Assessing the Impact of Seasonal Lighting Variations on Drone Visual Positioning

Che-Cheng Chang
Department of Information Engineering
and Computer Science
Feng Chia University
Taichung, Taiwan
checchang@fcu.edu.tw

Bo-Yu Liu
Department of Information Engineering
and Computer Science
Feng Chia University
Taichung, Taiwan
d1090698@o365.fcu.edu.tw

Bo-Ren Chen
Department of Information Engineering
and Computer Science
Feng Chia University
Taichung, Taiwan
gongjuem@gmail.com

Po-Ting Wu
Department of Information Engineering
and Computer Science
Feng Chia University
Taichung, Taiwan
wumoris@gmail.com

***Abstract*—Positioning systems and algorithms are essential for drone applications. Nowadays, Global Positioning Systems (GPS) are the most common method for drone positioning, but GPS may not always be precise and available. In the literature, a visual-based positioning study uses a Convolutional Neural Network (CNN) to match geometric features for drone positioning. However, they do not consider the impact of seasonal lighting variations. Hence, by incorporating several critical components into a CNN, we design a new architecture to position the drone despite the seasonal lighting variations. According to the experimental results, our method can deal appropriately with the issue above and provide enough accuracy and stability for drone positioning.**

Keywords—Drone Positioning, Convolutional Neural Network (CNN), Lighting Variations

I. Introduction

A positioning system is designed to determine the location of an object within a defined space [1]. While GPS (Global Positioning System) is the most well-known and significant advancement in traditional navigation, its accuracy and reliability are compromised by various factors, such as multi-path propagation and signal attenuation [2]. Recently, three major classes of methods for positioning objects are commonly discussed in the literature to ensure the safety of advanced and intelligent applications. The first is the sensor-fusion technique, which combines data from multiple sensors to estimate the pose. The second is the device-assisted method, which relies on ground sensors to determine the position and trajectory. Lastly, the third is the vision-based approach, which analyzes geometric features to identify the position [1, 2]. Notably, the latter two can also be considered parts of a sensor-fusion technique.

To address the limitations of GPS, our approach leverages onboard cameras, employing the vision-based method for task completion. In the literature, a visual-based positioning study uses a Convolutional Neural Network (CNN) to match geometric features for drone positioning [1]. However, they do not consider the impact of seasonal lighting variations. This may reduce the confidence level needed to realize the study in practice. Hence, by incorporating several critical components into a CNN, we design a new architecture to position the drone despite the seasonal lighting variations. More specifically, the authors introduce several kinds of CNN-based network blocks in [3], and we will consider these blocks to conduct our experiments for corresponding models.

In this study, via using an orthophotomap, we can train models and then estimate their performance, e.g., accuracy, computing load, and so forth. Our procedure is the same as that in [1]. Noteworthily, an orthophotomap is an image made by merging several orthophotos that drones or satellites photograph and have been corrected [1]. Consequently, our efforts can be regarded as actual experiments, not simulations. This will increase confidence in succeeding porting procedures to drone applications.

On the other hand, this topic differs from traditional supervised learning, where training and testing data are mutually exclusive. In the case of drones, they cannot position themselves using the vision-based approach without pre-existing map data. Accordingly, our training data must include parts of the testing data. Moreover, we further adopt the concept of data augmentation to augment the original training data, generating more testing data with seasonal lighting variations to assess our models. Remarkably, these data will be regarded as new testing data and not included in the original training data and original testing data.

The structure of this paper is as follows. Section 2 covers the essential background information for this work. In Section 3, we present our concept, detailing the designs and improvements. Section 4 shows our experiments and corresponding results. Based on these experimental results, we gather statistics and provide discussions. Finally, in the last section, we conclude this study and give potential directions for future research.

II. Background Knowledge

A. Data Augmentation with Lighting Variations

Data augmentation is a technique utilized in Machine Learning (ML) to increase the diversity of the training dataset without actually collecting new data. It involves applying various transformations to the existing dataset and generating new versions of the original dataset. This can help improve the ability of the ML model to generalize to unseen data and reduce overfitting.

In this paper, we use the concept of the conversion for global solar radiation in Taiwan [4, 5] to augment the original training data and generate more testing data with lighting variations to assess our models. More specifically, according to [4, 5], we obtain the difference and corresponding formula of light variations among arbitrary moments for the experimental area. Thus, we can adjust the brightness of the orthophotomap to generate new testing data.

B. Datasets

A large orthophotomap is used to conduct our experiments. Firstly, it is divided into many small orthophotomaps called the original training data. Since drones cannot position themselves using the vision-based approach without pre-existing map data, the original testing data are set to the original training data so far. Next, we further adopt the concept of data augmentation to augment the original training data, generating more testing data with seasonal lighting variations to assess our models. Remarkably, these data will be regarded as new testing data and not included in the original training data and original testing data.

Figure 1 shows the relationship among the original training data, the original testing, and the new testing data with lighting variations. It illustrates the scenario where a drone can position itself within a specific area but is not limited to lighting variations.

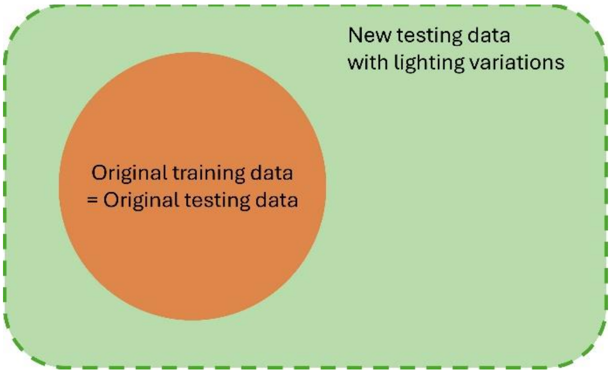

Figure 1. The relationship among three datasets.

C. Convolutional Neural Network (CNN)

Convolution is a concept in the structure of the animal visual cortex, where individual neurons respond to stimuli within their respective receptive fields [1]. Since these receptive fields partially overlap, they can collectively cover the entire visual field, effectively performing convolution operations. Convolutional layers are typically paired with pooling layers to condense the outputs from the previous layer into a single output for the next layer. This significantly reduces the number of weights and computational complexity. Incorporating one or more convolutional and pooling layers in the early stages of a neural network will bring the benefits of these two techniques into the traditional architecture [1, 6, 7].

D. Regression and Classification

Regression and classification are supervised learning algorithms that work with labeled datasets. The critical difference between them lies in the problems they address. Regression is used to predict continuous values, while classification is used to categorize data into discrete labels. In this paper, our positioning problem is treated as a classification problem rather than a regression one. In our positioning problem, by applying the sliding window technique [8] over a small range, the precision of classification can be close to that of regression, which is an important and interesting aspect of this study [1, 6, 7].

III. The Learning and Testing Procedures

Here, we introduce the learning and testing procedures in our vision-based approach, and the pseudocode is given in Table 1.

First, a large orthophotomap is inputted as the sole source (line 01). Then, several variables are defined for our algorithm: stride s is the distance the window moves at each step (line 02), *original-training-data* is a set of small orthophotomaps as the original training data (line 03), *original-testing-data* is a set of small orthophotomaps as the original testing data (line 04), *new-testing-data* is a set of small orthophotomaps as the new testing data (line 05), and *model* is our machine learning model (line 06).

In the initial phase, the technique of sliding window with stride s is used to obtain *original-training-data* from dividing the large orthophotomap (line 07), *original-testing-data* is set to *original-training-data* (line 08), and *new-testing-data* is obtained by augmenting *original-testing-data* by the concept of lighting variations (line 09). In the training phase, *model* is trained via only *original-training-data* (line 10). In the evaluating phase, *model* is evaluated via *original-testing-data* and *new-testing-data* (line 11). This fits the scenario where a drone can position itself within a specific area but is not limited to lighting variations.

IV. The Experiments

In this section, we start to detail our experiments, and the experimental settings are given below:

- Figure 2 shows the instances of small orthophotomaps (classifications) augmented for four seasons.

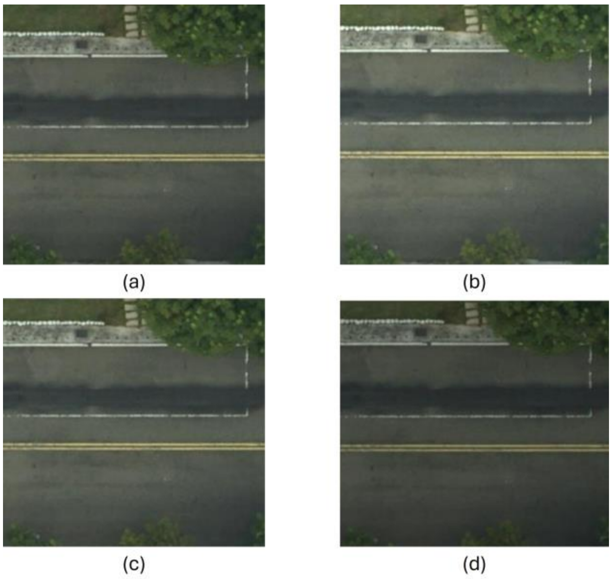

Figure 2. The instances of small orthophotomaps augmented for four seasons: (a) spring, (b) summer, (c) autumn, and (d) winter.

Table 1. The procedures of our vision-based approach.

Algorithm: The learning and testing procedures in our vision-based approach.	
input:	**initial phase:**
(01) *map*: a large orthophotomap;	(07) *original-training-data* ← **use** sliding window technique with stride *s* to divide *map*;
variables:	(08) *original-testing-data* ← *original-training-data*;
(02) *s*: stride, the distance the window moves at each step;	(09) *new-testing-data* ← **augment** *original-testing-data* by lighting variations;
(03) *original-training-data*: small orthophotomaps as original training data;	**training phase:**
(04) *original-testing-data*: small orthophotomaps as original testing data;	(10) **train** *model* via *original-training-data*;
(05) *new-testing-data*: small orthophotomaps as new testing data;	**evaluating phase:**
(06) *model*: our machine learning model;	(11) **evaluate** *model* via *original-testing-data* and *new-testing-data*;

- 16 areas are considered in our experiments, and their features are different (Figure 3).
- We divide each experimental area into 400 sub-areas (classifications).
- The resolution of small orthophotomaps in the dataset is 300*300.
- The distance the window moves at each step in practice is 0.5 meters.
- Thanks to the data augmentation by lighting variations, the scenario can describe how a drone can position itself within a specific area but is not limited to lighting variations.
- We use four critical blocks to realize our CNN models, i.e., inception block, dense block, MCDN block, and residual block [3].
- Dropout: 0.6, loss function: Cross entropy, learning rate: 0.0001, optimizer: Adam, and weight decay: $1e^{-3}$.
- The specification of the experimental computer: CPU: Intel Core i7-11370H, RAM: 16GB DDR4 3200MHz, and GPU: Nvidia GeForce RTX 3070 8GB.
- IDE (Integrated Development Environment): PyCharm [9], and API (Application Programming Interface): PyTorch [10].

During the experimental procedures, we use four critical blocks to test and realize our CNN models. However, only the design with the residual concept acquires satisfied results. For the sake of simplicity, only the design above is presented in the text. Figure 4 gives the design of our model, and Table 1 shows the positioning accuracy for all 16 sub-areas. More specifically, since our structure is utilized for image classification, we need the ability to feature extraction from high-dimensional image data. Hence, a CNN with residual-like connections via addition blocks to preserve information across layers is an effective design.

Next, Table 1 presents results for different sub-areas across four seasons: spring, summer, autumn, and winter. Each sub-area has a percentage value for each season. Most sub-areas maintain high performance across all seasons, while some decrease in the winter season, with sub-area 15 showing the most reduction. The overall pattern indicates strong stability in all areas except for a few seasonal fluctuations. Noteworthily, other designs also obtain the worst performance in the winter season, even with only 1%. Thus, the model has more difficulty positioning drones with lower brightness.

Figure 3. 16 areas are considered in our experiments.

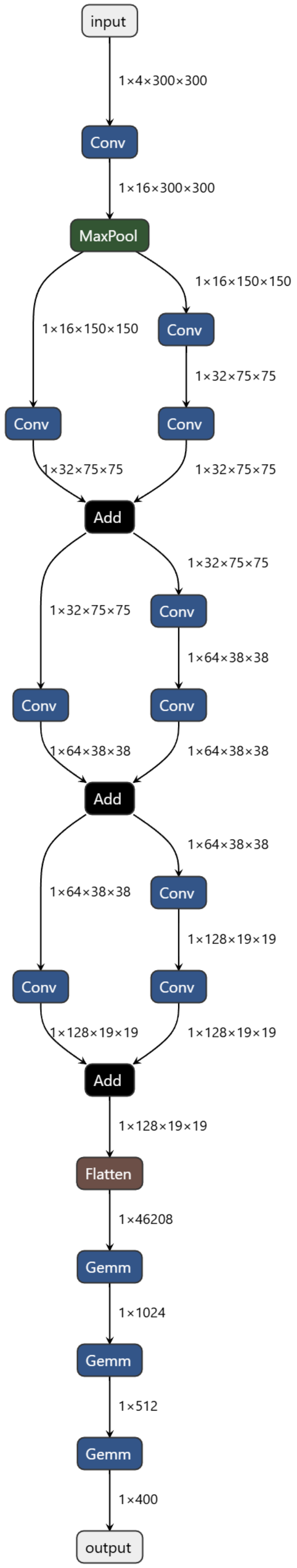

Figure 4. The design of our model.

Table 1. The positioning accuracy for all 16 sub-areas.

Sub-area \ Season	Spring	Summer	Autumn	Winter
01	100%	100%	100%	100%
02	100%	100%	100%	100%
03	100%	100%	100%	96%
04	100%	100%	100%	100%
05	100%	100%	100%	99%
06	100%	100%	100%	100%
07	100%	99%	100%	95%
08	100%	100%	100%	100%
09	100%	100%	100%	100%
10	100%	100%	100%	100%
11	100%	100%	100%	100%
12	100%	100%	100%	100%
13	100%	100%	100%	100%
14	100%	100%	100%	99%
15	100%	100%	100%	90%
16	100%	100%	100%	100%

V. Conclusions and Future Works

This paper investigates the challenges of drone visual positioning under varying seasonal lighting conditions and presents a CNN-based approach to address this issue. The experiments show that our model performs consistently well in most cases, while its accuracy decreases in lower brightness conditions (the winter season). It highlights the difficulty of positioning under low-light scenarios.

Several future directions would help improve the accuracy and applicability of visual drone positioning systems in real-world environments, e.g., incorporating additional environmental factors, improving low-light performance, larger-scale implementation, and so on.

Acknowledgment

This research is supported by National Science and Technology Council, Taiwan, R.O.C., under grant 112-2221-E-035-062-MY3.

Moreover, we extend our gratitude to SINJIE INSTRUMENT CO., LTD. for providing us with free access to the orthophotomap, which is instrumental in conducting our experiments.

References

[1] C.-C. Chang, D.-T. Lin, Y. Ikema, and Y.-M. Ooi, “Drone visual positioning algorithm with convolutional neural network,” *2023 IEEE 5th Eurasia Conference on IOT, Communication and Engineering (ECICE)*, 2023.

[2] C.-C. Chang, J. Tsai, P.-C. Lu, and C.-A. Lai, “Accuracy improvement of autonomous straight take-off, flying forward, and landing of a drone with deep reinforcement learning,” *International Journal of Computational Intelligence Systems*, Vol. 13, Iss. 1, pp. 914-919, 2020.

[3] Y. Lee, D. Jun, B.-G. Kim, and H. Lee, “Enhanced single image super resolution method using lightweight multi-scale channel dense network,” *Sensors*, Vol. 21, Iss. 10, 2021.

[4] P. R. Michael, D. E. Johnston, and W. Moreno, “A conversion guide: solar irradiance and lux illuminance,” *Journal of Measurements in Engineering*, 2020.

[5] T.-E. Hsieh, B. Fraincas, and K.-C. Chang, “Generation of a typical meteorological year for global solar radiation in Taiwan,” *Energies*, Vol. 16, Iss. 7, 2023.

[6] K. Hwang, Cloud Computing for Machine Learning and Cognitive Applications, *The MIT Press*, 2017.

[7] K. Hwang, and M. Chen, Big-Data Analytics for Cloud, IoT and Cognitive Computing, *John Wiley & Sons*, 2017.

[8] Window Sliding Technique - GeeksforGeeks [Online]. Available: https://www.geeksforgeeks.org/window-sliding-technique/.

[9] PyCharm: the Python IDE for Professional Developers by JetBrains [Online]. Available: https://www.jetbrains.com/pycharm/.

[10] PyTorch [Online]. Available: https://pytorch.org/.