ISSN 2007-9737

A First CNN-based Approach towards Autonomous Flight for Object Lifting

Manuel Lopez Garcia[1], Jose Martinez Carranza[1,2]

[1] Instituto Nacional de Astrofisica, Optica y Electronica,
Mexico

[2] University of Bristol,
UK

{mlg.cc, carranza}@inaoep.mx

Abstract. Cable-suspended load transportation with Micro Air Vehicles (MAV) is a well-studied topic as it reduces mechanical complexity, the weight of the system, and energy consumption. However, it is always taken for granted that the load is already attached to cable. In this work, we present a methodology to autonomously lift a cable-suspended load with a MAV using a Deep-Learning based Object Detector as the perception system, whose detections are used by a PID controller and a state machine to perform the lifting procedure. We report an autonomous lifting success rate of 40%, an encouraging result considering that we carry out this task in a realistic environment, not in simulation. The Object Detector model has been tailored to detect the 2D position and 3D orientation of a bucket-shaped load and trained with a fully synthetic dataset. However, the model is successfully used in the real world. The control system deals with the oscillatory behavior of the cable and ground effects using low-level controllers. Future work includes improvements to the perception system to also detect a hook-shaped grasper.

Keywords. MAV, load lifting, deep learning.

1 Introduction

Micro Aerial Vehicles (MAV) have an increasing impact in areas such as agriculture, construction, mining, logistics, etc. Currently, a MAV can be controlled in basic perception and navigation tasks, however, it is still necessary to develop techniques for aerial manipulation tasks.

One of these tasks is for a MAV to be able to collect or lift objects autonomously. Several methods have been developed to lift objects which include equipping the MAV with an actuated arm or a magnetic grasper at the cost of higher power consumption or a larger size of the MAV platform. Another approach uses a cable-suspended grasper to lift a load, which has the advantage of reducing mechanical complexity, energy consumption, and the weight of the system, but it introduces a variety of challenges related to the swinging motion of the load.

There is an extensive amount of work related to the autonomous transportation and take-off of aerial vehicles with a cable suspended load, but it is established that the load is already attached to the cable. This means that autonomous cable-suspended load lifting is an open task. In the 2016 International Micro Air Vehicles, Conferences and Competitions (IMAV) competition, it was shown that it is possible for a human pilot to lift a load with a cable-suspended grasper while observing only the vehicle's camera images.

Inspired by that competition, we propose a method for autonomous lifting of a bucket-shaped load using a cable-suspended hook-shaped grasper with a MAV as shown in Fig. 1. The method consist of two systems, a customized Deep-Learning based Object Detector as the perception system, and a control system for

doi: 10.13053/CyS-24-3-3482

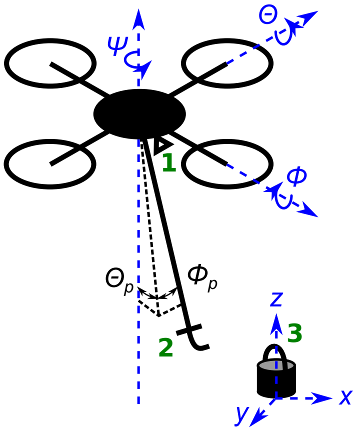

Fig. 1. Elements part of the lifting problem: (1) monocular camera, (2) cable-suspended hook-shaped grasper, (3) bucket-shaped load.

vehicle alignment and for generating the grasping and lifting trajectory.

The detector takes an aerial image captured by the onboard camera and outputs the load position and 3D orientation. To also estimate the orientation, the original convolutional box predictor of the detector is modified to include a quaternion output.

In order to generalize the orientation domain of the load, a fully synthetic dataset was used to train the object detector. The dataset is built with augmented data using 7 scales of the object and randomization of the image background, translation of the object in the image, the color of the object, and the brightness of the image.

Control system uses the position and orientation of the load to move the vehicle from the center of the image to the center of the detected object. Each time the center of the object is reached, the vehicle turns until the load handle is perpendicular to the vertical axis of the image. Then, a forward-up sequence is executed to grasp and lift the load.

To present our work, this paper has been organized as follows: section 2 describes our related work; section 3 describes our methodology; section 4 presents our results; and section 5 discusses our conclusions and future work.

2 Related Work

In the IMAV 2016 competition [8], a mission to pick and release a bucket-shaped load was presented; the same load shape will be used in this article. The problem of cable-suspended transportation with MAV has been widely studied using individual or cooperative vehicles. The dynamics of the forces for taking off [2] and trajectory control [3] with a MAV with a suspended load are analyzed. In [6] a model for the optimization of trajectories independent of the cable tension is presented. In [5] a neural network based controller trained by reinforcement learning to minimize the effects of swinging in cable-suspended load transportation is presented. However, in the aforementioned works it is assumed that the load is already attached to the cable end.

Among the methods for grasping and lifting a load that do not use a cable, in [19, 12] a method for lifting static and non-static ferrous discs with a magnetic gripper is presented. Their visual system detects the discs by their color, roundness and eccentricity, and calculates the 3D position of the object using the known size of the object and the camera parameters. In [26] a MAV uses a robot arm to grab a moving object while a motion capture system provides the poses for both the MAV and the object. An arm for grasping and a stereo camera used for pose estimation are used in [15]. In [21] a MAV uses a magnetic grasper to lift a cylindrical object using a monocular camera and a vision algorithm that recognizes the geometry of the cylinder.

For MAV vision systems using deep-learning based object detection, in [13] a Convolutional Neural Network (CNN), trained entirely with a synthetic dataset, is used for a MAV to pass through the gates of a race track autonomously. In [14] a CNN that allows a MAV to navigate the streets of a city safely by avoiding vehicles and pedestrians is presented. In [9] another CNN trained as a gate detector for a race track shows improvements compared to traditional image processing algorithms based on gate color and geometry. In [24] a CNN is designed upon state-of-the-art YOLO [16, 17] object detector trained to detect landing zone marks at a rate of 21Hz versus 7.5 and 5.3Hz for YOLO and YOLOV2, with similar precision. While those networks maintain a lean architecture for on board processing, they only provide a 2D position of the objects but not their orientation.

To detect the 3D orientation of objects, in [11] a CNN estimates the 6D pose for full scenes for the problem of camera relocalization using a quaternion regression for orientation. In [20, 23, 4, 10, 27] object detectors estimate the 2D projection of points of a 3D cuboid to solve the problem of camera localization to calculate 6D pose of the objects. Similarly, in [22] a 6D object detector is trained only with synthetic photorealistic images and random domain data to overcome the reality gap. Mostly, the architectures of these detectors allow an inference rate of 10-25Hz [27]; in [20] the proposed network manages to get the pose at a rate of 50Hz, but uses a heavy architecture that requires an NVIDIA Titan X graphics processor with 12GB of memory. Additionally, these networks require an Perspective-n-Point (PnP) algorithm to obtain the final pose of the camera using the estimated 2D points.

3 Methodology

The proposed methodology consist of two systems: perception and control. Perception is done through a customized CNN-based object detector that takes as input an RGB image from the MAV's camera and outputs the localization data of a bucket-shaped load. The data consists of the 2D position of the load in the image reference frame and its 3D orientation (roll, pitch, yaw), though, for our application, only yaw angle is used. With such data, the control system uses geometry to align the MAV itself with the load to execute a grasp and lift maneuver.

3.1 Dataset

For bucket detection, a dataset of 6384 images of 640x360 px was build with randomization [13, 25] of background, color, translation and brightness, at 7 different scales of the object (Fig. 2). We started with a set of 912 images from a rough 3D model of the object taken with a virtual camera using the Gazebo Simulator. The camera was held static at an altitude of 0.5m over the bucket and images were taken while changing the orientation of the bucket by 5° steps in a range of 180°, 80°, 80° for yaw, roll and pitch respectively. Then, color segmentation was used to separate the bucket from the background to create the other 6 scales of the object that range from 0.25 to 1.75. These scales help to train the network to recognize the object from different heights.

The background is randomly replaced by an image of the DTD dataset [1]; each channel of the object color in the HSV color space is randomly changed up to $\pm 50\%$ of its original value; the object is then randomly translated over the area of the image; the brightness of the full image is randomly changed up to $\pm 50\%$; finally, a JPG compression level of 0.3 is applied to blur the object with the background. Each image was labeled as: $((x_1, y_1, x_2, y_2), (q_w, q_x, q_y, q_z), SF)$ with the bounding box coordinates, a quaternion with the 3D orientation, and the scale factor SF applied to the object. Note that no data from the real object was used meaning that our dataset is fully synthetic.

3.2 Object Detector with Orientation

The detector is build around a Convolutional Neural Network with an output layer inspired by YOLO [16, 17], which uses MobileNetV2 [7, 18] as feature extractor. Taking PoseNet [11] as inspiration, the output layer is modified to also estimate a quaternion with the 3D orientation of the object. The detector takes a RGB image from as input and generates 7x7=49 estimates of the object. The detector is built with Tensorflow 1.15, using MobileNetV2 with an alpha=1.0 and an image input size of 224x224 pixels; images from the dataset are resized accordingly. The layer details of the model are shown in Fig. 3.

The convolutional box predictor output volume consist of three parts. The first part has a depth of 5 and each prediction contains the center coordinate (x_c, y_c) of the object, the width and height (h, w) of the object, and the estimation confidence C.

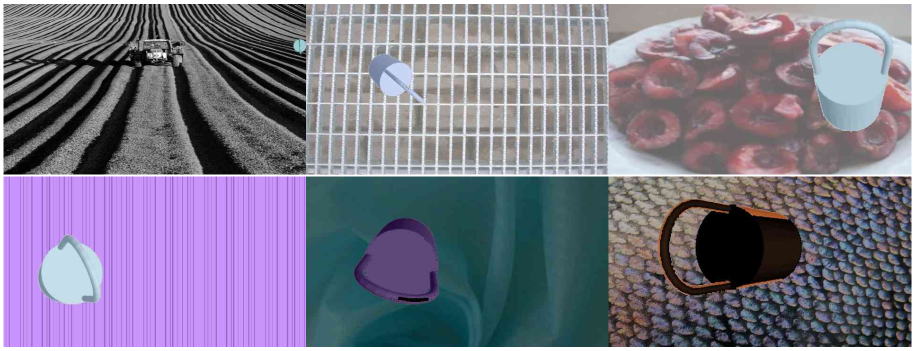

Fig. 2. Examples of images of the end dataset for the bucket-shaped load detection using background, color, translation and brightness randomization

The second part has a depth of 4 and each prediction contains the quaternion (q_w, q_x, q_y, q_z) of the object. The third part has a depth of 1, where each prediction contains the scale factor SF of the object. The detector outputs a $S \times S \times (5 + 4 + 1)$ tensor, where S is the grid size of the last convolutional layer before the output layer and, for MobileNetv2, $S = 7$. The coordinates (x_1, y_1, x_2, y_2) from a labeled bounding box are transformed into a normalized center coordinate (x_c, y_c), and into the normalized width and height of the bounding box (h, w) according to: $x_c = \frac{7}{w_{img}}\left(x_1 + \frac{x_2 - x_1}{2}\right)$, $y_c = \frac{7}{h_{img}}\left(y_1 + \frac{y_2 - y_1}{2}\right)$, $h = \frac{y_2 - y_1}{h_{img}}$, $w = \frac{x_2 - x_1}{w_{img}}$, where w_{img}, h_{img} is the size of the original image. If O is the $7 \times 7 \times 10$ tensor, then we put an object in one grid position and left the other grid cells empty:

$$
\begin{aligned}
&O_{\lfloor y_c \rfloor, \lfloor x_c \rfloor} = \\
&(h, w, y_c - \lfloor y_c \rfloor, x_c - \lfloor x_c \rfloor, 1, q_w, q_x, q_y, q_z, SF)\,.
\end{aligned} \tag{1}
$$

The three parts of the output volume are concatenated and the best estimate can be chosen from the prediction tensor $\hat{O}$ according to its highest probability confidence as follows: $y_f, x_f = \arg\max_{i,j} \hat{O}_{i,j,5}$; $h = \hat{O}_{y_f,x_f,1}$; $w = \hat{O}_{y_f,x_f,2}$; $y_c = y_f + \hat{O}_{y_f,x_f,3}$; $x_c = x_f + \hat{O}_{y_f,x_f,4}$; $q_w = \hat{O}_{y_f,x_f,6}$; $q_x = \hat{O}_{y_f,x_f,7}$; $q_y = \hat{O}_{y_f,x_f,8}$; $q_z = \hat{O}_{y_f,x_f,9}$; $SF = \hat{O}_{y_f,x_f,10}$.

For the loss function, squared error was used for the coordinates, quaternion and scale factor; binary cross entropy (BCE) was used for object confidence, as defined as: $BCE\,(g, p) = -\left(g \log(p) + (1 - g)\log(1 - p)\right)$. If O is the ground truth tensor, the loss function can be written as:

$$
\begin{aligned}
L\left(O, \hat{O}\right) = \sum_{b=1}^{B} \sum_{c=1}^{7} \sum_{d=1}^{7} BCE\left(O_{b,c,d,5}, \hat{O}_{b,c,d,5}\right) \\
+ \sum_{b=1}^{B} \sum_{i=1}^{4} \left(C_{b,i} - \hat{C}_{b,i}\right)^2 \\
+ \sum_{b=1}^{B} \sum_{i=1}^{4} \left(Q_{b,i} - \hat{Q}_{b,i}\right)^2 + \sum_{b=1}^{B} \left(S_b - \hat{S}_b\right)^2,
\end{aligned} \tag{2}
$$

where: B is the batch size; C, Q, S are the ground truth matrices for (y_c, x_c, h, w), quaternions and scale factors respectively; $\hat{C}$, $\hat{Q}$, $\hat{S}$ are the prediction matrices.

Training is performed in two stages taking the dataset of 6384 images divided in 5800 images for training and 584 for testing. The first stage takes MobileNetV2 pre-trained with ImageNet and only the customized output layers are allowed to train over 100 epochs.

The new trained weights are used as a start for the second stage, in which all the layers of MobileNetV2, and the output layers, are trained again over 100 epochs with no modifications to the dataset. The network is trained to maximize the Intersection Over Union (IOU) between the predicted objects and the ground truth as illustrated in Fig. 4b; the closer the IOU is to 1, the better the object detection boxes are.

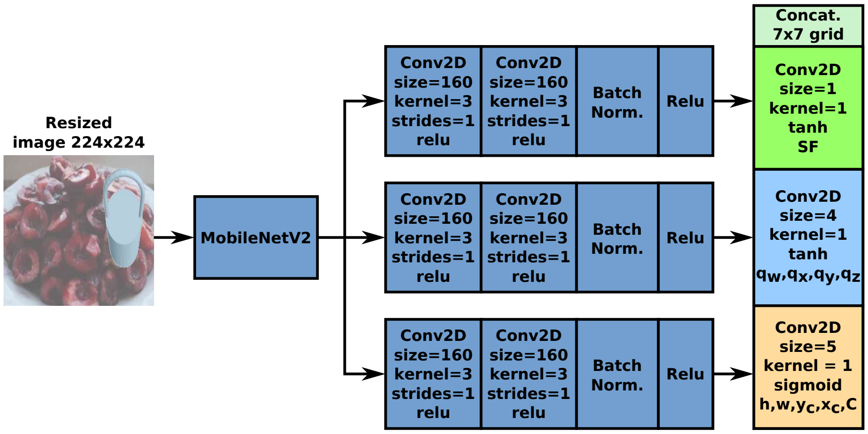
Resized image 224x224
MobileNetV2
Conv2D size=160 kernel=3 strides=1 relu
Conv2D size=160 kernel=3 strides=1 relu
Batch Norm.
Relu
Conv2D size=160 kernel=3 strides=1 relu
Conv2D size=160 kernel=3 strides=1 relu
Batch Norm.
Relu
Conv2D size=160 kernel=3 strides=1 relu
Conv2D size=160 kernel=3 strides=1 relu
Batch Norm.
Relu
Concat. 7x7 grid
Conv2D size=1 kernel=1 tanh SF
Conv2D size=4 kernel=1 tanh q_w,q_x,q_y,q_z
Conv2D size=5 kernel = 1 sigmoid h,w,y_c,x_c,C

Fig. 3. Object detector with orientation. The output volume consist of a grid of predictions of $7 \times 7 \times (h, w, y_c, x_c, q_w, q_x, q_y, q_z, SF)$

Table 1. Results for the two-stage training of the object detector

Metric	First Stage	Second Stage
Precision [IOU≥0.5]	0.9162	**0.9948**
Recall [IOU≥0.5]	0.8801	**0.9914**
Mean Error Q (deg)	22.0890± 16.17	**10.3374**± 17.77

A graph showing the loss and IOU gain is shown in Fig. 4a; a maximum IOU of 0.7020 and 0.9150 was achieved for the first and second training respectively, an increase of over 30%. The end model contains 3.7 million parameters.

Table 1 shows how the metrics improve once all the layers are allowed to train; in particular, recall shows that the sensitivity of the model is improved to avoid false negatives, that is, when the object is there but the model does not detect it.

Precision score shows that the model detects very few false positives, though, a decrease is expected when the model is used in the real world. A prediction is considered correct if the object is present, i.e. confidence≥0.5, and IOU≥0.5.

The quaternion error takes into account the full 3D orientation and is calculated as $\theta = \arccos \frac{Q \cdot \hat{Q}}{|Q||\hat{Q}|}$ where Q is the ground truth and $\hat{Q}$ is the predicted quaternion. A degree error of $10 \pm 18°$ is considered enough for the end application.

3.3 Single Object Detection Control

The control system uses the estimate with the highest confidence from the Object Detector to move the vehicle from the center of the image (x_{imgc}, y_{imgc}) to the center of the detected load (x_c, y_c) by simultaneously assigning values to u_x and u_y as shown in Fig. 5.

Altitude remains fixed to a specific reference by u_z. The diagonal line within the bounding box shows the yaw angle of the load obtained from the estimated quaternion. Each time the center of the load is reached, the vehicle yaw is controlled until a load angle reference is reached. Then, the control reduces the distance threshold to the center of the load. If distance is within such a threshold for at least 30/60 image frames, the control moves the vehicle in a forward-up sequence to grasp and lift the bucket.

The state machines for pose control and lifting control are shown in Fig. 6. For lateral and frontal lineal velocities u_x, u_y, two Proportional Integral (PI) controllers are used, one proportional controller is used for lineal velocity u_z for altitude

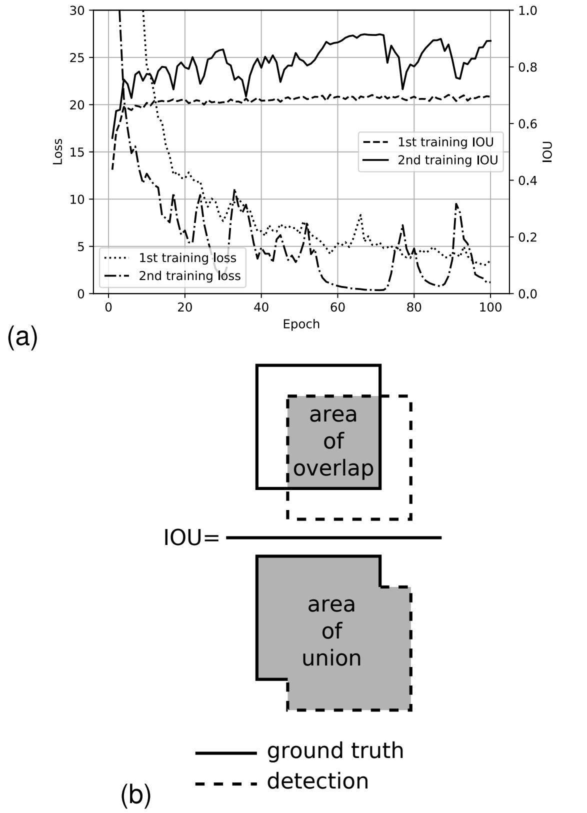

Fig. 4. (a) Loss and IOU for the two-stage training (b) Illustration of the IOU

displacement, and one PI controller is used for the angular velocity w_z for yaw rotation.

4 Experiments and Results

The proposed system was implemented in ROS Melodic Morenia, consist of four nodes (Fig. 7) that are executed in a laptop with a i5-9300H@2.4GHz CPU, 8G RAM, and a NVIDIA GTX 1660ti GPU. The MAV is a Bebop 2 from Parrot. The control node executes the state machines for pose and lifting control every time a new image from the Bebop's camera is published at a rate of 30Hz; it publishes a twist message with the velocities for the Bebop.

The CNN node executes the object detector and publishes the bucket detection prediction with the best confidence. The keyboard node allows a human pilot to take control of the vehicle or switch to autonomous control.

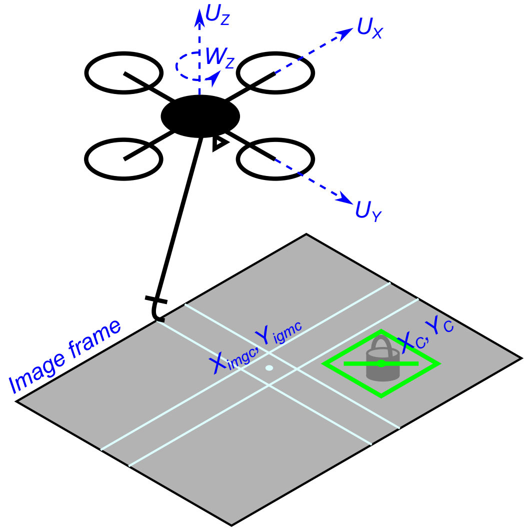

Fig. 5. Variables used for single object detection control

The ardrone_autonomy node publishes the images from the downward-facing Bebop's camera and subscribes to the commands from the control and keyboard nodes. The cable used is 55cm long and is attached to the back of the Bebop at one end and to a wire hook-shaped grasper at the other end.

Ten flights were carried out at an altitude of 70cm in different natural lighting conditions achieving a 40% success rate. A lift is considered successful if the vehicle lifts the load more than 30cm from the ground within a 3 minute window. Table 2 summarizes the results obtained. Mean detection accuracy is calculated over 200 images taken in each flight while autonomous control is ON, where true positive detections have a confidence greater than 0.5.

An example sequence is shown in Fig. 8:

— 1) a human pilot controls the vehicle until the load enters the camera's field of view;

— 2) the autonomous controller is ON and the vehicle start to descend to the specified altitude and towards the load;

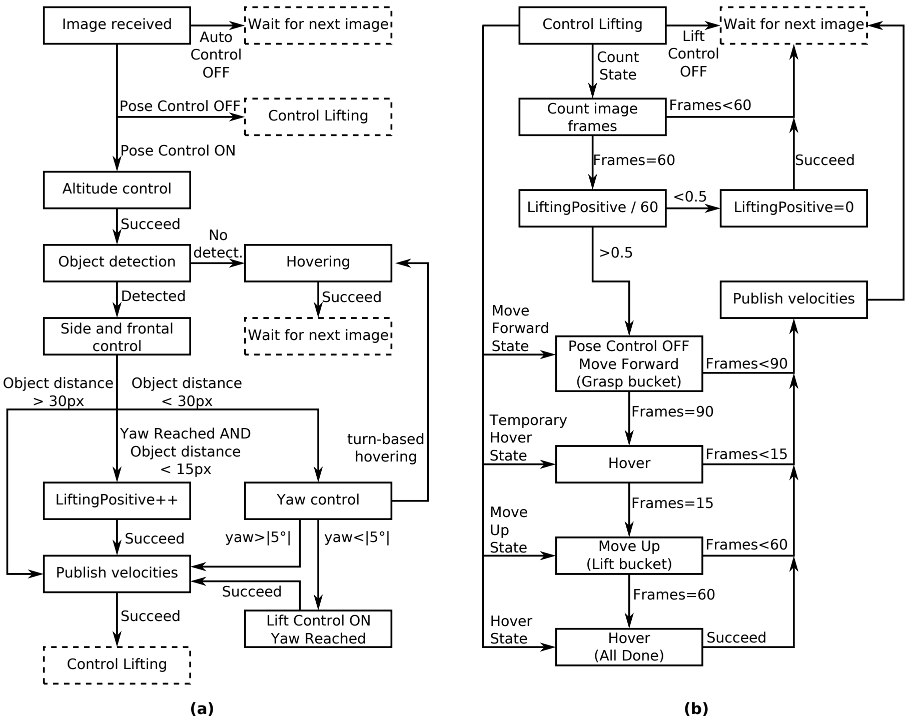

Fig. 6. Single object detection control. (a) Pose control. (b) Grasping and lifting control

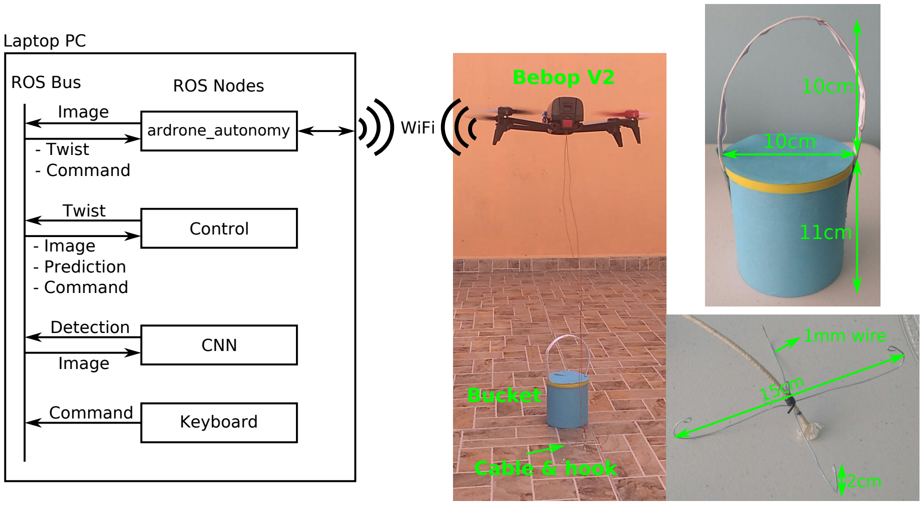

Fig. 7. Framework used to evaluate the single object detection control strategy

— 3,4) the vehicle rotates to reach grasp alignment;

— 5) when the distance to the object is maintained at least 30/60 frames, the control moves the vehicle to grasp and lift the load;

— 6) a message is shown when the lifting state is

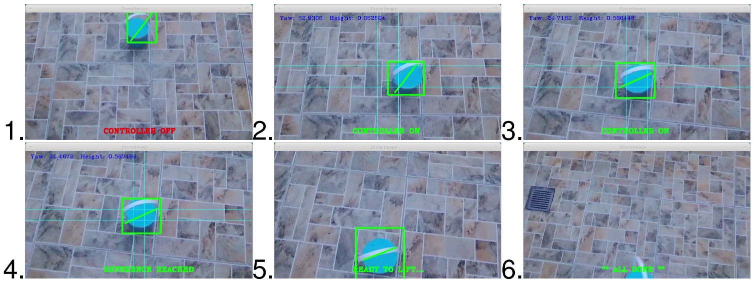

Fig. 8. Example sequence followed by the single object detection control strategy

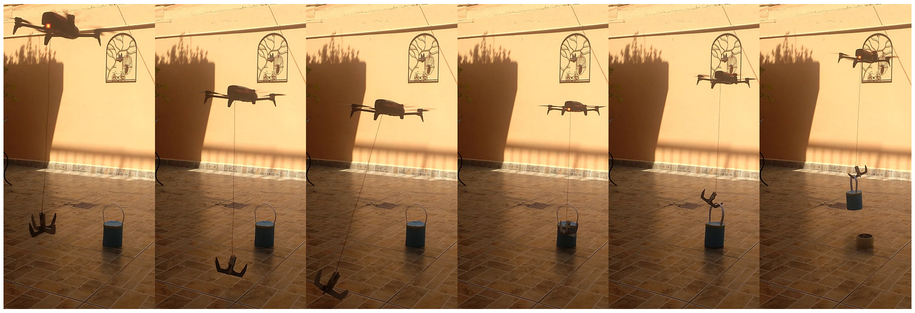

Fig. 9. Example of a lifting sequence with a plastic hook for better appreciation

completed and the human pilot takes control to land the vehicle. A lifting sequence with a plastic hook is shown in Fig. 9 and the full video can be viewed at this link[1].

Table 2. Results obtained over 10 flights for evaluating the lifting strategy

Metric	
Mean detection accuracy	0.93 ± 0.084
Object detector operating frequency	79.36 ± 7.56 Hz
Mean lifting time	55.7 ± 19 s
Lifting success rate	40%

Several sources of inaccuracy were observed that help explain a success rate of 40%: 1) as the vehicle moves forward in the grasping state, the air effect created by the bucket increases the oscillation of the grasper which in turn increases the grasping error; 2) in some cases, the grasper oscillation generates instability in the vehicle which changes a straight forward grasping trajectory to a diagonal forward trajectory; 3) there are offsets between the center of the image and the physical location of the camera inside the Bebop which add to the inaccuracy at the grasping state.

5 Conclusion and Future Work

A method designed to autonomously lift a load with a suspended cable using a Micro Aerial Vehicle has been presented. To address the problem, we used the perception capabilities of a deep learning based object detector to observe the object and use its detected position on the image to design a PID controller and a state machine to perform the lifting procedure. This perception along with a single object detection control system allowed us to achieve a 40% lifting success rate. This is an encouraging result since we perform our tests in a real environment, not in simulation. We are convinced that we could increase the rate success by fine tuning of the controller. In this sense, the control system uses only the position

[1] https://youtu.be/rvh3BWcd1Pg

and orientation data of the load to control the lifting routine.

The object detector was tailored to also estimate the 3D orientation of the object by increasing the depth of the original convolutional box predictor, to include a quaternion output. More over, the detector was fully trained with synthetic data and successfully used in a real environment.

Still, the method is sensitive to inaccuracies in flight stability and the randomness of grasper movement. Future work will focus on developing a more robust control strategy that includes both load and hook detection.

References

1. **Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., & Vedaldi, A. (2013).** Describing Textures in the Wild. *arXiv:1311.3618 [cs]*. ArXiv: 1311.3618.
2. **Cruz, P. J. & Fierro, R. (2017).** Cable-suspended load lifting by a quadrotor UAV: hybrid model, trajectory generation, and control. *Autonomous Robots*, Vol. 41, No. 8, pp. 1629–1643.
3. **de Angelis, E. L., Giulietti, F., & Pipeleers, G. (2019).** Two-time-scale control of a multirotor aircraft for suspended load transportation. *Aerospace Science and Technology*, Vol. 84, pp. 193–203.
4. **Do, T.-T., Cai, M., Pham, T., & Reid, I. (2018).** Deep-6DPose: Recovering 6D Object Pose from a Single RGB Image. *arXiv:1802.10367 [cs]*. ArXiv: 1802.10367.
5. **Faust, A., Palunko, I., Cruz, P., Fierro, R., & Tapia, L. (2017).** Automated aerial suspended cargo delivery through reinforcement learning. *Artificial Intelligence*, Vol. 247, pp. 381–398.
6. **Foehn, P., Falanga, D., Kuppuswamy, N., Tedrake, R., & Scaramuzza, D. (2017).** Fast Trajectory Optimization for Agile Quadrotor Maneuvers with a Cable-Suspended Payload. *Robotics: Science and Systems XIII*, Robotics: Science and Systems Foundation.
7. **Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017).** MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv:1704.04861 [cs]*. ArXiv: 1704.04861.
8. **IMAV (2016).** IMAV 2016 International Micro Air Vehicle Competition Rules v2.3, pp. 19–22.
9. **Jung, S., Hwang, S., Shin, H., & Shim, D. H. (2018).** Perception, Guidance, and Navigation for Indoor Autonomous Drone Racing Using Deep Learning. *IEEE Robotics and Automation Letters*, Vol. 3, No. 3, pp. 2539–2544.
10. **Kehl, W., Manhardt, F., Tombari, F., Ilic, S., & Navab, N. (2017).** SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again. *arXiv:1711.10006 [cs]*. ArXiv: 1711.10006.
11. **Kendall, A., Grimes, M., & Cipolla, R. (2016).** PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. *arXiv:1505.07427 [cs]*. ArXiv: 1505.07427.
12. **Loianno, G., Spurny, V., Thomas, J., Baca, T., Thakur, D., Hert, D., Penicka, R., Krajnik, T., Zhou, A., Cho, A., Saska, M., & Kumar, V. (2018).** Localization, Grasping, and Transportation of Magnetic Objects by a Team of MAVs in Challenging Desert-Like Environments. *IEEE Robotics and Automation Letters*, Vol. 3, No. 3, pp. 1576–1583.
13. **Loquercio, A., Kaufmann, E., Ranftl, R., Dosovitskiy, A., Koltun, V., & Scaramuzza, D. (2020).** Deep Drone Racing: From Simulation to Reality With Domain Randomization. *IEEE Transactions on Robotics*, Vol. 36, No. 1, pp. 1–14.
14. **Loquercio, A., Maqueda, A. I., del Blanco, C. R., & Scaramuzza, D. (2018).** DroNet: Learning to Fly by Driving. *IEEE Robotics and Automation Letters*, Vol. 3, No. 2, pp. 1088–1095.
15. **Ramon Soria, P., Arrue, B., & Ollero, A. (2017).** Detection, Location and Grasping Objects Using a Stereo Sensor on UAV in Outdoor Environments. *Sensors*, Vol. 17, No. 12, pp. 103.
16. **Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016).** You Only Look Once: Unified, Real-Time Object Detection. *arXiv:1506.02640 [cs]*. ArXiv: 1506.02640.
17. **Redmon, J. & Farhadi, A. (2016).** YOLO9000: Better, Faster, Stronger. *arXiv:1612.08242 [cs]*. ArXiv: 1612.08242.
18. **Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2019).** MobileNetV2: Inverted Residuals and Linear Bottlenecks. *arXiv:1801.04381 [cs]*. ArXiv: 1801.04381.
19. **Spurny, V., Baca, T., Saska, M., Penicka, R., Krajnik, T., Thomas, J., Thakur, D., Loianno, G., & Kumar, V. (2019).** Cooperative autonomous search, grasping, and delivering in a treasure hunt scenario by a team of unmanned aerial vehicles. *Journal of Field Robotics*, Vol. 36, No. 1, pp. 125–148.

20. **Tekin, B., Sinha, S. N., & Fua, P. (2018).** Real-Time Seamless Single Shot 6D Object Pose Prediction. *arXiv:1711.08848 [cs]*. ArXiv: 1711.08848.

21. **Thomas, J., Loianno, G., Polin, J., Sreenath, K., & Kumar, V. (2014).** Toward autonomous avian-inspired grasping for micro aerial vehicles. *Bioinspiration & Biomimetics*, Vol. 9, No. 2, pp. 025010.

22. **Tremblay, J., To, T., Sundaralingam, B., Xiang, Y., Fox, D., & Birchfield, S. (2018).** Deep Object Pose Estimation for Semantic Robotic Grasping of Household Objects. *arXiv:1809.10790 [cs]*. ArXiv: 1809.10790.

23. **Xiang, Y., Schmidt, T., Narayanan, V., & Fox, D. (2018).** PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. *arXiv:1711.00199 [cs]*. ArXiv: 1711.00199.

24. **Yu, L., Luo, C., Yu, X., Jiang, X., Yang, E., Luo, C., & Ren, P. (2018).** Deep learning for vision-based micro aerial vehicle autonomous landing. *International Journal of Micro Air Vehicles*, Vol. 10, No. 2, pp. 171–185.

25. **Zendel, O., Murschitz, M., Humenberger, M., & Herzner, W. (2017).** How Good Is My Test Data? Introducing Safety Analysis for Computer Vision. *International Journal of Computer Vision*, Vol. 125, No. 1-3, pp. 95–109.

26. **Zhang, G., He, Y., Dai, B., Gu, F., Yang, L., Han, J., Liu, G., & Qi, J. (2018).** Grasp a Moving Target from the Air: System & Control of an Aerial Manipulator. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Brisbane, QLD, pp. 1681–1687.

27. **Zhang, X., Jiang, Z., & Zhang, H. (2019).** Real-time 6D pose estimation from a single RGB image. *Image and Vision Computing*, Vol. 89, pp. 1–11.

Article received on 14/06/2020; accepted on 21/07/2020. Corresponding author is Manuel Lopez-Garcia.