Journal of
Marine Science and Technology

Volume 28 | Issue 6 Article 2

Vision Based Target Recognition for Cage Aquaculture Detection

Chao-Xun Chen
Department of Communications, Navigation and Control Engineering, National Taiwan Ocean University, Keelung, Taiwan.

Jih-Gau Juang
Department of Communications, Navigation and Control Engineering, National Taiwan Ocean University, Keelung, Taiwan., jgjuang@ntou.edu.tw

Recommended Citation
Chen, Chao-Xun and Juang, Jih-Gau (2020) "Vision Based Target Recognition for Cage Aquaculture Detection," *Journal of Marine Science and Technology*: Vol. 28: Iss. 6, Article 2.
DOI: DOI:10.6119/JMST.202012_28(6).0002
Available at: https://jmstt.ntou.edu.tw/journal/vol28/iss6/2

Vision Based Target Recognition for Cage Aquaculture Detection

Acknowledgements

This research was supported by the MOST AI Biomedical Research Center at NCKU, Taiwan.

DOI: 10.6119/JMST.202012_28(6).0002

VISION BASED TARGET RECOGNITION FOR CAGE AQUACULTURE DETECTION

Chao-Xun Chen and Jih-Gau Juang

Key words: Quadcopter, target recognition, visual tracking, position correction.

ABSTRACT

This study applies Unmanned Aerial Vehicle (UAV) to net-cage fish-farming. The UAV can fly thru cages along a predefined route. At each cage, the UAV can automatically drop sensors to collect the environment data around the cage. However, low cost commercial or assembled drone might not have precise Global Navigation Satellite System (GNSS). Also, there are many uncertain factors cause Global Positional System (GPS) instability, like weather, location…etc. The main idea of this study is using GPS to guide the drone to approximated location of each cage. Then, apply image recognition to obtain the net-cage and UAV relative position. After that, the drone can utilize this information to adjust its position to the desired target. In this study we use fully assembled drone that is controlled by the Pixhawk, and the main processing platform is the Raspberry Pi3.

I. INTRODUCTION

Unmanned Aerial Vehicle (UAV) development has a long history. In recent years, drone is widely used in many works such as advertising, filming, and delivering package. There are many references discussing the motion characteristics of drone (Luukkonen, 2011; An et al., 2012) and several control theories such as Proportional-Integral-Derivative (PID) control and fuzzy control can be applied to it. Bouabdallah (Bouabdallah et al., 2004) compared the differences between PID control and linear quadratic control applications on drone. Tiep et al (2017) used fuzzy proportional-derivative control to stabilize the flight status of the drone. Raharja et al. (2014) used fuzzy control to make drone hovering more stable. Those researches make drone more stable and reliable. Therefore, researchers use more various sensors to do more delicate works. In Yan et al. (2017), the authors used SAR radar to detect indoor objects. Dixit et al. (2017) used ultrasound sensors to avoid obstacles. In recent years, more researchers have integrated image processing technology with drones. Olivares-Méndez et al. (2010) and Carreira (2013) enhanced the stability of landing by using camera. The moving target also can be tracked in the works of references (Zhao et al., 2013; Gao et al., 2014). Nowadays, the techniques of object identification and classification neural networks are well-developed. Since the advent of neural networks, many network structures are proposed to improve the accuracy of the result. In recent years, because of Deep Convolutional Neural Networks (CNN) is proposed, the complexity and diversity of the object to be tested can be increased. In Zeiler and Fergus (2014), Zeiler visualized and revealed the layers of neural networks that are helpful to understand CNN. Krizhevsky et al. (2012) proposed CNN and reached better performance than some conventional neural networks and feature recognition techniques. The CNN can effectively classify various objects by training with big data hence there are many studies about intelligent robot using the CNN to recognize the target either does drone. Bian et al. (2016) trained a human Support Vector Machine (SVM) detection model by Histogram of Oriented Gradient (HOG) features. In Tzelepi and Tefas (2017), Tzelepi proposed a novel human crowd detection using CNN to identify the heatmaps for drone flight safety purposes. Lu et al. (2018) demonstrated the use of a single forward facing camera for obstacle avoidance on a quadrotor by using a deep CNN to identify depth estimation images.

Intelligent drone with camera is more and more common not only for aerial photography but also can assist drone in many jobs. In Razinkova and Cho (2015), Barták and Vykovský (2015) the authors used camera to track ground target. In Carreira (2013) the authors used camera to correct landing position. In Jiang et al. (2017) aerial photography was applied to classify different tree species. In Abdullah et al. (2016) the authors utilized drone to detect fire. Many standardized or dangerous works can be replaced by intelligent UAV which is equipped with camera and development board. In here, we intend to collect environmental information of net-cages by the use of UAV. To collect environmental information is time consuming and tedious, and some locations cannot be reached easily. However, a drone can be equipped with sensors and help people to finish the works efficiently. Pixhawk is a common open source flight platform, it has basic sensors and cruising

Paper submitted 12/20/19; revised 02/05/20; accepted 07/03/20. Corresponding Author: Jih-Gau Juang (Email: jgjuang@ntou.edu.tw)

Department of Communications, Navigation and Control Engineering National Taiwan Ocean University, Keelung, Taiwan.

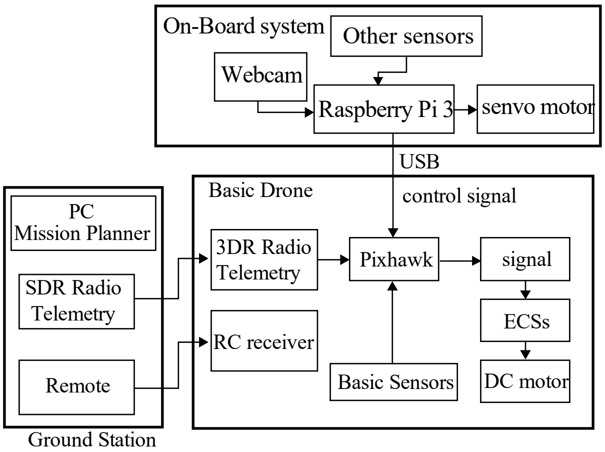


Fig. 1. Signal flowchart of the overall drone hardware

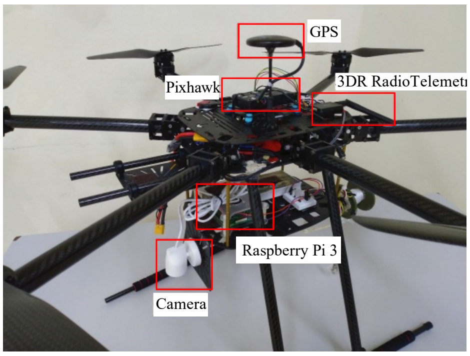
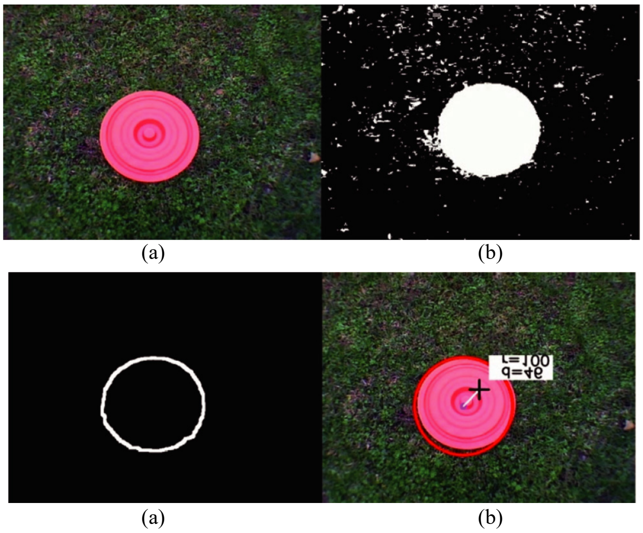


Fig. 2. Drone structure

function. But, for different environment it may cause different error. For example, when doing cruising task, it usually has 0.5m~2m error at each locations. So, we will place a simple target which is used to assist drone to correct its position at each location. To recognize the simple target, in this study, the Hue- Saturation-Value (HSV) color recognition with Hough transform circle recognition is applied. In addition, two image recognition methods are utilized for cage identification. They are HOG with SVM and CNN. These two methods can deal with complex target.

II. SYSTEM DESCRIPTION

The drone system consists of flight control board, 3DR Radio Telemetry, RC Radio, power strip of the motors, and sensors. Signal flowchart of the overall drone hardware is shown in Figure 1. The drone flight control board is the Pixhawk which is a common open source flight platform and has basic sensors. The operator can see the flight conditions, like flight altitude, flight speed, GPS, etc., through Mission Planner and send basic flight command through 3DR Radio Telemetry in ground station. In case of accident, the operator can also control the drone by the remote mode. In this study, we used a six-axis rotorcraft, also known as hexacopter, to perform the proposed task. The hexacopter can carry more devices than quadcopter. Figure 2 is the main body of the drone.


(a) (b)

(a) (b)

Fig. 3. (a) outdoor red target (b) mask image (c) canny edge detection (d) detect red circle target

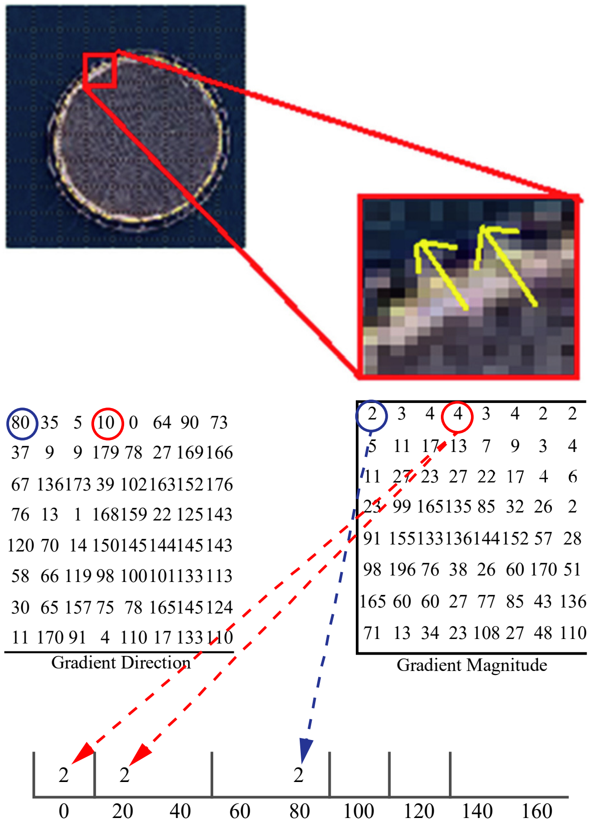
Although the operator can control the drone by the remote control mode or ground station. In order to finish the work automatically an on-board processor is needed. In this study, Raspberry Pi 3 is chosen to be the main processing core because of its lightweight and diverse functions. The operating system is an open source Linux system, by using python with dronekit module, Raspberry Pi 3 can be connected to the Pixhawk and peripheral devices easily. It can receive various parameters from the drone and assist the program in making judgments. All these systems are helpful to the automation of drone operation. The operator can also control the drone from ground station in case there is any error in the program.

III. IMAGE PROCESSING

In this study, we use two kinds of image target. First is a simple circular object of a specific color and the second is the complex net-cage of fishing farm.

1. Hough Transform

To detect a simple object on the captured picture, we transfer the image from Red-Green-Blue (RGB) color space to HSV color space. Because drone usually works in outdoor, the effect of light on the HSV color space images is lower than RGB color space images. After that, put a mask of a specific color on the image. Then, the region of the specific color on the image is detected. To distinguish whether the region is circular or not, we use Hough transformation circle detection. Hough transformation can calculate an object which has a certain class of shape by a voting procedure. Hough transformation usually is used to detect line on image. By adjusting the function, we can detect circle on the captured image



80	35	5	10	0	64	90	73
37	9	9	179	78	27	169	166
67	136	173	39	102	163	152	176
76	13	1	168	159	22	125	143
120	70	14	150	145	144	145	143
58	66	119	98	100	101	133	113
30	65	157	75	78	165	145	124
11	170	91	4	110	17	133	110

Gradient Direction

2	3	4	4	3	4	2	2
5	11	17	13	7	9	3	4
11	27	23	27	22	17	4	6
23	99	165	135	85	32	26	2
91	155	133	136	144	152	57	28
98	196	76	38	26	60	170	51
165	60	60	27	77	85	43	136
71	13	34	23	108	27	48	110

Gradient Magnitude

Fig. 4. Gradient of a cell

Table 1. HOG features with different HOG parameters

model	pixel per cell	cell per block	HOG features
1	10	2	6108
2	8	2	9564
3	6	2	16368

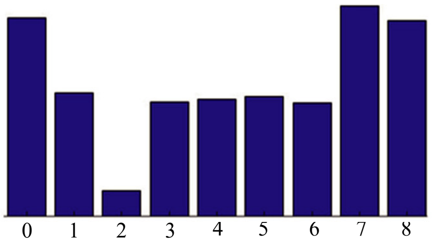


Fig.5. Histogram of a cell

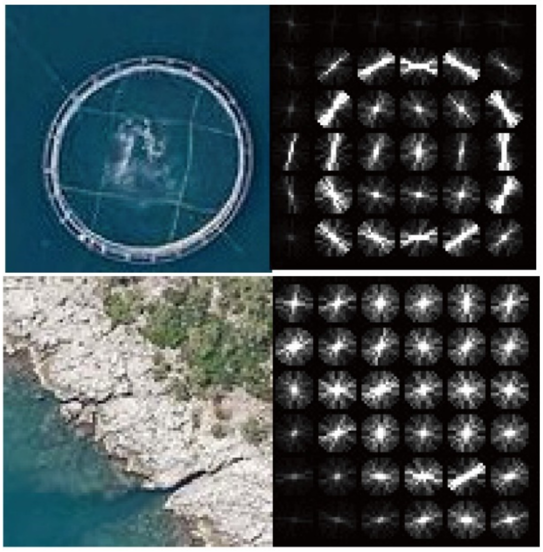


Fig. 6. Positive cage samples and the negative samples by HOG vector visualization

The main idea of Hough transform is to map the feature points in the two-dimension image space into the two or more high dimension Hough transform space. Next, by a finding-maximum procedure, determine what shape and its position is existed in the original image. Hough transform space coordinate is based on the model's transformation parameters which decide the shape of object. According to the Hough transform rule, in order to detect the circle in the image, use the circle function $(x\text{-a})^2+(y\text{-b})^2=r^2$ to be model's transformation parameter. In other words, the feature points in the image are mapped to Hough transform space which is a three-dimension space and its coordinate is (*a, b, r*). The center of circle in the image is represented by (*a, b*), and *r* is the radius of circle. The Hough transform with the image preprocessing has adaptability to handle different light source and noise. Finally, the specific single color circle can be detected, and by changing color mask it can detect other colors as well. Figure 3 shows the result of a red target. In the figures, *r* and *d* are radius of circle and distance between circle center and image center in pixel, respectively. After getting the location of target, we can take this information as reference and control the drone's position.

2. Histogram of oriented gradient (HOG)

In Olivares-Méndez et al. (2010), the authors used drone to detect and track human. If the target is in a regular shape, we can use HOG and SVM to train and identify this target. In this study the target is the fishing cage. HOG is used to extract images to a vector Ballard (1981). We collect 200 fish farming cage pictures. By resizing, rotating, and blurring, the number of images of our data base can be increased. The size of our samples is 80x80. By adjusting parameter of HOG, we can get different length of vector. The length of vector might affect training time and accuracy.

The first step of HOG vector calculation is the gradient computation. By filtering the color or intensity data of the image with the following filter kernels: [-1, 0, 1] and $[-1, 0, 1]^T$, we can get the *x*, *y* direction gradient of the image, as shown in Figure 4. In second step, the image is divided into several small cells (such as 6*6 pixels/cell). As mentioned above, every pixel has its own gradient within 0 to 180 degrees or 0 to 360 degrees, and is depending on whether the gradient is

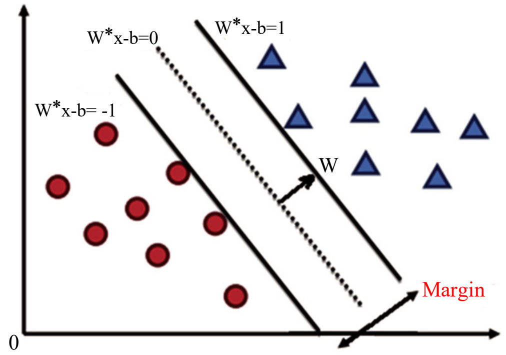


Fig. 7. Use SVM find two dimensions optimal hyperplane

"unsigned" or "signed". To create the histogram of a cell, the gradient within 0 to 180 degrees is divided into 9 parts. Each pixel within the cell votes for the section based on its own gradient to become the histogram of this cell, as shown in Figure 5. Then defining several connected cells as one block and concatenating the histogram of all cells in the block, the HOG feature vector of this block can be obtained. These blocks are overlapped. This means the characteristics of each cell appear multiple times in the vector with different type. At last, an image is converted into a feature vector that can be recognized. In HOG, "pixel per cell" and "cell per block" are the main factors deciding the length of the feature vector. Three models are shown in Table 1. Extract the HOG features of the positive samples and the negative samples respectively, as shown in Figure 6. The calculation time and accuracy of the SVM may be affected according to the different length of the HOG feature vector. Therefore, we use different HOG feature vectors to train the SVM and find the more suitable parameters which can be applied to this study.

3. Support Vector Machine (SVM)

SVM is used as a classification model. Given a set of HOG vector training samples it can find the optimal hyperplane that classify the positive samples and the negative samples. For the classifier, the data is the P-dimensional vector and the classifier can separate the data with the P-1 hyperplane, and there is not only one hyperplane can do it. SVM can find the optimal hyperplane which minimizes the margin. In addition to linear classification, SVM can efficiently perform a nonlinear classification using what is called the kernel trick, mapping the inputs into high-dimensional feature spaces. The following is taking linear separable classification for example. There are n points in the training data, ($\overrightarrow{x_1}$, y_1), ($\overrightarrow{x_2}$, y_2) … ($\overrightarrow{x_n}$, y_n). $\overrightarrow{x_i}$ is the data feature vector, y_i is 1 or -1 which represents the label of this data. Supposing that $\overrightarrow{w}\cdot\overrightarrow{x}$-$b$=0 is any hyperplane that can classify the training data. For each i in the data at least satisfy one of the following conditions:

$$\overrightarrow{W}\cdot\overline{x_i}-b\geq 1,\quad if\ y_i=1 \tag{1}$$

Table 2. Kernels of SVM

Linear kernel	K(x,y)=x^{T}y
Polynomial kernel	K(x,y)=(x^{T}y)^d
RBF kernel	K(x,y)=exp($-\gamma\|\|x-y\|\|^2$)
Sigmoid kernel	K(x,y)=tanh(ax^{T}y+b)

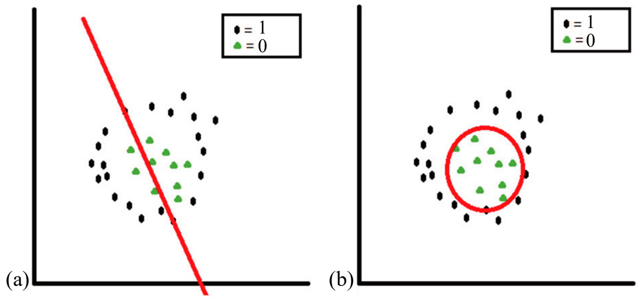


Fig. 8. (a) linear kernel (b) RBF kernel for nonlinear separation

$$\overrightarrow{W}\cdot\overline{x_i}-b\leq 1,\quad if\ y_i=-1 \tag{2}$$

The first condition can be written as $y_i(\overrightarrow{w}\cdot\overrightarrow{x_i}$-$b)\geq 1$, for i=1,…,n.

There are infinite hyperplanes which are satisfied this condition in this space. To find the optimal hyperplane, we can select two parallel hyperplanes that separate the two classes of data ($\overrightarrow{w}\cdot\overrightarrow{x}$-$b$=1, $\overrightarrow{w}\cdot\overrightarrow{x}$-$b$=-1), so that the distance between them is as large as possible. The region bounded by these two hyperplanes is $\frac{2}{||\overline{w}||}$ and called the "margin". The maximum-margin hyperplane is the hyperplane that lies halfway between them. For two dimensions point, we need minimize $\frac{1}{2}||w||^2$ which subject to

$$y_i\left(w^T x_i-b\right)-1\geq 0\ \forall i, \tag{3}$$

as shown in Figure 7 (Hsu et al., 2013).

The original SVM algorithm is used to solve linearly separable question. However, in 1992, Bernhard E. Boser, Isabelle M. Guyon and Vladimir N. Vapnik proposed a method that replaced dot product with kernel function to create the nonlinear classifiers (Boser et al., 1992). This allows the algorithm to find the maximum-margin hyperplane in a transformed feature space. Although the data in the original input space may be nonlinear, by mapping to the high-dimensional space, the classifier can be found in transformed feature space. Some common kernels include: linear kernel, polynomial kernel, Gaussian radial basis function (RBF), sigmoid kernel, as shown in Table 2. The choice of the kernel function will be based on the number of features of the sample or the amount of training data. In general, the RBF kernel is a reasonable first choice. This kernel nonlinearly maps samples into a higher dimensional space, unlike the linear kernel, it can

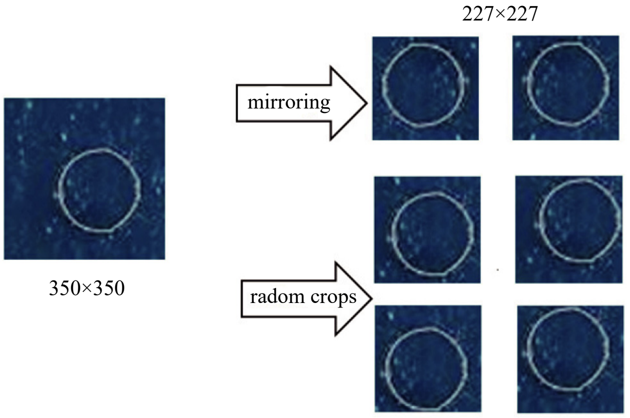


Fig. 9. Data augmentation

handle the case when the relation between class labels and attributes is nonlinear (Hsu et al., 2013), as shown in Figure 8.

4. Convolutional Neural Network

In recent years, there are many deep neural networks, such as: Region-CNN (R-CNN), Fast R-CNN, Faster R-CNN, You Only Look Once (YOLO), and Single Shot MultiBox Detector (SSD), are used in object classification. However, those neural networks take a long time to train and need complicated parameter adjustment. In this study, the objects to be recognized are just the cage and background, and we need to reduce computing time. Therefore, AlexNet which has lower layer neural network is selected as a reference model in this study. AlexNet provides few rules to improve training results and reduce overfitting. First is changing activation function. The most commonly used activation functions at the time were sigmoid and tanh functions. Sigmoid function compresses the input value between 0 and 1, but its disadvantage of gradient disappearing makes it difficult to optimize the neural network. ReLU function's slope is not close to zero for higher positive values. This helps the optimization to converge faster (Krizhevsky et al., 2012).

Table 3. AlexNet architecture

Layer	Dimension/Kernel	Activation
Input	227x227x3	
Convolution 1	11x11x3, 96	ReLU
Max Pooling		
Convolution 2	5x5x48, 248	ReLU
Max Pooling		
Convolution3	3x3x256, 384	ReLU
Convolution4	3x3x192, 384	ReLU
Convolution5	3x3x192, 256	ReLU
Max Pooling		
Fully Connect 6	4096	ReLU
Fully Connect 7	4096	ReLU
Output	2	softmax

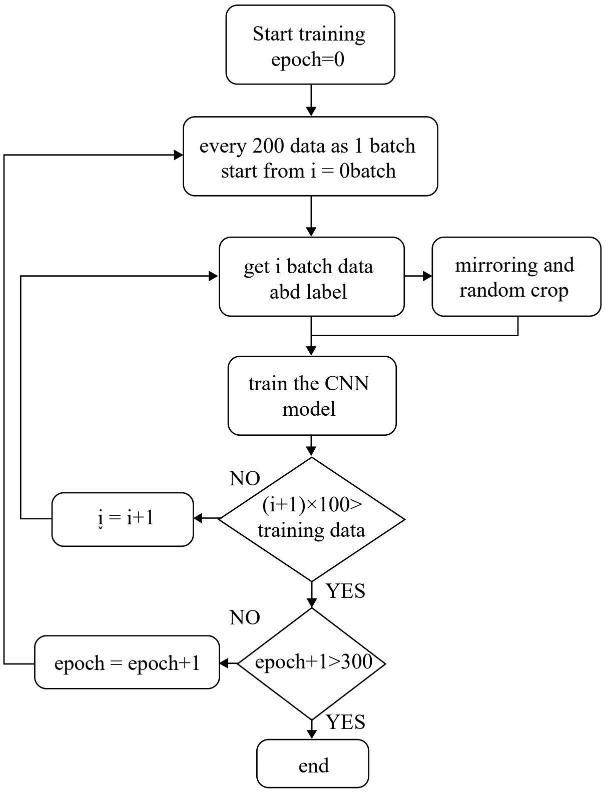


Fig. 10. CNN training flowchart

The size of the neural network is its capacity to learn. The neural network will try to understand the examples in the training data. As a result, the neural network may work well on the training data, but they fail to learn the similar object. In order to reducing overfitting, the authors of AlexNet use two kinds of methods. First is data augmentation, doing data augmentation on training data by mirroring and random crops. Generally, human figures or animals usually only use horizontal mirroring, and the shape of the cage is relatively fixed. Therefore, vertical mirroring is also used in this study. Random crops is that extracting random crops of size 227×227 from inside the original image boundary. These two ways add similar but not exactly the same data in the training data, a cage image is shown in Figure 9. The second way to reduce overfitting is dropout. With about 60M parameters to train, during the training process, the network will disconnect a certain network link and make its weight zero. The probability of dropped from the network that AlexNet set is 0.5. Those neurons will not contribute to either forward or backward propagation. Force some neurons to train the current data. It can increase network robustness and reduce overfitting.

AlexNet consists of 5 convolutional layers and 3 fully connected layers and there are three overlapping max pooling layers behind the convolutional layer 1, 2 and 5 (Nayak, Understanding AlexNet). Multiple convolutional layers extract

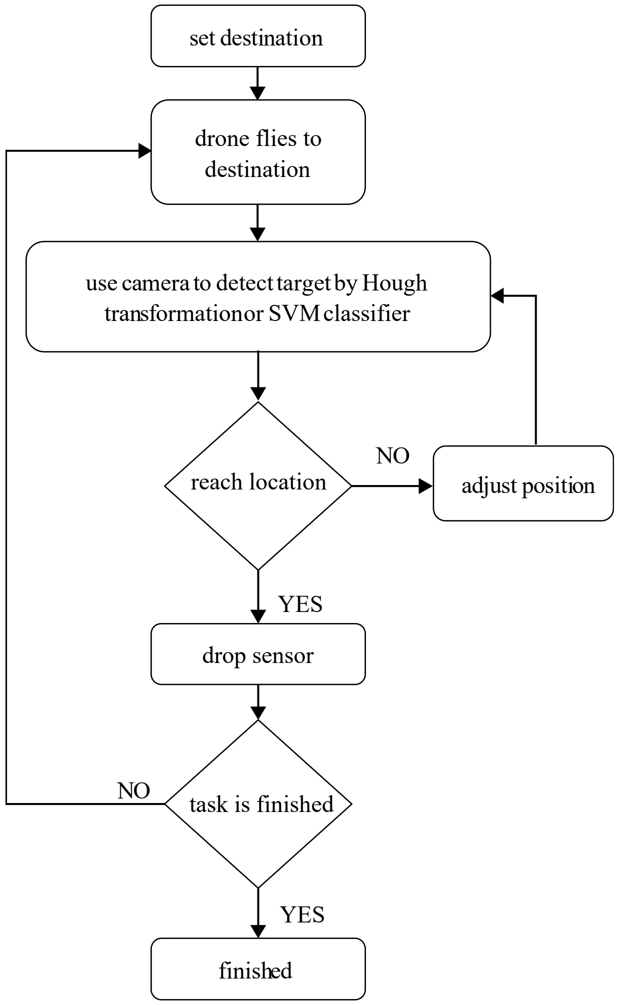

set destination
drone flies to destination
use camera to detect target by Hough transformationor SVM classifier
reach location
NO
adjust position
YES
drop sensor
task is finished
NO
YES
finished

Fig. 11. Flowchart of cruising targets

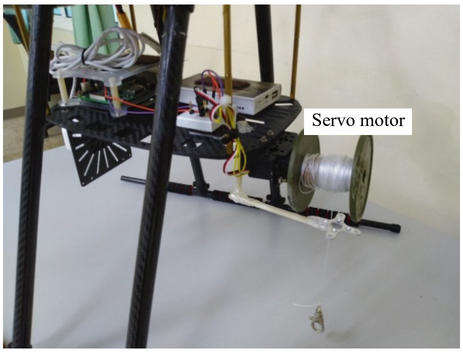


Fig. 12. 360 degrees servo motor

features in an image, by using many kernels of the same size to increase the depth of data. For 96 kernels in convolutional layer 1 and its size is 11x11x3. After convolution, an image which size is 227x227x3 is transferred into 55x55x96 data. Max Pooling layers down samples the width and height of the data but keeps the depth same. The last is two traditional fully connected neural network layers which are used as a classifier, as shown in Table 3.

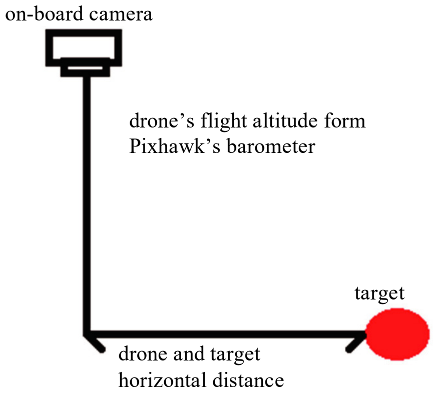

on-board camera
drone’s flight altitude form Pixhawk’s barometer
target
drone and target horizontal distance

Fig. 13. Distance of the drone and target

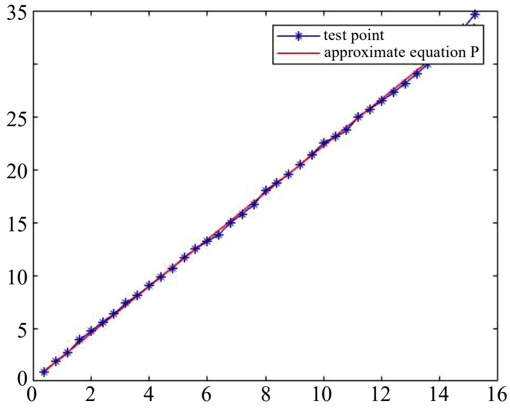


Fig. 14. Transfer function of the real distance and pixel

The original AlexNet is used to classify 1000 types of object so the last softmax classifier has 1000 class labels. In this study, we only need to classify cage and background, therefore, softmax classifier is reduced to 2 types. During the training process, every 200 of data in the training data as a batch and a certain proportion of the pictures are cut and added to the batch for training. Training data will be the same as SVM. We also adjusted some of CNN's parameters in order to improve accuracy and reduce computation time and compared with SVM results. The training flowchart is shown in Figure 10.

IV. CONTROL SCHEME

1. Mission Process

The flowchart of the experiment is shown in Figure 11. The drone is applied to cruise several fishing cages by GPS system. At each destination, we set a ground target or just identify the fishing cage to adjust drone’s position. While adjusting is done, the drone uses 360 ° servo motor to drop the sensor, as shown in Figure 12. After that, drone goes to the next destination until the task is finished. All steps are automatically controlled by Raspberry Pi 3 on the drone.

Table 4. Fixed error

number of times	GPS error(cm)	fixed error(cm)
1	103	33
2	140	25
3	194	52
4	112	80
5	228	65
6	138	40
7	210	15
8	123	20
9	163	45
10	207	75
average error	161.8	45

2. Image Tracking and Drone Position Control

While drone reaches approximated location by GPS, the on-board camera will start to detect the target.

(a) Distance transformation

Before the mission, we need to get the transfer function between the real distance and pixel in the image. We set a camera on a fixed position and put the same object in front of the camera every 40cm and record the object's pixels in the image. Then, we can get the following transfer function

$$P=2.2151*H+0.0903 \tag{4}$$

H is drone's flight altitude (m) which is obtained from Pixhawk, and P is the real distance (cm) of every 10 pixels in the image on current height. By this function, we can get the real horizontal distance of the drone and target from image [9], as shown in Figure 13 and Figure 14.

(b) Position control

While the drone reaches approximated location by GPS, the on-board camera will start to detect the target. Consider the center of image as drone itself. When the target is detected by camera, we can get the pixels which is the difference of x and y direction between the target and the center of image. Using (4), we can compute the real distance between the drone and target. We can control one position by fixed process in 3 seconds regardless of distance between the drone and target. According to x and y direction difference, we can give x and y direction velocity to control the drone to fix its position and fly on right angle.

Commercial GPS usually has certain error, the GPS used in this experiment has an average error of 1.67m. We use the same way to test GPS error. At first set a path then let drone cruise and return to the starting point. Before the drone takes off, we set a mark at start point and the drone will record the GPS position of starting point to be the end point after drone's cruising. Therefore, when the drone finished cruising mission it will return to the starting point. The difference between the mark and the drone landing position is the GPS error. A position adjustment program is applied to fix drone position when the drone reaches ending point. Although there is still error in the conversion of the distance and the movement of the drone, as shown in Table 4, the position adjustment program can reduce the average error from 161cm to 45cm.

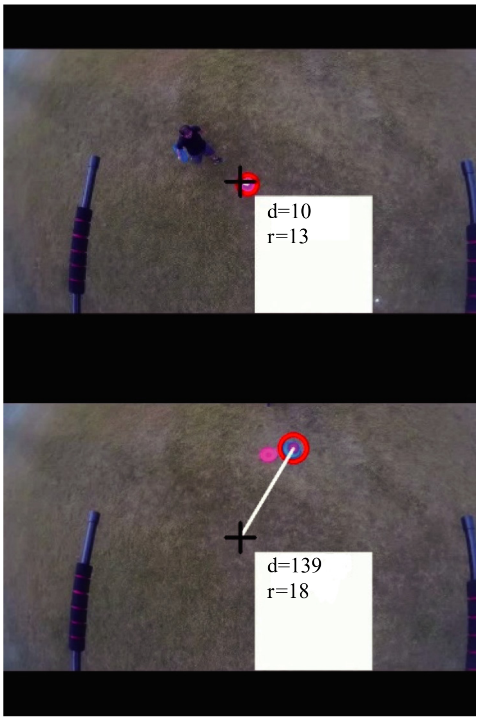


Fig. 15. Get different color targets and compute distance and radius in pixel number

V. EXPERIMENT RESULTS

By the use of HSV color space to recognize the target color, we can adapt to the intensity of different light. As long as the pixel of target's radius in the image is larger than 10 pixels, the target can be detected. According to the pixel to real distance transformation, we can set up a pixel value. If the distance of target and image center is smaller than this value, it represents the drone is close enough to the target. Then, we can control servo motor to drop the sensors. In addition, by setting different mask parameter, we can catch different color target to perform other mission, as shown in Figure 15. In Figure 15, ***d*** is the distance between the target and the center of image, ***r*** is target radius in pixels. The first figure shows a red target is tracked. The distance between the red target and the UAV is 10 pixels and the size of the target (radius) is 13 pixels. In the second figure, a blue target is tracked. The distance is 139 pixels and the size is 18 pixel.

Table 5. SVM identification result

Model	Kernel function	Training data size	HOG features	pos : neg samples	Accuracy
1	Gaussian	80x80	9564	2:1	96%
2	linear	80x80	9564	1:1	91%
3	linear	80x80	9564	2:1	68%
4	linear	80x80	9564	1:2	92%
5	linear	28x28	1284	1:2	87%
6	linear	40x40	2544	1:2	88%
7	linear	64x64	6108	1:2	92%

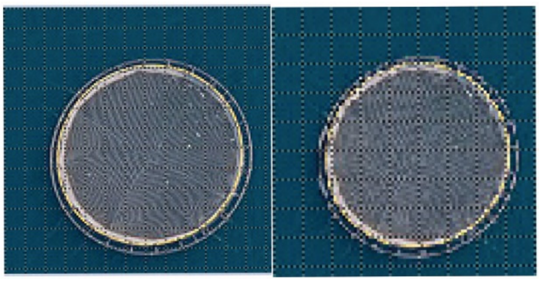


Fig. 16. Use diffierent HOG blocks can get diffierent length of vector.

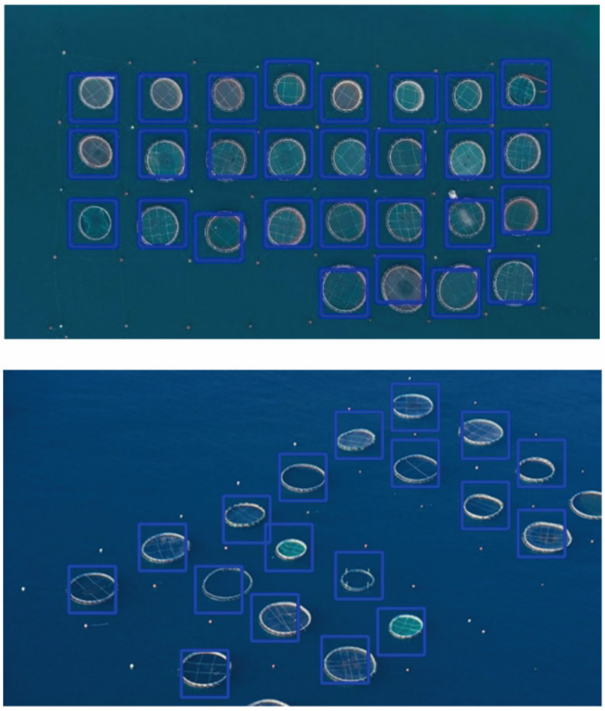
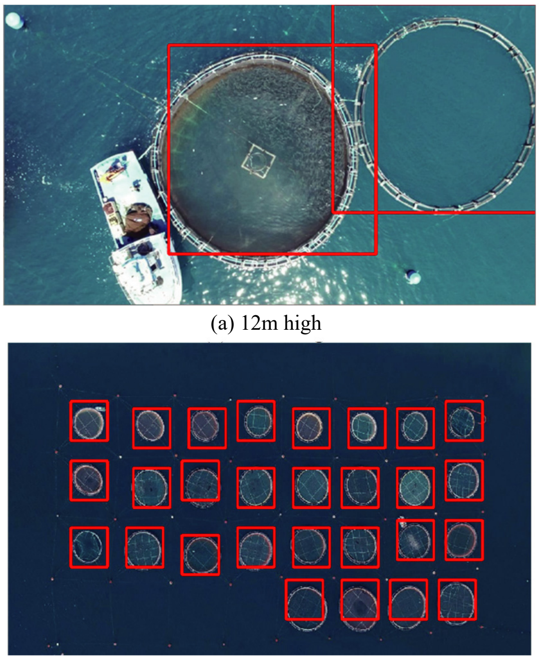


Fig. 17. Use SVM to detect fishing cages

Besides setting an obviously target near the desired cage to fix drone's position, we can use the shape of cage itself as well. As long as the destinations have regular or similar shape, we can use HOG and SVM to identify them, as shown in Figure 16 and Figure 17. In this study, we consider fishing cages which have the features as mentioned above. In the experiment, accuracy of SVM classification is concerned. By adjusting the numbers of the positive samples and the negative samples and length of HOG feature vector, we compare accuracy of 7 models. Increasing the length of HOG feature vector helps improve accuracy but also reduces calculation speed.



(a) 12m high

(b) 40m high

Fig. 18. Different sliding window sizes (a) 500x500 (b)100x100

For linear kernel, in the case that longer HOG feature vector with the negative sample is more than the positive sample will have better recognition results, as shown in Table 5. Reducing the feature vector also has an impact on accuracy. The use of Gaussian functions can increase the accuracy but greatly increase the computation time. In addition, the length of the feature vector length will also affect the computation time.

The size of the cage displayed in the image will change with the distance between the camera and the cage. In other words, the same cage at different flight altitudes shows different pixels in the picture. Therefore, different sliding window sizes

Table 6. SVM identification result – computing time (second)

processor Model	PC		Raspberry Pi		Accuracy
	100x100	500x500	100x100	500x500	
1	46.0	3.8	152.2	26.3	96%
2	7.1	0.5	47.3	3.3	91%
3	7.8	0.5	47.6	3.3	68%
4	6.9	0.7	47.2	3.3	92%
5	1.4	0.1	7.3	0.5	87%
6	2.5	0.2	13.4	0.9	88%
7	5.2	0.4	31.0	2.2	92%

Table 7. CNN identification result – accuracy

Model	training sample size	learning rate	epochs	batch	Accuracy
1	227x227	0.0001	600	200	95%
2	150x150	0.0001	600	200	92%
3	80x80	0.0001	600	200	93%

Table 8. CNN identification result – computing time (second)

Model	PC		Raspberry Pi		Accuracy
	100x100	500x500	100x100	500x500	
1	21.0	1.1	X	13.7	95%
2	6.1	0.40	X	8.6	92%
3	1.4	0.14	20.0	2.0	93%

are required in different situations, as shown in Figure 18. Because the size of the input data of the trained SVM module is based on the size of the training data, therefore, the data obtained through the sliding window must be resized to be the input data of the SVM module. The selected kernel function and the length of the HOG feature vectors also affect the computational complexity. All of the above cases are the main factors affecting the calculation time. Computation time on the PC and Raspberry Pi are shown in Table 6.

Performance of the CNN network is tested by changing the training sample size. The sample size that AlexNet recommends is 277x277. We also use two other sizes for comparison, the 3 models are given in Table 7. The size of the training sample will change the size of the parameters sent into the fully connected layer after passing through the convolution layer and it also affects the calculation and computing time. The training data used in the training process is the same as the SVM, during the training, each batch will increase the amount of data by data augmentation. As shown in the results, the training sample size has a smaller impact on the CNN accuracy than the SVM.

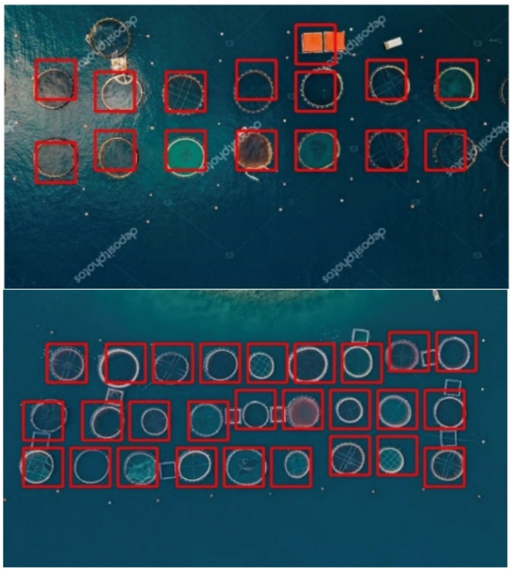


Fig. 19. CNN identification result

Like SVM, CNN must adjust the input data to the same size as the training data when inputting data. In addition, the sliding window is also used to capture the image as input data of the CNN. Therefore, the sliding window size and the required data size will be the main factors affecting the computing time. Test of the computing time on the PC and Raspberry Pi is shown in Table 8. Compared with SVM, CNN calculation is more complicated. When the input data is large, the Raspberry Pi hardware device cannot be loaded. However, as shown in the results, reducing the training sample size can effectively reduce the calculation time and has little effect on the accuracy. Identification results are shown in Figure 19.

At aquaculture farm, a red target is placed next to the fish pond. If the drone recognizes the target, it will fly to the fish pond along the direction and drop the sensor, as shown in Figure 20. The cruise mission is to obtain the environment information of the cage. The temperature data is obtained by the carry-on sensor, as shown in Figure 21. The sensor is RBRconcerto[3], a multiparameter water quality probe, multi-channel logger supports numerous sensors, offers flexible measurement

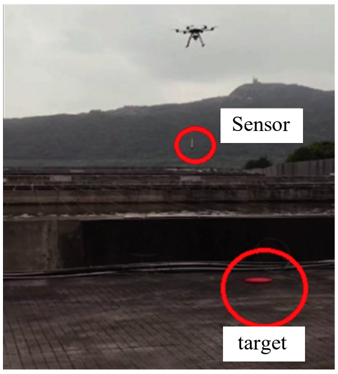


Fig. 20. Drone and target

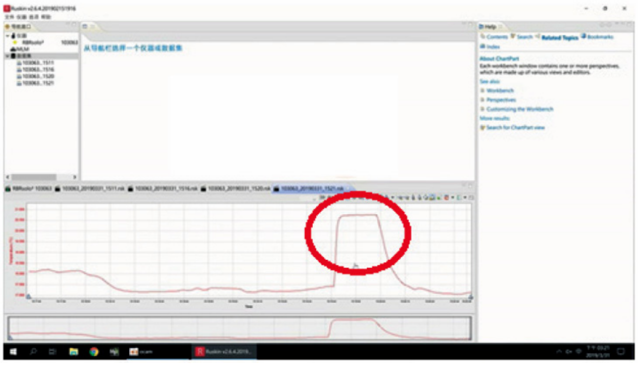


Fig. 21. Water temperature measured by the drone

schedules, standard sampling is up to 2Hz, optionally up to 32Hz, large memory, ample power for extended deployments, USB-C download for large data sets, and twist activation.

VI. CONCLUSIONS

This study proposes a control scheme to solve a cumbersome problem of environmental data collection and cage detection in the aquaculture by using intelligent drone. The drone is integrated with Raspberry Pi, camera, GPS, sensor, and servo motor to increase drone autonomy. Compared to commercial drones, we can adjust equipment and parameters as we want. The drone is guided by GPS. The GPS we used has an average error of 1.65 meters, and the error can be reduced to 0.45 meters after using the fix error program. After arriving at the destination, the drone uses a carry on camera to recognize the specified target and fine-tune its position to get to the destination accurately. The CNN, SVM, and HOG are successfully applied to image processing, and the captured target image can be used to adjust the drone's position. Experiment shows that the purpose of collecting environmental data from the fish farm is achieved by the sensor mounted on the servo motor of the drone.

ACKNOWLEDGEMENT

This research was supported by the MOST AI Biomedical Research Center at NCKU, Taiwan.

REFERENCES

Abdullah, M. F., I. Wijayanto and A. Rusdinar (2016). Position estimation and fire detection based on digital video color space for autonomous quadcopter using odroid XU4. International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC).

An, H., J. Li, H. Ma, and K. Chen (2012). Quaternion based geometric control of quadrotor UAV. 2012 24th Chinese Control and Decision Conference (CCDC), 2012.

B. E. Boser, I. M. Guyon and V. N. Vapnik (1992). A training algorithm for optimal margin classiers. Proceeding COLT '92 Proceedings of the fifth annual workshop on Computational learning theory, 144-152.

Ballard, D. H. (1981). Generalizing the hough transform to detect arbitrary shapes. Pattern Recognition 13(2), 111.

Barták, R. and A. Vykovský (2015). Any object tracking and following by a flying drone. Fourteenth Mexican International Conference on Artificial Intelligence.

Bian, C., Z. Yang, T. Zhang and H. Xiong (2016). Pedestrian tracking from an unmanned aerial vehicle. 2016 IEEE 13th International Conference on Signal Processing (ICSP).

Bouabdallah, S., A. Noth and R. Siegwart (2004). PID vs LQ control techniques applied to an indoor micro quadrotor. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems.

Carreira, T. G. (2013). Quadcopter automatic landing on a docking station. Semantic Scholar.

Carreira, T. G. (2013). Quadcopter automatic landing on a docking station. Master Theses, Instituto Superior Tecnico.

Dixit, K. R., P. P. Krishna and R. Antony (2017). Design and development of H frame quadcopter for control system with obstacle detection using ultrasound sensors. 2017 International Conference on Circuits, Controls, and Communications (CCUBE).

Gao, Q., Z. C. Zeng and D. Hu (2014). Long-term tracking method on ground moving target of UAV. Proceedings of 2014 IEEE Chinese Guidance, Navigation and Control Conference.

Hsu, C. W., C. C. Chang and C. J. Lin (2013). A practical guide to support vector classification. technical report. Department of Computer Science and Information Engineering, University of National Taiwan, Taipei, 1-12.

Jiang, H., S. Chen, D. Li, C. Wang and J. Yang (2017). Papaya tree detection with UAV images using a GPU-accelerated scale-space filtering method. Remote Sens 9, 721.

Krizhevsky, A., I. Sutskever and G. E. Hinton (2012). Image net classification with deep convolutional neural networks. Advances in Neural Information Processing Systems 25 (NIPS 2012).

Lu, Y., Z. Wang, Z. Tang and T. Javidi (2018). Target localization with drones using mobile CNNs. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

Luukkonen, T. (2011). Modelling and Control of Quadcopter. Mat-2.4108 Independent Research Project In Applied Mathematics.

Olivares-Méndez, M. A., I. F. Mondragón, P. Campoy and C. Martínez (2010). Fuzzy controller for UAV-landing task using 3D-position visual estimation. International Conference on Fuzzy Systems.

Raharja, N. M., I. M. Faris and A. I. Cahyadi (2014). Hover position quadrotor control with fuzzy logic. 2014 The 1st International Conference on Information Technology, Computer and Electrical Engineering.

Razinkova, A. and H. C. Cho (2015). Tracking a moving ground object using quadcopter UAV in a presence of noise. Advanced Intelligent Mechatronics (AIM), July 7-11.

S. Nayak, Understanding AlexNet, https://www.learnopencv.com/understanding-alexnet/Tiep, D. K., K. Lee, Y. J. Ryoo and S. J. Kim (2017). A fuzzy-PD controller for an autonomous aerial robot. 2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI).

Tzelepi, M. and A. Tefas (2017). Human crowd detection for drone flight safety using convolutional neural networks. 2017 25th European Signal Processing Conference (EUSIPCO).

Yan, J., Z. Peng, H. Hong, X. Zhu, Q. Lu, B. Ren and C. Li (2017). Indoor range-direction-movement SAR for drone-based radar systems. 2017 IEEE Asia Pacific Microwave Conference (APMC).

Zeiler, M. D. and R. Fergus (2014). Visualizing and understanding convolutional networks. ECCV 2014, Part I, LNCS 8689, 818–833.

Zhao, X., Q. Fei and Q. Geng (2013). Vision based ground target tracking for rotor UAV. 2013 10th IEEE International Conference on Control and Automation (ICCA).