CHARACTER RECOGNITION AND TRANSMISSION OF CHARACTERS USING NETWORK SECURITY

Subhash Tatale[1] and Akhil Khare[2]
[1]Student & [2]Assoc. Prof., Deptt. of Info. Tech., Bharti Vidyapeeth Deemed Uni., Pune, India.

ABSTRACT

This paper deals with character recognition of characters of vehicle number plate and these recognized characters are transmitted through secure network channel by using encryption & decryption techniques. This paper includes implementation of automatic number plate recognition, which ensures a process of number plate detection, processes of proper characters segmentation, normalization and recognition also it explains the implementation of respective algorithms. Automatic Number Plate Recognition is a real time embedded system which automatically recognizes the license number of vehicles. In this paper, the implementation of recognizing number plate is considered. After recognizing the characters from number plate by implementing by various algorithms, the characters are transmitted through secure Channel. For Secure transmission of recognized characters i.e. vehicle number, Steganography techniques are used. First recognized characters are embedded into image and that data is encrypted by using private key at sender's end. At the receiving end, the data is extracted from the image by using decryption technique.

KEYWORDS: *artificial intelligence, optical character recognition, encryption, decryption, KNN*

I. INTRODUCTION

Automatic Number Plate Recognition is a mass surveillance system that captures the image of vehicles and recognizes their license number. This project consists of two modules. First module describes the implementation of recognition of vehicle number from vehicle number plate. For this, a process of number plate detection processes of proper characters segmentation, normalization and recognition is used. It also explains the implementation of respective algorithms. In this paper, the implementation of recognizing number plate is considered.

Second module describes transmission of recognized characters i.e. vehicle number through secure network channel. The application of this concept is security and information hiding of the recognized data. For Secure transmission of recognized characters Steganography techniques are used. First recognized characters are embedded into image. An OutGuess algorithm is used to embed the characters into image. This embedded data is encrypted at the senders end and data is transmitted over network. At the receiver end, decryption technique is used to extract original data. A DES algorithm is used for encryption and decryption of data.

II. IMPLEMENTATION OF CHARACTER RECOGNITION

The first step in a process of character recognition of number plate is a detection of a number plate area. After detecting the number plate area the plate is segmented using horizontal projection. Once plate is segmented then characters are extracted from horizontal segments. Extracted characters are normalized by calculating parameters like brightness etc. and recognized using by KNN algorithm. The following describes the implementation of character recognition from number plate of vehicle.

2.1 Edge detection of number plate

Let us define the number plate as a rectangular area with increased occurrence of horizontal and vertical edges. The high density of horizontal and vertical edges on a small area is in many cases caused by contrast characters of a number plate, but not in every case. This process can sometimes detect a wrong area that does not correspond to a number plate. Because of this, we often detect several candidates for the plate by this algorithm, and then we choose the best one by a further heuristic analysis.

2.1.1 Convolution matrices

Each image operation is defined by a convolution matrix. The convolution matrix defines how the specific pixel is affected by neighboring pixels in the process of convolution. Individual cells in the matrix represent the neighbors related to the pixel situated in the centre of the matrix. The pixel represented by the cell y is affected by the pixels $x_0.....x_8$ according to the formula:

$$y = x_0 \text{ x } m_0 + x_1 \text{ x } m_1 + x_2 \text{ x } m_2 + x_3 \text{ x } m_3 + x_4 \text{ x } m_4 + x_5 \text{ x } m_5 + x_6 \text{ x } m_6 + x_7 \text{ x } m_7 + x_8 \text{ x } m_8 \qquad (1)$$

where m represents matrix, x represents row and y represents column.

2.1.2 Horizontal and vertical edge detection

To detect horizontal and vertical edges, we convolve source image with matrices m_{he} and m_{ve}.
The convolution matrices are usually much smaller than the actual image. Also, we can use bigger matrices to detect rougher edges.
In this section, technique of detection of number plate is explained. The edge of the number plate is detected horizontally and vertically.

2.2 Horizontal and Vertical Image Projection

After the series of convolution operations, we can detect an area of the number plate according to a statistics of the snapshot. There are various methods of statistical analysis. One of them is a horizontal and vertical projection of an image into the axes x and y.
The vertical projection of the image is a graph, which represents an overall magnitude of the image according to the axis y. If we compute the vertical projection of the image after the application of the vertical edge detection filter, the magnitude of certain point represents the occurrence of vertical edges at that point. Then, the vertical projection of so transformed image can be used for a vertical localization of the number plate. The horizontal projection represents an overall magnitude of the image mapped to the axis x.

Let an input image be defined by a discrete function $f(x, y)$. Then, a vertical projection p_y of the function f at a point y is a summary of all pixel magnitudes in the yth row of the input image. Similarly, a horizontal projection at a point x of that function is a summary of all magnitudes in the xth column.
We can mathematically define the horizontal and vertical projection as:

$$p_x(x) = \sum_{j=0}^{h-1} f(x, j); \; p_y(y) = \sum_{i=0}^{w-1} f(i, y); \qquad (2)$$

where w and h are dimensions of the image.
The detection of the number plate area consists of a “band clipping” and a “plate clipping”.
The band clipping is an operation, which is used to detect and clip the vertical area of the number plate (so-called band) by analysis of the vertical projection of the snapshot. The plate clipping is a consequent operation, which is used to detect and clip the plate from the band (not from the whole snapshot) by a horizontal analysis of such band.
In this section, horizontal and vertical projection technique is explained. This technique is used for detecting edge of number plate.

2.3 Segmentation of plate using a horizontal projection

Since the segmented plate is deskewed, we can segment it by detecting spaces in its horizontal projection. We often apply the adaptive thresholding filter to enhance an area of the plate before segmentation. The adaptive thresholding is used to separate dark foreground from light background with non-uniform illumination. You can see the number plate area after the thresholding in figure 1.

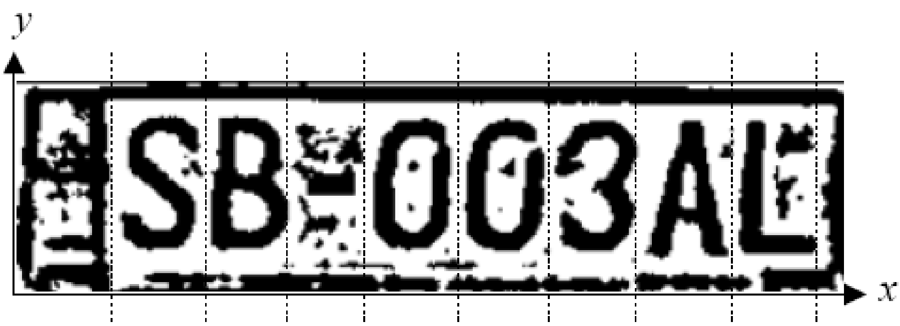

Figure 1: Number plate after application of the adaptive thresholding

After the thresholding, we compute a horizontal projection $p_x(x)$ of the plate $f(x, y)$. We use this projection to determine horizontal boundaries between segmented characters. These boundaries correspond to peaks in the graph of the horizontal projection (figure 2).

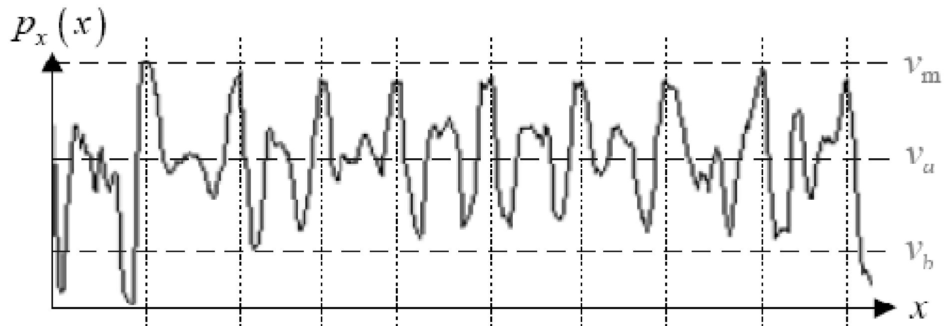

Figure 2: Horizontal projection of plate with detected peaks

The goal of the segmentation algorithm is to find peaks, which correspond to the spaces between characters. At first, there is a need to define several important values in a graph of the horizontal projection $p_x(x)$:

- V_m - The maximum value contained in the horizontal projection $p_x(x)$, such as $v_m = \max\{p_x(x)\}$ where w is a width of the plate in pixels.
- V_a - The average value of horizontal projection $p_x(x)$, such as $v_a = \frac{1}{w}\sum_{x=0}^{w-1} p_x(x)$
- V_b - This value is used as a base for evaluation of peak height. The base value is always calculated as $V_b = 2. V_a - V_m$. The V_a must lie on vertical axis between the values V_b and V_m.

The algorithm of segmentation iteratively finds the maximum peak in the graph of vertical projection. The peak is treated as a space between characters, if it meets some additional conditions, such as height of peak. The algorithm then zeroizes the peak and iteratively repeats this process until no further space is found. This principle can be illustrated by the following steps:

1. Determine the index of the maximum value of horizontal projection: $x_m = \arg\max_{0 \le x \prec w}\{x \mid P_x(x)\}$
2. Detect the left and right foot of the peak as:
$x_l = \arg\max_{0 \le x \prec x_m}\{x \mid P_x(x) \le c_x.P_x(x_m)\}$; $x_r = \arg\min_{0 \le x \prec x_m}\{x \mid P_x(x) \le c_x.P_x(x_m)\}$
3. Zeroize the horizontal projection $p_x(x)$ on interval, $\langle x_l, x_r \rangle$
4. If $p_x(x_m) < c_w \cdot v_m$, go to step 7.
5. Divide the plate horizontally in the point x_m.
6. Go to step 1.
7. End.

In this section, segmentation of number plate is explained. The number plate is segmented by using horizontal projection.

2.4 Extraction of characters from horizontal segments

The segment of plate contains besides the character also redundant space and other undesirable elements. We understand under the term "segment" the part of a number plate determined by a horizontal segmentation algorithm. Since the segment has been processed by an adaptive thresholding filter, it contains only black and white pixels. The neighboring pixels are grouped together into larger pieces, and one of them is a character. Our goal is to divide the segment into the several pieces, and keeps only one piece representing the regular character. This concept is illustrated in figure 3.

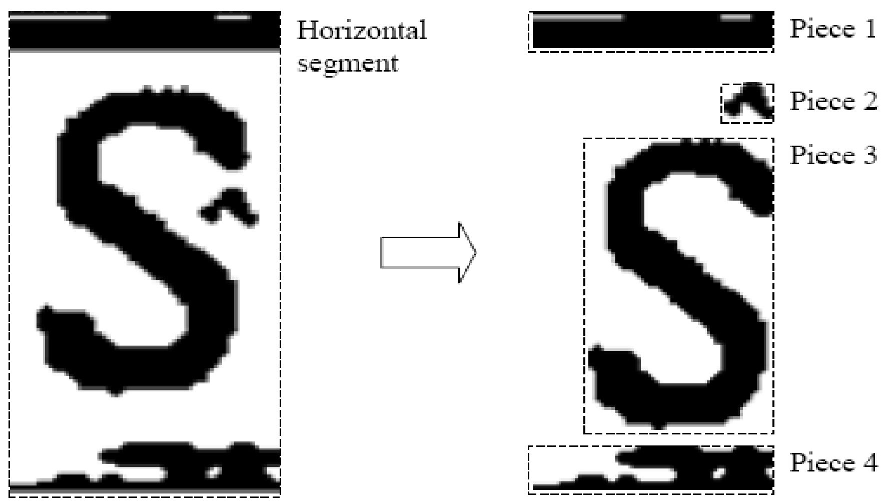

Figure 3: Horizontal segment of the number plate contains several pieces of neighboring pixels.

In this section, how the characters are extracted from horizontal segments are explained once the number plate is segmented.

2.5 Normalization of Characters

To recognize a character from a bitmap representation, there is a need to extract feature descriptors of such bitmap. As an extraction method significantly affects the quality of whole
OCR process, it is very important to extract features, which will be invariant towards the various light conditions, used font type and deformations of characters caused by a skew of the image.
The first step is a normalization of a brightness and contrast of processed image segments.
The second step is the characters contained in the image segments must be then resized to uniform dimensions. The third step is the feature extraction algorithm extracts appropriate descriptors from the normalized characters.

2.5.1 Normalization of brightness and contrast

The brightness and contrast characteristics of segmented characters are varying due to different light conditions during the capture. Because of this, it is necessary to normalize them. There are many different ways, but three most used: histogram normalization, global and adaptive thresholding.
Through the histogram normalization, the intensities of character segments are redistributed on the histogram to obtain the normalized statistics. Techniques of the global and adaptive thresholding are used to obtain monochrome representations of processed character segments. The monochrome (or black & white) representation of image is more appropriate for analysis, because it defines clear boundaries of contained characters.

2.5.2 Normalization of dimensions and resampling

Before extracting feature descriptors from a bitmap representation of a character, it is necessary to normalize it into unified dimensions. The term "resampling" is the process of changing dimensions of the character. As original dimensions of unnormalized characters are usually higher than the normalized ones, the characters are in most cases downsampled. When we downsample, we reduce information contained in the processed image.

There are several methods of resampling, such as the pixel-resize, bilinear interpolation or the weighted-average resampling. We cannot determine which method is the best in general, because the successfulness of particular method depends on many factors. For example, usage of the weighed-average downsampling in combination with a detection of character edges is not a good solution, because this type of downsampling does not preserve sharp edges. Because of this, the problematic of character resampling is closely associated with the problematic of feature extraction.

In this section, methods for normalization of characters are explained. The extracted characters of number plate can be normalized by calculating the brightness and contrast. It also be normalized by representing the dimensions and resampling of characters.

2.6 Character Recognition

After normalization, characters are recognized by using text classification. KNN algorithm is used for text classification. Text categorization also called text classification is the process of identifying the class to which a text document belongs. This generally involves learning, for each class, its representation from a set of characters that are known to be members of that class. KNN algorithm is used to achieve this task. The simplicity of this algorithm makes it efficient with respect to its computation time, but also with respect to the ability for non expert users to use it efficiently, that is, in terms of its prediction rate and the interpretability of the results. This section presents a simple KNN algorithm adapted to text categorization that does aggressive feature selection. This feature selection method allows the removal of features that add no new information given that some other feature highly interacts with them, which would otherwise lead to redundancy, and features with weak prediction capability. Redundancy and irrelevancy could harm a KNN learning algorithm by giving it some unwanted bias, and by adding additional complexity.

2.6.1 KNN algorithm

The main idea of KNN algorithm is that given a testing sample, we can use certain neighbor measure to calculate the neighbor degrees of testing and training samples on training sets, and then classify it with its label of the K nearest neighbor, if its K nearest neighbor contains a number of labels, the samples will be assigned to the majority class of their K nearest neighbor.

The following is the description of KNN algorithm.

a) Describe the training text vector according to the characteristics set, and the weight is always calculated in TF-IDF method.

b) It is necessary to do word segmentation for new text according to feature words, and then describe the vector of new text.

c) Find the K most similar neighbors of the new text among the training documents. To measure the similarity efficiently, we make use of the cosine distance as follows:

$$sim(d_i, d_j) = \frac{\sum_{k=1}^{M} w_{ik} * w_{jk}}{\sqrt{\left(\sum_{k=1}^{M} w_{ik} 2\right)\left(\sum_{k=1}^{M} w_{jk} 2\right)}} \qquad (3)$$

Where, d_i denotes the feature vector of test text, d_j denotes the center vector of j-type, M denotes the dimension of feature vectors, W_k denotes the k-dimension of text feature vector. So far, there is no good way to determine the value k. In general, it has an initial value, and then it will be adjusted according to the results of experiment.

d) In the K nearest neighbors of the new text, then calculate the weight of each category, calculated as follows:

$$p\left(\bar{x}, c_j\right) = \sum sim\left(\bar{x}, d_j\right) y\left(\bar{d}_i, c_j\right) \qquad (4)$$

Where, $\bar{x}$ denotes the feature vector of new text, $sim\left(\bar{x}, d_j\right)$ denotes the above similarity formula, $y\left(\bar{d}_i, c_j\right)$ denotes type function

e) To compare the weight of each category, move the new text to the category of the maximum weight.

In this section, the algorithm for character recognition is explained. To recognize the character k-nearest neighbor (KNN) algorithm is used.

III. CHARACTER TRANSMISSION USING NETWORK SECURITY

In this module, recognized characters of number plate are transmitted over network by using Network Security. Steganography techniques are used to hide the information that is recognized characters. First the recognized characters are embedded into image by using encryption technique. First the image is selected from source location; the recognized characters hide into selected image. Public key is used for encryption at sender side. After sending this image containing characters to the receiver, at the receiver end decryption technique is used to extract characters from image.

Steganography works by replacing bits of useless or unused data in regular computer files (such as graphics, sound, text, HTML, or even floppy disks) with bits of different, invisible information. This hidden information can be plaintext, cipher text, or even images and sound wave. In the field of Steganography, some terminology has developed. The adjectives cover, embedded and stego were defined at the Information Hiding Workshop held in Cambridge, England. The term "cover" is used to describe the original, innocent message, data, audio, still, video and so on. When referring to audio signal Steganography, the cover signal is sometimes called the "host" signal. The information to be hidden in the cover data is known as the "embedded" data. The "stego" data is the data containing both the cover signal and the "embedded" information. Logically, the processing of putting the hidden or embedded data, into the cover data, is sometimes known as embedding. Occasionally, especially when referring to image Steganography, the cover image is known as the container.

3.1 Hiding text message inside image

The following steps show in details the procedure of hiding secret text inside cover image Block diagram (Figure 4).

3.1.1 Preparing container image

1. Convert cover image to streams of binary bits.
2. Use two adjacent bits to hide one character.

3.1.2 Preparing secret text message

1. Convert each character of the secret message to decimal number. Example H = $(72)_{10}$ = $(0100\ 1000)_2$

(a) We take the 4 least significant bits alone; we can do that by perform AND operation:
$(72)_{10}$ AND $(15)_{10}$ = $(0100\ 1000)_2$ AND $(0000\ 1111)_2$ = (0000 1000) = $(8)_{10}$.
(b) We take the 4 upper significant bits alone; we can do that by perform shift operation by 4:

$(72)_{10}$ Shift to right be 4 = $(0000\ 0100)_2$ = $(4)_{10}$

2. Now we can add the secret message to the cover image by applying OR operation.

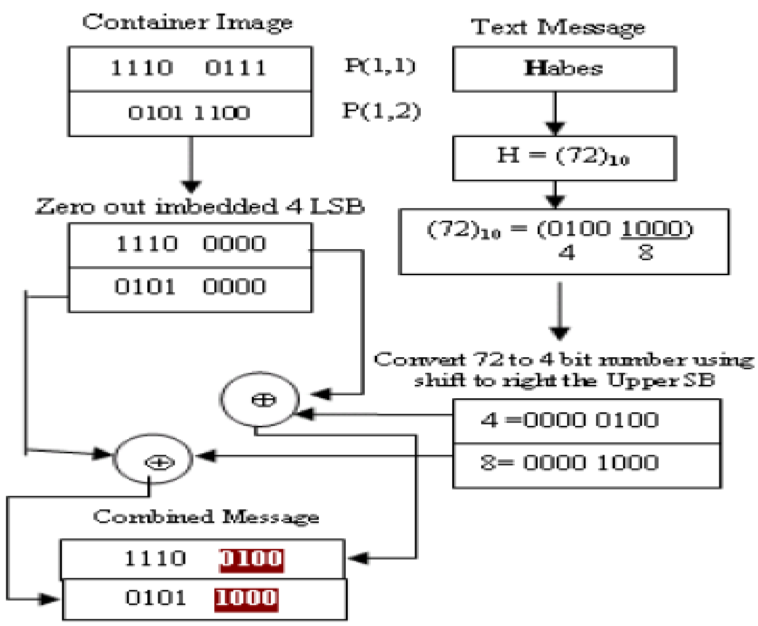

Figure 4: Block diagram to hide text into Image

As shown in the block diagram (figure 4), to hide each character of secret message we need two pixels. So the number of character that we can hide in (n x n) image is given by the following Equation:

$$\text{Numberofcharacters} \leq (n \cdot n) \div 2 - n \qquad (5)$$

In equation (5), we subtract n pixels because we don't set secret text in the first row of cover image; we start setting data from the second row of cover image. The first row of covered image used to store specific data, like position of last pixel in the covered image that contains secret data. The following two equations show how to calculate the pixels that determine of secret text data:

$$\text{Y pos} = \text{length (1strowofimage) modlength (secretmessage)} \times 2 \qquad (6)$$

$$\text{X pos} = (\text{length (secretmessage)} - \text{Y pos}) \div \text{length (1strowofimage)} \qquad (7)$$

3.1.3 Reconstruction the secret text file

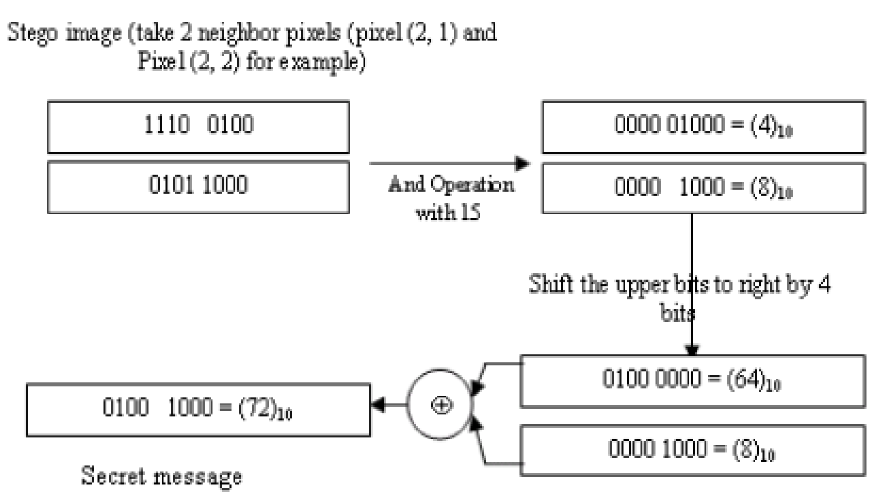

Figure 5: To extract secret message from image

Reconstruction of the secret text message is performed by reversing the process used to insert the secret message in the container image. The following steps describe the details of reconstruction the hidden text file (Figure 5):

1. Take two adjacent pixels from the stego image.
2. Shift the first pixel by 4 to right 1110 0100 shift to right by 4 = $(0100\ 0000)_2$
3. Perform AND operation with 15 to the second pixel

$$(0101\ 1000)\ \text{AND}\ (00001111)_2 = (0000\ 1000)_2$$

4. ADD the result of step 2 and 3 together and we get
$(0100\ 0000)_2 + (0000\ 1000)_2 = (0100\ 1000) = (72)_{10} = \text{H}$.

In this section, the embedding and encryption and decryption methods are explained for security and information hiding of characters.

IV. Results

According to the results, this system gives good responses only to clear plates, because skewed plates and plates with difficult surrounding environment causes significant degradation of recognition abilities.

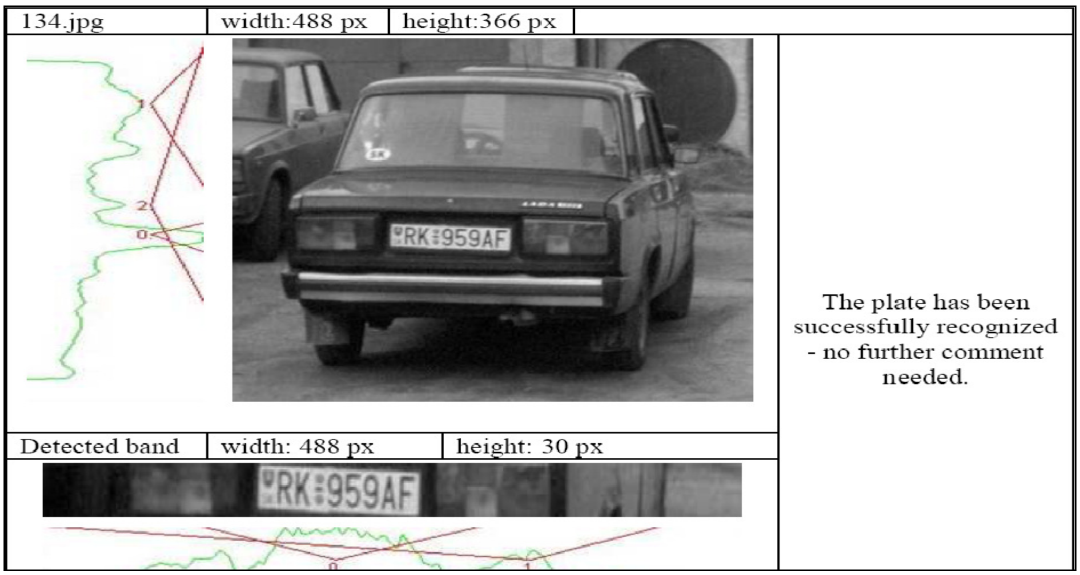

Figure 6: Example of plate recognition.

ANPR solution has been tested on static snapshots of vehicles, which has been divided into several sets according to difficultness.

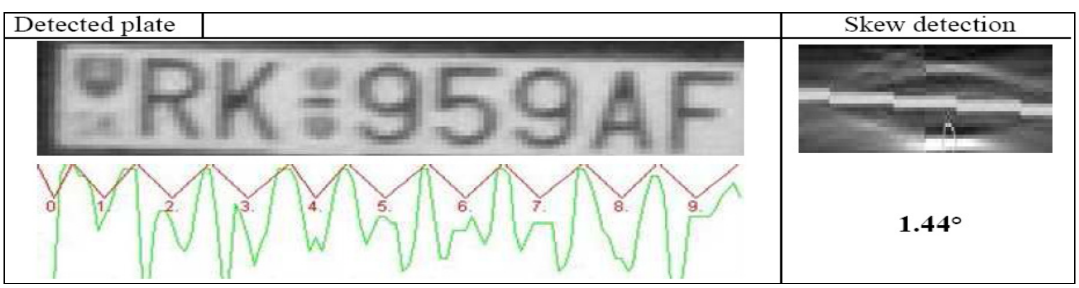

Figure 7: Example of plate detection.

Sets of blurry and skewed snapshots give worse recognition rates than a set of snapshots, which has been captured clearly. The objective of the tests was not to find a one hundred percent recognizable set of snapshots, but to test the invariance of the algorithms on random snapshots systematically classified to the sets according to their properties.

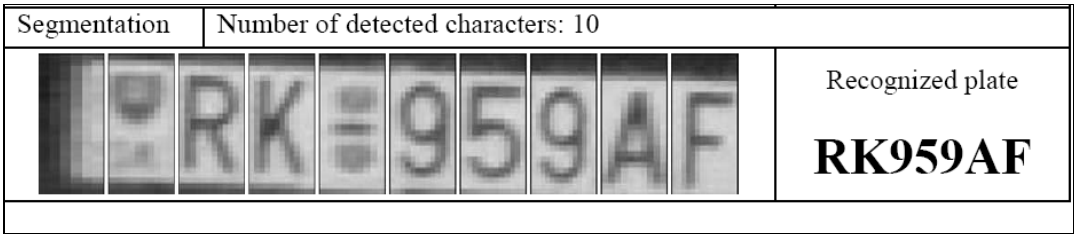

Figure 8: Example of character recognition.

V. CONCLUSIONS

The objective of this paper was to study and resolve algorithmic and mathematical aspects of the automatic number plate recognition systems, such as problematic of machine vision, pattern recognition, and OCR and KNN algorithm. This paper also contains demonstration of ANPR software, which comparatively demonstrates all described algorithms. The various algorithms of recognizing the characters from number plate are explained.

This paper also contains the Steganography techniques which are used for the information hiding and security. The recognized characters are embedded into images; encryption and decryption techniques are used for network security.

REFERENCES

[1] Peter M. Roth, Martin K¨ostinger, Paul Wohlhart, and Horst Bischof, Josef A. Birchbauer (2010): Automatic Detection and Reading of Dangerous, 2010 Seventh IEEE International Conference on Advanced Video and Signal Based Survillance.
[2] Zhiyong Yan, Congfu Xu: Combining KNN Algorithm and Other Classifiers
[3] Ping Dong, Jie-hui Yang, Jun-jun Dong (2006): The Application and Development Perspective of Number Plate Automatic Recognition Technique.
[4] Wenqian Shang, Haibin Zhu, Houkuan Huang, Youli Qu, and Yongmin Lin(IEEE 2006): The Improved ontology kNN Algorithm and its Application
[5] W. K. I. L. Wanniarachchi, D. U. J. Sonnadara and M. K. Jayananda (2007): License Plate Identification Based on Image Processing Techniques, Second International Conference on Industrial and Information Systems.
[6] Zhang Yunliang, Zhu Lijun, Qiao Xiaodong, Zhang Quan: Flexible KNN Algorithm for Text Categorization by Authorship based on Features of Lingual Conceptual Expression, 2009 World Congress on Computer Science and Information Engineering
[7] Ankush Roy Debarshi Patanjali Ghoshal(2011): Number Plate Recognition for Use in Different Countries Using an Improved Segmentation, IEEE.
[8] Yang Jun Li Na Ding Jun: A design and implementation of high-speed 3DES algorithm system, 2009 Second International Conference on Future Information Technology and Management Engineering
[9] YU WANG, ZHENG-OU WANG: A FAST KNN ALGORITHM FOR TEXT CATEGORIZATION , Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, 19-22 August 2007.
[10] Ping Dong, Jie-hui Yang, Jun-jun Dong (2006): The Application and Development Perspective of Number Plate Automatic Recognition Technique, IEEE.
[11] Tingyuan Nie, Teng Zhang: A Study of DES and Blowfish Encryption Algorithm, IEEE 2009.
[12] Muhammad Tahir Qadri, Muhammad Asif (2009): Automatic Number Plate Recognition System For Vehicle Identification Using Optical Character Recognition, International Conference on Education Technology and Computer.
[13] Chen-Chung Liu, Zhi-Chun Luo(2010): An Extraction Algorithm of Vehicle License Plate Numbers Using Pixel Value Projection and License Plate Calibration, International Symposium on Computer, Communication, Control and Automation.
[14] Pletl Szilveszter, Gálfi Csongor(2010): Parking surveillance and number plate recognition application, IEEE 8th International Symposium on Intelligent Systems and Informatics.
[15] Zhihua Chen, Xiutang Geng, Jin Xu: Efficient DNA Sticker Algorithms for DES, IEEE 2008.
[16] C. Sanchez-Avilaf , R. Sanchez-Reillot: The Rijndael Block Cipher (AES Proposal): A Comparison with DES, IEEE 2001.

[17] B. Raveendran Pillai, Prot: (Dr). Sukesh Kumar. A (2008): A Real-time system for the automatic identification of motorcycle - using Artificial Neural Networks, International Conference on Computing, Communication and Networking.
[18] Mohamed El-Adawi, Hesham Abd el Moneim Keshk, Mona Mahmoud Haragi: Automatic license plate recognition.
[19] Luis Salgado, Jose' M. Mene'ndex, Enrique Renddn and Narciso Garcia (1999): Automatic Car Plate Detection and Recognition through Intelligent Vision Engineering, IEEE.
[20] Hwajeong Lee, Daehwan Kim, Daijin Kim, Sung Yang Bang (2003): Real-Time Automatic Vehicle Management System Using Vehicle Tracking and Car Plate Number Identification, IEEE.
[21] B. Raveendran Pillai, Prot: (Dr). Sukesh Kumar. A (2008): A Real-time system for the automatic identification of motorcycle - using Artificial Neural Networks, International Conference on Computing, Communication and Networking.
[22] Mohamed El-Adawi, Hesham Abd el Moneim Keshk, Mona Mahmoud Haragi: Automatic license plate recognition.
[23] A. S. Johnson B. M. Bird, Department of Elect. & Electron. Engineering, University of Bristol: Number-plate Matching for Automatic Vehicle Identification.
[24] Maged M. M. FAHMY: Toward Low Cost Traffic Data collection: Automatic Number-Plate Recognition, The University of Newcastle Upon Tyne Transport Operations Research Group.

Authors Biographies

Subhash Tatale is a M.Tech Student.Having 4 yrs of experience in which 2 yrs of industry and 2 yrs of academic.My reasearch area is Image Processing.

Akhil Khare is an associate professor working in Department of Information Technology.Completed M.Tech.and Pursuing Ph.D. in software engineering field.