Persistent Airborne Surveillance using Semi-Autonomous Drone swarms

Akhil Bandarupalli
abandaru@purdue.edu
Purdue University

Noah Weston
noah.d.weston.civ@mail.mil
Army Research Laboratory

Dhruv Swarup
dswarup@purdue.edu
Purdue University

Somali Chaterji
schaterji@purdue.edu
Purdue University

ABSTRACT

Persistent surveillance is an essential tool in many damage prevention and mitigation scenarios. Existing approaches to conduct surveillance over an area either require deploying static equipment on the ground or are expensive and have a high ecological footprint. The era of commercial drones has facilitated the development of lightweight, agile vehicles that can carry a wide variety of payloads with a very low ecological footprint. In this paper, we present a new airborne surveillance system using drone swarms. We explore the concept of Swarm Utility and present a mobility model for persistent surveillance that achieves maximum utility while being energy efficient. We program the drones in the swarm to follow trajectories defined by the model, eliminating the need to control the drones manually. Additionally, we augment our system with an object detection framework that runs on accumulated video feed streamed by the swarm to detect objects of interest in the area with high accuracy.

CCS CONCEPTS

• Computer systems organization → Robotic control.

KEYWORDS

Persistent Surveillance, Drones, UAV, Drone swarms, Mobility model, Object Detection

ACM Reference Format:

Akhil Bandarupalli, Dhruv Swarup, Noah Weston, and Somali Chaterji. 2021. Persistent Airborne Surveillance using Semi-Autonomous Drone swarms. In *The 7th Workshop on Micro Aerial Vehicle Networks, Systems, and Applications (DroNet) (DroNet '21)*, June 24, 2021, Virtual, WI, USA. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3469259.3470487>

1 INTRODUCTION

Persistent surveillance is the process of persistently monitoring an area for any event of interest. It is an essential tool in many scenarios such as criminal apprehension, disaster management, reconnaissance missions, wildlife protection and traffic monitoring [3].

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs International 4.0 License.

DroNet '21, June 24, 2021, Virtual, WI, USA

© 2021 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-8599-2/21/06.

<https://doi.org/10.1145/3469259.3470487>

An area of interest is typically monitored by security cameras that transmit live video feed to control rooms. In big cities like Los Angeles and in remote areas, law enforcement agencies also use choppers to circle an area of interest at low heights to monitor for adverse events. The pilot relays information about the events to on-ground units that use this information and take necessary action. These approaches have their own limitations with respect to deployment, maintenance, and security. Deploying security cameras around an area requires on-ground infrastructure and regular maintenance to tolerate asset destruction and theft. Helicopter operations have triggered concerns about the safety of personnel in adverse weather conditions and the noise from whirring choppers [6].

Unmanned Aerial Vehicles (UAVs) or drones can carry a wide variety of payloads. Many commercial drones come with HD and Infrared Cameras, which are an ideal choice for providing HD video feed of on-ground events in real-time. They are relatively easy to deploy and maintain than security cameras, and their mobility gives them an edge over fixed infrastructure. They are also much safer to operate, cost-effective, and are quieter than choppers. Recent developments in communication technology have extended their operation ranges [4].

Drones are operated by pilots through a Ground Control Station (GCS). Piloting a drone has a learning curve that requires an understanding of the aerial dynamics of drone flights. Today's commercial drones are equipped with features like waypoint navigation and trajectory following using GPS, which ease the burden on the pilot. The drone also streams HD video to the GCS, using which the pilot can monitor the area. This approach, called a semi-autonomous drone flight, works well for an individual pilot-drone pair. In the semi-autonomous setting, a) Drones do not take decisions independently or coordinate with each other. They only move using GCS commands. b) The GCS sends commands according to pre-defined trajectories without the pilot having to control every drone in the swarm manually.

Challenges and Contributions. Individual drones have a restricted Field-of-View (FoV) and limited energy resources, which limits their effectiveness in an application like persistent surveillance over big areas. However, swarms of drones functioning as a distributed system can be used to overcome these limitations and scale surveillance to larger areas. Keeping the limitations of drones in mind, a drone swarm must achieve maximum utility, while not wasting energy. We formally define drone and swarm utility in Section 2. Various works that have been published in the past [1, 2, 12] lack emphasis on swarm utility. These swarm models are not suitable

for persistent surveillance because they are not optimized for utility and periodic motion. We present a swarm mobility model that achieves maximum utility and discuss a tradeoff between the frequency of visitation and the area covered by the swarm with a fixed number of drones. Additionally, we automate the deployment and control of the swarm by implementing a semi-autonomous trajectory setting. We also implement an object detection system based on YOLOv3 [13, 14] that processes video feed from the swarm to assist the drone pilot in surveillance. An added advantage of our system is that it is non-invasive, does not involve complex drone engineering, and is compatible with any commercial drone that supports waypoint navigation. We evaluated this system in a parking lot scenario with a testbed of 2 drones functioning as a swarm where the system tracks vehicles moving in and out of the area. To approximate the computation in future variants of our drone-based surveillance, we will leverage our approximate vision algorithms that also satisfy end-to-end latency requirements that may be needed for near-real time decision making [15, 21, 22].

2 PROBLEM STATEMENT AND SOLUTION APPROACH

Our surveillance system contains a swarm of drones of size n . We deploy our system for a coverage area of A . In A , we assume that adverse events can occur throughout the area uniformly randomly at any given time. Say an event e occurred at location l on the map. The time of occurrence is t_o , and the time of event sighting by a drone is t_s . We require:

$$t_d = t_s - t_o \leq \Delta \quad (1)$$

where Δ is the maximum event sighting delay. The objectives we are trying to achieve are (a) Keep the maximum time delay $\leq \Delta$ and (b) Maximize coverage A while objective (a) is true. We assume that the swarm operates in a semi-autonomous mode and takes commands from the GCS. The swarm streams video of the area to the GCS, where we run our object detection.

Traditionally, persistent surveillance using security cameras implied continuous observation with no time delay, i.e., $\Delta = 0$. However, a constant finite time delay Δ is a reasonable compromise for higher coverage. We make use of the mobility of drones to create a dynamic system with user-specified characteristics.

Drone Utility and Swarm Utility. The area covered by a drone $A_{drone}(t)$ is the total area captured by the drone's camera from time $t - \Delta$ to t . According to Eq. (1), the coverage of an area expires after time Δ . The utility of a drone ρ is the rate of change of $A_{drone}(t)$. Similarly, we define Swarm Utility σ to be the rate at which the swarm covers area.

$$\rho = \frac{dA_{drone}(t)}{dt} \quad (2)$$

$$\sigma = \frac{dA_{swarm}(t)}{dt} \quad (3)$$

We say that a drone swarm is at its maximum utility at time t when the areas covered by drones do not overlap at t .

We approach the problem in the following steps:

(1) We capture the dependence of ρ on drone height h and velocity v . Additionally, we analyze the impact of swarm structure on σ .

(2) Based on the principles of swarm utility, we design a swarm mobility model for surveillance that exhibits maximum swarm utility, based on the Circular Mobility Model [20].

(3) We map the tradeoff between delay Δ and the coverage A and present insights on optimizing swarm configurations.

3 RELATED WORK

Drones have been used in applications like search and rescue [9], mobile network coverage [5, 16], and surveillance [10]. Danilchenko et. al. [5] talks about statically deploying a drone swarm to provide network coverage to targets on the ground. Static deployments surveil lesser area compared to mobile swarms because they do not move. Bouachir et. al. and Bujari et. al. [1, 2] present mobility models for UAV swarms that prioritize ad-hoc network connectivity and area coverage. These works describe inherently randomized mobility models, which are not suited for persistent surveillance with strict Δ requirements. Pokhrel et. al. [12] talks about a convoy of drones that travel in a tightly packed manner. Tightly packed swarms have low utility because their coverage area overlaps with one another.

Montanari et. al. [10] presented a system where a drone autonomously decides its trajectory by assessing the situation on the ground using an object detection program on an NVIDIA Jetson TX2, mounted on the drone. This work is orthogonal to our system, where the job is not to act autonomously but to reliably transmit the feed of any perceived threats to the operator with minimal delay. Singh et. al. [17] also presents a surveillance system using a drone that recognizes violent humans through object detection. Mishra et. al. [9] uses object detection but the drone is trained to recognize humans calling for help in a disaster situation. These systems use only one drone and do not scale to large areas. Huang et. al. [8] describes a concept known as Robot Patrolling in which patrolling robots maintain a given interval between visits. However, their techniques are not optimized for drones and on-ground robots have distinct motion dynamics from surveillance drones.

4 SYSTEM OVERVIEW

4.1 Drone Utility

Drone utility ρ depends on the height h and horizontal velocity v of the drone. We assume that the angle-of-view of the drone's camera is θ . A modern consumer drone comes with a gimbal that positions the camera at a given angle α . The Field of View of the drone is shown in Fig. 1. The width of the FoV w is calculated as follows.

$$w = h(\tan(\frac{\theta}{2} - \alpha) + \tan(\frac{\theta}{2} + \alpha)) \quad (4)$$

Gimbal angle. The gimbal angle can be varied by the type of objects to be recognized and the corresponding object detection accuracy. However, we fix the $\alpha = 0$ for our analyses.

$$w = 2h \cdot \tan(\frac{\theta}{2})$$

$$\implies \rho = 2 \cdot h \cdot v \cdot \tan(\frac{\theta}{2}) \quad (5)$$

When $\alpha = 0$, the FoV can be assumed to be rectangular, with the larger side $\vec{w}$ and the smaller side $\vec{b}$. Given that the drone is moving with velocity $\vec{v}$, the covered area is maximum when its movement

Figure 1: FoV of a drone, flying at height h , camera view angle θ , and gimbal angle α

direction is perpendicular to $\vec{w}$, $\implies \vec{v} \perp \vec{w}$. We configure the drone and the camera such that this condition is always true. Drone utility:

$$\rho = \frac{dA_{drone}}{dt} = w \frac{db}{dt} = wv \quad (6)$$

As a drone moves in a specific direction, the area covered by it is a sweep of the FoV along the trajectory. ρ increases linearly with the velocity v and height h . However, the information captured about objects on the ground deteriorates with the increase in h and v . The relationship between h , v , and the object detector's accuracy depends on factors like the type of objects, topology of the training set, and the drone's camera quality. We assume that for a given drone, there will exist an optimal ρ_{max} when the object detector can detect all the objects on ground with minimal information loss. We empirically compute the optimal height and velocity.

4.2 Swarm Utility

We assume all the drones to move at the same height h and velocity v to ensure the homogeneity of the swarm. A swarm of n drones is at its maximum utility σ_{max} when their FoVs do not overlap with each other. For example, consider n drones arranged in a straight line and are moving perpendicular to the line. σ is maximum when the drones are separated by at least w . This swarm achieves a $\sigma_{max} = n\rho$.

Deviation from trajectories. Drones can deviate from their trajectories due to ambient forces. When the swarm utility is $\sigma_{max} = n\rho$, minor deviations from trajectories will result in surveillance gaps. We assume δ to be the maximum deviation from the intended trajectory.

4.3 Mobility Model

A drone conducting persistent surveillance must be in a periodic motion to satisfy Eq. (1). We deploy the drone swarm in concentric circular regions centered at point p and separated by $w_d = w_{max} - 2\delta$ as shown in Fig. 3. This deployment model restricts the surveillance

Figure 2: FoV of a swarm with drones at different distances.

Figure 3: Circular orbit model for drone swarms.

area to be circular. However, applications like criminal apprehension and disaster management do not generally have strictly defined boundaries for surveillance. Hence, we prioritize maximizing swarm utility σ and judiciously spending the swarm's energy resources while ensuring that Eq. (1) holds throughout the deployed area.

The radius of the first circle is $\frac{w_d}{2}$, and the subsequent radii increase as an arithmetic progression with common difference w_d . We call these circles orbits in which drones rotate. The area under surveillance in orbit i :

$$A_i = \pi w_d^2 ((i)^2 - (i-1)^2) = 2\pi(i-0.5)w_d^2$$

We allocate n drones to orbits by deploying $f(i)$ drones in the i th orbit. We assert that $f(i) \geq 1\forall i$. Δ in the i th orbit is:

$$\Delta_i = \frac{2\pi w_d(i-0.5)}{v_{max}f(i)} \quad (7)$$

Given that the maximum delay in all orbits is Δ ,

$$f(i) = \left\lceil \frac{2\pi w_d(i-0.5)}{v_{max}\Delta} \right\rceil \quad (8)$$

Initial Positioning. The drones rotating in a particular orbit must always be equidistant from each other. Before the swarm starts

surveillance, drones in orbit i must be positioned in an equidistant manner. Out of $f(i)$ drones in the orbit i , we position the drone j at location:

$$l_j = ((i - 0.5)w_d \cdot \cos(\phi_j), (i - 0.5)w_d \cdot \sin(\phi_j), h_{max})$$

$$\phi_j = \frac{2\pi(j - 1)}{f(i)}, \forall j \in [1, f(i)]$$

Swarm utility σ will depend on the estimate of trajectory deviation δ . We assume that the deviation will be negligible compared to w , implying $w_d \approx w$. Since consecutive orbits are separated by w_d , FoVs of drones in different orbits will not overlap. Hence, the swarm utility $\sigma \approx \sigma_{max}$. Additionally, we also need to keep in mind that packing way too many drones into a single orbit will result in overlapping FoVs for drones in the same orbit, reducing σ .

Predictability and Energy-Efficiency of Circular motion. Circular motion is the most energy-effective way to cover an area periodically. The energy spent by a drone in *changing its direction* to return to its starting point is zero in case of circular motion. The centrifugal force $F_c = \frac{mv^2}{r}$ acting on a drone moving in a circular path does not do any work on the drone because the force and distance vectors are perpendicular to each other. However, this property is true for other trajectories with smooth turns. The unique advantage of circular orbits over other smooth-turn trajectories is if the velocity of the drone moving in a circle is constant, the acceleration is also constant throughout the journey. This property ensures that the drone's motion is homogeneous throughout the trajectory, making it easy to estimate the worst case delay Δ . Additionally, this predictability is also of great use in predicting other parameters of drone flight, such as time and place of battery exhaustion and pinpoint location identification of perceived threats in the presence of network and detection delays. Wan et. al. and Wang et. al. [19, 20] capture other benefits of circular trajectories, e.g., low collision probability between drones.

Coverage vs. delay (Δ) tradeoff. When the size of the swarm is n , the value of Δ directly influences the swarm's coverage. However, there must be at least one drone in every orbit so the maximum area that the swarm can surveil is:

$$A_{max} = \pi w_d^2 n^2$$

$$\Delta = \Delta_{max} = \frac{2\pi w_d (n - 0.5)}{v_{max}}$$

On the other end of the spectrum, we can achieve $\Delta = 0$ by deploying the swarm statically over an area.

$$A_{min} = nA_{drone} = nwb$$

$$\Delta = \Delta_{min} = 0$$

Other forms of surveillance applications. $f(i)$ can be specified according to the type of surveillance required. In case of perimeter surveillance [18], we deploy more drones in the outer orbits to ensure stricter vigilance on the borders. Similarly, when the task is protecting a target from adversaries, inner orbits should have a higher drone concentration.

4.4 Object Detection

The pipeline used to process the video feed from the drones is based on the Real-Time Messaging Protocol (RTMP). The drone streams

Figure 4: Overview for object detection from drone feed.

video to the GCS. The GCS uploads the video feed to a web server using the RTMP protocol—a protocol commonly integrated with popular drone APIs. The RTMP server acts as an ingest for multiple incoming drone video feeds. The client then pulls these streams and passes them through the object detection API (YOLOv3). Whenever the client detects an object of interest, it sends notifications containing the following: the drone location, the timestamp, the position in the image frame (bounding boxes), class of object detected, and the detection confidence.

5 EXPERIMENTS AND PERFORMANCE ANALYSIS

We evaluated our system in a parking lot setting where we monitored the movement of cars. For our experiments, we used a system of two drones - DJI Phantom 4 Pro and DJI Mavic 2 Pro. We programmed the drone's trajectories using DJI Mobile SDK, more specifically, the Android API. The embedded Jetson-class device—Xavier AGX—hosted an nginx [11] RTMP server, to which the GCS uploads video feeds. The same Xavier also acted as the client that pulled the streams from the server using OpenCV, and passed the frames through YOLOv3 [14], trained on [7].

5.1 FoV of Drone

We experimentally determined the maximum height at which the drone can reliably detect all the vehicles on the ground. We progressively increased the Drone height h and measured the client's detection accuracy as Mean Average Precision (mAP) in Fig. 5a. After $h = 150\text{ft}$ or 45m , the accuracy of the program degrades gradually. Hence, we set the $h = 150\text{ft}$. The detection accuracy can be traded for coverage area by deploying the swarm at higher altitudes. We also study the variation of detection accuracy w.r.t. velocity in Fig. 5b. The maximum velocity of both the drones in our experiment in a circular trajectory is $v = 8.94\text{m/sec}$. At $h = 45\text{m}$, the mAP is maximum and does not vary much with changing velocity. Hence, we set the velocity of the swarm $v = 20\text{mph}$ or 8.94m/sec .

Trajectory deviation. We set the drone on a circular orbit of radius $\frac{w}{2}$. We measured the maximum deviation of the drone from its intended path by logging its GPS traces. We found the maximum deviation to be strictly less than 10ft . or 3m , even under wind speeds of 20mph . We experimented in a region with high GPS availability (12 satellites available). We expect the deviation to be higher under low GPS availability.

(a) mAP vs. h. mAP is 1 at $h = 45m$ or $150ft$, after which it reduces gradually.

(b) mAP vs. v. The mAP is at its maximum and stays relatively constant with respect to velocity at $h = 45m$. As we increase the height, the detection accuracy falls very fast.

Figure 5: Plots showing variation of Mean Average Precision (mAP) with respect to Height h and Velocity v . The mAP values are measured at 0.50 IoU threshold and class confidence greater than 70%

(a) Delay (Δ) heatmap for $\Delta = 60$ seconds over an area $1.64sq.km$. Number of drones required to cover this area with $\Delta = 60$ seconds is $n = 36$. Values in the color bar are mentioned are in seconds. The radius of each orbit is mentioned in meters.

(b) Plot showing the relationship between coverage area and Δ with swarm configured according to Eq. (8), with swarm sizes $n = 20, 30, 50$. The subplot shows the areas covered by static deployments (SD) of swarm sizes $n = 20, 30, 50$, with $\Delta = 0$. Drones in static deployments do not move, rather hover in place. The subplot also compares the areas covered by the SDs to areas covered by swarms following our mobility model with $\Delta = 15$ seconds.

Figure 6: Plots demonstrating the relationship between coverage area and delay (Δ).

5.2 Delay Heatmap

We simulated the mobility model in NS3 network simulator, where we plugged in values calculated above: $w_d = 90.51m$, $v_{max} = 8.94m/sec$, $h_{max} = 45m$, and observed the performance of the system. We set $\Delta = 60$ sec and deployed drones according to Eq. (8). The heatmap in Fig. 6a displays the Δ across the area.

The first few orbits have low delay values compared to the outer orbits. It is because of the step function and the constant term of 0.5 in Eq. (8). As the orbit number increases, the linear term overpowers

the constant, and the colors get darker, bringing the delay closer to 60 sec. The light concentric rings in between any two consecutive orbits are of width 2δ . Drones from both orbits visit these points in the 2δ rings because of which the delay is small.

5.3 Area vs Delay (Δ)

We plot the relationship between the coverage by the swarm and the Δ required throughout the map. In this experiment, we fix the number of drones in the swarm to be $n = 50$ and observe

Figure 7: The image is from a drone flying at $h_{max} = 150$ ft. and cars detected by the client are marked accordingly.

the relationship between Δ and deployed area. The plot indicates that the coverage (area) by the swarm increases with Δ . A swarm moving according to the mobility model covers more than 3 times the area covered by a static swarm with delay $\Delta = 15$ seconds. The

Table 1: We observe the relationship between A and delay (Δ) to be piece-wise linear (3 pieces) Fig. 6b; slope of each piece is proportional to swarm size. The slope values here are derived for a swarm with $n = 50$.

Regions	R^2	Equation
R1	0.994	$A = 0.03\Delta + 0.77$
R2	0.985	$A = 0.02\Delta + 4.09$
R3	0.994	$A = 0.01\Delta + 10.73$

relationship between *coverage* and *delay* (Δ) is piece-wise linear with 3 pieces. We derived the relationship by fitting the data in Fig. 6b into a linear regression function, which gave the data in Table 1.

6 CONCLUSION AND FUTURE WORK

In this paper, we studied semi-autonomous drone swarms in persistent airborne surveillance. We presented a mobility model with maximum swarm utility that guarantees a configured delay between drone visits. In future, we will explore fully autonomous coordinating swarms and the impact of faulty drones on these systems. We will also extend this framework to other applications and datasets like intrusion detection, wildlife surveillance and disaster management.

ACKNOWLEDGMENTS

We thank John Scott, Digital Agriculture Coordinator at Purdue Extension, for his help with the drone flights in the parking lots of ACRE, Purdue. This work has been sponsored by the Army Research Laboratory (ARL) under grant W911NF-20-2-0026.

REFERENCES

[1] Ouns Bouachir, Alinoe Abrassart, Fabien Garcia, and Nicolas Larrieu. 2014. A mobility model for UAV ad hoc network. In *2014 International Conference on*

Unmanned Aircraft Systems, ICUAS 2014 - Conference Proceedings. IEEE Computer Society, 383–388. <https://doi.org/10.1109/ICUAS.2014.6842277>

[2] Armir Bujari, Carlos T Calafate, Juan-Carlos Cano, Pietro Manzoni, Claudio Enrico Palazzi, and Daniele Ronzani. 2017. Flying ad-hoc network application scenarios and mobility models. *International Journal of Distributed Sensor Networks* 13, 10 (2017), 1550147717738192. <https://doi.org/10.1177/1550147717738192>

[3] M.J. Carlotto. 1997. Detection and analysis of change in remotely sensed imagery with application to wide area surveillance. *IEEE Transactions on Image Processing* 6, 1 (1997), 189–202. <https://doi.org/10.1109/83.552106>

[4] SZ DJI Technology Co. 2018. Specs. <https://www.dji.com/smart-controller/specs>.

[5] Kiril Danilchenko and Michael Segal. 2020. Connected Ad-Hoc Swarm of Drones. In *Proceedings of the 6th ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications (DroNet '20)*. Association for Computing Machinery, New York, NY, USA, Article 4, 6 pages. <https://doi.org/10.1145/3396864.3399699>

[6] Léon Dische-Becker. 2020. Police helicopters circling a city are environmental terrorism. <https://massivesci.com/articles/helicopter-los-angeles-environment-new-nature/>

[7] Meng-Ru Hsieh, Yen-Liang Lin, and Winston H. Hsu. 2017. Drone-Based Object Counting by Spatially Regularized Regional Proposal Network. In *2017 IEEE International Conference on Computer Vision (ICCV)*. 4165–4173. <https://doi.org/10.1109/ICCV.2017.446>

[8] Li Huang, Mengchu Zhou, Kuangrong Hao, and Edwin Hou. 2019. A survey of multi-robot regular and adversarial patrolling. *IEEE/CAA Journal of Automatica Sinica* 6, 4 (2019), 894–903. <https://doi.org/10.1109/JAS.2019.1911537>

[9] Balmukund Mishra, Deepak Garg, Pratik Narang, and Vipul Mishra. 2020. Drone-surveillance for search and rescue in natural disaster. *Computer Communications* 156 (apr 2020), 1–10. <https://doi.org/10.1016/j.comcom.2020.03.012>

[10] Alessandro Montanari, Fredrika Kringberg, Alice Valentini, Cecilia Mascolo, and Amanda Prorok. 2018. Surveying Areas in Developing Regions Through Context Aware Drone Mobility. In *Proceedings of the 4th ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications* (Munich, Germany) (*DroNet'18*). Association for Computing Machinery, New York, NY, USA, 27–32. <https://doi.org/10.1145/3213526.3213532>

[11] nginx. [n.d.]. An open source web server. <https://www.nginx.com/>.

[12] Shiva Raj Pokhrel, Jiong Jin, and Hai Le Vu. 2019. Mobility-aware multipath communication for unmanned aerial surveillance systems. *IEEE Transactions on Vehicular Technology* 68, 6 (2019), 6088–6098.

[13] Joseph Redmon. 2013–2016. Darknet: Open Source Neural Networks in C. <http://pjreddie.com/darknet/>.

[14] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You Only Look Once: Unified, Real-Time Object Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 779–788. <https://doi.org/10.1109/CVPR.2016.91>

[15] Karthick Shankar, Pengcheng Wang, Ran Xu, Ashraf Mahgoub, and Somali Chaterji. 2020. JANUS: Benchmarking Commercial and Open-Source Cloud and Edge Platforms for Object and Anomaly Detection Workloads. In *2020 IEEE 13th International Conference on Cloud Computing (CLOUD)*. IEEE, 590–599.

[16] Pankaj K. Sharma and Dong In Kim. 2019. Random 3D Mobile UAV Networks: Mobility Modeling and Coverage Probability. *IEEE Transactions on Wireless Communications* 18, 5 (may 2019), 2527–2538. <https://doi.org/10.1109/TWC.2019.2904564>

[17] Amarjot Singh, Devendra Patil, and S. N. Omkar. 2018. Eye in the Sky: Real-Time Drone Surveillance System (DSS) for Violent Individuals Identification Using ScatterNet Hybrid Deep Learning Network. In *2018 IEEE Computer Vision and Pattern Recognition CVPR Workshops 2018, June 18–22, 2018*. IEEE Computer Society, 1629–1637. <https://doi.org/10.1109/CVPRW.2018.00214>

[18] Cristina Gabriela Sărăcin, Ioan Dragoș, and Aurel Ionuț Chirilă. 2017. Powering aerial surveillance drones. In *2017 10th International Symposium on Advanced Topics in Electrical Engineering (ATEE)*. 237–240. <https://doi.org/10.1109/ATEE.2017.7905185>

[19] Yan Wan, Kamesh Namuduri, Yi Zhou, and Shengli Fu. 2013. A Smooth-Turn Mobility Model for Airborne Networks. *IEEE Transactions on Vehicular Technology* 62, 7 (2013), 3359–3370. <https://doi.org/10.1109/TVT.2013.2251686>

[20] Wei Wang, Xiaohong Guan, Beizhan Wang, and Yaping Wang. 2010. A novel mobility model based on semi-random circular movement in mobile ad hoc networks. *Information Sciences* 180, 3 (2010), 399–413. <https://doi.org/10.1016/j.ins.2009.10.001>

[21] Ran Xu, Rakesh Kumar, PengCheng Wang, Peter Bai, Ganga Meghanath, Somali Chaterji, Subrata Mitra, and Saurabh Bagchi. 2021. ApproxNet: Content and Contention Aware Video Analytics System for the Edge. *ACM-TOSN* (2021), 1–28.

[22] Ran Xu, Chen-lin Zhang, Pengcheng Wang, Jayoung Lee, Subrata Mitra, Somali Chaterji, Yin Li, and Saurabh Bagchi. 2020. ApproxDet: content and contention-aware approximate object detection for mobiles. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*. 449–462.