Joint frame rate adaptation and object recognition model selection for stabilized unmanned aerial vehicle surveillance

Gyu Seon Kim © | Haemin Lee © | Soohyun Park © | Joongheon Kim ©

Abstract
We propose an adaptive unmanned aerial vehicle (UAV)-assisted object recognition algorithm for urban surveillance scenarios. For UAV-assisted surveillance, UAVs are equipped with learning-based object recognition models and can collect surveillance image data. However, owing to the limitations of UAVs regarding power and computational resources, adaptive control must be performed accordingly. Therefore, we introduce a self-adaptive control strategy to maximize the time-averaged recognition performance subject to stability through a formulation based on Lyapunov optimization. Results from performance evaluations on real-world data demonstrate that the proposed algorithm achieves the desired performance improvements.

Keywords
adaptive control, object recognition, surveillance, UAV

1 | INTRODUCTION

The use of unmanned aerial vehicles (UAVs) has increased rapidly in a wide range of applications including autonomous aerial transportation, mobile base stations, aerial surveillance, and environmental observation. It offers advantages such as accessibility, cost-effectiveness, efficiency, safety, and accuracy. Therefore, UAVs are playing an important role in numerous industrial applications [1]. Because of their accessibility and adaptability, UAVs can record high-resolution aerial images and videos to monitor crowds, detect potential threats, and recognize visual events of unexpected situations [2, 3]. Modern UAVs are equipped with high-resolution sensors and machine learning algorithms to perform complicated tasks, such as detecting and tracking specific objects.

Object recognition is essential in computer vision for localizing object instances and can be used in applications that rely on remote sensing or tracking. For example, it can be used in urban planning, mapping, and monitoring to detect objects in harsh and inaccessible environments. Conventional object recognition relies on satellites and manned aircraft that generally follow predetermined paths and may temporarily alter their routes and hover to complete specific tasks. However, satellites and manned aircraft are expensive, limiting their applicability [1]. UAVs can replace conventional platforms for remote sensing to swiftly obtain high-resolution images at low altitudes. Moreover, UAVs can move with fewer restrictions compared than ground mobile vehicles and collect video data without geographic restrictions. As UAV data are highly informative in terms of content and temporal resolution, object recognition using UAVs has
increased dramatically in many applications. However, UAVs have limited battery capacity and computational resources, restricting their operational duration and capabilities. Therefore, the computational capabilities, accuracy, and operational speed of UAVs must be balanced.

In general, latency, accuracy, and energy allocation have been studied in UAVs and many other systems. For example, Ji and others [4] comprehensively studied the optimization of latency, accuracy, and energy consumption for wireless federated learning over long periods. Fang and others [5] proposed a wireless network that optimized the latency associated with high power consumption in wireless networks and achieved high performance even at low power levels. Zhang and others [6] proposed an algorithm to optimize the transmit power allocation in non-orthogonal multiple access systems. Wang and others [7] solved a resource allocation optimization problem using integrated sensing and communication wireless networks to achieve low latency and energy efficiency for green networks. To address poor environmental perception, they used low-complexity optimization to obtain near-optimal latency performance.

Most modern object recognition algorithms are based on convolutional neural networks (CNNs). High accuracy can be achieved by using many convolutional layers at the expense of a high computational cost. Consequently, model complexity restrains the deployment in devices with limited computational resources, particularly for long-term surveillance or large-scale image data processing. To avoid this problem, CNN-based object recognition algorithms have been optimized for efficient computation through architectures such as lightweight CNNs [8], thereby reducing the number of parameters [9] and pruning connections [10]. These methods aim to reduce the computational cost while maintaining the learning accuracy. Nevertheless, optimization inevitably sacrifices accuracy to reduce the computational burden. Hence, the tradeoff between computational cost and accuracy must be balanced when deploying optimized CNN-based object recognition models [11, 12].

We consider a real-time object recognition scenario using UAVs [13, 14]. Real-time object recognition algorithms consist of an image queue in which sampled images from videos are stored. The image processor components that detect and classify objects from the collected images are based on object recognition models. To consider and simulate real-world urban UAV-assisted surveillance scenarios, we use the VisDrone2019 dataset [15], which consists of 288 video clips containing 261 908 frames and 10 209 static images captured by various camera-mounted UAVs, covering a wide range of aspects including locations (taken from 14 different cities separated by thousands of kilometers in China), environments (urban and countryside), objects (e.g., pedestrians, vehicles, and bicycles), and densities (sparse and crowded scenes), as shown in Figures 1 and 2.

We observed a tradeoff between image/frame rates, object recognition models, and delays. If image/frame rates are higher, more arrivals are introduced into the image queue, consequently reducing stability. Although stability may be compromised, higher image/frame rates can enhance the learning accuracy and recognition. However, higher accuracy may introduce processing delays given the model complexity. Lyapunov optimization can be used to mathematically express the tradeoff between time-averaged utility maximization and stability [16]. Accordingly, we use joint image/frame rate adaptation and selection of object recognition deep learning models considering delays in terms of Lyapunov optimization.

The main contributions of this study are as follows:

- An adaptive UAV-assisted object recognition algorithm is proposed for urban surveillance applications and evaluated on real-world data.
- The proposed algorithm performs self-adaptive control for object recognition to maximize the time-average recognition performance subject to stability. It is based on queue backlog status on the UAV and inspired by Lyapunov optimization.

The remainder of this paper is organized as follows. Section 2 presents related work on conventional and UAV-assisted object recognition. Section 3 presents the reference system model and algorithm for UAV surveillance. Section 4 reports the evaluation of the proposed algorithm through simulations. Finally, we draw conclusions in Section 5.

2 | RELATED WORK

In this section, we discuss object recognition methods and control applications based on Lyapunov optimization.

2.1 | Object recognition methods

Object recognition based on deep learning algorithms can be divided into single- and two-stage approaches [17]. The single-stage approach offers fast computation for recognition. It simultaneously processes an image and directly predicts the location and size of an object. In other words, the model generates a certain number of
boxes at each location and then classifies whether the boxes correspond to objects. Among representative models of single-stage algorithms, you only look once (YOLO)/YOLOv2 [18] directly predicts the bounding box from an image through a CNN. The single-shot multibox detector is another representative algorithm designed to recognize objects of various sizes using feature maps of the convolutional layer in the middle of a CNN. It uses candidate boxes with different scales and aspect ratios per image block in the feature map and compares each box with a ground-truth box. On the other hand, two-stage approaches separately perform region proposal and detector learning. Two-stage algorithms generally reach a higher accuracy than single-stage ones, but they often incur a high computational cost. The region-based CNN [19] is a representative two-stage algorithm that first leverages a region proposal network to extract candidate regions likely containing an object and then uses a CNN to classify the candidate regions. Moreover, the fast region-based CNN extracts and applies features to candidate areas for classification and bounding box regression.

Owing to their efficient and flexible data acquisition capabilities, UAVs have gained popularity in computer vision and remote sensing. Inspired by recent advancements in deep learning, several advanced techniques for object detection have been extensively applied to a range of UAV-related tasks, including environmental monitoring, precision agriculture, and traffic management [1]. UAV-assisted object recognition using machine learning was developed in Budiharto and
leads plus-penalty (DPP) [25]. Minimizing the Lyapunov drift achieved with constant gap drift per timestep, time-averaged optimization can be minimize both the time-averaged objective and Lyapunov averaged objective function. By taking control actions to ing, in various fields [11, 12, 26-29] including video stream- subject to queue stability), is scalable and has been used optimization (i.e., time-averaged utility maximization punov optimization, which relies on stochastic network between utility and stability. Moreover, DPP-based Lyapunov optimization has been used to control super-resolution deep learning models for surveillance UAVs in security applications [12, 13]. Kim and others [11] proposed con- trollable neural networks and training principles for multi-rate super-resolution tasks.

3 | JOINT FRAME RATE ADAPTATION AND OBJECT RECOGNITION MODEL SELECTION

3.1 | Algorithm overview

Real-time object recognition requires considerable computational resources and rapid data transmission. In addition, it faces various challenges. First, the images can be large and complex, hindering real-time processing. Second, the processing of object recognition may be slow to process all incoming data. Finally, the recognition system may experience queue overflow if the incoming data rate is higher than the processing rate. Thus, controlling the frame rates may enable real-time object recognition. To this end, the rate at which information is gathered from the image source must vary. Specifically, adaptive sampling can control the frame rates depending on the complexity of the images. This method can aid in reduc- ing the data volume to be processed and enhance the efficiency of real-time processing.

3.2 | Algorithm details

We introduce an algorithm to simultaneously adjust the optimal frame rate and select the object recognition model to maximize the time-averaged object recognition performance with stability. We consider a single image queue with the arrivals and departures indicating the frame rates and processing by the selected object recognition model, respectively, as illustrated in Figure 3. Instead of adjusting the processing rate, the queue backlog can be evaluated to determine the current state and adjust the frame rate and object recognition model accord- ingly. We consider a scenario in which a UAV has its own image data buffer and the image data flow can be repre- sented by a queue. In addition, image data are assumed to be obtained at specific timesteps from a camera-mounted UAV.

In our system, a UAV should perform real-time object recognition under limited computational resources. If accurate object recognition models are used, a delay may occur in the image queue because the processing speed is lower than the arrival rate. Although less-accurate object recognition models reduce the delay, they undermine the
recognition accuracy to achieve fast inference. Therefore, a tradeoff exists between the objective/utility and stability. To handle this tradeoff, we propose a control action decision-making method to dynamically adjust the frame rate and select the object recognition model. To control the frame rates, the number of incoming image frames must be adjusted. A UAV keeps an image queue, and the frames in the queue are processed sequentially. These frames are fed into the object recognition model, and their processing indicates the departure from the queue. The arrival process is then determined by image acquisition, which provides image frames from a camera. A high frame rate (e.g., 60 fps) increases the computational burden of object recognition.

We formulate a mathematical problem for maximizing the time-averaged recognition accuracy for processing images from the queue, denoted by $F(a[t])$ for decision $a[t]$. It can be expressed as follows:

$$\max \lim_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} F(a(\tau))$$

subject to

$$\lim_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} Q[\tau] < \infty.$$

In (1), $F(a(\tau))$ is the object recognition model accuracy for queue departure $b(a[t])$ under decision $a[t]$ at timestep $t$. The decision generates a tradeoff between the maximization of the object recognition accuracy and stability of the queuing system. This tradeoff is related to the average queuing delay. A DPP-based Lyapunov optimization algorithm can be used to optimize the time-averaged utility function (i.e., the object recognition accuracy) subject to queue stability. For DPP [30, 31], the expectations can be opportunistically maximized. The corresponding Lyapunov function can be formulated as $L(Q[\tau]) = \frac{1}{2} Q^2[\tau]$, and the conditional quadratic Lyapunov function, $\Delta(\cdot)$, is given by $\mathbb{E}[L(Q[\tau + 1]) - L(Q[\tau])|Q[\tau]]$. As in Koo et al. [31], the dynamics are defined to achieve queue stability by minimizing the upper bound on DPP, that is, $\Delta(Q[\tau]) + \mathbb{V}[F(a[t])]$, where $\mathbb{V}$ is a tradeoff factor.

The upper bound of the drift of the Lyapunov function can be expressed as

$$L(Q[\tau + 1]) - L(Q[\tau]) = \frac{1}{2} (Q[\tau + 1]^2 - Q[\tau]^2) \leq \frac{1}{2} \left( a(a[t])^2 + b(a[t])^2 \right) + Q[\tau] (a(a[t]) - b(a[t])),$$

The upper bound of the conditional Lyapunov drift is given by

$$\Delta(Q[\tau]) = \mathbb{E}[L(Q[\tau + 1]) - L(Q[\tau])|Q[\tau]] \leq C + \mathbb{V}[Q[t](a(a[t]) - b(a[t]))|Q[\tau]],$$

and constant $C$ can be expressed as

$$\frac{1}{2} \mathbb{V}[a(a[t])^2 + b(a[t])^2|Q[\tau]] \leq C,$$

which assumes that both the arrival and departure rates are upper-bounded. Because $C$ is constant, the minimization of the upper bound on DPP leads to

$$\mathbb{V}[F(a[t])] - \mathbb{E}[Q[t] \cdot (-a(a[t]) + b(a[t]))].$$

Thus, the time-averaged maximization problem can be formulated as

---

**Figure 3** Diagram of model based on Lyapunov optimization and comprising image frame rate adaptation and model selection.
\[ V \in [F(a[t])] + \mathbb{E}[Q[t] \cdot (-a(a[t]) + b(a[t]))]. \]  

(7)

The proposed algorithm observes the current queue state, \( Q[t] \), and determines decision \( a[t] \) at every timestep \( t \) as follows:

\[
\alpha^*[t] \leftarrow \arg \max_{a \in A} V \cdot F(\alpha[t]) - Q[t] \cdot (a(a[t]) - b(a[t]))
\]

(8)

where \( \alpha^*[t], V, F(\alpha[t]), Q[t], a(a[t]), \) and \( b(a[t]) \) are the optimal decision control at time \( t \), tradeoff factor, object recognition accuracy depending on the decision control at time \( t \), queue stability, arrival depending on the decision control at time \( t \), and departure depending on the decision control at time \( t \), respectively. Let \( a[t] \) denote the decision control at time \( t \). For arrival, \( a(a[t]) \) is the number of frames per second. In the proposed algorithm, both \( a(a[t]) \) and \( b(a[t]) \) are controlled. \( V \) represents a predefined tradeoff factor that serves as the gain of the significance between object recognition accuracy and queue backlog. This significance is generally constant. If \( V \) is large, the system prioritizes recognition accuracy over queue stability. If \( V \) is small, the system prioritizes queue stability over recognition accuracy.

To validate (8), we first assume that \( Q[t] = 0 \). Then, (8) attempts to maximize \( V \cdot F(\alpha[t]) \), that is, the number of input images should increase, and a more accurate model should be selected to maximize the object recognition accuracy. This is semantically true because we can focus on the main objective, that is, accurate object recognition, because stability has already been achieved. Then, we assume that \( Q[t] \approx \infty \) and aim to minimize \( a(a[t]) \) (that is, the number of input images should decrease) and maximize \( b(a[t]) \) (i.e., the processing speed should increase) to maintain queue stability using a faster model. This is true because stability should be mainly considered for large \( Q[t] \), but we sacrifice the object recognition accuracy to avoid overflow. Hence, we confirm that the proposed closed-form formulation in (8) controls \( a[t] \) to maximize the time-averaged recognition accuracy subject to queue stability.

3.3 Pseudocode and computational complexity

The pseudocode of the proposed algorithm is presented in Algorithm 1. In Algorithm 1 and (8), \( A \) is the set of \( a[t] \) frames per second, with \( a[t] \in \{10, 30, 60\} \). In lines 1–3, the optimizations variables and parameters are initialized. The algorithm is executed at every timestep, as shown in line 5. In line 6, current state \( Q(t) \) is used in (8). Lines 8–14 describe the main computational procedure for (8).

**Algorithm 1 Maximization of stabilized recognition accuracy**

```
1: Input: \( Q[t], F[t], V \)
2: Output: \( \alpha^*[t] \)
3: Initialize \( t = 0, Q(0) = 0, \forall a[t] \in A \)
4: Maximization of stabilized recognition accuracy:
5: while \( t \leq T \) do
6: \( \text{Observe } Q[t]; \)
7: \( T^* \leftarrow \infty; \)
8: for \( a[t] \in A \) do
9: \( T \leftarrow V \cdot F(a[t]) - Q[t](a[a(t)] - b[a(t)]); \)
10: if \( T \leq T^* \) then
11: \( T^* \leftarrow T; \)
12: \( \alpha^*[t] \leftarrow a[t]; \)
13: end if
14: end for
15: end while
```

4 PERFORMANCE EVALUATION

4.1 Evaluation setup

YOLOv5 is designed for real-time object recognition, which is important for UAVs that must navigate rapidly changing environments. In addition, YOLOv5 has a lightweight architecture unlike other object recognition models, possibly reducing power consumption. YOLOv5 is focused on speed and efficiency, being convenient for saving energy in UAVs, whose computing platform can be active for relatively short periods. These characteristics of YOLOv5 make it suitable for our application because UAVs have limited power and computational resources. A real-world flight experiment was conducted using a quadcopter UAV flying at low altitude and using YOLOv5 for object recognition [32].

The proposed method was implemented using YOLOv5 on the PyTorch 1.7.1 library and the VisDrone2019 dataset. All the models used an NVIDIA RTX 2080 Super graphics processor for training and testing. During training, we used part of the pretrained model from YOLOv5n and YOLOv5s, which were trained using stochastic gradient descent with a momentum of 0.937 and weight decay of 0.0005. In addition, the batch size was set to 32, and training proceeded over 300 epochs. We set the size of the input images to 640 × 640 pixels for both models. The dataset had 10 categories: pedestrian, person, bicycle, car, van, truck, tricycle, awning tricycle, bus, and motor. It contained 6471 training images, 548 validation images, and 1580 test images. For
performance evaluation, we considered the inference time to recognize objects from input images, recognition accuracy, and queue backlog, which measures the queue stability.

4.2 Evaluation results

As listed in Table 1, the inference time for both the graphics and conventional processors is longer for YOLOv5s (heavy model). Both models achieve real-time performance in the graphics processor but not in the conventional processor. Object recognition using YOLOv5 typically relies on the algorithm proposed in Lao and Sundaramoorthi [22] and runtime object detection to maximize real-time accuracy (ROMA) [23]. The algorithm in Lao and Sundaramoorthi [22] focuses on queuing stability in the experimental environment used in this study, while ROMA focuses on performance. Unlike these algorithms, Lyapunov optimization balances queuing stability and performance. As the algorithm in Lao and Sundaramoorthi [22] and ROMA focus on similar control objectives, they are suitable for comparison with the proposed approach based on Lyapunov optimization.

Figure 4 shows the change in queue backlog, $Q(t)$, according to the change in tradeoff factor $V$. In (8), the object recognition accuracy, $F(\alpha[t])$, and queue backlog, $Q(t)$, are opposing variables. As the queue backlog is nearly empty per unit time, the queue stability of the minimum delay model is maintained at the expense of a reduction in the recognition accuracy. Consequently, zero latency can be achieved. The relative significance of the accuracy is adjusted according to the value of tradeoff factor $V$, which is the coefficient of $F(\alpha[t])$. A higher $V$ value increases the importance of $F(\alpha[t])$ and reduces the importance of the queue backlog. According to $V$, as a heavy model with many input images improves accuracy, the point of convergence is delayed, and the value of the queue backlog increases. Except for the maximum accuracy model, all the models converge at a specific point in time.

UAVs cannot keep large image queues owing to battery, storage, and throughput limitations. Therefore, an appropriate balance between recognition accuracy and queue backlog should be determined. In the maximum accuracy model, the stability of the system cannot be guaranteed for the UAV model because the value of the queue backlog increases over time. The queue backlog continues to increase and diverges over time, causing a large system delay. However, in the proposed model, both the delay and stability of the utility function can be

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Performance of object recognition for various models on VisDrone2019 dataset.</th>
</tr>
</thead>
<tbody>
<tr>
<td>mAP50</td>
<td>YOLOv5n</td>
</tr>
<tr>
<td></td>
<td>28.0</td>
</tr>
<tr>
<td>mAP95</td>
<td>14.5</td>
</tr>
<tr>
<td>Inference time (GPU)</td>
<td>4.5 ms</td>
</tr>
<tr>
<td>Inference time (CPU)</td>
<td>60 ms</td>
</tr>
</tbody>
</table>

Figure 4: Queue backlog according to tradeoff factor $V$. Figure 5: Average queue backlog.
secured by selecting an appropriate tradeoff factor $V$. For $V = 100\,000\,000$, the queue backlog in the model stabilizes after approximately 450 timesteps. Even for $V = 40\,000\,000$ and $8\,000\,000$, a heavy model with many input images is used to improve the utility until just before reaching the critical point, after which the queue backlog stabilizes. In a UAV with limited battery and throughput, the proposed method provides an appropriate balance between the utility function and queue backlog and improved system efficiency by properly adjusting the delay and accuracy.

Figure 5 shows the average cumulative queue backlogs. Each box plot indicates the median (red line), minimum and maximum values (whiskers), and 25% and 75% quartiles (box edges). The maximum accuracy model uses the heavy model to improve accuracy. Thus, over time, the queue backlog increases, and the average queue backlog reaches the highest value. The minimum delay model has a reduced accuracy to maintain the queue backlog stability. Therefore, the average queue has the lowest value in the minimum delay model. As the value of $V$ increases, the importance of accuracy increases, and because a heavy model is used for a longer time, more queues accumulate over time, leading to increase in the average queue. The proposed model maintains both the queue stability and accuracy between those of the maximum accuracy and minimum delay models.

Figure 6 shows the accuracy and queue backlog over time. The maximum accuracy model (green line) always uses the heavy model for ensuring high accuracy, and the minimum delay model (blue line) uses the light model to stabilize the queue backlog. In Figure 6A,D, for $V = 8\,000\,000$, the queue backlog becomes stable after approximately 40 timesteps. Before 40 timesteps, the heavy model increases the accuracy, and then the light model is selected. Around the critical point of 40 timesteps, the accuracy fluctuates, and then the light model is used. Simultaneously, the queue backlog is stabilized. For $V = 100\,000\,000$, the queue continues to diverge and then oscillates at the critical point of 460 timesteps, after which the queue backlog stabilizes. For the maximum accuracy model, the queue backlog becomes unstable and diverges, causing overflow of the queue backlog after a given period. Considering the UAV characteristics, the proposed algorithm can find a balance between accuracy and queue backlog. Hence, the queue stability and recognition accuracy are suitable to increase the UAV coverage time [33].

5 | CONCLUSIONS AND FUTURE WORK

We designed and implemented a novel adaptive UAV-assisted object recognition algorithm for urban
surveillance. We designed a UAV-assisted surveillance system for implementing a machine learning object recognition model. The system effectively collected surveillance image data by exploiting the UAV mobility and flexibility. As UAVs have limited power and computational resources, a balance between recognition accuracy and delay should be achieved because a higher CNN recognition accuracy incurs a higher computational cost (delay). Therefore, we adopted self-adaptive control to maximize the time-averaged recognition performance subject to stability through a formulation based on Lyapunov optimization. Experimental results on a real-world dataset confirmed that the proposed algorithm achieved the intended performance improvements.

CONFLICT OF INTEREST STATEMENT
The authors declare no conflicts of interest.

ORCID
Gyu Seon Kim https://orcid.org/0000-0002-5559-9749
Haemin Lee https://orcid.org/0000-0002-7524-1672
Soohyun Park https://orcid.org/0000-0002-6556-9746
Joongheon Kim https://orcid.org/0000-0002-1794-6076

REFERENCES
21. J. Shin, M. J. Piran, H.-K. Song, and H. Moon, UAV-assisted and deep learning-driven object detection and tracking for autonomous driving, (Proceedings of International ACM MobiCom Workshop on Drone Assisted Wireless Communications for 5G and Beyond, Sydney, Australia), 2022, pp. 7–12.
AUTHOR BIOGRAPHIES

Gyu Seon Kim is a PhD student in the Department of Electrical and Computer Engineering, Korea University, Seoul, Republic of Korea. He received the BS degree in aerospace engineering from Inha University, Incheon, Republic of Korea. His research focuses on deep reinforcement learning algorithms and their application to autonomous mobility systems.

Haemin Lee is with Korea University, Seoul, Republic of Korea, since September 2020, and is currently a PhD candidate in the Department of Electrical and Computer Engineering, Korea University, Seoul, Republic of Korea. She received the BS degree in statistics from Sookmyung Women’s University, Seoul, Republic of Korea, in 2019. Her research focuses on deep learning algorithms and their application to communication networks.

Soohyun Park is a postdoctoral scholar at Korea University, Seoul, Republic of Korea, since 2023, where she received her PhD in electrical and computer engineering. She received her BS degree in computer science and engineering from Chung-Ang University, Seoul, Republic of Korea, in 2019. Her research focuses on deep learning algorithms and their application to big data platforms and networking. She was also the recipient of the IEEE Vehicular Technology Society (VTS) Seoul Chapter Award in 2019.

Joongheon Kim is with Korea University, Seoul, Republic of Korea, since 2019, where he is currently an associate professor in the Department of Electrical and Computer Engineering and an adjunct professor in the Department of Communications Engineering (co-operated with Samsung Electronics) and the Department of Semiconductor Engineering (co-operated with SK Hynix). He received the BS and MS degrees in computer science and engineering from Korea University, Seoul, Republic of Korea, in 2004 and 2006, respectively, and the PhD degree in computer science from the University of Southern California (USC), Los Angeles, CA, USA, in 2014. Before joining Korea University, he was a research engineer at LG Electronics (Seoul, Republic of Korea, 2006–2009), systems engineer at Intel Corporation Headquarters (Santa Clara in Silicon Valley, CA, USA, 2013–2016), and an assistant professor of computer science and engineering at Chung-Ang University (Seoul, Korea, 2016–2019). He served as an editor and guest editor for IEEE Transactions on Vehicular Technology, IEEE Transactions on Machine Learning in Communications and
Networking, IEEE Communications Standards Magazine, Computer Networks (Elsevier), and ICT Express (Elsevier). He is also a distinguished lecturer at the IEEE Communications Society (ComSoc) and the IEEE Systems Council. He is also the Executive Director of the Korea Institute of Communication and Information Sciences (KICS). He was the recipient of Annenberg Graduate Fellowship with his PhD admission from USC (2009) and the Intel Corporation Next Generation and Standards (NGS) Division Recognition Award (2015), IEEE Systems Journal Best Paper Award (2020), IEEE ComSoc Multimedia Communications Technical Committee (MMTC) Outstanding Young Researcher Award (2020), IEEE ComSoc MMTC Best Journal Paper Award (2021), Best Special Issue Guest Editor Award by ICT Express (2022), and Best Editor Award by ICT Express (2023). He has also received several awards from IEEE conferences, including IEEE ICOIN Best Paper Award (2021), IEEE Vehicular Technology Society (VTS) Seoul Chapter Awards (2019, 2021, and 2022), and IEEE ITC Best Paper Award (2022).

How to cite this article: G. S. Kim, H. Lee, S. Park, and J. Kim, Joint frame rate adaptation and object recognition model selection for stabilized unmanned aerial vehicle surveillance, ETRI Journal 45 (2023), 811–821. DOI 10.4218/etrij.2023-0121