AIAA SciTech Forum
8–12 January 2018, Kissimmee, Florida
2018 AIAA Information Systems-AIAA Infotech @ Aerospace
10.2514/6.2018-2138

Real Time Embedded System Framework for Autonomous Drone Racing using Deep Learning Techniques

Sunggoo Jung*, Hanseob Lee† and David Hyunchul Shim‡
Unmanned Systems Research Group (USRG), KAIST, Daejeon, 34141, South Korea

We present an robust visual detection method for indoor autonomous drone racing (ADR). Our unmanned micro aerial vehicle (MAV), that is built with low-cost, off-the-shelf hardware, detect the racing gates with a monocular camera using deep-learning method. The biggest challenging tasks for this ADR is to detect quickly and reliably race gates while avoiding collisions. In this paper, we introduce convolution neural network (CNN) called single shot detector (SSD) to robustly detect the race gate in various indoor light condition and environment. All vision processing, except flight control, runs in real time on the NVIDIA embedded GPU (Tegra X1) and the learning process is implemented on the desktop GPU (NVIDIA GTX980) hardware platform. We provide details about the hardware system and software algorithms used to implement the proposed solution. Based on the learning results, we show that our proposed solution can provide quick and reliable results in the indoor environment through experiments that pass through the race gate.

I. Nomenclature

N	=	number of matched default boxes
l	=	predicted box
g	=	ground truth box
L_{conf}	=	confidence loss
L_{loc}	=	localization loss
α	=	weight term
c	=	multiple classes confidences
s_{min}	=	lowest layer scale
s_{max}	=	highest layer scale
m	=	number of feature maps

II. Introduction

Drone racing is gaining popularity as one of the new sports. The improved performance of avionics, FPV devices and quadrotor-type MAV makes it possible [1]. Furthermore, autonomous unmanned race competitions were tried in line with the development of indoor navigation technology and the flow of autonomous robot [2]. Vision-based object detection is an important component in such MAV applications. As the visual device getting smaller and the development of GPU computing algorithms, robot applications using complex vision processing tasks are now available in the MAV platforms. In the drone race, the drone must pass through the race gates placed along the course. Thus, fast and accurate gate detection is an important factor for a successful race.

In the drone-racing arena a number of gates are placed with the same shape and color along the course. Therefore, it is difficult to detect the gate with classical image recognition techniques due to the gate duplication problem. Classical method are mainly tuned by people according to feature information (i.e., shape or color) thus, it is quite sensitive to the circumstance. In particular, in the case of two or more gates are overlapped each other with a depth and slight position difference, there is a high possibility that the two gates recognized as one large gate. Therefore, other vision approaches are need to solve this classical image recognition problem.

*Graduate Researcher, Unmanned Systems Research Group (USRG)., KAIST, 291 Daehakro, Daejeon, South Korea, AIAA Member.
†Graduate Researcher, Unmanned Systems Research Group (USRG)., KAIST, 291 Daehakro, Daejeon, South Korea, AIAA Member.
‡Associate Professor, Unmanned Systems Research Group (USRG)., KAIST, 291 Daehakro, Daejeon, South Korea, AIAA Senior Member.

Deep learning techniques using convolution neural networks can resolve this image overlap problem [3, 4]. The main advantage is that it does not require human tuning process because it learns the object model from raw pixel data. However, deep-learning consumes a lot of computing power, it has been mainly applied in an environment where a relatively high-performance computer can be used such as a laboratory or an autonomous car environment [5]. The development of GPU computing technology and the introduction of high-performance small-embedded board, it enables on-board processing of computer vision technology using deep learning technique.

In this paper, we presents a hardware and software framework of a practical quadrotor system for autonomous drone racing. This system uses the CNN technique called SSD [6] to search the race gate and show the real-time recognition performance through the on-board processing. The learning process was performed in the desktop GPU environment (NVIDIA GTX980) using the caffe deep learning library and the learning results were applied using the NVIDIA TX1 embedded board.

The rest of this paper is structured as follows: The rest of this document is structured as follows: In section III we present an overview of the stat-of-the-art in the past decade. Section IV overviews of the hardware platform and software architecture. Details of the modified SSD network is discussed in section V. The experiment results of proposed CNN network with practical Implementation and evaluations are presented in Sect VI. Sect VII draws the conclusions and points out possible future works.

III. Related Work

In robotics computer vision, object detection has been an important role for the successful mission assignment. Among the various literature of object detection, we review the papers over the past decade in object detection history.

Object detection performance has been greatly increased by the Viola-Jones algorithm [7]. Their work shows real-time face detection performance from the webcam feed. They fed the collected features to the binary classifier called support vector machine and it works fairly well. However, this algorithm mainly focuses on facial detection. The HOG [8] method uses gradient to show the flow from light to dark. It shows impressive performance to convert the original image into simple basic structure of face. It still have hand-code problem to make the algorithm knows the feature map characteristics. Thanks to the development of GPU technology and vast data set collection, deep-learning method has been implemented in the practical object detection method [9]. It enables the computer can do the task of classification and detection simultaneously. But it still have slow speed problem. R-CNN [10] uses selective search algorithm to make the faster speed. There are various neural network based classification method such as VGG-16, GoogLeNet [11, 12] and it is combined with detectors so that faster and stronger object detection networks are presented called SSD and YOLO [6, 13]. However, it is not easy to apply deep-learning method in the small embedded system to work in real-time.

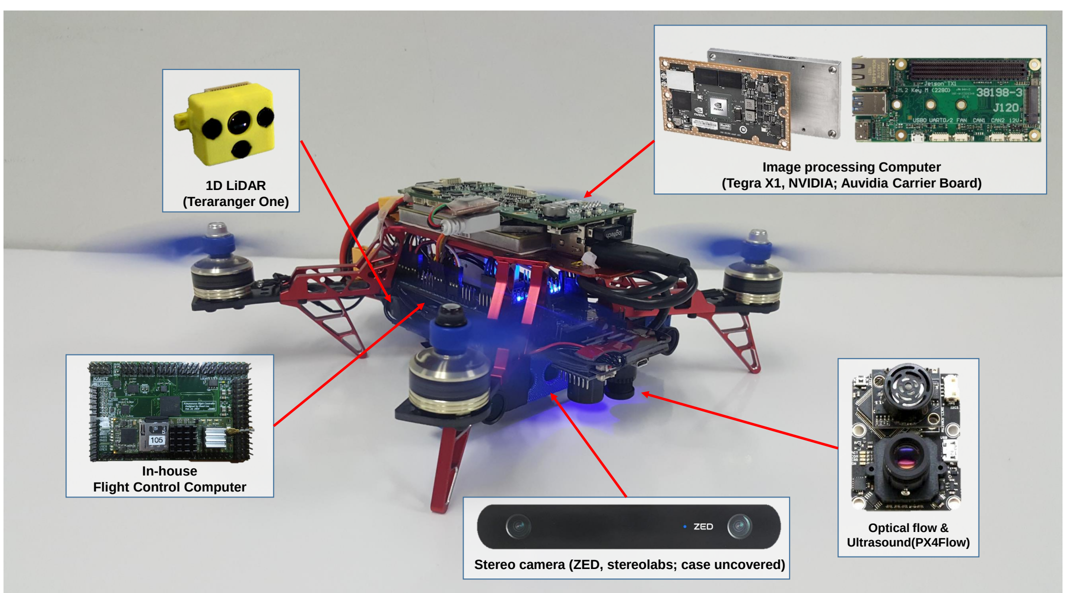


Fig. 1 MAV configuration for an autonomous drone racing.

IV. System Description

We developed a quadrotor type hardware platform as shown in Fig. 1. Specifically, we present our hardware platform and software architecture design considerations to make enable deep-learning assisted autonomous drone race.

A. Hardware Platform

The hardware requirements in this work is the deep-learning available MAV. Reaching such high computing performance during flight leads to small and powerful embedded computer. To make that we searched an off-the-shelf computers which have available computing power and weight. The available choices are listed in Table 1.

Table 1 Performance evaluation of commercially available embedded computer lists. To meet the design requirements, GPU computing should be available. We expect the payload for the embedded computer is less then 100g.

Embedded Computers	Size [mm]	Weight [g]	CPU	GPU Speed
Raspberry Pi3	85*56*2	42	Cortex-A53 1.2GHz	N/A
Odroid XU4	85*56*2	38	Cortex-A15 2.0GHz	N/A
NVIDIA Jetson TK1	133*133*30	120	Cortex-A15 2.0GHz	300 GFLOP/s
NVIDIA Jetson TX1	87*50*2	88	Cortex-A57 1.9GHz	1 TFLOP/s

The robot need to fly autonomously while pass-through the gate rely only on-board processing. Based on the available options, NVIDIA Jetson TX1 along with the Auvidea J120 carrier board is selected to our embedded computer. It has higher GPU computing performance with relatively light weight and the carrier board is well designed for small platform such as MAV. Thus, it intimately matches our hardware design requirements.

We have implemented a MAV set suitable for drone racing using 250 class frames (Fig. 1). For low-level controller we use our in-house flight control computer (FCC). Thanks to our custom FCC we can easily modify the firmware to meet our performance design condition. From the front facing monocular (FLIR FireFly-MV) camera VGA resolution image input is obtained at 60Hz. This image is used as the input of CNN to detect the gates. In addition, in the lower part of the drone, the PX4Flow optical flow sensor and TeraRanger One 1D LiDAR are also installed to get velocity measurement and Altitude information.

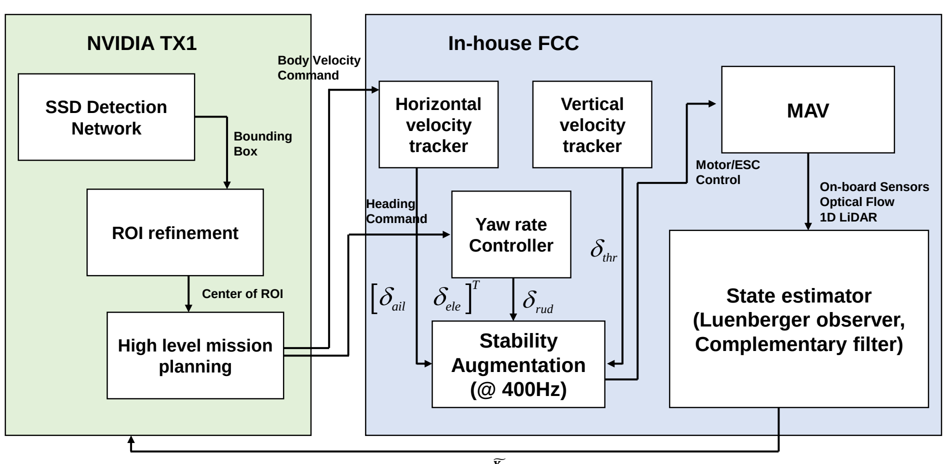


Fig. 2 Block diagram of high level control software architecture. Nvidia TX1 and the in-house FCC are communicating via RS-232 communication protocol

This system uses lithium polymer battery and requires 4S battery and we select 1800mAh capacity. Total weight of MAV is 1.1Kg including battery and propulsion system. FCC, embedded computer and propulsion system share that the single power source (4S li-po battery). From the flight test it's duration is around seven minuets using 80% of rated capacity of the battery while running whole detection and control process.

B. Software Details

To fly through the gates, mainly two tasks are performed which is detection and guidance control. Those individual tasks are functioned together in the hight level computer (Nvidia TX1). Fig. 2 shows the mission process from the high-level embedded computer to the MAV. The mission tasks in the NVIDIA TX1 are powered by ROS framework which are made for easy to test and debug in robot operations. In ROS, detection and guidance control are composed into two different packages and those are communicate as an message format at an 20Hz speed.

We use Caffe SSD library for train and inference the network. SSD network use VGG-16 as an base network and it modifies extra feature layer. This network structure is similar to YOLO. It uses only one network to find the bounding box of object and define the class. YOLO contains bouding box and class information only the final featur map, but SSD contains those information seperately to the hidden layers (six extra feature layer) to enhance the performance. According to SSD [6], The overall objective loss function is:

$$L(x, c, l, g) = \frac{1}{N} \left(L_{conf}(x, c) + \alpha L_{loc}(x, l, g) \right) \tag{1}$$

and they define the default box size from the feature map,

$$s_k = s_{\min} + \frac{s_{\max} - s_{\min}}{m - 1}(k - 1), \qquad k \in [1, m] \tag{2}$$

SSD detection method show 74.3 mAP and 40fps of high accuracy and fast speed in desktop environment. However it still not adequate for an embedded purpose. In our test it shows 2Hz speed (462.04 ms) on Nvidia TX1, therefore it is not suitable for real-time autonomous drone racing. Fig. 3 shows the modified SSD network to meet our real-time criteria. We modified abandoned extra feature layer to increase the detection speed. High lever feature layers which are not critical to gate detection and unnecessary low level feature map while bounding box regression are removed.

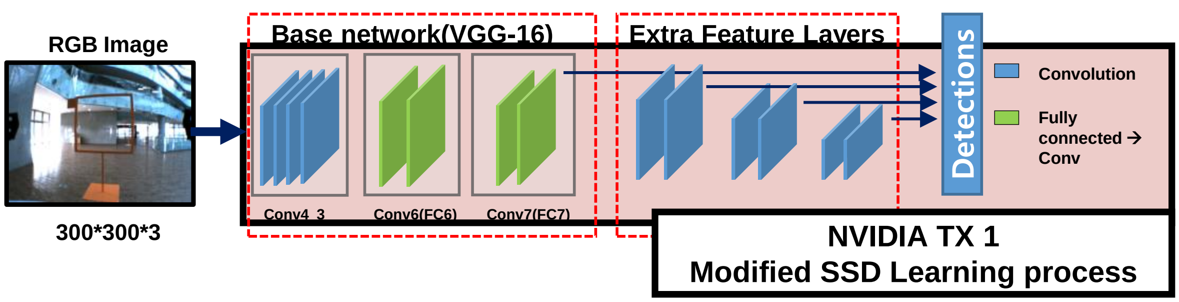


Fig. 3 Modified SSD network diagram to detect the gate to pass. Software packages are power by ROS environment

V. Race Gate Detection with VGG-16 based modified SSD

Table 2 Power consumption and object detection speed on different hardware platform

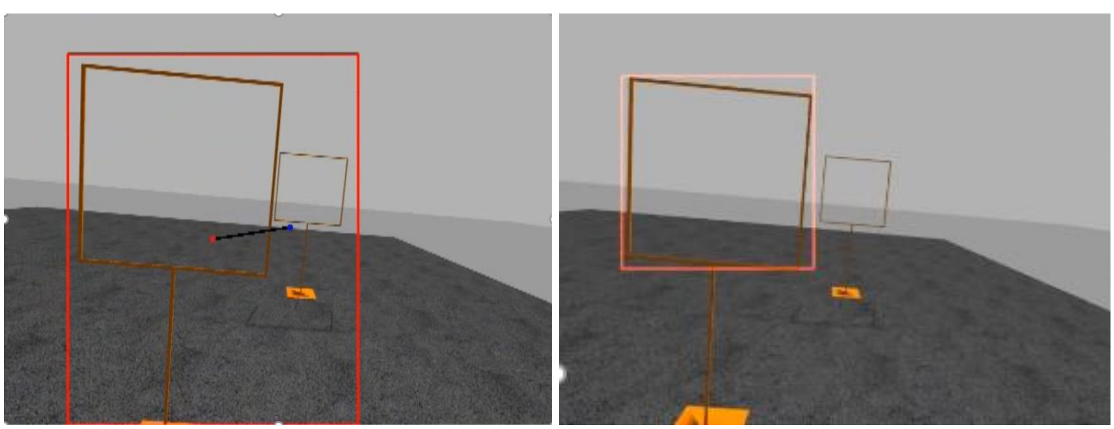
	Power consumption	Original SSD	proposed Network
TX1	10W	210.4ms	108.4ms
TK1	7W	1365.3ms	786.5ms

Our SSD algorithm builds on top of Caffe deep-learning framework. The SSD detects the gate given to the experimental environment. For each detected gates, the network provides image coordinates of left top and right bottom

Fig. 4 Gate detection result by SSD running on the Jetson TX1 of drone

pixel. From this coordinate we can get the center point of the detected gates. First, we compare the performance of the Original SSD and our modified algorithm. For more information, we tested it with NVIDIA TK1 also. Table 2 shows the results of gate detection using the same algorithm in TK1 and TX1. Compared to TK1, TX1 has about 6 times fast detection speed while consuming similar power. This is the main reason that we select the embedded computer to the NVIDIA TX1. At the world first autonomous drone racing competition held in 2016, high level computing was performed using TK1 [1]. However, as can be seen from the results of the table 2, it shows a very slow process speed to implementing deep learning methods. We studied the network with the PASCAL VOC data set structure. An example of the output of the learning result is shown in figure 4

VI. Flight Test Experiment



(a) (b)

Fig. 5 Simulation result of conventional color-based object detection method versus our proposed CNN based deep-learning method. Figure (a) shows the main negative points of color-based method. It cannot distinguish the two overlapped gates. Thus, it detects those two gates as an one big gates. Nevertheless, our proposed method robustly detect even the gates are overlapped each other.

Before the flight test, we briefly tested our network in the Gazebo simulation environment. Fig. 5 (a) shows the conventional color-based edge detection method. This method has advantage of consuming low computing power and easy to design however, hard to resolve the ambiguous problem when the two or more gates are overlapped each other. Our proposed deep-learning method properly detect the gates even in the overlapped situation. We set the algorithm to boxing the boundary of gate which have maximum probability. Our detection package give high probability to the bigger gate therefore, always grab close gate with MAV. To pass through the gate we use the center point matching method which are described our previous research result [1]. We set the image center point as a guidance target and control the MAV to match the gate center and the image center. Fig. 6 shows the snapshot of the quadrotor eye view. Path error is the pixel distance between the detected gate center and the image center. Distance means the gap between the MAV and the detected gate.

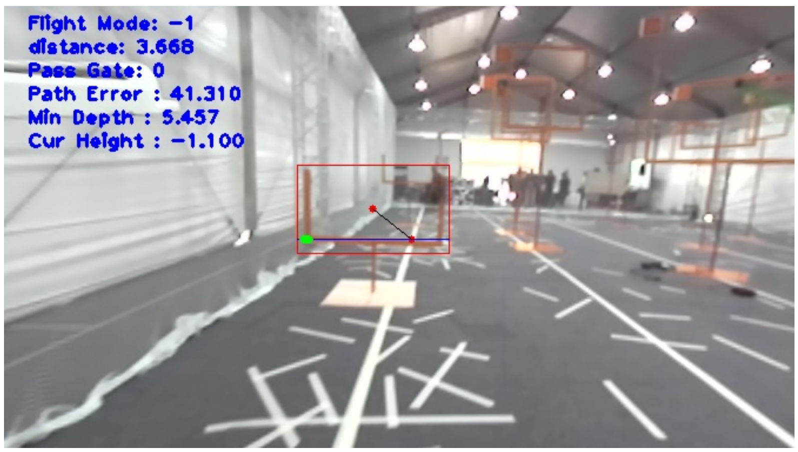


Fig. 6 Snapshot of the quadrotor eye view. Center point matching method to match the gate center and the image center.

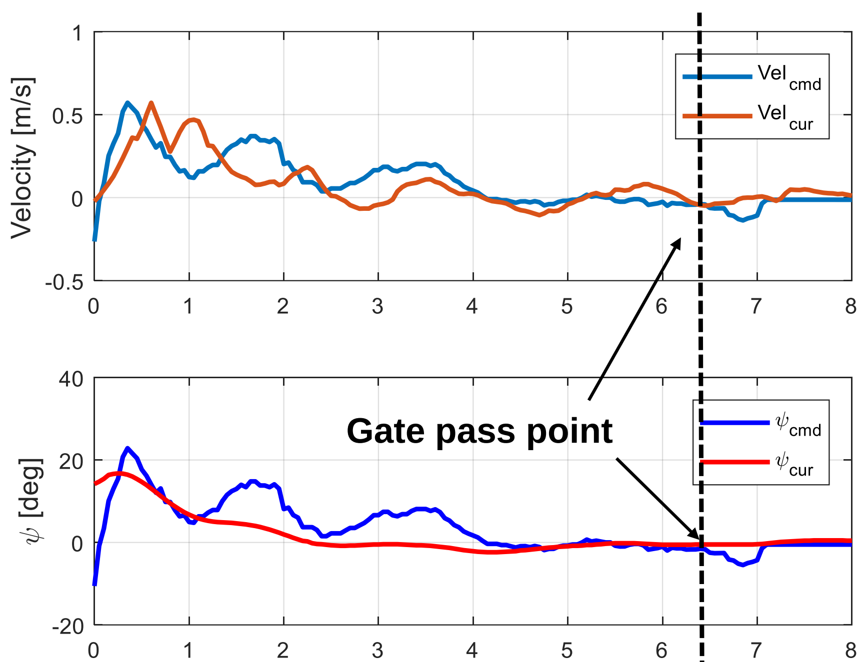


Fig. 7 Command tracking result while pass through the gate. Velocity command of horizontal axis and the yaw rate command are well followed.

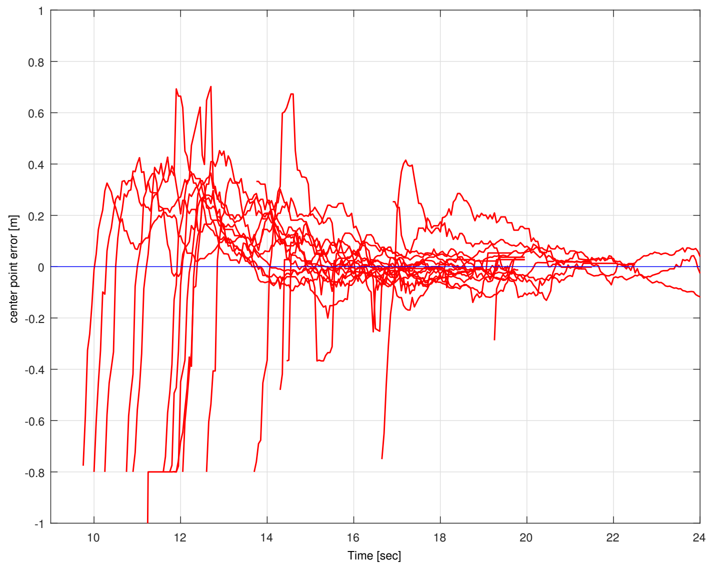


Fig. 8 Command tracking result while pass through the gate. Velocity command of horizontal axis and the yaw rate command are well followed.

High-level computer send the velocity control command to FCC via RS-232 communication protocol. Fig. 7 shows the horizontal velocity and yaw rate command while pass through the detected gate. As the center point difference decreases, the horizontal velocity command is also close to zero. Velocity command are followed by designing the low-level autopilot using luenberger observer with classical PID controller. We performed twenty repeated experiments to test the performance of our center point matching algorithm. Fig 8 present the result, it shows successful passing at every twenty trials.

VII. Conclusion and Future work

In this paper, deep-learning based gate detection is performed through the proposed autonomous drone racing framework. Our gate detection technique works at a speed of about 10Hz. We properly integrated the high-computing mission task into compact drone system. The mission of gate passing is well designed and it shows appropriate performance.

However, much faster recognition performance is required while considering the fast running drone racing characteristics. Therefore, after submitting this paper, we will improve learning part by changing the base network. Currently, we use VGG network as the base network for gate detection, but in our consideration alexnet or modified alexnet network will show faster recognition performance while maintaining accuracy. In addition, since NVIDIA recently released the TX2 board, the switch to the TX2 board is also subject to review. TX2 is about 20 times faster than TX1, therefore it will show faster and more accurate recognition performance. Currently, the learning and flight algorithms are integrated into the onboard embedded system, and we plan to increase the robustness of gate detection and pass through performance through numerous flight tests.

Acknowledgments

This paper is based upon work supported by the Ministry of Trade, Industry & Energy (MOTIE, Korea) under the Industrial Technology Innovation Program. No.10067202, "Development of robot system for indoor reconnaissance mission in complex disaster situations"

References

[1] Jung, S., Cho, S., Lee, D., Lee, H., and Shim, D. H., “A direct visual servoing-based framework for the 2016 IROS Autonomous Drone Racing Challenge,” *Journal of Field Robotics*, 2017. doi:10.1002/rob.21743, URL `http://dx.doi.org/10.1002/`

rob.21743.

[2] Moon, H., "IROS — 2016 Autonomous Drone Racing in Mos Espa Daejeon Arena," , 2016. URL http://ris.skku.edu/home/iros2016racing.html, [Online; accessed 31-May-2017].

[3] Bachrach, A., He, R., and Roy, N., "Autonomous flight in unknown indoor environments," *International Journal of Micro Air Vehicles*, Vol. 1, No. 4, 2009, pp. 217–228.

[4] Çelik, K., and Somani, A. K., "Monocular vision SLAM for indoor aerial vehicles," *Journal of electrical and computer engineering*, Vol. 2013, 2013, pp. 4–1573.

[5] Jung, S., Lee, U., Jung, J., and Shim, D. H., "Real-time Traffic Sign Recognition system with deep convolutional neural network," *Ubiquitous Robots and Ambient Intelligence (URAI), 2016 13th International Conference on*, IEEE, 2016, pp. 31–34.

[6] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C., "SSD: Single shot multibox detector," *European Conference on Computer Vision*, Springer, 2016, pp. 21–37.

[7] Viola, P., and Jones, M., "Robust Real-time Object Detection," *International Journal of Computer Vision*, 2001.

[8] Dalal, N., and Triggs, B., "Histograms of oriented gradients for human detection," *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, Vol. 1, IEEE, 2005, pp. 886–893.

[9] Krizhevsky, A., Sutskever, I., and Hinton, G. E., "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[10] Ren, S., He, K., Girshick, R., and Sun, J., "Faster R-CNN: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, 2015, pp. 91–99.

[11] Simonyan, K., and Zisserman, A., "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[12] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A., "Going deeper with convolutions," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[13] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., "You only look once: Unified, real-time object detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.