1 Discussion and Future Work

This section will discuss some of the current features developed in this project as well as some additional thoughts and ideas. However, not all of these are covered in this rapport but would be the next tasks to focus on, for improving the product as a whole. Both in terms of succession rate of the vision algorithm and additional features which would make the product more attractive for potential investors and end-users.

Robotics:
Due to time constraints the setup of the offboard controller node was only done in simulation and not testing on the actual drone, which was undesirable. Furthermore, testing this on the drone would also have allowed us to work with the optitrack system which would have yielded a great estimation of the trajectory and robustness of the offboard controller node, before talking it outside to test it.

Hardware limitation:
The chosen sensor (Intel Realsense Depth Camera D435, see **??**) had some limitations. It is a great all round sensor also for testing different sensors at a relative low cost. However, to get full HD video at 30 fps a USB 3.0 is needed when using this specific sensor type. However, the companion computer (CC) mounted on the drone, for on-board computations is a Raspberry Pi 3 Model B+ which only support USB 2.0. This resulted in a resolution of only 640x480 if 30 fps was wanted. Upgrading the CC to a Pi 4 could help this issue.

Since, a Pi4 was not available, we should have made an automated path for the drone to follow with a lot of intermediate points where it stops, and snaps a full HD image instead, this would increase the overall flight time, however this is not of importance right now, so this could have been a stable solution to this problem. Furthermore, a manual setup of camera settings by analysing the exposure triangle could have proved beneficial, but this comes with its own issues since these value will change as the environment changes e.g. with different lightings or weather conditions.

Additional sensors for improving robustness:
During the final acceptance test, we identified a potential critical issue with the current system. Using GPS coordinates to make the automated mission for the drone to follow has its issues. This is due to the value of the GPS fluctuating over time. This could have been neglected by using a Ground Control Station (GCS) and a GPS module which supported Real Time Kinematics (RTK). This system potentially has an accuracy below 2cm.
Alternatively one or two additional sensors could be mounted on the drone to maintain a specific ground height as well as distance from the fence. Using these could yield a more stable data capturing sequence of the fence and ensure that the whole fence is completely recorded by the camera for a better end result.

Final Acceptance Test:
To fully test this, the acceptance test should be ran under different weather conditions but due to time constraints as well as corona limitations, only one day of testing was performed.

Additional features:
There are identified some requirements which would be nice to have, given more development time.

A lockable landing box, both to protect the drone from theft as well as weather. This box should be able to open and close automatically when the drone needs to do a perimeter inspection. The team did a initial idea creation for this feature, and would have liked to make the inside landing platform for the drone as an N-fold marker with a smaller ArUco maker inside it, for an automated and precise landing procedure. This feature will further increase the product value as it puts even less strain on the end-user when deployment is needed. This box can also be equipped with a rain and wind sensor to make sure the weather is flyable, otherwise the end-user is alerted and has to manually do a perimeter search. Using these landing boxes makes it easy for larger airports who have a lot more fence to cover, so multiple of these systems could be integrated together to cover the whole fence in one sweep. e.g. having the drone(s) fly from one box to the next by e.g. enabling them at the same time or making use a swarm technology.

Within this box a battery recharging or replacement station must be incorporated for this solution to be fully autonomous.

The mask-rcnn algorithm used to detect breaches in the fence should be trained on a very large (100.000 as a minimum) training set. However, it was quite time consuming to make the set of 7000 which were developed for this project, and it was enough to show a proof of concept. The more versatile the training data, the better the network would be in handling e.g. different lightings and weather conditions. Furthermore to increase effectiveness of the algorithm, a very simple solution is to fly a little higher with the camera tilted abit towards the ground. This will effectively reduce the amount of background noise you have and issues that come with this e.g. see Figure **??**.

In case of hardware or software failure the drone, if possible, lands at it current position and alerts the end-user.

Additional camera facing forward could prove beneficial for obstacle avoidance or person detection in case of trespassing.

Building a new drone that is IP54 water resistant to handle light rain.

Integration with the cloud/stream real time data for processing.

Using high-end CC and sensors would make the system able to maintain a much higher fly speed while capturing data of the fence, which would increase the drones total fence clearance range.