AI Wings: An AIoT Drone System for Commanding ArduPilot UAVs

Kuan-Ting Lai, *Member, IEEE*, Yueh-Tsung Chung, Jun-Jia Su, Chien-Hung Lai*, Member, IEEE*, and Yu-Hsuan Huang

***Abstract*—Recent advances in drone and artificial intelligence (AI) technologies have enabled many innovative applications, such as package delivery, reconnaissance, and search and rescue, to name a few. In this article, we propose AI wings, an artificial intelligence of things (AIoT) drone system for commanding multiple unmanned aerial vehicles and deploying AI models. We integrated ArduPilot with the Android mobile platform, which equips DIY drones with AI computing power and 4G/5G connectivity. Embedded control software is developed to cooperate with the AI Wings cloud. Users can easily convert ArduPilot drones into AIoT drones using Android phones, and connect to a cloud server to create their own Internet of Drones. Our cloud server is also integrated with the drone simulation software AirSim for simulating drone missions in virtual reality (VR) worlds. The virtual simulation enables users to test software/hardware configurations as well as train AI models. Moreover, to ensure secure communication, we propose an authentication protocol based on elliptic-curve cryptography with pseudoidentities and time freshness check. In summary, AI Wings provides a cloud server for commanding drone fleets securely, software/hardware design for AIoT drones, and VR simulation for training and testing AI models. Users can install the AI models on the drones directly. To test the system, we built an experimental medical drone service, which delivers an automated external defibrillator to people with a sudden cardiac attack in the shortest time possible.**

***Index Terms*—Cellular networks, elliptic-curve cryptography (ECC), embedded deep learning, Internet of Things (IoT), unmanned aerial vehicles (UAVs).**

I. Introduction

UNMANNED aerial vehicles (UAVs), a.k.a. drones, have been improved rapidly in recent years, especially multirotor drones. Compared to traditional fixed-wing airplanes and helicopters, multirotor drones are more stable and have better control over all axes. The features make drones better for loitering and carrying, and have created many new applications such as drone photography [1], art performance [2], reconnaissance [3], search and rescue [4], [5], etc. In order to perform more tasks automatically, drones need to be more intelligent, i.e., running powerful deep learning algorithms. Meanwhile, drones should be able to communicate with a cloud server to receive commands and report status. One solution is to combine artificial intelligence (AI) and Internet of Things (IoT) to create AI of Things (AIoT) drones. This trend is best described by Cameron from IEEE Computer Society: “Drones: the New Flying IoT” [6].

Specifically, we refer to drones that can run real-time deep learning algorithms as AI drones, and the AI drones that can connect to the Internet as AIoT drones. The first challenge of building an AI drone regards drone hardware. Most AI drones are based on the NVIDIA Jetson family of modules [7], which contain powerful embedded AI processors. However, high computation means high power consumption and cost, which make Jetson modules unsuitable for small drones. The other solution is Qualcomm Flight Pro [8], which is based on the Qualcomm Snapdragon mobile platform [9] and provides less AI computation power but more integrated sensors including Wi-Fi, Bluetooth, and GPS. This solution can be used to build very lightweight autonomous drones. Nonetheless, both solutions require deep knowledge of hardware to integrate the embedded system modules with the drones.

To convert an AI drone into an AIoT drone, it is necessary to install additional modules for connecting to the Internet remotely. Considering transmission distance and bandwidth, 4G/5G is currently the best choice. There are a few commercial solutions including ModalAI [10] and XBStation [11]. However, the cost of one module is almost higher than a basic drone because the demand of drone 4G/5G modules is low. Moreover, some of the products may have long lead times, be out of stock, and therefore, difficult to acquire.

Once we have built AIoT drones, the next challenge is to efficiently command a group of drones. The most convenient method is to utilize IoT technology and connect all the drones to the Internet. Once again, current solutions are mostly commercial services. The service providers include small startups such as DroneCloud [12], SkyDrone [13], AltitudeAngel [14], 3DR [15], Auterion [16], etc. Those companies provide application programming interface (API) and software development toolkit (SDK) for users to connect their drones to the cloud with monthly fees. However, commercial services use proprietary protocols that cannot be modified. There is a need for an open-source drone cloud that allows researchers to study advanced algorithms.

Manuscript received 12 January 2022; revised 17 April 2022 and 2 June 2022; accepted 3 July 2022. Date of publication 21 July 2022; date of current version 8 June 2023. This work was supported by the Ministry of Science and Technology, Taiwan, under Grant MOST 109-2221-E-027-080. *(Corresponding author: Kuan-Ting Lai.)*

The authors are with the Department of Electronic Engineering, National Taipei University of Technology, Taipei City 10608, Taiwan (e-mail: ktlai@ntut.edu.tw; t109368006@ntut.org.tw; t104360071@ntut.org.tw; ee12245@ntut.edu.tw; t109c51534@ntut.org.tw).

Digital Object Identifier 10.1109/JSYST.2022.3189011

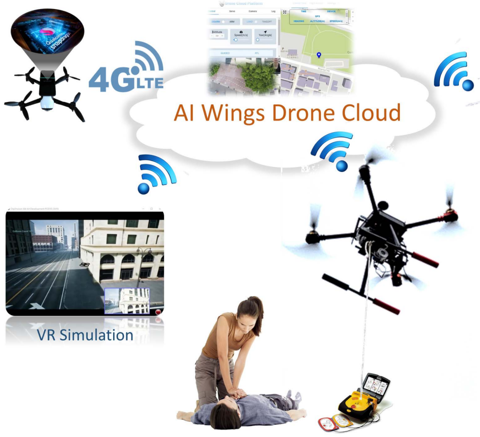


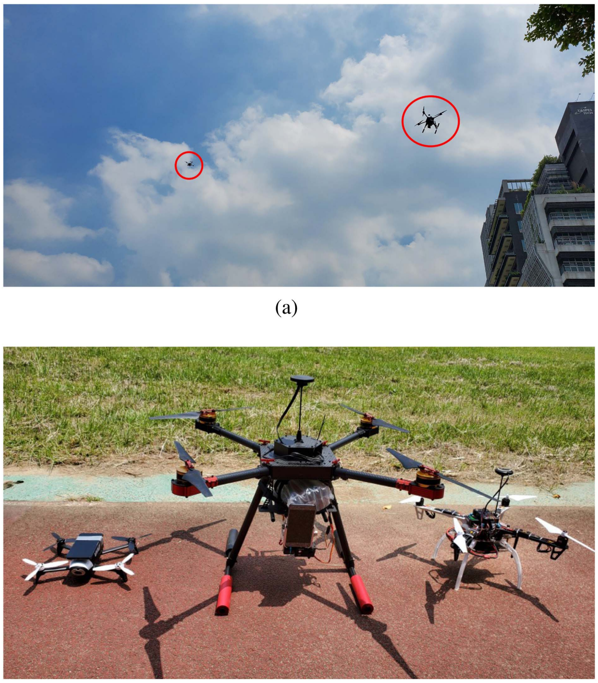
Fig. 1. Overview of AI Wings. We have created a complete AIoT drone system including a cloud server, embedded hardware/software of AIoT drones, drone authentication, and VR simulation for training and testing AI models.

The last challenge is deploying AI models on real drones. Due to strict constraints on weight and power, drones have limited computing capability, which makes it difficult to port deep learning models to drones. Different hardware specifications and operating system environments make this task even more daunting.

To address the challenges mentioned previously, we developed AI Wings, a complete drone system that enables users to create and command their own AIoT drone fleet. AI Wings has the following features: a cloud server for commanding drone fleets securely, software/hardware design of AIoT drones, and virtual reality (VR) simulation for training and testing AI models. The system architecture of AI Wings is shown in Fig. 1. Our system consists of four major components: the drone cloud, the embedded system module (ESM), the embedded software, and VR simulation. Our embedded software cooperates with ArduPilot using the MAVLink protocol [17] via the UART connection. Users can easily convert ArduPilot-based drones into AIoT drones using our embedded software and hardware reference design.

The key contributions of AI Wings are listed as follows.

1) *Drone cloud:* A cloud server with a broker has been created to control a group of AIoT drones. It supports command and control, video streaming, and flight planning on Google Map and MapBox [18]. Users can install the server on their own cloud to manage a drone fleet.
2) *Embedded hardware:* We leverage the Android mobile platform as our ESM, which provides neural network hardware accelerators and supports 4G/5G connection. The ESM cooperates with ArduPilot-compatible microcontroller unit (MCU). Any DIY UAV can be converted into an AIoT drone for as little as 100 USD and 100 grams in extra weight.
3) *Embedded software:* Embedded control software has been developed to control the drone via ArduPilot, run AI models, stream video, and communicate with the drone cloud.
4) *VR simulation:* AI Wings cloud server supports Microsoft AirSim [19], the most popular drone simulation software. A user is able to simulate the risky tasks or train/test AI models in VR worlds.
5) *AI model deployment:* The models trained using TensorFlow Lite can be directly installed on our AIoT drones, since we use Android as our embedded OS. We have run MobileNet SSD [20] for object detection on our drones at 30+ FPS using Snapdragon 855 as onboard ESM.


(a)

(b)

Fig. 2. AI Wings can convert various drones into AIoT drones. The AIoT drones built by the authors are shown in (b). The dimensions from left to right are 328 × 328, 800 × 800, and 450 × 450 mm. (a) Two drones flown and controlled by AI Wings cloud. (b) AIoT drones built with off-the-shelf mobile platforms.

To test the AI Wings system, we built three different AIoT drones to test our system with dimensions ranging from 328 × 328, 450 × 450 to 800 × 800 mm. The photos of the drones are shown in Fig. 2. Field experiments demonstrated that AI Wings can successfully command and control different AIoT drones based on our modules.

In order to enhance the security of our system, we adopted the latest Internet-of-Drones (IoD) authentication [21] and adapted it to meet practical applications. The protocol is based on elliptic-curve cryptography (ECC) with pseudoidentities of the user and the drones, which can resist most common attacks including user impersonation, man-in-the-middle, password guessing, denial-of-service (DoS), etc.

Furthermore, we used AI Wings to build an emergency delivery service for automated external defibrillators (AED). An AED is a portable device used for treating sudden cardiac arrest. As the brain will suffer permanent damage in 3–5 min after the heart stops beating, the time of first aid to a patient is extremely important. Therefore, we have built an AED service based on AI Wings to deliver AEDs in the shortest possible time.

In summary, AI Wings is a complete AIoT drone system for commanding multiple UAVs. Users can create AIoT drones using ArduPilot, Android phones and our embedded software, and run our server code to manage their own AIoT drone fleet. Furthermore, users can simulate the functions of their drones in VR worlds, and use the synthetic image data to train AI models. Researchers can leverage AI Wings to study new drone protocols or train and test new deep learning models.

II. Previous Work

Multirotor drone technology has developed rapidly during the past decade. Due to the advances in artificial intelligence, more innovative applications have been created. Such innovations include automatic packet delivery [22], [23], video analytics [24], remote action recognition [3], etc. Hernandez *et al.* [25] proposed to use AI algorithms to assess the natural disasters in drone images. Hao *et al.* [4] utilized a drone swarm to perform search and rescue missions. Zhu *et al.* [26] created the VisDrone dataset, which annotated over 2.6 million object bounding boxes in drone videos. Li *et al.* [27] created a large-scale drone action recognition dataset called UAV-Human with 67 428 video sequences. The AI algorithms of computer vision can also facilitate the aerial controls and create better autonomous drones. Weinstein *et al.* [28] developed a visual odometry system to control a drone swarm. Unlike previous methods that require fiduciary references points or GPS, visual odometry relies only on camera and inertia sensors. This approach needs a lot of computation power. The other popular research topic is training neural networks for drone autonomous navigation. Researchers have been actively studying how to use deep reinforcement learning (DRL) to learn control policies in virtual worlds and apply in the real world. This technique is called virtual-to-real learning [29]. Sadeghi *et al.* [30] proposed learning drone indoor collision avoidance in virtual environments created by computer-aided design (CAD) tools. Experiment results showed that the AI models trained purely in virtual rooms can be applied to real drone navigation, as long as there is enough randomness in the simulation. The experiments of Tobin *et al.* [31] demonstrated that object detectors trained using simulated RGB images can be accurate to 1.5 cm in the real world. Lai *et al.* [32] have built a large-scale virtual environments to train action recognition models for drones. Other drone DRL applications include obstacle avoidance [33], drone racing [34], etc.

Besides AI algorithms, another important research direction is to integrate drones with IoT technology. By connecting all drones to the Internet, the owner can easily command and control several drones to perform different tasks. This technology is also called Internet-of-Drones [35]. Gharibi *et al.* [36] introduced an Internet-of-Drones model including five layers: airspace, node-to-node, end-to-end, service, and application. Abualigah *et al.* [37] provided a good review of recent IoD applications. In fact, a drone cannot only utilize IoT technology, but also play an assisting role in the IoT network. As a result, the Internet of UAVs over cellular networks have become an emerging research topic. Bassoli and Granelli [38] proposed to use drones for rapid deployment of 5G services. Meng *et al.* derived a mathematical model that can optimize the 3-D trajectory of UAVs to maximize the transmission data of the ground devices [39]. Gao *et al.* [40] proposed to utilize the game theory for developing a cooperative scheme for wireless-powered device-to-device (D2D) networks.

The security of IoT network is another important research topic. To enhance the security of a drone communication network, Chaudhry *et al.* [41] proposed a novel user access control scheme for distributed IoT devices, which uses a unique key for each communication session, and thus, needs no key pairing function. Jia *et al.* [42] developed an identity-based anonymous authentication scheme for mobile edge computing, which completes mutual authentication in single message exchange round. Dao *et al.* [43] proposed a new framework called FOGshield, which can defend distributed denial-of-service (DDoS) attacks against IoT devices. In terms of the drone network security, Hussain *et al.* [21] proposed using ECC-based authentication scheme, and combine it with password, biometric keys, and pseudo identity. Ge *et al.* [44] proposed a provenance-aware distributed trust model for the network security of UAVs.

The key component of the IoT drone automation system is the drone cloud. Current solutions are mostly commercial services. The service providers include small startups such as DroneCloud [12], SkyDrone [13], AltitudeAngel [14], the drone pioneer 3DR [15], and Auterion founded by Pixhawk's creator [16], to name a few. Some of the companies mentioned previously provide API and SDK for users to connect their drones to the cloud with monthly fees. In addition to basic fleet management, some companies provide other functions including unmanned traffic management (UTM) [14], flight planning [12]–[14], [16], mapping, and 3-D modeling [15], [16]. Commercial services are not open-source and cannot be modified. There is a need for an open-source drone cloud system for advanced researches such as drone security, fleet management, drone AI algorithms, etc.

III. Architecture of AI Wings

In this section, we elaborate the architecture of AI Wings. The major components of our system include drone hardware, embedded software, drone cloud, and VR simulation. The system block diagram is illustrated in Fig. 3. A DIY AIoT drone is built by using an ArduPilot microcontroller and an off-the-shelf Android phone. Our embedded software runs on top of the Android operating system and cooperates with ArduPilot and the drone cloud. The drone cloud server supports video streaming and a message broker for multidrone communication. The details of each component are introduced in the following subsections. The source code has been released on GitHub (https://github.com/kuanting/aiwings).

A. Embedded Hardware

Building drones requires specialized hardware knowledge. It is a nontrivial task to create a hardware module with 4G connectivity and AI computation while meeting the strict limits of the weight and power consumption of drones. There are two major solutions for embedded deep learning computation on drones. One solution is based on NVIDIA Jetson family of modules [7]. The Jetson modules are the most powerful

TABLE I
COST ANALYSIS OF THE COMMERCIAL OR CUSTOM DRONES THAT HAVE DEEP LEARNING ACCELERATORS

Drones	Flight Controller	AI module	4G/5G module	Camera	Gimbal	Other Parts	size(mm)	Total (USD)
DJI M300 RTK	Proprietary	$*1280* (Jetson TX2)	N/A	Proprietary		Proprietary	895×895	≈ $*8,380*
Skydio 2	Proprietary	Built-in (Jetson TX2)	N/A	Proprietary		Proprietary	223×273	$*1,349*
ModalAI m500	VOXL: Pixhawk + Snapdragon 821		Optional	Proprietary		Proprietary	480×480	$*1,999*
ModalAI (DIY)	VOXL: Pixhawk + Snapdragon 821 ($*599*)		LTE add-on ($*299*)	≈ $200	≈ $100	≈ $100	480×480	≈ $1,298
Jetson TX2 (DIY)	Pixhawk (≈$250)	$*299* (Jetson TX2)	XBLink [45] ($*635*)	≈ $200	≈ $100	≈ $100	450×450	≈ $1,584
AI Wings (DIY)	Pixhawk (≈$250)	≈$100∼$300 (Snapdragon 675 - 860 + LTE)		≈ $200	≈ $100	≈ $100	450×450	≈ $750∼$950

The component cost of DIY drones is based on F450. The cost numbers in italics represent retail prices in 2022.

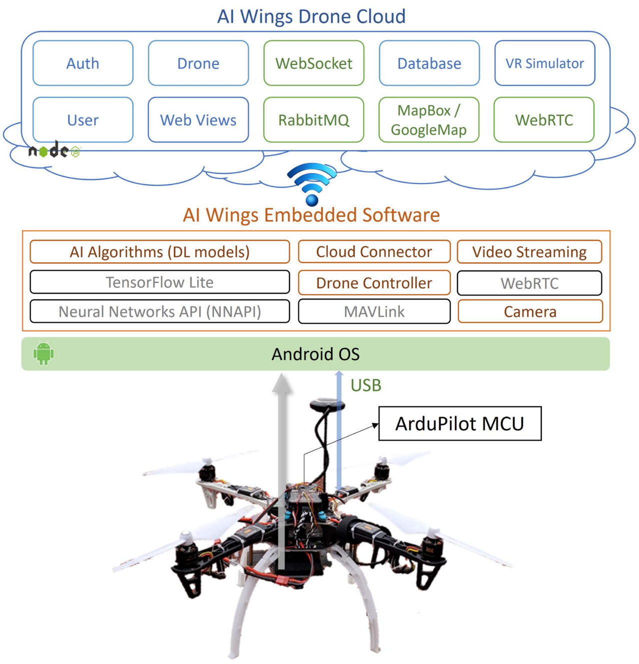


Fig. 3. System architecture of AI Wings on a DIY F450 drone. The off-the-shelf Android mobile platform is attached in the bottom of the main frame, and an ArduPilot-based MCU (Pixhawk) is installed on the top.

embedded AI processors to date, as shown in the benchmark created by Suzen *et al.* [46]. The other solution is Qualcomm Flight Pro [8], which provides less AI computation power but more integrated sensors and lower power consumption.

The aforementioned solutions both require extra cost of expensive modules and drone hardware knowledge to assemble. To address those issues, we propose to use off-the-shelf mobile platforms, a.k.a., mobile phones, as our embedded modules. There are several advantages in using off-the-shelf mobile phones. First, the latest mobile phones support 4G/5G connectivity and AI computation acceleration. Second, commercial mobile phones are optimized for price and weight and can be installed on a drone easily. Third, users can freely choose from low-end to high-end mobile phones. In AI Wings, we choose Snapdragon and Android OS because this combination has better supports for deep learning computation, but other hardware platforms can also be employed as long as they support Android OS and deep learning frameworks.

Table I lists the BOM cost of various AI drones in 2022. We selected the drones that support running custom deep learning models. The first one is the Matrice-300 (M300) RTK made by the leading drone company DJI [47]. M300 is a large industrial-grade drone with a 895-mm diagonal size, 55 min of flight time, and high precision RTK GPS. DJI created a Jetson TX2 companion onboard module for Matrice series for developers, which costs around $1 280. The total cost of DJI M300 with AI capability could be as high as $8 380. The other commercial AI drone is the Skydio 2 [48], which has a compact size (223×273 mm) with a built-in Jetson TX2 module. The drone contains advanced collision avoidance and tracking algorithms and sold at $1 349. However, the system of Skydio 2 is proprietary and cannot be modified. Another solution is ModalAI m500 [49], which uses open-source Pixhawk [50] as the flight controller and uses Qualcomm Snapdragon 821 as the onboard AI module. The preassembled m500 is sold at $1 999. Users can also buy a bundle of Pixhawk and Snapdragon module at $599, and assemble their own drones. The drones mentioned previously do not have 4G modules except for ModalAI, which provides an LTE upgrade option at $299. DJI M300 and Skydio 2 have a proprietary control system that are not designed for modification. Besides the ModalAI LTE module, the other solution is XBLink, which includes a 1-year 4G subscription and is sold at $635 [45]. The implementation costs are proportional to the complexity of the drone. It takes a drone expert about 2 working days to build a F450 AIoT drone and 5 working days for an AED drone. In terms of AI Wings drones, we use a Pixhawk controller and F450 drone DIY frame kit as a reference design. The cost of Pixhawk 4 with a bundle of sensors is around $250 including GPS and power management modules. Other basic components of a drone are propellers, motors, electronics speed control (ESC), frame, battery, wires, and screws, which cost around $100. This estimate may vary due to the quality of different components. The costs of off-the-shelf Qualcomm mobile phones range from $109 for Snapdragon 675 (Samsung Galaxy A70) to $265 for Snapdragon 860 (Xiaomi Poco X3 Pro) in 2022. Generally, our DIY AIoT drone costs around 30%∼40% less than the cheapest AI drone on the market.

B. Embedded Software

After the hardware specifications are determined, the next step is to develop embedded software that cooperates with the drone controller and the cloud. The block diagram of our software is illustrated in Fig. 4. Our software runs on Android. There are four main modules: drone control, cloud connector, video streaming, and AI module. The drone control module communicates with an ArduPilot controller via an USB cable using the MAVLink protocol [17]. MAVLink is the de facto standard

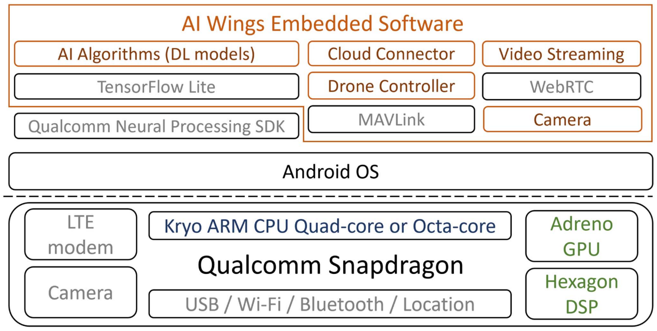

AI Wings Embedded Software
AI Algorithms (DL models)
Cloud Connector
Video Streaming
TensorFlow Lite
Drone Controller
WebRTC
Qualcomm Neural Processing SDK
MAVLink
Camera
Android OS
LTE modem
Kryo ARM CPU Quad-core or Octa-core
Adreno GPU
Qualcomm Snapdragon
Camera
USB / Wi-Fi / Bluetooth / Location
Hexagon DSP

Fig. 4. Software stack of AI Wings embedded software.

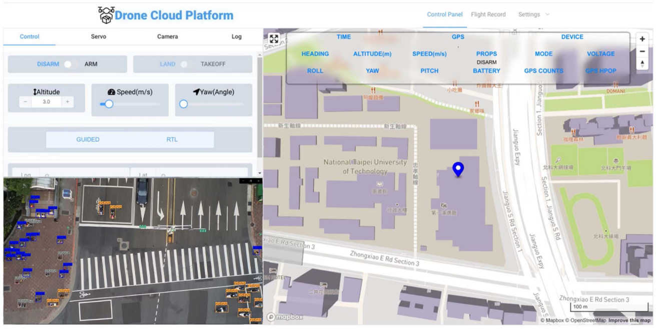


Fig. 5. User interface of AI Wings cloud. The right panel is the map, the upper left panel is the drone control UI, and lower left panel is the video stream from the drone.

drone control protocol. It is a lightweight protocol that contains both publish-subscribe and point-to-point design pattern. The primary mission of the drone control module is to assign a flight destination, report drone status, and perform specific tasks such as dropping cargo. The cloud connector is in charge of login authorization and server-drone communication. It is responsible for summarizing the data received by the drone control module and reporting the current status to the cloud. Meanwhile, the cloud server handles high level commands, such as traveling to specific location, dropping cargo, etc. We chose transmission control protocol (TCP) as our underlying transmission protocol for reliability. In terms of video streaming, we have tested several open-source projects and selected the Google's Web Real-Time Communication (WebRTC) [51] as our underlying library. WebRTC has the lowest average latency during our 4G streaming test. The deep learning models run on top of TensorFlow Lite library, which leverages the Qualcomm neural processing SDK to accelerate the deep learning computation via Android neural networks API (NNAPI). The Snapdragon Hexagon DSP can run quantized 8-bit integer models, while Ardeno GPU can run 16-bit and 32-bit models.

In the past, the robotic community preferred to use Linux as operating system and run the robot operating system (ROS). However, more robots have selected Android as the underlying OS [52]. The main reasons are that the reliability of Android has been significantly improved during recent years, and it is much easier to develop new applications. One open-source project that has a few features similar to our embedded software is Android DroneKit by 3DR [53]. However the last update of this project was in 2017. The key difference is that other Android drone software takes the Android system as an independent ground control station (GCS), while our software takes the system as an embedded module controlled by a cloud server.

C. Drone Cloud

We have built a cloud server to control the AIoT drones. Each user can install the source code to create a private cloud, or register drones on the AI Wings cloud. Users can also save flight GPS records and videos, and review them after completing a mission. The cloud server has been built using node.js [54], which is a highly efficient framework. The user interface (UI) of the cloud is shown in Fig. 5. The browser is divided into three panels. The right panel is the MapBox map service [18], which is a powerful map service containing 3-D information. The upper left panel is the drone control UI. The users can use the cloud to plan flight route, assign missions, monitor current positions, etc. Our cloud supports many functions of the most popular drone control tool MissionPlanner [55]. The user can set the drone to several default modes including Loiter, Guided, Land, and Return To Launch (RTL) mode, or add their own functions into the source code. The cloud does not support modes that requires real-time controls such as ACRO, Stabilize and Altitude Hold (AltHold) modes. The lower left panel in Fig. 5 shows the video stream from the drone. Users can leverage real-time video streaming to perform manual remote control using a browser. We employed a broker to let users command and control all their drones. RabbitMQ is currently selected as the default broker [56]. We utilize the topic mode to broadcast commands to a group of drones.

Theoretically, 4G/5G technology enables remote control from anywhere with network coverage. However, general drone speed is around 20 m/s, and 4G Internet latency is around 30–50 ms. The 4G network delay can lead to 0.6–1 m difference between the real and remote position. This issue makes it difficult to do real-time control from the cloud. Therefore, the collision avoidance and autonomous flight abilities are important for IoT drones. Another solution is to use 5G technology, which has much faster transmission speed and lower latency. However, the transmission range of 5G is shorter, and the service coverage is still low. As a result, 4G is still a better choice for general drone tasks. Nevertheless, our embedded module can support both 4G and 5G automatically according to the network conditions.

D. VR Simulation

The drone community has been using software simulator to test and verify drone configurations. Shah *et al.* [19] have developed a drone simulator called AirSim based on the Unreal Engine, which is one of the most advanced 3-D engines. Meanwhile, AI researchers started to utilize 3-D engine to generate visual training data [32] because deep learning requires a large amount of training data. Sadeghi *et al.* [30] successfully trained an autonomous drone model for real world using only synthetic images generated by CAD. Lai *et al.* [32] created several large virtual scenes for training drones to perform different tasks autonomously using deep reinforcement learning.

TABLE II
FMEA OF THE AI WINGS SYSTEM

	Item	Failure mode	Mission phase	Probability	Severity	Detection	Mitigation
1	Cloud Server	Crash or lose connection	Flight	2	3	2	Copters can fly autonomously
2	Networking	internet connection Lost	Flight	5	1	4	Copters can fly autonomously
3	Embedded SW	Software malfunction	Flight	2	3	2	Switch to ArduPilot return-to-launch mode
4	Motor / ESC	Copter loses some power	Flight	1	5	2	Copters with 6+ motors can fly without 1 motor
5	Battery	Copter loses all power	Flight	2	5	2	Use two batteries in parallel
6	GPS	No GPS, use IMU	Flight	1	4	2	Use inertial navigation or mobile phone's GPS
7	IMU	No inertial navigation	Flight	1	3	2	Copters can use GPS navigation
8	Compass	Lose direction. Use GPS	Flight	1	3	2	ArduPilot supports multiple compasses
9	Barometer	Lose altitude information	Flight	1	3	2	ArduPilot supports multiple barometers

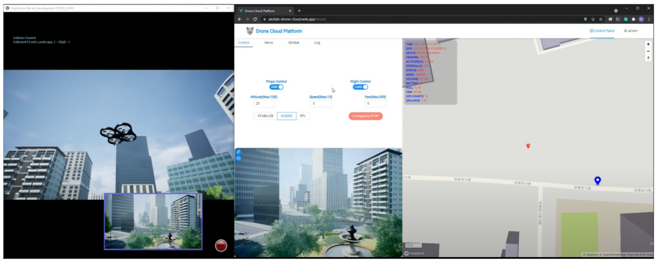


Fig. 6. Our system supports Microsoft AirSim drone VR simulation. The left window shows the city environment created by the Unreal engine. The subwindow below is the drone camera view. The right window is the cloud UI, and the lower left corner shows the videos streamed from the VR world.

TABLE III
RATING OF THE PROBABILITY (P), SEVERITY (S), AND DETECTION (D) IN OUR FMEA

Rating	Probability	Severity	Detection
1	Remote	Trivial	Certain: fault can be caught on test
2	Unlikely	Minor	Almost certain
3	Possible	Moderate	High
4	Likely	Serious	Low
5	Frequently	Fatal	Fault is undetected by operators

Sultani *et al.* [57] proposed to leverage VR environments to train models for recognizing human actions in drone videos. Other virtual-to-real applications including autonomous driving [58], robot hands [59], drone racing [34], etc. Fig. 6 demonstrates the drone VR simulation with our cloud server. We believe that VR simulation will become more and more important for both drone software test and AI model training in the future.

E. Failure Mode and Effects Analysis (FMEA)

We have performed a FMEA on the AI Wings system. The analysis focused on the flight phase and the results are shown in Table II. We classify the failure occurrence probability (P), severity (S), and ease of detection during testing (D) into five levels. The ratings of P, S, and D are shown in Table III.

There are four main sources of failure: cloud server, networking, embedded software, and drone hardware components. The cloud server may crash while drones are on missions. In this case, the drones can continue executing missions autonomously using the onboard ESM and ArduPilot. After completing the missions, the drones will return to the base automatically. However, users lose the control of their drones and cannot perform tasks that require human control. Therefore, the severity of server crashes is considered to be level 3: Moderate. Network connection issues are common causes of failure. We expect the connectivity problems to happen frequently. Since our drones can execute missions autonomously and reconnect to server when network is restored, we consider the severity of network issues to be level 1: Trivial. As network quality varies in each region, the detection of networking issue is considered to be low. Another failure item is embedded software. Our embedded software may crash or Android OS may fail. In this case, the drone will need to stop some missions. Fortunately, the drones can still fly back to home using ArduPilot GPS navigation, so we considered the severity to be level 3: Moderate.

In terms of drone hardware, the failure of motor/ESC will cause drone to lose fly ability, which is a fatal problem. The drone with more than six motors can tolerate loss of one or more motors, which can mitigate the issue. Battery failure is the most severe problem because it can cause drone to lose all the power. To alleviate this problem, we can use two batteries in parallel in case one of the them fails. Another important component is GPS. ArduPilot relies on GPS for accurate navigation. ArduPilot also supports inertial navigation using inertial measurement unit (IMU) and extended Kalman filter (EKF) algorithm. However, the inertial navigation is inaccurate and position deviation is large. Therefore, GPS failure is serious and the severity level is 4. The drone can still try to fly back to home using inertial navigation without GPS. In addition, we can use the GPS of the Android mobile platform when the drone GPS fails. This feature makes our AIoT drone more reliable. On the other hand, the failure of the IMU is trivial because the drone can fly with GPS, so the severity level is set to 2. The rest two components are compass and barometer. Compass failure causes loss of direction, while barometer failure leads loss of height information. The ArduPilot-based Pixhawk MCU supports multiple sensors to prevent one of them fails. Therefore, the failure probability of compass and barometer are very low and considered to be level 1: Remote. The failure severity of IMU, compass, and barometer are all considered as level 3 (Moderate) in our system.

IV. DRONE AUTHENTICATION PROTOCOL

Security is the major concern of Internet-of-drones, as hijacking drones can cause serious threats. We adopted the asymmetric cryptosystem to protect user and drone information. Specifically, we chose ECC [60] and developed a secure protocol based on [21]. ECC is considered as the next generation of public key cryptography because it can provide stronger security than

TABLE IV
NOTATION

Symbols	Description
U_i, D_j, C	The user i, drone j and cloud
k_{pvt}	Private key of the cloud C, a random integer.
G	Generator point (G_x, G_y) on the EC curve
P_{pub}	Public key of C, $P_{pub} = k_{pvt} * G$
$E_p(a, b)$	Elliptic curve function over the finite field $\mathbb{F}_p$
$*$	EC multiplication
$\|\|, \oplus$	Concatenation operation, XOR operation
$\stackrel{?}{=}$	Check if left-hand side equals right-hand side
$h(\cdot)$	One-way hash function [61]
UID_{D_j}	UUID of the drone's embedded mobile platform
ID_{U_i}, PID_{U_i}	Real ID and pseudo-ID of the i-th user
ID_{D_j}, PID_{D_j}	Real ID and pseudo-ID of the j-th drone
$PID^C_{U_i}$	Pseudo user ID encrypted by the cloud
$PID^C_{D_j}$	Pseudo drone ID encrypted by the cloud
$ID^*_{D_j}$	ID of drone j bound to user i

RSA. A 256-bit elliptic curve public key can achieve the same level of security as 3072-bit RSA public key. The elliptic curve function $E(a, b)$ used in ECC is defined as

$$y^2 = x^3 + ax + bx \tag{1}$$

where a and b are predefined parameters in different cryptography standard.

The private key in ECC is a random integer representing the time of reflections on the function $E(a, b)$. The public key in the ECC is a pair of integer coordinates on a finite elliptic curve. The encryption and decryption of ECC use an operation called EC scalar multiplication. Specifically, we employ the elliptic-curve Diffie–Hellman (ECDH) key agreement protocol. The basic equation of ECDH is

$$(k_\alpha * G) * k_\beta = (k_\beta * G) * k_\alpha = (x_n, y_n) \tag{2}$$

where $*$ represents EC scalar multiplication. k_α and k_β are private keys of user α and β, which are randomly selected integers. G is a generator point on a finite elliptic curve. $(k_\alpha * G)$ is the public key of user α and $(k_\beta * G)$ is the public key of user β. The EC point (x_n, y_n) is the shared secret for both parties, which is used for encryption.

We proposed an Internet-of-Drones authentication protocol based on [21] and made two key changes to meet the constraints of real applications. First, we use the mobile phone ID to replace the biometric ID. Despite that most mobile phones have supported biometric login today, the user's bio information is guarded by the operating system and cannot be accessed directly. Second, we use the UUID of the mobile platform installed on the drone as an additional identifier for user-drone binding and security enhancement.

The symbols in our protocol are shown in Table IV. The authentication scheme can be divided into the following stages.

A. Drone Registration

The AI wings embedded software can register the drone in the cloud server C at first installation. The cloud server generates an unique ID ID_{Dj} for each drone j and creates the pseudoidentity PID_{D_j} of the drone j using a private random integer key r_c and a one-way hash function $h(\cdot)$

$$\text{PID}_{D_j} = h(\text{ID}_{D_j}||\text{UID}_{D_j}||r_c) \tag{3}$$

The embedded software then saves ID_{Dj} and PID_{Dj} at local storage while cloud server C saves ID_{D_j} and UID_{D_j} to the database.

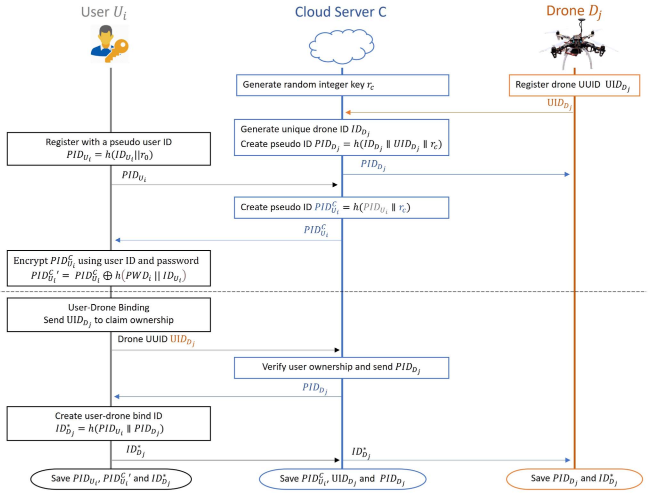

User U_i
Cloud Server C
Drone D_j
Generate random integer key r_c
Register drone UUID UID_{D_j}
UID_{D_j}
Generate unique drone ID ID_{D_j}
Create pseudo ID $PID_{D_j} = h(ID_{D_j} \| UID_{D_j} \| r_c)$
Register with a pseudo user ID
$PID_{U_i} = h(ID_{U_i}||r_0)$
PID_{D_j}
PID_{U_i}
Create pseudo ID $PID^C_{U_i} = h(PID_{U_i} \| r_c)$
$PID^C_{U_i}$
Encrypt $PID^C_{U_i}$ using user ID and password
$PID^C_{U_i}{}' = PID^C_{U_i} \oplus h(PWD_i \| ID_{U_i})$
User-Drone Binding
Send UID_{D_j} to claim ownership
Drone UUID UID_{D_j}
Verify user ownership and send PID_{D_j}
PID_{D_j}
Create user-drone bind ID
$ID^*_{D_j} = h(PID_{U_i} \| PID_{D_j})$
$ID^*_{D_j}$
$ID^*_{D_j}$
Save PID_{U_i}, $PID^C_{U_i}{}'$ and $ID^*_{D_j}$
Save $PID^C_{U_i}$, UID_{D_j} and PID_{D_j}
Save PID_{D_j} and $ID^*_{D_j}$

Fig. 7. Interaction diagram for drone registration, user registration, and user-drone binding.

B. User Registration

A user needs to register an account in the cloud C and generate a random number r_0 to create a pseudo ID $\text{PID}_{U_i} = h(\text{ID}_{U_i}||r_0)$. The PID_{U_i} is sent to the cloud in a secure channel. The cloud server then creates a new user ID $\text{PID}^C_{U_i}$ using a private random integer key r_c and send it back to the user.

$$\text{PID}^C_{U_i} = h(\text{PID}_{U_i}||r_c). \tag{4}$$

After receiving the new ID $\text{PID}^C_{U_i}$, the user chooses a password PWD_i and encrypts it using XOR cipher.

$$\text{PID}^C_{U_i}{}' = \text{PID}^C_{U_i} \oplus h(\text{PWD}_i||\text{ID}_{U_i}). \tag{5}$$

C. User-Drone Binding

The user needs to register the drone UUID UID_{D_j} under his/her account in the cloud to confirm the ownership of the drone D_j. Once confirmed, The cloud sends the pseudo drone ID PID_{D_j} to the user, and the user calculates a new ID using pseudouser ID and pseudodrone ID

$$\text{ID}^*_{D_j} = h(\text{PID}_{U_i}||\text{PID}_{D_j}). \tag{6}$$

The userdrone bind ID is sent to the cloud, and the cloud passes it to the drone. Finally, the user saves PID_{U_j}, $\text{PID}^C_{U_i}{}'$, and $\text{ID}^*_{D_j}$ at local storage; the cloud records $\text{PID}^C_{U_i}$, UID_{D_j}, and PID_{D_j} in the database; the drone saves PID_{D_j} and $\text{ID}^*_{D_j}$. The process of drone registration, user registration, and user-drone binding is shown in Fig. 7.

D. Login and Authentication

1) The user needs to login his/her mobile device using bio information and enter the username and password. Once verified, the embedded software retrieves current timestamp T_u and generates two random numbers $r_1, r_2 \in \mathbb{F}_p$ as private keys. A new elliptic point is created using public key of the cloud

$$N_i = r_1 * P_{\text{pub}} = (N_i^x, N_i^y). \tag{7}$$

We then decrypt the encrypted cloud assigned user ID using user's password

$$\mathrm{PID}_{U_i}^C = \mathrm{PID}_{U_i}^{C\,\prime} \oplus h(\mathrm{PWD}_i||\mathrm{ID}_{U_i}) \tag{8}$$

and add timestamp T_u to compute new encryption A_i as

$$A_i = h(\mathrm{PID}_{U_i}^C||T_u). \tag{9}$$

All the information is encrypted by *x*-coordinate of user's private EC key N_i^x as

$$A_i' = A_i \oplus N_i^x \tag{10}$$

$$\mathrm{PID}_i' = \mathrm{PID}_{U_i} \oplus N_i^x \tag{11}$$

$$\mathrm{ID}_{D_j}' = \mathrm{ID}_{D_j}^* \oplus N_i^x, \tag{12}$$

$$M_i = r_1 * G, r_2' = r_2 \oplus N_i^x \tag{13}$$

where M_i is the user's public EC key. The encrypted login request $\mathrm{REQ}_1 = \{M_i, r_2', T_u, A_i,' \mathrm{PID}_i^c, \mathrm{ID}_{D_j}^c\}$ is sent to the cloud server C via public channel.

2) On receiving the login request, cloud C checks the time freshness $|T_u - T_{\text{now}}| < \Delta T$. If passed, C decrypts M_i by computing

$$M_i * k_{\text{pvt}} = N_i = (N_i^x, N_i^y) \tag{14}$$

and uses N_i^x to decrypt other information

$$A_i = A_i' \oplus N_i^x \tag{15}$$

$$\mathrm{PID}_{U_i} = \mathrm{PID}_i' \oplus N_i^x \tag{16}$$

$$\mathrm{ID}_{D_j}^* = \mathrm{ID}_{D_j}' \oplus N_i^x \tag{17}$$

$$r_2 = r_2' \oplus N_i^x. \tag{18}$$

The cloud then checks if

$$A_i \stackrel{?}{=} h(\mathrm{PID}_{U_i}^C||T_u). \tag{19}$$

If passed, C is ready to send information to the drone D_j. The cloud C gets current timestamp T_s and encrypts the user information as access request

$$X_j = h(\mathrm{PID}_{U_i}||r_2||\mathrm{ID}_{D_j}^*||T_s) \tag{20}$$

$$X_j' = X_j \oplus \mathrm{PID}_{D_j} \tag{21}$$

$$W_j = h(\mathrm{PID}_{D_j}||X_j||T_s) \tag{22}$$

$$W_j' = W_j \oplus \mathrm{PID}_{D_j} \tag{23}$$

$$\mathrm{PID}_{U_i}^N = h(\mathrm{PID}_{U_i}||N_i) \tag{24}$$

$$\mathrm{PID}_{D_j}^C = h(\mathrm{PID}_{D_j}||\mathrm{ID}_{D_j}^*||T_s). \tag{25}$$

The server C sends the encrypted access request $\mathrm{REQ}_2 = \{\mathrm{PID}_{D_j}^C, \mathrm{PID}_{U_i}^N, T_s, X_j,' W_j'\}$ to the drone D_j.

3) The drone D_j checks the time freshness $|T_s - T_{\text{now}}| < \Delta T$. If passed, then D_j verifies that

$$\mathrm{PID}_{D_j}^C \stackrel{?}{=} h(\mathrm{PID}_{D_j}||\mathrm{ID}_{D_j}^*||T_s) \tag{26}$$

by using the PID_{D_j} and $\mathrm{ID}_{D_j}^*$ saved in local storage. If passed, the drone decrypts B_j' and W_j', and verify $W_j \stackrel{?}{=} h(\mathrm{PID}_{D_j}||X_j||T_s)$.

The drone gets current timestamp T_D and creates the session key

$$\mathrm{SK}_{D_j} = h(X_j||\mathrm{ID}_{D_j}^*||W_j||T_D) \tag{27}$$

and computes the message

$$Y_j = W_j \oplus X_j \tag{28}$$

$$Z_j = h(\mathrm{SK}_{D_j}||\mathrm{ID}_{D_j}^*||T_D). \tag{29}$$

The drone sends the response $\mathrm{RES} = \{\mathrm{PID}_{U_i}^N, Y_j, Z_j, T_s, T_D\}$ back to the user U_i.

4) The user U_i receives the response RES and checks the time freshness $|T_D - T_{\text{now}}| < \Delta T$ and $\mathrm{PID}_{U_i}^N \stackrel{?}{=} h(\mathrm{PID}_{U_i}||N_i)$. If both requirements are passed, U_i uses local information to compute X_j

$$X_j = h(\mathrm{PID}_{U_i}||r_2||\mathrm{ID}_{D_j}^*||T_s)) \tag{30}$$

and decrypts $W_j = Y_j \oplus X_i$. Finally, the user U_i calculates the session key SK_{U_i}

$$\mathrm{SK}_{U_i} = h(X_i||\mathrm{ID}_{D_j}^*||W_j||T_D). \tag{31}$$

SK_{U_i} should be equal to SK_{D_j}. The user U_i verifies the session key by checking

$$h(\mathrm{SK}_{U_i}||\mathrm{ID}_{D_j}^*||T_D) \stackrel{?}{=} Z_j. \tag{32}$$

The drone authorization process is illustrated in Fig. 8.

The authentication protocol can resist most known attacks including password guessing, denial-of-service, user impersonation, session key agreement, etc. Table V lists the comparison with other drone authentication algorithms. As shown in the table, our protocol has the most complete security features than other methods. For the detailed crypto analysis, please refer to [21].

TABLE V
COMPARISON OF AUTHENTICATION ATTACK RESISTANCE

Attack Type	[62]	[35]	[63]	Ours
User anonymity		✓		✓
Privileged insider	✓		✓	✓
Password guessing	✓	✓	✓	✓
Denial-of-service	✓		✓	✓
User impersonation	✓		✓	✓
Man-in-the middle	✓	✓	✓	✓
Mutual authentication	✓		✓	✓
Session key agreement	✓		✓	✓
Untraceability			✓	✓
Session key security			✓	✓

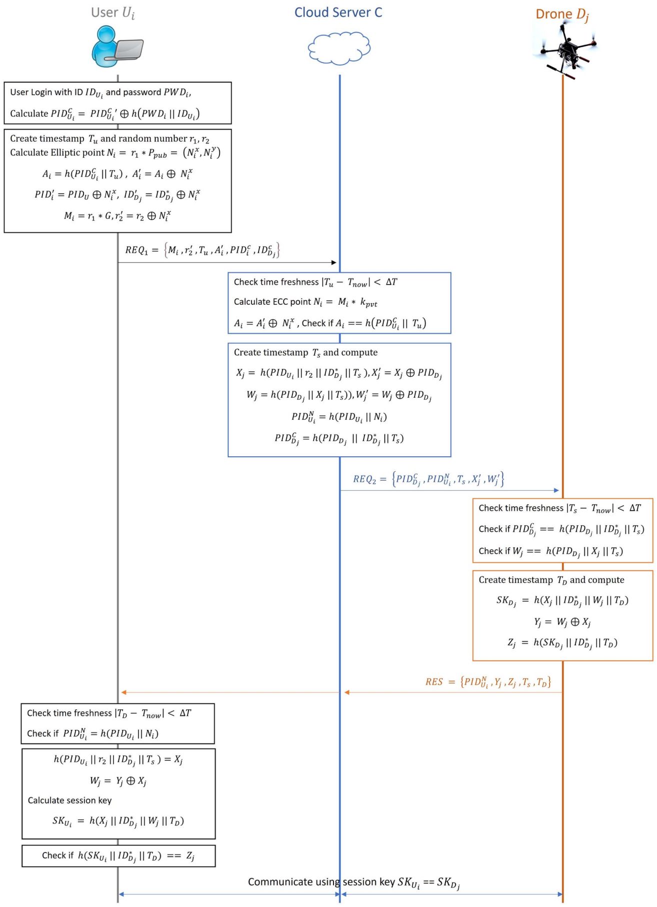
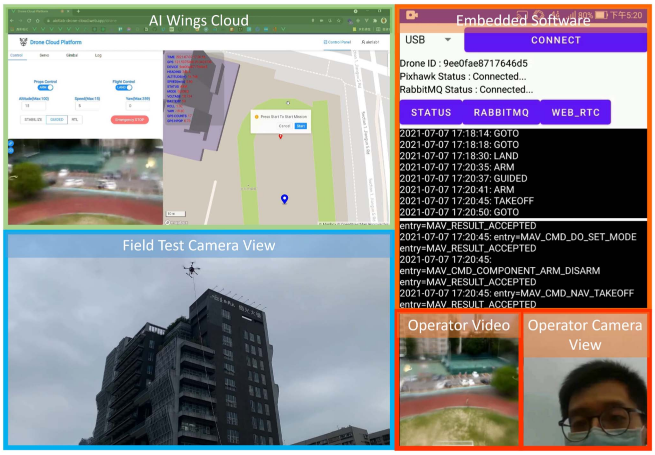

User U_i
Cloud Server C
Drone D_j
User Login with ID ID_{U_i} and password PWD_i,
Calculate $PID_{U_i}^{C} = PID_{U_i}^{C}{}' \oplus h(PWD_i \,||\, ID_{U_i})$
Create timestamp T_u and random number r_1, r_2
Calculate Elliptic point $N_i = r_1 * P_{pub} = (N_i^x, N_i^y)$
$A_i = h(PID_{U_i}^{C} \,||\, T_u)$, $A_i' = A_i \oplus N_i^x$
$PID_i' = PID_U \oplus N_i^x$, $ID_{D_j}' = ID_{D_j}^{*} \oplus N_i^x$
$M_i = r_1 * G, r_2' = r_2 \oplus N_i^x$
$REQ_1 = \{M_i, r_2', T_u, A_i', PID_i^c, ID_{D_j}^c\}$
Check time freshness $|T_u - T_{now}| < \Delta T$
Calculate ECC point $N_i = M_i * k_{pvt}$
$A_i = A_i' \oplus N_i^x$, Check if $A_i == h(PID_{U_i}^{C} \,||\, T_u)$
Create timestamp T_s and compute
$X_j = h(PID_{U_i} \,||\, r_2 \,||\, ID_{D_j}^{*} \,||\, T_s), X_j' = X_j \oplus PID_{D_j}$
$W_j = h(PID_{D_j} \,||\, X_j \,||\, T_s)), W_j' = W_j \oplus PID_{D_j}$
$PID_{U_i}^{N} = h(PID_{U_i} \,||\, N_i)$
$PID_{D_j}^{C} = h(PID_{D_j} \,||\, ID_{D_j}^{*} \,||\, T_s)$
$REQ_2 = \{PID_{D_j}^{C}, PID_{U_i}^{N}, T_s, X_j', W_j'\}$
Check time freshness $|T_s - T_{now}| < \Delta T$
Check if $PID_{D_j}^{C} == h(PID_{D_j} \,||\, ID_{D_j}^{*} \,||\, T_s)$
Check if $W_j == h(PID_{D_j} \,||\, X_j \,||\, T_s)$
Create timestamp T_D and compute
$SK_{D_j} = h(X_j \,||\, ID_{D_j}^{*} \,||\, W_j \,||\, T_D)$
$Y_j = W_j \oplus X_j$
$Z_j = h(SK_{D_j} \,||\, ID_{D_j}^{*} \,||\, T_D)$
$RES = \{PID_{U_i}^{N}, Y_j, Z_j, T_s, T_D\}$
Check time freshness $|T_D - T_{now}| < \Delta T$
Check if $PID_{U_i}^{N} = h(PID_{U_i} \,||\, N_i)$
$h(PID_{U_i} \,||\, r_2 \,||\, ID_{D_j}^{*} \,||\, T_s) = X_j$
$W_j = Y_j \oplus X_j$
Calculate session key
$SK_{U_i} = h(X_j \,||\, ID_{D_j}^{*} \,||\, W_j \,||\, T_D)$
Check if $h(SK_{U_i} \,||\, ID_{D_j}^{*} \,||\, T_D) == Z_j$
Communicate using session key $SK_{U_i} == SK_{D_j}$

Fig. 8. Authorization process for accessing a user's drone in an open channel.



Drone ID : 9ee0fae8717646d5
Pixhawk Status : Connected...
RabbitMQ Status : Connected...

```
2021-07-07 17:18:14: GOTO
2021-07-07 17:18:18: GOTO
2021-07-07 17:18:30: LAND
2021-07-07 17:20:35: ARM
2021-07-07 17:20:37: GUIDED
2021-07-07 17:20:41: ARM
2021-07-07 17:20:45: TAKEOFF
2021-07-07 17:20:50: GOTO
```

```
entry=MAV_RESULT_ACCEPTED
2021-07-07 17:20:45: entry=MAV_CMD_DO_SET_MODE
entry=MAV_RESULT_ACCEPTED
2021-07-07 17:20:45:
entry=MAV_CMD_COMPONENT_ARM_DISARM
entry=MAV_RESULT_ACCEPTED
2021-07-07 17:20:45: entry=MAV_CMD_NAV_TAKEOFF
entry=MAV_RESULT_ACCEPTED
```

Fig. 9. Test video of AIoT X800 drone. There are five screenshots in the picture: cloud UI, embedded software, operator video, operator camera and field test camera view. The video was published on YouTube.

V. Experiments

In order to test the functions of AI Wings, we have built three different types of drones, as shown in Fig. 2. The first one was based on a commercial drone, the Bebop 2 made by Parrot AR. The second and third drones, F450 and X800, were assembled by ourselves. The test flight of AIoT X800 is shown in Fig. 9. The figure consists of five screenshots: cloud UI, embedded software, operator video, operator camera, and field test camera view. We first assigned the destination using the cloud UI, and record the flight process. The map is shown in the AI Wings Cloud UI. The field camera records the view of the drone flying by a building. The embedded software view shows the MAVLink commands received by the drone. The operator video view is the video streamed from the drone, and the operator camera records the activities of the operator. All test videos can be found on YouTube.[1]

In the following subsections, we present the detailed specifications of each AIoT drone.

A. AIoT Bebop 2

Bebop 2 is a camera drone made by Parrot AR, whose dimension is 328 × 328 mm and weight is around 310 g without a battery and 524 g with a battery. The flight time is around 25 min with a 2700-mAh battery. We chose Bebop 2 because its control system is open and can be customized. We attached a Qualcomm mobile platform on top of the drone and developed a special version of embedded software to control Bebop 2 via our cloud server.

Although Bebop 2 is a very nice commercial drone with high freedom of customization, it is at the end of the line now. New users of AI Wings are suggested to assemble their own drones as presented in the following sections.

B. AIoT F450

The F450 frame kit is manufactured by DJI, which includes a PCB board (main body), frame arms, ESC, motors, propellers, and screws. The dimension of the assembled drone is 450 × 450 mm. The cost of the kit is around \$100 USD, while the weight is around 800 g without a battery, gimbal, and camera. A fully equipped AIoT F450 is around 1490 g. The takeoff weight is around 1600 g. Photos of the AIoT F450 drone test flight are shown in Fig. 10. The companion mobile phone can be installed at the bottom of the drone.

C. AIoT X800 and AED Rescue Service

One of the most important drone applications is the emergency transportation of medical supplies. Drones can overcome terrain restrictions, avoid traffic jams, and deliver goods in the shortest time. We have cooperated with the Taiwan Association of Public Access Defibrillation to develop an emergency rescue service. An AED is an electronic medical device that can diagnose and treat cardiac arrhythmia and pulse-less ventricular tachycardia automatically. Cardiac arrhythmias are anomalies in the heartbeat that is too fast or too slow. Ventricular tachycardia (VT) is a fast heart rate that is above 100 beats per minute in adults. As the brain start to suffer irreversible damage after 3–5 min [64], the time to first aid is the key to saving the lives and improving quality of life outcomes for people with sudden arrhythmia. An AED can diagnose the abnormal heart rhythm pulse and apply

[1] www.youtube.com/playlist?list=PL3S3ZnDPwJ-MV5H1yTzR0jPp5sz-ptSo3

TABLE VI
RUNNING TFLITE OBJECT DETECTOR ON VARIOUS ANDROID PLATFORMS

Phone	CPU	Released	Price (2022)	AI Benchmark [65]	Object Detector	Detection FPS
Samsung Galaxy S20	Snapdragon 865	2020	$517	88.5	YOLObile (320×320) [66]	≈ 20
ASUS ROG PHONE II	Snapdragon 855	2019	$456	90.2	MobileNet SSD (300×300)	≈ 33
Moto G9 Play	Snapdragon 665	2020	$110	14.1	MobileNet SSD (300×300)	≈ 16
Samsung Galaxy Note 8	Exynos Octa 8895	2017	$190	21.9	MobileNet SSD (300×300)	≈ 14

The performance of YOLObile was reported by the authors. The AI benchmark scores of the mobiles phones are shown for reference.

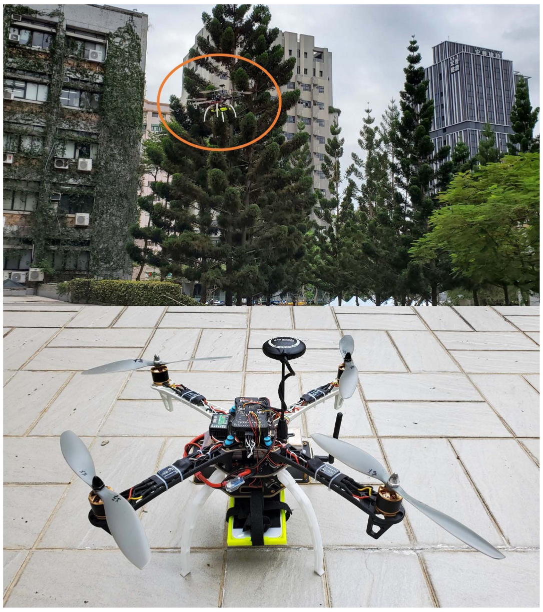
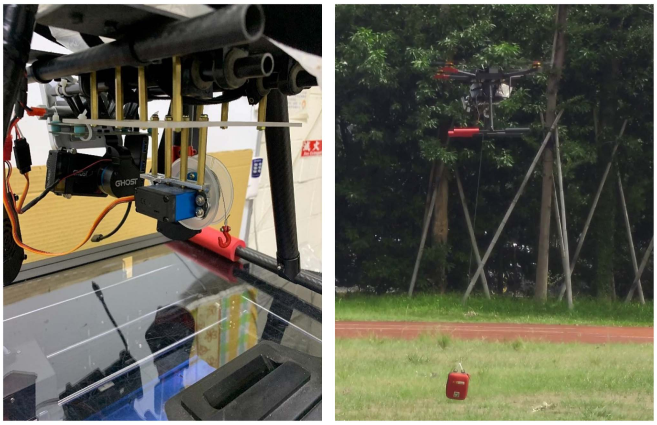


Fig. 10. DIY AIoT F450 drone with the mobile platform installed at the bottom.

Fig. 11. AED drone in the sky.

an electric shock to restore the heart to normal function. The portable AED used in our service is the FRED Easyport made by SCHILLER. The weight of the whole package is around 860 g. The dimension of the drone is 800 × 800 mm, the weight is around 3830 g without a battery, and 5600 g with a battery. We chose a 6S 22.2 V 16500 mAh Li-Po battery. The maximum takeoff weight is 7.8 kg. The flight time is around 18.5 min without additional cargo, 16 min with an AED. Fig. 11 shows the drone carrying an AED in the sky. Because the AED is fragile, it is not suitable for airdrop. We have designed a servo motor to lower the AED to the ground. The cargo delivery device and AED delivery test are shown in Fig. 12.



(a) (b)

Fig. 12. Cargo delivery device installed on the X800. (a) Servo motor to lower package. (b) Delivering AED to the ground.

D. AI Performance Evaluation

We evaluated the performance of state-of-the-art object detectors on our ESM, since object detection is the most fundamental task in computer vision. In general, the common object detectors can be directly applied to drones at low altitudes ($\leq$ 10 m), applied with overlapping grid detection at altitudes between 10 and 50 m, and require fine tuning at higher altitudes ($\geq$ 50 m). We applied the quantized MobileNet SSD [20] in TensorFlow Lite to detect humans on different Android platforms. We also added the test results of YOLObile [66]. In our experiments, 4 out of 8 CPU cores are used for the object detection, so 50% of the computation power is used for other tasks, including drone control, video streaming and cloud communication. The GPU and DSP are mainly responsible for running the deep learning model of the object detector. The experiment results are shown in Table VI. The general consensus of real time is 30 FPS. On the Snapdragon 855, the TFLite object detector can achieve real-time performance (≈ 33 FPS) for 300 × 300 input images. It can still detect around 16 frames per second on low-end Snapdragon 665. We have added the AI benchmark [65] scores of different mobile phones in the table for reference.

VI. CONCLUSION

This article presented AI Wings, an open-source AIoT drone system. As far as we know, this is currently the most complete open-source drone system that covers embedded software/hardware, drone cloud, drone authentication, and VR simulation. Users can convert any ArduPilot-based drone into an AIoT drone with low extra cost and weight. Researchers can simulate and train AI models for computer vision tasks in virtual worlds. The models can then be deployed on real drones directly. Furthermore, we developed an advanced authentication protocol based on ECC with pseudoidentity and time freshness check to ensure the security of the AI Wings drone network.

We have built three different types of AIoT drones and a medical delivery service using AI Wings. The experiments demonstrated the practicability of our system. About future work, we will develop more VR environments and train AI models to recognize more human actions such as waving for help or doing CPR, and deploy the models to our drones to make them more autonomous.

References

[1] E. Cheng, *Aerial Photography and Videography Using Drones*. Berkeley, CA, USA: Peachpit Press, 2015.

[2] SPH Engineering, "Drone show software." Accessed: Dec. 28, 2019. [Online]. Available: https://www.droneshowsoftware.com

[3] A. Singh, D. Patil, and S. N. Omkar, "Eye in the sky: Real-time drone surveillance system (DSS) for violent individuals identification using ScatterNet hybrid deep learning network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 1710–17108.

[4] J. Hao, J. Li, Y. Pi, and X. Fang, "A drone fleet-borne SAR model and three-dimensional imaging algorithm," *IEEE Sensors J.*, vol. 19, no. 20, pp. 9178–9186, Oct. 2019.

[5] B. Mishra, D. Garg, P. Narang, and V. Mishra, "Drone-surveillance for search and rescue in natural disaster," *Comput. Commun.*, vol. 156, pp. 1–10, 2020.

[6] L. Cameron, "Drones as the new "Flying IoT": They'll track people and deliver goods using a new low-power architecture to juice the Apps while staying aloft," 2017. Accessed: Jan. 12, 2022. [Online]. Available: https://www.heart.org/en/news/2019/11/15/drone-delivered-aeds-fly-a-step-closer-to-saving-lives

[7] NVIDIA Corporation, "Jetson modules." Accessed: Jan. 12, 2022. [Online]. Available: https://developer.nvidia.com/embedded/jetson-modules

[8] Qualcomm Technologies, Inc., "Qualcomm flight pro." Accessed: Jan. 12, 2022. [Online]. Available: https://developer.qualcomm.com/hardware/qualcomm-flight-pro

[9] Qualcomm Technologies, Inc., "Qualcomm snapdragon." Accessed: Jan. 12, 2022. [Online]. Available: https://www.qualcomm.com/snapdragon

[10] ModalAI, "ModalAI." Accessed: Apr. 10, 2022. [Online]. Available: https://www.modalai.com/

[11] XB R&D Co., LTD., "XBStation." Accessed: Apr. 10, 2022. [Online]. Available: https://xbstation.com/

[12] Dronecloud Ltd., "Dronecloud." Accessed: Jan. 12, 2022. [Online]. Available: https://dronecloud.io/

[13] Sky-Drones Technologies Ltd., "Sky-drones." Accessed: Jan. 12, 2022. [Online]. Available: https://sky-drones.com

[14] Altitude Angel Ltd., "Altitude angel." Accessed: Jan. 12, 2022. [Online]. Available: https://www.altitudeangel.com

[15] 3D Robottics, "3DR." Accessed: Jan. 12, 2022. [Online]. Available: https://3dr.com/

[16] Auterion Ltd., "Auterion." Accessed: Jan. 12, 2022. [Online]. Available: https://auterion.com/

[17] Dronecode, "MAVlink: Micro air vehicle communication protocol," Accessed: Jul. 13, 2022. [Online]. Available: https://mavlink.io

[18] Mapbox, "Mapbox." Accessed: Jan. 12, 2022. [Online]. Available: https://www.mapbox.com

[19] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*. Berlin, Germany: Springer, 2018, pp. 621–635.

[20] J. Huang et al., "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3296–3297.

[21] S. Hussain, S. A. Chaudhry, O. A. Alomari, M. H. Alsharif, M. K. Khan, and N. Kumar, "Amassing the security: An ECC-based authentication scheme for internet of drones," *IEEE Syst. J.*, vol. 15, no. 3, pp. 4431–4438, Sep. 2021.

[22] Wikipedia, "Delivery drone," Dec. 2019. Accessed: Jan. 12, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Delivery_drone

[23] D. Wang, P. Hu, J. Du, P. Zhou, T. Deng, and M. Hu, "Routing and scheduling for hybrid truck-drone collaborative parcel delivery with independent and truck-carried drones," *IEEE Internet of Things J.*, vol. 6, no. 6, pp. 10483–10495, Dec. 2019.

[24] J. Wang et al., "Edge-based live video analytics for drones," *IEEE Internet Comput.*, vol. 23, no. 4, pp. 27–34, Jul./Aug. 2019.

[25] D. Hernandez, J.-C. Cano, F. Silla, C. T. Calafate, and J. M. Cecilia, "AI-enabled autonomous drones for fast climate change crisis assessment," *IEEE Internet of Things J.*, vol. 9, no. 10, pp. 7286–7297, May 2022.

[26] P. Zhu, L. Wen, D. Du, X. Bian, Q. Hu, and H. Ling, "Vision meets drones: Past, present and future," 2020, *arXiv:2001.06303*.

[27] T. Li, J. Liu, W. Zhang, Y. Ni, W. Wang, and Z. Li, "UAV-human: A large benchmark for human behavior understanding with unmanned aerial vehicles," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 16261–16270.

[28] M. Campion, P. Ranganathan, and S. Faruque, "UAV swarm communication and control architectures: A review," *J. Unmanned Veh. Syst.*, vol. 7, no. 2, pp. 93–106, 2018.

[29] X. Pan, Y. You, Z. Wang, and C. Lu, "Virtual to real reinforcement learning for autonomous driving," in *Proc. Brit. Mach. Vis. Conf.*, London, UK, Sep. 2017. [Online]. Available: https://www.dropbox.com/s/qqb0qvt1q18dp9e/0221.pdf?dl=1

[30] F. Sadeghi and S. Levine, "CAD2RL: Real single-image flight without a single real image," *Robot.: Sci. Syst.*, 2016.

[31] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 23–30.

[32] K.-T. Lai, C.-C. Lin, C.-Y. Kang, M.-E. Liao, and M.-S. Chen, "VIVID: Virtual environment for visual deep learning," in *Proc. ACM Multimedia Conf.*, 2018, pp. 1356–1359.

[33] E. Cetin, C. Barrado, G. Munoz, M. Macias, and E. Pastor, "Drone navigation and avoidance of obstacles through deep reinforcement learning," in *Proc. IEEE/AIAA 38th Digit. Avionics Syst. Conf.*, 2019, pp. 1–7.

[34] R. Madaan et al., "AirSim drone racing lab," in *Proc. NeurIPS Competition Demonstration Track*, 2020, pp. 177–191.

[35] M. Wazid, A. K. Das, N. Kumar, A. V. Vasilakos, and J. J. Rodrigues, "Design and analysis of secure lightweight remote user authentication and key agreement scheme in internet of drones deployment," *IEEE Internet of Things J.*, vol. 6, no. 2, pp. 3572–3584, Apr. 2019.

[36] M. Gharibi, R. Boutaba, and S. L. Waslander, "Internet of drones," *IEEE Access*, vol. 4, pp. 1148–1162, 2016.

[37] L. Abualigah, A. Diabat, P. Sumari, and A. H. Gandomi, "Applications, deployments, and integration of Internet of Drones (IoD): A review," *IEEE Sensors J.*, vol. 21, no. 22, pp. 25532–25546, Nov. 2021.

[38] R. Bassoli and F. Granelli, "Rapid deployment of 5G services using drones and other manned and unmanned aerial vehicles," Accessed: Jul. 13, 2022. [Online]. Available: https://www.5gitaly.eu/2018/wp-content/uploads/2019/02/5G-Italy-White-eBook-Drones.pdf

[39] A. Meng, X. Gao, Y. Zhao, and Z. Yang, "Three-dimensional trajectory optimization for energy-constrained UAV-enabled IoT system in probabilistic LoS channel," *IEEE Internet of Things J.*, vol. 9, no. 2, pp. 1109–1121, Jan. 2022.

[40] X. Gao, D. Niyato, K. Yang, and J. An, "Cooperative scheme for backscatter-aided passive relay communications in wireless-powered D2D networks," *IEEE Internet of Things J.*, vol. 9, no. 1, pp. 152–164, Jan. 2022.

[41] S. A. Chaudhry, M. S. Farash, N. Kumar, and M. H. Alsharif, "PFLUA-DIoT: A pairing free lightweight and unlinkable user access control scheme for distributed IoT environments," *IEEE Syst. J.*, vol. 16, no. 1, pp. 309–316, Mar. 2022.

[42] X. Jia, D. He, N. Kumar, and K.-K. R. Choo, "A provably secure and efficient identity-based anonymous authentication scheme for mobile edge computing," *IEEE Syst. J.*, vol. 14, no. 1, pp. 560–571, Mar. 2020.

[43] N.-N. Dao et al., "Securing heterogeneous IoT with intelligent DDoS attack behavior learning," *IEEE Syst. J.*, vol. 16, no. 2, pp. 1974–1983, Jun. 2022.

[44] C. Ge, L. Zhou, G. P. Hancke, and C. Su, "A provenance-aware distributed trust model for resilient unmanned aerial vehicle networks," *IEEE Internet of Things J.*, vol. 8, no. 16, pp. 12481–12489, Aug. 2021.

[45] XB R&D Co., Ltd., "XBLink." Accessed: Mar. 10, 2022. [Online]. Available: https://xbstation.com/store/xblink

[46] A. A. Süzen, B. Duman, and B. Şen, "Benchmark analysis of Jetson TX2, Jetson nano and raspberry PI using deep-CNN," in *Proc. Int. Congr. Human-Comput. Interact. Optim. Robot. Appl.*, 2020, pp. 1–5.

[47] SZ DJI Technology Co., Ltd., "Matrice 300 RTK." Accessed: Jan. 12, 2022. [Online]. Available: https://www.dji.com/en/matrice-300

[48] Skydio Inc., “3DR.” Accessed: Jan. 12, 2022. [Online]. Available: https://www.skydio.com/skydio-2

[49] ModalAI, “ModalAI m500.” Accessed: Apr. 10, 2022. [Online]. Available: https://www.modalai.com/products/voxl-m500

[50] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, “PIXHAWK: A system for autonomous flight using onboard computer vision,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 2992–2997.

[51] B. Sredojev, D. Samardzija, and D. Posarac, “WebRTC technology overview and signaling solution design and implementation,” in *Proc. 38th Int. Conv. Inf. Commun. Technol. Electron. Microelectron.*, 2015, pp. 1006–1009.

[52] J. P. D. A. Barbosa et al., “ROS, android and cloud robotics: How to make a powerful low cost robot,” in *Proc. Int. Conf. Adv. Robot.*, 2015, pp. 158–163.

[53] DroneKit Android, “3DR.” Accessed: Apr. 10, 2022. [Online]. Available: https://github.com/dronekit/dronekit-android

[54] Node.js, “OpenJS foundation.” Accessed: Apr. 10, 2022. [Online]. Available: https://nodejs.org

[55] ArduPilot Dev Team, “Mission planner.” Accessed: Jan. 12, 2022. [Online]. Available: https://ardupilot.org/planner

[56] V. M. Ionescu, “The analysis of the performance of RabbitMQ and ActiveMQ,” in *Proc. 14th RoEduNet Int. Conf.- Netw. Educ. Res.*, 2015, pp. 132–137.

[57] W. Sultani and M. Shah, “Human action recognition in drone videos using a few aerial training examples,” *Comput. Vis. Image Understanding*, vol. 206, 2021, Art. no. 103186.

[58] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “CARLA: An open urban driving simulator,” in *Proc. Conf. Robot Learn.*, 2017, pp. 1–16.

[59] O. M. Andrychowicz et al., “Learning dexterous in-hand manipulation,” *Int. J. Robot. Res.*, vol. 39, no. 1, pp. 3–20, 2020.

[60] M. Amara and A. Siad, “Elliptic curve cryptography and its applications,” in *Proc. Int. Workshop Syst. Signal Process. Appl.*, 2011, pp. 247–250.

[61] S. Halevi and H. Krawczyk, “Strengthening digital signatures via randomized hashing,” in *Proc. Annu. Int. Cryptol. Conf.*, 2006, pp. 41–59.

[62] J. Srinivas, A. K. Das, N. Kumar, and J. J. P. C. Rodrigues, “TCALAS: Temporal credential-based anonymous lightweight authentication scheme for Internet of Drones environment,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6903–6916, Jul. 2019.

[63] B. Bera, D. Chattaraj, and A. K. Das, “Designing secure blockchain-based access control scheme in IoT-enabled Internet of Drones deployment,” *Comput. Commun.*, vol. 153, pp. 229–249, 2020.

[64] American CPR Training, “CPR Facts,” Accessed: Jul. 13, 2022. [Online]. Available: https://americancpr.com/cprfacts.html

[65] A. Ignatov et al., “AI benchmark: Running deep neural networks on android smartphones,” in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2018, pp. 288–314.

[66] Y. Cai et al., “YOLObile: Real-time object detection on mobile devices via compression-compilation co-design,” in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 955–963.

Kuan-Ting Lai (Member, IEEE) received the B.Eng. degree in electrical engineering, the M.Sc. degree in computer science, and the Ph.D. degree in computer science from National Taiwan University, Taipei, Taiwan, in 2003, 2005, and 2015, respectively.

He was a Visiting Scholar with the DVMM lab, Columbia University, from 2012 to 2013, and collaborated with IBM Thomas J. Watson Research Center to develop a large-scale video event detection system. He was a Postdoctoral Researcher with Academia Sinica, Taiwan, and led a group to win the ACM Multimedia Best Open-Source Software Award in 2018. He is currently an Assistant Professor with the Department of Electronic Engineering, National Taipei University of Technology, Taipei, Taiwan. His research interests include computer vision, deep learning, unmanned aerial vehicles, and the Internet of Things.

Yueh-Tsung Chung received the bachelor’s degree in electronic engineering from the National Taipei University of Technology, Taipei, Taiwan, in 2020. He is currently working toward the master’s degree with the Department of Electronic Engineering, National Taipei University, Taipei.

He was a Software Engineer with Tong Hsing Electronic Ind., Ltd., from 2016 to 2020. His current research interests include unmanned aerial vehicles, UI/UX design, and the Internet of Things.

Jun-Jia Su received the bachelor’s degree in electronic engineering and the master’s degree in electronic engineering from the National Taipei University of Technology, Taipei, Taiwan, in 2019 and 2021, respectively.

He is currently an Engineer with AU Optronics Corp. His research interests include unmanned aerial vehicles and the Internet of Things.

Chien-Hung Lai (Member, IEEE) received the Ph.D. degree in electrical engineering from the National Taipei University of Technology, Taipei, Taiwan, in July 2020.

Since August 2020, he has been an Assistant Research Professor with the Department of Electronic Engineering, National Taipei University of Technology. His current research interests include Internet of Things (IoT) applications, artificial intelligence, and software/hardware codesign.

Dr. Lai has proposed an IoT water quality monitoring system, which won the Taiwan Fisackathon Championship in 2018 and Taiwan Presidential Hackathon Best Proposal Team Award in 2019.

Yu-Hsuan Huang received the bachelor’s degree in business administration from Soochow University, Taipei, Taiwan, in 2017, with specialization in business data analysis. She is currently working toward the EMBA master’s degree in AI and big data with the National Taipei University of Technology, Taipei, Taiwan and the University of Texas at Arlington, Arlington, TX, USA .

She is currently the Section Chief with procurement department, YUANYU GROUP Co., Ltd., Taipei, Taiwan. Her research interests include data mining and drone applications.