License Plate Localization for Difficult Cases

Juan Miguel J. Bawagan, Patrick Dominic R. Menorca, Vladimir Y. Mariano
Institute of Computer Science
University of the Philippines Los Baños
College, Laguna, Philippines 4031
{jmbawagan,pdrmenorca}@gmail.com, vymariano@uplb.edu.ph

ABSTRACT
We explore an edge-based algorithm for extracting license plate images from a video of a traffic scene for use in vehicle recognition systems. We specifically focus on difficult-case Philippine license plates, which appear with varied styles and impeding factors such as dirt, scene clutter, low contrast, and partial occlusion. Given a video frame of the scene, we extract the vertical edge information using approximate derivative functions. From this gradient image, we make multiple hypotheses of the plate's location based on the areas of maximum edge density. Then, we adjust the dimensions of these bounding boxes to exclude portions that are unlikely to be part of the license plate. Finally, we eliminate those hypotheses whose dimensions do not fall within the valid range of plate sizes.

Keywords
computer vision, automatic number plate recognition, license plate localization

1. INTRODUCTION
License plate localization is an object detection method that extracts the license plate image from an image of a street scene. It is the first part of *license plate recognition* (LPR), the process that identifies a vehicle by reading its registration number. The LPR process may be divided into four stages: plate localization, enhancement, character segmentation, and recognition. Plate localization provides the input to all the other stages and it is regarded as the most critical stage of the process. Thus, a certain level of accuracy and efficiency is required of this particular stage.

1.1 Background of the Study
The framework used by most object recognition systems is *pattern recognition* - the act of taking in raw data and taking an action based on the category of the pattern [3]. LPR systems get raw data (i.e. video frame), derive patterns (e.g. salient license plate features), and make decisions (e.g. alert police station or store in database). Specifically, it accepts image frames from a camera and processes each frame to isolate the image of the license plate. The license plate is then read by dividing the image into blocks that are likely to represent the individual letters. Each block is then processed so that each may be classified into the country's set of letters and numbers, thus identifying the plate number.

Applications of LPR systems include surveillance and security systems. These systems are used by the police to monitor highway activity, including red-light violations, highway overspeeding, and car theft. LPR is most commonly employed in parking and access control systems. These are used for electronic toll and car park fee collection. They can also be used to prevent unauthorized entry of unknown or unregistered vehicles into restricted buildings.

Figure 1: Examples of difficult cases. (Left to right, and top to bottom) dirty plates; scene clutter: jeepney grills and misleading text; partial occlusion; low contrast: too bright, too dark, plastic cover; and uneven illumination

The complexity of the input images demands effective algorithms. Despite the fact that LPR systems are developed for a specific context (application- or country-specific), the population of all input images within that context may still have significant variations. The Philippine license plates alone have at least six various designs: dark green characters against white background, black characters against yellow background, red characters against white background, and their inverses. Natural and usage factors also contribute to the complexity: faded characters, dirt, uneven illumination, text on vehicles, occlusions, and angular distortions. An effective algorithm must be robust against most if not all of these factors.

1.2 Previous Works on License Plate Localization

There are many approaches to license plate localization. However, much of the available literature is *feature-based* - they rely on salient features of the object in order to detect or track them. Unlike *model-based methods*, which rely on detailed representations of an object, feature-based methods discard the idea of tracking objects as a whole; instead they track sub-features like noticeable points or lines on an object. The advantage of this approach is that features of a moving object remain visible even in the presence of partial occlusion [2]. In general, license plates are distinguishable from other objects by their defined aspect ratios, high local contrast, and occassionally, an enclosing rectangular frame. The observable width-to-height ratio of Philippine license plates is in the range of 3-4. Local contrast is the amount of difference between the neighboring pixels in an area, whether in color or brightness.

However, many other objects exhibit similar features. Jeepney grills, car stickers, reflective surfaces, shadow and mud streaks all exhibit or add unwanted high edge values to the wrong places, misleading most edge-based algorithms. Other factors on the other hand, diminish the edge density of the license plate areas, such as overexposure to light and faded characters.

In 2003, Shapiro and his colleagues [8] proposed an adaptive method for license plate localization. Their method is characterized by the use of a median filter on the edge image to produce an ellipsoidal smear, which is the license plate. The location and dimensions of the plate are then based on the row exhibiting the highest sum of pixel edge values. Their algorithm also adjusted the left and right boundaries using vertical and horizontal projections, some empirically calculated constants and application constraints (e.g. aspect ratio and plate size). Their method achieved 90% accuracy on their database.

Other features may be used with edge information in order to locate license plates. Mahini et al [6] presented another feature-based method, characterized by its use of three "basis images". These images are derivations of the same input image and describe different features: vertical edge value, pixel grayness, and brightness. The basis images are used to decide which pixels will be considered possible license plate characters. Each group of connected pixels is then tested for license plate similarity and one among these groups will be decided as the license plate. Their method took an average of 300 milliseconds to process 800x600 color images at 96.5% accuracy.

Jia et al [5], in 2005, presented a *region-based* method for candidate extraction. They used the mean shift algorithm, a statistical method that searches for the mode of a point sample distribution. By applying this method on the joint spatial-and-range domain, they were able to cleanly segment the image. Their method effectively divided the image into their object regions, forming logical groupings such that pixels within each group/region exhibit the similar intensities and are contiguous. Their method used region rectangularity, aspect ratio, and edge densities in their Mahalanobis classifier to distinguish the plate plate region from the rest of the groups. To measure their algorithm's performance, they used the *difference ratio* (1).

$$d = \frac{|A_{real} - A_{detected}|}{|A_{real}|} \tag{1}$$

On the average, each of their algorithm's output box missed out only 4.9% of the real license plate (95.1% accuracy).

Enyedi et al [4] proposed *fast localization algorithms* which they categorized into three groups. The first set of algorithms, intended for coarse localization, made use of edge values and a subsequent window filtering. The second set of algorithms, intended to refine the initial guess, applied adaptive threshold (ATH) procedures to refine the localized area. The third set was intended to realign tilted license plates and employed direction sensitive window filtering.

The preceding review presents only a few of the existing techniques in license plate localization. This study aims to develop a license plate localization algorithm that is both fast and reliable (i.e. it will successfully locate plates under difficult cases). The efficiency of the resulting algorithm may present a viable alternative to these existing solutions.

1.3 Our Proposed Solution

Some of the algorithms in the literature combine several complex procedures to achieve high accuracy, but to the expense of speed. These algorithms may be improved for speed by using simpler (and therefore faster) techniques. Our approach to plate localization is *edge-based*, i.e. we capitalize on the strength of contrast between the license plate characters and the plate surface. Figure 2 illustrates the process in detail.

We propose that candidates be extracted per frame by applying approximated first or second derivative filters on the grayscale image. Each possible window area will be scanned and those areas which exhibit the highest sum of edge/gradient values shall then be considered as possible plate candidates.

The use of multiple candidate extraction also helps in finding license plates in difficult cases. This allows the system to make secondary guesses of the plate location. By making multiple hypotheses, the plate localization step ensures that the license plate is included as much as possible.

To minimize the false positives and the amount of data passed on to the rest of the LPR process, we suggest a subsequent refinement and elimination procedure after the coarse localization step. The refinement step includes the adjustment of each candidate's location and boundaries. Our hypothesis is that by means of extracting only the vertical

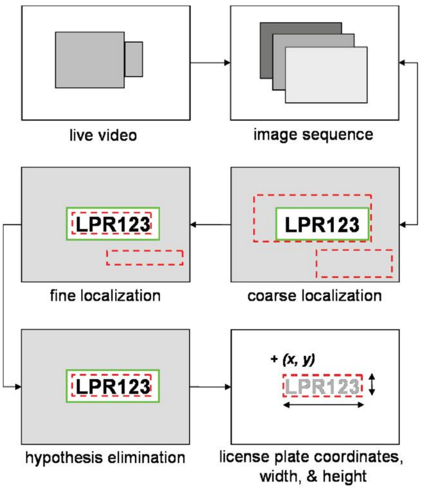

Figure 2: License plate localization: The main localization program receives and processes every frame of the live video. The program then makes rough hypotheses of the location of the plate, refines these hypotheses, and decides on the accuracy of each one. The output of the program is a set of zero, one, or more bounding boxes.

gradient or edge information using simple filters and employing application-dependent constants during refinement should allow for a faster yet reliable localization algorithm.

The rest of the paper is organized as follows: Section 2 describes the algorithm in more detail, where each of coarse localization and fine localization is allotted a separate subsection. The results, discussion, and recommendations for future work are presented in Sections 3 and 4.

2. METHODOLOGY

2.1 Preprocessing

The video camera feeds the system with UYVY frames, which are then converted to YUV. These frames are D1-sized (704x480 pixels) images. To reduce the data to be processed, we scale these frames to CIF-sized (352x240 pixels) images. Only the grayscale image $G(x, y)$ is passed on to the next stage.

2.2 License Plate Localization

Localization may be divided into two substages: *coarse localization*, where the goal is to cover the entire plate with a bounding box; and *fine localization*, where the goal is to exclude the non-plate areas from the coarse bounding box.

2.2.1 Coarse Localization

Coarse localization proceeds by the following steps:

1. **Define the bounding box's dimensions.** For this study, the dimensions are predetermined and must be adjusted according to the intended relative size of the area occupied by the license plate in the image. For our experiment, we defined the bounding box to be 60 pixels high and 150 pixels wide (using the CIF-sized images).

2. **Extract the edges of the image.** License plate characters generally exhibit higher intensity variations along the horizontal direction, as compared to its surroundings (e.g. surface of the car). Thus, we extract only the vertical edges of the image and refer to the edge image as the *gradient image*, $E(x, y)$. The pixels along each row of the grayscale image are used as input to either the first derivative approximating function,

$$E(x, y) = |G(x, y) - G(x + k, y)| \tag{2}$$

or the second derivative approximating function

$$E(x, y) = |G(x - k, y) - 2G(x, y) + G(x + k, y)|. \tag{3}$$

Each $E(x, y)$ (gradient value) is then compared against some threshold. If $E(x, y) < edgeThreshold$, it is set to zero. Otherwise, the original gradient value $(E(x, y))$ is kept. Applying the algorithm on the database of [1] using different values of k and edgeThreshold ($k = 1 \ldots 20$; $edgeThreshold = 10, 20, \ldots 250$), we found that $k = 6$ and $edgeThreshold = 40$ yields the highest average recall. Figure 4 show the output of these two filters.

Figure 3: Example of a difficult case (original image and its grayscale). The grills of the jeepney result in high edge content, misleading most edge-based algorithms.

Figure 4: Applying the approximate first-order and second-order derivative filters ($k = 6$) on the grayscale image will yield these two edge images. These filters highlight the high contrast vertical edges found in license plates.

3. **Locate the areas of highest edge concentration.** One by one, we scan every possible rectangular area

of gradient image and identify which area exhibits the highest edge density. The bounds of this scanning window (rectangular area) are defined by the dimensions of the bounding box in Step 1.

To find the location of the area with the maximum sum of edges, we use the *running-sum* algorithm, which minimizes the amount of addition operations. The idea is to map a coordinate-pair (x,y) to a particular rectangular area of the gradient image.

Figure 5 illustrates the running sum algorithm. Given a $M \times N$ gradient image and a $W \times H$ window, the running-sum algorithm produces the said mapping as follows:

(a) Define a two-dimensional array V (with M rows and (N-H+1) columns). Each value in V is:

$$V[x, y] = \sum_{i=y}^{y+H-1} E(x, i)$$

Each cell value in V is simply the sum of values in a particular $1 \times H$ strip of the gradient image.

(b) Define another two-dimensional array A (with dimensions (M-W+1) rows and (N-H+1) columns). Each value in A is:

$$A[x, y] = \sum_{i=x}^{x+W-1} V[i, y]$$

Each cell value in A is simply the sum of values in a particular $W \times 1$ strip of V.

The array A should store by now the actual sum of edges in every possible window area of the gradient image.

Once the values of A have been computed, finding the area of highest edge concentration becomes a matter of finding the maximum value in A. The x and y indices of the maximum value in A is then the resulting location coordinates of the area of highest edge concentration.

To get multiple plate candidates, once a maximum area is located, the gradient values within its area are set to zero. Then the running sum algorithm is repeated to locate the area with the *next highest* edge concentration. By setting the area to zero, we ensure that the next plate candidate does not overlap with any of the previous plate candidates in the frame.

For our experiments, we defined the number of plate candidates to be equal to 3 ($numOfGuesses = 3$). Figure 6 shows the result of the coarse localization. Each of the plate candidates are then subjected to refinement.

2.2.2 Fine Localization

Fine localization proceeds by the following steps:

1. **Adjust the bounding box towards the center of mass.** Ideally, the bounding box perfectly encloses

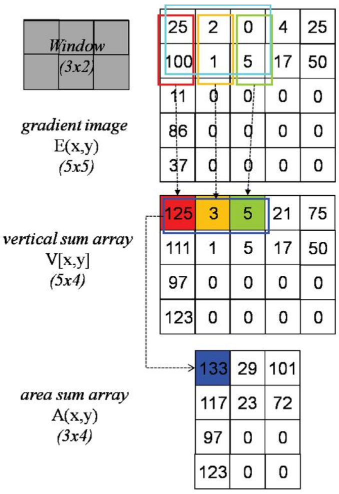

Figure 5: An illustration of the running sum algorithm. Given a license plate size of 3x2 pixels, and a gradient image size of 5x5, there are 12 possible areas. Note that the sum of edge values in the window area (aqua box) is equal to 133. Using an exhaustive summation algorithm, it would take $5 \times 12 = 60$ additions to compute this value. The running-sum algorithm obtains this same value in $(1 \times 5 \times 4) + (2 \times 3 \times 4) = 44$ additions, which is 1.36 times faster.

the *LP character row* (LPCR) [Fig. 7]. Most of the time, however, pixels of high edge values exist near the LPCR and these end up being part of the coarse bounding box. The result is that the output boxes are “off-the-center” or, worse, “broken” - the LPCR is not fully included in the bounding box.

To resolve this problem, the bounding box must be relocated such that its geometric center is also the center of the LPCR. Thus, the LPCR’s center must be approximated and this reference point will then be used as the bounding box’s new center. We propose that this reference point be the *center of mass of the edge values* inside the bounding box. The center of mass is the point where the weights (i.e. pixel edge values) of a given area appear to be concentrated or balanced. In computing for the center of mass, only the edge values within each bounding box will be used as weights.

$$center_X = \frac{\sum_{x=0}^{W-1} \sum_{y=0}^{H-1} xE(x, y)}{\sum_{x=0}^{W-1} \sum_{y=0}^{H-1} E(x, y)} \tag{4}$$

$$center_Y = \frac{\sum_{x=0}^{W-1} \sum_{y=0}^{H-1} yE(x, y)}{\sum_{x=0}^{W-1} \sum_{y=0}^{H-1} E(x, y)} \tag{5}$$

2. **Resize the bounding box.** Character segmentation step of the LPR process will work better if the output of localization bounding box neatly wraps around the license plate character row. This desired increase in

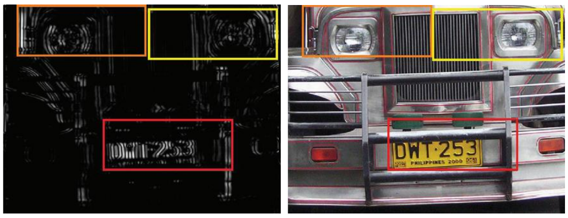

Figure 6: The areas of highest edge concentration using the 2nd derivative filter (left) and the corresponding plate hypotheses (right).

Figure 7: The LP character row (LPCR).

precision may be obtained by resizing the LP bounding box. The goal is to make the LP box more compact, which should remove the unnecessary parts in a LP image (retaining only the LPCR). To do this, the boundaries (upper, lower, left, and right) of the LPCR must be found.

Finding the upper and lower boundaries is done in a number of steps:

(a) For every row in the bounding box, count the number of *strong edges*. If an edge value $E(x, y) >= binary\_threshold_{row}$, pixel has a high edge strength. In this study, $binary\_threshold_{row} = 50$.

(b) Determine whether a row is an LPCR row or not. Since a license plate usually has six characters, there must be a minimum number of strong edges along any of the LPCR. In this study, if a row has at least 15 edges ($row\_threshold = 15$), it is marked as "LPCR row"; otherwise it is considered a "non-LPCR row".

(c) The *thickest band* of "LPCR" rows is determined. The top and bottom rows of this band shall form the upper and lower boundaries of the actual license plate character row.

The algorithm is almost the same in finding the left and right boundaries:

(a) For every column in the bounding box, count the number of *strong edges*. If an edge value $E(x, y) >= binary\_threshold_{col}$, pixel has a high edge strength. In this study, $binary\_threshold_{col} = 60$.

(b) Define the *column_threshold*. Since most characters have shorter horizontal lines, the LPCR has less edges along the vertical axis than along the horizontal axis, i.e. the columns have less edges than the rows. For this study, $column\_threshold = 2$.

(c) Search the left and right boundaries. Starting from the leftmost and rightmost columns, find the

Figure 8: A LP image with the coarse output box (yellow) and the adjusted box (black).

Figure 9: Three hypotheses from coarse localization (red=1st, yellow=2nd, green=3rd) and each of their adjusted boxes (nearest white box).

first columns that have a greater number of strong edges than *column_threhold*.

Figure 10: Clockwise from top left: (1) A license plate image, (2) its 1st derivative edges, (3) its 2nd derivative edges, and (4) the new plate hypothesis after resizing the bounding box.

3. **Initial elimination of guesses.** Derived bounding boxes that do not fit the characteristics of a LP should be eliminated. This should make the subsequent LPR processes more efficient because the number of false data is reduced (boxes that do not contain a license plate). In this research, the minimum width and height were considered as the parameters in removing "non-LPs". If one of the following conditions is true, then the bounding box is eliminated:

- if the width of the resized LP box is less than the set minimum width
- if the height of the resized LP box is less than the set minimum height
- if the width of the resized LP box is less than its height

In the processed images, the minimum width was set to 90 pixels while the minimum height was set to 30 pixels. For the real-time implementation (CIF size), the minimums were 37 (width) and 15 (height) pixels.

Figure 11: An image containing 3 hypotheses with the coarse (thin boxes) and the resized outputs(thicker boxes). After guess elimination, the only remaining box was the first guess (red).

3. EXPERIMENTAL RESULTS

The algorithm was implemented using C++ and was run in Fedora (Linux OS). During development, the database of images from the study by Castillo and Mariano [1] was used. Each of the 407 JPEG images contains one license plate.

Mariano et al [7], in their paper entitled *Performance Evaluation of Object Detection Algorithms*, described several performance measures, including recall and precision. *Recall* describes how much of the ground truth area is covered by the algorithm output; *precision* measures the "exactness" of the algorithm output in covering the ground truth area. To measure the algorithm's reliability, recall was used as the main measure.

$$Recall = \frac{|\ ground\_truth\ \cap\ algorithm\_output\ |}{|\ ground\_truth\ |} \quad (6)$$

$$Precision = \frac{|\ ground\_truth\ \cap\ algorithm\_output\ |}{|\ algorithm\_output\ |} \quad (7)$$

Since each frame may have more than one algorithm output (bounding box), only the highest among the multiple recall values will determine the actual recall for that frame. A frame with no algorithm output has a recall value of 0. Using only the coarse localization, the performance test on the database yielded an average frame recall of 93.45%.

Results show that adjusting the box towards the center of mass had little effect. Majority of the test images have shown small shifts towards the center of the character row (LPCR), but the movement was not significant [Fig. 8]. The reason for this is that the center of the bounding box was already close to the center-of-mass to begin with. While some showed good results [Fig. 8], some actually moved away from the LPCR's center.

In resizing the LP bounding box, most of the results have shown good box resizing. [Note: This is a qualitative judgment. No quantitative performance evaluation has been done.] Although some have not fully wrapped around the

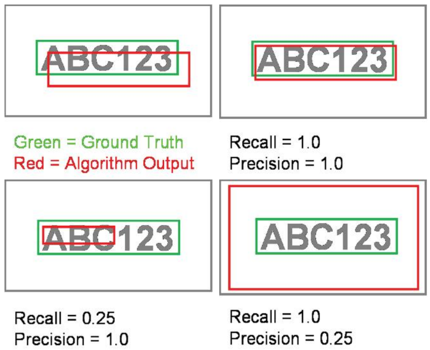

Figure 12: Some examples of recall and precision values. It is important that algorithms have good recall (low misdetection) and good precision (low false alarm).

LPCR [Fig. 13], there is a big increase in precision. Furthermore, many of false plate candidates were eliminated [Fig. 11, 13].

Figure 13: An example of a good resizing. Note that the red bounding box tightly encloses the character row of the license plate.

Algorithm performance on some images resulted to the elimination of true positive plate candidates. [Fig. 14]. This is caused either by low ambient light or low contrast on the original image.

Figure 14: A case where a true positive was eliminated. Notice that the edges in the edge image are weak due to low ambient lighting.

Some boxes have "over-wrapped" the LPCR and have cut some characters from the LP [Fig. 16]. Additionally, there were still a number of true negatives that were not eliminated [Fig. 15]. This is caused by the high amounts of contrast on those areas. Figure 16 shows a common result for a difficult case.

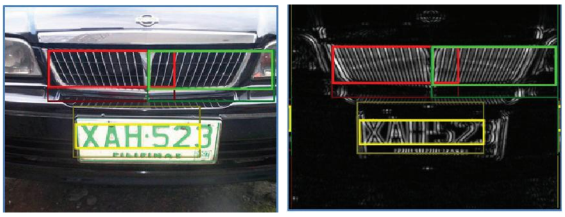

Figure 15: Example of bounding boxes that should have been eliminated. The grills on this vehicle produce enough edges to be mistaken as real license plates.

Figure 16: These images show the case of over-wrapping. In the left image, the digit '1' is not covered by the bounding box while on the right, the letter 'C' and some portions are also not covered.

Also, in many cases where the scene clutter exhibits high contrasts, the algorithm is still able to capture the LPCR through the second plate candidate. Figure 17 shows a successful case of complete license plate localization process.

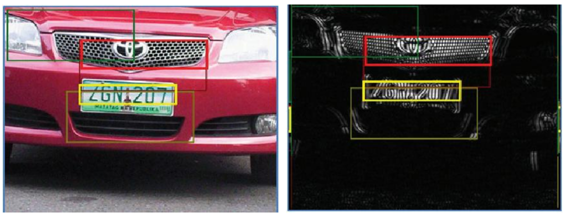

Figure 17: The LPCR in this image has been located successfully by adjusting towards the center of mass and resizing. This is also a good example where the use of multiple hypotheses enables the system to capture the actual license plate despite scene clutter.

4. CONCLUSION AND FUTURE WORK

Using multiple candidate extraction (guesses), the system increased its chances of localizing a license plate. Performance evaluation tests have indeed shown higher recall for coarse localization (93.45% average recall). Although there is no quantitative evaluation yet for the results in fine localization, qualitative assessments seem to show that overall recall is lower but has increased precision.

Future studies may improve the recall and precision values by statistically acquiring the best threshold values. Also, using video cameras that enhance the quality of images and good camera positioning may help in the overall performance of the system. Other studies may improve localization by deskewing for misaligned or highly tilted plates. And finally, false guesses that were not previously eliminated should be removed during the character recognition phase.

5. REFERENCES

[1] Castillo, M., and Mariano, V. License plate localization based on edges. Undergraduate special problem, University of the Philippines Los Baños, Laguna, Philippines, Mar. 2008.

[2] Coifman, B., McLauchlan, P., and Malik, J. A real-time computer vision system for vehicle tracking and traffic surveillance. Submitted for publication in Transportation Research-C, Mar. 2008.

[3] Duda, R., Hart, T., and Stork, D. *Pattern Classification, 2nd Edition*. Wiley, New York, USA, 2001.

[4] Enyedi, B., Konyha, L., and Fazekas, K. Real time number plate localization algorithms. *Journal of Electrical Engineering 57*, 2 (2006), 69–77.

[5] Jia, W., Zhang, H., He, X., and Piccardi, M. Mean shift for accurate license plate localization. In *Proceedings. IEEE Conference on Intelligent Transportation Systems, 2005* (Sept. 2005), pp. 566–571.

[6] Mahini, H., Kasaei, S., Dorri, F., and Dorri, F. An efficient features-based license plate localization method. In *ICPR 2006. 18th International Conference on Pattern Recognition, 2006* (Aug. 2006), pp. 841–844.

[7] Mariano, V., Min, J., Park, J., Kasturi, R., Mihalcik, D., Li, H., Doermann, D., and Drayer, T. Performance evaluation of object detection algorithms. In *Proceedings. 16th International Conference on Pattern Recognition* (2002), vol. 3, pp. 965–969.

[8] Shapiro, V., Dimov, D., Bonchev, S., Velichkov, V., and Gluhchev, G. Adaptive license plate image extraction. In *Proceedings. 5th International Conference on Computer Systems and Technologies. Rousse, Bulgaria* (2004), pp. 1–7.

APPENDIX

The images shown here represent some of the difficult cases and their corresponding localization outcomes. The first nine images are successful localizations while the last four images show failed localizations.

DWD 660
DMJ·358
PILIPINAS
DRR 365
TLM 638
AUTO MATE