DROOPI : Intelligent Operation Drone

Joël KY
Department of Computer Science and Telecommunications
Toulouse INP-ENSEEIHT
2 Rue Charles Camichel
31000 Toulouse
joelroman.ky@etu.toulouse-inp.fr

Abstract

The positioning of drones in front of a person is a subject that is still open and not much covered in the literature. We will therefore present the various works in the field of drones positioning, as well as our solution developed in a simulation environment, using Artificial Intelligence techniques for people detection (YOLO) and distance estimation (DisNet). This solution is complemented by data assimilation to improve accuracy. We will end with a presentation of the results making it possible to detect persons at about 40 meters and to position the drone in front of them.

***K*eywords** *Unmanned Aerial Vehicle, Human Detection, AirSim, DisNet, Data Assimilation, Deep Learning*

1 Introduction

The use of drones is increasingly present, in particular thanks to its multiple applications. As part of the long project proposed by the SII Group company, DROOPI, we had to work on a rescue drone prototype. This drone should be able to position itself and enter into conversation with the person in distress.
Positioning the drone requires being able to detect the person and estimate the distance between the person and the drone. The detection of the person is done based on the images taken by the camera installed on board the drone. Based on these images we can detect the person as well as the distance between the drone.
This paper will therefore present the first works in the field of drone positioning, the detection of people with the state of the art of YOLO detection, the distance estimation with DisNet and our implementation in the Unreal Engine[1] environment and AirSim[2].

2 Details of the project

2.1 State of the Art

Previous work on drone positioning is not abundant. Numerous articles deal with the detection of persons with drones as part of the [3] deliveries. The article [4] can detect people in hard-to-reach areas. These items are not suitable for positioning drones in front of a person but could be supplemented with step-by-step landing techniques [5].
The article [6] presents the state of the art in the detection of objects, in particular people. This solution can detect people in any environment. This detection of objects coupled to the DisNet [7] method makes it possible to estimate the distance between objects from images. By using these two methods, we can determine the distance to be traveled by the drone to position itself in front of a person in danger. We will describe our solution in detail in the sections to come.

2.2 Human Detection

The YOLO (You Only Look Once) architecture [6], made up of 24 layers of convolutional neural networks followed by 2 connected layers, is very fast and efficient for detecting people in an image.
As part of our project, we decided to use YOLOv4 [8], which is an improved version of the initial version of YOLO in terms of inference time and performance of 12%. This already trained architecture does not need to be retrained to be used in our project. We therefore had to re-use the model available from the following site: `https://github.com/pjreddie/darknet`.
We customized the existing YOLO model, to give with the bounding boxes, an information on the relative position of the person on the image. We can know if the person is on the **left,right or the front** of the image. This information will be used to help the drone for the positioning.

2.3 Distance Estimation

The DisNet solution is as self-supervised learning method that can estimate the distance on images from the bounding boxes predicted by the YOLO model. This solution was trained and tested from railroad images but can be adapted to our use case.
It consists of a neural network of 3 linear layers of 100 neurons each. We trained it on 2000 features of boxes predicted by YOLO model.

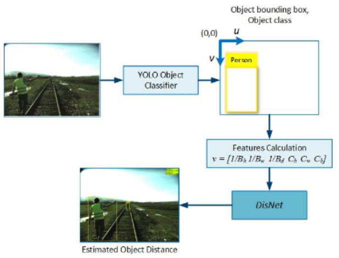

Figure 1: DisNet System

We take as input a 6 dimensional vector v:

$$v = [1/B_h, 1/B_w, 1/B_d, C_h, C_w, C_b]$$

with :
- B_h: height of the bounding box in pixels/image,
- B_w: width of the bounding box in pixels/image,
- B_d: diagonal of the bounding box in pixels/image,
- C_h: average height of an object of the particular class,
- C_w: average width of an object of the particular class,
- C_b: average breadth of an object of the particular class.

For example for the class person, $C_h = 175$ cm , $C_w = 55$ cm, $C_b = 30$ cm.
The model was trained on 1000 epochs, with the optimizer Adam, a learning rate $lr = 10^{-4}$ and an mean absolute error loss.
To get a better precision of our predicted distances, we use the data assimilation process to enhance our results. We then use the Extended Kalman Filter (EKF):

$$\begin{aligned} x^a &= x_B + K(y - Hx_B) \\ K &= BH^T(R + HBH^T)^{-1} \\ P^a &= (I - KH)P^f \\ x^f &= Mx^a \\ P^f &= MP^aM^T \end{aligned}$$

where :
- x_B: apriori state of the system,
- K: gain matrix,
- y: observation of the system,
- H: observation matrix,
- B: covariance matrix of apriori state of the system,
- R: covariance matrix of noise,
- P^a: assimilation matrix at analysis step,
- P^f: assimilation matrix at prediction step,
- x^a: state of the system at the analysis step,
- x^f: state of the system at the prediction step.

By using this on the distance predicted by DisNet as the observation, we can have a better prediction of the distance between the drone and the victim.

2.4 Drone Positioning

By integrating the aforementioned components, we can build, by using the Unreal Engine environment and AirSim packages, a drone that will positions in front of a person in need. The drone will first take images. These images will be pass through the YOLO model that will detect if a person is on this image or not. If there is a person, the drone will use the boxes predicted by YOLO to create the 6-dimensional vector v to estimate the distance and will move accordingly.

3 Results & Future applications

3.1 Results

The results obtained after a pass through the YOLO model, we can get these images with the score of confidence.

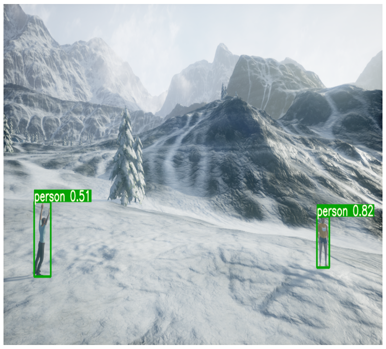

Figure 2: YOLO predictions

The results after training the DisNet model on the dataset, can be seen below on the loss curves of the training and validation set (Fig3).

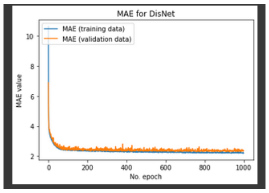

Figure 3: Loss curves

We can also have a look on the mean absolute error between the distances predicted by our trained model and the real ones (Fig 4).

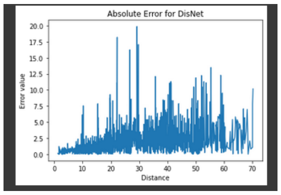

Figure 4: Mean Absolute Errors

3.2 Limits & Future applications

The predictions made by our models are quick enough to be considered as real-time. Nevertheless, our performance are influenced by the distance between the drone and the victim, the angle of view [9]. The code and experiments need to be implemented and tested on a real drone to have a better idea of the performance.
Moreover, for simplicity we assume that our person in distress will be standing and waving his hands, and the person is detected on day. This situation is not always the case in real life and our model will not performed well on detection on night due to bad visibility.
For future applications, we can test the solution of Reinforcement Learning which have proved better for autonomous navigation as well as for drones and for cars.

4 Conclusion

Our solution combining people detection with YOLO, distance estimation with DisNet coupled with data assimilation allows us to have satisfactory results.
However, points of improvement are to be expected to improve performance. Future work, making it possible to carry out detections at higher altitudes or to directly develop on-board and high-performance solutions on drones; are necessary before this tool can be used in a real situation.

5 Acknowledgements

We thank Mr. Sandy Vaslon and Mrs. Coline Moinet from SII Group for their assistance and advice throughout this project.

References

[1] Epic Games. Unreal engine.

[2] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017.

[3] Safadinho David, Ramos João, Ribeiro Roberto, Filipe Vítor, Barroso João, and Pereira António. Uav landing using computer vision techniques for human detection. *Sensors*, 20(3), 2020.

[4] Gotovac Sven, Zelenika Danijel, Marušić Željko, and Božić Štulić Dunja. Visual-based person detection for search-and-rescue with uas: Humans vs. machine learning algorithm. *Remote Sensing*, 12(20), 2020.

[5] Jamie Wubben, Francisco Fabra, Carlos T. Calafate, Tomasz Krzeszowski, Johann M. Marquez-Barja, Juan-Carlos Cano, and Pietro Manzoni. Accurate landing of unmanned aerial vehicles using ground pattern recognition. *Electronics*, 8(12), 2019.

[6] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2016.

[7] Muhammad AbdulHaseeb, JianyuGuan, Danijela Ristić-Durrant, and Axel Gräser. Disnet: A novel method for distance estimation from monocularcamera. 2018.

[8] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection, 2020.

[9] Hwai-Jung Hsu and Kuan-Ta Chen. Face recognition on drones: Issues and limitations. In *Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, DroNet '15, page 39–44, New York, NY, USA, 2015. Association for Computing Machinery.