Recognition of Landing Target of UAV by Vision Using Machine Learning

Yuya Homma and Seiichiro Moro
Graduate School of Engineering, University of Fukui
3-9-1 Bunkyo, Fukui, Fukui 910–8507, Japan
E-mail: je180311@u-fukui.ac.jp, moro@u-fukui.ac.jp

Abstract
In this paper, we propose a novel method to recognize an H-type landing target accurately based on information only from an image for an unmanned aerial vehicle (UAV). A UAV generally includes an ultrasonic sensor, an infrared sensor, and GPS to obtain its current location. However, there is a possibility that the GPS cannot estimate the correct position depending on the operating environment. By acquiring the height from an image taken with a camera, the UAV can be operated such an environment. In this paper, we focus on the recognition of landing targets, especially for autonomous landing.

1. Introduction
Research on autonomous landing of an unmanned aerial vehicle (UAV) has been conducted mainly from the viewpoint of reliability using GPS and sensors [1]. In addition, research on autonomous landing has been conducted by setting unique circular landing targets and calculating the altitude from the circularity, assuming an environment in which GPS cannot be used [2]. However, there is little research on control using information obtained by detecting a general H-type landing target with a camera. Normally, the landing target of UAVs is often set as a unique recognizable target comprising a circle added to the H-type landing target. However, the operation environments will be limited in such cases. Therefore, as a method that can be applied regardless of the environment at the time of landing, it is conceivable that the recognition of the target from an image of the landing target in advance is more suitable for autonomous landing.

On the other hand, regarding machine learning, which has been emerging in recent years, there are conditions widely used for human recognition. Therefore, machine learning can be used to recognize the landing target with more accuracy.

In this study, we use a camera equipped to provide a vision function, and we attempt to obtain the current altitude using only a camera image and a method of recognizing the landing target from the image on the basis of reliable machine learning. To enable operation even in a GPS-denied environment, the landing targets are extracted using only information from the camera. Then, we perform recognition by machine learning using the extracted image. It is possible to estimate the landing target with high accuracy and high speed by incorporating preprocessing.

2. Proposed Method
Table 1 shows the flow of the recognition algorithm proposed in this research. The input image is an RGB color image from an RPi camera.

2.1. Image Processing
In 2 of Table 1, image processing to estimate the landing target by image recognition is carried out and the distance from the target is determined. In 4 of Table 1, the target range is set to a region of interest (ROI) upon recognition of the landing target on the basis of the estimation. The definition of the ROI is an area to be selected as the operation target in the image data. In 5 of Table 1, whether the selected ROI is the actual landing target or not is evaluated by deep learning.

When the above process is finished, the process returns to 1 of Table 1 and repeated.

3. Processing of Image
For the fast and stable estimation of the target, image processing is performed as a preprocess before inference. The landing target used in this research is shown in Fig. 1. We suppose that the assumed operation environment is mainly a place that does not contain white for example, a green space such as a park or mountainous region.

Table 1: Flow of designed algorithm

<table>
<thead>
<tr>
<th>Landing Target Recognition Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. get image from RPi camera</td>
</tr>
<tr>
<td>2. area processing and altitude measurement of landing target</td>
</tr>
<tr>
<td>3. judge whether or not there is more than the desired area</td>
</tr>
<tr>
<td>4. cut out the landing target</td>
</tr>
<tr>
<td>5. judge whether it is the landing target</td>
</tr>
<tr>
<td>6. compare with previous frame and save position</td>
</tr>
<tr>
<td>7. save the current frame and return to 1</td>
</tr>
</tbody>
</table>

SELECTED PAPER AT NCSP’19
In this case, the load of the processing system increases because a large amount of data should be processed at high speed. By transforming data into a single black-and-white image called a grayscale image, it is possible to reduce the amount of data amount of data to about one-third of that of a color image.

Threshold processing is performed in many methods. In threshold processing, attention is focused on pixels of an image. \( I(p) \) the intensity of a pixel is converted to the grayscale image and \( I_B(p) \) is expressed by a new binary value as

\[
I_B(p) = \begin{cases} 
1 & (thr > I(p)) \\
0 & (thr < I(p)) 
\end{cases} 
\]

where \( thr \) is the threshold and \( p \) is the position of the pixel in the image. In Eq. (1), 1 is white and 0 is black. In this case, the base threshold is set to 210 (maximum: 255) so that the white part of the landing target is properly processed. Additionally, To vary the binarization threshold from luminance information obtained from an image, Otsu’s binarization method [3] is used in this research.

Next we classify the landing target using the area of the white region on the basis of the result of threshold processing. In this case, we measure whether the element is black or white. In other words, when the range enclosed by 1 is acquired, it finally becomes the area of the white region. For use in the deep-learning framework Caffe [4, 5], the reduction of the image of the landing target by nearest neighbor-interpolation is performed to make the vertical and horizontal dimensions equal.

4. Altitude Measurement

We perform altitude measurement to obtain the current position of the UAV. The distance from the camera to the landing target can only be determined if the vertical and horizontal lengths of the landing target are known and the camera is parallel to the landing target.

Figure 2 is a schematic diagram showing the relationship between the image pickup device (IPD) and the landing target. \( s \) is the length of the IPD and \( w \) is the length of the diagonal of the landing target.

The distance we want to evaluate is the distance from the IPD to the landing target. Because the distance to the focal point \( f \) is known, we need to get \( x \) from Fig. 2. The altitude can be obtained from

\[
f + x = \frac{f(s + w)}{s} 
\]

5. Machine Learning

In this research, we use machine learning to recognize the landing target. We also use deep learning based on a neural network. Estimating the landing target with machine learning enable high-accuracy, high-speed processing.

5.1 Deep learning framework Caffe

The deep learning framework adopted in this research is Caffe [4, 5]. Caffe’s features are specialized for image processing and implemented in C++, and it can run at high speed.

To compensate for learning accuracy, we added white noise and partially cut out from the original image.

5.2 Learning by Caffe

To let Caffe learn the landing target, 176 pure learning images and 60 images for evaluation were prepared from the pictures taken. For learning, the dataset was expanded from the 176 images to include, for example, increased noise, adjusted contrast, rotation and trimming.

The image input to the network is a 28 \( \times \) 28 24bit RGB color image. The binary output value of 0 means it is not the landing target and 1 means that it is the landing target.

We used a general two-layer convolution neural network (CNN) (see Fig. 3).

6. Distance from UAV to Landing Target

In the airframe control of the UAV, the movement is obtained by comparison of the previous image frame and the present image frame. In this research, we use two coordinate systems of the landing target and the center of the image. Using them, we estimate the direction of advancement from the
previous to the present frame by mutual conversion. The coordinate system at the center of the image is defined as the body coordinate system and the other is defined as the inertial coordinate system. Then, the UAV creates a map to recognize the surrounding environment. The following variables are used for map creation.

\[
\begin{align*}
    x & \text{ the axis in the lateral direction as seen from the upper part of the UAV} \\
    y & \text{ the axis in the longitudinal direction as seen from the upper part of the UAV}
\end{align*}
\]

The body coordinate system is expressed as \( U(x_1, y_1) \), and the inertial coordinate system is expressed as \( W(x_2, y_2) \).

We use the Euclidean distance, which can be easily calculated, as the measured distance between \( U \) and \( W \). The Euclidean distance is the distance from point to point in Euclidean space. The 2-dimensional Euclidean distance can be expressed by

\[
d_2 = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{3}
\]

By the above processes, 3-dimensional flight control becomes possible using Eq. (3) and the result of altitude measurement.

### 7. Results and Evaluation

#### 7.1 Altitude measurement

Using the video images of the landing target obtained from the camera, several distances are extracted, and the results are shown in Table 2. The relative error is less than 5% and the average error is 1.68% in Table 2. There are relatively good results.

<table>
<thead>
<tr>
<th>actual altitude[cm]</th>
<th>obtained altitude[cm]</th>
<th>relative error[%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>39.2</td>
<td>2.00</td>
</tr>
<tr>
<td>50</td>
<td>48.5</td>
<td>3.00</td>
</tr>
<tr>
<td>60</td>
<td>59.3</td>
<td>1.17</td>
</tr>
<tr>
<td>70</td>
<td>69.6</td>
<td>0.571</td>
</tr>
<tr>
<td>80</td>
<td>79.7</td>
<td>0.375</td>
</tr>
<tr>
<td>90</td>
<td>88.1</td>
<td>2.11</td>
</tr>
<tr>
<td>100</td>
<td>100.2</td>
<td>0.200</td>
</tr>
<tr>
<td>150</td>
<td>146.5</td>
<td>2.33</td>
</tr>
<tr>
<td>200</td>
<td>206.4</td>
<td>3.20</td>
</tr>
<tr>
<td>250</td>
<td>245.5</td>
<td>1.80</td>
</tr>
</tbody>
</table>

#### 7.2 Machine learning

The proposed system learned 1500 times while adjusting several parameters. The results are shown in Fig. 4.

There are seven Caffe parameters: base_lr, momentum, weight_decay, lr_policy, gamma, power and inv. Of these parameters, momentum and power were set to 0.9 and 0.75, respectively, by referring to [5]. The base_lr was set to 0.001 because good learning can be accomplished with a small value. weight_decay and gamma were set to 0.01 and 0.001, respectively, on the basis of the learning tendency by trial and error. In the lr_policy, we set inv for optimal learning.

The horizontal axis represents the number of iterations of learning, which is indicated as the solid purple line. The vertical axis represents learning loss and learning accuracy, indicated as dash green line.

We can that the learning accuracy attains 95%. Learning accuracy is the result of verification using an image randomly selected from the dataset for evaluation in the learned model. Although the learning accuracy normally increases gradually, test accuracy rapidly increases until the number of learning iterations exceeds about 60, as shown in Fig. 4. Because the discrimination is binary, when one is determined, the other is also determined. Therefore, the learning speed can be increased.

#### 7.3 Recognition result of landing target

To confirm the effectiveness of target estimation and altitude measurement by image processing, we first show the results of landing target estimation and recognition (see Table 3). The bottom row of Table 3 shows images not recognized
because they were excluded by preprocessing. These results indicate that we can achieve a better recognition rate by incorporating preprocessing.

Table 3: Recognition results with image processing

<table>
<thead>
<tr>
<th>input image</th>
<th>recognition result</th>
<th>input image</th>
<th>recognition result</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>1</td>
<td>H</td>
<td>0</td>
</tr>
<tr>
<td>H</td>
<td>1</td>
<td>I</td>
<td>1</td>
</tr>
<tr>
<td>L</td>
<td>excluded by</td>
<td>L</td>
<td>excluded by</td>
</tr>
<tr>
<td></td>
<td>preprocessing</td>
<td></td>
<td>preprocessing</td>
</tr>
</tbody>
</table>

7.4 Algorithm execution time

We obtain ten different execution times by inputting different images with the landing target taken every loop (see Table 4). The Raspberry Pi3 Model B+, which will be adopted in further research, is used as the implementation environment and Movidius Myriad2 operating at 600 MHz is used as an accelerator for recognizing landing targets.

Table 4: Results of execution time in one loop

<table>
<thead>
<tr>
<th>times</th>
<th>execution time [s]</th>
<th>times</th>
<th>execution time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0318</td>
<td>6</td>
<td>0.0311</td>
</tr>
<tr>
<td>2</td>
<td>0.0330</td>
<td>7</td>
<td>0.0343</td>
</tr>
<tr>
<td>3</td>
<td>0.0359</td>
<td>8</td>
<td>0.0291</td>
</tr>
<tr>
<td>4</td>
<td>0.0367</td>
<td>9</td>
<td>0.0328</td>
</tr>
<tr>
<td>5</td>
<td>0.0313</td>
<td>10</td>
<td>0.0293</td>
</tr>
</tbody>
</table>

In general, the hardware installed in the UAV is effective for processing at 5 frames per second (5 fps) or more [6]. If it is more than 5 fps, the minimum operation within the assumed range can be performed. In Table 4, the processing execution time of only 10 iterations is measured, but the average processing execution time throughout the whole operation was 0.0322. In terms of processing speed, it is possible to process at about 33 fps. Therefore, it is considered that actual operation is possible by the proposed method.

7.5 Recognition of landing target

Table 5 shows the results of recognition based on the images captured assuming actual flight. The fourth item shows the perfectly recognized landing target including the probability. The total number of frames was 600, and 598 frames were recognized correctly. Accurate recognition and a correct probability of 1 were achieved for 592 frames.

In this results, the probability label for successful recognition is 1. Thus, we achieved a recognition rate of 98.6% with preprocessing. The combination of preprocessing and machine learning was very effective improving recognition.

8. Conclusion

In this paper, we proposed a method for accurately recognizing landing target, such as a helipad, using information from images. By designing an algorithm for autonomous landing using information from a camera and without using a sensor, we were able to confirm the effectiveness of the method of UAV control by the new learning method. The recognition rate of 98.6% was achieved with preprocessing. We considered this to be a result of estimating the landing target in advance or correcting the landing target on the basis of altitude.

Acknowledgment

This work is partly supported by Yamada-Giken Co., Ltd., Fukui, Japan.

References