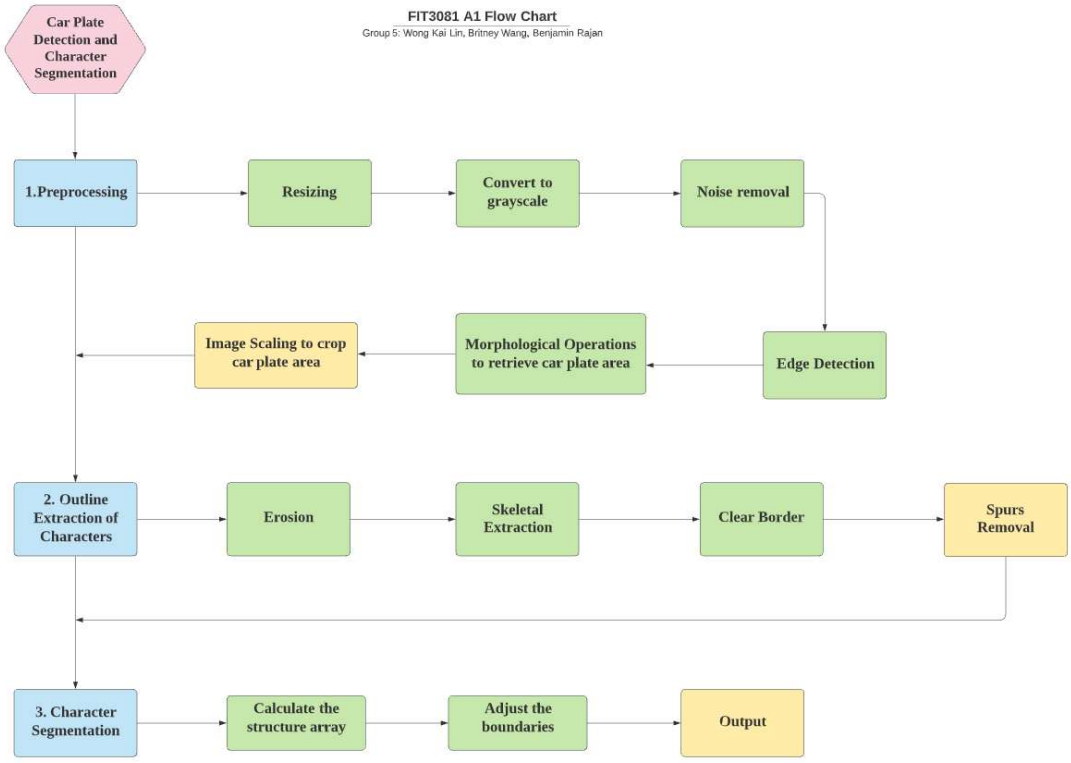
a. Introduction

This report details the steps taken to extract and segment car plate characters from an input image of a vehicle using MATLAB codes and image processing algorithms.

b. Overall Methodology

Flow chart:
https://lucid.app/lucidchart/invitations/accept/inv_a705cb1b-b144-4a18-90f5-9e6847a104c7?viewport_loc=-275%2C-136%2C2205%2C1008%2C0_0



The project of detecting, extracting and segmentation of car plate characters from an image involves 3 main steps, which are: preprocessing, outline extraction, and lastly, character segmentation.

Firstly, preprocessing covers the steps of detecting the car number plate from the image, and then scaling (cropping) the image to only the targeted area, which is the car number plate. Then, outline extraction of the car plate characters is the second step which includes extracting the skeleton of each character and removing some spurs from the extracted skeleton. Border removal is also applied to remove the edges shown of the car plate. Lastly,

character segmentation is done to separate each character. This includes getting the value of the structure array using regionprops function and adjusting the boundaries of characters in the image.

c. Algorithm Description

Preprocessing:

The sample images used can be categorized into 4 types: front view, rear view, side view and cropped plate. The general workflow of the preprocessing is similar. The image is read (`imread`) from the file and saved as a reference variable, we save it as I. For a cropped plate, the image is first resized (`imresize`) to 480x640. Firstly, the image is converted to grayscale (`rgb2gray`) and noise removal (`medfilt2` or `imdilate`) is performed. After that, the smoothen image is binarized (`imbinarize`) for future cropping. For front view, side view and cropped plate, median filtering (`medfilt2`) was used to filter out the noise and smoothen the image. For rear view, dilation (`imdilate`) is performed on the grayscale image to remove unwanted noise.

After that, edge detection (`edge`) is performed. We use Prewitt's edge detection method (`edge(I, 'prewitt')`) for front view, Sobel (`edge(I, 'sobel')`) for rear and side view, and canny (`edge(I, 'canny')`) for cropped plate. Different appropriate threshold values are applied for different algorithms.

The reason we use different edge detection methods is due to the nature of the picture. For canny's algorithm, we can detect edges with minimal loss but this also means that we will detect unwanted edges, thus a suitable threshold is needed. Canny is suitable for cropped plate because there are less objects in the image, as compared to an image of a car with multiple objects in the image. We use sobel edge detection for the rear and side view because sobel due to its nature where edge detection will lose linear gaps. Prewitt is used for front view because it can detect edges whilst efficiently identifying noise in the background.

We then perform vertical dilation (`imdilate`) and horizontal dilation (`imdilate`) to retrieve the lost linear gaps due to edge detection. Vertical and horizontal dilation each use different structuring elements (SE), for vertical SE we use (`VertSE = ones(6, 1)`), for horizontal SE, we use (`HoriSE = ones(1,6)`). This is also done to thicken the edges. After that, we fill (`imfill`) in the gaps of the edges and clear the border (`imclearborder`) to remove any unnecessary edges and connections.

For the front view scenario, edges have to be detected twice. Once is for detecting the estimated plate region, and the second time is to detect the actual plate region.

For the cropped plate scenario, it required an extra step. Since we used canny edge detection so we will have to perform erosion (`imerode`) twice to retrieve the area of the vehicle plate. For rear view detection, an extra step is required where we erode (`imerode`) the image twice to perform thinning. This allows the original image to look more natural.

Now that we have a nice edge detection highlighting the area of the number plate, we will extract the region (`regionprops`) that the number plate is detected. Then, we crop (`imcrop`) the binary image based on the boundary that we have retrieved. Finally, we have successfully identified and cropped the region of the number plates.

Outline Extraction:

The purpose of outline extraction is to reduce the bolded characters of a car plate into a single line of pixels. To achieve this, image erosion (`imerode`) is first used to thin the characters. Erosion also helps to remove some noise by shrinking and removing small specks of pixels which are not the main characters of the car plate. However, erosion operation is not used for the category: Rear view and front view images as some of the sample images in these categories have small plate characters that would be removed during erosion.

Then, skeletonization (`bwmorph(,'skel',N)`) is performed on the eroded image in order to extract the skeleton, also known as the medial axis of each character. This process is repeated until each character outline is reduced to one pixel thick lines. For different images, different repetitions of skeletonization is required to extract the skeletons of the characters, while making sure the characters can still be read clearly. Hence, different configurations of repetitions (`N`) are tried on each sample image until optimum results are obtained.

After that, spurs removal (`bwmorph(,'spur',N)`), also known as pruning, is performed on the skeletonized characters image in order to remove unwanted edge points produced by small irregularities in the boundary of the car plate characters in the original image. These spurs are removed to increase the readability of the characters. The optimum repetitions for the pruning process also differs with different images. Hence, different configurations of repetitions (`N`) are tried on each sample image to ensure spurs are removed without overdoing until the point where characters can't be identified.

Lastly, clear border (`imclearborder(, 8)`) is performed to remove the edges of the car plate that is seen to ease the segmentation process. However, this operation is only performed on the sample images that have clear distinction between characters and plate borders.

The generalized configurations according to the 4 categories:
Front view: no erosion, skeletonization, remove spurs with 1 repetition
Rear view: no erosion, skeletonization, remove spurs with 1 repetition
Side view: erosion, skeletonization, remove spurs with 5 repetitions
Plate only: erosion, skeletonization, remove spurs with 5 repetitions, clear border

Character Segmentation:

Given the spurred images output from the outline extraction, they are being used to segment the characters from the image using regionprops function. The purpose of regionprops, `regionprops(s1)` is usually returning the centroids in a structure array. But in this case, it is returning a value in the structure array based on the extracted outline of the image which will be a condition for the for-loop. The if-statement in the algorithm where the condition, `stats(index).Area > 20` checks the area of the value we retrieved using regionprops function of which it will restrict the boundaries we are looking for. This is why every number plate has a different value because all number plates are different depending on how the outline of extraction was worked on. Then, there is the BoundingBox, `stats(index).BoundingBox(3)*stats(index).BoundingBox(4) < 10000` function which serves the purpose of an imaginary rectangle that serves as a point of reference for object detection. In this case, the object would be the white pixels on the image which are also the characters. The value in the BoundingBox refers to the array of the objects. Then, to retrieve the sub image that contains a range of x and y values from start to end. By doing so, the output of the algorithm would be the number plate's characters separated into individual letters.

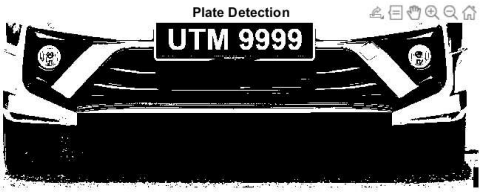
d. Results

Note. Optimal configurations (for part 2: erosion or not, number of repetitions of spurs removal operation, clear border or not) are used to display all results below instead of using the general configurations according to the 4 categories.

Front View Image of Vehicle (Plate 1, 2, 3, 4)

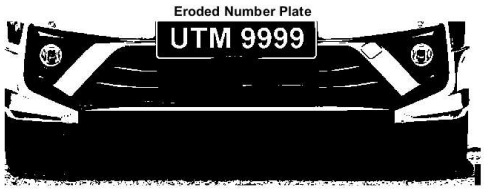
Plate 1:

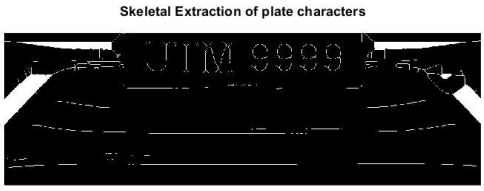
Task 01 - Preprocessing: Detecting & Extracting car number plate area

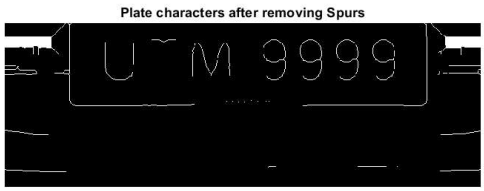


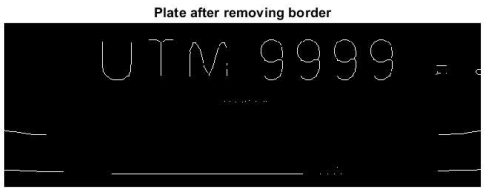
Task 02 - Outline Extraction of plate characters

Configurations applied:
Erosion, Skeletonization, Spur removal with 1 repetition, Clear border









*results are zoomed in (in MATLAB figure) to show that the character lines are actually continuous and there's no breaks (as seen when image results is zoomed out)

Task 03 - Character Segmentation

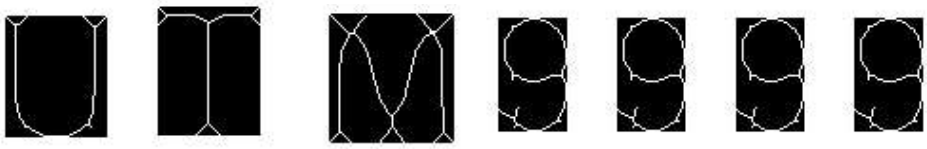
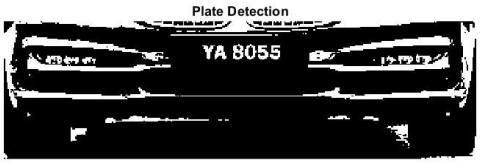


Plate 2:

Task 01 - Preprocessing: Detecting & Extracting car number plate area



Perform second preprocessing to obtain the desired cropped plate region

Task 02 - Outline Extraction of plate characters

Configurations applied:

Erosion, Skeletonization, Spur removal with 10 repetitions

Task 03 - Character Segmentation

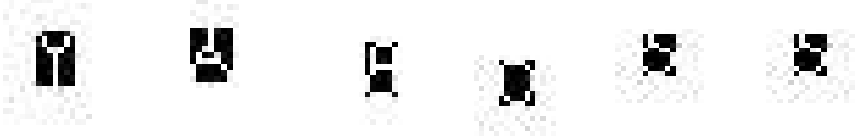


Plate 3:

Task 01 - Preprocessing: Detecting & Extracting car number plate area

Original Plate 3

Plate Detection

Task 02 - Outline Extraction of plate characters

Configurations applied:

Erosion, Skeletonization, Spur removal with 20 repetitions

Eroded Number Plate

Skeletal Extraction of plate characters

Plate characters after removing Spurs

*results are zoomed in (in MATLAB figure) to show that the character lines are actually continuous and there's no breaks (as seen when image results is zoomed out)

Task 03 - Character Segmentation

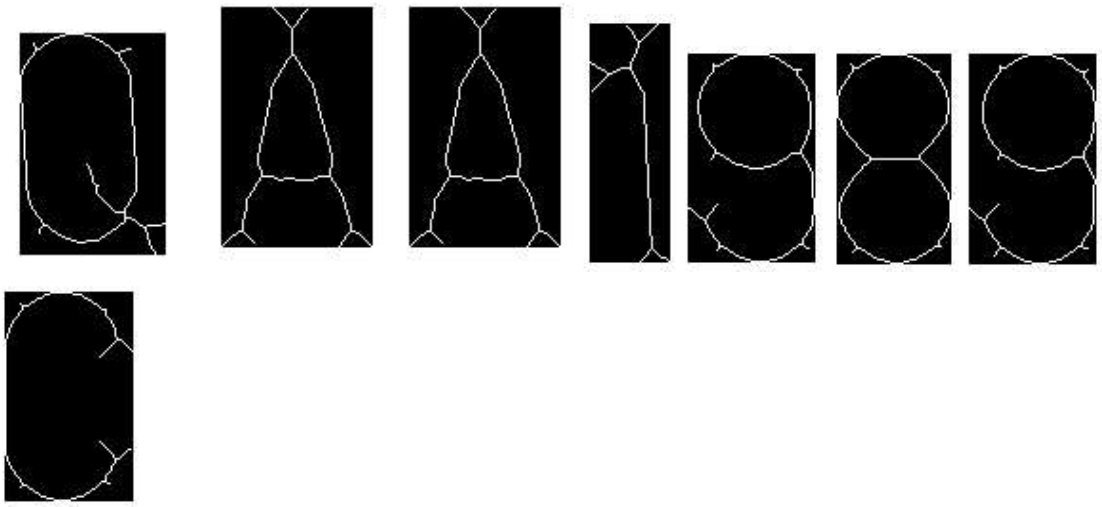


Plate 4:

Task 01 - Preprocessing: Detecting & Extracting car number plate area

Original Plate 4

Plate Detection

Task 02 - Outline Extraction of plate characters

Configurations applied:

Skeletonization, Spur removal with 1 repetition

Skeletal Extraction of plate characters

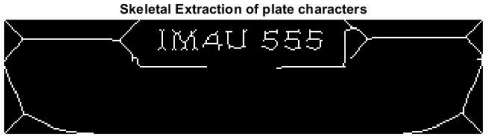
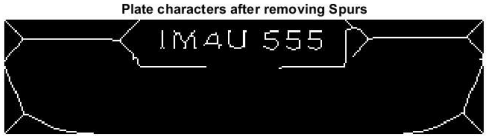
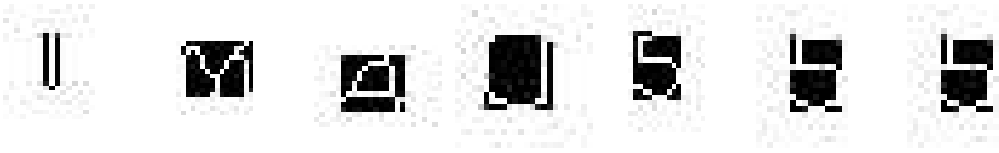


Plate characters after removing Spurs



Task 03 - Character Segmentation



<u>Rear View Images of Vehicle</u> (Plate 5, 6)

Plate 5:
Task 01 - Preprocessing: Detecting & Extracting car number plate area

Original Plate5

Plate Detection

Task 02 - Outline Extraction of plate characters
Configurations applied:
Skeletonization, Spur removal with 1 repetition

Skeletal Extraction of plate characters

Plate characters after removing Spurs

Task 03 - Character Segmentation

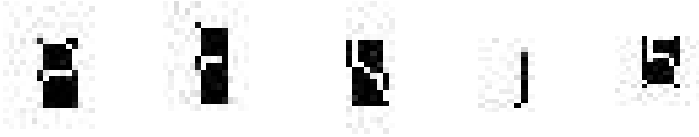


Plate 6:

Task 01 - Preprocessing: Detecting & Extracting car number plate area

Original Plate6

Plate Detection

Task 02 - Outline Extraction of plate characters

Configurations applied:

Skeletonization, Spur removal with 1 repetition

Skeletal Extraction of plate characters

Plate characters after removing Spurs

Task 03 - Character Segmentation

Side View Images of Vehicle (Plate 7)

Plate 7:
Task 01 - Preprocessing: Detecting & Extracting car number plate area

Original Plate7

Task 02 - Outline Extraction of plate characters
Configurations applied:
Erosion, Skeletonization, Spur removal with 5 repetitions

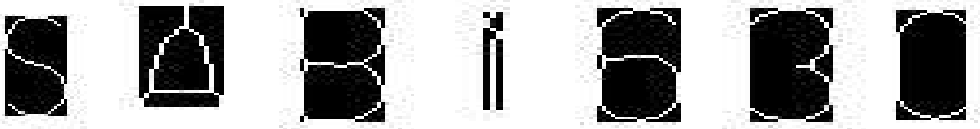
Cropped Number Plate

Eroded Number Plate

Skeletal Extraction of plate characters

Plate characters after removing Spurs

Task 03 - Character Segmentation



<u>Images of Vehicle Plates only</u> (Plate 8, 9)

Plate 8:

Task 01 - Preprocessing: Detecting & Extracting car number plate area

Original Plate 8

Plate Detection

Task 02 - Outline Extraction of plate characters

Configurations applied:

Erosion, Skeletonization, Spur removal with 5 repetitions, Clear border

Eroded Number Plate

Skeletal Extraction of plate characters

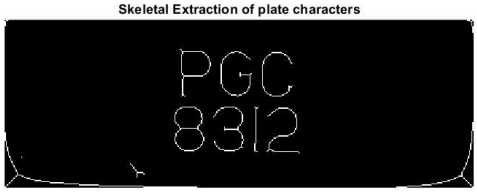


Plate characters after removing Spurs

Plate after removing border

Task 03 - Character Segmentation

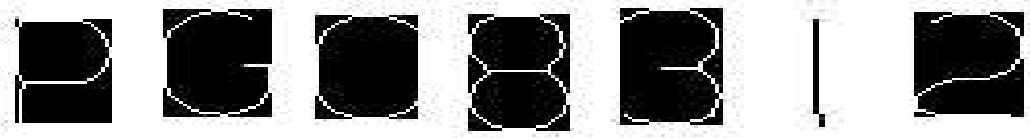


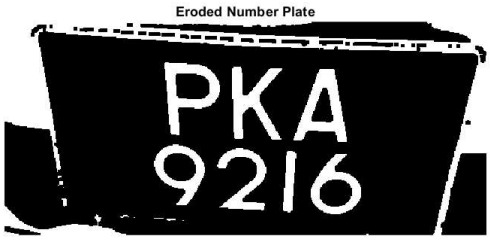
Plate 9:

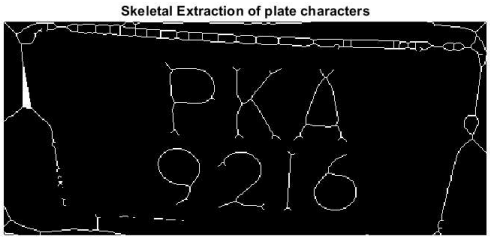
Task 01 - Preprocessing: Detecting & Extracting car number plate area

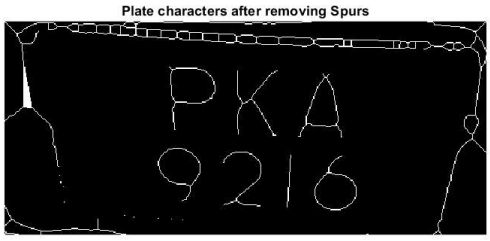
Task 02 - Outline Extraction of plate characters

Configurations applied:

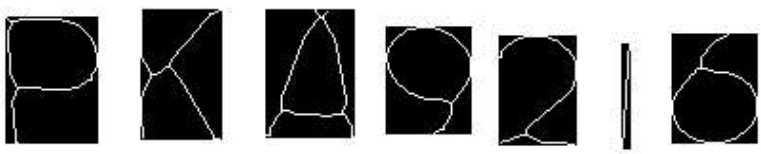
Erosion, Skeletonization, Spur removal with 5 repetitions, Clear border







Task 03 - Character Segmentation



e. Discussion/ critical evaluation on the outcomes and any observations

It is difficult to use one single algorithm that could effectively perform car plate detection, outline extraction of characters and character segmentation on every input image.

Hence, number plates are categorised differently according to the nature of the image (how the plates are captured). Due to this, we need different algorithms to process different types of plates. In our case, we realised that the plates can be categorised into 4 different types: Front view, rear view, side view and plate only.

During the preprocessing of the different types of car plates, different edge detection algorithms were used, including prewitt, sobel and canny. The edge detection is different due to the nature of the image, some images have more noise which includes the car outline or even environmental objects. Suitable edge detection algorithms were implemented to accommodate each category.

A critical evaluation of the preprocessing of images is that some car plates require multiple processing to obtain the perfect cropped car plate region. For example, in the front view scenario, plate 2 is processed twice using different edge detection and preprocessing methods to obtain the desired cropped number plate region. The other plates may not work as efficiently and can be improved by applying different kinds of edge detection and morphological operations. This is a loose bound to generalise the preprocessing methods that should be done on these images.

As for outline extraction, different images required different configurations of algorithms in terms of whether or not erosion is required, and also the number of repetitions for skeletonization and spurs removal. This is because each number characters have different sizes and have different degrees in terms of boldness. Image erosion is applied as a method to remove small pixels of noise. However, this would potentially remove car plate characters which are smaller in size as well. Spurs removal is also preferred after performing skeletonization to remove messy edge points. However, the number of repetitions depends on how bolded the characters are.

With this in mind, we tried to apply the optimal configurations for all sample images, but also generalizing to just 4 different configurations according to the 4 categories in order to minimize the number of codes.

Listed below are the optimal configurations for each sample image:*
Plate 1 (remove spurs with 10 reps, clear border)
Plate 2 (without erosion, remove spurs with 1 rep)
Plate 3 (remove spurs with 20 reps)
Plate 4 (without erosion, remove spurs with 1 rep)
Plate 5 (without erosion, remove spurs with 1 rep)