Onboard CNN-Based Processing for Target Detection and Autonomous Landing for MAVs

A. A. Cabrera-Ponce[1] and J. Martinez-Carranza[1,2](✉)

[1] Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Puebla, Mexico
{aldrichcabrera,carranza}@inaoep.mx
[2] University of Bristol, Bristol, UK

Abstract. In this work, we address the problem of target detection involved in an autonomous landing task for a Micro Aerial Vehicle (MAV). The challenge is to detect a flag located somewhere in the environment. The flag is posed on a pole, and to its right, a landing platform is located. Thus, the MAV has to detect the flag, fly towards it and once it is close enough, locate the landing platform nearby, aiming at centring over it to perform landing; all of this has to be carried out autonomously. In this context, the main problem is the detection of both the flag and the landing platform, whose shapes are known in advanced. Traditional computer vision algorithms could be used; however, the main challenges in this task are the changes in illumination, rotation and scale, and the fact that the flight controller uses the detection to perform the autonomous flight; hence the detection has to be stable and continuous on every camera frame. Motivated by this, we propose to use a Convolutional Neural Network optimised to be run on a small computer with limited computer processing budget. The MAV carries this computer, and it is used to process everything on board. To validate our system, we tested with rotated images, changes in scale and the presence of low illumination. Our method is compared against two conventional computer vision methods, namely, template and feature matching. In addition, we tested our system performance in a wide corridor, executing everything on board the MAV. We achieved a successful detection of the flag with a confidence metric of 0.9386 and 0.9826 for the Landing platform. In total, all the onboard computations ran at an average of 13.01 fps.

Keywords: CNN · SSD · Target detection · Autonomous landing

1 Introduction

Nowadays, target detection is a traditional problem in computer vision, which involves having to identify features describing relevant information about an object or set of objects. In robotics, target detection is a problem for robots that perform some tasks in real scenarios, mostly due to poor illumination.

K. M. Figueroa Mora et al. (Eds.): MCPR 2020, LNCS 12088, pp. 195–208, 2020.
https://doi.org/10.1007/978-3-030-49076-8_19

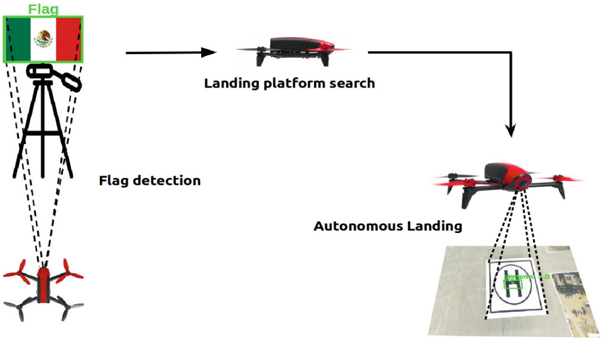


Fig. 1. Target detection for autonomous landing based on a mission of indoors competition in IMAV2019. A video of this work for reviewing purpose is found at https://youtu.be/sYn9mo-2hvA

Although the use of sensors can facilitate target detection, the information can be different in indoor and outdoor environments, thus varying the information around of the target.

Micro Aerial Vehicles (MAVs) have become popular in the research community for easy control and manipulation using the GPS devices and RGB cameras for solving multiple problems like inspection, detection, surveillance, rescue and localisation in indoors and outdoors environments. These tasks have been carried out with vision methods such as optical flow, segmentation, edge detector, morphological operations, feature extractor, feature matching and template matching. Besides, some methods have been combined with two or more techniques for suitable detection, while a MAV performs an autonomous flight in an unknown environment. Also, the combination of different types of cameras such as depth cameras, thermal cameras and stereo cameras enable to capture other types of information useful for detection. Nevertheless, the use of this information can be computationally expensive to perform detection onboard of the MAV in real-time, affect the speed performance. Likewise, it can be affected much for changes of illumination and environments, including oblique views, scale and rotations even that the object is partially occluded.

From the above, several events around the world have proposed competitions of robotics focused on the use of MAVs to solve tasks in real-time. The International Micro Aerial Vehicles and competition (IMAV) is an event focused on aerial robotics, including conference and competition in outdoors and indoors environments. The event consists of the development of new systems and methods to solve problems such as detection, control, pose estimation and autonomous navigation.

Deep learning has become a useful tool for classification, segmentation and detection without having to explicitly design a detector, descriptor and matcher components, typical of traditional computer vision techniques. Convolutional Neural Networks (CNNs) have been used to obtain results by training a dataset, allowing the learning of features to recognise multiple objects in one single pass without importance the views, occlusion and changes of illumination. YOLO, FRCNs and Single Shot Detector (SSD) are CNNs to detect classes of objects in an image, learning their features without using much computationally cost.

Therefore, motivated by the effectiveness of deep learning for the detecting task, in this work, we present a detection system to solve one of the missions included in the indoors competition of the IMAV2019. This mission consisted in detect a given flag, which is used to indicate the position of a landing platform. The goal is to a MAV navigate autonomously detecting the flag to fly towards its location, and then identify the landing platform. Once the landing platform is detected, the MAV has to maintain the detection, while performing autonomous flight to centre its position w.r.t. the platform, seeking to secure the landing on the platform in an autonomous manner, see Fig. 1.

Our detection system is based on Single Shot Detector architecture with seven convolutional layers (SSD7). We have manually generated a training dataset of the flag and platform in several views, environments and changes of illumination to obtain an improved result before realising the autonomous landing. The SSD network was chosen due to its fast performance on micro computer boards with low budget processing powers and without GPU. In average, we have tested and observed that detection tasks can be performed with an average processing speed of 15 fps; this includes the controller responsible for the autonomous flight and landing routines.

In order to present our work, this paper is organised as follows. Section 2 provides related works about object detection and autonomous landing using deep learning and vision methods. Section 3 describes the dataset generation, the hardware used for the training and experiments, and our approach for detection. Section 4 shows the experimental design and the comparison of our approach with other methods for the flag and platform detection. In Sect. 5, we present the results running on board the MAV. Finally, conclusions and future work are outlined in Sect. 6.

2 Related Work

Object detection is a problem that has addressed for a long time in image processing, pattern recognition, and robotics using multiples techniques of recognition. In aerial robotics, recent works have sought out new techniques for target detection using sensors or vision during autonomous flight. However, due to onboard cameras of the MAVs, vision methods have used to perform tasks of detection, search and tracking with visual descriptors being the most widely used due to its fast application. For instance, in [6] detect regions of interest to the runway of wind-fixes UAVs applying sparse coding spatial pyramid matching (ScSPM), others create a keypoints database for feature matching [17] or the improvement of a descriptor using CamShift based on colour information [24]. Others prefer the use patterns or marks to detect a landing platform [2,3] and template-based matching in an image pyramid scheme for the target detection in multiple scales [5]. Likewise, methods based on RANSAC allow the search and detection of landing sites with multi-scale features using 3D maps for pose estimation of landing sites [21,22].

For one hand, machine learning and Artificial Neural Networks have leveraged the learning to detect and recognise landing targets using several methods in combination. For instance, the use of nearest neighbour with CNN layers to have effective in recognition [7] and category maps using counter propagation networks (CPNs) to identify multiple objects from aerial images [8]. Also, they are suitable for learning the skill of pilots through generated models from datasets [1], even to cooperative detections and tracking onboard [13]. Likewise, deep reinforcement learning can identify the position of the land the UAV on uniform textures using a Deep Q-Networks (DQNs) for vertical descent on a variety of simulated and real-world environments [10] or in several simulated environments with relevant noise [11]. Some works employ different deep reinforcement learning methods for the autonomous landing. Thus, [19] they show an improved deep reinforcement learning (DRL) trained on Gazebo simulation for the autonomous landing. In [12], use Deep Q-Networks (DQNs) to perform autonomous landing on the deck of a USV subject to perturbations induced by sea, and [15] use a Gazebo-based reinforcement learning framework for UAV landing on a moving platform.

On the other hand, the target detection onboard of the MAVs using deep learning has promising results, such as YOLO, FRCN and SSD. The training of CNN models is an alternative for target detection, estimating heading angles to guide the aircraft to runway landing [4] or to obtain high-level commands directly to MAV respect to target [20]. Furthermore, some CNN allows detecting broad zones for autonomous landing using depth estimation networks in real environments from a simulate dataset [16]. However, it is necessary to take into account that some sites are not wides and a precise landing is required, providing a bounding box of the landing target [14]. Hence, the detection of the targets is one of the main tasks in aerial robotics before to do an autonomous landing, in [23] uses YOLO and SqueezeNet to detect marks on the landing zone in synthesised and real-world scenarios. Finally, another work performs deep learning-based reconstruction and marker detection for MAV landing with YOLOv2 [18], and [9] uses lightDenseYOLO in combination with Kalman Filter for detecting markers and estimating the direction to perform the autonomous landing.

Despite detect targets and landing zones with deep learning, these works perform an onboard detection using computers with GPU architecture like Nvidia TX1, Nvidia TX2 and Snapdragon. Therefore, in this paper, we present a detection system using an SSD network for target detection and autonomous landing onboard of a MAV without a computer with GPU architecture.

3 Methodology

Our detection system is based on SSD architecture with seven convolution layers (SSD7). This CNN is an optimised network build to be used in computers with low performs or without GPU architecture, including on micro computer boards. The SSD is a CNN for detect multiples objects through predictions of bounding boxes around them with the capability to learn up to twenty classes of an only image and being faster to train than Yolo, FRCN and tinyYolo. In this paper, we have been trained the SSD7 with two classes: Mexican flag and Landing platform, using images captured with the Drone Bebop 2 through ROS (Robot Operating System) establishing communication between the MAV and the computer. The images were resized to QVGA (320 × 240) resolution to accelerate the training of the network and manually labelled selecting the bounding box around of the interest object. The configuration in Fig. 2 has used to detect the flag, and the platform to then send control commands in Roll, Pitch and Yaw to the aerial vehicle.

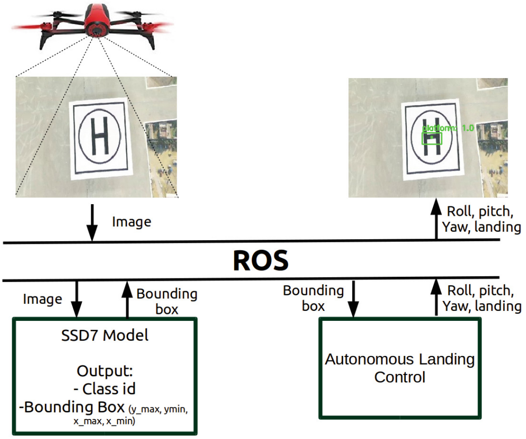
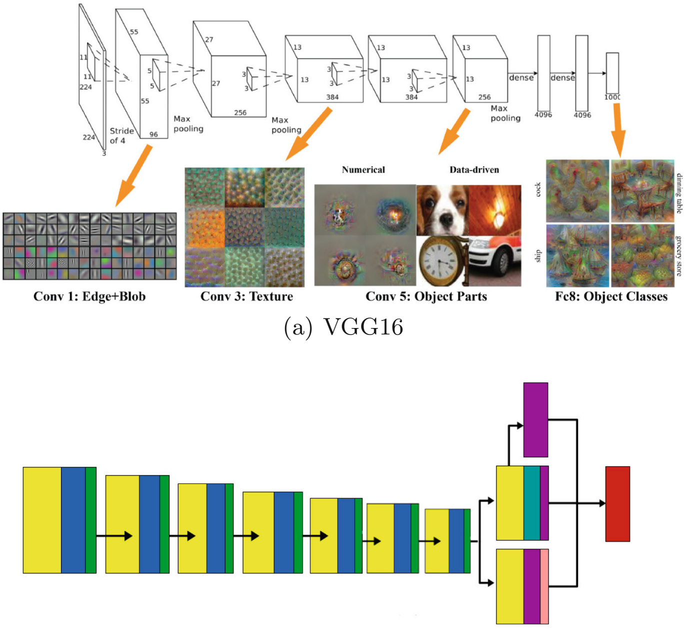


Fig. 2. System of communication used to send control commands in Roll, Pitch and Yaw to the MAV.

3.1 Single Shot Detector (SSD)

The SSD is a detection network composes of 2 parts: extract feature maps, and apply convolution filters to make predictions and detect objects, using a VGG16 network as a feature extractor (Fig. 3). Each convolution filter makes a prediction composes of bounding boxes and scores for each class. In contrast to other detection techniques, learn main features such as the form, colour, aspect, scale, saturation and texture regardless of illumination changes, partial occlusion and changes in the environment that may impair the appearance of the object. Therefore, in this paper, we use SSD7 architecture (Fig. 3) to perform the object detection onboard of the MAV in the Intel Stick Computer without GPU architecture. The input of the network is an RGB image with QVGA (320 × 240) resolution passing for filters in each convolution layer to producing bidimensional maps that generate bounding boxes around of the object.



(a) VGG16

(b) Single Shot Detector (SSD7).

Fig. 3. Network architectures. (Color figure online)

To cover more forms of bounding boxes, the SSD uses Multi-scale features maps and data augmentation to improve the accuracy, flipping, cropping and distorting the colour of the image to handle variants in various object sizes and shapes. Our SSD architecture makes 6340 predictions for better coverage of location, scale and aspect ratios, more than many other detection methods. Besides, the predictions are classified as the intersection over the union and are a measure of the ratio between the intersected area over the joined area for two regions. This strategy makes that each prediction have shapes closer to the corresponding ground truth (Fig. 4), where its value is of 0.0 to 1.0, being the value 1.0 the proper detection.

In the last layer, is apply Non-maximum Suppression (NMS) to clear the unnecessary bounding boxes and remove duplicate predictions to the same object. On this way, we keep 200 predictions per image and drawing the bounding box whose confidence value is above 0.8. In Fig. 5, we show an example of the bounding boxes predicted and the final result applying the threshold. Finally, in the output of the network, we obtain a vector whose information have the bounding box coordinates (x_min, x_max, y_min, y_max), class_id and a confidence metric where the object is localised in the image.

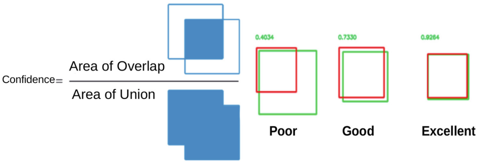


Fig. 4. Confidence metric to object detection. (Color figure online)

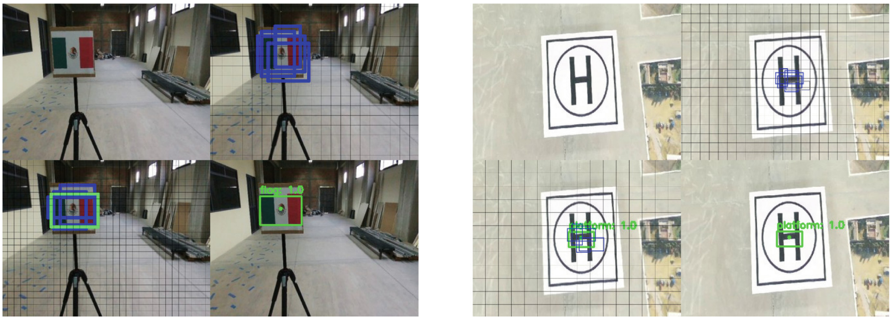


Fig. 5. Top left: original image; Top right: bounding boxes obtain with SDD; Bottom left: bounding box selected applying the confidence threshold; Bottom right: final result. (Color figure online)

3.2 System Overview

Our system has tested with two different computers Fig. 6. The first computer was used to train the SSD network and to validate our system offboard the MAV, whose specifications are: Lenovo Y700 with 16 GB of RAM, Nvidia GTX 960M with 640 CUDA cores, with CUDA 9.0, Keras 2.2.4 and TensorFlow 1.12.0. The second computer was used to validate our system onboard the MAV, whose specifications are: Intel Computer Stick with a processor core M3-Y30 2.20 Ghz, 4 Mb, 64 GB, 4 GB DDR3 without GPU, with Keras 2.1.4 and TensorFlow nightly (optimised version to computers without GPU architecture).

(a) Lenovo Y700 (b) Intel Stick M3

Fig. 6. Computers used by our detection system.

3.3 Dataset Generation

The dataset was generated inside of our laboratory using the Bebop 2 drone and ROS to obtain images of the Mexican flag and landing platform. The dataset consists of 9000 images to the "Mexican flag" class and 5000 to "Landing platform" class in multiple views, rotations, scales and changes of illumination. We labelled whole the images manually using the *LabelImg* tool selecting with a bounding box in the image the object that we require to identify. It is important to carefully label the bounding boxes since the predictions start based on those. In Fig. 7, we show an example of the images captured to train the SSD whose training parameters are: the Batch size of 16, Adam Optimiser and 100 epochs with 1000 steps of training.

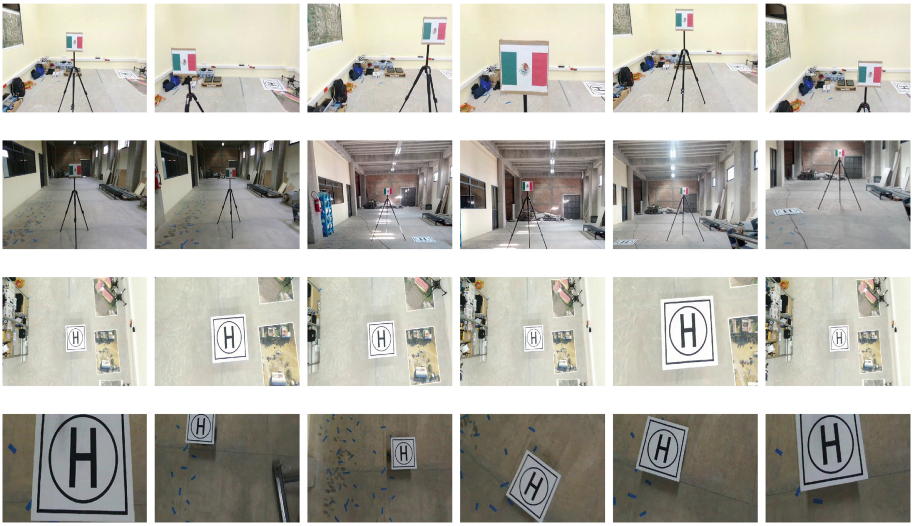


Fig. 7. Training dataset generated for the "Mexican flag" and "Landing platform".

4 Experiments

The carried out experiments focus on the two target detection in different rotations, scales and change of illumination, using our system, and the comparison with Template matching and Feature matching. The tests have performed in a wide corridor with low illumination Fig. 8, adapting to an indoor environment like the one presented in the competition. In addition, we use the same search template with respect to the bounding box labelled for detection Fig. 9, validating the effectiveness of the features extracted and learned by our network and the features of the methods of comparison.

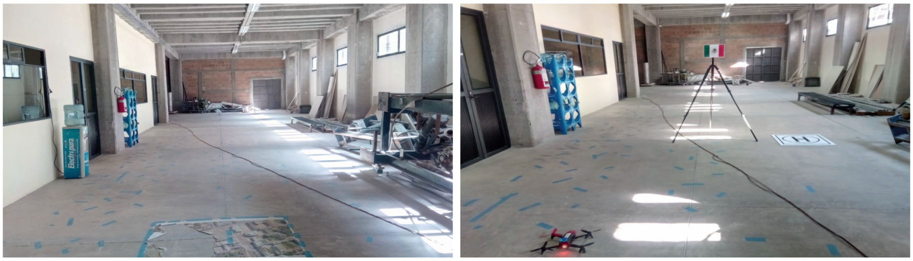


Fig. 8. Wide corridor where we perform the experiments.

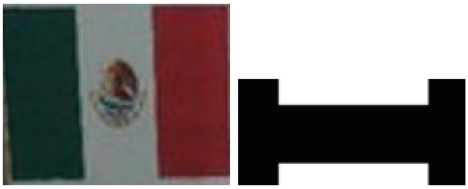


Fig. 9. Search templates used for template matching and feature matching.

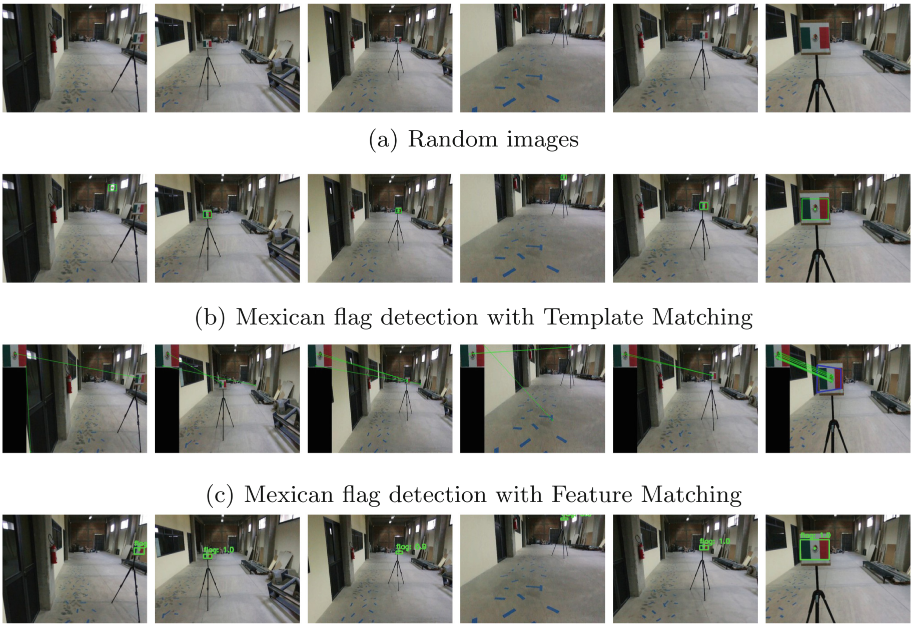
4.1 Mexican Flag Detection

We evaluated detection performance using 999 images of validation, splitting into 333 images rotated, 333 scaled and 333 with changes of illumination. The results obtained are shown in Table 1, presenting the number of times the flag is detected by each method and the percentage of success. In Fig. 10, we show the results of the detection.

Table 1. Mexican flag detection with different methods.

Method	Rotated	Scaled	With illumination	% successful
Template matching	117	294	33	74.47
Feature matching	85	107	238	43.04
Our system	308	326	331	**96.59**

The results obtained with Feature Matching achieves a 43.04% due to the lack of features in the template, causing the search for the flag to be missed in some cases. Instead, Template Matching obtains a suitable result 74.47% by using the cross-correlation and pyramidal scale, detecting the flag more times than Feature Matching. However, that method has problems of detection with rotated images in different angles. Nonetheless, our system implement with the SSD network finds the majority of images no matter the illumination, scales and rotations.



(a) Random images

(b) Mexican flag detection with Template Matching

(c) Mexican flag detection with Feature Matching

(d) Mexican flag detection with our system

Fig. 10. Detection of the Mexican flag with different methods.

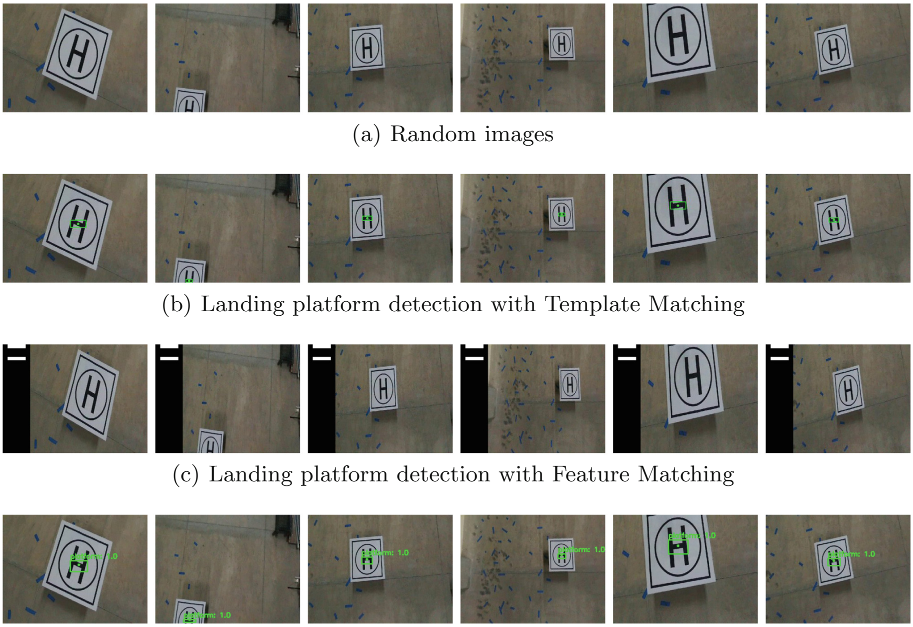
4.2 Landing Platform Detection

The landing platform detection was performed in the same way that the flag detection. We evaluated our system using 999 images which 333 are rotated, 333 scales, and 333 in the presence of changes of illumination, presenting the results in Table 2. Also, we show the landing platform detection using our system and the comparison with other methods in Fig. 11.

Table 2. Landing platform detection with different methods.

Method	Rotated	Scaled	With illumination	% successful
Template matching	25	280	184	48.94
Feature matching	10	17	29	5.60
Our system	298	329	333	**96.09**

The second result shows that the feature matching is not suitable for this test due to not finding enough features. The template matching method achieves 48.94% by realises a sweep in all the input image to localise the search template, obtaining a better result that feature matching method. Notwithstanding the



(a) Random images

(b) Landing platform detection with Template Matching

(c) Landing platform detection with Feature Matching

(d) Landing platform detection with our system

Fig. 11. Detection of the landing platform with different methods.

result, the speed performance is slow by performing the sweep in the whole image; therefore, it is not suitable for real-time tasks. For another hand, our system achievement 96.09% finding the landing platform in the different conditions of the image and faster than the other methods.

5 Autonomous Landing Results

The final test consists of the target detection and autonomous landing onboard of the MAV using our system in the Intel Computer Stick, communicating the vehicle with the computer via WIFI to obtain the images in real-time. This test is focused on the problem presented in the mission of the indoors competition in IMAV2019, which consist of autonomous navigation to detect a flag and then perform an autonomous landing. We validate our detection system carry out 40 autonomous flights split into 20 to offboard and 20 onboard of the MAV, taking the average confidence metric when detecting the targets, and the computationally cost of our system in fps. Table 3 shows the data of the autonomous flight offboard and onboard, obtaining a constant velocity of 90 fps offboard and 13 fps onboard. Fps represents the speed performance in frame per second.

Table 3. Autonomous landing results offboard and onboard of MAV.

Flight	Average flag confidence	Average platform confidence	Fps
Offboard	0.8971	0.9843	**90.5913**
Onboard	0.9386	0.9826	**13.0199**

Figure 12 and Fig. 13, we present a set of images that show all the process since that MAV taking off to detect the flag until detecting the landing platform to perform the autonomous landing.

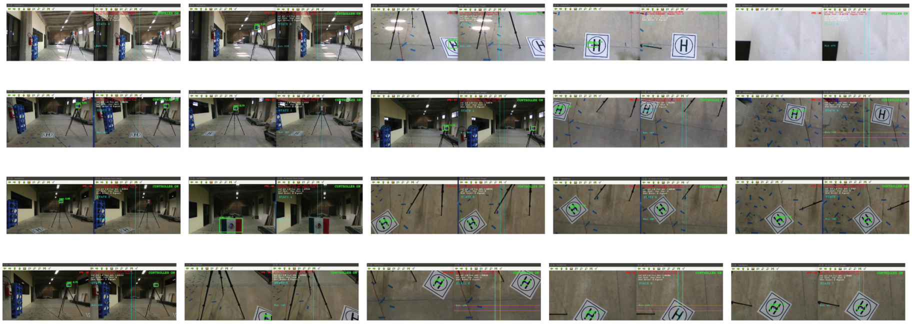


Fig. 12. Images sequence of the flights performed where we show the detection of the Mexican flag and landing platform. A video o this work for reviewing purpose is found at https://youtu.be/sYn9mo-2hvA

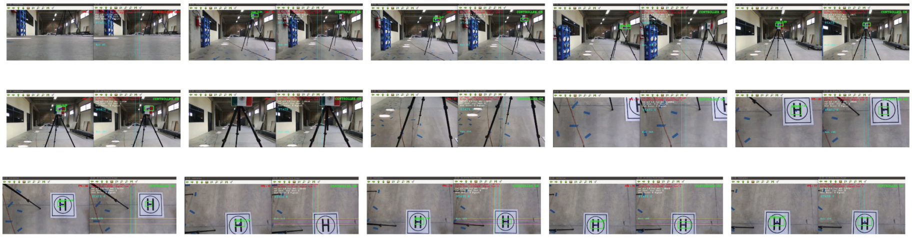


Fig. 13. Images sequence that shows all the process since the drone is taking off until detecting the landing platform to perform the autonomous landing. A video o this work for reviewing purpose is found at https://youtu.be/sYn9mo-2hvA

6 Conclusion

We have presented a target detection system using a deep learning implementation based on the SSD network to detect a flag and a Landing platform. This work is motivated by the challenge of having to perform autonomous landing as part of a mission included in the indoors competition of the IMAV 2019. The mission represents an existing problem in aerial robotics which consists of target detection while a MAV performs autonomous navigation, where the place to land has to be located by detecting a flag and then, the landing has to be performed by centring on a landing platform performing a landing routine autonomously. Thus, we have presented a detection system using the SSD7 network running on the Intel Computer Stick without GPU and architecture carried by the MAV, thus enabling it to perform onboard processing. This enabled the MAV to detect a flag and later on the landing platform while performing an autonomous flight. We validated our detection system with image datasets under multiple conditions of illumination even when the object is scaled or rotated, obtaining success of 96.59% for the flag detection and 96.09% for the landing platform detection. We compared our approach against other methods based on traditional computer vision techniques such as template and feature matching. Also, we test our system in real-time with offboard and onboard flights, obtaining metric confidence output of 0.9386 for the flag, and 0.9826 for the Landing platform, everything running on the Intel Stick at an average of 13.01 fps.

Future work involves the use of this framework for more sophisticated tasks such as object tracking during autonomous flight, involving much more targets and in outdoor environments.

References

1. Baomar, H., Bentley, P.J.: Autonomous navigation and landing of airliners using artificial neural networks and learning by imitation. In: 2017 IEEE Symposium Series on Computational Intelligence (SSCI) (2017)
2. Bartak, R., Hraško, A., Obdržálek, D.: A controller for autonomous landing of AR. Drone. In: The 26th Chinese Control and Decision Conference (2014 CCDC), pp. 329–334. IEEE (2014)
3. Barták, R., Hrasko, A., Obdrzalek, D.: On autonomous landing of AR. Drone: hands-on experience. In: The Twenty-Seventh International Flairs Conference (2014)
4. Bicer, Y., Moghadam, M., Sahin, C., Eroglu, B., Üre, N.K.: Vision-based UAV guidance for autonomous landing with deep neural networks. In: AIAA SciTech 2019 Forum, p. 0140 (2019)
5. Cabrera-Ponce, A.A., Martinez-Carranza, J.: A vision-based approach for autonomous landing. In: 2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS), pp. 126–131. IEEE (2017)
6. Fan, Y.M., Ding, M., Cao, Y.F.: Vision algorithms for fixed-wing unmanned aerial vehicle landing system. Sci. China Technol. Sci. **60**(3), 434–443 (2017). https://doi.org/10.1007/s11431-016-0618-3
7. Homma, Y., Moro, S.: Recognition of landing target of UAV by vision using machine learning. J. Signal Process. **23**(4), 193–196 (2019)
8. Madokoro, H., Kainuma, A., Sato, K.: Non-rectangular RoI extraction and machine learning based multiple object recognition used for time-series areal images obtained using MAV. Procedia Comput. Sci. **126**, 462–471 (2018)

9. Nguyen, P.H., Arsalan, M., Koo, J.H., Naqvi, R.A., Truong, N.Q., Park, K.R.: LightDenseYOLO: a fast and accurate marker tracker for autonomous UAV landing by visible light camera sensor on drone. Sensors **18**(6), 1703 (2018)
10. Polvara, R., et al.: Autonomous quadrotor landing using deep reinforcement learning. arXiv preprint arXiv:1709.03339 (2017)
11. Polvara, R., et al.: Toward end-to-end control for UAV autonomous landing via deep reinforcement learning. In: 2018 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 115–123. IEEE (2018)
12. Polvara, R., Sharma, S., Wan, J., Manning, A., Sutton, R.: Autonomous vehicular landings on the deck of an unmanned surface vehicle using deep reinforcement learning. Robotica **37**(11), 1867–1882 (2019)
13. Price, E.: Deep neural network-based cooperative visual tracking through multiple micro aerial vehicles. IEEE Robot. Autom. Lett. **3**(4), 3193–3200 (2018)
14. Recker, S., Gribble, C., Butkiewicz, M.: Autonomous precision landing for the joint tactical aerial resupply vehicle. In: 2018 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), pp. 1–8. IEEE (2018)
15. Rodriguez-Ramos, A., Sampedro, C., Bavle, H., De La Puente, P., Campoy, P.: A deep reinforcement learning strategy for UAV autonomous landing on a moving platform. J. Intell. Robot. Syst. **93**(1–2), 351–366 (2019). https://doi.org/10.1007/s10846-018-0891-8
16. Rojas-Perez, L.O., Munguia-Silva, R., Martinez-Carranza, J.: Real-time landing zone detection for UAVs using single aerial images. In: Watkins, S. (ed.) 10th International Micro Air Vehicle Competition and Conference, Melbourne, Australia, pp. 243–248, November 2018
17. Skoczylas, M.: Vision analysis system for autonomous landing of micro drone. Acta Mechanica et Automatica **8**(4), 199–203 (2014)
18. Truong, N.Q., Nguyen, P.H., Nam, S.H., Park, K.R.: Deep learning-based super-resolution reconstruction and marker detection for drone landing. IEEE Access **7**, 61639–61655 (2019)
19. Xu, Y., Liu, Z., Wang, X.: Monocular vision based autonomous landing of quadrotor through deep reinforcement learning. In: 2018 37th Chinese Control Conference (CCC), pp. 10014–10019. IEEE (2018)
20. Xu, Y., Zhang, Y., Liu, H., Wang, X.: Deep learning for UAV autonomous landing based on self-built image dataset. In: Eleventh International Conference on Machine Vision (ICMV 2018), vol. 11041, p. 110412I. International Society for Optics and Photonics (2019)
21. Yang, S., Scherer, S.A., Schauwecker, K., Zell, A.: Onboard monocular vision for landing of an MAV on a landing site specified by a single reference image. In: 2013 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 318–325. IEEE (2013)
22. Yang, S., Scherer, S.A., Schauwecker, K., Zell, A.: Autonomous landing of MAVs on an arbitrarily textured landing site using onboard monocular vision. J. Intell. Robot. Syst. **74**(1–2), 27–43 (2014). https://doi.org/10.1007/s10846-013-9906-7
23. Yu, L.: Deep learning for vision-based micro aerial vehicle autonomous landing. Int. J. Micro Air Veh. **10**(2), 171–185 (2018)
24. Zhao, Y., Pei, H.: An improved vision-based algorithm for unmanned aerial vehicles autonomous landing. Phys. Procedia **33**, 935–941 (2012)