drones

*Article*

# The HDIN Dataset: A Real-World Indoor UAV Dataset with Multi-Task Labels for Visual-Based Navigation

**Yingxiu Chang [1], Yongqiang Cheng [1], John Murray [2,*], Shi Huang [3] and Guangyi Shi [4]**

[1] Department of Computer Science and Technology, University of Hull, Hull HU6 7RX, UK
[2] Faculty of Technology, University of Sunderland, Sunderland SR6 0DD, UK
[3] Department of Engineering, University of Hull, Hull HU6 7RX, UK
[4] School of Software & Microelectronics, Peking University, Beijing 102600, China
* Correspondence: john.murray@sunderland.ac.uk

**Abstract:** Supervised learning for Unmanned Aerial Vehicle (UAVs) visual-based navigation raises the need for reliable datasets with multi-task labels (e.g., classification and regression labels). However, current public datasets have limitations: (a) Outdoor datasets have limited generalization capability when being used to train indoor navigation models; (b) The range of multi-task labels, especially for regression tasks, are in different units which require additional transformation. In this paper, we present a Hull Drone Indoor Navigation (HDIN) dataset to improve the generalization capability for indoor visual-based navigation. Data were collected from the onboard sensors of a UAV. The scaling factor labeling method with three label types has been proposed to overcome the data jitters during collection and unidentical units of regression labels simultaneously. An open-source Convolutional Neural Network (i.e., DroNet) was employed as a baseline algorithm to retrain the proposed HDIN dataset, and compared with DroNet's pretrained results on its original dataset since we have a similar data format and structure to the DroNet dataset. The results show that the labels in our dataset are reliable and consistent with the image samples.

**Keywords:** supervised learning; indoor visual-based navigation; real-world UAV dataset; multi-task labels; convolutional neural network (CNN); scaling factor labeling

check for updates

**Citation:** Chang, Y.; Cheng, Y.; Murray, J.; Huang, S.; Shi, G. The HDIN Dataset: A Real-World Indoor UAV Dataset with Multi-Task Labels for Visual-Based Navigation. *Drones* **2022**, *6*, 202. https://doi.org/10.3390/drones6080202

Academic Editors: Mehdi Ghommem, Jawhar Ghommam, Brahim Brahmi and Quanmin Zhu

Received: 28 June 2022
Accepted: 7 August 2022
Published: 11 August 2022

## 1. Introduction

A survey report in [1] shows the significant growth of UAV applications in commercial markets, such as precision agriculture [2], traffic monitoring [3] and warehouse management [4]. These applications predominately use visual information for model training containing sub-technologies for localization and mapping, obstacle avoidance and path planning [5]. Currently, Deep Supervised Learning, especially CNNs, is a good way to extract the specific features from visual information for corresponding tasks [6] but requires a large amount of UAV visual datasets.

Current public UAV visual datasets for CNN-based supervised learning can be divided into two broad categories, '*Object-Oriented*' and '*Navigation-Oriented*', where the former is to analyze the objects and situations in the UAV's Field-of-View (FOV). For example, the highD [7] and AU-AIR [8] datasets contain various car-type annotations (e.g., car, truck) along with their status (e.g., speed, acceleration) for ground vehicle tracking. The CADP [9] and ERA [10] datasets contain video samples collected from CCTVs on roads to train the CNN model for traffic accident prediction. Other than traffic surveillance applications, Okutama-Action [11] and Stanford Drone [12] datasets focused on predicting specific human social behaviors (e.g., hugging, handshaking, trajectory) while the UAV-GESTURE [13] dataset allowed the CNN model to recognize human gestures for corresponding control (e.g., landing, hovering, moving). Object Following is also a widely used application such as the UAV123 [14] and Hui et al. [15] datasets, respectively, which track video game characters and other UAVs from visual capture. University-1652 [16] is a cross-view geo-localization dataset which aims at globally localizing the UAV from drone view-to-satellite and satellite-to-drone view based on the collected buildings. Although parts of these '*Object-Oriented*' datasets, such as Object Following datasets, contain UAV control labels, none of them are suitable for autonomous navigation since they lack mapping between the captured image and the next moving control in unknown spaces.

In comparison, there are limited '*Navigation-Oriented*' datasets that can be used for visual-based autonomous navigation. These datasets with their corresponding CNN model can be further grouped as supervised classification or regression tasks. For example, Padhy et al. [17] applied DenseNet-161 [18] as the backbone CNN to classify the current position of FOV (i.e., Center, Left, Right or End of the corridor) where then, the control unit shifts the UAV away from the side walls to the Center of the corridor. However, the limitations of supervised classification navigation are stiff control and availability only in uncomplicated scenarios. Supervised regression navigation applies a CNN model as a regressor to map the input image to a specific data variable to achieve flexible flight control in more complicated environments. For example, Chhikara et al. [19] proposed a DCNN-GA model to regress the input image to an angle between the central axis of the corridor and the bottom of the image, while Kouris et al. [20] used a two-stream CNN model to extract the spatial-temporal features and regress to the distance-to-collision values. A ground vehicle dataset, GS4 [21], allows the end-to-end CNN model to predict the angular velocity of all wheels, i.e., predicting the next moving direction and velocity. Furthermore, as the combination of supervised classification and regression tasks, the multi-task CNN model such as DroNet [22,23] predicts the steering value and collision probability simultaneously, enabling smoother navigation control in complex spaces. Nevertheless, the common challenges of these '*Navigation Oriented*' datasets with supervised regression tasks can be summarized as below:

1. Incompatible vehicle platforms challenge: The GS4 dataset [21] was collected from a Turtlebot 2 platform, where the visual samples are low-altitude viewing and the corresponding labels are Turtlebot control (i.e., the angular velocity of wheels). These samples and labels are not suitable for the UAV platform.
2. Specific Sensor challenge: The visual samples of the GS4 dataset [21] were collected from a 360° fisheye camera, which are different from the captured images based on the common UAV monocular camera; The labels of the ICL dataset [20] are distance-to-collision, which require additionally installing three pairs of Infrared and Ultrasonic sensors and sensor fusion with time-synchronization.
3. Generalization challenge raised from label regression: Since the continuous regression labels existing data jitters, that is, large gaps between consecutive data, the CNN model cannot regress the inputs to the expected values; Unidentical data units with varying ranges such as DroNet [24] and ICL datasets [20], respectively, used steer wheel angles ($-1$~1 of radians) and distance-to-collision (0~500 cm) as training labels.

Motivated by the aforementioned challenges, we propose an HDIN UAV dataset with multi-task labels (i.e., regression task for steering, classification task for collision) for visual-based navigation collected from real-world indoor environments based on the following objectives:

1. Collecting data based on a widely available UAV platform from its original onboard sensors and simplifying the processes of multi-sensor synchronization.
2. Defining a novel scaling factor labeling method with three label types to overcome the learning challenges due to the data jitters during collection and unidentical label units.

To share our findings with the robotics and drone community, we publicly release our dataset with code as: https://github.com/Yingxiu-Chang/HDIN-Dataset, accessed on 21 July 2022.

The rest of the paper is organized as follows: Section 2 introduces the related existing '*Navigation-Oriented*' datasets along with their collection methodologies. Section 3 describes our dataset collection setup including the UAV platform and experimental environments. Section 4 introduces the details of the collection methodology containing dataset format and structure along with the scaling factor labeling method. Section 5 evaluates the dataset and Section 6 concludes the paper with contributions, limitations and potential future works.

## 2. Related Works

There exist special datasets such as Blackbird [25,26] and the Mid-Air [27] datasets that focus on improving the accuracy of Visual Positioning (e.g., Visual-SLAM and Visual Odometry) for visual-based navigation. However, they focus on localization calibration and have no UAV control commands.

The other '*Navigation-Oriented*' datasets focus on using the captured visual information to directly control the UAV's movements in unknown environments. For example, the DroNet [22,23] dataset has multi-task labels, i.e., the regression labels from the Udacity self-driving-car dataset are used for steering control and the classification labels self-collected from bicycle-riding are manually annotated as Collision and Non-collision. However, this outdoor collection leads to generalization challenges for indoor navigation tasks due to the background features being only available on public roads such as road markings (i.e., "Line-like pattern" dependence). Moreover, specific label units such as steering wheel angles are generally not available for UAV platforms.

The ICL dataset [20] specifically installed three pairs of infrared and ultrasonic sensors on a UAV heading towards $[-30^\circ, 0^\circ, 30^\circ]$ of the FOV. While remotely controlling UAV flight in corridors, complicated sensor fusion with time-synchronization is also required to match each visual sample with three distance-to-collision values ranging from 0~500 cm.

The GS4 dataset [21] from Stanford University captured front and back fisheye images and combined them into a large figure as training samples. During collection, the associated linear and angular velocities of the ground vehicle Turtlebot 2 for the regression task are represented by the angular velocity of all wheels. The collections were conducted in complicated environments with dynamic obstacles such as libraries and laboratories, but the low-altitude fisheye viewing was different from the common UAV's onboard RGB cameras.

We summarize key features of the existing '*Navigation-Oriented*' datasets with our proposed HDIN dataset in Table 1 for intuitive comparison.

**Table 1.** Navigation-Oriented Datasets.

| Datasets | | ICL [20] | DroNet [22,23] | GS4 [21] | Our HDIN |
|---|---|---|---|---|---|
| Vehicles | Collected | UAV | Car [24] and Bicycle | UGV [a] | UAV |
| | Applied | UAV | UAV | UGV [a] | UAV |
| Environments | | Real indoor | Real outdoor | Sim [b] and Real indoor | Real indoor |
| Samples | | Front RGB | Front Gray and RGB | 360° fisheye | Front RGB |
| Labels | | $[-30^\circ, 0^\circ, 30^\circ]$ distances | Steer wheel angle; Collision label [c] | UGV [a] control | UAV's orientation; Collision label [c] |
| Characteristics | | Sensor customized installation, fusion and time-sync [d]. | "Line-like" pattern dependence. | Incompatible vehicle platforms and sensors. | Common UAV platform and onboard sensors. |

[a] UGV, Unmanned Ground Vehicle. [b] Sim, Simulated collection environments. [c] Collision label, Manual-designed collision label. [d] time-sync, Time-synchronization.

"Vehicle" includes Collected and Applied platforms, "Environments" indicates outdoor/indoor and real/simulated collection environments simultaneously, "Samples" represents the viewing direction and image types, "Labels" shows the corresponding labels to the image samples, and the "Characteristics" are summarized based on their common features.

## 3. Data Collection Setup

### *3.1. UAV Platform*

Figure 1 shows the UAV collection framework used in this paper.

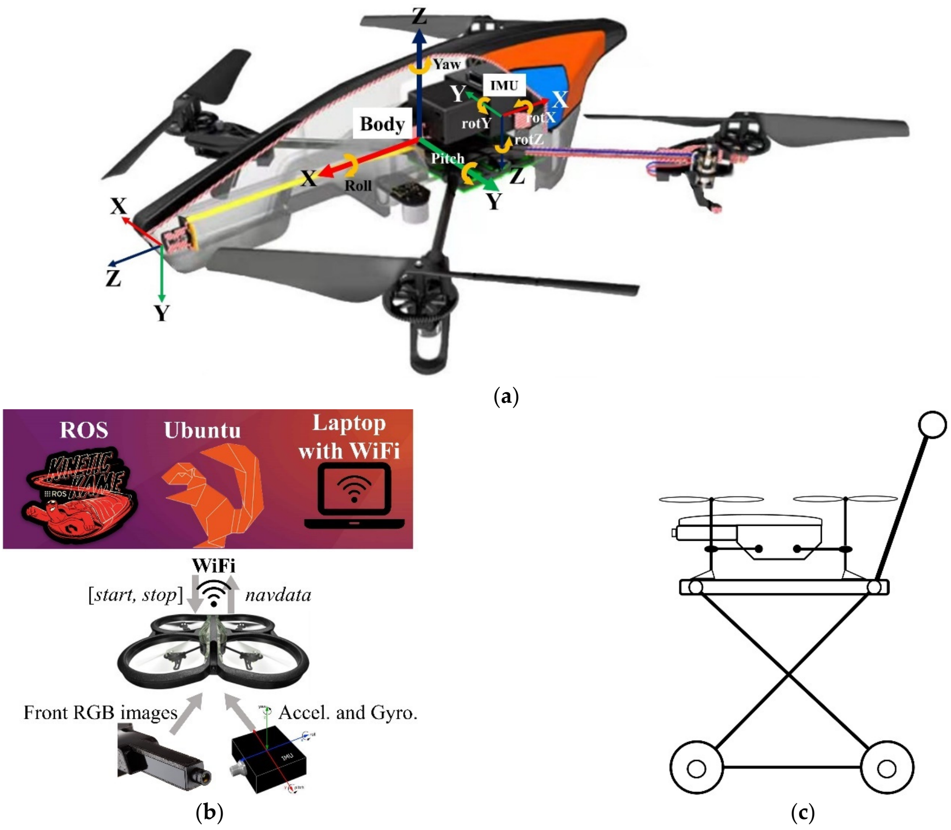


**Figure 1.** UAV collection framework. (**a**) The UAV with onboard sensors; (**b**) The overall framework; (**c**) An illustration diagram of the UAV's placement during data collection.

Figure 1a is the perspective view of the open-source Parrot AR.Drone 2.0 (https://www.parrot.com/us/support/documentation/ar-drone, accessed on 21 July 2022) and various sensors along with their coordinates. The red, green and blue axes in all coordinate systems represent the X, Y and Z axes, respectively. The Front 92° wide-angle lens Camera (720 p, 30 fps) captures the visual information. The onboard embedded device carries a MEMS-based IMU which consists of a 3-axis accelerometer ($\pm$0.05 g, g = 9.8 m/s$^2$) and a 3-axis gyroscope ($\pm$2000°/s), along with an extra 3-axis magnetometer shown as the *IMU coordinate*. Based on the left-hand coordinate system, the axes point towards the positive direction of the acceleration while the yellow arrows of rotX, rotY and rotZ (i.e., rotation X, rotation Y and rotation Z) indicate the positive rotation direction based on the left-hand rotation rule. Different from the *IMU coordinate,* the axes of the rigid *Body coordinate* based on the right-hand principle head toward the positive direction of the UAV's movement. The yellow arrows on the *Body coordinate* show the positive rotation direction along the X, Y and Z axes, that is Roll, Pitch and Yaw of the UAV. According to Figure 1a, the value of rotZ of the *IMU coordinate* is a normalized yaw angle ($\psi$) from the north provided by the magnetometer sensor and its positive rotation direction is the same as the Yaw of UAV rigid body. Therefore, rotZ can be used as the orientation of the UAV's movements during data collection.

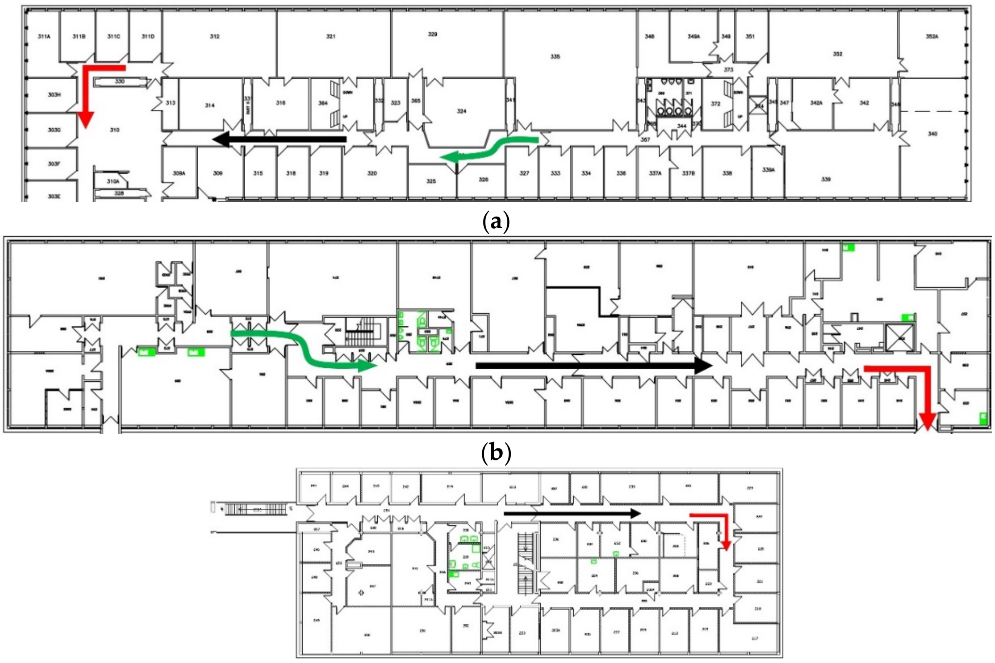
The Parrot AR.Drone is equipped with a WiFi device. In Figure 1b, the UAV exchanges the collected data (i.e., RGB images and *navdata*) and the UAV's controls commands (i.e., [*start, stop*]) between the laptop and the UAV via WiFi connection. Based on the ardrone_autonomy package (http://wiki.ros.org/ardrone_autonomy, accessed on

21 July 2022), the [*start, stop*] commands, respectively, power on and off the UAV to activate or deactivate the onboard sensors. The sequential RGB images are captured from the front camera as visual samples and the rotZ of *navdata* from IMU indicating the UAV's orientation is recorded as the original unprocessed labels simultaneously.

For purposes of security, smoothing steering and imitating the flight status during collection, the UAV was placed on the flat plate of a wheeled trolley as shown in Figure 1c. The relative height of the trolley plate to the ground is 0.7 m.

### 3.2. Experimental Environment

According to our observation and summaries, most buildings in the University of Hull campus are regular where the corridor components are straight with fixed-angle turning such as L-shape corners. We selected three different commonly used buildings and across multiple floors in the campus. Figure 2 shows the example floor plans in the different buildings used, where the straight corridors are shown as black arrows.



(**a**)

(**b**)

(**c**)

**Figure 2.** Example floor plans. (**a**) Building 1; (**b**) Building 2; (**c**) Building 3.

The corners can be divided into L-shape corners with one steering (red arrows) and S-shape corners with two continuous steering paths (green arrows). Depending on the UAV's movement direction, the L-shape corner contains L-shape left and right steering while the S-shape corner has S-shape left–right and right–left steering.

As it is the first version of the HDIN dataset, the initial sets take a risk assessment into consideration, where the corridors were kept with sufficient illumination and free of obstacles. We will be adding more to this dataset with more complex scenarios such as insufficient illumination and loop-circle corridors as we progress. Another noteworthy point concerns specific corridor components such as T-shape corners which can be regarded as two L-shape steering paths with different directions (i.e., L-shape left and right). Navigating a UAV towards the left or right at a T-shape corner relies on the objective of path planning such as information gain based on self-exploration, which means requiring an extra layer as a global planner for decision making.

## 4. Collection Methodology

Similar to the DroNet [22,23] dataset, the steering and collision subsets are collected and respectively seen as regression and classification prediction. The image samples in the steering subset are labeled by the scaling factor. It ignores the absolute values with various units such as absolute steering wheel angles (e.g., range of [−90°, 90°]) of DroNet [22,23] and distance-to-collision (e.g., cm) of ICL which both require appropriate transformation for corresponding applications, but uses proportions to indicate the identical steering intensity in the range of [−1, 1].

Since the scaling factor labeling method converts the original data (i.e., rotZ of *navdata* from the IMU) to steering intensity, it is necessary to estimate the accumulative steering errors. The UAV was placed heading toward a wall and recorded steering values ten times consecutively during three complete rotations (1080°) positively and negatively for accumulative error estimation. The results are shown in Table 2.

**Table 2.** Results of Steering Errors.

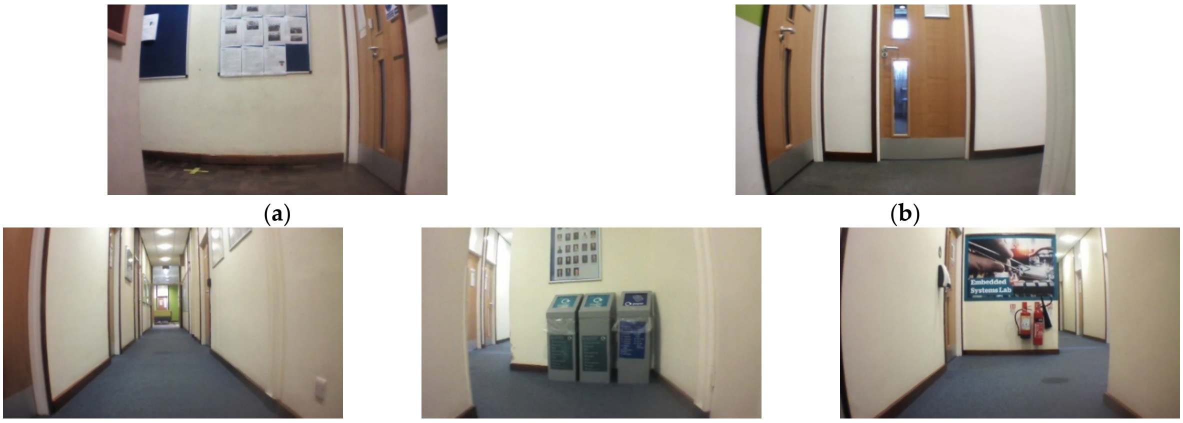
| | *MAE* [a] | *UE* [b] | *UA* [c] |
|---|---|---|---|
| Positive rotation | 11.352° | 0.0105° | 98.95% |
| Negative rotation | 4.154° | 0.0038° | 99.62% |

[a] Mean Absolute Error (*MAE*) of the 10-time three complete rotations (1080°), $MAE = \frac{1}{n}\sum_{i=1}^{n}\left|S_l - S_f\right|$, where $S_l$ and $S_f$ are respectively the last and the first steering values of each time, $n = 10$. [b] Unit Error (*UE*) of 1° rotation, that is $UE = \frac{MAE}{1080}$. [c] Unit Accuracy (*UA*) of 1° rotation, that is $UA = \frac{(1-UE)}{1}100\%$.

Though the *UE* of steering values is small and has sufficient *UA*, the accumulative steering errors caused by long-term rotations still need to be minimized. Moreover, the original data jitters lead to large numerical gaps between similar corresponding images, which may result in learning difficulties. Therefore, the following Section 4.1 will introduce the segmented collection inspired by ICL [20] and the scaling factor labeling method with Polynomial Fitting and Low-pass Filter to smooth data jitters.

### 4.1. Steering Subset

In comparison with collecting data along the entire trajectory such as the DroNet [22,23] dataset, the method of the segmented collection which just places the UAV in front of turning corners proposed by ICL has two benefits: (1) shortens the time required for data collection; (2) contains only one or two turnings within the range of [−90°, +90°], which both effectively reduce the accumulative steering errors. As discussed in Section 3.2, images were collected along with steering values based on the corridor components of L-shape left, L-shape right, straight forward, S-shape left–right and S-shape right–left shown in Figure 3.



(**a**) (**b**)

(**c**) (**d**) (**e**)

**Figure 3.** (**a**) The entrance of L-shape left steering; (**b**) The entrance of L-shape right steering; (**c**) Straight forward; (**d**) The entrance of S-shape left–right steering; (**e**) The entrance of S-shape right–left steering.

A total of five different corridor components were identified in three campus buildings and the UAV was initially placed at the positions shown in Figure 3a–e, then manually controlled with the trolley passing through these bends and straight corridors. The UAV simultaneously records the front sequential images at 30 fps and the original unprocessed steering values at 200 Hz without time-synchronization. The image naming convention adopted in our dataset is timestamps (19-bit of a nanosecond) while the original unprocessed steering values with corresponding timestamps are stored in the label text file.

The following Algorithm 1 describes the scaling factor labeling method with three label types to process original unprocessed steering values.

**Algorithm 1:** Scaling Factor Labeling Method.

**Label type 1: Expected Steering**

**Input**: Steering label text, Image path
**Output**: Synchronize steering text

```
$[L_{ts},\ S]^{n*2}$ ← Load Steering label text;
$[I_{ts}.jpg]^{m*1}$ ← Load images from Image path;
for $i = 1$ to $m$ do
    $[S_m]^{m*1}$ ← Matching($[L_{ts},\ S]^{n*2}$, $[I_{ts}.jpg]^{m*1}$);
$[S_t]^{m*1}$ ← Transformation($[S_m]^{m*1}$);
for $i = 1$ to $m$ do
    $S_{e_i} = S_{t_{(i+1)}} - S_{t_{(i)}}$;
$[\theta]^{m*1}$ ← Low-pass filter($[S_e]^{m*1}$);
Output $[\theta]^{m*1}$ to Synchronize steering text;
```

**Label type 2: Fitting Angular Velocity**

**Input**: Steering label text, Image path
**Output**: Synchronize steering text

```
$[L_{ts},\ S]^{n*2}$ ← Load Steering label text;
$[I_{ts}.jpg]^{m*1}$ ← Load images from Image path;
$[S_t]^{n*1}$ ← Transformation($[S]^{n*1}$);
$\left[S_f\right]^{n*1}$ ← Fitting($[S_t]^{n*1}$);
$[S_d]^{n*1}$ ← Derivative ($\left[S_f\right]^{n*1}$);
$[S_m]^{m*1}$ ← Matching($[L_{ts},\ S_d]^{n*2}$, $[I_{ts}.jpg]^{m*1}$);
$[\theta]^{m*1}$ ← deg2rad($[S_m]^{m*1}$);
Output $[\theta]^{m*1}$ to Synchronize steering text;
```

**Label type 3: Scalable Angular Velocity**

**Input**: Steering label text, Image path
**Output**: Synchronize steering text

```
$[L_{ts},\ S]^{n*2}$ ← Load Steering label text;
$[I_{ts}.jpg]^{m*1}$ ← Load images from Image path;
$[S_t]^{n*1}$ ← Transformation($[S]^{n*1}$);
$\left[S_f\right]^{n*1}$ ← Fitting($[S_t]^{n*1}$);
$[S_d]^{n*1}$ ← Derivative ($\left[S_f\right]^{n*1}$);
$[S_m]^{m*1}$ ← Matching($[L_{ts},\ S_d]^{n*2}$, $[I_{ts}.jpg]^{m*1}$);
$[\theta]^{m*1}$ ←$[S_m]^{m*1}$/max angular velocity;
Output $[\theta]^{m*1}$ to Synchronize steering text;
```

In all the above algorithms, the $L_{ts}$ and $I_{ts}$ respectively represent the current timestamp of steering values and images. The variables containing '$S$' are all related to steering values, where the subscripts indicate the specific processed steering values such as $S_m$, $S_t$, $S_f$ and $S_d$ which were obtained from Time Matching, Data Transformation, Polynomial Fitting and

First Derivative, respectively. The $S$ without a subscript represents the original unprocessed steering values. Furthermore, $\theta$ in different algorithms is the processed correct label that will be written into the steering text corresponding with the image.

The Expected Steering is motivated by the DroNet steering which records the steering wheel angles during car driving. The steering wheel angle of each image indicates the desired moving direction of the vehicle based on the current status. After loading the original Steering text and images, Label type 1 firstly chooses the steering value with the timestamp that is larger but closest to the timestamp of each image for Matching. Secondly, because the original steering values are the absolute angle from north provided by the magnetometer within the range of $[-180°, 180°]$ as previously mentioned in Section 3.1, each steering value is transformed based on Equation (1) by setting the first one as $S_0 = 0°$. $S$ and $S_t$ are the untransformed and the transformed steering values, respectively.

$$S_r = S - S_0 \begin{cases} S_t = S_r + 360; if\ S_0 > 0\ and\ S_r < -180 \\ S_t = S_r - 360;\ if\ S_0 < 0\ and\ S_r > 180 \\ S_t = S_r; others \end{cases} \tag{1}$$

Moreover, the expected steering values $S_e$ are calculated as step 7 and finally, we used a low-pass filter to smooth the fluctuations of sequential $S_e$ as Equation (2) where $\alpha = 0.1$ and $i \in (1,\ 2,\ 3\ldots, m)$.

$$\theta_i = (1-\alpha)S_{e_{i-1}} + \alpha S_{e_i} \tag{2}$$

Label type 2 mostly focuses on obtaining angular velocity for scaling factor steering. The Transformation and the Matching processes are same as the ones in Label type 1. The original steering values after Matching are processed by Polynomial Fitting which helps to smooth the steering fluctuations and then, using the First Derivative to obtain the angular velocity. The Time-Matching steering values are finally converted to radians.

The Scalable Angular Velocity is similar to the one in Label type 2 with the only difference being that the Time-Matching steering values are converted to the ratio of maximum angular velocity in step 7 (we set the default as 40°/s).

### *4.2. Collision Subset*

The collision subset collection is located far away from an obstacle (>0.5 m) and stops when it is very close ($\leq$0.5 m). The sequential images can be manually annotated: the frames far away from collision are annotated as 0 (non-collision), and frames close to collision as 1 (collision). Here, we set up the distance threshold to the front obstacle between non-collision and collision as approximately 0.5 m. The following Figure 4 shows the example samples in the collision subset.

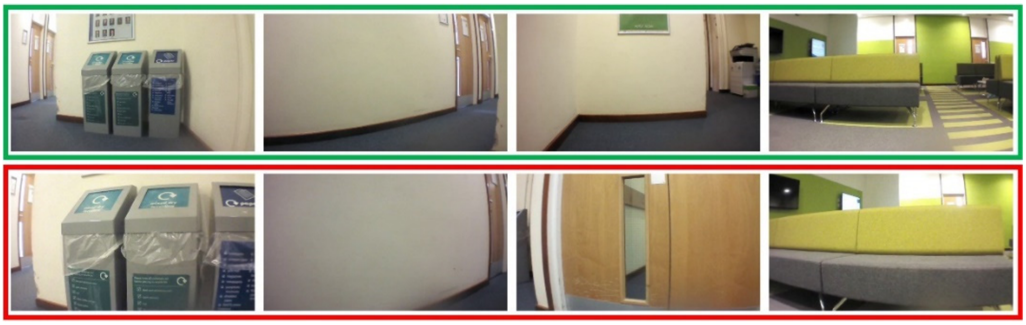
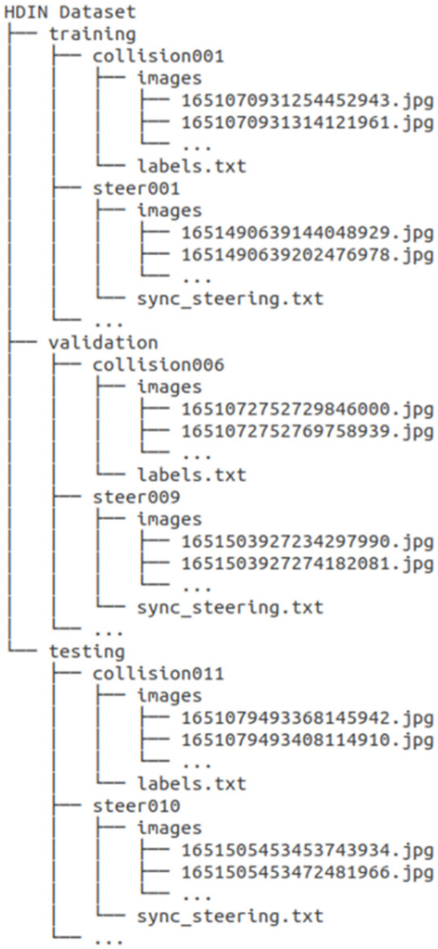


**Figure 4.** The example images of non-collision (green box, top row) and collision (red box, bottom row).

### *4.3. Dataset Structure*

The steering subset contains five components (i.e., L-shape left, L-shape right, straight forward, S-shape left–right and S-shape right–left) in three different buildings, which have

36 different backgrounds. Each component randomly selects at least two different backgrounds to re-collect data for validation and testing, respectively, here are six backgrounds for validation and testing, respectively. The collision subset contains 15 random different backgrounds. Each background is defined as a trajectory, so there are 48 trajectories for steering and 15 trajectories for collision (63 trajectories in total). The HDIN dataset structure is organized as shown in Figure 5 which separates the steering and collision subsets into training, validation and testing for the evaluation experiment. Each folder of collision or steer contains RGB images and their corresponding labels where the collision labels (i.e., label.txt) indicate the collision probability and the steering labels (i.e., sync_steering.txt) are scaling factor steering from Section 4.1.



```
HDIN Dataset
├── training
│   ├── collision001
│   │   ├── images
│   │   │   ├── 1651070931254452943.jpg
│   │   │   ├── 1651070931314121961.jpg
│   │   │   └── ...
│   │   └── labels.txt
│   ├── steer001
│   │   ├── images
│   │   │   ├── 1651490639144048929.jpg
│   │   │   ├── 1651490639202476978.jpg
│   │   │   └── ...
│   │   └── sync_steering.txt
│   └── ...
├── validation
│   ├── collision006
│   │   ├── images
│   │   │   ├── 1651072752729846000.jpg
│   │   │   ├── 1651072752769758939.jpg
│   │   │   └── ...
│   │   └── labels.txt
│   ├── steer009
│   │   ├── images
│   │   │   ├── 1651503927234297990.jpg
│   │   │   ├── 1651503927274182081.jpg
│   │   │   └── ...
│   │   └── sync_steering.txt
│   └── ...
└── testing
    ├── collision011
    │   ├── images
    │   │   ├── 1651079493368145942.jpg
    │   │   ├── 1651079493408114910.jpg
    │   │   └── ...
    │   └── labels.txt
    ├── steer010
    │   ├── images
    │   │   ├── 1651505453453743934.jpg
    │   │   ├── 1651505453472481966.jpg
    │   │   └── ...
    │   └── sync_steering.txt
    └── ...
```

**Figure 5.** Dataset structure.

## 5. Dataset Evaluation

The open-source DroNet [22,23] (https://github.com/uzh-rpg/rpg_public_dronet, accessed on 21 July 2022) (https://rpg.ifi.uzh.ch/dronet.html, accessed on 21 July 2022) network with Multi-task labels has been selected as the baseline model to evaluate the proposed HDIN dataset by comparing the results with the provided pretrained DroNet network. DroNet has high-correlation experiments for indoor navigation which the shared-weights CNN regresses the steering values and classifies Collision or Non-collision simultaneously shown as Figure 6. The rest of this section will respectively illustrate the accuracy and consistency of values in the HDIN dataset based on quantitative comparison and data distribution visualization for regression and classification.

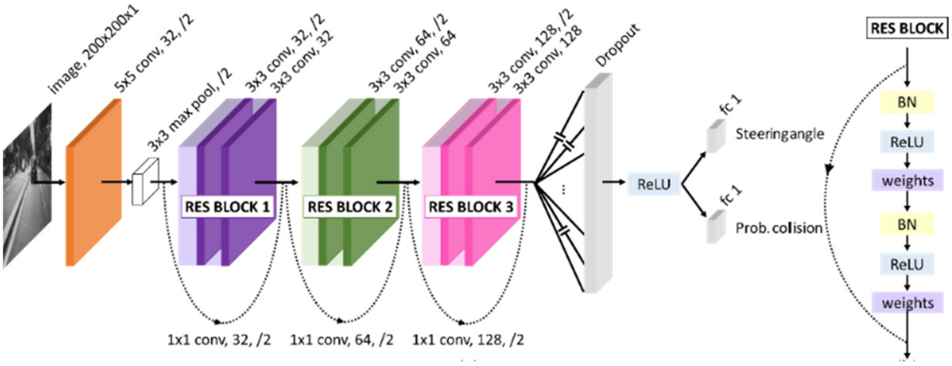


**Figure 6.** DroNet architecture [22].

### *5.1. Quantitative Comparison*

The DroNet network was retrained on the HDIN dataset, and compared with the existing pretrained DroNet network. The results in Table 3 are evaluated using the testing sequences of DroNet and HDIN datasets.

**Table 3.** Quantitative Dataset Comparison.

| Dataset | Label Type | Img Type | *EVA* [a] | RMSE | Ave. Accuracy | *F-1* Score [b] |
|---|---|---|---|---|---|---|
| DroNet [22,23] | Steer wheel angle | Gray | 0.737 | 0.110 | 95.3% | 0.895 |
| HDIN (Ours) | Expected steering | RGB | 0.778 | 0.193 | 84.9% | 0.784 |
| | | Gray | 0.798 | 0.184 | 88.2% | 0.822 |
| | Fitting angular velocity | RGB | 0.808 | 0.090 | 86.7% | 0.804 |
| | | Gray | 0.810 | 0.089 | 86.2% | 0.799 |
| | Scalable angular velocity | RGB | 0.853 | 0.113 | 85.3% | 0.789 |
| | | Gray | 0.827 | 0.123 | 85.8% | 0.794 |

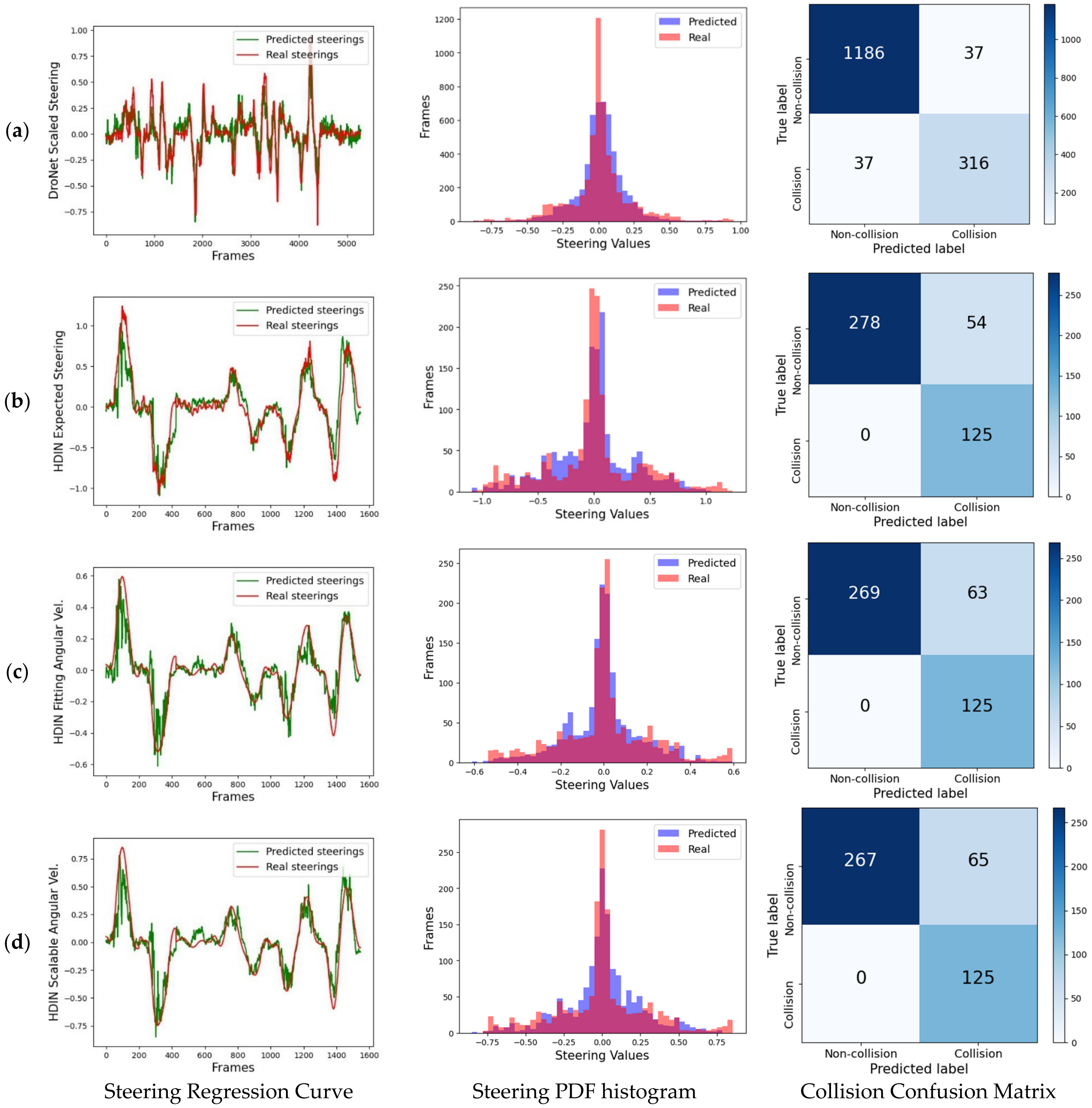
[a] Explained Variance (*EVA*), used to quantify the quality of regression at the same variance level, defined as $EVA = \frac{Var[y_{true} - y_{pred}]}{VAR[y_{true}]}$. [b] *F-1* score, used to evaluate the quality of classification at the same variance level, defined as $F\text{-}1 = 2\frac{precision \times recall}{precision + recall}$.

The DroNet network is a Multi-task model which contains the regression for steering and the classification for collision probability. Although the Root Mean Square Error (RMSE) is used as the common evaluation metric of regression tasks, *EVA* is still required to validate the variance ratio at the same level since the fluctuations of samples in different datasets are diverse. From Table 3, one can observe that the image type (RGB or Grayscale) does not affect the performance and three different label types present similar regression (*EVA* = 0.815 ± 0.037) which also outperforms the regression performance based on the DroNet dataset.

The classification task of the DroNet network predicts the collision probability within the range of [0, 1] and the threshold of collision probability is 0.5. We assessed the average accuracy and the *F-1* score of our dataset, and even though they are smaller than the DroNet dataset, maintain a considerable classification performance.

### *5.2. Data Distribution Visualization*

Since the image type does not affect performance and the pretrained DroNet is based on gray images, we select the direct outputs from Gray image rows in Table 3 to draw the following graphs as Figure 7.



Steering Regression Curve Steering PDF histogram Collision Confusion Matrix

**Figure 7.** (**a**) Provided Pretrained DroNet; (**b**) HDIN Expected steering; (**c**) HDIN Fitting angular velocity; (**d**) HDIN Scalable angular velocity.

The rows (Figure 7a–d) represent the testing steering and collision results, respectively, based on the DroNet original dataset and HDIN dataset with three label types (i.e., Expected Steering, Fitting Angular Velocity and Scalable Angular Velocity). The first column draws the curves of predicted and real steering values to visually show the steering regression performance. The second column visually displays the data distribution as the probability density function (PDF) of predicted vs. real steering values. The third column is the Confusion Matrices which are used to illustrate the classification performance of collision prediction.

The predicted steering values fluctuate along the real steering values with acceptable RMSE shown in the first column figures of Figure 7, but one can observe that the large steering values are hard to be learned and have more fluctuations. The predicted steering values cannot be directly used to control the UAV since they are the scaling factor of the steering intensity. Therefore, a constant maximum steering value is required such as maximum steering velocity. Moreover, a customized smoothing function such as a Low-pass filter in DroNet [22,23] which takes the $t-1$ predicted steering values into consideration is better.

The Collision Confusion Matrices in Figure 7 show that the retrained DroNet network based on the HDIN dataset achieved 100% accurate classification for collision samples, but misclassified non-collision samples as collision samples more than the pretrained DroNet network. The potential reason relies on the collection. Since the HDIN dataset indoor collection moves the UAV slower than the car-driving of the original DroNet dataset, the differences at the edge between the collision and non-collision samples are less obvious.

Table 4 quantitatively shows the data distribution according to Figure 7. The first column indicates the Datasets (i.e., DroNet or HDIN) with their corresponding label types. We used the Mean, Variance (Var) and Range to show the distribution of the steering label types from DroNet and HDIN datasets, where both our HDIN datasets and the DroNet datasets have similar mean steering values in all three label types (i.e., a baseline of 0.0067, close to 0 means).

**Table 4.** Data Distribution.

| Dataset with Label Types | Steering Subset | | | Collision Subset | |
|---|---|---|---|---|---|
| | Mean | Var | Range | Collision | Non-Collision |
| DroNet with Steer wheel angle | 0.0067 | 0.045 | $(-0.87, 0.94)$ | 77.6% | 22.4% |
| HDIN with Expected steering | $-0.0027$ | 0.167 | $(-1.07, 1.24)$ | 72.6% | 27.4% |
| HDIN with Fitting angular velocity | $-0.0017$ | 0.042 | $(-0.52, 0.60)$ | 72.6% | 27.4% |
| HDIN with Scalable angular velocity | $-0.0024$ | 0.087 | $(-0.75, 0.85)$ | 72.6% | 27.4% |

Moreover, the Expected Steering label type in HDIN dataset has the largest variance (i.e., 0.167) which means more difficulty for regression by the CNN model. Despite this large variance, the 0.798 *EVA* of HDIN dataset with Expected Steering label type in Table 3 outperformed the baseline 0.737 *EVA* of the DroNet dataset, where *EVA* normalizes the different variances into the same level. This indicates the labels in our HDIN dataset are more consistent and trainable than DroNet dataset. The collision subset includes Collision and Non-collision samples, where their corresponding proportions are shown in the final two columns.

## 6. Conclusions

In this paper, we proposed a real-world indoor UAV dataset along with its collection methodology and scaling factor labeling method for visual-based navigation. Our HDIN dataset compensates for the current public datasets with real-world indoor data, and further benefits the generalization capability of autonomous navigation using supervised learning. The Data Collection Setup indicates that the data can be collected without bespoke sensor installation while the Collection Methodology proposes the scaling factor labeling method with three label types, i.e., Expected steering, Fitting angular velocity and Scalable angular velocity, which overcome the challenges of data jitters and unidentical steering labels. The Dataset Evaluation evidenced that our dataset is valid for training visual-based UAV autonomous navigation networks.

The limitations of the datasets: (1) The samples do not have a vast diversity of backgrounds in various corridor components; (2) The Multi-task labels in HDIN dataset only enable the UAV to navigate in the 2D plane (i.e., collision classification for adjusting forward speed and steering regression for correcting direction), which does not efficiently take advantage of UAV accessibility in 3D space. Our future research will firstly expand the dataset with scaling factor shifting and altitude-rising labels in more buildings with dynamic obstacles such as human volunteers, not just enriching the backgrounds but also increasing the current 2-DOF control of the DroNet network (i.e., moving forward and yawing) to 4-DOF control (i.e., moving forward, yawing, shifting and rising altitude in 3D spaces). Secondly, the DroNet network retrained on the HDIN dataset requires field tests on the real UAV and environments.

**Author Contributions:** Data curation, Y.C. (Yingxiu Chang) and S.H.; Formal analysis, Y.C. (Yingxiu Chang); Investigation, Y.C. (Yingxiu Chang); Methodology, Y.C. (Yingxiu Chang), Y.C. (Yongqiang Cheng) and G.S.; Project administration, Y.C. (Yongqiang Cheng) and J.M.; Resources, Y.C. (Yongqiang Cheng), J.M. and G.S.; Software, Y.C. (Yingxiu Chang); Validation, Y.C. (Yingxiu Chang), J.M. and S.H.; Visualization, S.H.; Writing—original draft, Y.C. (Yingxiu Chang); Writing—review & editing, Y.C. (Yongqiang Cheng), J.M. and G.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by [the China Scholarship Council (CSC)] 202008010003.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Acknowledgments:** The authors gratefully acknowledge the financial support from the China Scholarship Council (CSC).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Schroth, L. The Drone Market 2019–2024: 5 Things You Need to Know. 2019. Available online: https://www.droneii.com/the-drone-market-2019-2024-5-things-you-need-to-know (accessed on 21 July 2022).
2. Daponte, P.; De Vito, L.; Glielmo, L.; Iannelli, L.; Liuzza, D.; Picariello, F.; Silano, G. A review on the use of drones for precision agriculture. IOP Conference Series: Earth and Environmental Science. In Proceedings of the 1st Workshop on Metrology for Agriculture and Forestry (METROAGRIFOR), Ancona, Italy, 1–2 October 2018; Volume 275.
3. Khan, N.A.; Jhanjhi, N.; Brohi, S.N.; Usmani, R.S.A.; Nayyar, A. Smart traffic monitoring system using Unmanned Aerial Vehicles (UAVs). *Comput. Commun.* **2020**, *157*, 434–443. [CrossRef]
4. Kwon, W.; Park, J.H.; Lee, M.; Her, J.; Kim, S.-H.; Seo, J.-W. Robust Autonomous Navigation of Unmanned Aerial Vehicles (UAVs) for Warehouses' Inventory Application. *IEEE Robot. Autom. Lett.* **2020**, *5*, 243–249. [CrossRef]
5. Lu, Y.; Xue, Z.; Xia, G.-S.; Zhang, L. A survey on vision-based UAV navigation. *Geo-Spat. Inf. Sci.* **2018**, *21*, 21–32. [CrossRef]
6. Carrio, A.; Sampedro, C.; Rodriguez-Ramos, A.; Campoy, P. A Review of Deep Learning Methods and Applications for Unmanned Aerial Vehicles. *J. Sens.* **2017**, *2017*, 3296874. [CrossRef]
7. Krajewski, R.; Bock, J.; Kloeker, L.; Eckstein, L. The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (Itsc), Maui, HI, USA, 4–7 November 2018; pp. 2118–2125.
8. Bozcan, I.; Kayacan, E. AU-AIR: A Multi-modal Unmanned Aerial Vehicle Dataset for Low Altitude Traffic Surveillance. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 8504–8510.
9. Shah, A.P.; Lamare, J.; Nguyen-Anh, T.; Hauptmann, A. CADP: A Novel Dataset for CCTV Traffic Camera based Accident Analysis. In Proceedings of the 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand, 27–30 November 2018; pp. 1–9.
10. Mou, L.; Hua, Y.; Jin, P.; Zhu, X.X. ERA: A Data Set and Deep Learning Benchmark for Event Recognition in Aerial Videos [Software and Data Sets]. *IEEE Geosci. Remote Sens. Mag.* **2020**, *8*, 125–133. [CrossRef]
11. Barekatain, M.; Marti, M.; Shih, H.-F.; Murray, S.; Nakayama, K.; Matsuo, Y.; Prendinger, H. Okutama-Action: An Aerial View Video Dataset for Concurrent Human Action Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 2153–2160.
12. Robicquet, A.; Sadeghian, A.; Alahi, A.; Savarese, S. *Learning Social Etiquette: Human Trajectory Understanding in Crowded Scenes*; Springer International Publishing: Berlin/Heidelberg, Germany, 2016; pp. 549–565.
13. Perera, A.G.; Law, Y.W.; Chahl, J. UAV-GESTURE: A Dataset for UAV Control and Gesture Recognition. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops 2018, Munich, Germany, 8–14 September 2018; Volume 11130, pp. 117–128.

14. Hui, B.; Song, Z.; Fan, H.; Zhong, P.; Hu, W.; Zhang, X.; Lin, J.; Su, H.; Jin, W.; Zhang, Y.; et al. Dataset for Infrared Image Dim-Small Aircraft Target Detection and Tracking under Ground/Air Background(V1). Science Data Bank. 2019. Available online: https://www.scidb.cn/en/detail?dataSetId=720626420933459968&dataSetType=journal (accessed on 21 July 2022).
15. Mueller, M.; Smith, N.; Ghanem, B. A Benchmark and Simulator for UAV Tracking. In Proceedings of the Computer Vision-Eccv 2016 Pt I 2016, Amsterdam, The Netherlands, 11–14 October 2016; Volume 9905, pp. 445–461.
16. Zheng, Z.; Wei, Y.; Yang, Y. University-1652: A Multi-view Multi-source Benchmark for Drone-based Geo-localization. In Proceedings of the 28th ACM International Conference on Multimedia, Seattle, WA, USA, 12–16 October 2020; pp. 1395–1403.
17. Hirose, N.; Xia, F.; Martin-Martin, R.; Sadeghian, A.; Savarese, S. Deep Visual MPC-Policy Learning for Navigation. *IEEE Robot. Autom. Lett.* **2019**, *4*, 3184–3191. [CrossRef]
18. Kouris, A.; Bouganis, C.S. Learning to Fly by MySelf: A Self-Supervised CNN-based Approach for Autonomous Navigation. In Proceedings of the 2018 IEEE/Rsj International Conference on Intelligent Robots and Systems (Iros), Madrid, Spain, 1–5 October 2018; pp. 5216–5223.
19. Udacity. An Open Source Self-Driving Car. 2016. Available online: https://www.udacity.com/self-driving-car. (accessed on 21 July 2022).
20. Padhy, R.P.; Verma, S.; Ahmad, S.; Choudhury, S.K.; Sa, P.K. Deep neural network for autonomous uav navigation in indoor corridor environments. *Procedia Comput. Sci.* **2018**, *133*, 643–650. [CrossRef]
21. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
22. Chhikara, P.; Tekchandani, R.; Kumar, N.; Chamola, V.; Guizani, M. DCNN-GA: A deep neural net architecture for navigation of UAV in indoor environment. *IEEE Internet Things J.* **2020**, *8*, 4448–4460. [CrossRef]
23. Loquercio, A.; Maqueda, A.I.; del-Blanco, C.R.; Scaramuzza, D. DroNet: Learning to Fly by Driving. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1088–1095. [CrossRef]
24. Palossi, D.; Conti, F.; Benini, L. An Open Source and Open Hardware Deep Learning-powered Visual Navigation Engine for Autonomous Nano-UAVs. In Proceedings of the 2019 15th International Conference on Distributed Computing in Sensor Systems (Dcoss), Santorini, Greece, 29–31 May 2019; pp. 604–611.
25. Antonini, A.; Guerra, W.; Murali, V.; Sayre-McCord, T. The Blackbird UAV dataset. *Int. J. Robot. Res.* **2020**, *39*, 1346–1364. [CrossRef]
26. Antonini, A.; Guerra, W.; Murali, V.; Sayre-McCord, T.; Karaman, S. The Blackbird Dataset: A Large-Scale Dataset for UAV Perception in Aggressive Flight. In Proceedings of the 2018 International Symposium on Experimental Robotics, Buenos Aires, Argentina, 5–8 November 2018; Volume 11, pp. 130–139.
27. Fonder, M.; Van Droogenbroeck, M. Mid-Air: A multi-modal dataset for extremely low altitude drone flights. In Proceedings of the 2019 IEEE/Cvf Conference on Computer Vision and Pattern Recognition Workshops (Cvprw 2019), Long Beach, CA, USA, 16–17 June 2019; pp. 553–562.