Chapter 2

Literature Review

2.1 Facets of vehicle number plate recognition systems

Generally, vehicles are identified by their number plates, which are easily readable by humans, but not by computer. For computer, a number plate is only a grey picture defined as a two-dimensional function $f(x, y)$, where x and y are spatial coordinates, and f is a light intensity at that point. Because of this, it is necessary to design robust mathematical machinery, which will be able to extract semantics from spatial domain of the captured image. These functions are implemented in VNPR systems. VNPR system transforms data between the real environment and the information systems.

The design of VNPR systems is a field of research in artificial intelligence, machine vision, pattern recognition and neural networks.

Image processor recognizes static snapshots captured by the camera, and returns a text representation of the detected license plate. VNPR units can have own dedicated image processors (all-in-one solution), or they can send captured data to a central processing unit for further processing (generic VNPR). The image processor run on special recognition software, which is a key part of whole VNPR system. Because one of the fields of application is the usage on road lanes, it is necessary to use a special camera with the extremely short shutter. Otherwise, quality of captured snapshots will be degraded by an undesired motion blur effect caused by a movement of the vehicle. For example, usage of the standard camera with shutter of 1/100 sec to capture a vehicle with speed of 80 km/h will cause a motion skew in amount of 0.22 m. This skew means the significant degradation of recognition abilities. There is also a need to ensure system invariance towards the light conditions. Normal camera should not be used for capturing snapshots in darkness or night, because it operates in a visible light spectrum. Automatic number plate recognition systems are often based on cameras operating in an infrared band of the light spectrum. Usage of the infrared camera in combination with an infrared illumination is better to achieve this goal. Under the illumination, plates that are made from reflexive material are much more highlighted than rest of the image. This fact makes detection of license plates much easier.

2.2 VNPR Process

The VNPR system that extracts a license plate number from a given image can be composed of four stages [19].The first stage is to acquire the car image using a camera. The parameters of the camera, such as the type of camera, camera resolution, shutter speed, orientation, and light, have to be considered. The second stage is to extract the license plate from the image based on some features, such as the boundary, the color, or the existence of the characters. The third stage is to segment the license plate and extract the characters by projecting their color information, labeling them, or matching their positions with templates. The final stage is to recognize the extracted characters by template matching or using classifiers, such as neural networks and fuzzy classifiers. Figure 2.1 shows the structure of the VNPR process. The performance of a VNPR system relies on the robustness of each individual stage.

A review on each elements of the VNPR structure, shown in Figure 2.1 is described as follows.

2.2.1 Image Acquisition

The study of light has been pursued for centuries. Early work on the subject dates back to Ibn al-Haytham's "Book of Optics" (11th century) and Leonardo

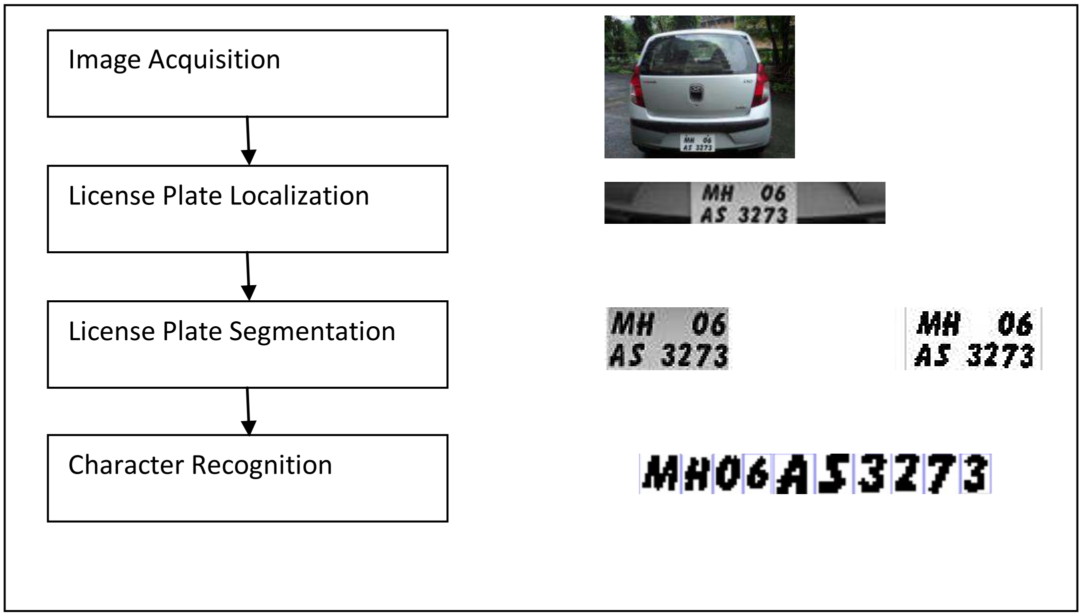

Figure 2.1: Phases in VNPR System.

da Vinci's notebooks (15th century). Only recently, a comprehensive, ray-based model of light encompassing all perceivable properties has been proposed: the plenoptic function [20].

Consider a pinhole camera. Each sample on the sensor plane is a ray passing though the pinhole with a different direction of propagation, but at a fixed position in space. The recorded image is a 2D subset of all light rays in the scene, the full set being a 5D function that models the directional light distribution at all possible positions in 3D space. Radiance of light rays, however, does not change along their paths in free space. Hence, a 4D slice of the plenoptic function fully describes the spatio-angular light variation in a space free of occluders. In computer graphics and vision, this 4D set of rays is referred to as the light field [21, 22]. The camera acquires samples of the plenoptic function [20].The plenoptic function describes the light intensity passing through every viewpoint, in every direction, for all time, and for every wavelength. Thus, the samples of the plenoptic function can be used to reconstruct a view of reality at the decoder.

As opposed to light fields, the full 7D plenoptic function also considers the wavelength of light as well as temporal variation. Polarization and phase are usually disregarded, because these properties are associated with wave-based light models. The multi-spectral light fields are acquired by mounting a custom multi-spectral camera on a programmable X-Y translation stage [23]. Each of the different light field views is colour-coded with one of the acquired spectral channels. The larger dataset contains 15x15 perspectives of a scene, each containing 23 narrow-band colour channels ranging 460 nm to 680 nm. Each channel has a spatial resolution of 341x512 and is stored as a high dynamic range image composited from multiple exposures. Without image compression, this dataset alone requires 27 GB of storage and about 16 hours to be captured. The shear size and the amount of required resources make it difficult to capture and process any such plenoptic data. Most acquisition setups therefore only consider subsets of the plenoptic function.

Limited resources for sensing and processing are not only a problem for machine vision, but also for the visual systems of animals. Natural evolution has resulted in a variety of different tradeoffs in these systems, each of which is optimized for survival in its natural environment. Animals with compound eyes, for example, often have an impressive spectral and temporal resolving power, but a limited spatial resolution [24]. The human visual system (HVS), on the other hand, has its own limitations. The tristimulus nature of human colour perception severely limits our ability to resolve spectral distributions. Without adaptation, the maximum contrast that the HVS can resolve at any given time is about 10,000:1 [25]. Humans have a temporal resolving power of about 30 images per second [26] and sample the plenoptic function at the two different positions of the eyes. Unlike mantis shrimps, we are not, to any significant amount, sensitive to polarization. Standard sensors integrate over all of the dimensions of the plenoptic function. In order to record specific aspects, however, a lot of research has been done which will be outlined in the following.

2.2.1.1 High Dynamic Range Imaging

High dynamic range (HDR) image acquisition has been a very active area of research for more than a decade. With the introduction of the HDR display prototype [27] and its successor models becoming consumer products today, the demand for highcontrast photographic material is ever increasing. For a comprehensive overview of HDR imaging, including applications, radiometry, perception, data formats, tone reproduction, and display, the reader is referred to the textbooks by Reinhard *et al.* [25, 28].

2.2.1.2 Spectral Imaging

The RGB image contains three gray scale channel images which are acquired through 3 filters. The spectral image can contain multiple gray scale channel images, which can be acquired through narrow band filters. Each channel image contains information about one narrow spectral channel band. The spectral image can be converted to other color spaces such as RGB using common conversion algorithms [29, 30]. Spectral image acquisition can be done using three different approaches. In the first approach, the spectral image is captured line by line [31]. Each pixel on each single line contains the full spectral information of the target. The spatial domain can be captured by moving the target or the camera. In the second method, the spectral image is captured by using a filter, for example, a Liquid Crystal Tunable Filter (LCTF) [32]. Each channel image is captured with a different filter transmittance. This approach captures x and y spatial domains at once for each wavelength channel. In the third approach, the spectral image is formed by capturing the spectral and spatial domains at the same time [33]. The measured image is divided into multiple sub images by using optical elements. A different wavelength region with full spatial information goes to different places on the CCD-cell. The spectral image can be reconstructed from the divided sub images. Each method has its own benefits. The first approach is much better for the industrial line applications where targets cannot be stopped and where the imaging has to be done in real time, as in a conveyor belt production. The second approach provides quite fast image capture, but the spectral resolution is not as good as in the first approach and it is not so convenient for moving targets. However, the scanning of the camera or the object is not needed. The third approach is the fastest, but in this case usually the spatial or the spectral resolution is poor.

2.2.1.3 Light field acquisition

The concept of a light field predates its introduction in computer graphics. The term itself dates to the work of Gershun [34], who derived closed-form expressions for illumination patterns projected by area light sources. Ashdown [35] continued this line of research. Moon and Spencer [36] introduced the equivalent concept of a photic field and applied it to topics spanning lighting design, photography, and solar heating. The concept of a light field is similar to epipolar volumes in computer vision [37]. As demonstrated by Halle [38], both epipolar volumes and holographic stereograms can be captured by uniform camera translations. The concept of capturing a 4D light field, for example by translating a single camera [21, 22] or by using an array of cameras [39], is predated by integral photography [40], parallax panoramagrams [41], and holography [42].

2.2.1.4 Phase and Fluid Imaging

Fluid imaging is a wide and active area of research. Generally, approaches to measure fluid flows can be categorized into optical and non-optical methods. An extensive overview of fluid imaging techniques can be found in the book by Merzkirch [43].

2.2.2 License Plate Localization

The second step in a process of automatic number plate recognition is a detection of a number plate area. This problem includes algorithms that are able to detect a rectangular area of the number plate in an original image. Humans define a number plate in a natural language as a "small plastic or metal plate attached to a vehicle for official identification purposes", but machines do not understand this definition as well as they do not understand what "vehicle", "road", or whatever else is. Because of this, there is a need to find an alternative definition of a number plate based on descriptors that will be comprehensible for machines. The number plate is 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 sometime detect a wrong area that does not correspond to a number plate. Hence, generally several candidates for the plate are detected by this algorithm, and then best one is chosen by a further heuristic analysis. These several candidates are known as features of the license plate. Processing of each pixel in the car image, for plate recognition, requires lot of computational effort and hence the license plates are identified by its features so that the processing of only those pixels which are the part of the license plate are required. These features are derived from the various formats of license plates and the style of texts, as given in section 2.2. The existing classification of extraction methods based on the features is as follows.

2.2.2.1 Boundary information

Since the license plate normally has a rectangular shape with a known aspect ratio, it can be extracted by finding all possible rectangles in the image. Edge detection methods are commonly used to find these rectangles. A periodical convolution of the function f with specific types of matrices m is used to detect various types of edges in an image:

$$f'(x,y) = f(x,y)\tilde{*}m[x,y] = \sum_{i=0}^{w-1}\sum_{j=0}^{h-1} f(x,y).m[mod_w(x-i), mod_h(y-j)]$$

where w and h are dimensions of the image represented by the function f.

Convolution

Convolutions are commonly used in a wide array of engineering and mathematical applications. A simple high level explanation is basically taking one matrix (the image) and passing it through another matrix (the convolution matrix). The result is convoluted image. The matrix can also be called the filter. 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 in the destination image, figure 2.2, is affected by the pixels $x_0, \ldots, x_8$ according to the formula:

$$y = x_0 \times m_0 + x_1 \times m_1 + x_2 \times m_2 + x_3 \times m_3 + x_4 \times m_4 + x_5 \times m_5 + x_6 \times m_6 + x_7 \times m_7 + x_8 \times m_8$$

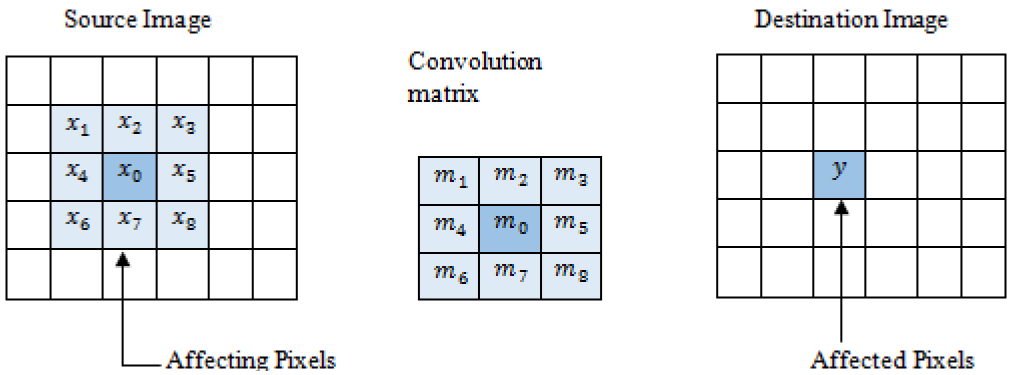

Figure 2.2: The pixel is affected by its neighbors according to the convolution matrix.

where m_0, m_1,. . . , m_8 are elements of convolution matrix.

Horizontal and vertical edge detection

Edge detection is a common image processing technique used in feature detection and extraction.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.

$$m_{he} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} ; m_{ve} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Sobel Edge Detection

The amount of data needed to be processed at a later phase, while maintaining the important structure of the image, can be significantly reduced by applying edge detection. The idea is to remove everything from the image except the pixels that are part of an edge. These edges have special properties, such as corners, lines, curves, etc. A collection of these properties or features can be used to accomplish a bigger picture, such as image recognition. An edge can be identified by significant local changes of intensity in an image [45]. An edge usually divides two different regions of an image. Most edge detection algorithms work best on an image that has the noise removal procedure already applied. The main ones existing today are techniques using differential operators and high pass filtration.

A simple edge detection algorithm is to apply to the Sobel edge detection algorithm. It involves convolving the image using a integer value filter, which is both simple and computationally inexpensive.

The Sobel filter is defined as:

$$S_1 = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, S_2 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

To apply the sobel algorithm on an image, we first find the approximate derivatives with respect to the horizontal and vertical directions. Let A be the original image, G_x be the derivative approximation on the horizontal axis and G_y be the derivative approximation on the vertical axis. Then

$$G_x = S_1.A$$
$$G_y = S_2.A$$

The resulting gradient image is the combination of G_x and G_y. Each pixel $G(x, y)$ of the resulting image can be calculated by taking the magnitude of G_x and G_y:

$$G(x, y) = \sqrt{G_x^2 + G_y^2}$$

The gradients direction is calculated by:

$$\theta = \arctan \frac{G_y}{G_x}$$

Finally, to determine whether a pixel of the original image A is part of an edge, the following relationship is applied:

if $G(x, y) >$ threshold, then $A(x, y)$ is part of an edge.

Some approaches have been proposed by authors in [46], [47], [48-51] to detect the edge of the license plate based on the Sobel filter. The edge of the license plate is identified based on the color difference between the license plate and the car body. While performing horizontal edge detection, the edges are two horizontal lines and similarly while performing vertical edge detection; the edges are two vertical lines. Both the operations are done at the same time to identify the boundary of the license plate. Nelson and Nallaperumal [52] have proposed an approach to detect license plate by using the geometric attribute for locating license to form rectangle. In [46-48] and [53], candidate regions are generated by matching between vertical edges only. The magnitude of the vertical edges on the license plate is considered a robust extraction feature, while using the horizontal edges only, it can result in errors due to car bumper [54]. Some approaches are block-based [55], in which blocks with high edge magnitudes are identified as license plate areas. The accuracy of 180 pairs of images is 92.5 %.

Canny is one of widely used edge detectors, based on first-order derivatives. A Compute Unified Device Architecture(CUDA)implementation of Canny operator is described in [59]. Open GL and Cg versions were presented by Ruiz *et al.*[60]. A more detailed study on Canny operator was presented by Luo and Duraiswami [61].The algorithm was implemented in CUDA and subsequently integrated in MATLAB, achieving approximately 3× speedup over an optimized CPU implementation in the OpenCV library, tested on Intel Core 2 Duo 6600 and NVidia GeForce 8800 GTX. Kong *et al.* [62] presented GPU-accelerated procedure for MATLAB, performing the Prewitt edge detector.

Color Edge detection Another distinctive feature of the license plate is that it contains many edges from white to black and vice versa. An edge detection algorithm finds sharp shifts in intensity of the images, i.e. edges. When checking for edges that are strong in all color layers, primarily the black/white edges are found. The color edge detection is performed as a convolution between the image and the horizontal mask, h = [0.5 0 -0.5]. This means that it traverses the picture row by row, takes the value for the preceding pixel and subtracts the value for the following pixel. If the values of all the three color layers mark an edge, then the pixel will be filled. The horizontal mask mark vertical edges only, and this gives less marked edges than a 2-D algorthim. The plate does however contain many vertical edges and by only searching for edges in one dimension, much computational time is saved. By only marking black/white edges, less relevant edges are excluded.

Mathematically an edge can be described as a transition from high intensity to low or the reversed. If an edge is plotted as an intensity histogram, it looks like a slope. The derivative can be used to find the slope of a curve. The maximum gradient in the image intensity curve is equal to the maximum of the first derivative, or equal to the zero-crossings of the second derivative's curve. The zero-crossings of the second derivative even highlight edges from high to low intensity. For a discrete source such as a pixel image, the changes in intensity can be approximated by calculating the change in magnitude, Δy over Δx, instead of the derivative. In equation 2.1 this approximation is shown. This is equal to a one dimensional convolution with the mask h=[0.5 0-0.5], if Δx has the value 1, which is the smallest step in a discrete grid. The generalized discrete function of convolution is given by equation 2.2, where f is the function and h is the convolution kernel. This is similar to a one-dimensional version of the Sobel edge detection [63]. Since the image source is positive real-valued the mask h=[1 0-1] is used, and the output will still be within the 8 bit boundaries.

$$\frac{\partial f(x,y)}{\partial x} = \lim_{x\to\infty} \frac{f(x+h,y)-f(x,y)}{h} \approx \frac{f(x+\Delta x,y)-f(x-\Delta x,y)}{2\Delta x} \tag{2.1}$$

$$f'(x,y) = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} h(k,l) f(x-k,y-l) \tag{2.2}$$

Filtering the image with this mask is equal to assigning every pixel the difference of its two closest neighbors in the row. The color image is filtered over all three color-layers and the resulting value from each layer is compared with the sign of the edge value has to be same for all layers. The result is that only edges that are reasonable black/white are marked and e.g. red/black edges are suppressed. With this mask, only positive edges are found, i.e. from white to black, but not from black to white. This would not be a big problem for this mask, except that the output would be asymmetric to the license plate. The dissymmetry would though be static and easy to compensate for. However, in the latest versions of the implementation, the absolute value is used so that even negative edges are counted. The edges are also filtered with respect to the magnitude so that edges with small noise are discarded.

$$\Delta f_c = p(x+1, y) - p(x-1, y); \quad \text{(c=R,G,B)}$$

$$e(x,y) = \begin{cases} 1 & (\sin(\Delta f_R)\text{=}\sin(\Delta f_G)\text{=} \sin(\Delta f_B)\text{and} \min(|\Delta f_R|, |\Delta f_G|, |\Delta f_B|) > 25) \\ 0 & \text{otherwise} \end{cases}$$

where $p(x, y)$ represent the pixel value at the point (x, y), each pixel value holds the three values for each color intensity (R, G, B), $e(x, y)$ represents the binary output value. The threshold of minimum edge strength filtering is set to 25. The exact value of the threshold is not very sensitive. For good quality images, it can be varied rather much, ±20, while still being able to detect the plate. Noisy low contrast images are more sensitive though; a large value suppresses noise edges, but a low contrast images produces weaker edges, why some might be missed. The selected value is a good compromise between these effects.

2.2.2.2 Texture

Texture is a regional property on which the internal representation of an image is based. Several approaches based on the texture have been proposed by various authors [64]-[71], which depends on the presence of characters in the license plate, and on the difference in the grey-scale level between the colors of the text and background color of the license plate. In some approaches [71, 72], scan-line techniques are used. The change of the grey-scale level results in a number of peaks in the scan line. This number equals the number of the characters. R. Zunino and S. Rovetta used vector quantization to identify the text in the image.

Haralick proposed co-occurrence matrix, which is also referred to as a co-occurrence distribution [77]. In this method, distribution of co-occurring values in an image at a given set are calculated in order to form a co-occurrence matrix, and several features are extracted from this matrix for texture recognition task. Co-occurrence matrix is sensitive to spatial frequencies of the texture, however it is not recommended for textures with large primitives. In [78], second and third order statistics are used directly to differentiate texture images. However this method cannot be used to differentiate a large database of texture images. Lin *et al.* [79] adopted a structural approach for texture retrieval from an image database, rather than using frequency domain methods.

In [80], covariance matrix as a new region descriptor for texture recognition task is introduced. Covariance matrix is shown to outperform the previous feature extraction methods in several texture retrieval and classification experiments.

2.2.2.3 Color

Color of the license plate is governed by the transport authorities of respective countries. Hence the license plates can be identified by locating the color of license plates in the image. The color combination of a plate and text is unique [81], which forms a feature for license plate extraction. Shi *et al.* [81], authors devised an approach for license plate extraction based on color feature, proposing that all the pixels in the input image are classified using the hue, lightness, and saturation (HLS) color model into 13 categories. Some authors use neural network to identify the color of each pixel after converting the RGB image into HLS. Neural network outputs are green, red and white in colors for the license plates in Korea. The same license plate color is projected vertically and horizontally to determine the highest color density region that is the license plate region. In [82-83], authors have used Genetic algorithm to locate the license plate color. It determines upper and lower thresholds for the plate color in varying lighting conditions. In this approach a function is used to describe the relation between the average brightness and these thresholds. In the image, any pixel with a value between these thresholds is labeled. The region is considered as license plate region if the connectivity of the labeled pixels is rectangular with the same aspect ratio of the license plate. In [84] and [85], authors use mean shift algorithm into candidate regions to identify the location of the license plate in the image. The plate identification methodologies, based on color feature suffer from illumination variation problem. Hence Wang *et al.* [86] devised a methodology for localization of the license plate that is based on fuzzy logic. In this method the components of the color- hue, saturation and value are considered. All these three components are first mapped to fuzzy sets according to different membership functions and thereafter, the fuzzy classification function is described by the fusion of three weighted membership degrees.

2.2.2.4 Global Image Information

In the papers [87-90], authors use connected component analysis for binary image processing. In this approach, a binary image is scanned and its pixels are labeled into components based on pixel connectivity. Chacon and Zimmerman [91] use contour detection algorithm on the binary image to detect the connected objects and which have same geometrical features as license plate are considered to be the candidates. Miyamoto *et al.* [132], use 2D cross correlation with a pre-stored license plate template to locate the license plate but it's very time consuming, in the order of n^4 for $n \times n$ pixels.

2.2.2.5 Character features

There are several approaches to extract the license plate which are based on locating its characters. Such methods look for the character region by identifying the characters. Matas and Zimmermann [89] use region based approach to propose an algorithm to find the character like regions in the image. The classification of the regions is done by neural network. The license plate region is declared if the combination of character-like regions is found. Draghici [94] proposed an approach for license plate recognition using neural networks, which is based on character feature. It scans the image horizontally and looks for contrast changes at the interval of 15 pixels. But the assumption was made that there is sufficient contrast between the characters and the background to localize the license plate. It also assumes that there are at least 3-4 characters, having height of at least 15 pixels. In [95], scale-space analysis is done to identify the license plate characters for its localization. In this approach, large size blob-type figures are extracted that contain line-type figures as character candidates. Cho *et al.* [96] recognize characters by their widths and the difference between background and character region. Then distance between the adjacent characters is found in the plate region to localize the plate. Its extraction rate is 99.5%. In [97], authors have proposed a methodology for extraction of license plate using AdaBoost and SVM. In this there are two stages. In the first stage, an initial set of possible character region is obtained. The second stage takes the inputs from the first stage and reject non-character regions. In first stage 36 AdaBoost classifiers are used and in second stage SVM trained on scale-invariant feature transform descriptors for identifying the character regions. In Lim and Tray [98] maximally stable extreme regions are used to obtain a set of character regions. Highly unlike regions are removed with a simplistic heuristic-based filter. The remaining regions with sufficient positively classified SIFT key points are retained as the likely license plate regions.

The methods based on the character features, for license plate identification are time consuming because the algorithm has to scan and process all binary objects. Also, they fail when the image contains text, other than plate region also.

2.2.2.6 Histogram of Oriented Gradient (HOG) features

Histogram of Oriented Gradient descriptors provides excellent performance in object detection over other existing feature sets including Haar wavelets [99, 100]. HOG descriptors are reminiscent of edge orientation histograms, SIFT descriptors and shape contexts, but they are computed on a dense grid of uniformly spaced cells and they use overlapping local contrast normalizations for improved performance [101]. The HOG descriptors extract the edge or gradient information which is very characteristic of local shapes, and performs with a controllable degree of invariance to local geometric and photometric transformations. There are different types of HOG descriptors like Rectangular HOG, Circular HOG, Bar HOG, and Centre-Surround HOG. Rectangular HOG is most conveniently to be implemented among all these descriptors. To search the location of license plates in an image, a scanning window needs to be defined and slide over the image at different scales and locations. In each location, the HOG feature will be extracted from the window and sent to the SVM classifier which will decide if the region is a license plate or not. The regions with large positive response from SVM classifier will be collected as potential detection, and fused with non-maximum suppression algorithms into one final detection result. The step size of the sliding window is related to the algorithm efficiency and detection precision.

2.2.2.7 Eigenspace approach

Turk and Pentland (1991a) introduced the use of eigenfaces for face recognition. They used a set of training samples to formulate a lower dimensional object space or face space by finding the vectors which best account for the distribution of the samples (PCA). An image from a test set may then be projected onto this space and classified using a minimum distance classifier. This provides information on how close the test image is to the face space, and may then be used to accept or reject the test image as a candidate face, as well as assign a class to the test image [102]. The problem of face recognition is similar to that of VNPR in that both image representations share major features (e.g. strokes and edges) and minor ones (e.g. tone and texture). The definition of an average face or license plate template in principal component space provides a characteristic texture or similarity grouping. Furthermore, in both algorithm complexity and computation times are reduced by the lower dimensionality representation provided by this approach, while still maintaining great accuracy in recognition. As a consequence, the possibility of adapting this method to the detection of license plates is explored.

2.2.2.8 Connected Component (CC) Based Methods

Connected component based methods use a bottom-up approach by grouping pixels and components into larger components until all the regions are identified. Geometrical analysis and spatial arrangement features are used to filter the non-text components and mark the text regions. Although each CC-based method utilizes different approaches, there are four common steps:

1. preprocessing, such as color clustering and noise reduction
2. connected component generation
3. filtering non-textual components
4. component grouping

Zhong [103] used a CC-based method which uses color reduction. This method assumes that text regions occupy a significant portion of an image. Shim *et al.* [104] utilized intensity homogeneity of text regions and grouped the pixels with similar intensity values. After removing large objects that are marked as backgrounds, each candidate region is subjected to verification using size, area, fill factor and contrast. Lienhart *et al.* [105] regarded text regions as connected components with the same or similar color and size, and applied motion analysis to enhance the results. Segments, which are too small or too large, are filtered out. Due to their relatively simple implementation, CC-based methods are widely used in text information extraction. On the other hand, CC-based methods can segment a character into multiple components in case of polychrome texts and low resolution noisy images. Moreover, several input-dependent threshold values are required to filter out non-text components on the image.

Horizontal and vertical rank filtering

Horizontally and vertically oriented rank filters are often used to detect clusters of high density of bright edges in the area of the number plate. The width of the horizontally oriented rank filter matrix is much larger than the height of the matrix ($w >> h$), and vice versa for the vertical rank filter ($w << h$)

To preserve the global intensity of an image, it is necessary to each pixel be replaced with an average pixel intensity in the area covered by the rank filter matrix. In general, the convolution matrix should meet the following condition:

$$\sum_{i=0}^{w-1} \sum_{j-1}^{h-1} m_{hr}[i,j] = 1.0$$

where w and h are dimensions of the matrix and m_{hr} elements of horizontal rank matrix.

2.2.3 License Plate Segmentation

Image segmentation is the process of reducing processing complexity by examining only those regions in the image that is of interest. An attempt is made to separate useful information from the background. Segmentation implies a higher level description of the image than that provided by the original grey level pixels. It separates 'raw' image into different areas which can be represented by parameters other than grey levels. These parameters are known as features.

After locating the license plate from an image, it needs to be segmented to extract the characters for recognition. There is a pre-processing step in segmentation, which is responsible to overcome the problems of the output data from the previous phase: 'License Plate Localization'. The problems may include license plate sizing, orientation, non-uniform brightness, etc.

This process is highly dependent on the format of the plate being processed. Because different countries 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 [106] have dark background with character in lighter color, the algorithm used to recognize characters here cannot be directly applied to the plates in Alberta, Canada [107], 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 [107], or by using an adaptive clustering technique by finding white spaces between columns of higher density of dark pixels [108]. However, these techniques do not take into account that sometimes there could be frames that are partially connected to the plate content. The use of simple projection or clustering techniques will not yield adequate results.

In [109] and [110], authors use bilinear transformation to map the tilted extracted license plate to a straight rectangle. Pan *et al.* [111] used a least square method to treat horizontal tilt and vertical tilt in license plate images. In [112], according to Karhunen-Loeve transform, the coordinates of characters are arranged into a 2-D covariance matrix. After this the eigenvector and the rotation angle α are computed and in turn image tilt correction is performed. The vertical tilt angle is found by K-L transform, the line fitting based on K-means clustering and the line fitting based on least squares. Deb *et al.* [113] introduced a line fitting method based on least-squares fitting with perpendicular offsets for correcting a license plate tilt in the horizontal direction. To correct the tiltness in vertical direction, the variance of coordinates of the projection points is minimized. Character segmentation is performed after horizontal correction and character points are projected along the vertical direction after shear transform.

Binarization and CCA are used for segmentation. Binarization is the process of labeling pixels foreground or background. Optimal thresholds are computed based upon local or global histogram analysis. The pixels above the thresholds are classified as foreground, and the rest as background. Binarization methods may be fully automatic or may require manual inspection and fine-tuning of parameters [114]. For several special US license plate designs, binarization algorithms fail to segment the characters correctly. If the binarization - threshold is inaccurate, it may lead joined characters and whose segmentation is difficult [115]. Comelli *et al.* [116] claim that the license plates which have frames, are difficult to segment because some characters may get join with the frame. They propose a methodology to conduct gradient analysis on the whole image to detect the license plate, which is then enhanced by grey level transformation.

The methods for license plate segmentation are broadly classified based on the features they used as follows.

2.2.3.1 Segmentation of plate using binary image processing

Several approaches are proposed based in binary image processing that includes horizontal and vertical projections [117-119] and mathematical morphology [119, 120].

2.2.3.2 Segmentation of plate using Grey scale processing

Several techniques have been proposed by different authors for segmentation using grey scale processing. The most common techniques are adaptive threshold [12, 13], histogram processing [129, 71], and using classifiers [89, 90].

2.2.4 Character Recognition

In this final phase, the characters of the license plate are recognized to identify the license plate number. There are some issues involved to identify the characters, which are quality of camera, variant in character size [131], [116], noise [131] or tilted plate. Normalization, smoothing, thinning are the techniques to overcome these issues. After feature extraction, different types of neural networks, like Hidden Markov Model [92], SVM [133, 121], and ANN [11, 15, 121, 134, 135] are commonly used to classify the characters. Template matching [19,107] is also most commonly used technique to recognize the characters. Some authors use hybrid approach [136-140].