A CNN-based Drone Localisation Approach for Autonomous Drone Racing

Jos Arturo Cocoma-Ortega *[1] and J. Martinez-Carranza[1,2]

[1]Instituto Nacional de Astrofisica, Optica y Electronica , Puebla, Mexico
[2]University of Bristol, Bristol, UK

ABSTRACT

In this paper we present a CNN architecture to automatically estimate the position of a drone, in metres, relative to a gate in a race track. The latter arises in the context of the autonomous drone racing competition where the challenge is to design a drone that can beat a human in a drone race. There have emerged different proposals to address this problem. Notably, localisation of the drone in the race track is one of the first capabilities that could lead to a solution. However, global localisation may require sophisticated methods such as odometry or SLAM that may become expensive to be computed on board. Furthermore, global localisation may drift as the drone runs the track. Motivated by the latter, we present a CNN architecture based on the Posenet network, which was designed for camera relocalisation in real time. Nevertheless, we have adopted, modified and re-trained such network to the context of relative localisation w.r.t to a gate in the track, which can be exploited by the autonomous navigation algorithms for the race. We report an average performance of 50 fps and a maximum up to 100 fps in a low budget computer with a modest GPU, thus outperforming similar works in the state of the art.

1 INTRODUCTION

Autonomous Drone Racing (ADR) is an open challenge that focuses on having to beat a human in a drone race. This task leads to various challenges, such as localisation and drone control navigation. To know where the drone is, represent a fundamental task in the planning for autonomous navigation, in the last decade several works were focused on estimating the pose of a robot by means of using a single camera and a techniques such as visual odometry or visual simultaneous localisation and mapping, with good accuracy in the estimation, but with the caveat that such estimates may be obtained at low frame rates (20 - 30 Hz). Pose estimation at high frequency is desirable as it could be exploited in agile flights, such as those expected in a drone race. Even proposals that have been employed in ADR competitions operates at 10 fps.

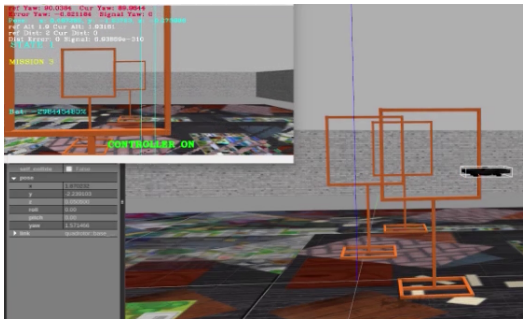

Figure 1: We design a method for Autonomous Drone Racing based on CNN for pose estimation and an algorithm for autonomous navigation. See `https://youtu.be/5rboqinFXYo`

Motivated by the above, in this work we proposed an algorithm for Autonomous Drone Racing based on Convolutional Neural Networks aiming at estimating the pose of the drone relative to the gate and at a high frequency. Similar works have achieved this but at a frame rate of 10 fps. In contrast, our proposal achieves an estimation speed of 100 fps on average with GPU and 20 fps on average with CPU.

To describe our approach, first we will discuss the related work in section 2, then we will describe the methodology used to design and train the network and how we use the pose estimation for autonomous navigation in section 3. Next, we will present the testing results showing that we can estimate the pose up to 100 fps, in section 4. Finally, conclusions are discussed in section 5.

2 RELATED WORK

In recent years, the problem of estimating the position of the camera has been widely studied. There are two main approaches to Visual Odometry: geometrical approach and deep learning approach.
Visual SLAM is one of the most used algorithms to known the robot (camera) position in navigation. V-SLAM solves the problem of localisation and mapping the environment by

*Department of Computer Science at INAOE. Email addresses: {cocoma, carranza}@inaoep.mx

landmarks and features from the frame observed [1]. Deep learning-based algorithms have explore different ways to estimate camera pose. We can found in literature works that uses CNN as main algorithm and shows the viability of the results instead geometric ones [2, 3], other ones resolves localisation via V-SLAM in where estimates VO and also generates a map of the environment [4, 5]. One relevant work is the reported in [6, 7] where they propose a Network they called Posenet. Posenet is based in GoogLeNet [8]. The main contribution of the work is the change of the softmax classifier in the last three layers by a regressor to estimate the pose of the camera. They report high accuracy in their results. Also, it is reported a real-time computation for pose estimation, a time of $5ms$.
There are some works, focus in the estimation of the pose of an object in the image, this is the scope of the works published in [9, 10, 11, 12].
Seminal works addressed the problem to autonomous navigation by using visual odometry or visual SLAM algorithms to resolve the drone's localisation [13] and then generate a planning based on the pose of the drone. In this same context, the works [14, 15] describe an algorithm to autonomous navigation by detecting the gate objective and develop a planning route for the drone flying. Using traditional computer vision, the works presented in [16, 17] propose a strategy based on colour pixels of the gate for detection (four corners) and subsequently, the problem of perspective n-point (PnP) is solved to estimate the relative position of the drone. Other approach based on gate detection by the use of deep learning is presented in [18], they propose a modified SSD network they called ADR-Net to gate detection and then they propose a guidance algorithm based on LOS vector guidance to performs autonomous flight to cross the gate.

3 METHODOLOGY

For autonomous navigation in drone racing, the principals approach have shown an efficient way to planning navigation knowing the position of the gate.

In this work we propose an approach to pose estimation based on gate position, this means, not to estimate the pose of camera based on the whole scene, instead take the gate as reference an estimate how far is the camera from the gate.

We propose a CNN solution based on Posenet [6]. Posenet allows to estimate 6D camera pose for a complete scene outdoors and indoors. We are only interested in 3D camera pose, it means only translation is required for this work, thus we modify the regressors layers to outputs only position (x, y, z) and set the euclidean distance only for translation for the learning algorithm. Also is eliminated a image normalization (mean subtraction) due the use in continuous video images (real-time). For this propose, this modifications are made in a complete network and also is designed a reduced one for increasing network rate predict (see figure 2).

The dataset was designed in simulated environment using gazebo. The scene is created with two gates only, the reason

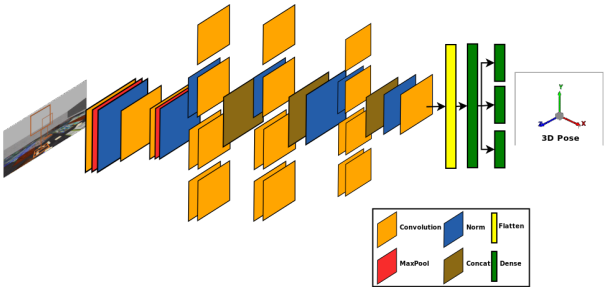

Figure 2: Reduced Posenet architecture.

for this is because when the drone is far enough of gate one, it can see the both gates. Then the drone flies towards the gate and when it is close to the gate one the camera will be in a blind point from that gate, this means that the drone will no be able to see the gate one. Thus, gate two will appear in the line vision producing a new estimation of the pose related to gate two. It is for that reason that the dataset is design in that way, figure 3 shows an example of the gates in the Gazebo scene.

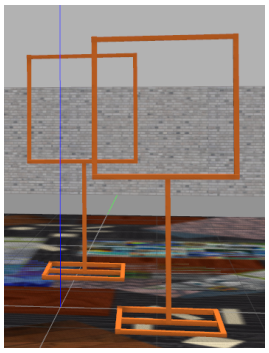

Figure 3: Example of gates used for training.

Using the gates designed as shown, the pose of the drone is calculated used gazebo model state, but not from scene origin, the pose is related to the gate one. Thus the groundtruth is created related the distance of the drone from the gate one, the figure 4 illustrates how is the pose of the drone taken in the simulator.

The pose estimation calculated by the CNN, is used to develop autonomous navigation. The algorithm developed calculates the trajectory adjustment necessary to fly through the gate. In the first step, the drone aligns its position to the center of the gate (y position), and then when it is centered the algorithm commands to fly the distance necessary to close the gate. As we describe early, when the drone is in the blind point of the gate, then predict the position to the next gate, with this new position, it is estimate the distance left to cross the gate. When the gate has been through, the algorithm restarts the process to fly and cross the next gate. The methodology described is illustrated in the figure 5.

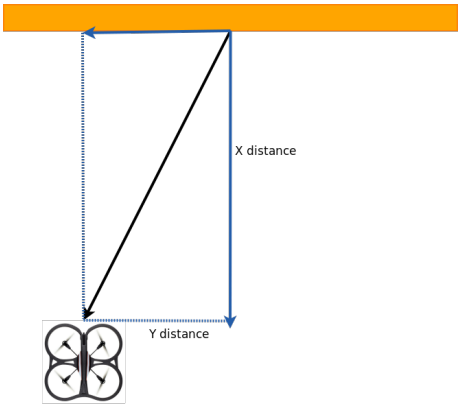

Figure 4: Pose of the drone related to the center of the gate, top view.

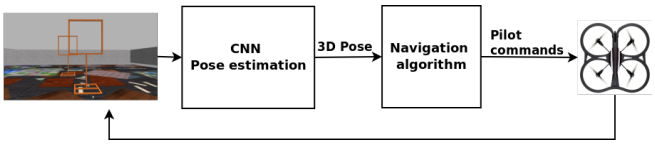

Figure 5: Proposed methodology.

4 EXPERIMENTS AND RESULTS

The experiments were conducted by the use of simulated environment using gazebo. This section describe the results obtained in each experiment.

4.1 Pose evaluation

To evaluate the pose predicted. ROS framework was used to communicate simulated environment (Gazebo) with the Predictor (Modified Posenet) and RVIZ. The experiments performed showed that exist a precision zone for the prediction due to the design of the training dataset. Inside the precision area (this area has size 2.2m x 2.5m), the mean error decreases and prediction is close to the groundtruth, the figure 6 shows the evaluation of the pose predicted displayed in RVIZ, also is attached to figure a white rectangle indicating the precision area detected.

The error calculated inside and outside the area indicates that when the gate is the line vision of the drone the error decreases (inside area), but the error increases as the drone flies away leaving out of the line of vision to the window. The poses were compare calculating the distance between them (error). Figure 7 plots the errors in the navigation test. As the drone flies insider of the precision area, the error decreases to a mean of 0.16 m. Even if the drone is inside the area, the orientation also affects the pose estimation, the more oriented to the front of the gate, implies the less error in prediction.

To evaluate the performance of the Reduced Posenet, navigation tests are carried out in the same way as the Modified Posenet. It can observes from figure 9 that the pose predicted is close to groundtruth inside the precision area in the same

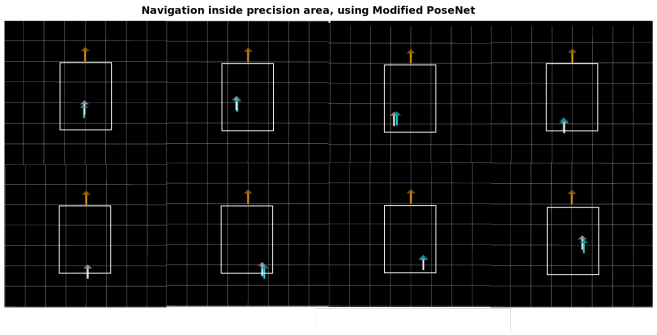

Figure 6: Predicted pose compared with groundtruth using RVIZ. White arrow shows Groundtruth and Blue arrow Predicted.

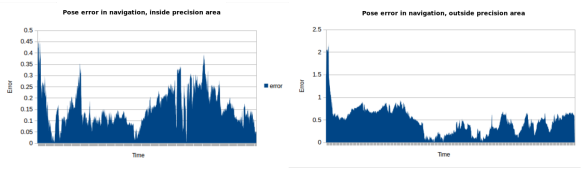

Figure 7: Error over time in navigation. Left graph shows the error while navigating inside the precision area. Right graph shows the error while navigating outside the precision area.

way that Modified Posenet, besides the error increases more outside the precision area. This is not really significant, because when the drone is flying towards the center of the window automatically will be placed inside the area and the prediction will be best for the navigation algorithm.

We have found a similar behavior for the error in the inside and outside area when we plot the error (position differences between predicted and groundtruth) over the time in a navigation. This is showed by figure 8.

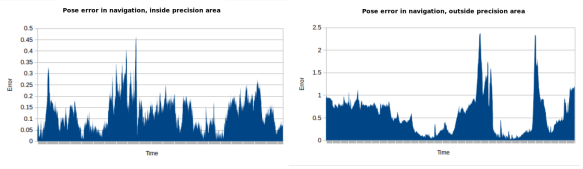

Figure 8: Error over time in navigation. Left graph shows the error while navigating inside the precision area. Right graph shows the error while navigating outside the precision area.

We have test the algorithm for autonomous drone racing. Using the scenario from Gazebo, via ROS framework, we communicate the pose prediction with the algorithm for navigation to the Gazebo world. In the world presented in figure 10, we put three gates in the line vision of the drone to evaluate at first the prediction in the autonomous navigation. The algorithm correctly estimate the pose from the gate and command the drone to center the gate to fly across. As the se-

quence shows, the drone flies satisfactorily through the gate and then stop and oriented to the next gate.

In the table 1, we summarise the error results of both approaches as well as the frequency of process of the pose prediction. The best performance is for the Reduced Posenet, that has minimum error inside the precision area and has the highest frame rate operation for prediction.

	Inside ε	Outside ε	Frame rate (GPU)
MPoseNet	0.1597 m	0.4866 m	50 fps
RPoseNet	0.1285 m	0.5867 m	100 fps

Table 1: Results in navigation testing Modified PoseNet (MPoseNet) and Reduced PoseNet (RPoseNet) for both inside and outside precision area of prediction. ε is the mean error of the predictions over the time of navigation.

All the test of the algorithm were conducted in a computer with a GTX 860m, 16Gb of RAM and an i7-4710HQ CPU.

5 CONCLUSIONS

Autonomous Drone Racing represents a big challenge to develop efficient algorithms that can beat a human pilot in navigation. Localisation at high-speed is still one of the principal problems to solve.
In this work, we have shown that it is possible to estimates 3D pose of a drone relative to a gate in real-time, and at high frame rate.
To achieve this, we have developed a dataset that allows the proposed CNN to learn the pose of the drone with respect to the gate with a low error. In addition, we have designed an algorithm for Autonomous Drone Racing based on the pose obtained from the CNN.
The tests performed in simulation shows goods results with low error.
We report the highest rate for pose prediction at 100 fps (20 fps with CPU) with our reduced Posenet for and with a low error of around 13 centimetres, which still enables our navigation algorithm to centre the drone w.r.t the gate to then command it to cross the gate.

As future work, we will improve the test for real-world scenarios to evaluate the pose estimation and the autonomous drone navigation.

REFERENCES

[1] H. Casarrubias-Vargas, A. Petrilli-Barcelo, and E. Bayro-Corrochano. Ekf-slam and machine learning techniques for visual robot navigation. In *2010 20th International Conference on Pattern Recognition*, pages 396–399, Aug 2010.

[2] Nolang Fanani, Alina Strck, Matthias Ochs, Henry Bradler, and Rudolf Mester. Predictive monocular odometry (pmo): What is possible without ransac and multiframe bundle adjustment? *Image and Vision Computing*, 68:3 – 13, 2017. Automotive Vision: Challenges, Trends, Technologies and Systems for Vision-Based Intelligent Vehicles.

[3] Alec Graves, Steffen Lim, Thomas Fagan, et al. Visual odometry using convolutional neural networks. *The Kennesaw Journal of Undergraduate Research*, 5(3):5, 2017.

[4] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Toward geometric deep slam. *CoRR*, abs/1707.07410, 2017.

[5] K. Tateno, F. Tombari, I. Laina, and N. Navab. Cnn-slam: Real-time dense monocular slam with learned depth prediction. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6565–6574, July 2017.

[6] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2938–2946, Dec 2015.

[7] A. Kendall and R. Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4762–4769, May 2016.

[8] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[9] Patrick Poirson, Phil Ammirato, Cheng-Yang Fu, Wei Liu, Jana Kosecka, and Alexander C Berg. Fast single shot detection and pose estimation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 676–684. IEEE, 2016.

[10] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[11] EV Shalnov and AS Konushin. Convolutional neural network for camera pose estimation from object detections. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 42, 2017.

[12] Thanh-Toan Do, Ming Cai, Trung Pham, and Ian Reid. Deep-6dpose: Recovering 6d object pose from a single rgb image, 2018.

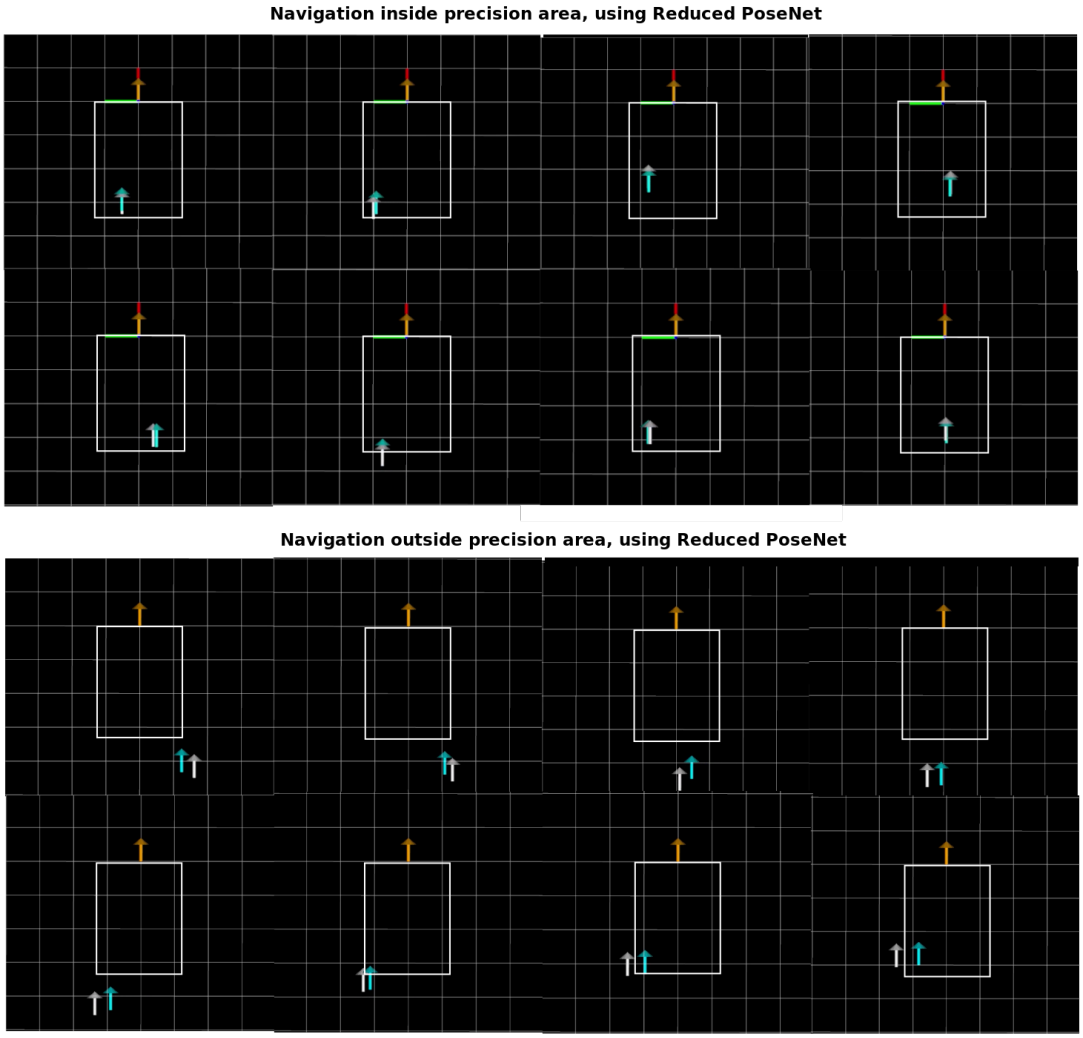

Figure 9: Predicted pose compared with groundtruth using RVIZ. First row shows Groundtruth (white) and Predicted (Sky) comparison results by Reduced PoseNet inside the precision area. Second row shows comparison results by Reduced PoseNet outside precision area.

[13] Hyungpil Moon, Jose Martinez-Carranza, Titus Cieslewski, Matthias Faessler, Davide Falanga, Alessandro Simovic, Davide Scaramuzza, Shuo Li, Michael Ozo, Christophe De Wagter, Guido de Croon, Sunyou Hwang, Sunggoo Jung, Hyunchul Shim, Haeryang Kim, Minhyuk Park, Tsz-Chiu Au, and Si Jung Kim. Challenges and implemented technologies used in autonomous drone racing. *Intelligent Service Robotics*, 12(2), Apr 2019.

[14] Elia Kaufmann, Mathias Gehrig, Philipp Foehn, René Ranftl, Alexey Dosovitskiy, Vladlen Koltun, and Davide Scaramuzza. Beauty and the beast: Optimal methods meet learning for drone racing. *CoRR*, abs/1810.06224, 2018.

[15] Elia Kaufmann, Antonio Loquercio, Rene Ranftl, Alexey Dosovitskiy, Vladlen Koltun, and Davide Scaramuzza. Deep drone racing: Learning agile flight in dynamic environments. *CoRR*, abs/1806.08548, 2018.

[16] Shuo Li, Michaël MOI Ozo, Christophe De Wagter, and Guido CHE de Croon. Autonomous drone race: A computationally efficient vision-based navigation and control strategy. *arXiv preprint arXiv:1809.05958*, 2018.

[17] Shuo Li, Erik van der Horst, Philipp Duernay, Christophe De Wagter, and Guido CHE de Croon. Visual model-predictive localization for computationally efficient autonomous racing of a 72-gram drone. *arXiv preprint arXiv:1905.10110*, 2019.

[18] S. Jung, S. Hwang, H. Shin, and D. H. Shim. Perception, guidance, and navigation for indoor autonomous

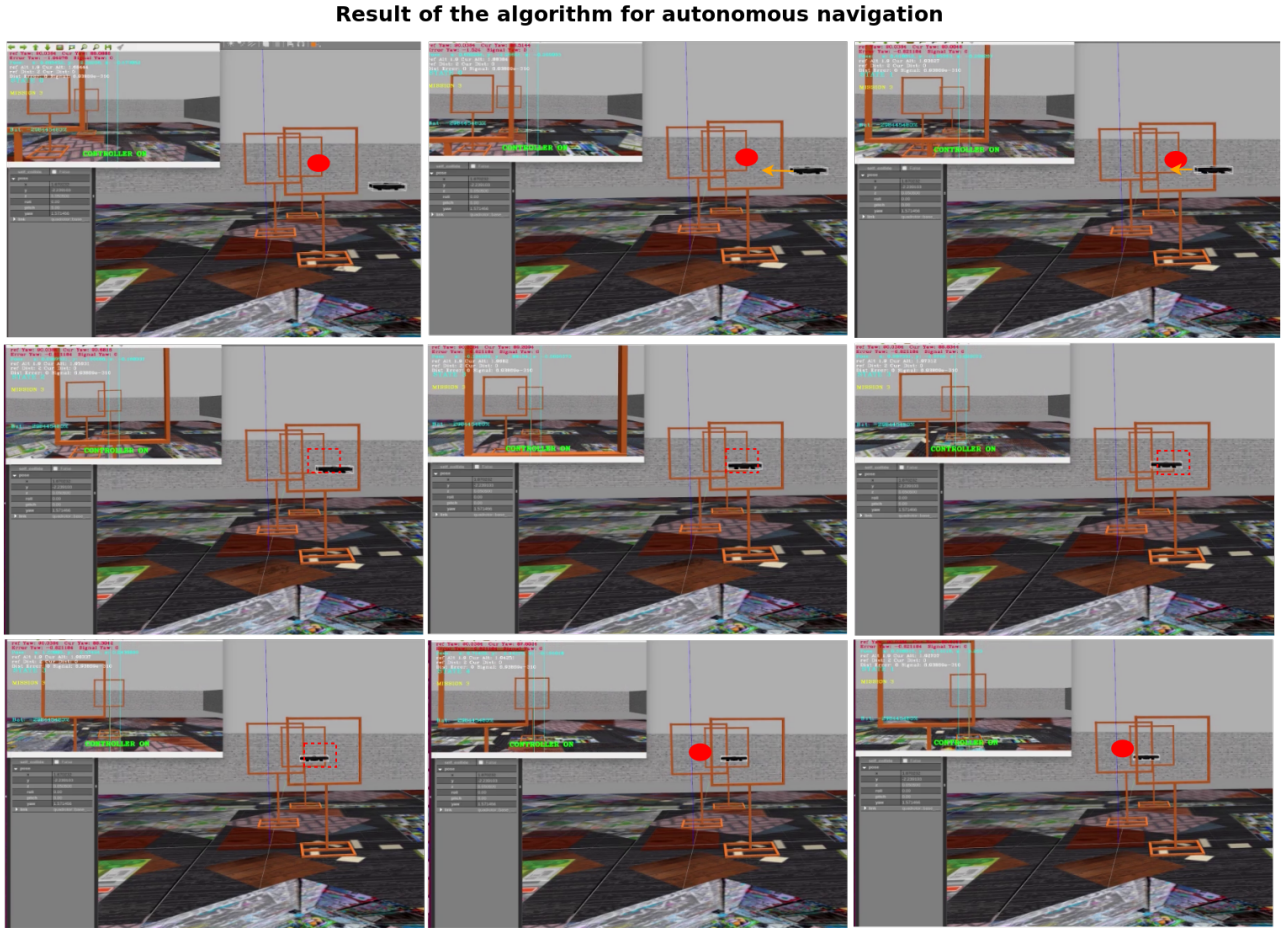

Figure 10: Autonomous navigation algorithm flying through one gate. In the first row, the algorithm centre the drone flying to the left, once the drone is centred, the drone flies to cross the gate (row two), this is marked with a dotted rectangle. Finally in row three when the drone complete cross the gate, the algorithms predict the pose of the drone and centres it again.

drone racing using deep learning. *IEEE Robotics and Automation Letters*, 3(3):2539–2544, July 2018.