Computer Vision Assignment 2

Report

Koh Li Hang 18313798

Work Declaration

"This is my own work. Any material taken from other sources has been fully referenced in the text of the work. I have not worked with anyone else on this assignment and have not shown or provided my work to any other person. I have also not made my work accessible to others, both physically and electronically. All sources used in the preparation of this work have been listed in the Bibliography. I have read the statement on plagiarism in the College Calendar and understand that plagiarism is an offence that may result in penalties up to and including expulsion from the University."

Koh Li Hang 7th Nov 2021

Bibliography

No extra material is used, except the template code and libraries provided from TIPS, opencv and lecture slides.

Theory underlying my solution

There are 3 sections in this assignment:

- **Section 1 is the part to obtain the moving object from the video.**
- **Section 2 is to locate the license plate from the moving object.**
- **Section 3 is to calculate the speed of the moving object that we located.**

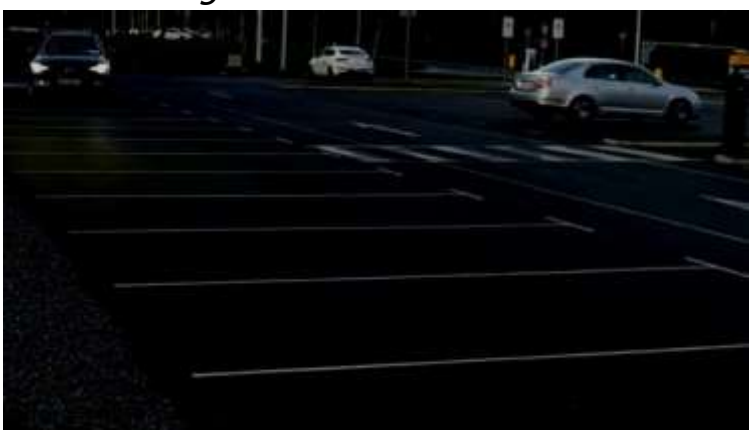
Section 1

Starting with section 1, I decided to apply GMM model on the video to obtain the moving object from the frames. Here are the steps I taken to get a clean object image in the iteration.

1. **Create a GMM model outside the loop, let it keep learning in the process.**
2. **Reduce the brightness of each frame, this will help to reduce the noise that will obtain during the process.**
3. **Apply GMM to current frame.**
4. **Threshold the image obtained from GMM for further cleaning process.**
5. **First apply closing to fill the small holes, and join close regions.**
6. **Then apply opening to remove noise, and smooth the shape.**
7. **Grayscale the image, then apply OTSU thresholding to enhance what is left in the frame.**

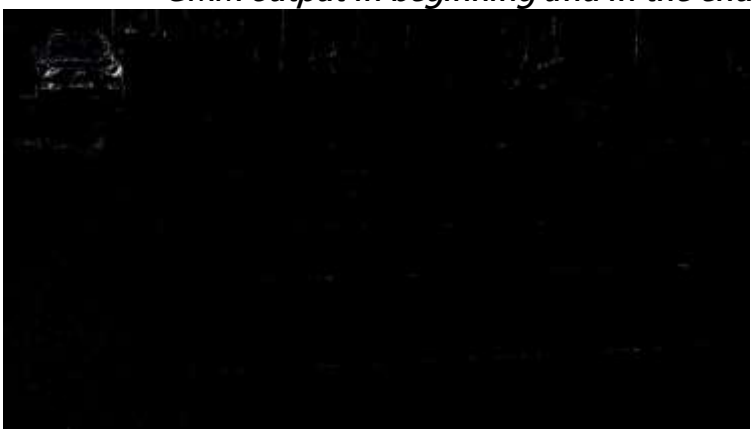
Screen shots

Brightness difference



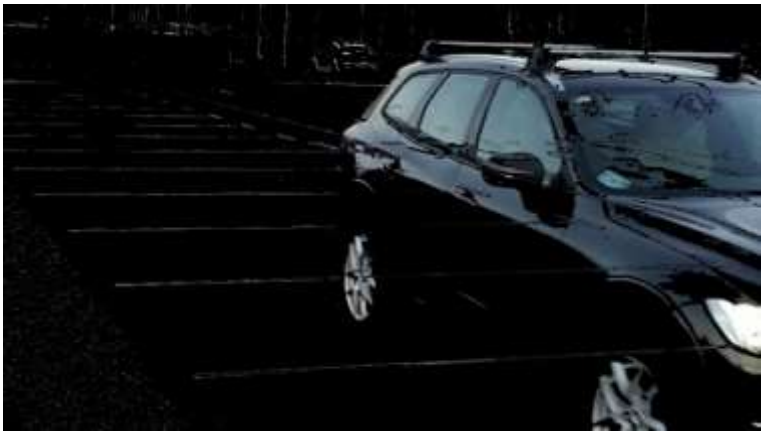

Lowering the brightness meaning it is also lowering the contrast of the object in the frame. But it is helpful here where the object we want to locate is having a higher contrast comparing with others, where part of the noise will not be considered as the contrast has been lowered.

GMM output in beginning and in the end




GMM model take time to learn, so it's ability of obtaining the moving object is weak in the beginning (Left image is taken from frame 8), but it improves when time goes. In the second image, it has successfully obtained the whole moving object (Right image is taken from frame 100).

Otsu thresholding

Otsu thresholding is applied here for 2 reasons, the first is to enhance the object in the frame, and second is to prepare for the next process, which is to pass into the function findContours(). This function only accepts binary image; therefore, it is necessary to threshold the input image.

Section 2

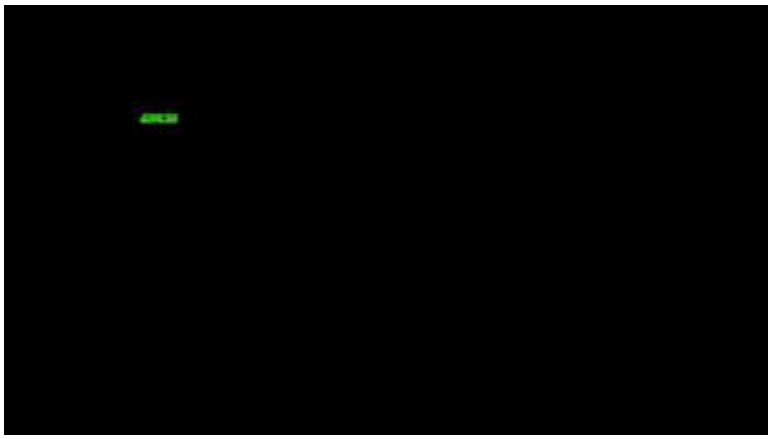
Moving on to section 2, after getting the moving object from section 1, it has lowered the difficulty on locating the license plate. Here are the steps taken to locate the most rectangular object from the frame:

After obtaining a clear moving object from the frame, move on to the license plate location process:

1. **Use Connected Component Analysis to find all the contours.**
2. **Iterates each contour we obtained, and calculate their area, width and height.**
3. **Calculate the aspect ratio, due to the given license plate details, we will want to get the contour with aspect ratio close to 4.65, I set up a range between 4.1 and 4.95 which gave me the best accuracy.**
4. **While iterations, keep track with the ratio and also the area, which we want to find the biggest rectangular object in the frame.**
5. **After iterates all of the contour, if there are multiple contours that are in the pre-set range of the aspect ratio, we will then pick the largest contour.**
6. **Finally, if a potential license plate is found draw a bounding box around it on the current frame.**

Screen shots

Found contours



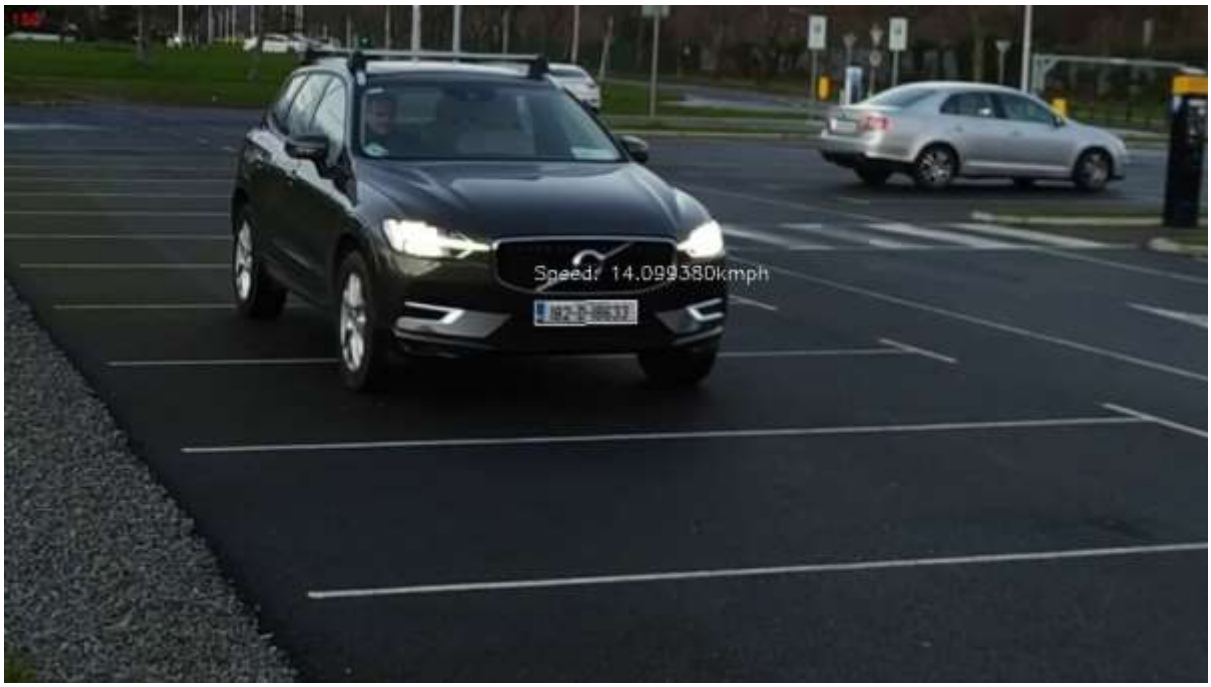
Above are the success and failure examples of locating the plate. But we can see that both of the located contours are having a similar aspect ratio, due to the condition we set.

Section 3

After we have located the potential license plate, we are able to calculate the speed. I use the "Camera Model" method mentioned in lecture, but somehow the speed and distance keep fluctuating, and is unable to get the right travelled distance and speed. Here is how I calculate:

1. **Created a Rect object to keep track of the last X co-ordinate and width to calculate the D_camera(distance from focus to camera) first with Pythagorean theorem as focal length is given.**
2. **Then calculate the D_mm(distance from focus to object) by with formula [D_mm = D_camera*W_mm/W_pixels]. (W_mm is the given plate width, and W_pixels is the width we found)**
3. **Calculate the radian between D_camera and D_mm with SOHCAHTOA rule for further calculation.**
4. **After obtained both D_mm1 and D_mm2, and the radian between them, we can calculate the distance travelled with cosine rule.**
5. **Calculate the time difference between 2 frames, and finally calculate the speed between them.**

Here is the screen shot on how the speed present in the processed video:



Here is the result of travelled distance and speed in frame specified:

Frame	54	70	86	101	115	129	143	158	172
Distance(mm)	5.78	5.10	519.43	390.04	296.59	225.39	162.36	101.71	53.57
Speed(kmph)	0.62	0.55	56.04	42.08	31.99	24.31	17.51	10.97	5.78

Solution Criticism & Other Issues

Brightness reliable Issue

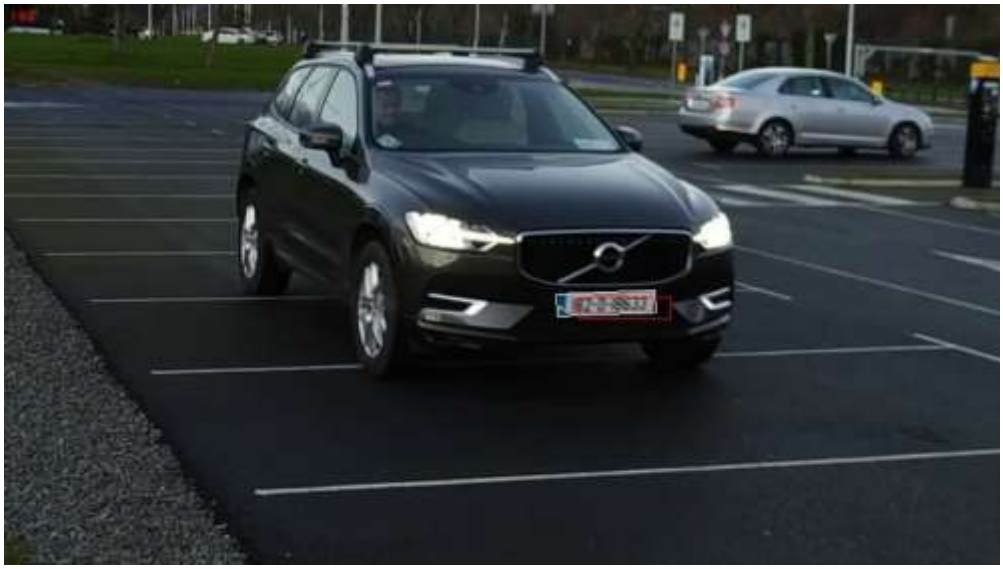
Depend on the video given, the brightness is having a huge impact on the model performance. If high brightness video is given, noise with high exposure will be hard to remove. Thus, increase the difficulty of locating the object we want. Due to my implementation, there's no mechanism to dynamic adjust the brightness of the frame. Therefore, it might cause trouble if high brightness video is given, or low brightness video is given and the model lower the brightness again causing the input to be unrecognizable.

GMM Training Issue

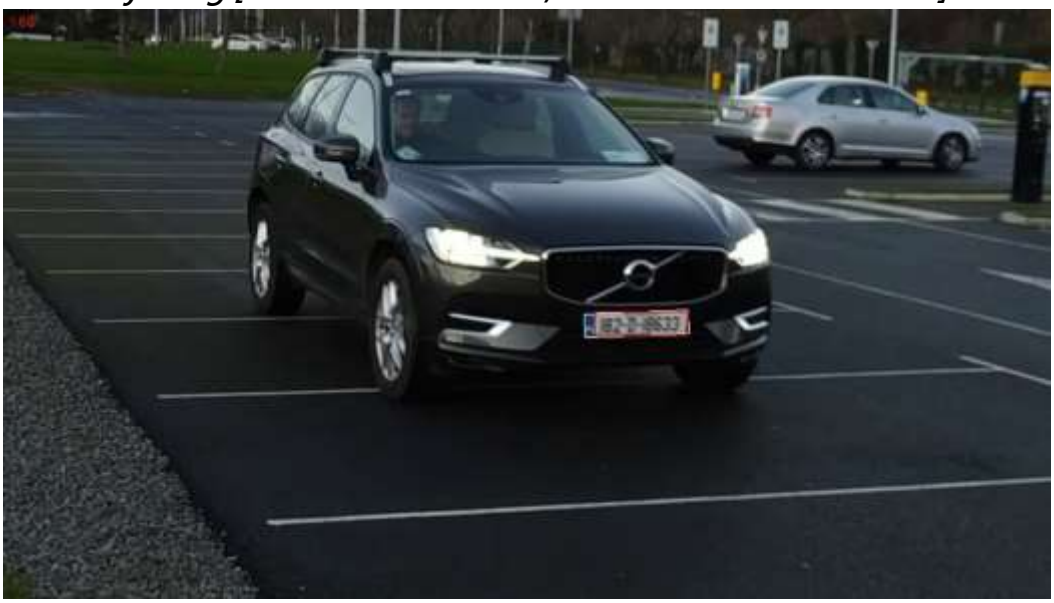
We know that GMM model require time for training to get good performance, since the moving object stays enough long in the given video. Therefore, it is having a nice result after first few frames. But if the object moves really fast, and only stay like 5-10 frames in the video, then the GMM will unable to trained well, and obtain a good result comparing with this case.
Also, if there are multiple moving objects on screen and keeps overlapping on each other will also cause the performance to be bad. GMM will require a longer period to process, and for overlapping objects GMM might be confuse with those situations and lose accuracy in this case.

Plates Location Ground Truth Issue

While playing around the code, I also plotted the bounding box of the given plate location on the video. For some reason, the bounding box is always 1 frame off, it is not obvious in the beginning, but it went obvious during the end of the video. After adjusted the frame by subtracting 1, the bounding box is now appeared in the correct location. Here is the screen shot before adjusting and after adjusting:
Before adjusting [Red—Given Location, White—Detected Location]:



After adjusting [Red—Given Location, White—Detected Location]:



To avoid if it is a mistake, I have added both calculated confusion matrix before adjusting the frame and after adjusting the frame.

Performance Metrics

According to the instructions, we are meant to implement a function DICE to calculate the accuracy. The function will then calculate the ratio of the 2 times intersection area and the ground truth area plus the located area. Here is how I update the confusion matrix, and the performance table.

When an object is located:
If the plate does not exist in the frame, it is a false positive (FP).
If the ratio is greater than 0.8, it is a true positive (TP);
Else if the ratio is lower or equal to 0.8, it is a false negative (FN), as it is being seen as no plate is located.

When an object is not located:
If the plate exists in the frame, it is a false negative (FN).
If the plate does not exist in the frame, it is a true negative (TN).

Confusion Matrix Table without adjusting the frame number

True Positive	151
True Negative	0
False Positive	45
False Negative	25

Recall = TP / (TP + FN) = 0.8579
Precision = TP / (TP + FP) = 0.7704

According to the value of recall and precision calculated above, our model is performing well. As recall can be seen as a measurement of quantity, and precision can be seen as a measurement of quality. Therefore, the model is having a better performance on the quantity comparing with the quality.
Looking into the confusion matrix, there's no true negative spotted. Which means that our model keeps locating the plate even though there's no plate exist on the frame. As we can see there are 45 false positive is spotted, and it is exactly the difference of the total frames and frames that plate exists (221-176=45).
For the 25 false negative, a portion of them is from the beginning. Since I'm using GMM model, and there will be a situation that when the model is learning, it can't obtain the moving object from the video. Therefore, the located plate might be not accurate, or even no plate is located. And I obtained 151 true positive from the model, which is nearly 70% of all frames.
After looking into the matrix and the reasons causing the false judgement, the overall performance is still good

Confusion Matrix Table after adjusting the frame number

True Positive	149
True Negative	0
False Positive	45
False Negative	27

Recall = TP / (TP + FN) = 0.8466
Precision = TP / (TP + FP) = 0.7680
Surprisingly, the model obtained a sightly bad result, with getting 2 extra false negative and less 2 true positive. The given slightly off plate location might create a bigger opportunity for the plate to intersect, so with the exact true location, we might lose some true positives.