BSc. Software/Multimedia Year 3

IPCV - Image Processing and Computer Vision

Intelligent Obstacle and Vehicle Braking Alert System

A03 - Home Assignment

Assignment Guidelines

This assignment is a **home assignment** that is to be completed by **9.00am**, **30th January 2023,** and submitted in the A03 Submission Area within the IPCV VLE course**.**

Some parts of the assignment require a certain amount of **research** to be completed successfully. It is entirely your responsibility to do so.

Assignments without a proper Harvard style bibliography will not be accepted.

The College operates a cheating/**plagiarism policy** and any copied work will be penalized according to this policy.

All the institute procedures rules and regulations apply to this assignment.

At the end of the assignment you should submit the following in one Moodle submission:

- Document with Research tasks. (append any references at the end)
- Project Solution (zipped including source code, outputs and any data sets used)

Project Background

A vehicle obstacle detection system can sense slow-moving or stationary objects when driving at low speeds. Some may even brake for you to avoid obstacles. This feature activates at slow speeds and will provide warnings of impending collisions. For some versions, it will brake the car automatically. Such solutions make use of sensors and/or a camera mounted in the front/rear bumpers and windscreen respectively to determine the distance between the car and nearby objects.

You have been engaged to develop an AI solution for the computer vision based approach using a camera mounted at the top of the windshield to detect:

1. Close obstacles which may be other vehicles and pedestrians, alerting the driver with an 'Obstacle Close' alert
2. Imminent collisions and output a 'Brake!' alert which may also be used by the car anti-collision braking system.

   You have been provided with a short publicly available input demo video and another one demonstrating a sample final output, (https://vle.mcast.edu.mt/course/view.php?id=57#section-18) However, as long as you demonstrate your solution working **you can shorten the video clip** or **use any other appropriate video clip** you like. You are also free to develop your own detection system and a user interface of your choice.

Task 1 – Prepare the scenario

1. Create a new project within your Python+OpenCV environment. You are free to use any distribution, but for convenience you may wish to follow the instructions to set up an Anaconda environment supplied at: https://vle.mcast.edu.mt/course/view.php?id=57#section-1

2. Given you are required to open and process video using windows, for this project it is suggested you make use of Python (.py) files rather than a Jupyter notebook.

3. Download the images and video supplied at: https://vle.mcast.edu.mt/course/view.php?id=57#section-18

4.

   a. Research and explain using an example of each the difference between **affine** and **non-affine** geometric transformations.

   b. Identify **one** geometric transformation which is required to reduce the **size** of the video feed.

KU2.7 - Discuss the effects of geometric transformations on images		
	Maximum	Awarded
4a. Geometric transformation 1 (1 mark for choice, 1 mark for explanation)	2	
4a. Geometric transformation 2 (1 mark for choice, 1 mark for explanation)	2	
4b. Correct choice of transformation to reduce video size	1	
Total	5	

Task 2 – Research object detection algorithms.

SE4.1 -Evaluate and research possible scenarios for the application of object detection and tracking algorithms

5. Three popular algorithms for object detection are:

    - Colour-based object detection.
    - Region-Based Convolutional Neural Networks.
    - YOLO.

    Research each of the above detection mechanisms and write down:

    a. A brief definition of each method.
    b. One advantage and one disadvantage of each.
    c. Choose **one** of the above methods to develop your **Vehicle and Pedestrian detection** project and justify your answer.

SE4.1 -Evaluate and research possible scenarios for the application of object detection and tracking algorithms		
	Maximum	Awarded
5a. Explanation of each method (1 mark each)	3	
5b. One advantage and one disadvantage of each method.	6	
5c. Justification of method chosen for project.	1	
Total	10	

Task 3 – Process a vehicle image.

AA2.8 - Develop and test an algorithm which involves the use of geometric transformations

6. Download the sample street1.jpg image provided with this assignment. The image shows the vehicle with brake lights on. Research and implement an algorithm to:

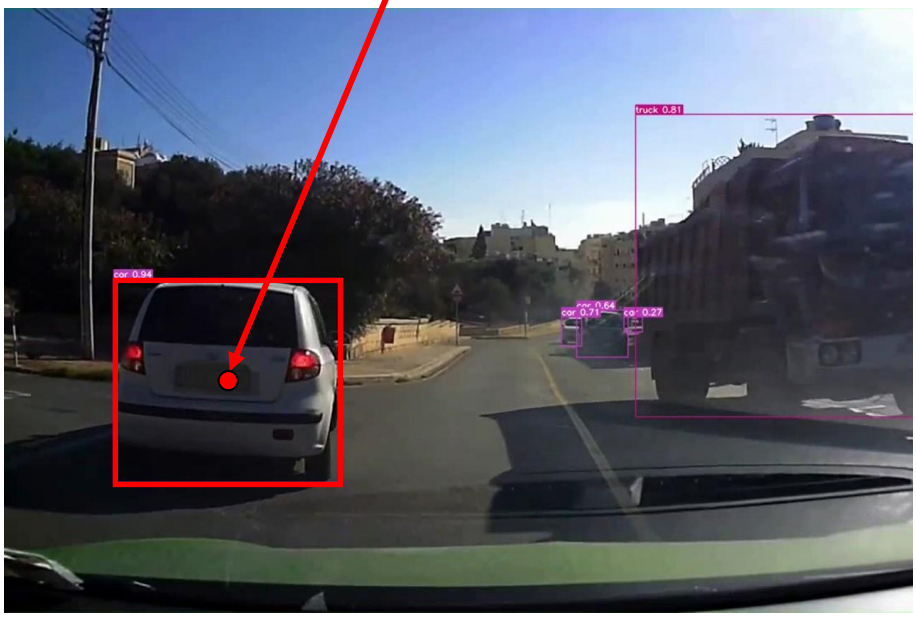
   a. Load and Resize the image to a **width of 1000 pixels** while maintaining the same aspect ratio.

   b. Crop the frame as marked by the red box in the sample below, using the **normalized** (0 to 1) red bounding box centre coordinates, width and height as follows:

```
centroidX=0.251
centroidY=0.611

width=0.240
height=0.324
```

Sample output:



   c. Print in your console the shape of the resized and cropped frame.

   d. Display and save the resized and cropped image.

AA2.8 - Develop and test an algorithm which involves the use of geometric transformations		
	Maximum	Awarded
6a. Load image	1	
6b. Resize image	2	
6c. Crop frames to desired ROI	3	
6d. Display and Save output	1	
Total	7	

Task 4 – Research algorithms that can identify and segment brake lights.

KU3.1 - Discuss the relevance of edge-finding and feature-finding algorithms

7. Research **two** algorithms that can **detect** and **segment** brake lights on or off. Write down a brief description of each algorithm.

8. Using your Task 3 result or a braking car image of your choice, implement **one** algorithm to **detect** if a vehicle's brake lights are on.

KU7 - Discuss the relevance of edge-finding and feature-finding algorithms		
	Maximum	Awarded
7. Research 2 segmentation algorithms	2	
8. Detect vehicle brake lights	3	
Total	5	

Task 5 – Create an intelligent obstacle and vehicle braking alert system

SE3.3 - Compare algorithms that can be used for feature / corner detection
SE4.2 - Combine algorithms in order to create a working object recognition application

9. Implement one or more algorithms discussed in Task 2 to create the intelligent obstacle and vehicle braking alert system:

    a. Choice and setup of a relevant dataset and algorithm for vehicles detection.

    b. Detect a vehicle.

    c. Detect **pedestrians** and **vehicle type**. E.g. **Bicycle**, **Car**, **Truck**, **Motorcycle** or **Bus**.

    d. Distinguish when a vehicle is in front of the windshield camera. Vehicles on the left and right should not be considered by your obstacle detection solution.

SE3.3 - Compare algorithms that can be used for feature / corner detection		
	Maximum	Awarded
9a. Good algorithm choice and implementation of object detection.	4	
9b. Detect vehicle.	1	
9c. Detect pedestrian and vehicle types.	2	
9d. Distinguish vehicles in front of camera.	3	
Total	10	

10.

Using the algorithm implementing task 7 load the **supplied dash-cam video sequence** and perform the following tasks:

a. Load video and display in a named window.

b. Generate output values in the console with coordinates x1, y1, x2, y2 for each vehicle bounding boxes.

c. Highlight bounding boxes with an **orange** colour and display the message **'Obstacle Close'** of top of a vehicle bounding box when a vehicle is close by **approximately** more than a standard car's length, but less than 5 car lengths away.

d. Highlight bounding boxes with a **red** colour and display the message **'Imminent Collision – Brake!'** of top of a vehicle bounding box when a vehicle is very close by **approximately** less than a standard car's length.

e. Save a short output video showing your collision detection system working. (including vehicle bounding boxes and alerts) *(You do not need to output whole video. You can save a few seconds using a sequence of frames that demonstrate your solution working.)*

SE4.2 - Combine algorithms in order to create a working object recognition application		
	Maximum	Awarded
10a. Load video and display in a named window.	1	
10b. Generate output values in the console with coordinates x1, y1, x2, y2 for each vehicle bounding boxes.	2	
10c. Highlight bounding boxes with an **orange** color and display the message **'Obstacle Close'**	3	
10d. Highlight bounding boxes with an **red** color and display the message **'Imminent Collision – Brake!'**	2	
10e. Output video to disk	2	
Total	10	