

# Smart Drone Controller Framework—Toward an Internet of Drones



Chandra Sekar Veerappan, Peter Kok Keong Loh, and Reny Joy Chennattu

**Abstract** There has been an increasing trend to use multiple drones to cooperate autonomously beyond visual line-of-sight missions such as remote services, digital governance and planning, control of safety and security in a smart nation/smart city. In addition, machine learning (ML) has emerged as a key enabler to achieve efficiency in missions such as object detection and intruder detection. In this context, most of the commercially off-the-shelf Wi-Fi drones have limited resources and do not offer any firmware customization; these inherent limitations and technical gaps highlight the need for a software-based smart controller framework to realize support for a team of autonomous drones working together as an Internet of Drones (IoD). This can form the basis for strategic management of new Smart Cities that aim to optimize resources utilization and autonomize services. In this chapter, we present a preliminary architectural design to support needed capabilities and features of a cross-platform Smart Drone Controller (SDC) framework. An SDC framework supports a deployed team of Wi-Fi-based drones to conduct assigned missions collaboratively. The SDC's ML engine has an option to choose algorithms according to the assigned mission. Overall, our SDC framework prototype improves the reliability of the team-based mission and enables a mixed selection of commercial drones to be deployed remotely and collaboratively as an IoD to create positive impact in service autonomy offered to smart city residents. This chapter details framework's implementation and results with multiple Tello Edu drones assigned to an intruder drone detection mission.

**Keywords** UAV · Drones · Controller framework · Smart city · Machine learning · Object detection · Computer vision · Surveillance

---

C. S. Veerappan (✉) · P. K. K. Loh · R. J. Chennattu  
Singapore Institute of Technology, Singapore, Singapore  
e-mail: [chandra.veerappan@singaporetech.edu.sg](mailto:chandra.veerappan@singaporetech.edu.sg)

## Abbreviations

UAV	Unmanned Aerial Vehicle
UAS	Unmanned Aircraft Systems
SDC	Smart Drone Controller
C3I	Command Control Communication and Intelligence
MME	Mission Management Engine
MLE	Machine Learning Engine
FCE	Flight Control Engine
IoD	Internet of Drones
YOLO	You Only Look Once
CNN	Convolution Neural Networks
SSD	Single Shot Detection

## 1 Introduction

Drones/UAV's are increasingly becoming part of a smart city's infrastructure [1]. Smart city core applications include traffic monitoring, crowd monitoring, critical infrastructure inspection, management of limited resources such as air/water quality, tourism-related activities, geodetic works for remote sensing, delivery of goods, solution to natural disasters, and healthcare applications[2]. With the advent of the Internet of Things (IoT), drone networking has been given a new terminology called "Internet of Drones" (IoD). IoD works as a layered network control architecture designed mainly for coordinating the access of Drones in controlled airspace and providing navigation services between locations referred to as nodes [3]. Since drone communication lacks standards, they face several challenges in interoperable implementation, design, and deployment in smart city applications and other domains. For example, multiple Wi-Fi Drones working on large-scale missions or swarm operations face the following issues.

**Cross Platform:** Drone vendor supplied Software Development Kit (SDK)/Application Programming Interface (API) may be limited to mobile operating system (OS) platforms only. Usually, an SDK may be used to execute custom code in the drone during the flight. When missions involve heterogenous drones then controllers with more interoperable platforms, e.g., Windows/Linux OS may be needed for special-purpose applications and services.

**Data Harnessing:** With multiple missions, the drone's flight control data and sensor data need to be stored in the backend (example: video surveillance footage). This historical data helps enhance future drone incidents; mission statistics and outcomes can help to improve the efficiency and effectiveness of operational protocols and procedures. Further, to get actionable insights from this data, support from ML-based analysis with implementation/customization of appropriate ML algorithms

and computer vision-based solutions are needed. In some remote missions, it would be operationally effective if drone behavioral autonomy can be extended in real time, e.g., able to change the altitude of the drone or the camera angle, when certain infrastructural or object points or areas require closer observation.

**Extending Range:** Wi-Fi drones supported by the 802.11 family of protocols offer less than 250 m range in open space [4]. Operating environment interference, device used, and other parameters can affect the operating range which leads to degradation of operational effectiveness of the team-based drone mission. These issues clearly point to the need for a Smart Drone Controller (SDC) framework, which supports the Internet of Drone (IoD) infrastructure to realize better operational decision-making and consensus achievement for the mission's drone team. For example, the SDC can be implemented on mobile endpoint nodes inside a mesh network to increase the drone team's operating range. In this context, we propose a cross-platform (Linux, Windows, Android OS) SDC framework. It communicates directly with the Drone's access point via Wi-Fi. The SDC comprises four core sub-systems as follows

- (1) Command, Control, Communications, and Intelligence (**C3I**)
- (2) Mission Management Engine (**MME**)
- (3) Machine Learning Engine (**MLE**)
- (4) Flight Control Engine (**FCE**).

The rest of the chapter is organized as follows. Related work and the SDC framework's technical overview are reviewed in Sect. 2. The proposed SDC framework's design and functionality are then discussed in Sect. 3. Section 4 examines the application of the SDC framework in an Intruder drone detection mission. The results in Sect. 5 shows the outcome of the deployed mission—a real-time object detection task in the Intruder drone detection mission. Object detection is implemented using SDC's ML Engine with YOLO4/Darknet library, and performance evaluation is done empirically. In Sect. 6, we present our conclusions and future work.

## 2 Related Work

Ground drone controller or smart drone controller-related works are not readily available in the literature. However, we have reviewed some of the important articles related to IoT for drones or swarm operation's domain. In Ref. [5], the authors describe how the multiple UAV or drone services with IoT platform operate with oneM2M global standard. An automated UAV-based surveillance system using multiple drones (**Patrol UAV** and **Tracking UAV**) was also implemented. However, their surveillance mission depends on the video being transmitted through the camera mounted on the UAV, for a user to act and no implementation of machine learning approaches is involved for the mission. In Ref. [6], authors have attempted to provide vision-based drone swarms by Imitation Learning (another approach of reinforcement learning). It is more toward a fully decentralized, vision-based swarm with

no communication. In Ref. [4], the authors analyze UAV IoT framework views and challenges to efficient deployment of drones as “Things” with architecture, security, and privacy points of view. Furthermore, the literature covers how to extend the IoT security layers to UAV including vision-based security enhancements that support self-protection and path/destination identification, safe landing locations by using video-processing and computer-vision (CV) techniques. However, no implementation details of these techniques were discussed. In reference [7], the authors explained well about machine-learning-based real-time object detection on an Android OS mobile platform with some benchmarking results. We note the details mentioned and consider these points when implementing SDC on the Android OS platform. A lot of related works found in the literature [8–11].

As an alternative to drone firmware customization, commercial drones like Tello Edu can be programmed via predefined code blocks to do basic flight navigation. Currently, there is no support for autonomous teamwork. Similar limitations are inherent in the AR Parrot Drone system. A comparison analysis overview of related commercial offerings follows:

- **Services** companies offer drones equipped with computer vision and AI. AI-processed image data from such drones yields actionable insights, however, no autonomous teamwork is supported. Examples include Drone Sense, Neurala, Scale, Skycatch, Alive, Skydio, etc. [12].
- **Domo** uses IP mesh technologies [13] that provide decision-support with predictive analytics to business users. Drones need to be coordinated and controlled via human operators via the mesh-based network.
- **Skylark Drones** uses cloud-based intelligence with system-integrated drones to enhance a customer’s insights in sectors like utilities and infrastructure [14]. Machine learning is used only to process drone data. User-managed services that the company offer includes analysis, tracking, and detection.
- **Optelos** is a leading provider of secure drone data management and AI analytics software. However, the software executes on backend servers that depend on a cloud-based platform for connectivity. An app is available to streamline delivery of actionable drone data [15].
- **DroneDeploy** is a leading cloud software platform for commercial drones. Software automates individual drone’s flight and facilitates the capture of aerial data with a mobile app. The DroneDeploy software platform processes the imagery using computer vision. There are solutions for user-controlled drone-based mapping, modeling, and data analytics capabilities [16].
- **Garuda Robotics** uses a BVLOS platform that will be controlled by a Drone Operations Centre [17]. There is currently no reported work on autonomous teamwork in drones.

Often, the remote physical environments that drones operate in, may prohibit the use of enterprise cloud-based platforms. Further constraints include the data transfer overheads and the limited computing resources on each drone. Beyond these, the above offerings apply machine learning to drone-collected data and not operational mission-focused data, e.g., mission progress, drone actions/status, or event history

etc. They also still require human intervention and control via internet/cloud connectivity with no support for smart autonomous teamwork among multiple drones. Unlike any other approach in recent literature, our main contribution in this chapter is to design and develop smart drone controller (SDC) framework with implementation of mission-based deep learning algorithm like YOLO/Darknet for real-time object detection while SDC receiving video streaming from the drone/UAV; further in this process, we evaluate the deep learning algorithms and benchmark it.

### 3 Smart Drone Controller Design and Functions

The Smart Drone Controller (SDC) is designed in Qt C++, a cross-platform application development technology for desktop, embedded and mobile applications on Linux platform. The SDC framework prototype is installed on laptop computers running Ubuntu 20.04 LTS platforms. Each laptop computer has an external Wi-Fi access card to enable wireless connection to a mission drone. The laptop's internal Wi-Fi card connects, in turn, to a wireless local area network (LAN) supported by router. This forms the mesh network and facilitates other SDCs to connect within network range. For user identification purpose, we label the first connected laptop computer as SDC 1. Subsequent laptop computers joining the LAN network are identified as SDC 2, SDC 3, etc., though each SDC's IP address act as unique identifier referenced as *SDCi@192.160.0.102, where  $i = 1, 2, 3...$*

Tello Edu drones are used drone team (Patrol and Tracking), which can capture 5MP photos, streaming 720p HD video, and around 13 min flight time per fully charged battery. Its SDK connects to the aircraft through a Wi-Fi UDP port, allowing programmatic control of the drone with text-based commands. A GUI (Graphical User Interface) dashboard provides administrative control during development, testing and mission specification, and selection phases. The completed SDC framework is envisaged to be operationally deployed as depicted in Fig. 1.

The SDC framework deployed in a basic network is shown with SDC 1 and SDC 2 running different operating systems in Fig. 2. One of the SDC design goals is to support cross-platform deployment. It has four core sub-systems with dedicated functionalities as shown in Fig. 3.

#### i. **Command, Control, Communications, and Intelligence (C3I) Sub-system**

The C3I sub-system is used to support the sharing of mission-focused data and actionable insights, assist the correlation and generation of decisions as well as the achievement of team consensus. Mission-focused data is based on specified surveillance and inspection scenarios. Examples include mission progress, drone actions/status or event history, etc. The (C3I) sub-system is designed to run cross-platform and supports multi-media data transmissions. It is customized from an existing C3I sub-system used in an earlier IoT Real-time Security Framework [18].

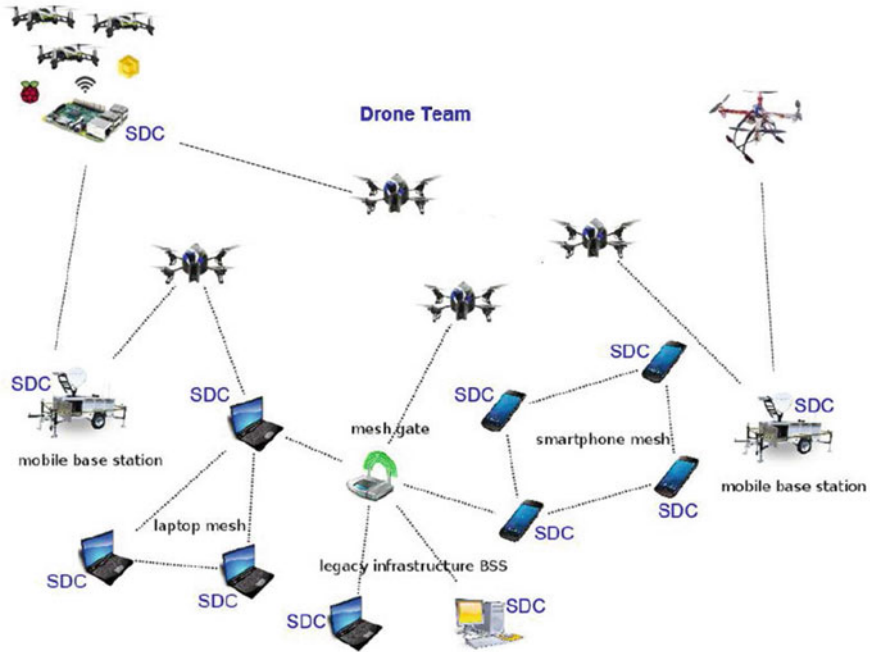


Fig. 1 SDC framework architecture

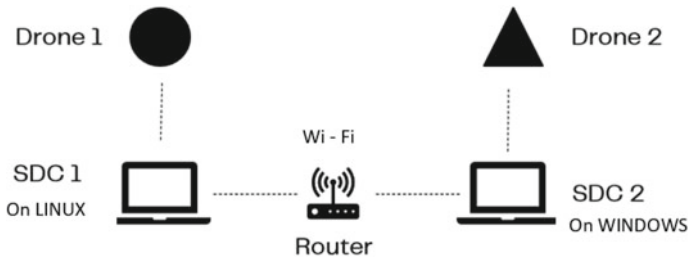


Fig. 2 SDC framework basic network

ii. **Mission Management Engine (MME)**

The MME references and correlates mission-focused data, actionable insights shared by the C3I sub-system to decide on the next course of action. Its main objective is to coordinate mission flight progress and drone team actions. A prioritized set of decisions will be derived and validated by each drone controller to achieve team consensus. The Plug-and-Play Mission Interface facilitates the integration of a range of mission-specific surveillance and inspection apps with no change of the core SDC framework code.

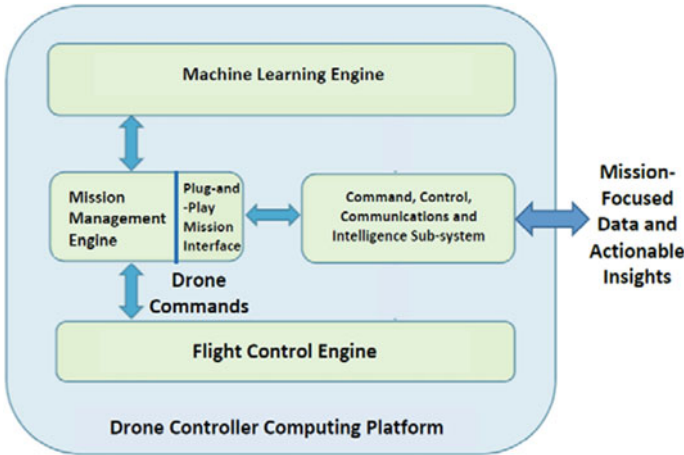


Fig. 3 SDC core sub-systems

### iii. Machine Learning Engine (MLE)

Mission-specific directives are disseminated to the drone team via the C3I sub-system. A mission-focused machine learning algorithm considers the data (mission flight configuration, team actions, events history, and mission progress) to generate actionable insights. These insights will be shared with the drone team before consensus is derived. This differs from commercial offerings that apply machine learning only to collected drone data.

### iv. Flight Control Engine (FCE)

The Flight Control Engine's primary purpose is to control drone flight behavior. It will support the integration of a set of customized flight navigation parameters and code libraries for the specified deployed drones. In addition, we design the SDC framework prototype with a Graphical User Interface (GUI) dashboard application that supports developer administration during the implementation, integration, and testing phases as well as mission administration.

## 3.1 GUI Support for C3I

The core functionalities of C3I may be programmed pre-mission in the SDC dashboard's home page where mission selection, network connectivity of SDC, and drone's network and battery status updates are displayed. Figure 4 shows the screenshot of C3I page. It has the following functions:

- Get the IP address (self) of SDC
- Show the available SDC in the network
- Get the IP address of drone and network status

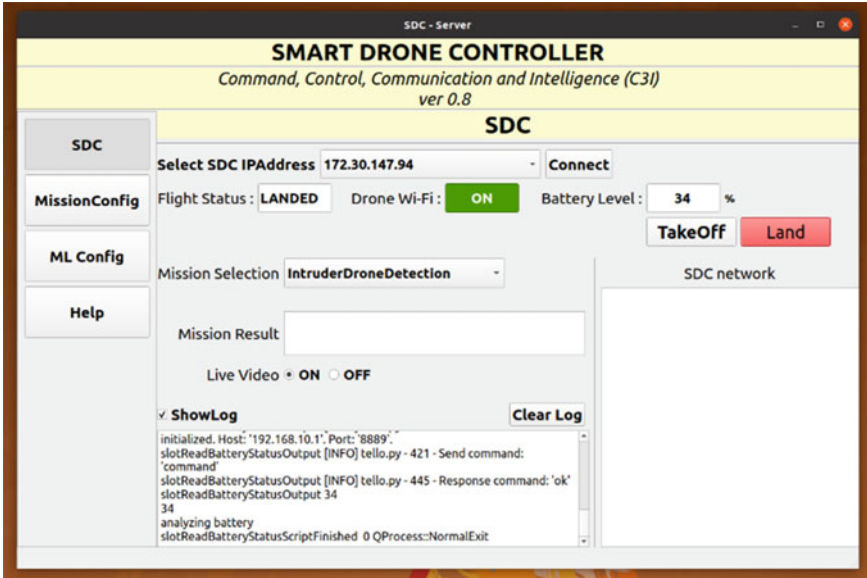


Fig. 4 Dashboard (C3I) home page

- List the available missions for user selection
- Send the Take-off and other mission commands
- Showing the streaming video during flight
- All SDC messages are shown as log and stored in SDCLog folder.

The selected mission routine starts when the user clicks the “**TakeOff**” button. The C3I sub-system verifies that the required mission and machine learning parameters are configured correctly. If there are errors, error messages will be shown in the log. Overall, the C3I plays a critical role to support successful mission completion.

### 3.2 GUI Support for MME

The drone mission flight details such as Takeoff altitude <UP>, distance to move <FORWARD/BACKWARD>, movement direction <RIGHT/LEFT> options are available to user on the MME page. These mission configuration details are saved in the **SDC.ini** file for each mission and passed to the C3I sub-system. The distance to move is referred as the distance between “Point A to Point B” where point A is the starting point of the mission and Point B is the destination. Figure 5 shows the MME’s configuration page.



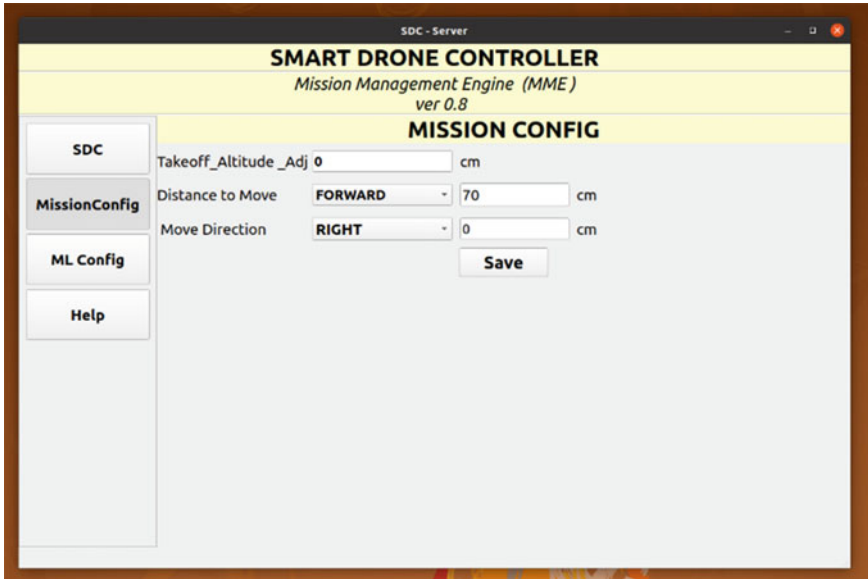


Fig. 5 MME configuration page

### 3.3 GUI Support for MLE

When a new mission is created, the MME’s configuration is uploaded by user as shown in Fig. 6. Following this, the selected mission’s ML configuration will be loaded automatically by MLE from the SDC.ini file. Clicking the **Add** button, a new mission is created, the related machine learning weights, configuration, data file, and names file are uploaded by the user. These files are saved in the **SDC.ini** file and mission folder. The selected mission and files are deleted from mission folder and SDC.ini file by the **Delete** button. The **Edit** option serves to update the ML configuration file.

The values are passed to C3I module where the selected mission’s machine learning model, configuration files, and other related modules are loaded. Hence, the SDC mission routine starts when the user clicks the “Take-off” button on the C3I Home page. We have implemented the real-time object detection using YOLOv4/Darknet [19]—Neural Networks library for Object Detection. The following section describes the Intruder drone detection mission used to verify the functionality of the SDC framework prototype.

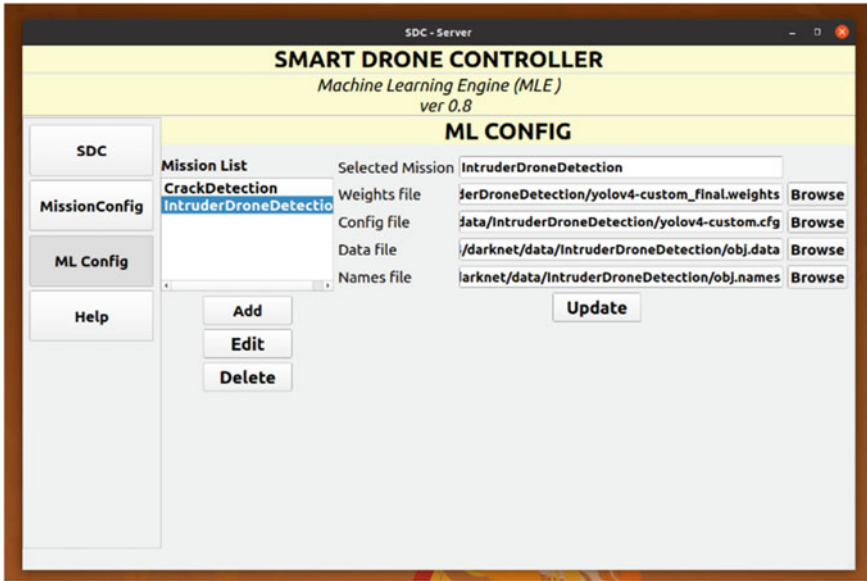


Fig. 6 MLE configuration page

## 4 Mission Deployment

Deployment of the drone team (**Patrol and Tracking drone**) in the Intruder drone detection mission involves the following steps

- (1) At point A, SDC 1's—Patrol drone 1 checks the mission configuration, takes off and flies to waypoint B.
- (2) At waypoint B, the Patrol drone searches for the Intruder drone and detects the target. The drone then informs SDC 1 and alerts the user.
- (3) SDC 1 then informs the other SDCs in the local network, i.e., SDC 2, SDC 3 via the C3I sub-system.
- (4) The corresponding Tracking drones 2 and 3 are remotely tasked by SDC 2 and SDC 3, independently, to fly to waypoint B to confirm the detection of the intruder drone. Upon confirmation, drones 2 and 3 informs SDC 2 and SDC 3, respectively.
- (5) With receipt of intruder alert messages from all 3 SDCs via the C3I sub-system, SDC 1 correlates all received alerts and reaches consensus that there is an intruder detected at waypoint B. The mission flow is described in Fig. 7.

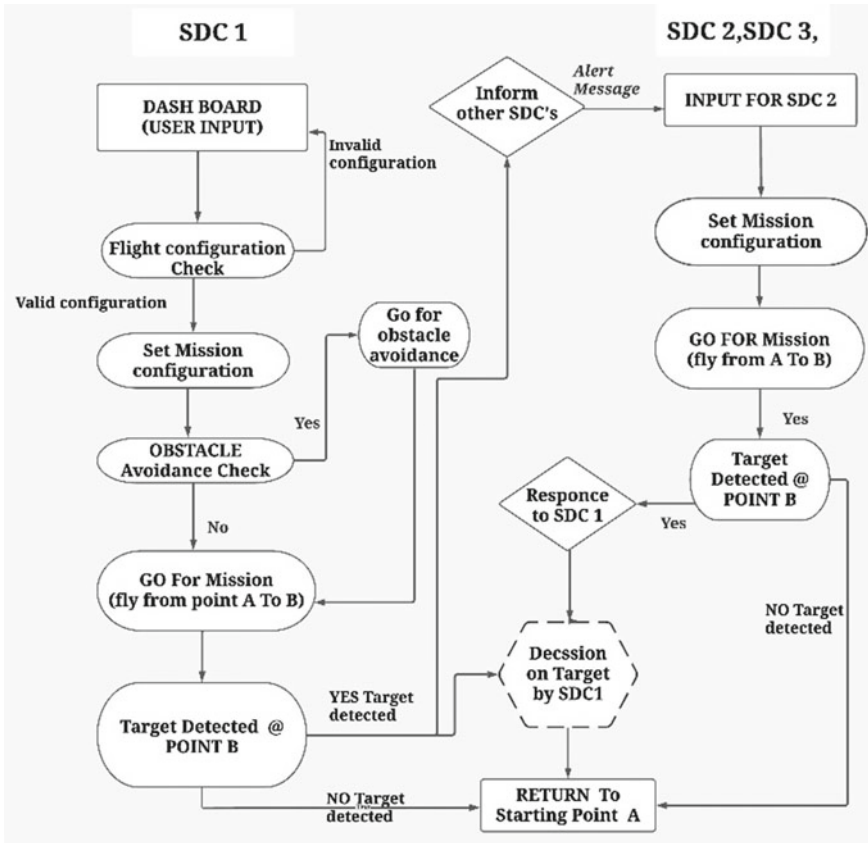
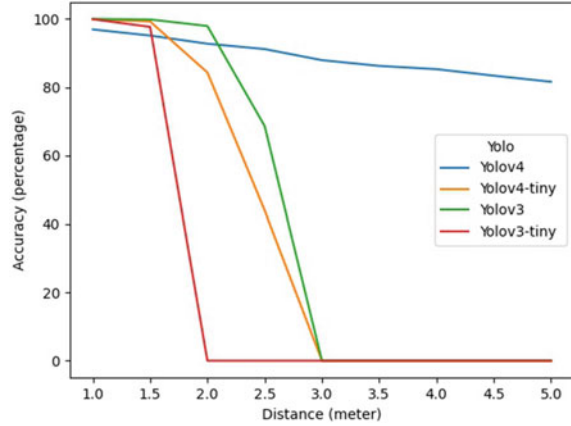


Fig. 7 Intruder drone detection routine

## 5 Intruder Drone Detection Mission Results

Intruder drone detection is an object detection problem. Recent developments in convolution neural networks (CNN) have increased the deep learning methods in the image detection algorithms. Algorithms such as Single Shot Detection (SSD), Faster Region-based Convolutional Neural Networks (Faster R-CNN), and You Only Look Once (YOLO). Among this YOLO version 3 (YOLOv3) shows the best overall performance [20]. Last year, the YOLOv4 was released by Alexey Bochkovskiy, and there were a large number of features that are said to improve Convolutional Neural Network (CNN) accuracy [21]. In this preliminary research work, we have experimented with and used YOLOv4/YOLOv4 Tiny as deep learning library with Darkent as a backbone in Machine Learning Engine (MLE). Our training dataset parameters are:

**Fig. 8** YOLO version comparison



- 200 DJI MINI drone images to train as an intruder drone
- Transfer learning used with pre-trained YOLOv4 weights
- Image size:  $416 \times 416$
- Batch size 6000
- Size of yolov4-custom\_best.weights = 256 MB
- Size of yolov4-tiny- custom\_best.weights = 23.5 MB
- Test System configuration: i7 i7-10750H CPU @ 2.60 GHz/RTX 2060 (6 GB)/Memory 32 GB RAM.

The training dataset consists of DJI Mini’s images from the public-domain and images captured by our team. YOLOv4 Tiny/YOLOv3 Tiny are simplified versions of YOLOv3/YOLOv4. YOLO-Tiny algorithm developed for embedded devices with poor data processing capabilities. Its model structure is simple but the detection accuracy is low.

Figure 8 shows our experimental outcomes with different YOLO models on detecting intruder drone. In this test, we found that only YOLOv4 can detect the Intruder drone target with  $>80\%$  accuracy up to a 5 m distance. But the YOLO-Tiny version is faster and more lightweight than others. This mission requires detection of the Intruder drones at maximum distance. To conclude, the YOLO v4 model is the best choice.

The next stage of our experiment is to improve the accuracy in the YOLO v4 model with different sized image sets— $320 \times 320$ ,  $416 \times 416$ ,  $608 \times 608$ . The image size is dimensions width and height of the neural network in the YOLOv4 configuration file during training. As shown in Table 1, we can see that when the image size increases, we get more accuracy percentage on the detection, but it also incurs more processing time during training the dataset. We have decided to use YOLOv4 with  $416 \times 416$  frame size model for this performance verification stage for near optimality.

**Table 1** YOLOv4 training image size comparison

Distance (m)	Accuracy at image size		
	320 × 320	416 × 416	608 × 608
1.0	97.65	96.92	99.16
1.5	88.53	95.12	98.28
2.0	75.25	92.73	98.36
2.5	55.23	91.16	94.35
3.0	55.23	87.91	93
3.5	Not detected	86.23	85.69
4.0	Not detected	85.27	76.43
4.5	Not detected	83.36	65.64
5.0	Not detected	81.58	Not detected

## 6 Conclusion and Future Work

In this work, we have presented the context, design, and functionality of the Smart Drone Controller (SDC) framework. The real-time object detection mission implemented with ML engine in the SDC framework prototype was verified. The intruder drone was detected with 80% accuracy consistently. This is an important use case that leverages the current developments in Machine Learning/Deep Learning domain and introduce value-added benefits to smart city planning, governance, and support. Our Smart Drone Controller (SDC) enables access to more collaborative drone-based services and technologies, support standardization of cross-platform development efforts, facilitate seamless integration of drone networks in a digital economy and lifestyle.

At present, the SDC framework design and features are still at an early stage of development and the software is in beta mode. There is potential for improvement, and we plan to add:

- Collision avoidance to ensure safe operations of mission routes.
- Collect the metrics like route length, minimum distance, battery used, total time, time per each computation step, no. of unsuccessful missions, etc.
- Optimization of ML algorithms to reduce the latency and improve accuracy for a chosen mission.
- Add features to enhance the drone’s network security and operational security.
- More real-time object detection applications like finding building wall cracks.
- Develop the SDC version to work in Windows OS and mobile device platform with Android OS.

**Acknowledgments** This research work is funded by the Ministry of Education, Singapore Translational R&D and Innovation Fund under grant MOE2019-TIF-0027.

## References

1. Drones are a part of Singapore's Smart Nation strategy. <https://opengovasia.com/drones-are-a-part-of-singapores-smart-nation-strategy/>. Accessed 1 Sep 2021
2. Mohamed N, Al-Jaroodi J, Jawhar I, Idries A, Mohammed F (2020) Unmanned aerial vehicles applications in future smart cities, Technological forecasting and social change, vol 153. <https://doi.org/10.1016/j.techfore.2018.05.004>
3. Gharibi M, Boutaba R, Waslander SL (2016) Internet of drones. *IEEE Access* 4:1148–1162. <https://doi.org/10.1109/ACCESS.2016.2537208>
4. Lagkas T, Argyriou V, Bibi S, Sarigiannidis P (2018) UAV IoT framework views and challenges: towards protecting drones as “Things”. *Sensors* 18(11):4015. <https://doi.org/10.3390/s18114015>
5. Park J, Choi S, Ahn I, Kim J (2019) Multiple UAVs-based surveillance and reconnaissance system utilizing IoT platform. In: 2019 International conference on electronics, information, and communication (ICEIC), 2019, pp 1–3. <https://doi.org/10.23919/ELINFOCOM.2019.8706406>
6. Schilling F, Lecoer J, Schiano F, Floreano D (2019) Learning vision-based flight in drone swarms by imitation. *IEEE Robot Autom Lett* 4(4):4523–4530. <https://doi.org/10.1109/LRA.2019.2935377>
7. Martínez-Alpiste I, Casaseca-de-la-Higuera P, Alcaraz-Calero J, Grecos C, Wang Q (2019) Benchmarking machine-learning-based object detection on a UAV and mobile platform. *IEEE Wirel Commun Netw Conf (WCNC) 2019*:1–6. <https://doi.org/10.1109/WCNC.2019.8885504>
8. Mandal S, Biswas S, Balas VE, Shaw RN, Ghosh A (2021) Lyft 3D object detection for autonomous vehicles. *Artif Intell Fut Gener Robot* 119–136. <https://doi.org/10.1016/B978-0-323-85498-6.00003-4>
9. Mandal S, Md Basharat Mones SK, Das A, Balas VE, Shaw RN, Ghosh A (2021) Single shot detection for detecting real-time flying objects for unmanned aerial vehicle. *Artif Intell Fut Gener Robot* 37–53. <https://doi.org/10.1016/B978-0-323-85498-6.00005-8>
10. Biswas S, Bianchini M, Shaw RN, Ghosh A (2021) Prediction of traffic movement for autonomous vehicles. In: Bianchini M, Simic M, Ghosh A, Shaw RN (eds) *Machine learning for robotics applications*. Studies in computational intelligence, vol 960. Springer, Singapore. [https://doi.org/10.1007/978-981-16-0598-7\\_12](https://doi.org/10.1007/978-981-16-0598-7_12)
11. Soni A, Dharmacharya D, Pal A, Srivastava VK, Shaw RN, Ghosh A (2021) Design of a machine learning-based self-driving car. In: Bianchini M, Simic M, Ghosh A, Shaw RN (eds) *Machine learning for robotics applications*. Studies in computational intelligence, vol 960. Springer, Singapore. [https://doi.org/10.1007/978-981-16-0598-7\\_11](https://doi.org/10.1007/978-981-16-0598-7_11)
12. Sam Daley, “Fighting Fires and Saving Elephants: How 12 Companies are using the AI Drone to solve Big Problems”, 30 April 2019, <https://builtin.com/artificial-intelligence/drones-ai-companies/>
13. Domo Tactical Communications. <https://www.unmannedsystemstechnology.com/company/domo-tactical-communications/>. Accessed 26 July 2021
14. Skylark Drones. <https://skylarkdrones.com/>. Accessed 26 July 2021
15. Kaur K (2021) One-north to be designated as Singapore's first drone estate. *The Straits Times*, 5 Feb 2018: <https://www.straitstimes.com/singapore/transport/one-north-to-be-designated-as-singapores-first-drone-estate>. Accessed 26 July 2021
16. DroneDeploy: <https://www.dronedeploy.com/solutions/>. Accessed 26 July 2021.
17. Garuda Robotics: <https://garuda.io/>. Accessed 26 July 2021
18. Peter KK Loh, Tang LZ, Forest SL Tan, Wee C (2017) CELLS—a cross-platform, real-time security framework for IoT endpoints. Grant MOE 2016-TIF-1-G-02, 27 Jan 2017
19. YOLO v4: <https://github.com/AlexeyAB/darknet>
20. Srivastava S, Divekar AV, Anilkumar C et al (2021) Comparative analysis of deep learning image detection algorithms. *J Big Data* 8:66. <https://doi.org/10.1186/s40537-021-00434-w>
21. Paper YOLO v4: <https://arxiv.org/pdf/2004.10934.pdf>