# Drone-Based Object Tracking with Convolutional Neural Networks

**Henrique Antônio Guanais Corneau[1], Ramon Vinícius Ferreira de Souza[1], Murilo de Lima Barros[1], Vinícius Wallace Rebouças Silva[1], César Henrique Tittoto Melo[1], Helton Maia[1]**

*[1]Universidade Federal do Rio Grande do Norte, Natal, Brasil (helton.maia@ufrn.br)*

*Abstract*: A Python algorithm was implemented to control a drone's movement using a pre-trained Convolutional Neural Network (CNN) and the YOLO (You Only Look Once) framework. The algorithm utilizes a dataset of 2043 images to send movement commands based on the detected object's pixel area and screen position.

*Keywords*: Convolutional Neural Network; Drone; YOLO.

## INTRODUCTION

In recent years, UAVs (unmanned aerial vehicles), known as drones, have gained increasing traction in various fields, including research, agriculture, and construction. Their versatility in takeoff and landing and the ability to be remotely controlled make them suitable for missions in urban environments or hard-to-reach areas. (Otto et al., 2018).

In parallel, the increasing availability of computational resources, such as GPUs and cloud computing, has enabled the development of deep learning techniques, specifically artificial neural networks. In this context, CNNs (convolutional neural networks), a type of deep neural network, have become widely used for recognizing and extracting generic features in images and videos, solving problems such as segmentation, detection, and object tracking. (Saqib et al., 2018).

Currently, a challenging problem is a real-time object tracking by a drone in a dynamic environment (Chen et al., 2018). The most challenging part of object tracking is its detection in a continuous video stream captured by the drone's camera. The target's exact location is then used to navigate it, which can change as the object and the drone move (Barták and Vykovský, 2015). Thus, integrating convolutional neural networks and drones presents significant potential for solving this problem, enabling real-time processing directly on the device or in combination with cloud computing.

In this context, the primary aim of this study is to devise a straightforward solution for the object tracking issue using an affordable drone and a convolutional neural network to achieve real-time performance.

## MATERIAL AND METHODS

To develop the study, the necessary steps were outlined through a flowchart, as shown in Figure 1.

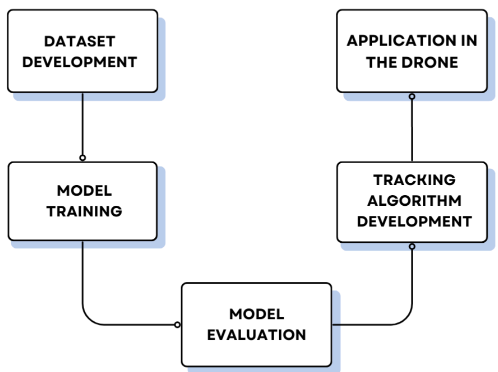


Figure 1. Flowchart of the developed steps.

At first, a dataset containing 2043 images was assembled, using the two types of landing targets adopted in the Competição Brasileira de Robótica (CBR) in the Flying Robots Trial League category as a basis. For this, the Roboflow tool was used, which allows users to preprocess the image set and manually annotate it to delimit the objects of interest.

It was decided to divide the dataset into three smaller sets: a training set, a validation set, and a test set, each containing 84%, 10%, and 6% of the original dataset images, respectively. The classes adopted were named "takeoff" and "movel", in reference to the base names in the CBR.

For the model training, a free tool called Google Colab was used, a cloud-based development environment that allows writing and executing

Python code directly in the browser. Using the Ultralytics library, it was possible to train the network using YOLO (You Only Look Once) version 8, a convolutional neural network architecture specialized in performing computer vision tasks. Its choice was due to YOLO being fast and efficient, making it suitable for applications where quick processing is crucial.

Upon completing the training of the model, the tracking algorithm was developed. The implemented logic relies on the data provided by YOLO, which identifies bounding boxes around the objects captured in an image. Each bounding box is characterized by three components: the box's coordinates, the confidence level, and the category to which the detected object belongs. By utilizing the x and y coordinates (in pixels) of the top-left and bottom-right points, we extract the coordinates of the detection center and calculate the occupied area.

By using the height and width of the window where the images are displayed, it is possible to calculate the center of the screen and implement a simple PID (proportional integral derivative) control, which is a widely used feedback control mechanism in industrial systems. Assuming that the drone would be aligned with the detected object if it were centered on the screen, an error measure is created. The proportional term of the PID indicates how far the center of the detection is from the middle of the window. This concept applies to both the x and y coordinates, as shown in equations 1 and 2.

$$errorX = cxDetect - centerX \quad (1)$$

$$errorY = centerY - cyDetect \quad (2)$$

The variables chosen to be adjusted by the PID, excluding the integral part, were the speeds in the vertical z-axis direction (speedZ) and the rotational speed around this axis (speedYaw). The error is calculated with each detection and saved to be used in the following speed calculation.

As seen in equations 3 and 4, derX and derY are calculated based on the previous error (prevErrorX and prevErrorY) and refer to the derivative term of the PID control. The rate of change of the error in the system defines their magnitude.

$$derX = (errorX - prevErrorX) \quad (3)$$

$$derY = (errorY - prevErrorY) \quad (4)$$

The parameters kp and kd, seen in equations 5 and 6, act in a corrective manner, helping to achieve a response value closer to the ideal.

$$speedYaw = kp * errorX + kd * derX \quad (5)$$

$$speedZ = kp * errorY + kd * derY \quad (6)$$

The choice of vertical speed as the adjustment variable is because the drone can move up or down, aiming to center the object along the y-axis. To center the drone along the x-axis, it was decided to rotate it around the z-axis. After the vertical and horizontal adjustments to place the object in the center of the screen, it is necessary to correct the distance between the drone and the object of interest. Therefore, a detection area at an ideal distance is estimated so that the approach toward the target occurs as the detection area increases on the screen.

The test vehicle was the Tello drone, a mini drone developed by Ryze Tech in partnership with DJI and Intel. Its choice was because it has an SDK (Software Development Kit) that allows the drone to be controlled through network commands sent from any device connected to the same Wi-Fi. Through this connection, it is also possible to obtain the video stream from the Tello.

Using Python API, the connection to the drone was established, and the video stream was obtained, capturing the frames received with the OpenCV library. Afterward, YOLO and the tracking algorithm were applied, controlling the drone in an indoor environment and adjusting the parameters of the PID.

## RESULTS AND DISCUSSION

After the dataset creation and network training stages, the generated results and the model's behavior were evaluated. In Figure 2, the obtained confusion matrix can be observed. It is noticeable that the result is considerably satisfactory, with few classification errors.

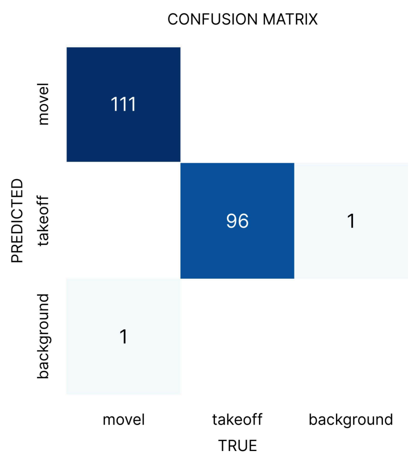


Figure 2. Confusion matrix of the training.

Just like in the confusion matrix, the values shown by the training metrics indicate an optimistic result. In Figure 3, for example, a final precision value above 97% can be observed, along with excellent values for all other metrics.

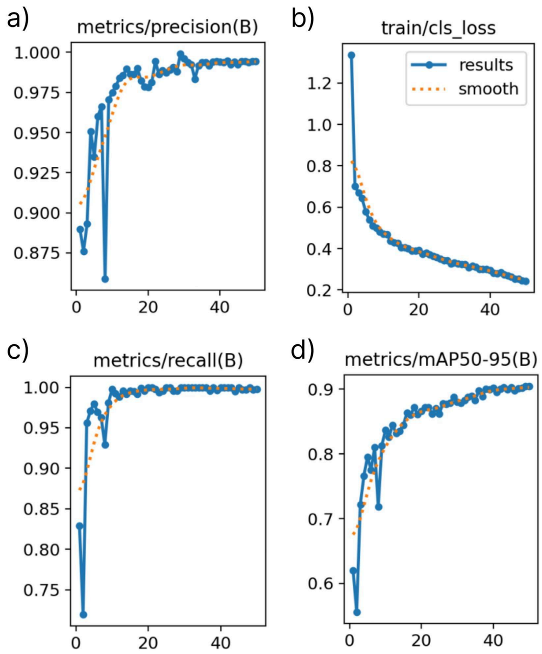


Figure 3. Training curves obtained.

To test the model, the detection of the two classes, "takeoff" and "movel," was performed in different environments with varying lighting conditions, angles, and perspectives. Figure 4 shows an example of the detection performed.

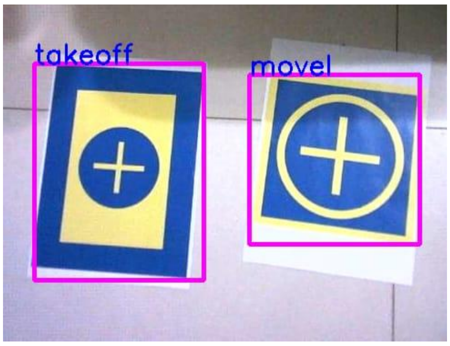


Figure 4. Detection of the bases by YOLOv8.

One of the failure points in detection is the inability to recognize objects at very long distances. In highly illuminated environments with other objects of similar patterns and colors, false positives appear, which occasionally can cause issues for drone control.

With the implemented tracking algorithm, it was noticed that the object's tilt significantly affects the control of distance to the target, which depends solely on the detection area. In cases where the base is perpendicular to the camera, progress occurs steadily and consistently until the limit is reached. However, at other angles, this progress is not as regular.

An additional factor that requires examination is the loss of object visibility. If there is a sudden change in the target's position on the screen, the drone may fail to track it and instead remain stationary in the air. This issue could be attributed to connectivity problems or image processing speed. It's important to acknowledge the constraints of both the camera and the motors.

Despite encountering these challenges, the overall implementation was deemed successful. Through adjustments to the parameters, the drone was able to effectively track and follow the target. As the project progresses, efforts will be made to address these shortcomings, leading to new potential applications.

## CONCLUSION

The study demonstrated that it is possible to develop a successful solution for the tracking problem using convolutional neural networks and drones. Although inaccuracies in detection pose challenges in resolving the problem in some scenarios, and the limitations of drone hardware hinder quick responses, there is still ample room for improvement. Adopting other control techniques, exploring different neural network architectures, and even developing new hardware are possibilities that will enable the continuous evolution of the work.

## ACKNOWLEDGEMENTS

We would like to thank the support of the Escola de Ciências e Tecnologia (ECT), the Departamento de Engenharia de Computação e Automação (DCA), and the Núcleo de Pesquisa e Inovação em Tecnologia da Informação (NPITI) at UFRN.

## REFERENCES

Otto, Alena et al. “Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey.” Networks, v. 72, n. 4, p. 411-458, 2018.

P. Chen, Y. Dang, R. Liang, W. Zhu and X. He, "Real-Time Object Tracking on a Drone With Multi-Inertial Sensing Data," in IEEE Transactions on Intelligent Transportation Systems, vol. 19, n. 1, p. 131-139, 2018.

R. Barták and A. Vykovský, "Any Object Tracking and Following by a Flying Drone," 2015 Fourteenth Mexican International Conference on Artificial Intelligence (MICAI), Cuernavaca, Mexico, p. 35-41, 2015.

Saqib, Muhammad et al. "A study on detecting drones using deep convolutional neural networks." 2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS). IEEE, 2017.