Individual assignment AE4317 – Q3 of 2020-2021

Assessment weight and learning objective

This individual assignment will form 50% of the grade for AE4317, the other 50% being determined by the group assignment on obstacle avoidance with a Bebop drone. The goal of the assignment is to assess the learning objectives of the AE4317 course at an individual level.

In particular, we will assess how well students are able to select, implement and validate a computer vision algorithm that is intended to be used on an autonomous drone. The product of the assignment is a report, for which the elements will be specified in more detail below. The length of the report (text, main figures and references) should be 3-4 pages. After the fourth page, you are encouraged to include more detection results of your algorithm, so that we can see how well it works / when it fails.

Background of the assignment: autonomous drone racing

Artificial Intelligence is a disruptive technology that can profoundly change aspects of our society. Since 2012 there has been great progress in AI, thanks to the rapid developments in "deep neural networks". The world champion in chess was beaten by IBM's Deep Blue in 1997, humans were beaten at Jeopardy by IBM's Walter in 2011, in Go by Google Deepmind's AlphaGo in 2016, and in StarCraft II in 2019. In aerospace, the next achievement may consist of beating human drone race pilots with an autonomous drone.

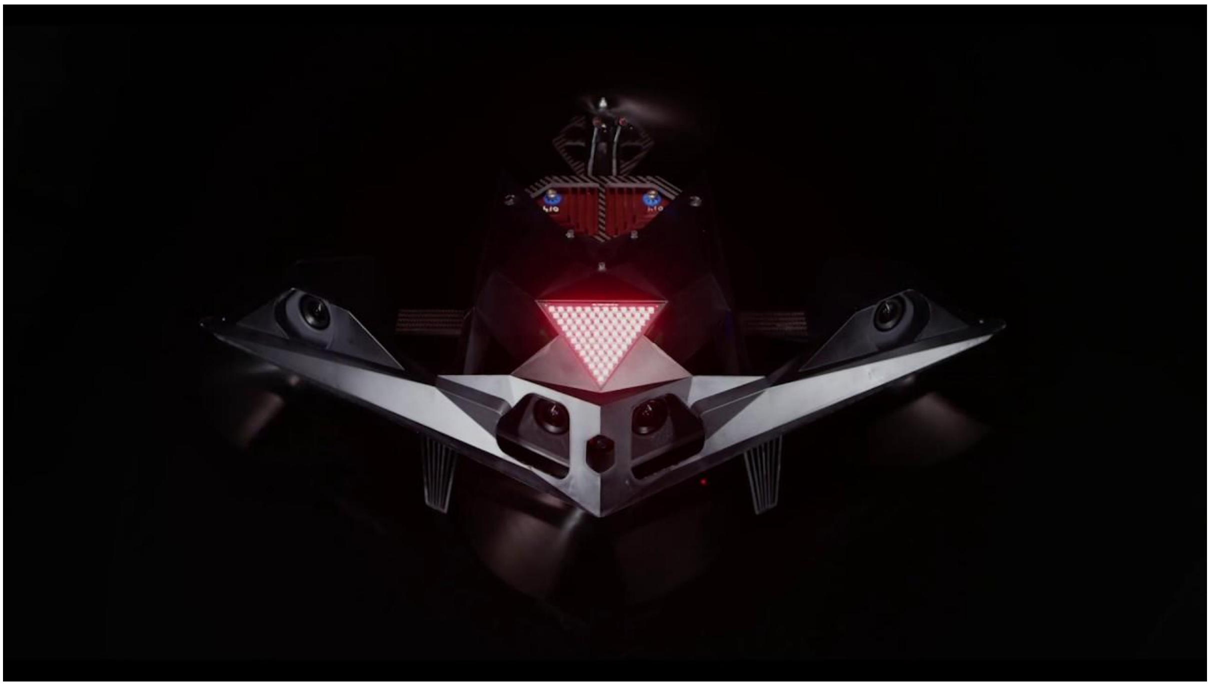

Figure 1: The AIRR racing drone, it has an Inertial Measurement Unit (IMU), a downward pointing laser ranger, 4 high-resolution cameras, and an NVidia Xavier board: 512-Core Volta GPU with Tensor Cores, 8-Core ARM v8.2 64-Bit CPU, 8 MB L2 + 4 MB L3., 16 GB 256-Bit LPDDR4x Memory., 32 GB eMMC 5.1 Flash Storage.

In 2019, the first AI Robotic Racing competition was organized. The goal was to (1) have autonomous drones race together utilizing different approaches to AI and drone racing, and (2) have the winner compete with one of the world's best human drone race pilots, Gab707. 424 teams from 81 countries participated in the competition. At the end, 9 teams competed against each other in 3 seasonal races and 1 world championship race. The TU Delft team became 1st in the championship but lost against the human pilot. For more details: http://mavlab.tudelft.nl/mavlab-wins-the-alpha-pilot-challenge/

Assignment specification

In the AIRR races, the drone had to pass through large gates designed by the competition organizers, the Drone Racing League and Lockheed Martin. Detecting these gates in the images from the drone's cameras was a vital task for the competition.

The assignment's goal is to design an automatic gate detection method. A relevant data set with images from an autonomous drone race and labels for the gates in the images will be provided on Brightspace. **Please note that the code for this assignment can be Python / MATLAB only!** In contrast to the group assignment, here it is not expected that you actually implement the method on embedded hardware.

The **product** of the assignment is a **3-4 page report**, in which the student describes:

1. The choice for a specific object detection algorithm. The student is free to choose any object detection method, e.g., a method based on SIFT-feature matching to a gate template image or a deep neural network method, as long as the choice is motivated in the context of autonomous drone racing.
2. Explanation of the detection algorithm. If the choice fell on a machine learning algorithm, there should be an explanation of the learning process (What part of the data set was used for learning? What were the learning parameters? What was the learning progress?)
3. An analysis of the computational effort spent by the algorithm. What kind of processing hardware do you think is necessary for the algorithm, and how will it affect the weight / safety of the drone? Can you show a relation between parameters of the algorithm and the resulting computation time? Where is the "sweet spot" according to you?
4. Validation of the algorithm, both qualitative (show the resulting detections in images) and quantitative (generate a ROC curve on the basis of the test set, and, e.g., evaluate mean Average Precision or Intersection over Union).
5. A link to (private or public) github with the student's code. **IMPORTANT: do NOT put the dataset on your personal github.** Share a private github with the following users: guidoAI, JuSquare and dewagter.

The **deadline** for the report is four weeks after the competition, **April 30, 2021**. Please send your report to g.c.h.e.decroon@tudelft.nl, j.j.g.dupeyroux@tudelft.nl and c.dewagter@tudelft.nl with as subject:

AE4317 name [your name] id [your student ID]

In order for the students to know how we will assess the 3-4 page report, please see the rubric on the next page.

Rubric

In order to give an idea to the students how the report will be graded, we give here the rubric that will form the basis for the assessment.

	4-6	6-8	8-10
Selection and motivation algorithm	Insufficient motivation for the choice of algorithm, other than that it can detect objects. No reasoning about trade off between performance and computational cost.	Selection of a suitable algorithm. Motivation makes clear why it has been selected for use on a drone, taking into account the trade-off between performance and computational cost.	Selection of a very suitable algorithm. Motivation includes drone racing aspects, showing insight in the wide range of available methods and the specificities of the problem. May provide novel insights in gate detection.
Explanation algorithm / learning process	Explanation is lacking, incorrect, and / or would make reproduction of the learning results difficult at best.	Well-explained algorithm. Main properties of the learning process are explained / demonstrated.	Clear explanation of the algorithm. The learning process is set up well, obeying standards in machine learning, and variations of the learning setup are explored with corresponding effects.
Computational effort	Investigation of computational effort is lacking or highly inaccurate.	A test is performed measuring the time spent by the algorithm on processing the images. An explanation is included on which processor would be suitable.	Solid tests are performed for measuring computation time, e.g., taking into account different images and varying parameters or image sizes. Effects on the required processor and the drone specifications are included in the discussion.
Validation algorithm	Only a few qualitative results are included. Only a single point on the ROC curve is given (TP, FP combination). Lacking interpretation of the results.	A ROC curve is generated for the object detector. Qualitative results show the performance of the algorithm. Good interpretation of the results.	Multiple ROC curves are shown for different settings of the algorithm. Qualitative results show the performance of the best version of the algorithm in important, difficult cases. Very good

			interpretation of the results.
Code	No github link, or no README how to run the code.	README and code allow the assessors to run the code on the test images themselves. Reproduction of the training and testing costs only a moderate amount of effort. Code is understandable.	Effortless reproduction of both the training and test runs of the algorithm. Code is well-documented. Can form a good basis for others to use the detection algorithm for drone race gates or other objects.