IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 21, 2021, accepted September 8, 2021, date of publication September 13, 2021, date of current version September 29, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3112185

Vision-Assisted Landing Method for Unmanned Powered Parachute Vehicle Based on Lightweight Neural Network

MENGXUAN ZHANG[1], WEI HU[2], SHUDE JI[1], QI SONG[2], PENG GONG[1], AND LINGPEI KONG[1]

[1]College of Aerospace Engineering, Shenyang Aerospace University, Shenyang 110136, China
[2]College of Automation, Shenyang Aerospace University, Shenyang 110136, China

Corresponding author: Shude Ji (superjsd@163.com)

This work was supported in part by Liaoning BaiQianWan Talents Program under Grant 2020921030, and in part by the Education Department Foundation of Liaoning Province under Grant JYT2020074.

ABSTRACT With the development of airdrop technology, the intelligence degree of unmanned powered parachute vehicles (UPPVs) need to be improved. To achieve the accurate landing of UPPVs in complex environments, a landing runway recognition model based on a deep learning algorithm is trained and five actual flight tests are conducted. A six-degree-of-freedom (6-DOF) mathematical model of an unmanned powered parachute vehicle is established, and a landing runway offset controller is designed. The lightweight landing runway recognition model was trained by combining the YOLOv4 framework and the lightweight neural network MobileNet-V3 (Large) and validated in various scenarios. The runway recognition model was transplanted into the airborne image processor, and an unmanned powered parachute vehicle test platform was built for actual flight testing. The test results showed that the comprehensive accuracy of the runway recognition was 97.81% during visual landing and the offset correction was completed within 15s.

INDEX TERMS Unmanned powered parachute vehicle, visual landing, YOLOv4, lightweight neural network, offset controller.

I. INTRODUCTION

Unmanned powered parachute vehicles (UPPVs) are a new type of unmanned aerial vehicle based on manned powered parachutes. It is a complement to low-altitude and ultra-low-altitude flight operations in mission areas where manned aircraft, fixed-wing UAVs and unmanned helicopters cannot reach. Owing to its large payload, long endurance and simple operation, UPPV is considered safer and longer-range than the UAV. With the rapid development of AI technology, UPPV can be used in military and civil fields in the future [1], [2]. For example, equipment and supplies can be accurately delivered to the target location over a long distance, especially for large-scale disaster rescue and escort missions [3]. Critical for such missions is the autonomous and safe landing of the UPPV. However, there are few studies on the terminal homing and landing of parachutes. Santoso *et al.* [4] used neural network to predict the landing of parachute at the desired point. Murali *et al.* [5] used a line-of-sight guidance algorithm to guide a parachute to the specified target. As powered parachute vehicles work in increasingly diverse scenarios, it is difficult to achieve accurate landing in unknown environments.

Traditional navigation technologies such as GPS and INS cannot meet the navigation requirements of the landing stage owing to the influence of precision and disturbance. Visual navigation has the advantages of strong anti-interference ability, high precision, low cost and low power consumption, and is widely applied to UAV navigation especially in the autonomous landing phase [6]. For an unmanned flight platform, the fast and accurate identification of its landing runway is key for autonomous landing. Nazir *et al.* [7] acquired landing runway images using an airborne camera, and estimated the exact location of the landing runway using edge detection algorithms. Fadhil *et al.* [8] proposed a sensor fusion algorithm based on the Hough transform to detect runways. Zhang *et al.* [9] designed a fuzzy Canny edge extraction algorithm and combined it with a curve-fitting method to

The associate editor coordinating the review of this manuscript and approving it for publication was Amin Zehtabian.

obtain the centerline of the runway. The above-mentioned methods belong to the traditional runway identification methods [10]–[12] which are vulnerable to environmental changes and poor recognition accuracy despite their simple calculation.

Compared with traditional identification methods, deep learning methods can automatically learn the target features based on the existing data which has better adaptability and universal applicability [13]–[15]. The convolutional neural network (CNN) is most widely used in the field of target recognition [16]. Zhang *et al.* [17] and Xiao *et al.* [14] used a CNN to extract runway information but they did not port the algorithm to the mobile device platform for real experiments. Although this algorithm has high detection accuracy, the division of the target recognition task into multiple processes such as target area prediction and category prediction leads to a slower detection speed. You Only Look Once (YOLO) [18] is the first single-stage detection method, which is faster and better suited for mobile applications. Cintas *et al.* [19] propoesd the YOLOv3 network model to an airborne processor to detect and track flight targets. Chen *et al.* [20] used a UAV to monitor pests using a Tiny-YOLOv3 neural network model built on an embedded system. Considering the limited computing power of the UPPV's image processor Raspberry Pi, we replaced the backbone with a lightweight neural network Mobilenet-V3 [21] to simultaneously promote detection speed and accuracy.

In this paper, we propose the application of vision techniques to the autonomous landing of an unmanned powered parachute vehicle (considering only daytime work). To meet the lightweight requirement for use on mobile devices, we chose the lightweight network Mobilenet-V3 (Large) to replace the underlying network structure CSPDarkNet-53 in the YOLOv4 algorithm. In addition, we designed a runway offset controller based on a powered parachute vehicle model. Finally, to verify the feasibility of the trained runway recognition model in practical applications, an unmanned powered parachute vehicle platform was built for actual autonomous landing and actual flight testing.

The rest of this paper is organized as follows: In Section 2, a 6-DOF model of an unmanned powered parachute vehicle is developed and an offset controller is designed. In Section 3, a lightweight landing runway recognition model is trained and validated in various scenarios. In Section 4, practical flight tests are conducted to verify the feasibility of the proposed scheme. A summary of this study is presented in Section 5.

II. SYSTEM MODEL

A. STRUCTURE AND PARAMETERS OF THE WING PARACHUTE

To establish the mathematical model of the unmanned powered parachute vehicle, the relevant parameters of the wing parachute are given as shown in Figure 1. When the unmanned powered parachute vehicle is flying in air, the air fills the wing parachute. At this time, the length of the horizontal projection of the parachute jacket is spread b, the width is the chord length c, and the farthest vertical distance between the upper and lower airfoil is e. The vertical distance from the top of the leading edge of the parachute jacket to the two end points is h. The virtual intersection point of the parachute cord is point C, and its distance from the parachute jacket is r. The roll center is R, and the pitch center is P.

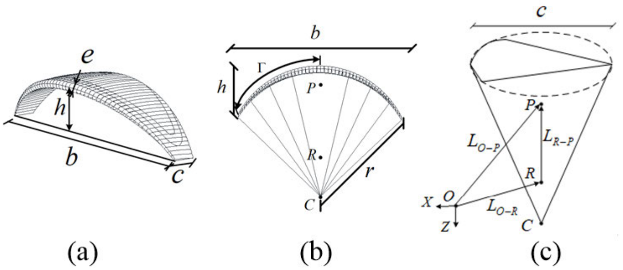


FIGURE 1. Related parameters of the wing parachute: (a) parachute shape, (b) frontal view and (c) side view.

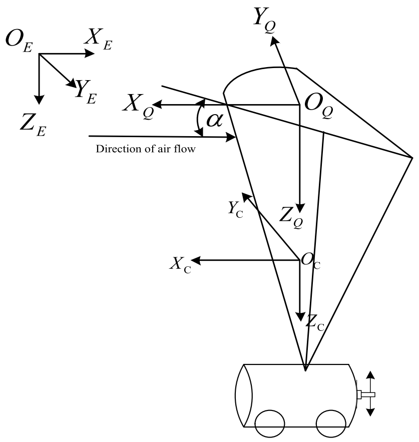


FIGURE 2. Unmanned powered parachute vehicle coordinate system diagram.

The coordinate system of the UPPV system mainly consists of the body coordinate system $(O_TX_TY_TZ_T)$, geodetic coordinate system $(O_EX_EY_EZ_E)$ and airflow coordinate system $(O_QX_QY_QZ_Q)$.The camera axis and the parachute are aligned. A schematic of the coordinate system of an unmanned powered parachute vehicle is shown in Fig. 2.

B. MOTION EQUATION OF UNMANNED POWERED PARACHUTE VEHICLE

The dynamical equations of the UPPV system are based on momentum and momentum moments and consist of two parts: the additional mass and true mass. Assume that $V_O = (u, v, w)$ denotes the momentum tri-axial velocity, and denotes $W = (p, q, r)$ the tri-axial angular velocity. Then, the matrix form of the moment $P_{ZS,O}$ and momentum moment

$H_{ZS,O}$ generated by the real mass can be expressed as (1):

$$\begin{bmatrix} P_{ZS,O} \\ H_{ZS,O} \end{bmatrix} = \begin{bmatrix} m_{ZS}E & -m_{ZS}L^{\otimes}_{O-MC} \\ m_{ZS}L^{\otimes}_{O-MC} & J_{ZS,O} \end{bmatrix} \begin{bmatrix} V_O \\ W \end{bmatrix} = A_{ZS,O} \begin{bmatrix} V_O \\ W \end{bmatrix} \quad (1)$$

where m_{ZS} denotes the true mass of the unmanned parachute vehicle, E denotes a 3 × 3 unit matrix, $J_{ZS,O}$ denotes the rotational inertia of the true mass with respect to the origin; L_{O-MC} denotes the lost path from the origin of the parachute vehicle system to the center of mass, and $\otimes$ denotes the anti-symmetric matrix of the modified matrix.

The matrix form of the momentum $P_{FJ,O}$ and momentum moment $H_{FJ,O}$ generated by the additional mass can be expressed as (2):

$$\begin{bmatrix} P_{FJ,O} \\ H_{FJ,O} \end{bmatrix} = \begin{bmatrix} M_a & -M_a(L^{\otimes}_{O-R} + L^{\otimes}_{R-P}T_2) \\ M_a(L^{\otimes}_{O-R} + L^{\otimes}_{R-P}T_2) & J_{a,O} \end{bmatrix} \begin{bmatrix} V_O \\ W \end{bmatrix} = A_{a,O} \begin{bmatrix} V_O \\ W \end{bmatrix} \quad (2)$$

where $A_{a,O}$ denotes the matrix for calculating the additional mass of the unmanned powered parachute vehicle, M_a denotes the rotation added mass component matrix, and T_2 is the selection matrix. The motion equations of the unmanned powered parachute system are as follows:

$$\begin{bmatrix} \dot{V}_O \\ \dot{W} \end{bmatrix} = [A_{ZS,O} + A_{a,O}]^{\otimes} \begin{bmatrix} F_{S,A}+F_g+F_{ZS,ZT}+F_{FJ,ZT}+F_{TL} \\ M_{S,A}+M_g+M_{ZS,ZT}+M_{FJ,ZT}+M_{TL} \end{bmatrix} \quad (3)$$

where $F_{S,A}$ and $M_{S,A}$ denote the aerodynamic force and pneumatic moment, respectively; F_g and M_g denote gravity and gravitational force, respectively; F_{TL} and M_{TL} denote the propeller thrust and thrust moment, respectively.

C. OFFSET CORRECTION

When the unmanned powered parachute vehicle arrives near the destination area via GPS, the runway detection system is turned on. After the runway recognition model identifies the runway, the detection system sends the runway center coordinate, width, height, and offset information to the flight controller. Offset correction was then performed. During the visual adjustment phase, the parachute vehicle maintains a constant descent speed and the overall process is shown in Fig. 3.

The lateral control of the UPPV is mainly achieved by pulling down the left and right servos. Using PID control, where P denotes the proportional, I denotes the integral, and D denotes the differential. The input quantity was the offset quantity. Then the output amount of the PID is converted into a pulse width signal and input to the input interface of the single-side servo, and the offset correction is completed by controlling the pull-down amount of the left and right servos.

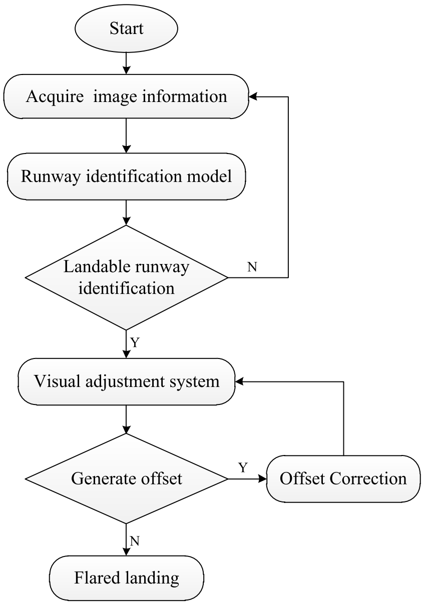

Start
Acquire image information
Runway identification model
Landable runway identification
N
Y
Visual adjustment system
Generate offset
Y
Offset Correction
N
Flared landing

FIGURE 3. Flow diagrams for Visual landing.

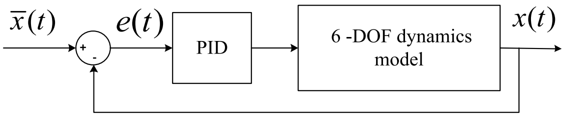


FIGURE 4. Schematic diagram of PID controller structure.

Define the expected value of the system as $\bar{x}(t)$ and the actual displacement of the system output as $x(t)$. Then the system control error is $e(t) = x(t) - \bar{x}(t)$. To eliminate the steady-state error, the integration term was set to $\int e(t) * e^{-0.1|e(t)|}dt$. When the deviation value is large, $e^{-0.1|e(t)|}$ is small and the integration link does not work, in which case the correction time can be shortened. When the deviation value was small, the integration link worked. At this time, it can eliminate the static difference and improve the control accuracy. The control law can be expressed as:

III. RUNWAY RECOGNITION MODEL

A. MOBILENET-V3 BASED YOLOV4 NETWORK MODEL

The YOLO network is an end-to-end fast detection method that transforms the target detection problem into a regression problem. The YOLOV4 [22] algorithm used in this study gradually makes up for many defects after several iterations. Its backbone network CSPDarknet53 has higher accuracy and real-time performance in target detection.

Lightweight networks are characterized by a small number of parameters, a simple structure, and fast operation. It is suitable for application to mobile hardware platforms with limited storage and computational resources [23]. SqueezeNet [24] replaces the 3 × 3 convolution with 1 × 1 convolution to reduce the parameters. ShuffleNet [25] takes advantage of group convolution and channel shuffle to improve performance. MobileNet families separate

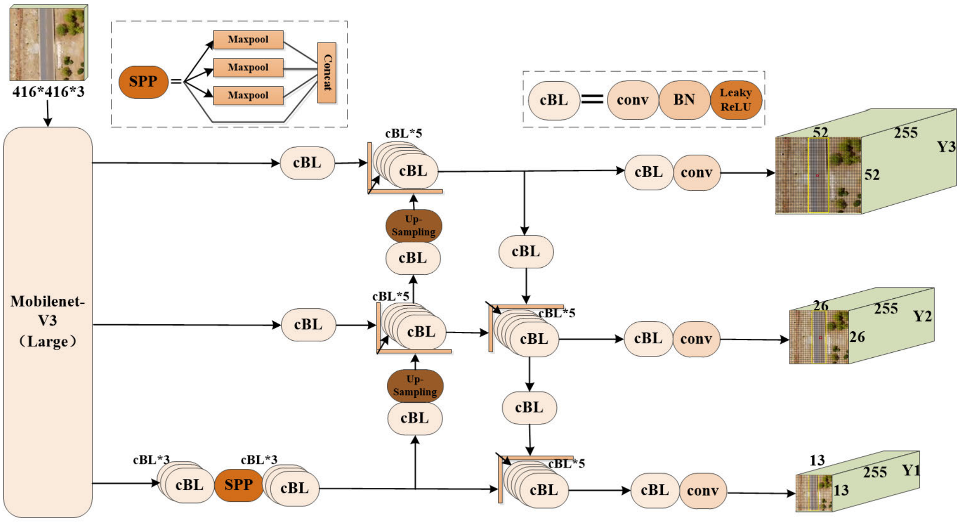


FIGURE 5. Algorithm structure for runway recognition model.

convolutional operations into channel-level and depth operations to accelerate the computation. MobileNet-V3 uses a network architecture search algorithm called NAS and a hard swish activation function to improve performance with almost no speed loss. It is less computationally intensive than the SqueezeNet network and is faster in detection compared to the ShuffleNet network. In order for port the trained model to the airborne image processor, Mobilenet-V3 (large) is the most suitable choice. This network has higher accuracy and faster speed on the VOC dataset compared to Mobilenet-V1 and Mobilenet-V2. The structure of Mobilenet-V3 (Large) is shown in Table 1.

TABLE 1. The overall structure of Mobilenet-V3 (Large) network.

Symbol	Operator	exp size	out	SE	NL	s
224^2*3	conv2d3*3	-	16	-	HS	2
112^2*16	bneck 3*3	16	16	-	RE	1
112^2*16	bneck 3*3	64	24	-	RE	2
56^2*24	bneck 3*3	72	24	-	RE	1
56^2*24	bneck 5*5	72	40	√	RE	2
28^2*40	bneck 5*5	120	40	√	RE	1
28^2*40	bneck 5*5	120	40	√	RE	1
28^2*40	bneck 3*3	240	80	-	HS	2
14^2*80	bneck 3*3	200	80	-	HS	1
14^2*80	bneck 3*3	184	80	-	HS	1
14^2*80	bneck 3*3	184	80	-	HS	1
14^2*80	bneck 3*3	480	112	√	HS	1
14^2*112	bneck 3*3	672	112	√	HS	1
14^2*112	bneck 5*5	672	160	√	HS	2
7^2*160	bneck 5*5	960	160	√	HS	1
7^2*160	bneck 5*5	960	160	√	HS	1
7^2*160	conv2 1*1	-	960	-	HS	1
7^2*960	Pool 7*7	-	-	-	-	1
1^2*960	conv2 1*1,NBN	-	1280	-	HS	1
1^2*1280	conv2 1*1,NBN	-	k	-	-	1

In Table 1, bneck is the bottleneck layer, exp size is the number of channels within each bneck after the inverse residual structure rises; out indicates the number of channels input to the features in the benck; SE is the introduction of the attention mechanism; and NL is the type of activation function. where HS is the activation function, RE is the ReLU activation function, and s is the step length used for each block structure.

The YOLOv4 algorithm was applied to UPPV to identify landing roads and the lightweight target detection network where the backbone network of feature extraction is replaced by Mobilenet-V3 (Large). Fig. 5 provides the architecture of the runway detection method. First, the input image is normalized to a size of 448 × 448 pixels by the model. Then, the image was uniformly partitioned into a grid of 13 × 13 passed into the detection network of the framework. The algorithm outputs image features at three different scales (Y1, Y2, and Y3), which means that multiple scales are used to detect targets of different sizes.

YOLOv4 uses the k-means clustering method to default on the size of the bounding box. Using the IOU distance, the relative coordinates of the center point of the bounding box at the prediction relative to the upper left corner of the grid cell can be directly predicted, and the location, size, and score of the target are obtained through the calculation of equation (4).

$$\begin{cases} b_x = \sigma(t_x) + c_x \\ b_y = \sigma(t_y) + c_y \\ b_w = p_w e^{t_w} \\ b_h = p_h e^{t_h} \\ p_r(object) * IOU(b, object) = \sigma(t_o) \end{cases} \tag{4}$$

where are the model's predicted output detection target's centroid coordinates, bounding box size and score. Fig.6 shows the calculation schematic.

Where (c_x, c_y) is the coordinate of the grid cell, (p_w, p_h) is the size of the bounding box before prediction, and (b_x, b_y, b_w, b_h) is the center coordinate and size of the prediction.

B. DATA MAKING

Deep learning models require sufficient training samples to avoid overfitting the training data and negatively affect the recognition rate of runways. We collected 1738 images of the

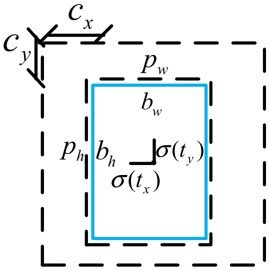
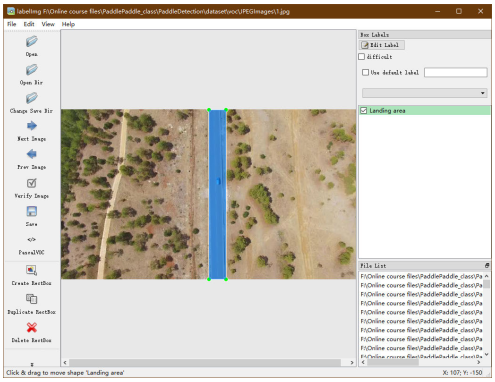


FIGURE 6. Schematic diagram of the bounding box calculation.


labelImg F:\Online course files\PaddlePaddle_class\PaddleDetection\dataset\voc\JPEGImages\1.jpg

☑ Landing area

F:\Online course files\PaddlePaddle_class\Pa
F:\Online course files\PaddlePaddle_class\Pa
F:\Online course files\PaddlePaddle_class\Pa
F:\Online course files\PaddlePaddle_class\Pa
F:\Online course files\PaddlePaddle_class\Pa
F:\Online course files\PaddlePaddle_class\Pa
F:\Online course files\PaddlePaddle_class\Pa
F:\Online course files\PaddlePaddle_class\Pa
F:\Online course files\PaddlePaddle_class\Pa
F:\Online course files\PaddlePaddle_class\Pa

Click & drag to move shape 'Landing area'

X: 107; Y: -150

FIGURE 7. The interface of LabelImg labeling tool.

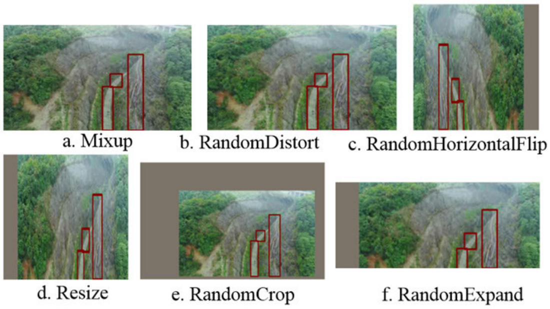

a. Mixup
b. RandomDistort
c. RandomHorizontalFlip
d. Resize
e. RandomCrop
f. RandomExpand

FIGURE 8. Data enhancement effect for six methods.

runway in different environments from the Internet and aerial videos and used data enhancement methods to expand them to more than 5000 as training samples.

The training, validation and test samples were set to account for 90%, 6% and 4% of the total data samples. We used LabelImg to label the collected sample images to establish the target pixel information (as shown in Fig.7). In this study, we designed the MobileV3-YOLOv4 model and adjusted the parameters of the model during training to set runways as the recognition target.

The dataset used for our training model was a manually labeled VOC dataset of landing runways for unmanned powered parachute vehicles. We used the RandomDistort, RandomExpand, RandomCrop, RandomHorizontalFlip, Resize and Mixup methods to augment the training samples. (as shown in Fig. 8).

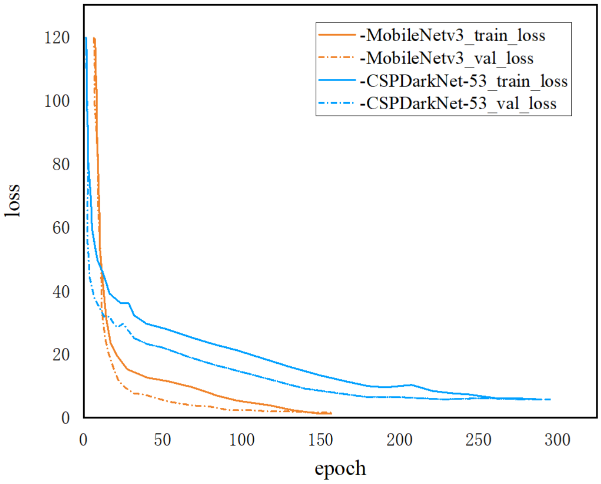


FIGURE 9. Comparison of the training effect of two models.

C. MODEL TRAINING

The experiment was completed in an environment of NVIDIA GTX1660S GPU with 14GB memory, WIN10 operating system, and Python 3.6. In the model training process, the total number of iterations was set to 270 and the batch size was set to eight. The initial learning rate was set to 0.001 and the learning rate was decayed by cosine annealing.

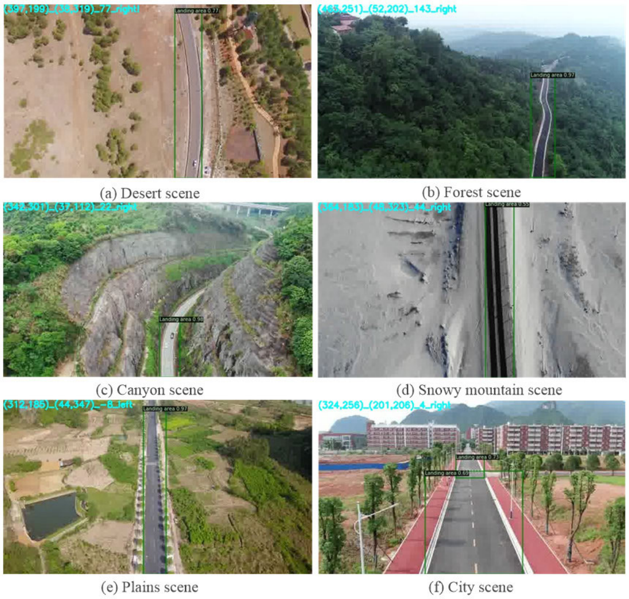
Under the same experimental environment and parameter settings, we trained the YOLOv4 network and compared it. The training loss curves for the two models are shown in Fig. 9. From the figure, it can be seen that the model combined with the lightweight network MobileNetv3 converges faster compared to the original Yolov4 and can reach a minimum of 1.

To test the recognition effect of the model, one frame image from multiple scene video streams is extracted for the road recognition test, and the detected coordinates of the center of the landing runway, the width and height of the recognition area, and the offset and deflection direction decisions are displayed in the upper left corner of the picture information, as shown in Fig.10.

The experimental results from training Darknet-53, CSPDarknet-53, Tiny, MobileNet-v2, and MobileNet-v3 are shown in Table 2. The CSPDarknet-53 backbone network is excessively complicated with high model weight of 235 MB and long average detection time of 164.6ms. Therefore, it is difficult for deeper backbone networks to meet high-speed requirements on embedded devices. MobileNet-v3 simplifies the backbone network with an AP value of 98.21%, which was slightly lower than CSPDarknet-53, but it reduces the average detection time by 113.6ms compared to the CSPDarknet-53. Furthermore, the model weight was only 90.3 MB, which significantly reduced the operating cost of the embedded devices. In summary, MobileNet-v3 has the characteristics of high detection accuracy, high detection speed, and low model memory consumption, making it distinctly advantageous for embedded devices.

TABLE 2. Performance comparison of different backbone networks.

Backbone network	Model size	mAP	Predicted speed
DarkNet53	223.3MB	91.32%	132.5ms
CSPDarkNet53	235.0MB	98.96%	164.6ms
Tiny	123.0MB	81.32%	62.5ms
MobileNetV2	102.5MB	95.47%	67.3ms
MobileNetV3	90.3MB	97.81%	51.0ms


(397,199)_(38,319)_77_right!
Landing area 0.77
(463,251)_(52,202)_143_right
Landing area 0.97
(342,301)_(37,112)_22_right
Landing area 0.98
(364,183)_(48,323)_44_right
Landing area 0.55
(312,185)_(44,347)_-8_left
Landing area 0.97
(324,256)_(201,206)_4_right
Landing area 0.73
Landing area 0.69

(a) Desert scene (b) Forest scene (c) Canyon scene (d) Snowy mountain scene (e) Plains scene (f) City scene

FIGURE 10. Offset correction effect in different scenes.

IV. LANDING TEST

A. TEST PLATFORM CONSTRUCTION

The unmanned powered parachute vehicle test platform chosen for this test has a double oval shape of the wing parachute jacket. The load capacity of the platform was 15∼25 kg and the designed maximum speed was 45 km/h. Referring to the relationship between the parameters of wing loading ratio, span ratio and parachute cord length of the wing parachute given by Steven Lingard [26], the selected parameters of the designed wing parachute are listed in Table 3.

TABLE 3. Parameters of unmanned powered parachute vehicle.

Parameter	Numerical value
Spread Length	6 m
Chord length	1.2 m
Spread Sine Ratio	5
System length	2.6m
Cord length	2.2m
Area	7.5m^2
Wing loading ratio	2.7
Installation angle	7°

The system for autonomous landing of a UPPV can be divided into an image processing system and a flight control system as shown in Fig. 11. In the image processing system, the trained runway recognition model needs to be deployed on an airborne image processor for sitting and offset the information to the flight controller. Among them, the high-speed wide-angle camera of the UVC protocol was selected as the image acquisition equipment, and one end of the lens was connected with an external polarization lens, which can effectively solve the problem that the characteristics of the measured object cannot be extracted owing to the reflection. Raspberry PI 4B was selected as the airborne image processor, and Paddle-lite 2.0 was used as the model transplantation tool in this study. Next, the flight control system adjusts the displacement and attitude according to the coordinates and other information sent by the image processing system, and combines it with GPS and barometer data collection to control the unmanned powered parachute vehicle for visual adjustment and autonomous landing. The flight controller hardware consists of a core board and an expansion board. The core board adopts the STM32F765IIK6 processor as the core unit module, and peripherally connects the accelerometer, gyroscope, magnetometer, barometric altimeter and other airborne sensor modules.

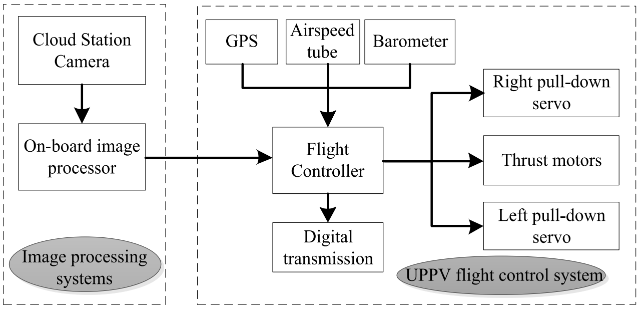


FIGURE 11. Block diagram of autonomous landing system for UPPV.

B. FLIGHT TEST AND RESULTS

To verify the feasibility of the algorithm in a practical environment, an unmanned powered parachute vehicle

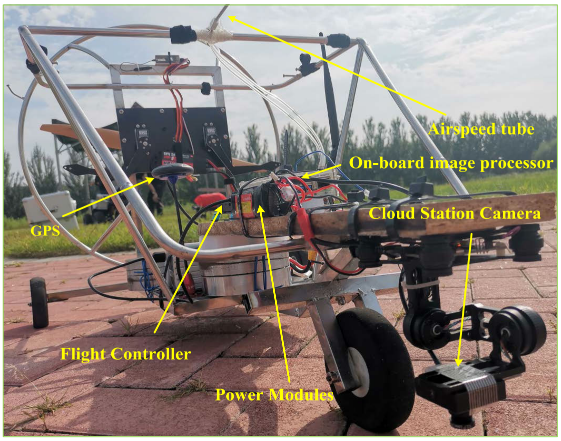


FIGURE 12. Physical image of unmanned powered parachute vehicle.

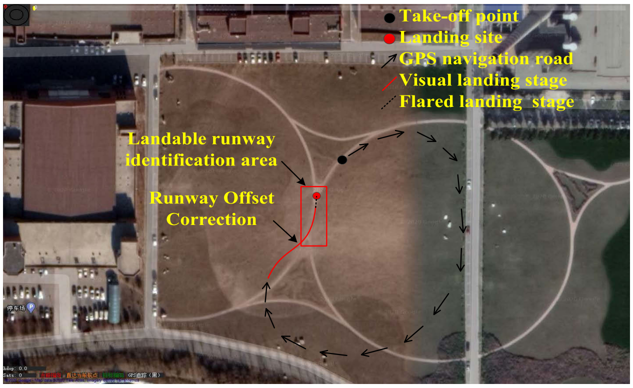


FIGURE 13. Schematic diagram of the test site and flight path.

FIGURE 14. Light test of UPPV platform: (a) taking off; (b) turning right; (c) landing.

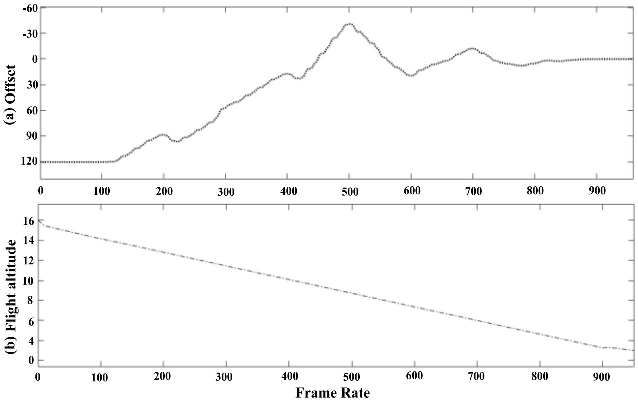


FIGURE 15. Offset (a) and flight height (b) at the same frame rate.

TABLE 4. Comparison of data of multiple flight tests.

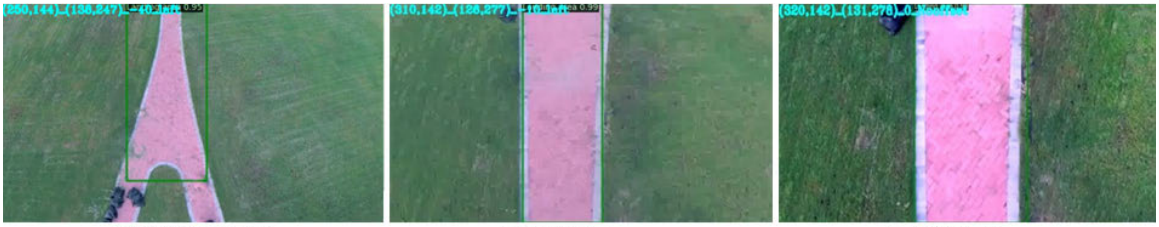
No.	Initial offset(m)	Accuracy (%)	Time (s)
1	120	97.56	14.79
2	148	98.47	14.85
3	135	97.47	14.93
4	96	98.16	13.30
5	104	98.25	13.87

autonomous landing visual system test platform was built as shown in Fig. 12 and tested in an autonomous landing flight at the school. Fig. 13 provides a site that is safe for flight testing and a schematic of the flight path.

Fig. 14(a)-(c) show the state of the UPPV platform: being taken off, turning right in the air, and landing during this flight test. The autonomous landing phase of the UPPV based on the visual positioning system lasted 15s, and the high-speed wide-angle camera worked for 15s, shooting 60 frames per second to capture a total of 900 frames. The offset values of each image frame when the UPPV enters the offset correction phase are shown in Figure 15(a). After entering the offset correction phase, Fig. 15(b) shows the flight altitude of the test platform at the time of taking each image frame.

As shown in Fig. 15, the UPPV test platform enters the initial phase of the visual landing with an offset of 120 and a flight altitude of 16 m. The test platform then enters the offset correction phase while decreasing the flight altitude at a speed of 3.6 km/h. Finally, the test platform enters the flared-landing phase, and both forward and vertical speeds are greatly reduced. Fig. 16 (a) provides real-time feedback from the airborne image processor when the UPPV test platform correct the runway offset; at this time, the offset is -40 and the test platform should be adjusted to the left; Fig. 16 (b) provides the middle stage of visual adjustment; at this time, the offset is -10 and it should be fine-tuned to the left; Fig. 16 (c) provides the end of visual adjustment, at this time, the offset is 0 and the test platform enters the flared landing phase.

The autonomous landing test of the UPPV was performed five times. Table 4 lists the initial offset of the visual landing phase, the integrated accuracy of landing runway recognition, and the total time of the offset correction process. In Table 4, the initial offset of the UPPV test platform is large owing to the error in GPS navigation when it enters the visual landing phase. However, the offset can be corrected to 0 within 15s with the effect of the autonomous landing visual positioning system, and the comprehensive recognition accuracy of the landing runway was higher than 97.47% for each test during the entire visual landing process.


(250,144)_(136,247)_-40_left 0.95
(310,142)_(126,277)_-10_left 0.99
(320,142)_(131,278)_0_Nooffset

(a) 500th frame (b) 600th frame (c) 900th frame

FIGURE 16. Offset correction of test platform in the air.

V. CONCLUSION

In this present paper, a vision-based autonomous landing system for unmanned powered parachute vehicles is proposed. A landing runway recognition model based on the lightweight Mobilenet-V3 and YOLOv4 algorithm was trained for the vision-based autonomous landing problem of unmanned powered parachute vehicles. Compared with models trained under different backbone networks, the model achieved a combined accuracy of 97.81% in identifying landing runways and a 68.92% improvement in prediction speed over YOLOv4-CSPDarkNet53. Furthermore, this model has the advantage of a small weight and makes it more suitable for use in embedded devices and mobile terminals. The controller of the landing runway deviation was designed to shorten the landing time. To verify the feasibility of the method in this study, an unmanned powered parachute flying machine test platform was built to conduct actual autonomous flight and landing tests. The results show that the unmanned powered parachute vehicle can complete the offset correction and enter the flared-landing stage within 15s, meeting the real-time and high-precision requirements of the autonomous landing visual positioning system.

In the future, there is still room for improvement of the model and how to make the lightweight network achieve such detection speed without loss of accuracy and apply it to richer detection scenarios. For example, recognition and landing in dark conditions and stability in windy conditions are the next problems to be solved.

VI. ACKNOWLEDGMENT

(Wei Hu and Mengxuan Zhang contributed equally to this work.)

REFERENCES

[1] V. Devalla and O. Prakash, "Developments in unmanned powered parachute aerial vehicle: A review," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 29, no. 11, pp. 6–20, Nov. 2014.

[2] Q. Chen, Y. Sun, M. Zhao, and M. Liu, "A virtual structure formation guidance strategy for multi-parafoil systems," *IEEE Access*, vol. 7, pp. 123592–123603, 2019.

[3] F. Lv, W. He, and L. Zhao, "A multivariate optimal control strategy for the attitude tracking of a parafoil-UAV system," *IEEE Access*, vol. 8, pp. 43736–43751, 2020.

[4] F. Santoso, M. A. Garratt, and S. G. Anavatti, "Visual–inertial navigation systems for aerial robotics: Sensor fusion and technology," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 1, pp. 260–275, Jan. 2017.

[5] N. Murali, M. Dineshkumar, and A. K. Wc, "Parafoil trajectory comparison for optimal control and proportional controller," in *Proc. Int. Conf. Control Commun. Comput. (ICCC)*, Dec. 2013, pp. 227–232.

[6] Y. Zhigui and L. ChuanJun, "Review on vision-based pose estimation of UAV based on landmark," in *Proc. 2nd Int. Conf. Frontiers Sensors Technol. (ICFST)*, Apr. 2017, pp. 453–457.

[7] S. Nazir, S. Aziz, Y. Khaliq, and S. M. Adnan, "Vision based autonomous runway identification and position estimation for UAV landing," in *Proc. Int. Conf. Artif. Intell. Data Process. (IDAP)*, Sep. 2018, pp. 28–30.

[8] A. F. Fadhil, R. Kanneganti, L. Gupta, H. Eberle, and R. Vaidyanathan, "Fusion of enhanced and synthetic vision system images for runway and horizon detection," *Sensors*, vol. 19, no. 17, p. 3802, Sep. 2019.

[9] H. Zhang, J. Liang, H. Jiang, Y. Cai, and X. Xu, "Lane line recognition based on improved 2D-gamma function and variable threshold Canny algorithm under complex environment," *Meas. Control*, vol. 53, nos. 9–10, pp. 1694–1708, Nov. 2020.

[10] S. Tandra and Z. U. Rahman, "Robust edge-detection algorithm for runway edge detection," *Proc. SPIE*, vol. 6813, Feb. 2008, Art. no. 68130L.

[11] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

[12] Ö. Aytekin, U. Zöngür, and U. Halici, "Texture-based airport runway detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 3, pp. 471–475, May 2013.

[13] M. Zhang, H. Li, G. Xia, W. Zhao, S. Ren, and C. Wang, "Research on the application of deep learning target detection of engineering vehicles in the patrol and inspection for military optical cable lines by UAV," in *Proc. 11th Int. Symp. Comput. Intell. Design (ISCID)*, Dec. 2018, pp. 97–101.

[14] Z. Xiao, Y. Gong, Y. Long, D. Li, X. Wang, and H. Liu, "Airport detection based on a multiscale fusion feature for optical remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 9, pp. 1469–1473, Sep. 2017.

[15] J. Akbar, M. Shahzad, M. I. Malik, A. Ul-Hasan, and F. Shafait, "Runway detection and localization in aerial images using deep learning," in *Proc. Digit. Image Comput., Techn. Appl. (DICTA)*, Dec. 2019, pp. 1–8.

[16] I. Rocco, R. Arandjelovic, and J. Sivic, "Convolutional neural network architecture for geometric matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 11, pp. 2553–2567, Nov. 2019.

[17] P. Zhang, X. Niu, Y. Dou, and F. Xia, "Airport detection from remote sensing images using transferable convolutional neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 2590–2595.

[18] Z. Menghan, L. Zitian, and S. Yuncheng, "Optimization and comparative analysis of YOLOV3 target detection method based on lightweight network structure," in *Proc. IEEE Int. Conf. Artif. Intell. Comput. Appl. (ICAICA)*, Jun. 2020, pp. 20–24.

[19] E. Cintas, B. Ozyer, and E. Simsek, "Vision-based moving UAV tracking by another UAV on low-cost hardware and a new ground control station," *IEEE Access*, vol. 8, pp. 194601–194611, 2020.

[20] C.-J. Chen, Y.-Y. Huang, Y.-S. Li, Y.-C. Chen, C.-Y. Chang, and Y.-M. Huang, "Identification of fruit tree pests with deep learning on embedded drone to achieve accurate pesticide spraying," *IEEE Access*, vol. 9, pp. 21986–21997, 2021.

[21] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324.

[22] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*. [Online]. Available: http://arxiv.org/abs/2004.10934

[23] Y. Yang, Y. Liao, S. Ni, and C. Lin, "Study of algorithm for aerial target detection based on lightweight neural network," in *Proc. IEEE Int. Conf. Consum. Electron. Comput. Eng. (ICCECE)*, Jan. 2021, pp. 422–426.

[24] A. G. Santos, C. O. de Souza, C. Zanchettin, D. Macedo, A. L. I. Oliveira, and T. Ludermir, "Reducing SqueezeNet storage size with depthwise separable convolutions," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–6.

[25] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.

[26] S. Lingard, "Basic analysis of ram-air parachute," in *Precision Aerial Delivery Systems: Modeling, Dynamics, and Control*. New York, NY, USA: AIAA, 2015, pp. 73–125.

MENGXUAN ZHANG was born in Tieling, Liaoning, in 1996. She received the B.S. degree in information and computing science from Shenyang Aerospace University, in 2018, where she is currently pursuing the master's degree.

Her research interests include image processing and machine learning applied to computer vision.

WEI HU received the Ph.D. degree in mechatronics engineering from Shenyang Institute of Automation, Chinese Academy of Sciences, in 2009.

He was a Visiting Scholar at Virginia Tech, from 2014 to 2015. He is currently a Master's Supervisor with the School of Automation, Shenyang Aerospace University. His research interests include robotics, autonomous control of intelligent equipment, and image processing.

SHUDE JI received the Ph.D. degree from Harbin Institute of Technology, in 2006.

He is currently a Professor and a Master's Supervisor with the School of Aerospace Engineering, Shenyang Aerospace University. He is a member of Liaoning Province's "Hundred Million Talents Project" and a Core Member of Liaoning Province's "Xingliao Talent" innovation team. In recent years, he has undertaken more than 20 projects, including the National Natural Science Foundation of China, the National Defense 863 Project, the Eleventh Five-Year Plan of Science and Technology Support Project, and major scientific research projects of Liaoning Provincial Science and Technology Department.

QI SONG received the Ph.D. degree in pattern recognition and intelligent systems from Shenyang Institute of Automation, Chinese Academy of Sciences, in 2007.

She is currently a Master's Supervisor with the School of Automation, Shenyang Aerospace University. Her research interests include UAV control technology, robot control technology, and the application of artificial intelligence in aviation equipment research.

PENG GONG received the Ph.D. degree in vehicle engineering major from Beijing Institute of Technology, in 2007.

He is currently a Master's Supervisor with Shenyang Aerospace University. His research interests include intelligent vehicle chassis control technology, vehicle automatic transmission control technology, intelligent equipment structure design, and lightweight technology.

LINGPEI KONG received the master's degree in aeronautical engineering from Shenyang Aerospace University.

His research interests include image processing and machine learning applied to computer vision.