

**GROUP ASSIGNMENT**

**TECHNOLOGY PARK MALAYSIA**

**CT029-3-2-ISE**

**Imaging and Special Effects**

**UC2F1908IS**

**HAND OUT DATE : 5 MARCH 2020**

**HAND IN DATE : 29 MAY 2020**

**WEIGHTAGE : 50%**

**INSTRUCTIONS TO CANDIDATES:**

1. **Submit your assignment at the administrative counter**
2. **Students are advised to underpin their answers with the use of references (cited using the Harvard Name System of Referencing)**
3. **Late submission will be awarded zero (0) unless Extenuating Circumstances (EC) are upheld**
4. **Cases of plagiarism will be penalized**
5. **The assignment should be bound in an appropriate style (comb bound or stapled).**
6. **Where the assignment should be submitted in both hardcopy and softcopy, the softcopy of the written assignment and source code (where appropriate) should be on a CD in an envelope / CD cover and attached to the hardcopy.**
7. **You must obtain 50% overall to pass this module.**

# Acknowledgement

Foremost in our acknowledgement in this project, we are very grateful to out teacher/lecturer in this module (Image and Special Effects), Mr. Hamam Mokayed for guiding us throughout this project, giving his time to answer our questions when we were lost, letting us student grow on our own for improvement and the lastly giving us the basic to advance knowledge and skill in image processing on C++. Without his excellent guidance we would not have been able to achieve on completing this project and its understanding.

We would also like to thank our fellow classmate for helping one group to another, answering some of our minor questions for us and lastly motivating us to finish this project earlier than the deadline.

Table of Contents

[Acknowledgement 2](bookmark://_Toc41666229#_Toc41666229)

[1. Objectives 4](bookmark://_Toc41666230#_Toc41666230)

[2. Problem Domain 5](bookmark://_Toc41666231#_Toc41666231)

[2.1 Licence Plate Detection 5](bookmark://_Toc41666232#_Toc41666232)

[2.2 Optical Character Recognition (OCR) 5](bookmark://_Toc41666233#_Toc41666233)

[2.3 Character Segmentation 5](bookmark://_Toc41666234#_Toc41666234)

[2.4 Tesseract 6](bookmark://_Toc41666235#_Toc41666235)

[3. Algorithm 7](bookmark://_Toc41666236#_Toc41666236)

[3.1 Grey Scale Function 7](bookmark://_Toc41666237#_Toc41666237)

[3.2 Equalize Function 7](bookmark://_Toc41666238#_Toc41666238)

[3.3 Blur Function 7](bookmark://_Toc41666239#_Toc41666239)

[3.4 Edge Detection Function 7](bookmark://_Toc41666240#_Toc41666240)

[3.5 Erosion Function 8](bookmark://_Toc41666241#_Toc41666241)

[3.6 Dilation Function 8](bookmark://_Toc41666242#_Toc41666242)

[3.7 Segmentation Function 8](bookmark://_Toc41666243#_Toc41666243)

[3.8 Binarize Function 8](bookmark://_Toc41666244#_Toc41666244)

[4. Test Plan 9](bookmark://_Toc41666245#_Toc41666245)

[5. Critical Analysis and Comment 13](bookmark://_Toc41666246#_Toc41666246)

[6. Conclusion 14](bookmark://_Toc41666247#_Toc41666247)

[7. References 15](bookmark://_Toc41666248#_Toc41666248)

# Objectives

Below are the objective and function of Licence Plate Recognition on the real world:

* Stolen Vehicle Recognition
* Securing an Area such as a Gated Entrance
* Public Safety by helping Law enforcement
* Speed cameras
* Electronic Toll Collection
* Monitoring Border Crossing
* Recognizing Vehicles at Gas Filling Stations
* Traffic Management
* Drive Through Vehicles Recognition
* Vehicles Park Recognition

# Problem Domain

## Licence Plate Detection

Society nowadays must be familiar with the plate detection system. Almost every part of the city now uses this system to automatically get the license plate of a vehicle whether it is moving or not, but the most common one is located mostly in the parking lot. To get the system to work, it will need a real-time image of a vehicle with a plate on it. Then, the system will convert the image into greyscale, segment it to get the plate and apply recognition using the neural network (Gala et al., 2017). Normally, the sequence of plate recognition would be as follows: capturing, pre-processing, plate region detection, noise reduction, segmentation & identification (Yimyam, W. and Ketcham, M., 2017). Capturing is when the system receive an image with a plate, pre-processing will be process altering the attributes of the image so that it is easy for the program to detect, while plate region detection is pinpointing the location of the plate in the image and then reduce the noises that exist outside the paint and lastly segment it and identify the plate’s character using OCR (Optical Character Recognition).The result of the identification will then be utilized to do the next task the system is designed.

## Optical Character Recognition (OCR)

OCR involves the technique of addressing the character that is inside an image whether it is hand-written or computer-generated text and translate it into language that the computer understands. It will segment & analyse the character inside the image then translate it into American Standard Code for Information Interchange (ASCII) or in other form that the machine could encode (Rao et al., 2016). OCR has been utilized in many apps that has been developed with every possible usage of it and one of the tremendous technologies that has been achieved by OCR now is helping the blind.

## Character Segmentation

Segmentation is the part of the plate recognition process that is also crucial in reading the character inside the plate. Because the OCR module will recognize each of the character one at a time and organize it so it is rearranged to the proper actual initial arrangement. To get the character, program will create a bounding box that surround each location of the candidate character region in the image (Ganapathy and Lui, 2020). Before continuing into the recognition process, the plate will be normalized so that it is allowed in the recognition module by the artificial neural networks.

## Tesseract

Artificial Intelligence has evolved and upgraded many aspects of human life, and one of the technologies developed is Tesseract, an open-source engine which turns texts inside an image into machine-encoded text (Rao et al., 2016). The Tesseract function with the Recurrent Neural Network’s architecture of Long Short-Term Memory (LSTM). What Tesseract expect from the original image is a binary image and then it will process the image bit by bit (Pawar, et al., 2019). The Tesseract will be used in this project as the OCR engine accompanied by the OpenCV to read the image.

# Algorithm

## Grey Scale Function

The Grey Scale Function converts an RGB image (3 channel) to a greyscale image (1 channel) by taking the average of the value of each 3 channels to the 1 channel, changing the new pixel into a grey-scale pixel that retains the brightness of the original RGB image. The justification of using this function first is to modify the image in order to have easier simpler algorithm and faster processing, when only finding and cropping out the plate position.

## Equalize Function

The Equalize Function modifies a greyscale image to more or less even out the brightness of an image. It does this by using the histogram equalization algorithm that first procures the G grey levels (which in this case is 256), then counts for every same value from 0 – 255 in the greyscale image. The result from the last step will then be converted into a *cumulative probability* (CP), and lastly the CP will be multiplied by the G grey level minus 1 (G - 1) to get the value of the new pixel’s value. The new value will then replace the old value in the grey scale image. The reason using this function is to brighten or darken the image accordingly to get finer details/contrast of the image and to easily tell shapes apart with the better contrast.

## Blur Function

The Blur Function will soften the greyscale images by the taking the average of the 8 surrounding pixel and its own value and average it out to get the new pixel value. Continue the process to every pixel in the image and it will blur/soften the image. This is necessary to remove some noises out of the image and blurs into the surrounding pixel.

## Edge Detection Function

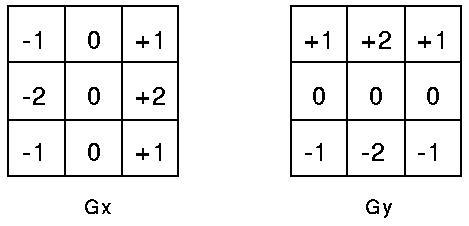
The Edge Detection Function is a function that will transform a greyscale image into an image that represents the outer lines of a shape. This works by applying 2 mask which will be called Gx (Figure 4.4a) and Gy (Figure 4.4 b), which we add together along with the neighbouring pixel to get the new value according to each mask. Both masks result’s absolute value will then be added together to get the new value. Finally, the new values will be binarize according to a fixed threshold.

Figure 2. Gy Mask

Figure 1. Gx Mask

## Erosion Function

The Erosion Function is a function that removes any outlier pixels in the black and white image. This is done by checking if the pixel’s value is 255/white. If yes, it then proceeds to check if the neighbouring pixels have a value of 0/black too. If there is at least one neighbouring black pixel, it will set the new value of the new pixel into 0/black too. This function is for reducing further noises in the black and white image for better and faster processing.

## Dilation Function

The Dilation Function is the exact opposite of the Erosion Function. It further expands/enlarge any white pixel in a black and white image into its surrounding. This works by checking if the pixel itself is black or has a value of 0, then continues by checking if the surrounding has a value of 255 or is white. If they found a white pixel in the surrounding 8 pixels, it will make the new pixel white as well. This is done for expanding the shapes that has already been filtered out by the erosion function to have better processing to find the shapes in the image in later functions.

## Segmentation Function

This function segments out the plate on the image according to some criteria given. It first groups the image’s pixel together according to their proximity by labelling them. It labels by checking if the pixel is white, then checks if the neighbour on top or left is white too. If yes, it will follow the label. If yes but both have label, it will use the minimal label (For example: a & b, a is more minimal than b, and so on). If no, it will create a new label for itself. It will then repeat itself until no changes were made. It will then proceed to check the neighbour on the bottom and right, with the same criteria as before. It repeats until no changes were made. It then segments out all the shapes that has not met the criteria given beforehand (e.g. height, width, position, ratio, etc). This is to finally segment out the plate of the car from the image and to segment out the letters and numbers in the plate into separate images.

## Binarize Function

The Binarize Function is to turn a greyscale image into a black and white image. To decide if a pixel is black or white, it will have a threshold to see. If the image is above the threshold, it will be white and vice versa. This is done to further ease the processing time and efficiency for better segmentation in later parts.

# Test Plan

Detection Accuracy

18/19 = 94.73%

18 out of 19 images are pre-processed the images and extracted the plate by the team successfully. The only image we couldn’t detect and extract the plate is the image 11 (the image with the small red car). The reason why the team couldn’t detect is because this image has too much noises that is attached to the plate. Thus, it makes the plate hard to be extract out of the image.

Recognition Accuracy

13/18 = 72.22%

|  |  |  |  |
| --- | --- | --- | --- |
| Image Index | Plate Number | OCR Output | Image |
| 1 | AHD 6131 | AHD 6131 |  |
| 2 | WRP 525 | WRP 525 |  |
| 4 | NAV 5969 | NAY 5969 |  |
| 5 | WA 5008 C | WA 5008 C |  |
| 6 | WCV 9605 | WCV 9605 |  |
| 7 | RK 8255 | RK 8255 |  |
| 8 | BDF 1490 | BOF 1490 |  |
| 10 | PGE 523 | PGE 523 |  |
| 12 | WWP 9229 | WWP 9229 |  |
| 13 | BHM 9492 | B4H9HE9 |  |
| 14 | BKQ 9784 | BKQ 9/84 |  |
| 15 | JLP 911 | JLP 911 |  |
| 16 | WSY 8789 | WSY 8789 |  |
| 17 | WYM 757 | JVH 7S7 |  |
| 19 | WXA 2198 | WXA 2198 |  |

Table 1. Test Plan Table

The accuracy of the recognition is debatable depends on how you see the result. As you can see from the result, we managed to identify 10 out of 18 images perfectly. There are some images like Image 8 and 16 that we considered it as OCR misread the character properly since we managed to extract the characters of the plate and none of them were hard for the human to recognise the character as other characters. Thus, we considered ourselves as 13/18 since it is OCR issue at here. There are some images like Image 3, 9, 13 are failed to extract by us because the plate is not same as the usual one, the number of the plate is at their bottom side of the characters. Thus, we failed to make the algorithms to go to the next line and read again.

As the table shown, 3 car plates have 1 letter off the actual car plate. This can be seen in car 4, 8 and 14. The reason for this is the OCR which is tesseract tend to mistake some letters and numbers to other letters and numbers such as B to 8, V to Y, I to 1, 5 to S, 2 to Z, vice versa, etc. Without the images being as clear as a letter in a computer or in printed paper, the OCR may make these mistakes. As such, it is possible for the OCR to have these kinds of mistake albeit sometimes.

Another problem in the OCR is that it can only read texts that are align, meaning that lines that are not align such a plate number will fail to be read by the OCR. An example of this can be seen on car 3, 13 and 18. The explanation of this may come from the fact that the OCR training model was only trained on printed text, not from real text written by people and other form text in the real world. One of the images was able to be read albeit with some mistakes, this is because the number, that is not align with the first line, was accidentally erased in the reprocessing stage after the plate segmentation part.

Finally, a reason that the car plate couldn’t be read or had several mistakes in reading it, is because the pre-processing stage after the car plate segmentation couldn’t make the text clear in black on white. Some needed a specific threshold on the binarize function, that it could not be automated and had to be input manually. Another had the same problem but in the plate segmentation part of it, that it needed more blur window size and specific dilation window size as well as edge detection threshold.

# Critical Analysis and Comment

During the assignment, we found encounter some issues that decrease the performance of our algorithms to detect and recognise the plate in the image. Thus, we would like to perform critical analysis to these issues to provide some possible solution for these issues.

First, the algorithm itself is not feasible for commercial use. In fact, our algorithm would only able to produce good result for this dataset because the algorithm is not automated, it is relying on the values that the team assigned for these plates. Therefore, it will fail to recognise most of the images. The solution for this is use Machine Learning to save the data and train the model to tune up the parameter by itself. According to CSDN, the author claimed that the license plate detection is done by the SVM model, and he implemented a class called plate\_Locate to locate the plate and extract as the image to save as a dataset to train the dataset. (CSDN, 2020) The team believe that if we use Machine learning method to perform the detection, the code will be less messy, and it will produce far better result than our algorithm.

Second, the OCR is not trained for the recognising the Malaysian plate number. The team encountered some issues with the OCR couldn’t recognise the characters that we extracted from the plate. The reason for this is the lack of the training for the model, the model hasn’t been trained enough to recognise the character accurately, it makes the model hard to recognise some characters despite these characters which are failed to be recognised is very easy to identified by the human. Hence, the solution for this is to gather the Malaysian plate dataset and spilt the dataset into three sets which are training set, test set and validation set. The training set will be used to feed to the OCR to train the model, the validation model will be used to produce the result and tune up the hyperparameter inside the model. Last, the test set will be used to test the performance of the OCR to identify any overfitting occurs inside the model.

# Conclusion

In conclusion, during the journey on finishing and completing this project, we as a group has learned a bountiful of knowledge and skill on image processing and the programming language, C++. We learned how to tweak an image to better suit a process in the later stages as well as the steps and processes for an image to be readable by a computer. We also learned that the steps on how a computer recognizes an object is similar to that of a human’s thinking, albeit with a few differences. We will continue to use this knowledge and skill gained in this module in the upcoming future project or assignments given to us. We would like to thank everyone again that has helped us on this journey, especially Mr. Hamam.

# References

Blog.csdn.net. 2020. *非常详细的讲解车牌识别easypr\_人工智能\_Qq\_31186123的博客-CSDN博客*. [online] Available at: <https://blog.csdn.net/qq\_31186123/article/details/78661566> [Accessed 29 May 2020].

Epic.org. 2011. *EPIC - License Plate Recognition Systems*. [online] Available at: <https://epic.org/privacy/licenseplates/> [Accessed 28 May 2020].

Gala, R., Doshi, S., M. Jain, Y., Patidar, V. and Nadkarni, S., 2017. Vehicle Number Plate Detection and Recognition. International Journal of Advance Research, Ideas and Innovations in Technology, [online] 3(2). Available at: <https://www.semanticscholar.org/paper/Vehicle-Number-Plate-Detection-and-Recognition-Gala-Doshi/3cf0c6ea27317e74c731f7fc626d71978b81cb73> [Accessed 25 May 2020].

Ganapathy, V. and Lui, W., 2020. A Malaysian Vehicle License Plate Localization and Recognition System. SYSTEMICS, CYBERNETICS AND INFORMATICS, 6(1).

Pawar, N., Shaikh, Z., Shinde, P. and Warke, P., 2019. Image to Text Conversion Using Tesseract. International Research Journal of Engineering and Technology (IRJET), [online] 06(02). [Accessed 27 May 2020].

Rao, N.V., Sastry, A.S.C.S., Chakravarthy, A.S.N. and Kalyanchakravarthi, P., 2016. OPTICAL CHARACTER RECOGNITION TECHNIQUE ALGORITHMS. Journal of Theoretical & Applied Information Technology, 83(2).

Videosurveillance.com. 2015. *What Is License Plate Recognition (LPR)? | Videosurveillance.Com*. [online] Available at: <https://www.videosurveillance.com/tech/license-plate-recognition.asp> [Accessed 28 May 2020].

Yimyam, W. and Ketcham, M., 2017, March. The automated parking fee calculation using license plate recognition system. In 2017 International Conference on Digital Arts, Media and Technology (ICDAMT) (pp. 325-329). IEEE.

# Workload Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| Work | Matthew Axell TP049057 | Ricky Marco  TP048884 | Tan Wei Xian  TP048417 |
| Programming | 33.3% | 33.3% | 33.3% |
| Documentation | 33.3% | 33.3% | 33.3% |