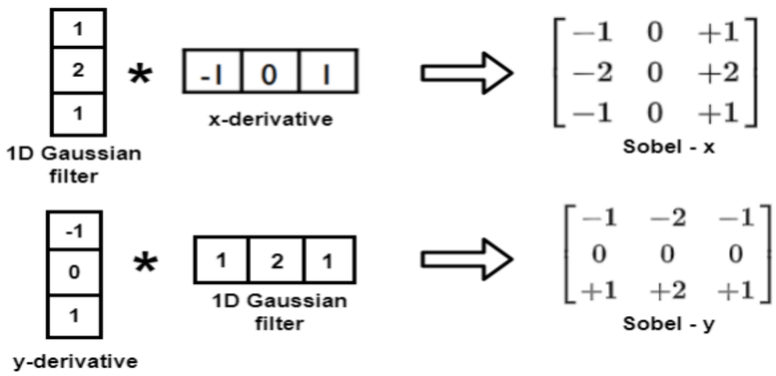
$$f(x,y) = \frac{f(x,y+h) - f(x,y-h)}{2h}$$



Here it is assumed that the *x*-coordinate is increasing in the "right"-direction, and *y*-coordinate in the "down"-direction. By weighting these x and y derivatives, we can obtain different edge detection filters. The first order derivatives of image computed using gradient operators are described below.

Sobel Operator

This is obtained by multiplying the x, and y-derivative filters obtained above with some smoothing filter(1D) in the other direction. For example, a 3×3 Sobel-x and Sobel-y filter can be obtained as



Derivatives in x and y directions are

$G_x = (z_3+2z_6+z_9) – (z_1+2z_4+z_7)$

$G_y = (z_7+2z_8+z_9) – (z_1+2z_2+z_3)$

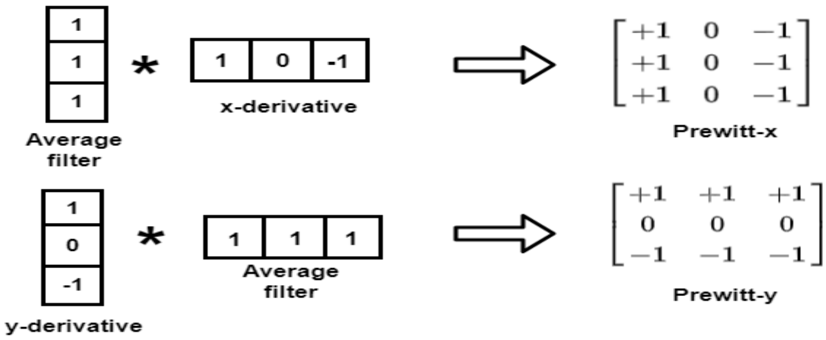
When we convolve these Sobel operators with the image, they estimate the gradients in the x, and y-directions as G_x and G_y. For each point, we can calculate the gradient magnitude and direction as

$$\mathbf{G} = \sqrt{\mathbf{G}_x{}^2 + \mathbf{G}_y{}^2} \qquad \mathbf{\Theta} = \operatorname{atan}\left(\frac{\mathbf{G}_y}{\mathbf{G}_x}\right)$$

Magnitude **Direction**

Prewitt Operator

In this, the x, and y-derivative filters are weighted with the standard averaging filter as shown below.



Derivatives in x and y directions are $G_x = (z_1+z_4+z_7) – (z_3+z_6+z_9)$ and $G_y = (z_1+z_2+z_3) – (z_7+z_8+z_9)$

Robert's cross gradient Operator

These are 2 X 2 masks. They can be represented as below.

-1	0
0	1

-1	0
0	1

By using these masks at point z_5 gives $G_x = z_9 - z_5$ and $G_y = z_8 - z_6$

B. Second Order Derivative Kernels for Edge Detection

A maximum of the first derivative will occur at a zero crossing of the second derivative. To get both horizontal and vertical edges we look at second derivatives in both the *x* and *y* directions.

The Laplacian

The Laplacian of a 2D function gives second order derivative and mathematically the Laplacian is defined as

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

Unlike first-order filters that detect the edges based on local maxima or minima, Laplacian detects the edges at zero crossings i.e. where the value changes from negative to positive and vice-versa. Kernels for Laplacian are similar to the kernels using finite difference approximations for the first-order derivative.

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x) \longrightarrow \underset{\text{x kernel}}{\begin{bmatrix} 1 & -2 & 1 \end{bmatrix}}$$

$$\frac{\partial^2 f}{\partial y^2} = f(y+1) + f(y-1) - 2f(y) \longrightarrow \underset{\text{y kernel}}{\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}}$$

Adding these two kernels together we obtain the Laplacian kernel as shown below

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

This is called a negative Laplacian because the central peak is negative. Other variants of Laplacian can be obtained by weighing the pixels in the diagonal directions also. Make sure that the sum of all kernel elements is zero so that the filter gives zero response in the homogeneous regions.

Properties of the Laplacian

- Unlike first-order that requires two masks for finding edges, Laplacian uses 1 mask but the edge orientation information is lost in Laplacian.
- Laplacian gives better edge localization as compared to first-order.
- Unlike first-order, Laplacian is an isotropic filter i.e. it produces a uniform edge magnitude for all directions

Laplacian Of Gaussian (LoG)

To reduce the noise effect, the image can be first smoothed with a Gaussian filter and then find the zero crossings using Laplacian. This two-step process is called the **Laplacian of Gaussian** (LoG) operation. But this can also be performed in one step. Instead of first smoothing an image with a Gaussian kernel and then taking its Laplace, we can obtain the Laplacian of the Gaussian kernel and then convolve it with the image. This is shown below where f is the image and g is the Gaussian kernel.

$$\nabla^2 (f * g) = f * \nabla^2 g$$

Mathematically, LoG can be written as

$$LoG(x, y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2 + y^2}{2\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

C. **Optimal Edge Detection Algorithms**

Optimal edge detection algorithm used here is Canny edge detection method.

The Canny edge detector

Canny edge detector first smoothens the image to remove noise and then determines the image gradient to highlight regions with high spatial derivatives. It was developed by John F. Canny in 1986. It is a multi-stage algorithm that provides good and reliable detection. Canny edge detector is probably the most commonly used and most effective method. It is having following steps.

- Smooth the image with a Gaussian filter to reduce noise.
- Compute gradient
- Extract edge points: Non-maximum suppression.
- Linking and thresholding: Hysteresis

Non-maximum suppression is trying to relate the edge direction to a direction that can be traced along the edges based on the previously calculated gradient strengths and edge directions. At each pixel location there are four possible directions. Check all directions if the gradient is maximum at this point. Perpendicular pixel values are compared with the value in the edge direction. If their value is lower than the pixel on the edge then they are suppressed. After this step the outputs are broken thin edges that needs to be fixed. Hysteresis is a way of linking the broken lines. This is done by iterating over the pixels and checking if the current pixel is an edge. If it’s an edge then check surrounding area for edges. If they have the same direction then mark them as an edge pixel. Two thresholds can be used for this, high and low. If the pixels are greater than lower threshold it is marked as an edge. Then pixels that are greater than the lower threshold and also are greater than high threshold, are also selected as strong edge pixels. When there are no more changes to the image the procedure is treated as complete.

D. Fuzzy Domain Edge Detection Techniques

Fuzzy logic method is a very important tool in the field of artificial intelligence. Fuzzy models represent mathematical means of representing vagueness. Fuzzy systems use a logic to map an input space to an output space .For doing this a list of ‘if then’ statements called rules are evaluated in parallel .These Rules are useful because they use variables and adjectives that describes those variables. Typical Fuzzy Logic Controller (FLC) consists of a fuzzification module, fuzzy inference engine, defuzzification module and pre- and post-processing modules. Fuzzy image processing has three main stages: image fuzzification, modification of membership values, image defuzzification as shown in **Fig2**. The coding of image data (fuzzification) and decoding of the results (defuzzification) are steps that make possible to process images with fuzzy techniques. The main power of fuzzy image processing is in the middle step (modification of membership values). After the image data are transformed from gray-level plane to the membership plane (fuzzification), appropriate fuzzy techniques modify the membership values.

Figure 2 Fuzzy logic approach for image processing

The fuzzy logic approach for image processing allows to use membership functions to define the degree to which a pixel belongs to an edge or a uniform region. Fuzzy image processing is a powerful tool which gives us the detailed knowledge of edge. An edge is a boundary between two uniform regions. You can detect an edge by comparing the intensity of neighboring pixels. So for detecting crack in glasses we can apply fuzzy logic.

The fuzzy logic edge-detection algorithm used here relies on the image gradient to locate breaks in uniform regions. Calculate the image gradient along the *x*-axis and *y*-axis. Gx and Gy are simple gradient filters. Then convolve image with Gy to obtain the *y*-axis gradients. Then create a Fuzzy Inference System (FIS) for edge detection. Specify the image gradient as the inputs and use a zero-mean Gaussian membership function for each input. If the gradient value for a pixel is 0, then it belongs to the zero-membership function with a degree of 1. Then specify the standard deviation for the zero-membership function for the inputs. Specify the intensity of the edge-detected image and the triangular membership functions. Then we have to include FIS rules. Add rules to make a pixel white if it belongs to a uniform region and make the pixel black if it is not so. Then evaluate the output of the edge detector for each row of pixels. So, the edges in the image are detected using a FIS by comparing the gradient of every pixel in the *x* and *y* directions. If the gradient for a pixel is not zero, then the pixel belongs to an edge (black).

III. RESULTS AND DISCUSSION

In this analysis, license plate images have been chosen from web and Matlab is used for the experimentation of edge detection. The image is processed using various Matlab codes to identify the irregularities. The performance is analyzed using the parameter peak signal to noise ratio (PSNR). The Mathematical equation for PSNR is

$$\mathbf{PSNR=20\,\log_{10}\frac{255}{\sqrt{MSE}}}$$

Where MSE is the mean square error of the image, which can be evaluated pixel by pixel according to the size of image. Usually higher values of PSNR shows better quality results. PSNR values are calculated in dB. The processed results are given in table1.

Table 1 Result analysis

Operating Method	**PSNR values**	
	Image1	Image2
LoG	18.4335	15.0009
Canny	19.2216	14.096
Roberts	14.4283	15.2778
Sobel	18.5839	18.5081
Prewitt	19.9694	18.5269
Fuzzy	21.6776	20.5481

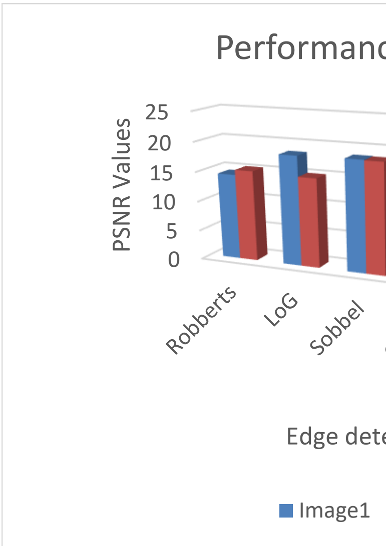
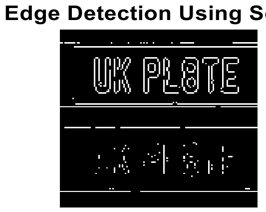
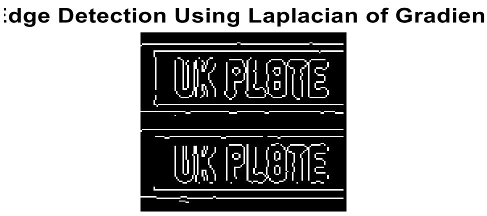
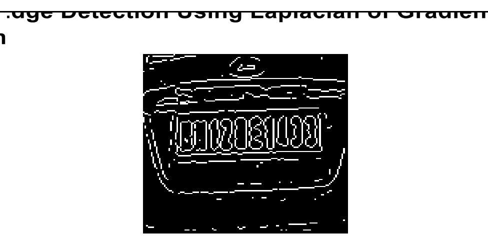
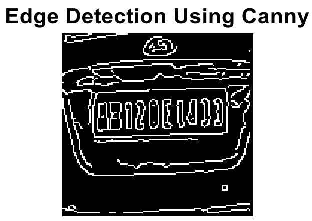


Figure 3 PSNR values for different edge detection methods

Sobel Operator	 
Prewitt	 
Roberts	 
LoG	 
Canny	 

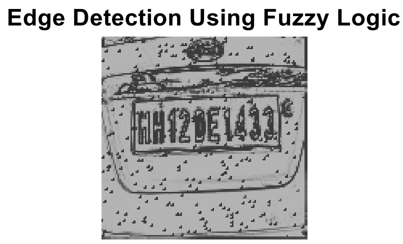
Fuzzy Based Operator	 
Original image	 

Figure 4 Results of various edge detection methods

IV. CONCLUSION

License plate images are analyzed in this paper using edge-based image processing technique. Edge detection is a part of segmentation method, which is implemented with the help of different first order kernels, second order kernels, optimal edge detection algorithm and fuzzy based method. Experimental analysis is performed in Matlab. The parameter used for the analysis was PSNR. From the results obtained, fuzzy based method performs better than the other methods.

V. ACKNOWLEDGMENT

We are expressing our sincere gratitude to all, who supported us for submitting this work.

REFERENCES

[1] S. E. El-Khamy, I. Ghaleb, N. A. El-Yamany, "Fuzzy edge detection using minimum entropy", Proceedings of 11th Mediterranean Electrotechnical Conference MELECON, Cairo, Egypt, 2002.

[2] P. Mamoria, D. Raj, "An Analysis of Images Using Fuzzy Contrast Enhancement Techniques", 3rd 2016 International Conference on Computing for Sustainable Global Development, INDIACom-2016 (IEEE Conference ID: 37465), BVICAM, New Delhi, India.

[3] R. C. Gonzalez and R. E. Woods," Digital Image Processing", 3rd ed. Prentice Hall, 2009.

[4] Rothwell, Ch., J. Mundy, B. Hoffman and V. Nguyen, 1994.Driving Vision by Topology. TR-2444 – Programme 4, INRIA, pp. 1-29.

[5] J. F. Canny, “A computational approach to edge detection”, IEEE Trans. On Pattern Analysis and Machine Intelligence. 8(6), 679/698, 1986.

[6] P. Wu, H.-H. Chen, R.-J. Wu, and D.-F. Shen, “License plate extractionin low resolution video,” in Proc. 18th ICPR, Hong Kong, 2006, vol. 1, pp. 824–827.

[7] Raman Maini and J. S. Sohal, "Performance Evaluation of Prewitt Edge Detector for noisy Images", GVIP Journal, Vol. 6, Issue 3, December 2006.

[8] Bezdek, J. C., R. Chandrasekhan and Y. Attikiouzel, “A geometric approach to edge detection”. IEEE Transactions. On Fuzzy Systems, 6, pp. 52-75, 1998.

[9] J. Cano and J. C. Perez-Cortes, “Vehicle license plate segmentation in natural images,” in Lecture Notes on Computer Science, vol. 2652, F. J. Perales et al., Eds. New York: Springer-Verlag, pp. 142–149, 2003.