MONASH University
Information Technology

GROUP ASSIGNMENT COVER SHEET

Student ID Number	Surname	Given Names
32457375	Soo	Yong Qi
31989632	Tuan Muhammad	Zafri
31861393	Ong	Yi See

***** Please include the names of all other group members.

Unit name and code	Image Processing, FIT3081		
Title of assignment	Assignment 2		
Lecturer/tutor	Dr.Raveendran Paramesran		
Tutorial day and time	Tuesday 12pm	**Campus**	Monash University Malaysia
Is this an authorised group assignment?	☑ **Yes** ☐ **No**		
Has any part of this assignment been previously submitted as part of another unit/course?	☐ **Yes** ☑ **No**		
Due Date	2/5/2023	**Date submitted**	2/5/2023

All work must be submitted by the due date. If an extension of work is granted this must be specified with the signature of the lecturer/tutor.

Extension granted until (date) .... **Signature of lecturer/tutor** ....

Please note that it is your responsibility to retain copies of your assessments.

Intentional plagiarism or collusion amounts to cheating under Part 7 of the Monash University (Council) Regulations

Plagiarism: Plagiarism means taking and using another person's ideas or manner of expressing them and passing them off as one's own. For example, by failing to give appropriate acknowledgement. The material used can be from any source (staff, students or the internet, published and unpublished works).

Collusion: Collusion means unauthorised collaboration with another person on assessable written, oral or practical work and includes paying another person to complete all or part of the work.

Where there are reasonable grounds for believing that intentional plagiarism or collusion has occurred, this will be reported to the Associate Dean (Education) or delegate, who may disallow the work concerned by prohibiting assessment or refer the matter to the Faculty Discipline Panel for a hearing.

Student Statement:

- I have read the university's Student Academic Integrity Policy and Procedures.
- I understand the consequences of engaging in plagiarism and collusion as described in Part 7 of the Monash University (Council) Regulations http://adm.monash.edu/legal/legislation/statutes
- I have taken proper care to safeguard this work and made all reasonable efforts to ensure it could not be copied.
- No part of this assignment has been previously submitted as part of another unit/course.
- I acknowledge and agree that the assessor of this assignment may for the purposes of assessment, reproduce the assignment and:
    i. provide to another member of faculty and any external marker; and/or
    ii. submit it to a text matching software; and/or
    iii. submit it to a text matching software which may then retain a copy of the assignment on its database for the purpose of future plagiarism checking.
- I certify that I have not plagiarised the work of others or participated in unauthorised collaboration when preparing this assignment.

Signature .... *Date*...

* delete (iii) if not applicable

Signature Tuan Date: 2/5/2023 Signature ______________ Date:________

Signature ______ Date: 2/5/2023 Signature ______________ Date:________

Signature Ong Date: 2/5/2023 Signature ______________ Date:________

Privacy Statement

The information on this form is collected for the primary purpose of assessing your assignment and ensuring the academic integrity requirements of the University are met. Other purposes of collection include recording your plagiarism and collusion declaration, attending to course and administrative matters and statistical analyses. If you choose not to complete all the questions on this form it may not be possible for Monash University to assess your assignment. You have a right to access personal information that Monash University holds about you, subject to any exceptions in relevant legislation. If you wish to seek access to your personal information or inquire about the handling of your personal information, please contact the University Privacy Officer: privacyofficer@adm.monash.edu.au

Table of contents

Part I

Literature review

Introduction

Licence plate is the unique identification of every vehicle. A lot of research has been done on automatic licence plate recognition due to it being very useful in areas such as law enforcement, surveillance, toll and car park payment systems. Licence plate localization is the detection and extraction of licence plates. It is an essential pre-requisite for subsequent tasks in licence plate recognition, such as character segmentation and recognition. In recent years, there are numerous methods to tackle number plate recognition, for this report we will analyse a few of them.

Background

Licence plate localization is a challenging task due to the various factors that affect the quality and condition of the input image. These factors include poor lighting conditions which can cause shadows and reflections that obscure the licence plate as well as varying angle and orientation of the licence plate. Additionally, blurry images, and variation in plate design, such as black or white edges, can also pose difficulties to locate the licence plate.

The localization process generally consists of 2 stages: preprocessing and segmentation. Preprocessing involves the application of filters and other methods to reduce noise and highlight key features of the plate. Segmentation refers to the process of separating the license plate from its background and cropping it out from the image.

Existing methodologies

Ganapathy at el. (2008)

A study by Ganapathy at el. (2008), proposed a method using morphological processes and connected components analysis to locate the licence plate. Based on the paper, the original RGB image needs to be converted into grayscale and subsequently into a binary image in order to be able to work with morphological processes.

The author uses Otsu's thresholding to binarize the image. This is a global thresholding method, the optimal threshold is suggested by minimizing the sum of intra-class variances of the object and background pixels. The results from Otsu's method shown in the study, shows that the characters of the licence plate are clearly separated from its background. Compared to other thresholding methods such as adaptive thresholding and Maximum Entropy Thresholding, the advantage of Otsu's method is that the optimal threshold is automatically selected from a gray level histogram and it is not computationally expensive. (Bangare, 2015). The disadvantage of it is it can be sensitive to noise. The study have a 91% successful detection rate. However, the study recommends the use of adaptive thresholding techniques to increase the detection rate as adaptive thresholding is effective for images with uneven illumination or variable background intensity.

The next step is to use opening to deal with noise in the binarized image, particularly the areas around the number plates. Erosion removes the small objects around the licence plate and smooths the edges and dilation is used to fill in any gaps or holes which further smooth the edges. The segmentation stage involves the extraction of the licence plate region from the binary image. This is achieved by using a region growing algorithm that identifies connected components in the image. The connected component that is most likely to be a licence plate is then extracted and subjected to further processing.

The study uses a modified hough transform. Normally, it is used to detect the borders of the number plate, but in the study, the focus is directed to the characters of the licence plate instead. However, the modified hough transform will usually detect more than one candidate region. To tackle this problem, vertical projection is used to evaluate the candidate regions. The most suitable candidate region is then selected and cropped out from the image and this will be the licence plate.

Shi et al. (2018)

In 2018, Shi et al. proposed a different approach from the one above. The detection algorithm proposed uses a convolutional neural network. The algorithm contains 2 main parts, candidate box generation and candidate box classification and regression. The candidate box is generated based on edge detection methodologies, using grayscale conversion, gaussian blur, Sobel edge detection, binarization and closing operation. These preprocessing steps are similar to the study above, the difference is the study above uses hough transform and vertical projection analysis to evaluate the candidate box, whereas this study uses CNN to evaluate the candidate box.
Two cascaded CNN namely C-Net and R-Net are applied, C-Net is used to locate the area where the licence plate is present, while R-Net is employed to refine the position and size of the detected licence plate based on the output of C-Net. The algorithm is fast and accurate when working with their own collected datasets. However the authors acknowledged that there are still limitations such as, miss detection in actual application where there are multiple licence plate and suboptimal candidate box regression.

Fomani et al. (2017)

Fomani et al. uses grayscale conversion, gaussian blur and histogram equalisation before any other steps are taken.

The authors described the disadvantages of using the Canny edge detection algorithm for licence plate detection. First, it is too sensitive to noise and will result in false edge detection. This will also produce spurious edges in the image, especially in regions with texture or fine details. The next limitation is edge fragmentation, canny algorithm tends to fragment long connected edges into multiple smaller edges. These limitations will cause detection of insignificant areas, which refers to areas in the image that are detected but do not correspond to the actual edges of the licence plate. Therefore the author uses morphological operators and color segmentation techniques to extract candidate boxes. After the candidate regions have been identified using colour segmentation, the authors apply a series of morphological operations, such as erosion and dilation, to refine the regions and remove any spurious regions. After colour segmentation, their algorithm uses local adaptive thresholding for binarising the image. The local adaptive thresholding is calculated based on the maximum and minimum value of width and height of licence plate. Finally, the Canny edge detection algorithm is applied to the refined candidate regions to detect the licence plate edges.

Kaur et al. (2014)

This study uses grayscale conversion and bilateral filter instead of gaussian blur. The study also uses adaptive histogram equalisation for contrast enhancement. For edge detection, the authors used the Sobel operator. This study uses Otsu's thresholding to binarize the image. The authors use opening and closing to improve the quality of edges. Lastly, connected component analysis is used to locate the licence plates among the candidate boxes. This is done by finding the top 10 contours sorted by area in descending order.

Similarities

- All four papers use grayscale conversion and edge detection algorithms to locate the licence plate.
- Fomani et al. (2017) and Kaur et al. (2014) uses histogram equalisation for contrast enhancement.
- All four papers use some form of thresholding, either global or local, to binarize the image.
- All four papers use morphological operations such as opening and closing to refine the candidate regions.
- All four papers use some form of candidate box generation and classification or segmentation to locate the licence plate.

Differences

Evaluating candidate regions

i) Connected components

- Ganapathy at el. (2008), Vertical projection (target licence plate characters)
- Kaur et al. (2014), Find contours (target licence plate edges)

ii) CNN

- Shi et al. (2018), CNN and Regression

Binarize image

- Global Otsu's thresholding
- Local adaptive thresholding

Task Allocation

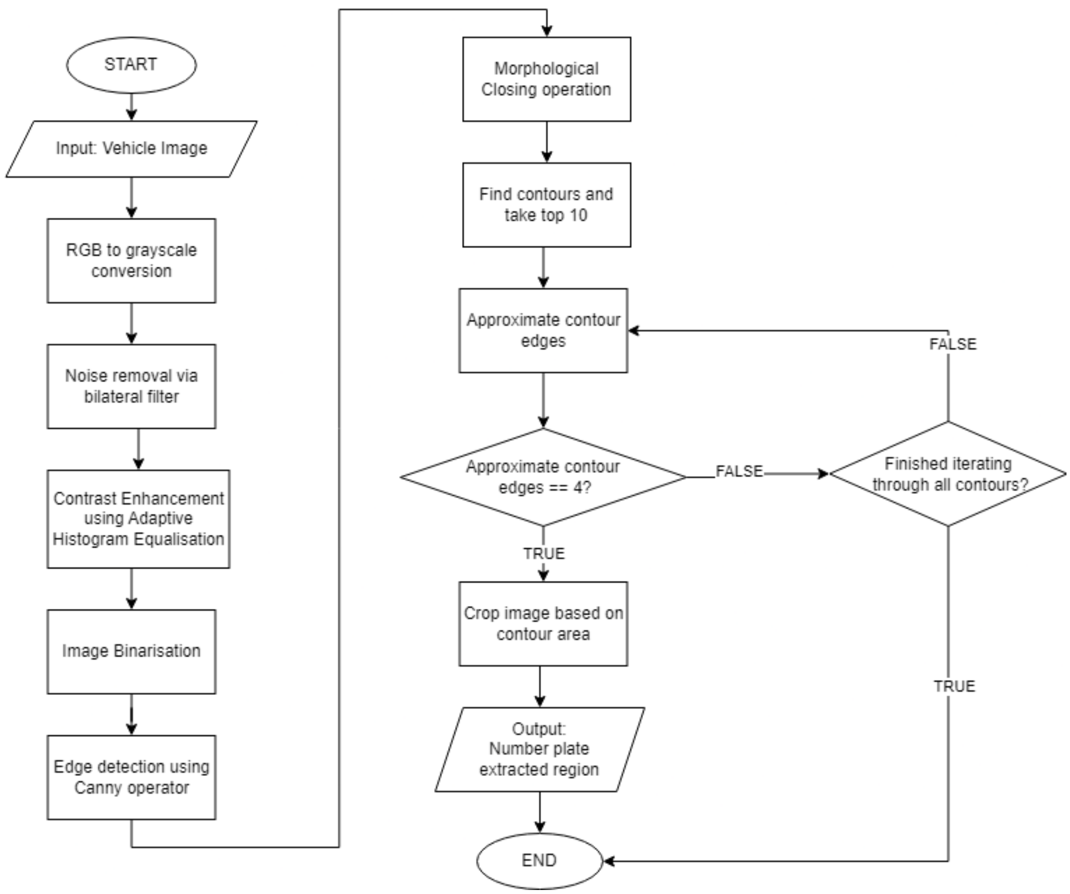
Part I
-All members of the group

Part II
Pre-processing
-All members of the group

Locating
-All members of the group

Part II

Overall Methodology



We are using an algorithm by Sarbjit Kaur et. al. (2014), modified using the algorithm from Praveen (2021). What we are trying to do here is to preprocess the image first by removing noise and balancing out the contrast levels. Then, we want to binarise the image to remove unwanted elements and detect edges. We then close any small disconnected edges and find the top 10 largest contours of the image. We approximate each contour's shape and find the first quadrilateral which is the potential number plate. Then, we crop the number plate region out.

Detailed description of your developed algorithms and the Python functions used in each step.

The main library we are using is OpenCV, as it reads and processes images in NumPy arrays which are easy to manipulate.

Firstly, we read in the image using cv2.imread(). The read image is now in the form of a NumPy array. Then, we perform the grayscale conversion using cv2.cvtColor() with the code cv2.COLOR_BGR2GRAY, telling the function that we want to convert an RGB image to grayscale. The purpose of this step is to reduce the amount of colours we have to work with. Then, we perform the noise removal using an iterative bilateral filter. The noise can occur during taking of the image because of factors such as movement when taking the image, weather conditions, lighting conditions and also camera quality. The bilateral filter is chosen because it does noise removal similar to a linear filter or Gaussian blur, but retains the edges more effectively. We used cv2.bilateralFilter() to perform this.

Next, we performed contrast enhancement. We chose to use an adaptive histogram equalisation because regular histogram equalisation may cause the contrast of the image to become very dark or bright, as it takes the extrema of the pixels. Adaptive histogram equalisation equalises the image in blocks, hence it provides a better and more accurate result. To do this, we used cv2.createCLAHE() with a clip limit of 5, and applied it to the image. Contrast Limited Adaptive Histogram Equalisation (CLAHE) is a variant of adaptive histogram equalisation in which the contrast amplification is limited to reduce this problem of noise amplification. Clip limit is the contrast limit for localised changes in contrast.

In the original paper, subtraction of the grayscale image and a mask created by performing opening using a disc shaped kernel was performed. However, we did not perform this operation as it made the image lose a lot of edges because the image made it darker, resulting in almost a 100% fail rate. It was also hard to reproduce the mask as shown because we did not know the size of the kernel used and the number of iterations of erosion and dilation performed. We then just performed the next step, which was image binarisation. The grayscale image was binarised using cv2.threshold(), using Otsu's threshold. This removes unwanted elements and also made the edges more prominent.

Next, we want to detect edges. We used the Canny operator instead of the Sobel operator as per the original paper. This was because Canny did a better job at detecting edges and the edges detected are smoother, albeit slower than Sobel. Moreover, Sobel needed to be run twice, once for vertical edges and the other for horizontal, whereas Canny only needed a single run. We used cv2.Canny() to perform this with a lower threshold of 50 and an upper threshold of 200. We then performed closing using cv2.dilate() for 2 iterations and then cv2.erode() for 1 iteration with a kernel of size 5. This is done in an attempt to close off gaps in between lines.

Lastly, we used cv2.findContours() to find all contours in the image. This will help us remove straight lines from the image and potentially locate the number plate. We then looped through the top 10 contours sorted by area in descending order, and then approximated its shape by using cv2.approxPolyDP() with its arc length. If it returns 4 edges, then it's a quadrilateral and the potential number plate. We then crop that region out and return it.

Detailed description of the experimental results.

The results were not as good as expected. Out of the images given to us, only 7 plates were detected from a total of 45 images from Set 1, and only 1 plate out of 15 from Set 2. We have observed that the algorithm does a bad job at detecting images with larger quadrilaterals than the number plate. It will detect that larger quadrilateral instead. It also does a bad job when the number plate does not have distinct 4 edges (i.e. when there is text in between a border). It also fails to detect images which are taken through glass as there may be reflections, images which are blurry and grainy and also low quality images. Images where the number plate has edges seemingly combined with a larger edge (underneath the boot handle) will also not be detected.

Discussion / critical evaluation on the outcomes and any observations

We have observed that the algorithm sometimes removes too much information from the image during preprocessing, especially at the binarisation part. It will treat broken parts of the number plate border to be noise and remove it. This will cause the borders of the number plate to not be clear and hence not detected by Canny.

We have also seen that using Canny sometimes provides overly detailed lines around smooth edges. This will cause jagged edges and hence does not work well with dilation, which may make those jagged edges more prominent. Hence, when approximating the shape of the contour, it may detect it as more than 4 edges. The same goes for images without 4 distinct edges around the number plate. Closing may not be able to close the gap fully and it will lead to a broken edge. approxPolyDP() may not be able to identify it as a quadrilateral due to that broken edge.

Images which are taken through glass normally have reflections on them. This will impact the image as we have not done sufficient preprocessing to remove that contour. Canny will pick up the reflection's contour, hence potentially covering the edge of the number plate. Lastly, images where the number plate has edges seemingly combined with a larger edge will cause findContours() to detect the entire combined edge and combine the number plate's contour along with that edge's contour, hence being unable to identify the quadrilateral of the number plate.

In conclusion, we have to take more preprocessing steps to remove as much noise and unwanted components as possible. Instead of taking the top 10 contours, we could instead take all contours and remove overly big and small areas, as the number plate's contour may not be the largest.

References

1. Musoromy, Zuwena & Ramalingam, Soodamani & Bekooy, Nico. (2011). Edge detection comparison for license plate detection. 11th International Conference on Control, Automation, Robotics and Vision, ICARCV 2010. 1133 - 1138. 10.1109/ICARCV.2010.5707935. Retrieved from https://www.researchgate.net/publication/224216764_Edge_detection_comparison_for_license_plate_detection/citation/download
2. Sunil L. Bangare, Amruta Dubal. (2015) Reviewing Otsu's Method For Image Thresholding. Retrieved from http://www.ripublication.com/ijaerdoi/2015/ijaerv10n9_20.pdf
3. Ganapathy, V. & Lui, Dennis. (2008). A Malaysian Vehicle License Plate Localization and Recognition System. Journal of Systemics, Cybernetics and Informatics. 6. https://www.researchgate.net/publication/238757836_A_Malaysian_Vehicle_License_Plate_Localization_and_Recognition_System
4. Y. Shi, and Y. Chen, "License plate detection based on convolutional neural network and visual feature", International Conference on Mechanical, Control and Computer Engineering (ICMCCE), pp. 514-519, 2018.
5. Babak Abad Fomani, and Asadollah Shahbahrami, "License plate detection using adaptive morphological closing and local adaptive thresholding", 3rd International Conference on Pattern Recognition and Image Analysis (IPRIA), pp. 146-150, 2017. https://www.semanticscholar.org/paper/License-plate-detection-using-adaptive-closing-and-Fomani-Shahbahrami/17482f01446fba0638e6fd2c8e872ab6c5cbbe99
6. Kaur, Sarbjit & Kaur, Sukhvir. (2014). An Efficient Approach for Number Plate Extraction from Vehicles Image under Image Processing. https://www.researchgate.net/publication/262843164_An_Efficient_Approach_for_Number_Plate_Extraction_from_Vehicles_Image_under_Image_Processing/citation/download
7. Praveen. (2021, December 15). License Plate Recognition using OpenCV Python. Medium. Retrieved from: https://medium.com/programming-fever/license-plate-recognition-using-opencv-python-7611f85cdd6c