IET Intelligent Transport Systems

Research Article

Improved license plate localisation algorithm based on morphological operations

ISSN 1751-956X
Received on 18th July 2017
Revised 29th January 2018
Accepted on 7th February 2018
E-First on 6th March 2018
doi: 10.1049/iet-its.2017.0224
www.ietdl.org

Juan Yepez[1], Seok-Bum Ko[1] ✉

[1]*Department of Electrical and Computer Engineering, University of Saskatchewan, Saskatoon, Canada*

✉ *E-mail: seokbum.ko@usask.ca*

Abstract: Automatic license plate recognition (ALPR) systems have become an important tool to track stolen cars, access control, and monitor traffic. ALPR system consists of locating the license plate in an image, followed by character detection and recognition. Since the license plate can exist anywhere within an image, localisation is the most important part of ALPR and requires greater processing time. Most ALPR systems are computationally intensive and require a high-performance computer. The present algorithm differs significantly from those utilised in previous ALPR technologies by offering a fast algorithm, composed of structural elements which more precisely conducts morphological operations within an image, and can be implemented in portable devices with low computation capabilities. The present algorithm is able to accurately detect and differentiate license plates in complex images. This method was first tested through MATLAB with an on-line public database of Greek number plates and was 100% accurate in all clear images, and achieved 98.45% accuracy when using the entire database which included complex backgrounds and license plates obscured by shadow and dirt. Second, the efficiency of the algorithm was tested in devices with low computational processing power, by translating the code to Python, and was 300% faster than previous work.

1 Introduction

Automatic license plate recognition (ALPR) is a computer technology where an optical device captures an image and this is then processed to obtain a string from a license plate. In general, the ALPR systems require a high-performance workstation to work [1–3]. However, when the same algorithm is used on a low computational power device, the processing time is extremely high relative to a workstation and it takes many seconds to achieve a result. ALPR systems, when used with real-time video feeds, are inefficient and slow [4], therefore more efficient methods and techniques need to be implemented in order to work quickly and with devices having low processing power.

The processing method in ALPR systems consists of three stages: license plate localisation (LPL), character segmentation, and optical character recognition (OCR) [5, 6]. LPL scans all of the pixels within an image to detect and localise the position of a license plate. Character segmentation is the stage where detection and separation of each character on a license plate occurs. OCR receives the character's information, validates, and encodes it to an ASCII symbol as an alphabetic letter or number. The stage which is crucial and takes the most processing time is that of LPL and is the main focus of this paper.

This paper presents an improved LPL algorithm based on morphological operations with a high and accurate detection rate suitable for low-cost devices such as Raspberry Pi, Odroid or any other embedded platform. The algorithm is faster compared to existing methods because it does not use the traditional edge detection methods, which are based on matrix multiplication, but instead a morphological Top-hat operation based on structural comparisons within a greyscale image. The algorithm highlights the license plate region, compares it to the background, thereby simplifying the detection process. The morphological operation utilises a structural element (SE) to detect and compare shapes within an image [5, 7]. The main difference between the present and previous LPL methods is that previous systems use a SE with randomly chosen size and dimensions that relied on no calculations or optimisation of shapes with an image [5, 8]. This lack of optimisation results in low accuracy and these deficiencies are overcome by our developed LPL SE. We developed an algorithm with SEs that detect appropriate structural profiles within an image and result increased accuracy despite environmental challenges such as illumination and plate orientation.

The present algorithm was designed in MATLAB so that it could be compared to previous works for accuracy and speed. To demonstrate that the system can run in a low-cost device without issues, the system was implemented in a Raspberry Pi so that detection speed can be measured and compared to other LPL's in frames per second (FPS) using a video camera. For training and testing purposes, a MediaLab-NTUA database was used [9]. The database is comprised of 573 images of individual cars with varying image resolutions of 640×480, 800×600, and 1792×1312 pixels. The images were grouped into eight sets using varying criteria such as distance, illumination, shadows, and dirty license plates.

The rest of this paper is organised as follows. Related work is reviewed in Section 2, whereas Section 3 describes the mathematical morphology (MM). The proposed method is then described in Section 4, whereas Section 5 is concerned with implementation and results. Section 6 then concludes the paper.

2 Related work

Different LPL methods have been presented in the scientific literature and are categorised based on the features, such as edge detection, global image information, texture, colour, and character features they use.

Edge detection algorithms find a rectangular shaped region with a known aspect ratio and extract all rectangles from within the image. This technique examines the changes in pixel amplitudes to transform the greyscale image into an edge image. The Sobel operator algorithm is a filter used for edge detection to define the boundary between two regions in a two-dimensional image, with its kernel scanning in both vertical and horizontal axes [10, 11]. The Canny operator uses a Gaussian filter for smoothing or convolution and can work with images of varying environmental conditions, distances, and angles [12]. Unfortunately, since Sobel and Canny are based on matrix multiplication, they are reliant on powerful processing power, which is costly, and suffer performance lag when used on low-cost devices. Morphological detection systems, such as that of Zhai *et al.* [5], eliminate non-plate regions thereby selecting the license plate and are faster than other edge methods. However, there are several drawbacks in Zhai's system in that it did not use complex images which challenge the algorithm, and second, was not tested in a low-cost device. All previous edge detection systems are highly accurate and efficient but are compromised when presented with complex images and were not designed to be used in low-cost devices.

Global image systems use connected component analysis (CCA) and are often used for license plate detection in low-resolution videos, including live feeds [3, 12, 13]. CCA algorithm scans a binary image and divides it into different components based on pixel connectivity. A contour detection algorithm is applied to identify connected objects based on their geometrical and spatial features such as area and aspect ratio, and are used for license plate extraction. Global image information methods work reliably regardless of the position of the license plate, but may generate license plates regions that are incomplete so that only a portion of a number plate is detected [1].

Another ALPR technology utilises textual features, such as colour transition between characters and the license plate background, for detecting license plate numbers. Gabor filter, which can differentiate textures in unlimited orientations and scales, is one of the major tools for texture analysis [14, 15]. However, a major drawback of Gabor filters is that they are time consuming. Another popular method is wavelet transform and is based on small wavelets with limited duration [2, 6, 16]. In this method, vertical features are extracted using wavelet transform and the position parameters of the plate are determined by analysing the projection features in both the time and frequency domains. Another texture detecting technology uses Haar-like methods for object detection [2, 15, 17]. Haar-like methods classify size, colour, brightness, and location of license plates thereby aiding in the detection of characters on a plate. A more complex textural method is that of the sliding concentric windows, and can detect the boundary even if the license plate in the image is deformed [3]. The main disadvantage of textural methods, however, is that they require high computational power and are thus expensive.

Colour features methods exploit the differences in shapes and colour of the text versus that of the background on a number plate in order to detect license plate numbers. The colour modes used are red, green, and blue (RGB), hue, lightness, and saturation (HLS), and hue, saturation, and value (HSV). One algorithm uses the RGB colour space to detect a license plate and the characters on it [18]. In this method, the colour features were joined with the greyscale features to eliminate the background so that the characters stood out and are able to be detected. RGB, however, is limited by illumination conditions. HLS has also been used for ALPR by determining the highest colour density, which typically are the characters, from the license plate region [19]. A drawback of HLS is that it is compromised with environmental conditions, such as an image being taken in the dark or a reflection on a number plate. HSV methods are better able to deal with the problem of illumination conditions and identifies the colour features of a license plate even when the letter is inclined and deformed [20]. Similar to other colour methods, HSV is limited by environmental conditions and also requires powerful processing systems.

Character features methods examine the image for the existence of characters, assuming the characters are from the license plate region. These methods search for characters in the image. If the characters are found, their region is extracted as the license plate region. In [21], instead of using features of the license plate directly, the algorithm finds all character-like regions in the image. These methods are robust to the rotation, but they have to scan through the entire image.

Edge detector methods are the fastest methods for image processing, for that reason we chose one of these methods to work with [1].

3 Mathematical morphology

MM is a technique utilised for the processing of geometrical structures and consists of sets of operators which transform images according to their geometrical features accounting for size, shape, convexity, connectivity, and geodesic distance. MM is excellent for pre- and post-image processing.

Opening is an operator in MM based on the dilation of the erosion of an image by an SE. Opening smooths the contour of an object. In other words, the opening operator removes small objects from the foreground of an image by placing them in the background

$$A \circ B = (A \ominus B) \oplus B \tag{1}$$

In (1), A is the greyscale image, and B is SE. B is subsequently scanned over A. The pixels of A below the anchor point of B are substituted with the calculated value. In the case of opening the calculated value is the result of the erosion of A by B, followed by the dilation of the result by B. For dilation and erosion, the union and intersection operations are used for processing a binary image, while maximum and minimum values are used for greyscale images.

Closing is an operator in MM based on the erosion of the dilation of an image by an SE. Closing smooths regions of contours and removes small holes in the foreground, converting small holes in the background into the foreground

$$A \cdot B = (A \oplus B) \ominus B \tag{2}$$

In (2), A is the greyscale image, and B is the SE. The closing operation is the dilation of A by B, followed by the erosion of the result computed by B. In this case, closing is the reverse of opening.

Top-hat is a morphological operator that extracts elements and details from the image. The operation returns an image with an object brighter than their surroundings. Top-hat is also called 'peak detector'

$$T_{f,b} = f - f \circ b \tag{3}$$

where f is a grey image, b is a structure element, and $f \circ b$ denotes the opening operation of f by b. From (3), the opening operator gets a region with dimension bigger than b, then using original greyscale image f to subtract the result from the opening operator, most of the background is suppressed.

The Top-hat operation has an operator counterpart which is known as the Bottom-hat. It is useful for the extraction of small darker structures

$$T_{f,b} = f \cdot b - f \tag{4}$$

where f is a grey image, b is a structure element, and $f \cdot b$ denotes the closing operation of f by b. In (4), the operation is defined by the residue of closing and the original image.

4 Proposed method

The work presented in this paper focuses on developing a fast, accurate, and stable LPL algorithm suitable for low-cost devices. The developed method can be divided into two operating stages: a LPL algorithm and an optimisation to determine the appropriate values within the SE.

4.1 LPL algorithm

Initially for the LPL, pre-processing starts by converting the original colour image into a greyscale image and then resizing the image. This then is followed by enhancing the license plate through Top-hat morphology transformation. Then, binary thresholding by the Otsu method is used. The next step is noise removal using closing and opening morphology features. Finally, the last step is extracting license plate candidates by finding appropriate contours and validating them against geometrical conditions. The overview of the developed LPL method is delineated in Fig. 1.

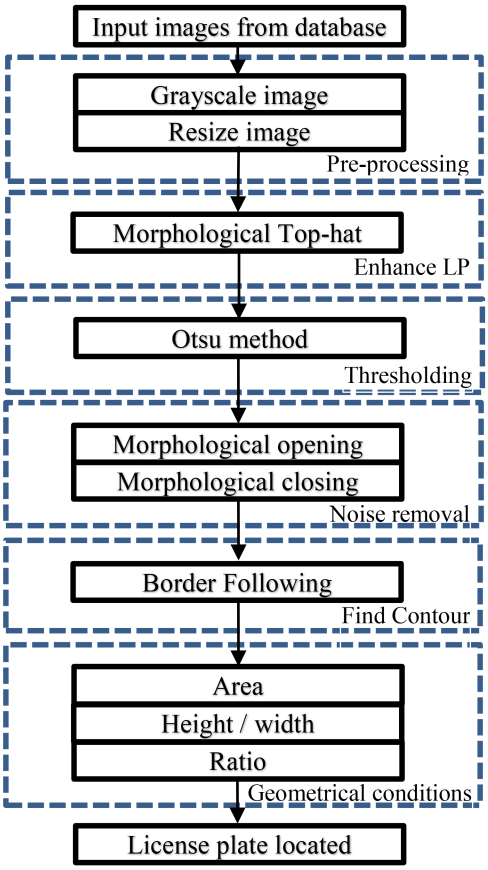

Fig. 1 *Developed LPL*

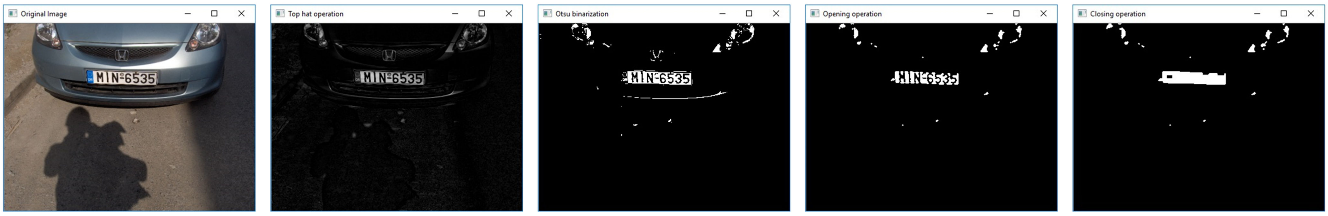

Fig. 2 *Image processing with morphological operations and Otsu binarisation*
(a) Original image, ***(b)*** Greyscale image after Top-hat, ***(c)*** Image after Otsu, ***(d)*** Image after opening, ***(e)*** Image after closing

4.1.1 Pre-processing: License plate cameras capture an image in colour with the captured image being composed of three layers (RGB). The layers are combined, with mathematical processes, into a single layer which is a greyscale image. Then, a resizing process is applied which reduces the numbers of pixels in order to make detection faster [22].

4.1.2 License plate enhancement: The function of the Top-hat transformation is to remove shapes which do not fit with the SE, suppress the background, and enhance the license plate region based on the shape and size parameters defined within the SE. A small SE is necessary to achieve the objective of removing most of the background, but our condition is maintained in the license plate region. It is possible that the SE is slightly larger than the distance between characters and borders.

The horizontal value parameter of the SE should be larger than the maximum distance between the letter and the numerical characters on the license plate [23]. For vertical value, the SE parameter should be larger than the maximum distance between the characters and the outer boundary of the license plate. Since the database contains many different sizes of license plates, these SE parameters are critical to the performance of the system [23]. So, the parameters within the SE will be determined by an algorithm that will help us to optimise the SE to encompass all of the sample images. The algorithm which optimises the SE is detailed in Section 4.2. Since the shape of a license plate is a rectangle, the shape definition within the SE used to localise the license plate should also be a rectangle.

Fig. 2*b* shows the result of using an appropriate SE for a Top-hat transform. As can be observed, most of the background details are removed but the license plate region and its brightness is increased. The Top-hat operation maintains all of the texture features of the objects that fit within the defined SE parameters.

4.1.3 Thresholding or binarisation process: Image segmentation is a way to separate a digital image into various sections. The normal technique for sectioning an image is to partition it according to the intensities of the light and dark areas. Thresholding, e.g. 0–1 or 0–255, pushes pixels which are dim (e.g. those which are below the threshold limit) to zero and all pixels above the threshold limit are pushed towards one.

If $g(x, y)$ is a threshold of $f(x, y)$ at global threshold T

$$g(x, y) = \begin{cases} 1, & \text{if } (x, y) < T \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

The major issue with thresholding is that it considers only the intensity and not the connections between adjacent pixels. There is no assurance that the pixels recognised by the thresholding procedures are adjoining. The thresholding procedure simply disconnects pixels inside the locale. Thresholding is often

Fig. 3 *Proposed LPL method*
(a) Original image, *(b)*Contours detection, *(c)* License plate localisation

subjective so that in one instance we may lose areas by moving too much towards zero, or gain unnecessary details by moving towards 1 or 255.

Otsu's technique [24] is utilised to differentiate between two types of relatively homogenous things such as the foreground versus background [8, 23, 25]. When an image is converted to its greyscale, this single-band image has a bimodal (black and white) pixel distribution. Otsu's technique performs a two-class segmentation by automatically finding an optimal threshold based on the observed distribution of pixel values thereby separating the two classes. Fig. 2*c* shows the result of a binarised image after applying the Otsu technique.

4.1.4 Noise removal: As can be seen in Fig. 2*c*, the license plate is visible, and the background is greatly reduced. However, even though we are able to eliminate a large portion of the background, some white regions remain, and these may cause false positives. To eliminate these areas, a filtering strategy based on morphology operations is designed within the system. In this case, the algorithm applies a morphological opening followed by closing to the binary image. The dimension and the shape of the two SEs are not randomly chosen but are determined by an algorithm which obtains the best shape fit in order to achieve high accuracy.

In Fig. 2*d*, when the binary image is applied in the opening operation, most of the noise information is removed and the characters on the license plate are maintained. It is important to keep the characters to preserve the license plate region otherwise the contour function would divide the plate region into multiple parts.

The next step is to remove the characters and delineate the license plate region based on its rectangular shape. For the closing operation, our system focuses on the delineated rectangular shape. Morphological closing is used to remove small black holes in a white region. The condition to remove the characters from the license plate region is done by ensuring that the dimensions are wider and longer than the line of the characters. These dimensions are determined by the algorithm detailed in Section 4.2.

Fig. 2*e* shows the result of applying the closing operation after the opening operation. As can be seen, most of the characters have been removed, and the license plate region is a big rectangle that could be easily identified through pixel connectivity methods.

The aforementioned steps are applied to an image which contains license plates with dark characters on a bright background. In order to find license plates with white characters on dark backgrounds, the bottom-hat operation can be used instead of Top-hat in the license plate enhancement step. In the noise removal step, closing followed by opening replaced with opening followed by closing. With these changes on the license plate, the region shows as a black rectangle in a white background. Inversion of the colours of the image is required before finding contours. The image will obtain a white rectangle around the area of the license plate and the following steps can be applied for both cases.

4.1.5 Finding contours: After applying noise removal, the resultant image consists of groups of white objects. Contour detection methodology is used to identify the location of the white objects. Contour detection is a technique applied to digital images to outline their external boundaries. This technique is very useful for shape analysis and object recognition. It can be explained as a curve joining all the continuous points which have the same colour or intensity. In the developed work, contour detection is used with the binary image after applying noise removal. Our goal is to apply it to the image with a black background which has only a few white objects thereby making detection simple and fast.

Fig. 3*b* shows the result of applying a contour detection technique. As can be seen, the license plate and a few additional objects are detected. As the license plates have geometrical features which are unique from the other objects, this can be used to differentiate between them.

4.1.6 Geometrical conditions: Once the contour detector finds the candidate objects, the candidates can be selected using the following geometrical conditions: area, inclination angle, ratio, width, and height [8].

The first criterion is to select the area. The system sorts the first ten most likely candidates by area in order to avoid processing smaller and unlikely objects which cannot be a valid license plate region. The second criteria analysed is the inclination angle, which is the angle between the vertical and horizontal vectors. As most license plates are in the horizontal position, this angle analysis can further reduce non-license plate candidates. This absolute angle should be $<15°$. The third criterion is the analysis of the ratio of the height and width of the object candidates. In the database, the ratio is pre-determined to be between 1.7 and 7.56. The larger ratio is needed because, for example, a white car has license plate with a white background. In such a case, the entire back of the car may be selected. The last criteria to be determined is the range of width (W) and height (H). The W range is between 9 and 44% of the width of the image while the H range is between 2 and 13% of the image. Once these geometrical conditions are applied to all of the candidates, the license plate region is successfully localised as can be seen in Fig. 3*c*.

A localisation is deemed to be correctly detected when the external boundary of the license plate is located. The localisation is further validated by using the pattern analysis, statistical modelling, and computational learning (PASCAL) criterion [26]. Since license plates vary in size, this localisation process can become complicated especially when the image quality is compromised because of shadows and dirt on the plate. However, by designing an SE which is flexible and can accommodate many types of plate morphologies allows the system to be accurate while also needing lesser processing time.

4.2 Determining the appropriate values in the SE

The parameters set with the SE are crucial in achieving high accuracy in LPL. The developed system uses three different SEs. Further, when the parameters of one SE changes, it affects the accuracy of the entire system. Consequently, determining appropriate SE parameters can be a significant problem. However, this can be made simpler with the help of another algorithm designed to measure the accuracy of the localisation process.

a

b

Fig. 4 *Different sizes of the license plate image*
(a) License plate of 9% of the width of the image, ***(b)*** License plate of 44% of the width of the image

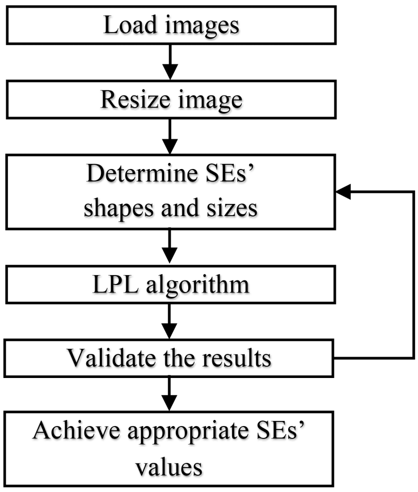

Fig. 5 *Algorithm to determine the appropriate SEs*

Table 1 Optimal SE for varying resolution images

Resolution, pixels	Top-hat SE	Opening SE	Closing SE	LPL rate, %
320 × 240	3 × 13	1 × 4	3 × 8	96.55
425 × 319	3 × 18	3 × 3	3 × 9	98.1
640 × 480	3 × 25	2 × 4	3 × 11	97.59
976 × 732	5 × 38	5 × 4	5 × 19	98. 45
1024 × 768	5 × 38	4 × 4	5 × 22	98.1

First, a training sample set is created by taking several random images from different datasets. All of the selected images are resized to a width of 800 pixels. The license plates on training images are manually located and marked using a software, Objectmarker. In our case, we manually created two rectangles with one covering only the characters on the license plate, and the other rectangle being twice as large as the license plate. When the LPL algorithm identifies a candidate license plate and if this is located between inner and the outer rectangle, it is considered to have successfully located a candidate region.

The training objective of the LPL was to detect the appropriate SEs using three morphological operations: Top-hat, opening, and closing. Image resolution is another feature which can be optimised to achieve accuracy and less processing time. While a low-resolution image is faster to process, high-resolution images are better for segmentation and optical recognition. The technique we utilised was to first resize the license plate image to a lower resolution one. The algorithm is then run to localise the license plate. Lastly, the image is cropped so that only the identified license plate region remains, and the rest of the image is removed. This makes localisation faster and the license plate region clearer. By doing this, the remaining process becomes more efficient and accurate.

The SE used for morphology can accommodate varying sizes and shapes of license plates. Since the majority of license plates are rectangular in shape, the SE for Top-hat and closing is comprised of a rectangle. Top-hat is used to highlight the license plate region and closing to remove black characters inside the license plate. In the case of the opening algorithm, even though it has been tested with different shapes, its objective is to focus on the characters on the license plate while removing the salt noise in the background.

Zhai *et al.* [5] specified the license plate character measurement parameters of the SE (e.g. distance between characters, distance of characters from the plate boundary). As can be observed in Fig. 4, the license plates can be of different sizes within an image. Fig. 4*a* shows the license plate which is only 9% of the image size as compared to Fig. 4*b* where it is 41% of the image. Therefore, it is challenging to find the optimum SE dimensions which can work with differing image sizes. So, to achieve the appropriate size of SEs, an algorithm was designed to analyse the manually labelled training image set and tested for accuracy. Once the algorithm successfully and accurately analysed the training dataset, it was further challenged against the testing dataset.

Fig. 5 shows visually the algorithm used to determine the appropriate SEs for the LPL. First, the images are loaded and these then are resized to a resolution of 800 × 600 but still maintaining the same aspect ratio. Next, loops are created to check the result with different values for each variable. There are two variables for the height (H), one is used in Top-hat and closing operation, and the other in the opening operation. In the case of width (W), each SE has a different variable. Next, a LPL operation is applied to the images and the result is further validated to determine if the localisation operation was precise. When the algorithm completes the entire cycle, the appropriate values for each variable (H and W) are achieved. This process is repeated for different resolution images (e.g. 320 × 240, 640 × 480, see Table 1) in order to obtain the variable values.

5 Implementation and results

The developed approach was tested in MATLAB on a PC with a 2.7 GHz core i7 and 8 GB of RAM. Further, it was implemented in Python on a low-cost Raspberry Pi device with a 700 MHz processor and 256 MB of RAM. The algorithm was first tested by challenging it against the public MediaLab database. We then compared our results, for both the PC and Raspberry Pi, with that of previous works which had used the same database for accuracy and processing times.

5.1 Database

The LPR database used for testing the algorithm is provided by MediaLab of the National Technical University of Athens (NTUA) [9]. The database contains images with different background

Table 2 Successful LPL rate by different sets

Categories	Set 1, %	Set 2, %	Set 3, %	Sets 1–3, %	Set 4, %	Set 5, %	Set 6, %	Sets 1–6, %	Set 7, %	Set 8, %	Sets 1–8, %
Proposed system	100	100	100	100	97	100	100	99.48	100	95.65	98.45
Zhu *et al.* [22]	92.02	82.48	88.73	87.74	87.24	74	90.84	89.45	N/A	N/A	N/A
Zhai *et al.* [5]	97.8	98.3	97.9	98	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Table 3 LPL localisation rates and time comparisons

References	Localisation rate, %	Localisation timer, ms
Proposed system	98.45	20
Anagnostopoulos *et al.* [3]	89.1	117
Zhai *et al.* [5]	98	143
Le *et al.* [12]	97.37	—
Zhu *et al.* [22]	89.45	35.01

scenes which show the front or back of vehicles. The database has images in JPEG format with different resolutions such as 1792×1312, 800×600, and 640×480 pixels. The width of the license plates in the database images varies from 9 to 44% of the width of the image. The license plates are from Europe and have black colour characters on a white background. The license plates have two or three alphabetic characters on the left-hand side, followed by a dash and four numerical characters on the right-hand side. The files within the dataset are composed of three categories: still images, difficult cases, and videos. Previous researchers who employed the MediaLab database used only the still images for their work. However, in our work both still and difficult cases images were used.

The image database contains eight different cases as follows:

- Sample set 1 (136 images): daytime colour images (large), the license plate region is clear, and the image was captured in daylight.
- Sample set 2 (122 images): close view, with large and clear license plate areas on a less complex background.
- Sample set 3 (49 images): images with shadows on the license plate area and the background, and some with uneven illumination.
- Sample set 4 (67 images): day time colour images (small) with images similar to sample set 1.
- Sample set 5 (7 images): blurred images captured by the unsteady camera.
- Sample set 6 (3 images): night colour images captured at night using a flashlight.
- Sample set 7 (26 images): difficult and complex images with shadows on license plates.
- Sample set 8 (161 images): difficult images with shadows and dirty license plates.

5.2 Appropriate SE result

The SE is very important to achieve high accuracy. In the present work, the focus was to determine the appropriate SE parameters in order to best localise license plates with varying image types (e.g. easy, difficult, varying resolutions).

Table 1 shows the appropriate SE parameters for different resolution images and the localisation success rate.

As can be observed in Table 1, with 320×240 resolution images the localisation rate was 96.55%. This is because, even though processing time is the fastest, the characters become blurred in low-resolution images and accuracy is undermined. As the resolution increases, the accuracy of localisation increases as well. This is because the characters on the license plate are clear and easily detectable, albeit at the expense of processing time. However, even though opening SE values are small, a small change in values (e.g. from 1×4 to 2×4) leads to considerable changes in the localisation success rate along with changes to processing time. There are some resolutions that fit an SE better than others. For 976×732 pixel images, even though this resolution is not standard, the localisation accuracy was the highest with 98.45%. On the other hand, when using the standard resolution of 1024×768 the localisation accuracy was 98.1% and also takes a longer to process. For this reason, it is highly recommended that for a given SE, varying the resolution of images to be tested to see which image size results in the highest accuracy and the best processing time. Of course, there will always be a trade-off in the sense that even though a higher resolution image may provide the highest localisation accuracy, this may be at the cost of processing time and power. The resolutions which achieved better results were 425×319, 1024×768, and 976×732 pixels.

The processing time needed to determine the appropriate SEs using this method is dependent on the range of SEs values. For example, the SE dimension range used in Top-hat for the resolution 320×240 was 1–15. First, the algorithm evaluates possible SEs values to determine an appropriate SE value for Top-hat operation. Next, the algorithm evaluates opening and closing combinations and determines their SEs values for better performance. Using a PC with 2.4 GHz Intel Xeon CPU E5-2620 and 32 GB of RAM, the algorithm was able to evaluate the SEs values after 0.2 s per each combination. Overall, completing the entire process needed 4.4 min for 320×240 and 6.9 min for 1024×768.

5.3 MATLAB implementation and results

The MATLAB program developed loads the images from the sample set and displays the license plate location through a green boundary box over the license plate in the image. To evaluate the LPL performance in terms of accuracy, the PASCAL criterion was used. As per the criterion, each predicted boundary box and ground truth boundary box can be encompassed according to the following equation:

$$\frac{c \cap t}{c \cup t} \geq 0.5 \tag{6}$$

In (6), c is the detected region through the algorithm and t is the ground-truth license plate region. All sample sets were tested using the developed algorithm. The localisation success rates for each of the eight sample sets are shown in Table 2.

As seen in Table 2, Zhai *et al.* (2013) used only sample sets 1–3, the images of which are all sharp and well illuminated, to test for the localisation accuracy of their algorithm. Zhu *et al.* (2015) tested their algorithm against images in sample sets 1–6, the images from which vary in resolutions, with some being clear while others are blurred, dark, and with shadows. In comparison, our algorithm was subjected to all of the sample sets and achieved greater accuracy. Our developed algorithm was 100% accurate in all of the clear images 1–3, 99.48% in sets 1–6, and 98.45% for all of the database sample set including the very difficult images with complex backgrounds, shadows, and dirt on the license plates.

Although other researchers used the MediaLab database for their LPL work, they did not specify their results by categories. Therefore, it is difficult to compare results. However, it is useful to compare both localisation accuracy and processing times and they are shown in Table 3.

As can be seen in Table 3, in terms of localisation rate, the system developed by Zhai *et al.* (2013) achieved 98% accuracy but they used the clear images from sample sets 1–3. Zhu *et al.* (2015) achieved 89.45% accuracy when using sample sets 1–6. Le *et al.* (2015) used only 452 images from a database of 573 images and achieved 97.37% accuracy. Anagnostopoulos *et al.* (2006) achieved an accuracy of 89.1%. Our system not only surpasses previous algorithms in terms of localisation accuracy (98.45%) but does so even when challenged with very difficult images, something which the other algorithms did not. It is likely that if the

Table 4 Average FPS in a Raspberry Pi

Reference	FPS
Proposed method	7.14
Weber and Jung [4]	2.15

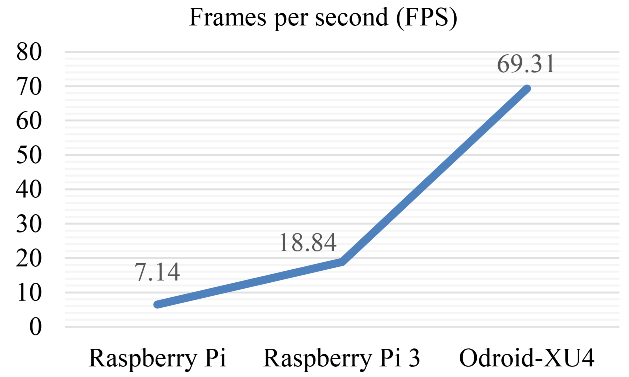

Fig. 6 *Algorithm performance using low-cost devices*

other algorithms were challenged against the difficult images, they would have achieved lower accuracy rates.

In terms of processing time, the data in Table 3 provides the average run times for processing a 640×480 image. As can be seen, our approach was the fastest requiring only 20 ms for LPL. The times of the other researchers varied from 35.01 to 143 ms, with the caveat that one of the authors did not report the processing time.

5.4 Raspberry Pi implementation and result

Our ultimate goal was to create a system which could run on a low-cost and low processing power portable devices. With this in mind, our system was implemented in a Raspberry Pi device. Since MATLAB software runs slower in a Raspberry Pi when compared to other platforms, our system was rewritten in Python 2.7 and OpenCV 3.1 using the same architecture. Linux Raspbian 3.18 operating system was installed in the Raspberry Pi. In addition, a USB camera was connected to the computer board. The developed Raspberry Pi system works as follows:

- The system captures video frames from the camera.
- The frames are processed with the LPL algorithm.
- The license plate is located and its location is shown as a green boundary box.

A video frame is one of the many still images which comprises the complete video. Humans can process and individually perceive 10–12 images per second. A rate higher than 10–12 is perceived as motion. FPS is a useful measurement to show how many video frames a system can process per second. The higher the FPS rate, the faster the image processing. The results compare with previous works as shown in Table 4.

Table 4 shows the average FPS localisation speed of our system versus that of Weber and Jung (2015) with both algorithms being run in a Raspberry Pi. In both cases, the video resolution for testing the platform was 320×240 pixels. Our system reached 7.14 FPS for LPL in real time, more than three times the speed of Weber and Jung (2015). If the algorithm was to be implemented on a device with a more powerful computational capability, the algorithm will achieve higher speeds while maintaining the same accuracy. This shows that our algorithm has utility for real-life situations, such as in a parking system with automatic license plate detection.

The Raspberry Pi was released in 2012, with version 3 being the latest 2017 version. The latter is comprised of a single board with 1.2 GHz 64-bit quad-ARMv8 chip and 1 GB of RAM. We updated our algorithm so that it could work with a pipeline architecture taking advantage of the four cores available in the latest Raspberry Pi 3. By doing this, we were able to increase the FPS to 18.84. Furthermore, we also tested the algorithm in a device called Odroid-XU4. Odroid has a 2 GHz octa-CortexA7 processing power chip and 2 GB of RAM. On this device, we were able to reach LPL detection speeds of 69.31 FPS, as can be seen in Fig. 6.

6 Conclusion

Owing to the importance of the LPL in an ALPR system, an improved algorithm based on morphological operations was developed. In this study, the LPR algorithm developed involves the use of only three morphologic operations which reduces the computational complexity compared to other methods. Thus, license plate enhancement, thresholding, noise reduction, and contour detection were utilised during experiments. Initial license plate enhancement by Top-hat operations reduced the background and highlighted the license plate region. Otsu threshold reduced the impact of uneven luminance regions through a binarisation process. Discontinuous and salt and pepper noise are filtered through opening and closing operations.

A method to determine appropriate SE dimensions for the morphological operations is used for optimisation purposes. A contour method determines regions and using geometrical conditions the license plate regions are located. The MediaLab license plate image database was used for training and testing the system. The developed method achieved 98.45% accuracy of LPL even when challenged with very complex images. Moreover, the MATLAB processing time was 20 ms and were much faster than those of other algorithms. In addition, the system was implemented in several low-cost, low processing power devices. The algorithm was able to locate license plates at 7.14 FPS on a Raspberry Pi version B, 18.84 FPS on a Raspberry Pi version 3, and 69.31 FPS on Odroid-XU4 devices. As demonstrated, the developed algorithm is able to work on both a computer and a low processing power portable device with high accuracy and low processing time.

7 References

[1] Du, S., Ibrahim, M., Shehata, M., *et al.*: 'Automatic license plate recognition (ALPR): a state-of-the-art review', *IEEE Trans. Circuits Syst. Video Technol.*, 2013, **23**, (2), pp. 311–325

[2] Zhao, M., Zhao, C., Qi, X.: 'Comparative analysis of several vehicle detection methods in urban traffic scenes'. Int. Conf. Sens. Technol., 2016, pp. 0–3

[3] Anagnostopoulos, C.-N.E., Anagnostopoulos, I.E., Psoroulas, I.D., *et al.*: 'License plate recognition from still images and video sequences: a survey', *Intell. Transp. Syst. IEEE Trans.*, 2008, **9**, (3), pp. 377–391

[4] Weber, H., Jung, C.R.: 'Low-cost plate detection using a calibrated camera'. Int. Conf. Image Process., 2015, pp. 4763–4767

[5] Zhai, X., Ramalingam, S., Bensaali, F.: 'Improved number plate localisation algorithm and its efficient field programmable gate arrays implementation', *IET Circuits Devices Syst.*, 2013, **7**, (2), pp. 93–103

[6] Sharma, J., Mishra, A., Khushboo Saxera, S.K.: 'A hybrid technique for license plate recognition based on feature selection of wavelet transform and artificial neural network'. 2014 Int. Conf. Reliab. Optim. Inf. Technol., 2014, pp. 347–352

[7] Tarabek, P.: 'Fast license plate detection based on edge density and integral edge image'. IEEE 10th Jubil. Int. Symp. Appl. Mach. Intell. Informatics, SAMI 2012 - Proc., 2012, pp. 37–40

[8] Asif, M.R., Chun, Q., Hussain, S., *et al.*: 'Efficient method for vehicle license plate identification based on learning a morphological feature', *IET Intell. Transp. Syst.*, 2016, **10**, (6), pp. 389–395

[9] MediaLab of National Technical University of Athens.: 'Medialab Lpr database', http://www.medialab.ntua.gr/research/LPRdatabase.html

[10] Ascar Davix, X., Oshin, L., Shamili, P., *et al.*: 'License plate localization by sobel vertical edge detection method', *Int. J. Emerg. Technol. Eng. Res.*, 2016, **4**, (6), pp. 48–53

[11] Salahshoor, M., Broumandnia, A., Rastgarpour, M.: 'Application of intelligent systems for Iranian license plate recognition'. Intell. Syst. (ICIS), 2014, Iran. Conf., 2014, pp. 1–6

[12] Le, T.S., Tran, V.N., Hamamoto, K.: 'A robust and flexible license plate detection method'. Int. Conf. Adv. Technol. Commun., 2015, pp. 326–331

[13] Davis, A.M.: 'Automatic license plate detection using vertical edge detection method'. Int. Conf. Innov. Inf. Embed. Commun. Syst., 2015, pp. 1–6

[14] Tadi, V., Trpovski, Ž., Odry, P.: 'License plate detection using gabor filter banks and texture analysis'. Int. Symp. Intell. Syst. Informatics, 2011, pp. 381–386

[15] Han, H.: 'Method of license plate location based on edge detection and color information'. Int. Conf. Transp. Mech. Electr. Eng., 2011, pp. 1477–1480

[16] Phule, S.: 'License plate identification using artificial neural network and wavelet transformed feature selection'. Int. Conf. Pervasive Comput., 2015, pp. 3–7

[17] Zheng, K., Zhao, Y., Gu, J., *et al.*: 'License plate detection using haar-like features and histogram of oriented gradients'. IEEE Int. Symp. Ind. Electron., 2012, pp. 1502–1505

[18] Yu, S., Xu, Z., Zhang, B., *et al.*: 'A novel algorithm for license plate location based on the RGB features and the texture features'. Int. Conf. Biomed. Eng. Informatics, 2012, pp. 156–159

[19] Hongliang, B.H.B., Changping, L.C.L.: 'A hybrid license plate extraction method based on edge statistics and morphology'. Proc. 17th Int. Conf. Pattern Recognition, 2004. ICPR 2004., 2004, pp. 2–5

[20] Dong, Y., Pei, M., Qin, X.: 'Vehicle color recognition based on license plate color'. Proc. - 2014 10th Int. Conf. Comput. Intell. Secur. CIS 2014, 2015, pp. 264–267

[21] Liu, Y., Yang, J.: 'Research of license plate character features extraction and recognition'. Proc. 2nd Int. Conf. Comput. Sci. Netw. Technol., 2012, pp. 2154–2157

[22] Zhu, S., Dianat, S., Mestha, L.K.: 'End-to-end system of license plate localization and recognition', *J. Electron. Imaging*, 2015, **24**, (2), p. 23020

[23] Ibarra Arenado, M., Maria, J., Oria, P., *et al.*: 'Monovision-based vehicle detection, distance and relative speed measurement in urban traffic', *IET Intell. Transp. Syst.*, 2014, **8**, (8), pp. 655–664

[24] Otsu, N.: 'A threshold selection method from gray-level histograms', *IEEE Trans. Syst. Man. Cybern.*, 1979, **9**, (1), pp. 62–66

[25] Zhi-Yong, Z., Yang, S.: 'The license plate image binarization based on otsu algorithm and MATLAB realize'. Int. Conf. Ind. Control Electron. Eng., 2012, pp. 1657–1659

[26] Everingham, M., Van Gool, L., Williams, C.K.I., *et al.*: 'The pascal visual object classes (VOC) challenge', *Int. J. Comput. Vis.*, 2010, **88**, (2), pp. 303–338