License Plate Localization, Segmentation and Recognition

Amod Agrawal
IIIT-Delhi
New Delhi, India
amod13125@iiitd.ac.in

Protichi Basak
IIIT-Delhi
New Delhi, India
protichi13075@iiitd.ac.in

Abstract

In this project we build a robust Automatic License Plate Detection (ALPR) system. ALPR involves localization, segmentation and recognition of License Plate. We explore a plethora of options for each of the major steps. We try to combine certain methods to achieve higher accuracy on our data set. We discover that localization is the bottleneck for accuracy of the system. We explain the complete process and pipeline involved in our system. We also talk about all the techniques that we have tried.At the end of the paper, we talk about it's accuracy on each step - localization and recognition.

1. MOTIVATION

Automatic License Plate Detection (ALPR) systems based on localization, segmentation and recognition of license plates from images, form a fundamental problem in computer vision. Rapid developments in Image Processing and need for faster, cheaper and more efficient systems in automotive security and steering systems has resulted in development of sophisticated solutions to this problem. A solution to such problem can be applied in many areas and can form a basis of automatic systems providing access to protected areas, route traffic monitoring for offenses and traffic system violations. Most countries have an elaborate network of surveillance camera throughout cities and yet dont have systems to automatize car identifying process. Solutions are still expensive and not deployed at a very large scale.

2. PROBLEM STATEMENT

The problem statement is to localize License Plates in images taken under a uncontrolled environment and identify the characters on the plate.

3. PREVIOUS WORK DONE

License plate recognition has three main steps: 1) Locating License Plate(LP) region 2) Segmenting the characters of the LP 3) Recognition of each character. The accuracy of the last two steps depends on how well the the license plate region is localized in the first step. Hence, Localization is the bottleneck for accuracy of the entire system.

3.1. Localization

The aim is to identify the region in an image that may probably contain the license plate. Using this localized region, the area is segmented for further recognition. So far there are four approaches to localize license plate.[1].

3.1.1 Binary Images

In binary images, license plates detected through mathematical morphological processing and vertical edge detection yield very good results. However, this is very sensitive to complex images with shadows and variation in gradient across the image. Since morphological operations depend on object dimensions, the above method is also dependent on the distance between camera and car. Accuracy of 80-93% [10] has been achieved through this method. Another approach is label connected components in the binarized image using 4-connectivity or 8-connectivity. Finally, region of interest is extracted by filtering disjoint components on the basis of thresholds on aspect ratio, area, orientation, etc.[9].

3.1.2 Grayscale Images

License plates usually have black texts over white background. This change in contrast is used to detect probable LP region. The idea is to scan the rows of the image to look for such drastic change in contrast that is repeating over a scale of 15 pixels[3] assuming that the contrast between background and character is good and there are more than 4 characters on the LP. A similar approach scans N-row

distance and counts number of edges. If number exceeds certain threshold then a probable LP is detected.

3.1.3 Color Images

These methods rely on color of license plates and character. As a result, it does not provide enough reliability as color changes under different illumination and LP colors vary country wise. Certain intuitive rules such as LP is rectangle, LP dimensions were used as Fuzzy sets and membership functions were defined to locate the LP.[11]. Another robust method used textures to train SVM and adaptive mean shift to measure proximity to locate LP by scanning the image through window of fixed size. This method achieved accuracy of 91.3%[5].

3.2. Segmentation

After localizing the plate, the character of the plates need to segmented. Character recognition relies on how well the characters on the LP can be isolated. For this the image is first binarized. The idea is to add of rows or columns to obtain a vector whose minimum value then allows us to segments the image.[7] Connected Components are also extracted from the image and then based on threshold on properties such as length, width, etc, components corresponding to characters are extracted. The image can also be adaptively binarized to detect noise through thickening and pruning. Finally, the components corresponding to same character are merged. [8]

3.3. Recognition

Various character recognition methods are implemented such as statistical classifiers, computational intelligence and common-matching pattern techniques.[1]. Template matching is useful for characters that are single font, not rotated and of fixed size. If template images are of high quality then using normalized correlation, high accuracy can be achieved. Other than normalized correlation, Hausdorff distance may also be used to find correlation between template image and segmented character.

4. APPROACH

4.1. Localization

Our approach to this problem uses morphological operations along with vertical edge detection to detect license plates in images. [4] The first step in our approach is to pre process the image. We use Gaussian mask followed by sharpening filter to remove noise and making edges stronger. We also use Median filter to get rid of black pepper noise. The next step is to process the image and localize license plate in the image. For this step, we start with Vertical edge detection. Based on Vertical edges in each row of the image, we make an edge histogram. We analyze this histogram to find candidate rows with maximum edge variations. We mask the image to only leave out the candidate rows.

This edge image is then followed by dilation in vertical and horizontal direction. We then find the common regions from X-dilated image and Y-dilated image. The final image is then dilated in X direction and then eroded with a different structuring element. We find the largest binary region in the image, to locate the license plate.

Finally we find the coordinates of the plate to locate it in the original image.

4.2. Segmentation

We have used connected components to extract the characters of license plates based on the approach in [2]. First, the plate image is binarized using adaptive thresholding. This ensures that shadows do not affect our segmented image. Since, different plates may have different orientation: in order to make the system robust to rotation, skew correction was performed using Hough Transform. After orientation was perfected, we tried to remove extra noise in the binarized image such as LP boundary. This was followed by extraction of connected components. Next, bounding box corresponding to each connected component was extracted and based on various properties such as aspect ratio, ratio of area of box with total area of image, etc, boxes corresponding to characters were identified.

4.3. Recognition

We are using the inbuilt OCR function in MATLAB to recognize characters. Instead of running the OCR on the entire number plate, we are running OCR on every character segment obtained from the previous step.

5. METHODOLOGY

5.1. Image Pre-Processing

5.1.1 Noise Removal

First step is to apply Median Filter to image to remove black pepper noise. Next step is highly sensitive to noise, hence, median filtering is applied first. Next step is to apply Gaussian Low Pass filter to remove noise.

5.1.2 Sharpening Image

We use unsharp sharpening mask to sharpen the image to complete preprocessing.

5.2. Localization

5.2.1 Vertical Edge Detection

We use sobel filter (Sy) to detect the vertical edges in the image. For each pixel of the edge image, we raise it to power of 2. This makes sure that small magnitudes are ignored and strong edges become stronger.

Figure 1. Vertical Sobel Operator

5.2.2 Normalization

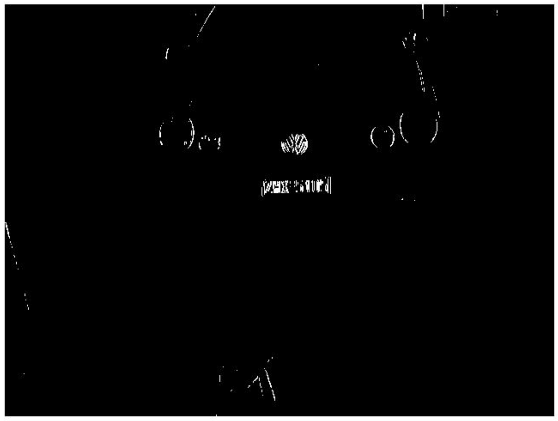
We normalize the pixel intensities of the image. The intensity values are proportionally scaled between minimum and maximum intensity of the the image.

Figure 2. Normalized Image

5.2.3 Thresholding

We use Otsu's Thresholding method to get a threshold level. This is followed by thresholding the image based on this level.

Figure 3. Ostu's Thresholding



5.2.4 Histogram Analysis in Vertical

For this edge image, we find the number of white pixels (pixel value = 1) in each row. We make a histogram for each row of the image. The rows having white pixels more than a threshold percentage: c percent are considered as candidiate rows.

To avoid having non-contiguous candidate rows, we add another condition: for a row to be a candidate row, number of white pixels in that row should be more than threshold c percent and number of white pixels in it's neighbouring rows should be more than d percent. This condition makes sure we get contiguous set of rows in our candidate image.

We are using the fact that rows that contain license plate will have maximum number of vertical edges.

Figure 4. Histogram for Vertical Edges

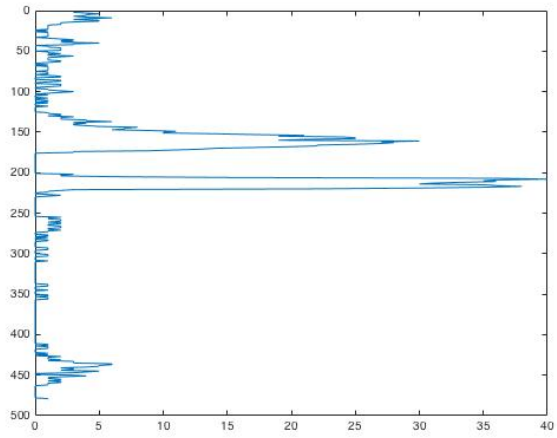
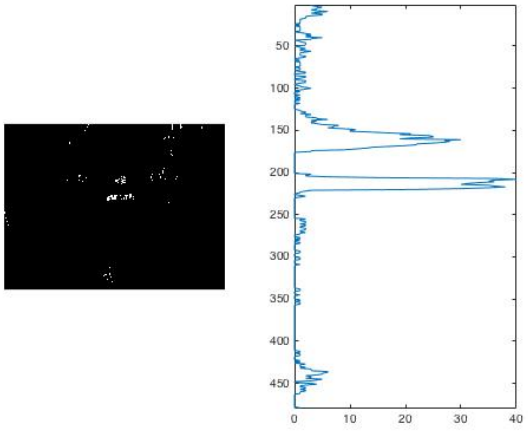


Figure 5. Vertical Histogram Analysis



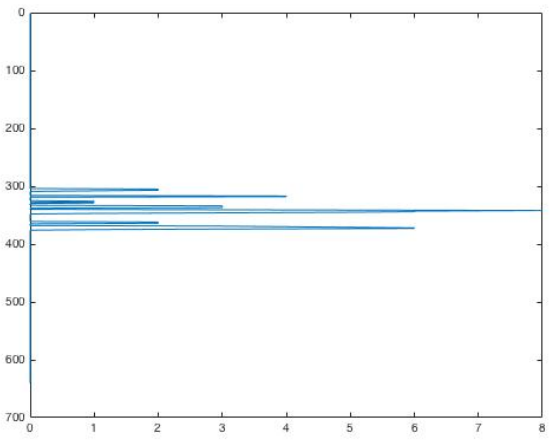
5.2.5 Histogram Analysis in Horizontal

In the previous step we might get a lot of extra rows that are not part of the number plate. We apply similar histogram analysis in horizontal direction as well. This is again using the fact that the number plate area will have a lot of horizontal edges.

We similarly find histogram for each column in the image and find the columns which have white pixels more than a certain threshold c percentage and it's neighbouring columns have white pixels more than a certain threshold d percentage.

This step helps us ignore all the irrelevant rows and columns and only stick to the part of the image that has high frequency changes in both Y and X direction.

Figure 6. Histogram for Horizonatal Edges

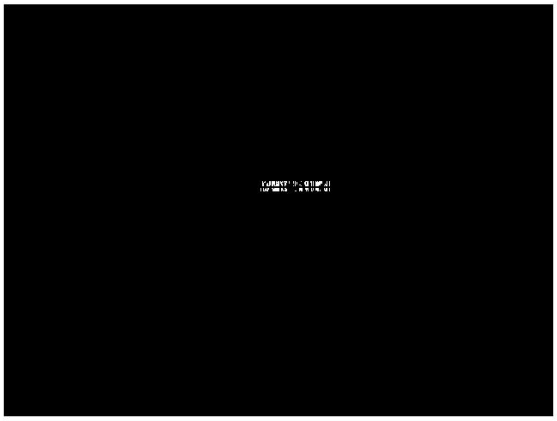


5.2.6 Masking Image

We create a mask based on the above histogram analysis in both directons. This mask will render all the non-candidate rows and columns in the edge image to 0. Hence, we left with an edge image with only candidate rows and columns. We can now crop the image.

Figure 7. Masked Image (Original Image)

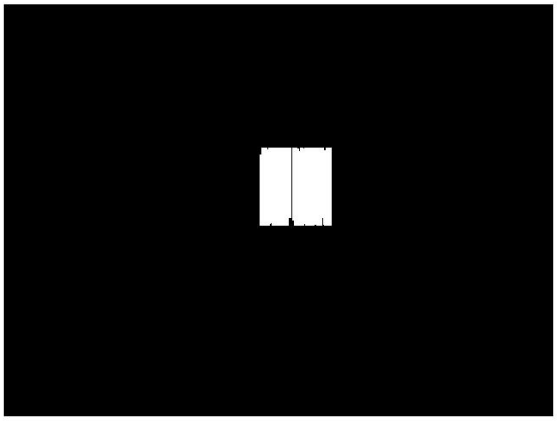
Figure 8. Masked Image (Edge Image)



5.2.7 Morphology (Dilation in Vertical)

We use a 80x4 rectangle structuring element to dilate the edge image. We use MATLAB's imfill() function to fill the holes so that don't have any holes in the image.

Figure 9. Dilation in Vertical

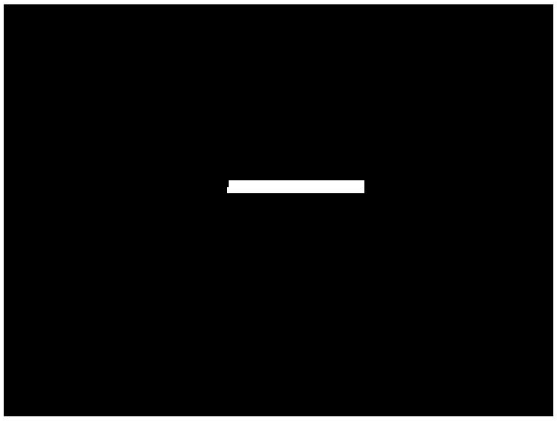


5.2.8 Morphology (Dilation in Horizontal)

We use a 4x30 rectangle strucuring element to dilate the edge image. We use MATLAB's imfill() function to fill the holes so that don't have any holes in the image.

80 pixels of height and 30 pixels of width is carefully chosen to detect the particular kinds of number plate present in our dataset (8:3 aspect ratio).

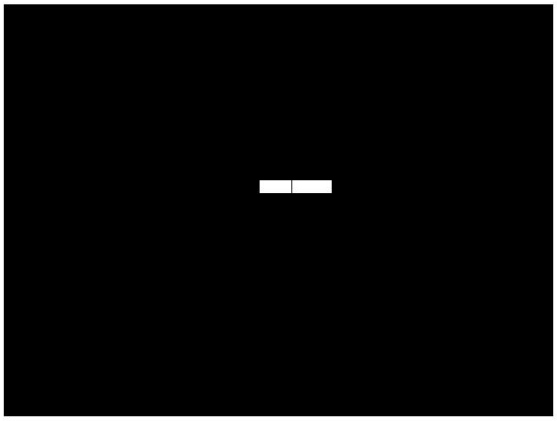
Figure 10. Dilation in Horizontal



5.2.9 Joint Image

We multiply the two binarized images to get the area final area from both of the above images. It is the intersection of above two images. This gives the holes filled area for the final license plate area.

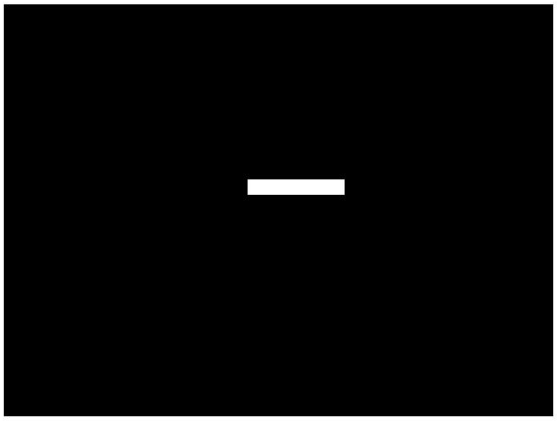
Figure 11. Joint Image (Intersection)



5.2.10 Morphology (Dilation in Horizontal)

We use a 30x4 rectangle structuring element to dilate the edge image. We use MATLAB's imfill() function to fill the holes so that don't have any holes in the image. This step is make sure the horizontal direction is of appropriate length.

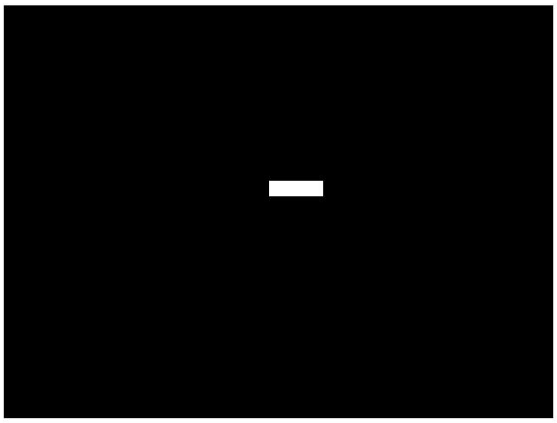
Figure 12. All holes filled



5.2.11 Morphology (Erosion)

We now use a line structuring element of length 50 pixels and 0 degrees orientation to erode the image.

Figure 13. Vertical Sobel Operator.



5.2.12 Biggest Binary Region

We now find areas of every region in the image. And we find the largest binary region in the image. After all the processing done above, the area of licence plate should be the largest area and it should be clearly detected.

5.2.13 Post Processing

Now we find the coordinates of the largest binary region. We take an extra of 10 pixels on each side to make sure the

complete license plate is detected. Using these coordinates we draw a red box on the image and also crop the license plate region.

Following this, we perform segmentation on this cropped image. Localization is the bottleneck of the accuracy of the complete system.

Figure 14. Detected Licence Plate

Figure 15. License Plate

6. Segmentation

We are using binary connected components to segment the characters and then running OCR on these connected components. The colored cropped image of license plate is first converted to grayscale image. Since some license plates are rotated, we used Hough Transform to detect lines and perform skew correction. In this, co-linear dots set is identified by drawing straight line. Result after skew correction is given in [Fig 17]

Next, we found the complement of the image and binarized it using Adaptive Threshold. Window size of the adaptive threshold is 50x50 and 0.1 is magnitude of C is 0.1. [Fig 18]

On this binary image, we performed morphological reconstruction using a line of thickness varying between 1-5 and 90 degrees in order to remove any unnecessary artifacts. After this, we extracted the connected components and their corresponding bounding box. In order to eliminate non character components, we enforced certain limits such as the height of the bounding box should be approximately equal to the height of the image, the width of each bounding box should not exceed width image divided by number of characters on LP (7 in our case). Result of bounding box of each connected component corresponding to characters is in [Fig 19].

Figure 16. Hough Lines

Figure 17.

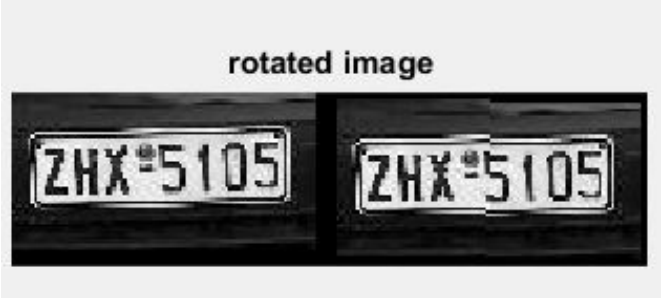


Figure 18. Binarized image

7. Recognition

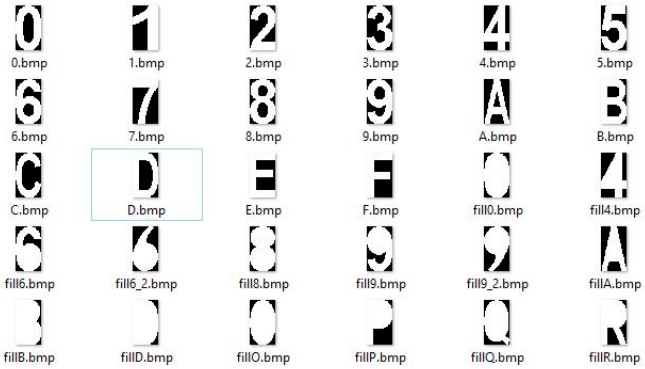
To find similarity, we first tried using Tesseract OCR, but it was not able to detect text in almost 40% images and for rest it produced gibberish.

Next we tried to perform template matching. However, since the data set we used does not provide template images, we built our own template images with the best possible matching font.[Fig 20]

We used 2-D correlation coefficient as a distance function for matching. Since our template images has resolution of 42x24 pixels, we resized every segmented character.

Figure 19. Bounding Box on components

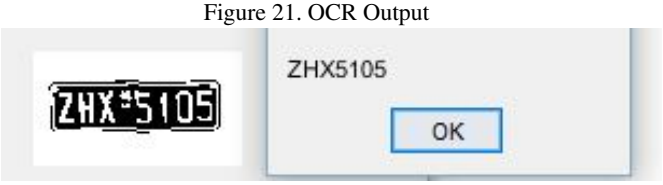
Figure 20. Bounding Box on components



But this lead to pixelation of the segmented characters because our data set had varied size of license plates. Moreover since the font we used in our template images was not same as the font of the LP, correlation was not working fine.

Next we tried OCR function of MATLAB. For the data set we used, MATLAB's inbuilt OCR performed better than both template matching and Tesseract.[Fig 20]

Figure 21. OCR Output



8. RESULTS AND ACCURACY

The size of our data set is 228 images. This data set involve all kinds of pictures: pictures shot in broad daylight, at night with flash, at night without flash, with strong shadows on number plates, also the ones with some projective transform.

8.0.1 Accuracy in Localization

Out of 228 images, we were able to successfully localize plates on 179 images. This represents an accuracy of 78.5%.

8.0.2 Accuracy in Recognition

We defined ground truth for 25 images based on random sampling of our data set. The distance measure between ground truth and OCR result is Edit Distance.

Number of characters on each plate is 7. Hence, similarity = 7 - EditDistance(image (i)). Accuracy is defined as 100 * similarity(i) /7.

Accuracy for our recognition OCR was around 85% on these 25 images.

9. OTHER METHODS TRIED

Details about some other approaches that we tried.

9.1. Contours for License Plate

Another way to detect rectangles in the image is to use Contours. After Pre-Processing the image as mentioned earlier, we used the canny edge output. We adjusted the threshold to remove all the clutter. In an attempt to detect the rectangular license plate, we used contours. We got all the contours including the characters on the number plate and the number plate itself. However, we were successful in rejecting contours based on geometry. It isnt very effective and lead to poor accuracy.

9.2. MSER Features

We tried to detect text regions in an image using MSER features. The aim was to detect all probable regions which might contain text and then use threshold to narrow down to the region containing text. We tried out the inbuilt MSER feature detector in MATLAB. But we found that command detectMSERFeatures in MATLAB does not return the connected components which is required to detect text regions based on various properties such as aspect ratio, eccentricity, etc. We also tried VLFeat MSER region detector to extract the connected components data structure. However, we chose not to explore this further since license plate localization through MSER is slow. Moreover, we had to manually change threshold for every image and we weren't successful in making it as robust as our current approach.

10. Color Features with edges

We used Sobel Operator in vertical direction and found vertical edges in the image. Then we added Gaussian blur to the image. Because of high contrast between characters and background, we applied the area filtering by vertical edge density for all 3 RGB channels, and filter out single colored edges, such as that from rear light or environmental

background. Using region labeling, we can label different regions and reject the areas based on the aspect ratio of a standard license plate.[6] However, this was not very successful and proved to be awfully inaccurate on our data set.

11. DIVISION OF LABOUR

Since, we were working in a team of two we made sure each of us gets equal contribution in the project. Both of us spent equal hours on this project.

12. CODE

We have uploaded the code along with the submission on Google Classroom.

References

[1] Christos-Nikolaos E Anagnostopoulos, Ioannis E Anagnostopoulos, Ioannis D Psoroulas, Vassili Loumos, and Eleftherios Kayafas. License plate recognition from still images and video sequences: A survey. *Intelligent Transportation Systems, IEEE Transactions on*, 9(3):377–391, 2008.

[2] Leandro Araújo, Sirlene Pio, and David Menotti. Segmenting and recognizing license plate characters. *International Journal of Computer Vision*, 24(3):251–270, 1997.

[3] Sorin Draghici. A neural network based artificial vision system for licence plate recognition. *International Journal of Neural Systems*, 8(01):113–126, 1997.

[4] Farhad Faradji, Amir Hossein Rezaie, and Majid Ziaratban. A morphological-based license plate location. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, volume 1, pages I–57. IEEE, 2007.

[5] Kwang In Kim, Keechul Jung, and Jin Hyung Kim. Color texture-based object detection: an application to license plate localization. In *Pattern Recognition with Support Vector Machines*, pages 293–309. Springer, 2002.

[6] Qing Li. Video-based license plate reader.

[7] David Llorens, Andrés Marzal, Vicente Palazón, and Juan M Vilar. Car license plates extraction and recognition based on connected components analysis and hmm decoding. In *Pattern Recognition and Image Analysis*, pages 571–578. Springer, 2005.

[8] Shigueo Nomura, Keiji Yamanaka, Osamu Katai, Hiroshi Kawakami, and Takayuki Shiose. A novel adaptive morphological approach for degraded character image segmentation. *Pattern Recognition*, 38(11):1961–1975, 2005.

[9] Cemil Oz and Fikret Ercal. A practical license plate recognition system for real-time environments. In *Computational Intelligence and Bioinspired Systems*, pages 881–888. Springer, 2005.

[10] Cheokman Wu, Lei Chan On, Chan Hon Weng, Tong Sio Kuan, and Kengchung Ng. A macao license plate recognition system. In *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, volume 7, pages 4506–4510. IEEE, 2005.

[11] Nikolaj Zimic, Jelena Ficzko, Miha Mraz, and Jernej Virant. The fuzzy logic approach to the car number plate locating problem. In *Intelligent Information Systems, 1997. IIS'97. Proceedings*, pages 227–230. IEEE, 1997.