Deep learning for UAV autonomous landing based on self-built image dataset

Xu, Yinbo, Zhang, Yongwei, Liu, Huan, Wang, Xiangke

Yinbo Xu, Yongwei Zhang, Huan Liu, Xiangke Wang, "Deep learning for UAV autonomous landing based on self-built image dataset," Proc. SPIE 11041, Eleventh International Conference on Machine Vision (ICMV 2018), 110412I (15 March 2019); doi: 10.1117/12.2522751

Event: Eleventh International Conference on Machine Vision (ICMV 2018), 2018, Munich, Germany

Deep Learning for UAV Autonomous Landing Based on Self-built Image Dataset

Yinbo Xu[1], Yongwei Zhang[1], Huan Liu[2], Xiangke Wang[1]*
[1] College of Mechatronic Engineering and Automation, National University of Defense Technology
[2] Jiuquan Satellite Launch Center Technology Department
E-mail: xkwang@nudt.edu.cn

ABSTRACT

An end-to-end deep learning (DL) control model is proposed to solve autonomous landing problem of the quadrotor in way of supervised learning. Traditional methods mainly focus on getting the relative position of the quadrotor through GPS signal which is not always reliable or position-based vision servo (PBVS) methods. In this paper, we have constructed a deep neural network based on convolutional neural network(CNN) whose input is raw image. A monocular camera is used as only sensor to capture down-looking image which contains landing area. To train our deep neural network, we have used our self-built image dataset. After training phase, the well-trained control model is tested and the results perform well. Light changes and background interferences have little influence on the model`s performance, which shows the robustness and adaptation of our deep learning model.

Keywords: Deep learning (DL), supervised learning, convolutional neural network(CNN), autonomous landing, quadrotor.

1. INTRODUCTION

Unmanned aerial vehicle (UAV) has attracted people`s eyes in both academia and industry in last decades and gained rapid development. Its applications include but not limit to resource exploration, disaster relief and logistics delivery. In these scenes, autonomous landing is a significant function which can improve the efficiency of executing tasks in complex unknown environments. Many existing methods is based on IMU and GPS which is light, small and low-cost. But the disadvantages are also obvious. IMU will accumulate errors during the working process and the GPS signal is not reliable. Camera is another available sensor which is cheap and fully developed.

Present auto-landing methods based on camera mainly focus on position based vision servo (PBVS) and image based vision servo (IBVS). PBVS is to determine characteristic points of specific mark in vision firstly, and then calculate the relative pose of UAV. Control law is designed based on the relative pose to the mark. The mark usually uses ApriTag or Aruco mark system, which consists of black and white area of same size. Different combination corresponds to an unique id. While IBVS does not need estimate relative pose. Its vision system is integrated into the design of control law. IBVS utilizes the information of image pixel layer directly, which not only reduce the complexity of calculation but also can avoid the influences brought by noise when measuring data. Many control laws based on IBVS have been developed [1] [2]. Xu G [3] proposes an IBVS auto landing method which uses a T shape pattern as land mark. This method uses the actual shape and disappearing line to conduct the calculation. Due to the fact that the image is highly relied on the light condition, the mark is supposed to have good reflection characteristic and contrasts with background.

The rise of deep learning in recent years provides a new think to the IBVS, namely an end-to-and direct control method whose input is the sensor image and output is the commands of executor. Since Hinton [4] opened an era of developing deep learning (DL) in 2006, DL has subverted many fields including voice recognition, natural language understanding, image classification and detection, machine translation etc. And a new pattern, namely the end-to-end deep neural network outputting predicting results, has gradually formed. Appling deep learning to autonomous landing of UAV is studied in some literatures. Chen [5] proposed a Faster Regions Convolutional Neural Network (Faster R-CNN) recognition algorithm to realize auto landing of UAV in power patrolling with the well-designed landing sign. The result has showed better identification accuracy compared with other three target identification methods. The proposed algorithm aims to estimate position and direction of UAV, is not an end-to-end method. Polvara [6] tried to use deep neural network to handle the quadrotor auto landing problem by using an end-to-end method. The method is based

Eleventh International Conference on Machine Vision (ICMV 2018), edited by Antanas Verikas,
Dmitry P. Nikolaev, Petia Radeva, Jianhong Zhou, Proc. of SPIE Vol. 11041, 110412I
 · doi: 10.1117/12.2522751

on reinforcement learning not supervised learning. The control model is trained in large amounts of “trial and error” experiments in simulation environment. When the well-trained model is tested in real world, the mark and background must be same with that in simulation world to narrow the gap between the real and simulation world.

In this paper, we proposed an end-to-end control model based on deep learning in supervised way to solve the auto landing problem of quadrotor. More specifically, we use a down-looking monocular camera as the only sensor to capture images of ground where a land mark is placed in landing region. Raw image is input into deep neural network, and our model outputs high-level commands directly. The dataset used in training phase is collected based on our previous work [7]. In the dataset, the labeled data are the down-looking images while the label is the corresponding correct speed. After our deep learning model being trained with the well-constructed dataset, the model is tested in both indoor and outdoor environments. With different light conditions and extra interferences being applied during test, our model still demonstrates good robustness and adaptability.

The main contributions are: **1**. Have developed an end-to-end deep learning control model for quadrotor autonomous landing in supervised learning way. **2**. Tests are conducted in real world including indoor and outdoor.

2. PROBLEM DESCRIPTION

The description of UAV auto landing problem is shown in Fig. 1. A tag plane is fixed on the ground, used for visual guidance landing sign. The QR code tag in Fig. 1 is just an example, the proposed solution has no restrictions with the ground signs. The aerial vehicle is equipped with a down-looking camera. After the deep neural network being trained, the model can exploit the features of tag plane, output suitable control command to land the UAV. At the same time, UAV communicates with ground station whose control commands have higher priority in case of emergency.

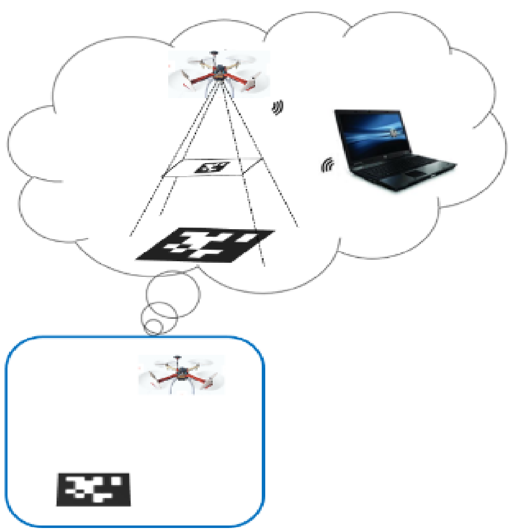


Figure 1. Problem description of UAV visual auto landing.

3. CNN-BASED AUTO LANDING

Deep convolution neural networks (CNNs) have been widely used in deep learning. CNNs have achieved better results than the previous state-of-the-art in obtaining features from image. The current neural network is deep enough to achieve the perfect result. But the amount of calculation is huge which requires strong hardware as a support. For UAV auto landing application in real world, the miniaturized onboard processor's computing capability is severely limited, the real-time control needs to be guaranteed at the same time. So, a proper CNN based deep-learning model should be explored and chosen to construct the end-to-end model as shown in Fig. 2. End-to-end means the deep network only needs to be input image and directly output a UAV velocity control command.

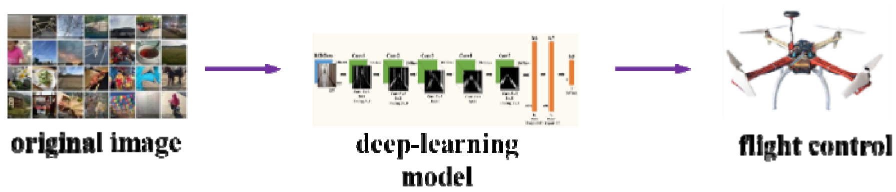


Figure 2. Deep learning end-to-end control programs

3.1 A brief Introduction of the CNN

A CNN typically contains three types of layers, including convolution layers, subsampling layers and fully connected layers. Denote by I_i^{l-1} the input data of the l-th layer and I_j^l the output data, where l∈{1, L}. The convolution layer is described as follows:

$$I_j^l = sigma\left(\sum_{i=1} I_i^{l-1} \otimes k_{ij}^l + b_j^l\right) \tag{1}$$

where k_{ij}^l is the convolution kernel and b_j^l is the bias term, *sigma* () is the activation term using the sigmoid function, and $\otimes$ denotes a convolution operation. The convolution operation is performed first with convolution kernels. The bias term is added to the resulting feature maps, in which an activation operation is performed subsequently.

The sub-sampling layer is usually used to select the dominant features over nonoverlapping square windows per feature map as follows:

$$I_j^l = downsample\left(I_i^{l-1}, m\right) + b_j^l \tag{2}$$

where b_j^l is the bias term, *downsample*( ) performs the average sub-sampling operation in each $m \times m$ region that does not overlap with each other.

The hierarchical feature extraction architecture consists of several convolution layers and sub-sampling layers. The final layer is combined into a feature vector by several fully connected layers. The last classification layer is usually a softmax layer with the same number of neurons with the number of output classes.

3.2 CNN-based Model for Auto Landing

The process of solving the UAV auto landing by deep learning method mainly consists of two steps, namely offline training and online applying, which is shown in Fig. 3.

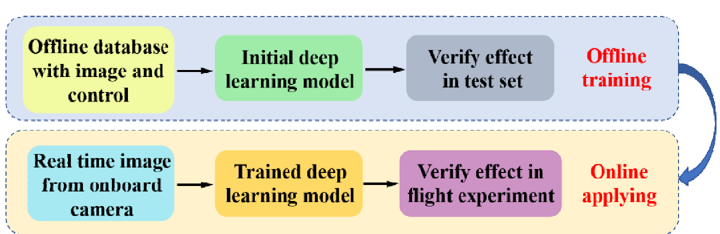


Figure 3. Process of deep learning end-to-end control

The landing dataset include 35000 frames in total, covering different initial landing position and attitude, lighting and background conditions. Fig.4 is the snapshot of one sample randomly selected from the dataset. Our dataset has been split into disjoint training (30000 frames), verifying (2000 frames) and testing (3000 frames) sets. The samples with various conditions are evenly distributed in each set. The label is the corresponding correct velocity value in four dimensions.

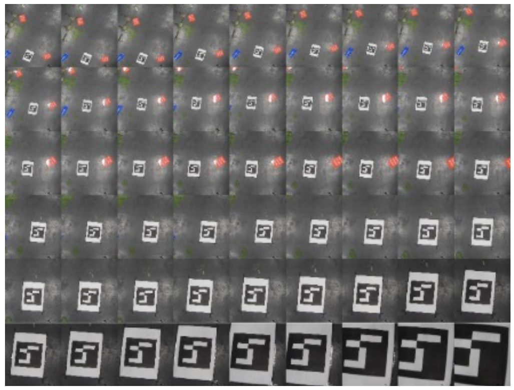


Figure 4. The snapshot of one sample in dataset.

Considering the limited computing capability of onboard processor and real-time requirements with UAV control, a CNN-based model similar to the *CaffeNet* model is chosen. *CaffeNet* has been proven to perform well when applied to many image classification problems. For regress problem of "image-to-velocity control" UAV auto landing, we made some modification as follows,

- Modify input layer from RGB three channels to single channel
- Modify the output number of the last fully connected layer to match the four-dimensional velocity control output
- Increase the *Sigmoid* activation layer to normalize the output vector since the velocity label has been normalized.
- Modify the loss layer function type as the Euclidean distance loss function for regress application

The modified network architecture is shown in Fig.5. We consider a matrix of $1\times227\times227$ neurons as the input layer, followed by a number of hidden layers. The last three fully connected layers act as a general regressor. For a given input, the CNN outputs four values, representing the predicting velocity value in X, Y and Z directions and angular velocity in Z direction respectively.

The original weights of *CaffeNet* model is used. And the modified parts of the model are initialized with a gaussian initializer whose mean and standard deviation is 0.0 and 0.01. The learning rate of modified parts is 10 times than the basic learning rate. The model weights are optimized using stochastic gradient descent(SGD) to minimize the Euclidean distance error over the training set.

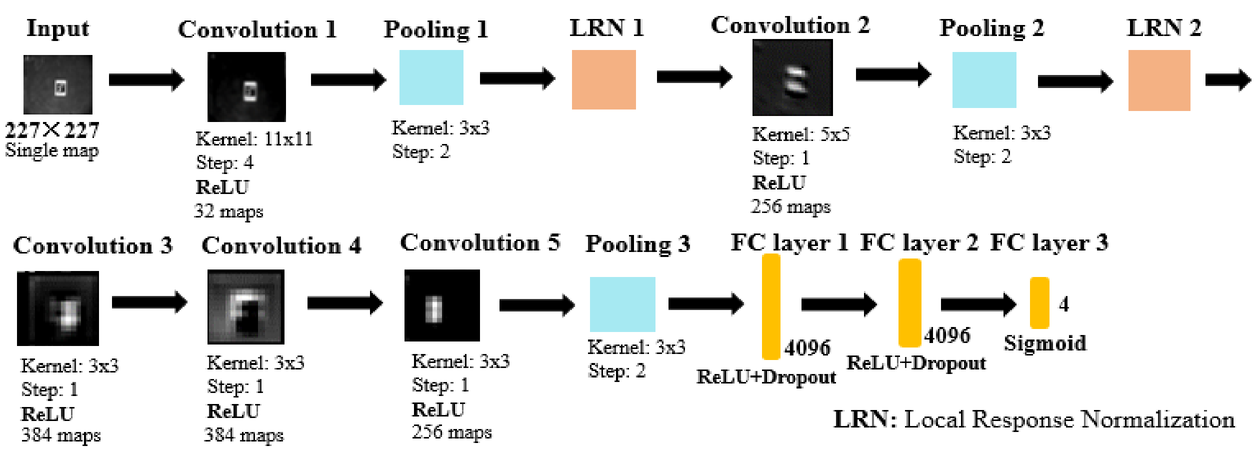


Figure 5. Architecture of the CNN used in our system.

30000 training data are used. Data preprocessing procedures includes greying original RGB images, resizing to 227×227 pixels, normalizing image and label of velocity to [0,1], and disrupting the data randomly. The CNN model is trained using backpropagation for 90 epochs, which requires about 4 hours on a workstation equipped with an Nvidia Tesla P100 GPU. The learning rate is initially set to 0.005, then scaled by a factor of 0.9 per epoch.

4. EXPERIMENT

Loss value is the training error during the training phase. We have collected the loss value and demonstrated the curve in Fig 6. We can see that the loss convergences with the training going on and keep stable near 0. And we test our trained model on test set. The testing set contains a variety of environmental conditions to ensure the reliability of test results. The average errors between the predict and true value of four-dimensional velocity control with linear velocities in X, Y and Z direction and angular velocity in Z direction are 0.0183m/s, 0.0194m/s, 0.0035m/s and 0.0132rad/s respectively. Considering the velocity amplitudes of four dimensions are 0.2m/s, 0.2m/s, 0.15m/s and 0.2 rad/s respectively, the average error in test set is quite small. So, both loss curve and average testing error indicate the training phase is successful.

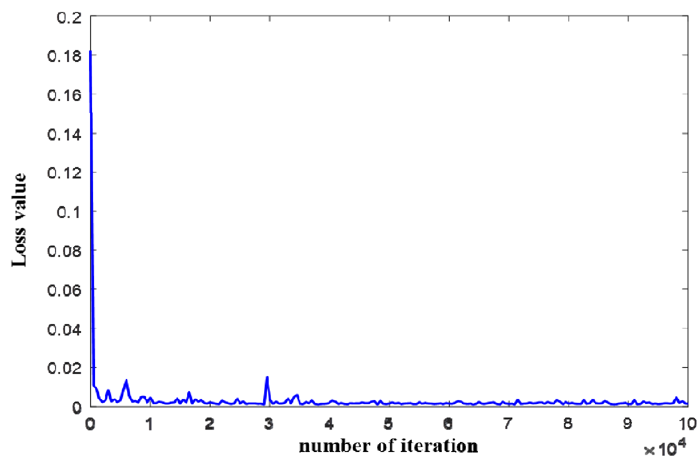


Figure 6. Loss curve during training phase.

Consider the good performance of the model on the test set, the trained model is applied to actual online flight experiments. A quadrotor UAV based DJI F450 is used in our flight experiments, as shown in Fig.7. Onboard processor is Nvidia TX1 that equipped with a camera facing the ground vertically.

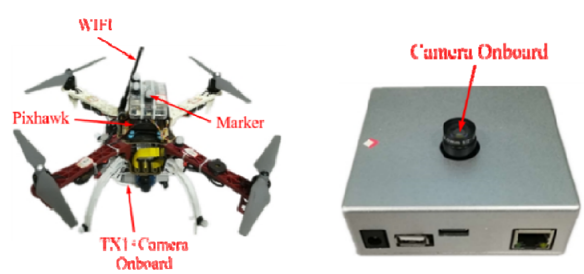


Figure 7. Quadrotor vehicle and TX1 processor used in the experiment.

VICON motion capture system is used to provide accurate position and attitude information of the UAV. During the flight experiment, the original image captured from TX1 onboard camera as the input of the trained CNN-based model, after the forward real-time computing, the model output predicted four-dimensional velocity control values, which be directly send to pixhawk flight control unit to implement the corresponding action of UAV. Due to simplified CNN-based model and GPU advantage of TX1, the image processing rate can reach 140fps, which is enough for control UAV.

We conducted 20 experimental tests with different condition, including different initial landing orientations, different lighting conditions and ground background. The real-time position and attitude, velocity control value and image during the UAV auto landing are recorded. We select 12 representative images from the record, which are shown in Fig.8.

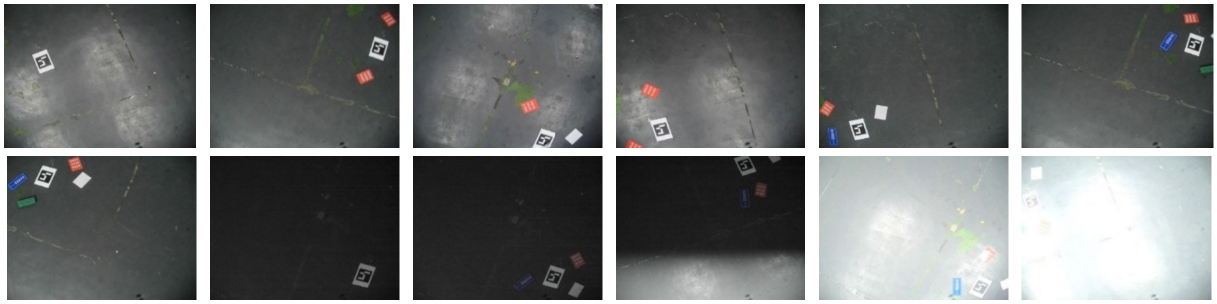
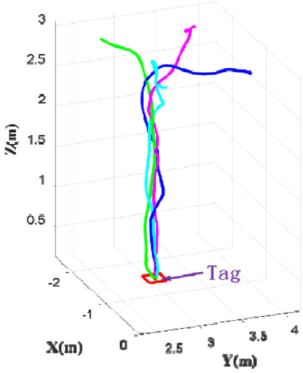


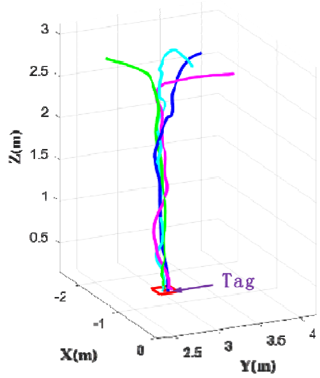
Figure 8. Selected down-looking images in the process of UAV auto landing.

The trajectory of UAV in the whole autonomous landing process in each group experiment are shown in Fig.9. Group 1-4 are the condition of moderate light and simple background; group 5-8 are the condition of dark lighting and simple background; group 9-12 are the condition of moderate light and background with interferences; group 13-16 are the condition of dark lighting and background with interferences; group 17-20 are the condition of changing lighting and background with interferences.

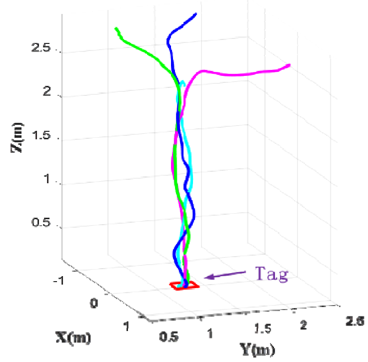
From the position trajectory of UAV in the whole autonomous landing process and statistics of errors between UAV final landing position and the central location of ground sign, we can analysis that the overall UAV auto landing effect is relatively stable, each experimental group UAV's positions eventually converge to the location of ground marker, total average errors are 4.33cm and 3.02cm in X and Y directions, note that the size of the ground sign is 16cm×16cm, the stability and accuracy of the algorithm are verified.



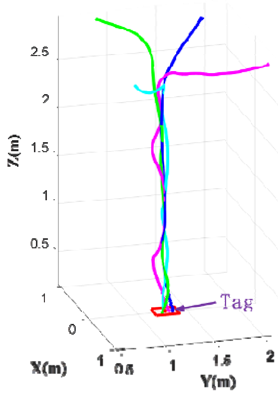
(a) Group 1-4



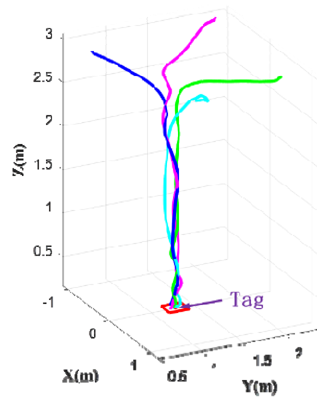
(b) Group 5-8



(c) Group 9-12



(d) Group 13-16



(e) Group 17-20

Figure 9. The trajectory of UAV in the whole autonomous landing process.

Errors between UAV final landing position and the central location of ground sign are shown in Table 1. The error of each group is very small, which demonstrates the accuracy of our deep learning model. Comparing the errors of 5 groups, we can also find there are not much difference between different groups. The robustness and environmental adaptability of our model are verified.

Table 1. Error between UAV final landing position and central location of ground tag

Group	1-4	5-8	9-12	13-16	17-20	Average
X/cm	7.04	4.43	4.54	2.81	2.85	4.33
Y/cm	2.70	2.64	2.86	4.05	2.83	3.02
Average/cm	4.87	3.54	3.70	3.43	2.84	3.68

We have two methods of velocity control for UAV landing now. One is based on position error using VICON positioning system which used in establishing dataset. While the other is an end-to-end velocity control model based on trained deep neural network. We Choose two different initial landing positions, and randomly select a set of experimental data in each method to draw the landing trajectory of UAV in Fig.10. We can see that the method using VICON positioning system tends to overshoot and disturbance in horizontal direction, and the stability is slightly worse. While the end-to-end velocity control model always maintain perfect stability and accuracy.

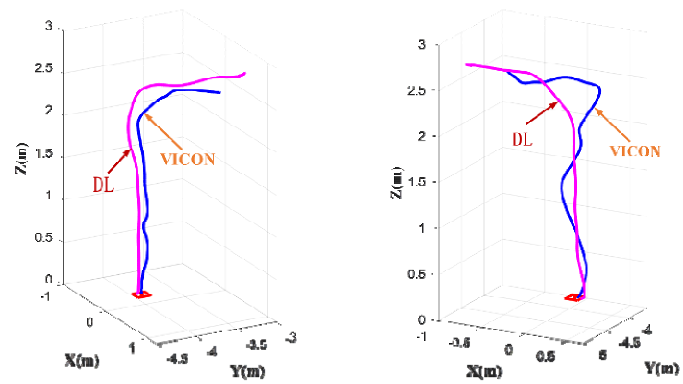


Figure 10. Landing trajectory comparison of two methods.

5. CONCLUSION

In this paper, a deep learning based end-to-end control model is proposed to solve autonomous landing problem of quadrotor. The model adopts widely-used monocular camera to capture down-looking images as sensor input to deep neural network, and outputs high-level command directly. A self-built dataset is utilized in training phase. The well-trained model is tested in different environment including indoor and outdoor. Besides, different light conditions and disturbances are added to our tests, and the results shows good robustness and adaptation of our trained model.

REFERENCES

[1] Xie, Hui, and Kin Huat Low. "Dynamic visual servoing of a rotary-wing unmanned aerial vehicle without velocity measurement." AIAA Guidance, Navigation, and Control Conference. 2017.

[2] Asl, Hamed Jabbari, and Jungwon Yoon. "Robust image-based control of the quadrotor unmanned aerial vehicle." Nonlinear Dynamics 85.3, 2035-2048, (2016).

[3] Xu, Guili, et al. "Use of land's cooperative object to estimate UAV's pose for autonomous landing." Chinese Journal of Aeronautics 26.6, 1498-1505, (2013).

[4] Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. "Reducing the dimensionality of data with neural networks." Science 313.5786, 504-507, (2006).

[5] Chen, Junjie, et al. "Identification of autonomous landing sign for unmanned aerial vehicle based on faster regions with convolutional neural network." Chinese Automation Congress (CAC), 2017. IEEE, (2017).

[6] Polvara, Riccardo, et al. "Autonomous Quadrotor Landing using Deep Reinforcement Learning." arXiv preprint arXiv:1709.03339 (2017).

[7] Zhang, Yongwei, et al. "Autonomous landing on ground target of UAV by using image-based visual servo control." Control Conference (CCC), 2017 36th Chinese. IEEE, (2017).