applied sciences

MDPI

Article

A Multi-Stage Approach to UAV Detection, Identification, and Tracking Using Region-of-Interest Management and Rate-Adaptive Video Coding

Dongkyu 'Roy' Lee [1], Sanghong Kim [1], Namkyung Yoon [1], Wonki Seo [2] and Hwangnam Kim [1,*]

[1] School of Electrical Engineering, Korea University, Seoul 02841, Republic of Korea; roylee6315@korea.ac.kr (D.'R.L.); sanghongkim@korea.ac.kr (S.K.); nkyoon93@korea.ac.kr (N.Y.)
[2] Nextcore Technology Co., Ltd., Seoul 05854 , Republic of Korea; all@next-core.co.kr
* Correspondence: hnkim@korea.ac.kr; Tel.: +82-2-3290-4821

Abstract: The drone industry has opened its market to ordinary people, making drones prevalent in daily life. However, safety and security issues have been raised as the number of accidents rises (e.g., losing control and colliding with people or invading secured properties). For safety and security purposes, observers and surveillance systems must be aware of UAVs invading aerial spaces. This paper introduces a UAV tracking system with ROI-based video coding capabilities that can efficiently encode videos with a dynamic coding rate. The proposed system initially uses deep learning-based UAV detection to locate the UAV and determine the ROI surrounding the detected UAVs. Afterward, the ROI is tracked using optical flow, which is relatively light in computational load. Furthermore, our devised module for effective compression, XROI-DCT, is applied to non-ROI regions, so a different coding rate is applied depending on the region during encoding. The proposed UAV tracking system is implemented and evaluated by utilizing videos from YouTube, Kaggle, and a video of 3DR Solo2 taken by the authors. The evaluation verifies that the proposed system can detect and track UAVs significantly faster than YOLOv7 and efficiently encode a video, compressing 70% of the video based on the ROI. Additionally, it can successfully identify the UAV model with a high accuracy of 0.9869 ROC–AUC score.

Keywords: UAV; object detection; object classification; video encoding

check for updates

Citation: Lee, D.'R.; Kim, S.; Yoon, N.; Seo, W.; Kim, H. A Multi-Stage Approach to UAV Detection, Identification, and Tracking Using Region-of-Interest Management and Rate-Adaptive Video Coding. *Appl. Sci.* **2024**, *14*, 5559. https://doi.org/10.3390/app14135559

Academic Editor: Douglas O'Shaughnessy

Received: 17 May 2024
Revised: 16 June 2024
Accepted: 19 June 2024
Published: 26 June 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Unmanned Aerial Vehicles (UAVs), commonly known as drones, have evolved from niche gadgets into pivotal tools in various applications, ranging from military operations to commercial and research endeavors. UAVs are now more prevalent than ever. Initially confined to military usage, UAVs have increasingly permeated numerous civilian sectors, including simple navigation, environmental monitoring, agricultural surveying, and notably, security and surveillance [1]. Crucial to such applications, the development of advanced UAV tracking systems that ensure precise operation and navigation—often in complex, dynamic environments—is in demand. A critical component of effectively leveraging UAVs lies in developing advanced UAV tracking systems, mainly through video surveillance. These systems are designed to enhance the capabilities of UAVs in complex environments by providing real-time data and insights essential for navigation, obstacle avoidance, and mission-specific tasks. Video surveillance, integrated with state-of-the-art computer vision technologies, plays a pivotal role in the real-time object tracking and operational management of UAVs. According to Statista [2], the market for commercial UAVs is expected to reach 13 billion U.S. dollars (USD) by 2025. As of October 2023, there are 863,728 UAVs registered with the Federal Aviation Administration in the United States alone [3]. As the number of UAVs—regardless of whether they are used for commercial or military purposes—increases, the number of incidences caused by UAVs has increased [4].

Incidences involving UAVs flying into restricted aerial zones are increasing along with the total number of incidences. To prevent such incidents, detecting and tracking the object—in this case, a UAV—reliably and accurately would be the first essential step.

Early works were primarily focused on enhancing the detection and tracking accuracy under constrained environments by employing classical computer vision techniques such as background subtraction and global/local contrast [5]. As the field progressed, more sophisticated approaches were introduced, leveraging the power of artificial intelligence and machine learning. These approaches have been utilized in many other UAV-related areas, such as topology control [6] and location-based routing [7]. Notably, algorithms like Kalman and particle filters were adapted for dynamic and uncertain environments to improve tracking robustness [8]. In recent studies, integrated deep learning algorithms, such as convolutional neural network (CNN) [9], are adopted to autonomously learn from vast amounts of data, significantly improving the detection of UAVs against varied backgrounds and during challenging conditions.

Furthermore, real-time object detection, such as YOLO (You Only Look Once) [10], has revolutionized the field by reducing latency and increasing object recognition accuracy in videos. This feature is indispensable in scenarios where speed and accuracy are crucial, such as monitoring fast-moving targets or navigating through rapidly changing environments, making it highly suitable for UAV surveillance and tracking.

In this paper, a UAV surveillance and tracking system with a suitable video coding system for limited resource devices is introduced. The proposed system is based on the region-of-interest (ROI) video coding technique and adopts object tracking to classify and track ROI. Efficient ROI management is attained by using YOLOv7 [11] for setting ROI based on object detection with deep neural network and optical flow [12] for tracking ROI. Video coding alone is a power-consuming process. Moreover, such deep learning tasks and image processing for small devices can be overwhelming due to limited resources and lack of GPU [13]. For capturing a moving object in a 30 frame-per-second or above video [14], object detection becomes a power-consuming algorithm. Applying UAV detection and video coding on every frame is unsuitable for such devices. On the other hand, calculating optical flow over a small area with a small number of features to track can run even on a mobile device—usually known for its limited resources. Therefore, the proposed system adopts optical flow. The optical flow of the ROI is calculated to track the movement of the object inside the ROI; the movement is utilized to adjust the ROI in the next frame, achieving efficient ROI management. Before encoding, we apply our devised scheme called XROI-DCT. With XROI-DCT in the proposed system, UAV tracking videos can achieve high transmission efficiency. The result shows that the accuracy of our approach is almost identical to that of YOLOv7 at a lower computation price [15]. The performance of the proposed system is evaluated by comparing it to YOLOv7 in various aspects. The contribution of this paper is presented as follows:

- We present a UAV tracking system. It utilizes YOLOv7's real-time detection capabilities and optical flow for a lightweight and fast UAV tracking system for limited resource devices. In the evaluation section, experiment results show that the proposed scheme can perform UAV detection 315.86% faster than YOLOv7.
- Our proposed system is devised as a surveillance/tracking system that considers video transmission. Due to XROI-DCT, the system is suitable for efficient video transmission. The file size of the transmitted video is significantly smaller (averaging 70% compression) than the file size of the same video processed by YOLOv7.
- For predicting UAV models, our proposed system performs UAV identification based on a CNN model. Our UAV identifier can distinguish 16 UAV models with high accuracy, and its ROC–AUC score is 0.9869.

The rest of the paper is organized as follows. In Section 2, we present the relative research regarding ROI-based video coding. In Section 3, an overall description of the system is presented. Section 4 presents the prototype implementation and performance evaluation

of the implementation. The paper is concluded in Section 5 with a brief presentation of our future research.

2. Related Works

UAV tracking involves several technologies to meet user demands. UAV detection requires a real-time processing capability; UAV identification should achieve great accuracy so that the system can determine the model and distinguish the target UAV from other UAVs. An effective coding system is needed for video storage and data transmission on limited-resource devices, such as IoT and mobile devices. We can summarize this paper as an image-based UAV identification and tracking system focusing on deep learning and image processing. Coding systems must produce quality video with adequate file size to be used by limited resource devices. Consisting of all the capabilities above, this presents a challenging research topic, and the following are studies related to our work.

2.1. Object Detection Based on Deep Learning

While many object detection algorithms were already introduced, R-CNN was the first deep learning-based object detection algorithm [16]. After R-CNN, its successors, such as Fast R-CNN [17] and Faster R-CNN [18] have been devised. While these successors could not achieve real-time processing due to expensive training costs and region proposals, YOLO was introduced and has been evolving ever since. YOLO is fundamentally based on a Convolutional Neural Network (CNN). By processing the entire image in a single evaluation, YOLO accomplishes bounding box prediction (object detection) and inferring class probability (object classification) in a single pass through CNN. This makes YOLO different from other detection systems where multiple networks or passes might be necessary. YOLOv7 is a more recent model of the YOLO family. Its architectural design improved performance by focusing on optimization. These modules and optimization methods are called trainable bag-of-freebies; they improve detection accuracy without increasing the inference cost. Additionally, it is already proven that the YOLOv7 can detect small flying objects such as UAVs [19]. Due to the nature of the UAV surveillance setting—capturing images of flying UAVs from a distance—and the fast maneuverability of UAVs capable of flying 60 km/h, quickly determining the bounding boxes accurately makes YOLOv7 a suitable choice for the proposed system.

2.2. ROI-Based Video Coding

Various researchers adopt ROI-based video coding schemes to increase video quality without increasing the file size. ROI-based video coding schemes divide video images into two regions, ROI and non-ROI, and allow a system to encode with two different bit rates. The means of setting and managing ROI is a popular research topic in ROI-based video coding schemes. The research by Liu et al. uses the Means–Absolute–Difference (MAD) prediction method to set ROI and adjust the Quantization Parameter (QP) value of H.264 encoder to differentiate the compression rates of ROI and non-ROI [20]. In the research by Arachchi et al., the authors utilize the Flexible Macroblock Ordering (FMO) tool in H.264 video coding standard [21]. The work encodes the background and foreground regions separately by dividing them into two slices. The writers claim that the proposed technique improves the objective quality of the foreground by more than 1 dB. Zheng et al. use the visual perceptual mechanism of the Human Visual System (HVS) on the H.264 standards [22]. The paper claims that using HVS improves overall visual quality and states that the PSNR of the important region improves by 1.2 dB. The aforementioned researchers set ROI in a video to separate the important and unimportant regions. By following the standards, the researchers present a method to improve the quality of the important region. In this paper, the system predicts the ROI differently from H.264 or VP9 [23] so that regions of detected UAVs in the video are determined more accurately, and eventually improving QoE by applying encoding at the correct region.

2.3. Link-Aware Video Encoding and Streaming

CoSA, a cognitive video streaming system, streams video with compression rates set according to the communication state [24,25]. The important area in a video is identified using an IR depth camera. Then, CoSA streams the important area with higher quality and the unimportant area with lower quality. Image encoding and decoding are handled by the CoSA encoder and decoder. The CoSA encoder adopts high-frequency deletion techniques to encode a video, and the decoder adopts high-frequency restoration techniques to decode the video. The CoSA decoder not only decodes the video for viewing but also periodically transmits the link state to the encoder to adjust the compression rate. The experiments show that the system can deliver video data with two varying compression rates.

Even though CoSA introduced a new image streaming system that is robust to the dynamic link state, it is not compatible with H.264 standards. Videos encoded using the CoSA codec cannot be decoded with the original H.264 standard decoder, so the system's practicality is limited. CoSA's codec is efficient and well-suited for conversational video, but the incompatibility with the standardized codec limits its applicability. Moreover, CoSA utilizes an IR depth camera to identify important areas. The requirement of an IR depth camera further limits the applicability of CoSA. The proposed system in the paper produces a more lightweight video and uses image processing for ROI selection, which does not require additional hardware.

3. System Overview

This section describes the system's structure and explains the algorithm associated with it in detail.

3.1. Operation Flow

Figure 1 shows a simplified operation flow of the system. The system contains an efficient ROI detection module, an ROI management module, an identification module, and an XROI-DCT/encoding module.

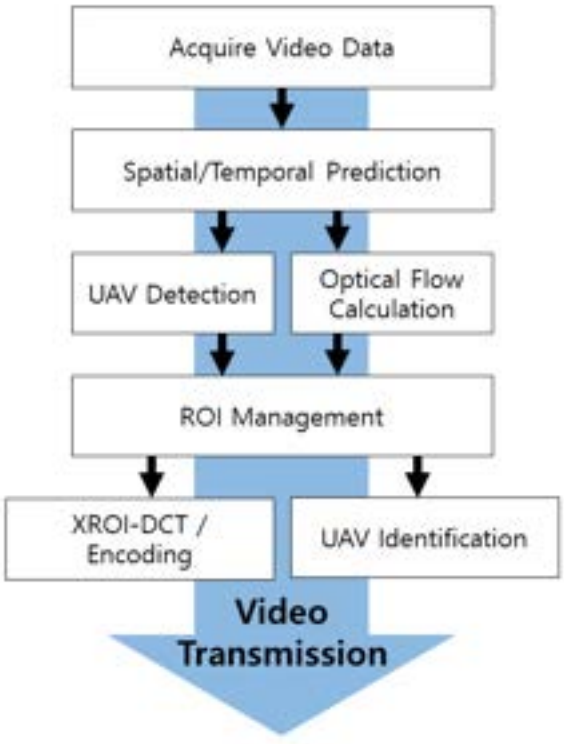

Figure 1. Operation flow diagram of the proposed system.

The video is first loaded onto the system, and then each frame is analyzed to determine its frame type. When video frames are loaded for the prediction, the proposed system also performs UAV detection and calculates optical flow. If the frame is predicted to be an I-frame, the system performs UAV detection to locate the UAV in the video. For frames predicted as non-I frames, optical flow is calculated based on the current non-I frame and the previous frame. The UAV detection and optical flow calculation operation saves the detected UAV information and the optical flow information for each frame, which are used to in ROI management operations. The ROI management operation is after the UAV detection and optical flow calculation operation. The ROI management operation manages the location of ROI using the detected UAV information and optical flow information of each frame. Before ROI management operations are conducted, the system analyzes the frame and performs frame prediction [26]. Such prediction based on predictive coding exploits spatial and temporal redundancy between video frames for file size compression. The reason for conducting such a process is to find key frames called I-frames. For I-frames, the system sets ROI according to the area where the UAV is predicted to be by the UAV detector based on YOLO. For other frames, the system manages the ROI by moving ROI (from the previous frame) according to the motion vector derived from the optical flow information. Then, frames are ready for XROI-DCT. The frames are divided into smaller macroblocks, which are discrete cosine transformed for compression. XROI-DCT is based on discrete cosine transform (DCT) [27]. During XROI-DCT, the high frequencies are deleted to provide extra compression on non-ROI macroblocks. This forces the encoder to give a high compression rate to the non-ROI. Transmission efficiency can be dramatically improved by giving a different compression rate on ROI and non-ROI areas. Afterwards, the proposed system follows the standard MPEG4 encoding process.

As shown in Figure 1, from acquiring video to ROI management is executed in a single flow. However, precisely speaking, 'XROI-DCT/Encoding' is included in the single flow. Inference by our UAV identifier does not guarantee real-time execution. So, UAV identification—querying for inference from our CNN—is conducted in parallel. This decision is based on the fact that the model of an already-detected UAV is highly unlikely to change as long as it is constantly tracked. Therefore, UAV identification does not have to be executed for every frame: this saves execution time and computing resources.

3.2. ROI Management

In this subsection, a detailed description of ROI management operation is depicted, and the algorithm that manages the ROI displacement is presented. ROI management, consisting of ROI classification and ROI tracking, is performed after the frame prediction.

3.2.1. ROI Classification

For frames that are determined as I-frames, detected UAV information, achieved using an object detection algorithm, is employed to classify ROI during ROI management operation. The proposed system adopts YOLOv7 for the UAV detection algorithm. YOLOv7 is a real-time object detector based on a well-known deep-learning architecture called CNN.

By processing an image through CNN, the algorithm detects objects. YOLO achieves high speeds and high efficiency by integrating detection tasks—typically performed in multiple stages—into one. YOLO detects objects in real time. This speed makes it suitable for applications requiring real-time processing, such as video analysis, autonomous driving, and real-time surveillance. The details of YOLO can be found in [28,29]. YOLO shows the detected object, in this case, a UAV, on the video frame (or image) by drawing a bounding box on the image. Figure 2a shows an example of the result. Each bounding box indicates that the detection algorithm determines the region to contain an object. This region in the bounding box is set as ROI in the proposed system. Even though the adopted UAV detection algorithm is an efficient algorithm that claims to support real-time detection, it would only mean anything if such a live stream could be sent to others. By performing UAV detection on only I-frames, the proposed system significantly decreases the overhead

of UAV detection. For other frames, the target UAV is tracked by applying optical flow and moving the ROI according to the target's movement. Every time an I-frame is detected, the ROI is set as the bounding box by YOLO. This prevents the ROI from drifting due to accumulated errors.

a

b

Figure 2. ROI management effect. (**a**) ROI classification using UAV detection. (**b**) ROI tracking using optical flow.

3.2.2. ROI Tracking

The proposed system adopts optical flow to manage the movement of ROI for the frames that were not used by UAV detection. Figure 2b demonstrates the information that results from optical flow calculation, and Figure 3 depicts how optical flow is adopted to calculate the motion vector of ROI in the proposed system. When an object shown in a video moves between frames, the motion vector of the object between two frames can be calculated using optical flow. In Figure 3a, motion vector v refers to the movement of a square object. The proposed system calculates the motion vector v using optical flow

values. Figure 3b depicts the optical flow values of pixels inside a particular area. Optical flow calculation produces motion vectors of pixels, as shown in the figure. Figure 2b shows an example of optical flow values extracted from a video. The figures imply that the optical flow values of the UAV area are consistent, and the UAV's motion vector can be calculated. The motion vector of ROI is the average of the optical flow values of pixels in the ROI. The proposed system estimates the displacement that needs to be applied to the ROI to contain the important content inside it by utilizing optical flow values of the ROI. Thus, the movement of UAVs can be tracked, and the ROI can be adjusted according to it.

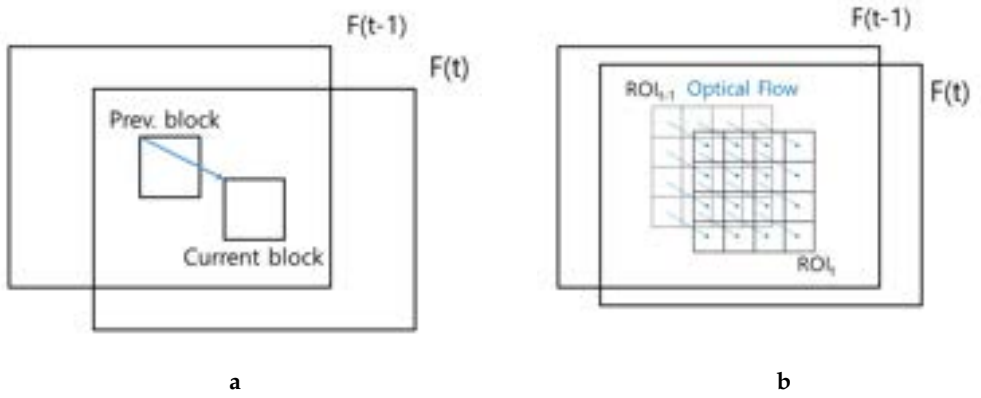

Figure 3. Optical flow for ROI displacement. (**a**) Motion vector between subsequent frames. (**b**) Motion vector calculation.

The proposed system only calculates the optical flow of the pixels inside the ROI to minimize calculation. Calculation of optical flow over such region returns optical flow values of every pixel in the ROI. However, simply averaging the optical flow values does not produce an accurate motion vector of the ROI. The ROI will not only contain UAVs, but the backgrounds, too. The portion of background in the ROI is not negligible. Therefore, optical flow values of the background pixels will count as errors and hinder the accuracy of the resulting motion vector. Thus, when the proposed system calculates the motion vector of the ROI, it ignores the optical flow values of the background by ignoring the optical flow values close to zero. However, errors by optical flow values of the background can be accumulated. Thus, updating the ROI based on the UAV detector is essential for accurate ROI tracking. The displacement of the ROI can be formulated as the following equation:

$$\begin{aligned} v_x &= \overline{OF_x}, \quad \text{where } OF_x = \{\forall \| OF(x,y)_x \| > t\}, \\ v_y &= \overline{OF_y}, \quad \text{where } OF_y = \{\forall \| OF(x,y)_y \| > t\}. \end{aligned} \tag{1}$$

In Equation (1), v_x and v_y represent the displacement of ROI in the x-axis and y-axis, respectively; $OF(x,y)_x$ and $OF(x,y)_y$ represent the horizontal and vertical component of optical flow value at (x,y); and t is the previously set threshold value. This equation calculates the mean of the movement of valid pixels inside ROI. With Equation (1), the proposed system estimates the area of the UAV on a frame without using the UAV detection algorithm on every frame by utilizing optical flow.

3.2.3. ROI Adjustment

Often the size and shape of the ROI must be adjusted to match the movement of the object inside the ROI. If an object inside the ROI moves toward the camera, the size of the ROI should increase to accommodate the increasing size of the object; the size of the ROI should decrease if an object moves away from the camera. This can be determined by evaluating the optical flow values of ROI subregions. In the proposed system, optical values are calculated for four ROI subregions: the top, bottom, left, and right subregions.

The optical values of ROI subregions are compared to adjust the width and height of ROI. Process is shown in Algorithm 1.

Algorithm 1 ROI adjustment

Initialize: $MV_l \leftarrow zero, MV_r \leftarrow zero, MV_t \leftarrow zero, MV_b \leftarrow zero$;
$count_l \leftarrow zero, count_r \leftarrow zero, count_t \leftarrow zero, count_b \leftarrow zero$;
$OF(x, y) \leftarrow$ optical flow values for all pixels (x, y) inside ROI

```
for each pixels in ROI do
    if x ≤ midpoint of ROI then
        if OF(x,y)[h] ≤ −1 || OF(x,y)[h] ≥ 1 then
            MV_l ← MV_l + OF(x,y)[h]
            count_l ← count_l + 1
        end
        if y ≤ midpoint of ROI then
            if OF(x,y)[v] ≤ −1 || OF(x,y)[v] ≥ 1 then
                MV_t ← MV_t + OF(x,y)[v]
                count_t ← count_t + 1
            end
        end
        else
            if OF(x,y)[v] ≤ −1 || OF(x,y)[v] ≥ 1 then
                MV_b ← MV_b + OF(x,y)[v]
                count_b ← count_b + 1
            end
        end
    end
    else
        if OF(x,y)[h] ≤ −1 || OF(x,y)[h] ≥ 1 then
            MV_r ← MV_r + OF(x,y)[h]
            count_r ← count_r + 1
        end
        if y ≤ midpoint of ROI then
            if OF(x,y)[v] ≤ −1 || OF(x,y)[v] ≥ 1 then
                MV_t ← MV_t + OF(x,y)[v]
                count_t ← count_t + 1
            end
        end
        else
            if OF(x,y)[v] ≤ −1 || OF(x,y)[v] ≥ 1 then
                MV_b ← MV_b + OF(x,y)[v]
                count_b ← count_b + 1
            end
        end
    end
end
if count_l > number of pixels in ROI/4 then
    Update left bound of ROI
end
if count_r > number of pixels in ROI/4 then
    Update right bound of ROI
end
if count_b > number of pixels in ROI/4 then
    Update bottom bound of ROI
end
if count_t > number of pixels in ROI/4 then
    Update top bound of ROI
end
```

Conclusively, Algorithm 1 describes the overall process of ROI management. First, the variables that manage the boundary of the ROI are initialized. Then, the object bounded by the ROI is tracked by calculating the optical flow values of the pixels inside the ROI. The ROI is divided into two subgroups to analyze the horizontal and vertical boundary movement. The ROI is divided into two subregions, the left and right regions, to check its horizontal boundary adjustment. The optical values of pixels in the left subregion are added up for left-side boundary adjustment. The same applies to the right-side boundary adjustment with the average optical flow value of the right subregion. The top and bottom boundary adjustments are processed in the same way using top and bottom subregions, which are retrieved by dividing the ROI horizontally from the middle. The average optical flow value is calculated by averaging all of the optical flow values that are greater than one or less than minus one. This is to minimize the effect of the background on the UAV motion vector calculated by optical flow values. Since background pixels show minimal movement between frames, pixels with small optical flow values are ignored. In Figure 2b, based on the optical flow values inside the ROI, one could recognize that the background pixels show almost no movement between frames. The accuracy of the algorithm is later evaluated in Section 4.2.

3.3. XROI-DCT

The proposed system sets different compression rates for ROI and non-ROI to increase the compression, thus increasing network transmission efficiency. The high-frequency deletion technique is applied to achieve extra compression on the non-ROI area to achieve such an effect. The high-frequency deletion technique produces additional compression by zeroing the tail values of the discrete cosine-transformed macroblocks. After applying a discrete cosine transform, the valuable information in the macroblock tends to be concentrated in the low-frequency section. By masking the high-frequency section, we can achieve additional compression by sacrificing the minimum amount of image quality. XROI-DCT is only applied to non-ROI regions, where information loss is affordable. By giving up the higher frequencies, higher compression can be achieved.

3.4. UAV Identification

One of the goals of the proposed system is to integrate the UAV identification module into the system. UAV identification is the task of determining the product model of the target UAV. Like other well-known classifiers, such as AlexNet [30], Inception [31], and SSD [32], authors have implemented a UAV identifier—a UAV model classification based on CNN—to recognize the type of threat that the system is tracking. The implemented CNN is trained with UAV image data collected from the web and manually labeled to learn and distinguish UAV models. Our CNN is trained to detect multi-rotors. We chose to train our CNN with multi-rotors because such UAVs are more difficult to distinguish by models—even with human eyes. The images used to train were 1835 images, applied with data augmentation of 16 different UAV models, and 390 images were used for testing. Although it is a relatively small set of images for training a CNN, due to limited classes of UAV models, the neural network was able to adapt well enough to identify UAV models.

4. Performance Evaluation

This section provides a brief description of the experiment setting with a thorough evaluation of the system conducted with the prototype of our proposed algorithm. The system is evaluated by encoding UAV videos from Kaggle and YouTube and videos taken by the authors with Samsung Galaxy S10 Ultra. To further evaluate the system performance, the encoded video is compared to the video produced by YOLOv7. The prototype of the system was implemented on a standard desktop machine running Ubuntu 20.04 LTS with Intel Xeon 16 processor CPU and 16GB RAM. The usage of GPU was restricted to mimic the limited resource environment. YOLOv7 was adopted for object detection, and OpenCV libraries were adopted for Optical Flow calculation and DCT operation. FFMPEG was used

for spatial/temporal prediction. Since YOLOv7 could not classify UAVs into models while being able to classify other objects (cars, trucks, desks, etc.) in the environment, the authors have implemented CNN for UAV identification with Tensorflow. The models that the CNN was trained for are listed in the following: 2DR Solo, DJI Phantom, DJI Inspire, DJI mavic, Chroma, Yuneec Typhoon, Yuneec Typhoon 4K, Xplorer, AscTec FireFly, Ehang 184, DJI Matrice, DJI S1000, XAG V40, Wingsland S6, MJX X916H, and Walkera QR Y100.

4.1. Experimental Configuration

The proposed system is evaluated by encoding a sequence of images retrieved from YouTube, Kaggle, and recorded flying UAVs. The number-labeled videos, a flying Solo2 in an urban setting, were taken by the authors. The videos all have different numbers of frames and different resolutions. Table 1 lists details of the videos encoded for the evaluation. The quality and overall file size of the video are compared to the video produced by YOLOv7.

Table 1. Video information.

Video Name	Resolution	Original File Size	No. of Frames
0	1920 × 1080	81.4 MB	1180
1	1920 × 1080	36.8 MB	654
2	1920 × 1080	30 MB	839
kaggle01	3840 × 2160	41.4 MB	1735
kaggle02	3840 × 2160	16.3 MB	665
Youtube	1280 × 720	18.2 MB	300

4.2. Accuracy of ROI Tracking

In this subsection, the accuracy of the optical flow-assisted ROI tracking technique is evaluated. The system can accurately set ROI by performing UAV detection on every frame. However, since UAV detection is a costly algorithm on limited resource devices, the proposed system adopts optical flow to track the displacement of the ROI. Figure 4 shows a visual representation that compares the ROI center tracked by using optical flow and UAV detection. The x-axis indicates the frame number of a UAV video, and the y-axis indicates the coordinates of the ROI center. The accuracy of the ROI tracking using optical flow is illustrated in the figure. If the x and y coordinates calculated using the ROI tracking scheme introduced in this paper are close to the coordinates retrieved by using the UAV detection algorithm, the proposed scheme is accurate. As depicted in the figure, the ROI tracked by using optical flow matches the ROI derived using UAV detection. Figure 4c shows huge spikes near $x = 0$ and $x = 120$. These are not outliers: the UAV briefly jerked, and the ROI responded abruptly. However, the target UAV never goes out of the ROI set. This is the same case for Figure 4d,f. Authors have noticed that the slower the UAV, the better it is for tracking. Prior research of [33,34] support our claim.

In Table 2, the average differences of the predicted midpoint are shown. The proposed algorithm predicts the location of the UAV and bounds it with the ROI box. The differences were measured in pixels by comparing its prediction to the prediction of YOLOv7 applied to every frame. Since the proposed system's technique is a dead reckoning process, the error tends to accumulate over time. The accumulated error is resolved by applying the periodic UAV detection of I-frames. Overall, the locations of the two ROIs matched, indicating that our proposed scheme was as accurate as YOLO.

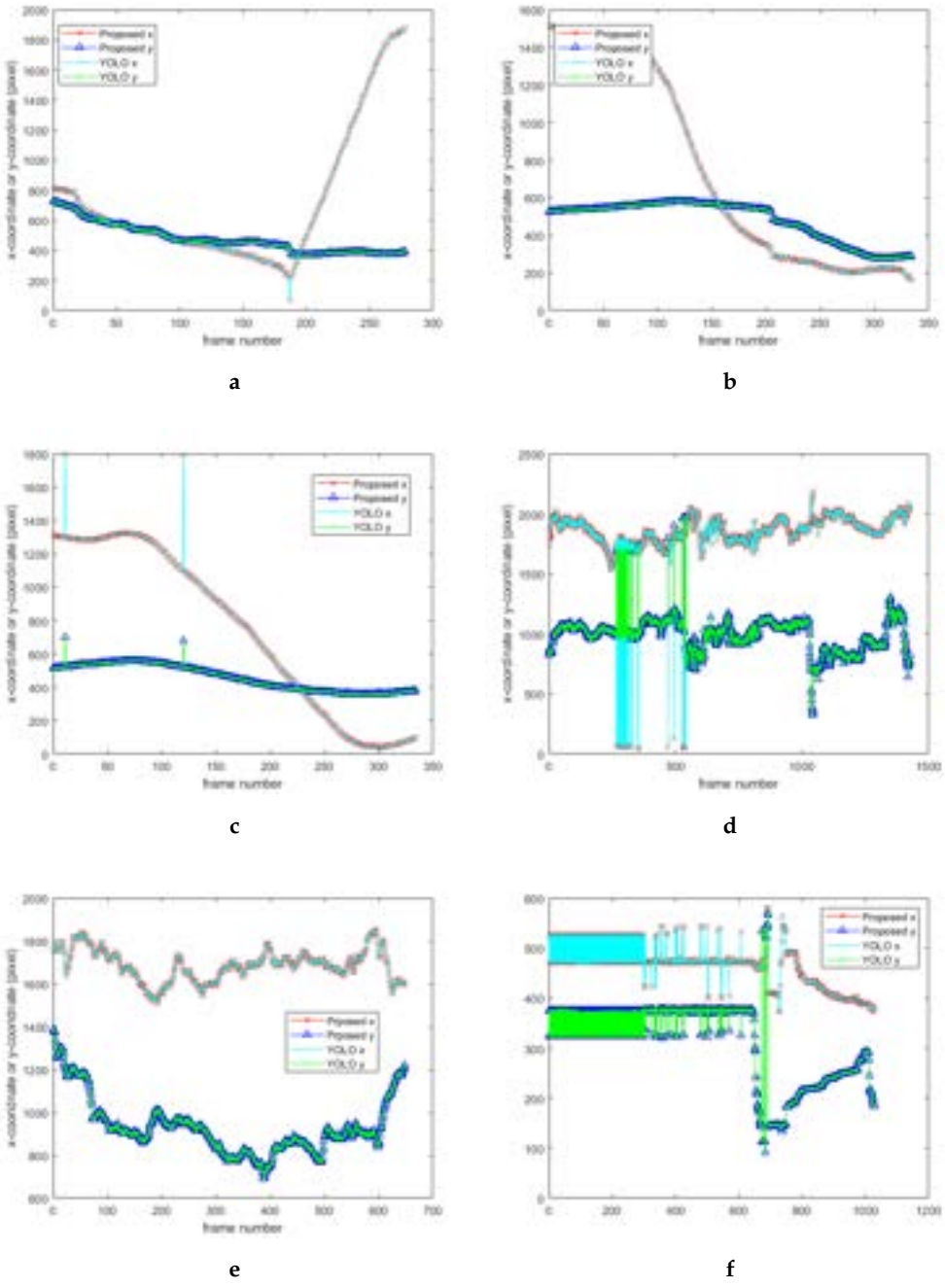

Figure 4. Accuracy of ROI management using optical flow. (**a**) Video 0. (**b**) Video 1. (**c**) Video 2. (**d**) Kaggle01. (**e**) Kaggle02. (**f**) Youtube.

Table 2. Pixel difference of ROI midpoint.

Video Name	*X*-Coordinates Avg. Difference (pixels)	*Y*-Coordinates Avg. Difference (pixels)
0	0	0
1	0.007393715	0.007393715
2	0.003726708	0.002484472
kaggle01	3.395935529	2.116327961
kaggle02	0.004622496	0.001540832
Youtube	0.389105058	0.892023346

4.3. Image Quality Analysis

In this section, the visual quality of the proposed system's output is evaluated by comparing the encoded videos using the proposed scheme to videos produced by YOLOv7. Figure 5 presents a visual representation of the encoded videos. The right images are frames from the video encoded with the proposed system, and the left images are frames from the video processed by YOLOv7. As shown in the figure, the image quality around the UAV is maintained at a high level, while the quality of other regions is degraded. The two different visual qualities within a single image depicted in the figure indicate that our proposed system can successfully encode with varying compression rates. The slight degradation of visual quality is due to the extra compression on non-ROI. The quality within ROI is maintained. Also, the dampness in image quality was only in the non-important area, and the image quality in the important area was maintained. Thus, we can conclude that the proposed system efficiently encodes ROI and non-ROI with two different compression rates.

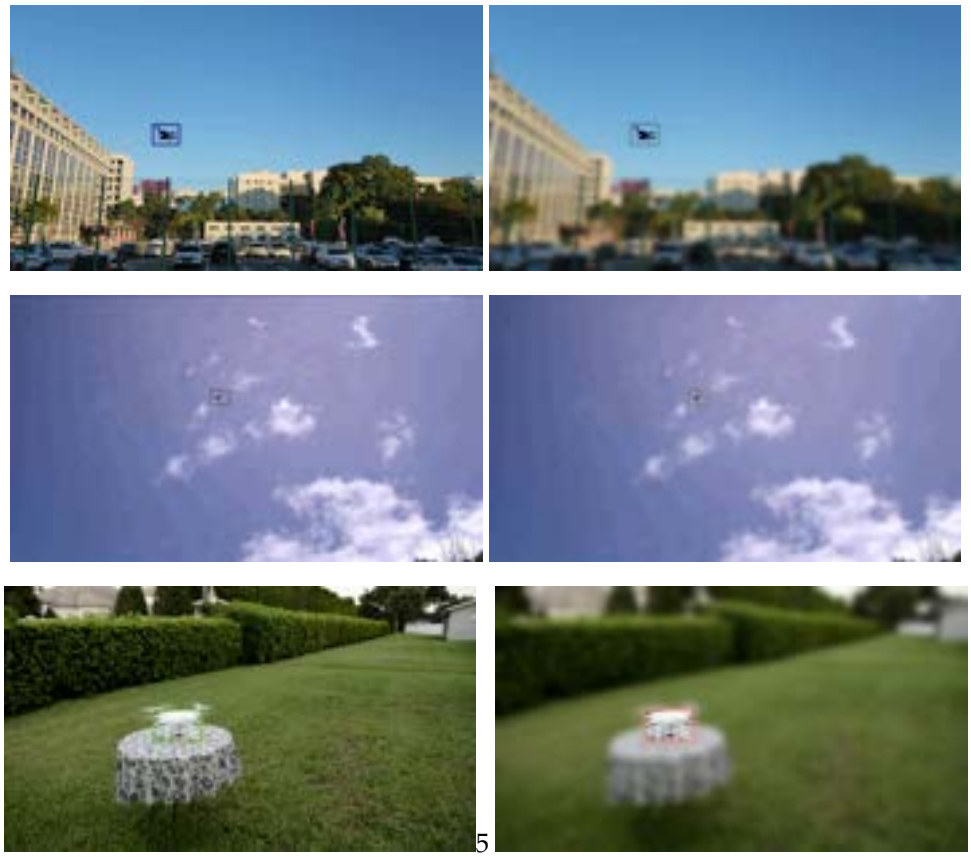

Figure 5. Comparison of videos (0, kaggle02, Youtube) processed with YOLOv7 (**left**) and the proposed system (**right**).

4.4. Volume Size Analysis

This subsection investigates the effect of the XROI-DCT on file size. Table 3 shows the video file size produced by the proposed system and YOLOv7. Detailed information regarding the file size of the encoded videos is addressed. By comparing the file size of the videos processed by the proposed system to the videos produced by YOLOv7, we observed that the file size of the video processed by the proposed algorithm is significantly smaller than the file size of the video processed by YOLOv7. In Table 3, the compression rate differs depending on the video. All codecs are somewhat optimized and do not support the same standards. All original videos used in the experiment were optimally encoded by some codec. Such codecs are optimized differently, have different basic settings, and encode video with different parameter sets. As evidenced, authors have found that ffmpeg—a well-

known open-source codec—and OpenCV give different size outputs when they encode the same video. The resulting videos produced by the proposed algorithm were encoded by using OpenCV video compression with the same setting and parameters of YOLOv7.

Videos with smaller ROI experienced smaller file sizes, and videos with larger ROI had larger file sizes. This is due to different encoding rates: the ROI area, where XROI-DCT is not applied, is encoded with a lower encoding rate so that it can be decoded without as much loss as possible, and the non-ROI area, where XROI-DCT is applied, is encoded with a higher encoding rate because it is efficient to have a compressed file size even if there could be some loss in a non-important area during decoding. By observing the encoded videos' file size, we were able to verify the effectiveness of the extra compression on non-ROI.

Table 3. File size of video processed by YOLOv7 and the proposed system.

Video Name	YOLOv7	Proposed Algorithm	Compression Rate
0	182.9 MB	34.7 MB	81.03%
1	85.2 MB	12.1 MB	85.8%
kaggle02	115.1 MB	52.7 MB	54.21%
Youtube	46.7 MB	29.4 MB	37.04%

4.5. Detection Speed

In this subsection, we evaluate how fast our proposed algorithm can detect a UAV on a video frame. As said above, our proposed algorithm adapts optical flow to predict the location of the UAV in the next frame so that it can track it. If the ROI is small and accurate enough, optical flow is expected to be faster than the trained neural network's inference speed. Evaluation results can be seen in Table 4. The inference speed of a neural network and optical flow calculation could vary depending on the hardware specification, image size, video codec, etc. Comparing raw latency will be inaccurate and meaningless. Therefore, we have compared two types of video—YOLOv7 applied to every frame and proposed algorithm—and presented them in a time-latency ratio. As shown in the table, the proposed algorithm adopting optical flow takes less time than applying YOLO to every frame. While Video1 takes about 64.86% of YOLOv7 process time, YouTube only takes about 24.05%. The processing time depends on the size of the ROI and, more importantly, the number of I-frames in a video. A large ROI accounts for more background pixels in the ROI and more feature points. Thus, more optical flow calculations were conducted on every frame except for I-frames. I-frames, where YOLOv7's detection is applied, take more time than non-I-frames, where optical flow is applied. As a note, there was one I-frame for every 30 frames. This shows that optical flow is faster than YOLOv7's inference.

Table 4. Detection speed improvement.

Video Name	Proposed Algorithm Compared to YOLOv7
0	315.86%
1	54.17%
kaggle02	86.98%
Youtube	201.49%

4.6. UAV Identification

The UAV identification of the proposed system is evaluated in this subsection. UAV identification of the proposed system is a UAV model classifier based on CNN. It was trained to classify 16 models, including Mavic Pro, Phantom, and Solo2—all models in the experiment videos. During evaluation, a small set of images, 1825 images for training and 390 images for testing, was used. To overcome the performance ceiling due to the small number of images, data augmentation was applied in various ways, such as rotations, left–right flips, saturation, brightness control, etc. Figure 6 depicts the confusion matrix [35]

of the UAV identifier in a heatmap. As shown on the scale to the right, the brighter the color of the tile, the more frequently the identifier shows respected results. The diagonal tiles on the heatmap represent cases where the prediction was correct, while other tiles represent cases where the prediction was incorrect. Diagonal values refer to the number of correct predictions during the testing phase. Therefore, the higher the diagonal values of the confusion matrix, the more accurate the UAV identifier is. As shown in Figure 6, most of the test trials are clustered around the diagonal tiles, meaning that the classifier performs desirably.

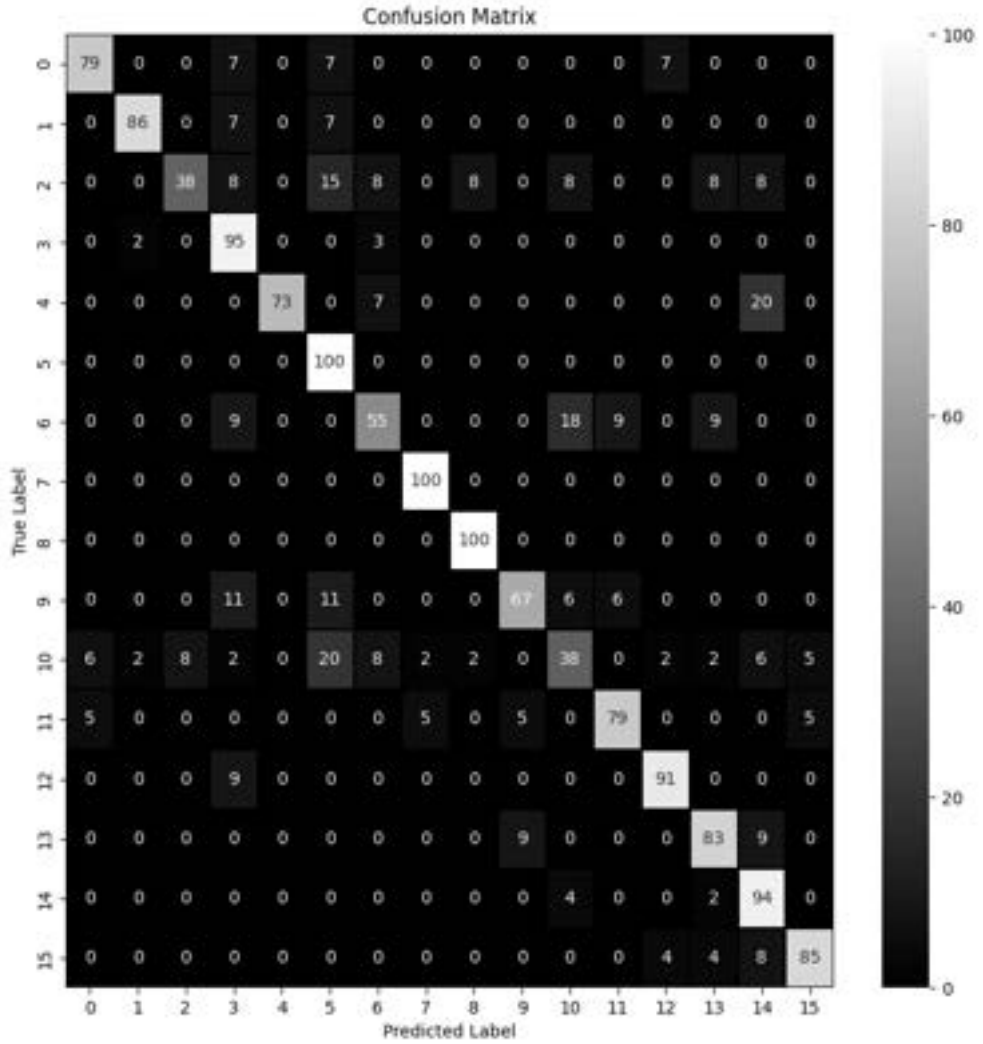

Figure 6. Confusion-matrix of the UAV identifier in a heatmap.

Figure 7 shows the Area Under the ROC Curve (AUC) [36], giving a graphical representation of the performance of the UAV identifier. The AUC of the ROC curve is a widely accepted metric for evaluating the performance of a classification model [37]. The ROC curve describes the performance of a binary classifier based on its discrimination threshold. On the AUC graph, the x-axis represents the true positive rate of the classifier, while the y-axis represents the false positive rate of the classifier. Since the UAV identifier is a 16-class classifier, merely applying outputs of the UAV identifier would not be able to be diagnosed. Therefore, it will be fed a scoring label representing the cases in which the identifier correctly predicted the class. Each ROC curve in the graph refers to the classifier's performance on one of the 16 classes. The ROC–AUC score—ranging from 0 to 1—was 0.9869. A score of 0.5 is the equivalence of random guessing, and a score of 1 indicates that the model is an omniscient classifier. The dash line in the graph represents the score of 0.5. This indicates that the UAV identifier is performing near to its best.

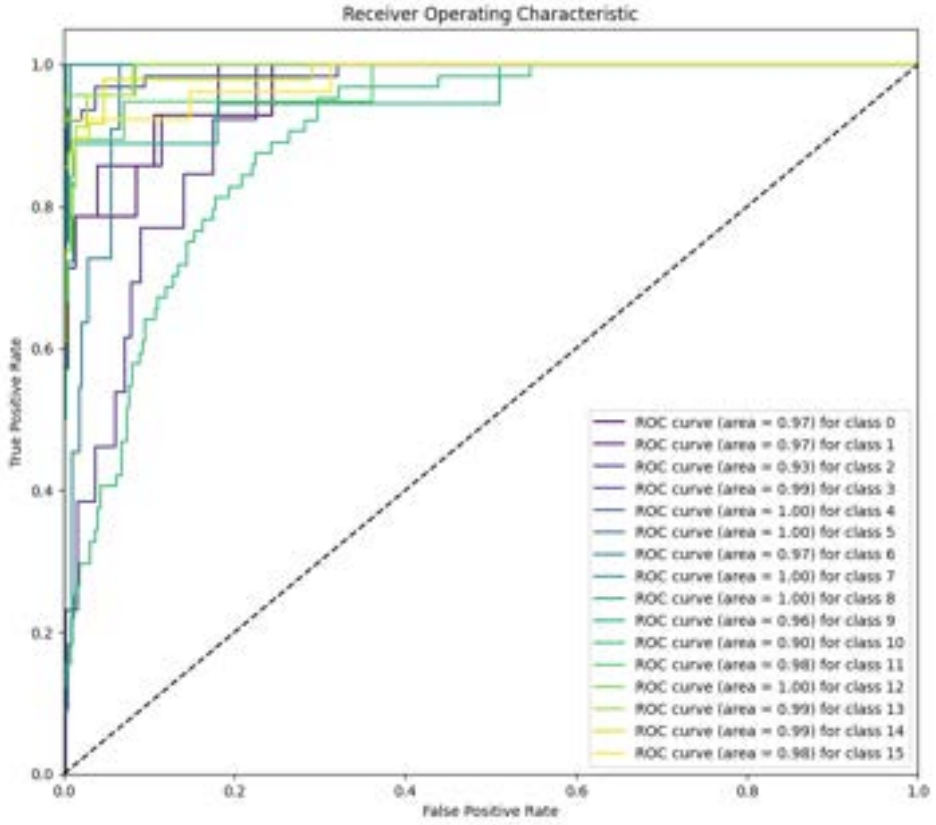

Figure 7. AUC of ROC.

5. Conclusions

In this paper, an accurate UAV tracking/surveillance system with an efficient encoding method was designed. The proposed system uses an ROI-based approach to apply two different compression rates on a single video. On every frame of the input video, the ROI region and the non-ROI region are determined using UAV detection, and a lower compression rate is applied to the ROI region while a higher compression rate is applied to the non-ROI region. This way, non-ROI regions, holding relatively less important information, can be compressed without the concern of decoding. The movement of ROI is managed using optical flow. Efficient transmission is achieved by applying XROI-DCT for effective compression. The evaluation, composed of the implemented prototype, shows that the proposed system can successfully encode with two different compression rates. The evaluation shows that the videos encoded with the proposed system experienced an average of 70% reduced file size without hindering the UAV region. The reduced file size of the encoded video shows that it requires less bandwidth, and the fact that ROI is located on the bounding box of the leading object detector, YOLOv7, most of the time shows that detection is applied accurately.

For our future work, we plan to apply a rate decision to our system. By applying a rate decision system, we can adapt the rate of a whole UAV tracking stream system. We plan to explore the proposed system's capabilities using infrared and thermal cameras, assuming the UAV is visible in these imaging modalities. We aim to enhance the UAV identification capacity beyond the current 16 classes to detect various UAVs, further improving its robustness and versatility in identifying various UAV models. Additionally, we would like to lessen the computation cost of the encoding phase so that it is suitable for IoT devices.

Author Contributions: Conceptualization, D.'R.L., W.S. and H.K.; Methodology, D.'R.L., S.K., N.Y. and H.K.; Software, D.'R.L., S.K. and N.Y.; Validation, D.'R.L. and N.Y.; Formal analysis, D.'R.L.; Investigation, D.'R.L., S.K., N.Y., W.S. and H.K.; Resources, W.S. and H.K.; Data curation, D.'R.L., S.K., N.Y. and W.S.; Writing—original draft, D.'R.L. and H.K.; Writing—review & editing, D.'R.L.,

S.K., N.Y., W.S. and H.K.; Visualization, D.'R.L. and S.K.; Supervision, H.K.; Project administration, H.K.; Funding acquisition, H.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea funded by the Korean Government (grant 2020R1A2C1012389). This work was supported by Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Korea government (MOTIE) (No. 20224B10300090).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: [https://www.kaggle.com/datasets/nelyg8002000/4k-eo-uav-videos?resource=download] (accessed on 16 May 2024) and [https://www.youtube.com/watch?v=Ah3Up2AbJZM] (accessed on 16 May 2024).

Conflicts of Interest: Author Wonki Seo was employed by the Nextcore Technology Co., Ltd. The remaining author declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Abro, G.E.M.; Zulkifli, S.A.B.M.; Masood, R.J.; Asirvadam, V.S.; Laouti, A. Comprehensive Review of UAV Detection, Security, and Communication Advancements to Prevent Threats. *Drones* **2022**, *6*, 284. [CrossRef]
2. Statista. Commercial UAVs—Statistics & Facts. 2023. Available online: https://www.statista.com/topics/3601/commercial-uavs/ (accessed on 11 March 2024).
3. Federal Aviation Administration. UAV Regulations and Policies. 2024. Available online: https://www.faa.gov/node/54496 (accessed on 16 May 2023).
4. Dronesec. Dronesec Reports 60% Increase in Drone Incidents in 2022 Compared with 2021. 2022. Available online: https://www.unmannedairspace.info/latest-news-and-information/dronesec-reports-60-increase-in-drone-incidents-in-2022-compared-with-2021/ (accessed on 16 May 2023).
5. Gupta, A.K.; Seal, A.; Prasad, M.; Khanna, P. Salient object detection techniques in computer vision—A survey. *Entropy* **2020**, *22*, 1174. [CrossRef] [PubMed]
6. Yoo, T.; Lee, S.; Yoo, K.; Kim, H. Reinforcement Learning Based Topology Control for UAV Networks. *Sensors* **2023**, *23*, 921. [CrossRef] [PubMed]
7. Park, C.; Lee, S.; Joo, H.; Kim, H. Empowering adaptive geolocation-based routing for UAV networks with reinforcement learning. *Drones* **2023**, *7*, 387. [CrossRef]
8. Zitar, R.A.; Mohsen, A.; Seghrouchni, A.E.; Barbaresco, F.; Al-Dmour, N.A. Intensive review of drones detection and tracking: Linear Kalman filter versus nonlinear regression, an analysis case. *Arch. Comput. Methods Eng.* **2023**, *30*, 2811–2830. [CrossRef]
9. O'shea, K.; Nash, R. An introduction to convolutional neural networks. *arXiv* **2015**, arXiv:1511.08458.
10. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
11. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
12. Barron, J.L.; Fleet, D.J.; Beauchemin, S.S. Performance of optical flow techniques. *Int. J. Comput. Vis.* **1994**, *12*, 43–77. [CrossRef]
13. Ullah, M.B. CPU based YOLO: A real time object detection algorithm. In Proceedings of the 2020 IEEE Region 10 Symposium (TENSYMP), Dhaka, Bangladesh, 5–7 June 2020; pp. 552–555.
14. Kim, Y.; Choi, J.S.; Kim, M. A real-time convolutional neural network for super-resolution on FPGA with applications to 4K UHD 60 fps video services. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *29*, 2521–2534. [CrossRef]
15. Lab, W. A Multi-Stage Approach to UAV Tracking using ROI Management and Rate-Adaptive Video Coding. 2024. Available online: https://youtu.be/gFQwzg6yLTs (accessed on 15 May 2023).
16. Chen, W.; Huang, H.; Peng, S.; Zhou, C.; Zhang, C. YOLO-face: A real-time face detector. *Vis. Comput.* **2021**, *37*, 805–813. [CrossRef]
17. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
18. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2015; Volume 28.
19. Li, X.; Wei, Y.; Li, J.; Duan, W.; Zhang, X.; Huang, Y. Improved YOLOv7 Algorithm for Small Object Detection in Unmanned Aerial Vehicle Image Scenarios. *Appl. Sci.* **2024**, *14*, 1664. [CrossRef]

20. Liu, Y.; Li, Z.; Soh, Y.; Loke, M. Conversational Video Communication of H.264/AVC with Region-of-Interest Concern. In Proceedings of the 2006 IEEE International Conference on Image Processing, Atlanta, GA, USA, 8–11 October 2006; pp. 3129–3132. [CrossRef]
21. Arachchi, H.; Fernando, W.; Panchadcharam, S.; Weerakkody, W. Unequal Error Protection Technique for ROI Based H.264 Video Coding. In Proceedings of the Canadian Conference on Electrical and Computer Engineering, CCECE '06, Ottawa, ON, Canada, 7–10 May 2006; pp. 2033–2036. [CrossRef]
22. Zheng, Y.; Feng, J.; Ma, H.; Chen, Y. H. 264 ROI coding based on visual perception. In Proceedings of the 5th International Conference on Visual Information Engineering (VIE 2008), Xi'an, China, 29 July–1 August 2008; pp. 829–834. [CrossRef]
23. Mukherjee, D.; Bankoski, J.; Grange, A.; Han, J.; Koleszar, J.; Wilkins, P.; Xu, Y.; Bultje, R. The latest open-source video codec VP9-an overview and preliminary results. In Proceedings of the 2013 Picture Coding Symposium (PCS), San Jose, CA, USA, 8–11 December 2013; pp. 390–393.
24. Lee, S.K.; Ryoo, J.; Yoo, S.; Jung, J.; Lee, W.; Kim, H. CoSA: Adaptive link-aware real-time streaming for mobile devices. In Proceedings of the IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Lyon, France, 7–9 October 2013; pp. 681–686. [CrossRef]
25. Lee, S.K.; Yoo, S.; Jung, J.; Kim, H.; Ryoo, J. Link-Aware Reconfigurable Point-to-Point Video Streaming for Mobile Devices. *ACM Trans. Multimed. Comput. Commun. Appl.* **2015**, *12*, 9. [CrossRef]
26. Huang, Y.; Rao, R.P. Predictive coding. *Wiley Interdiscip. Rev. Cogn. Sci.* **2011**, *2*, 580–593. [CrossRef] [PubMed]
27. Ahmed, N.; Natarajan, T.; Rao, K.R. Discrete cosine transform. *IEEE Trans. Comput.* **1974**, *100*, 90–93. [CrossRef]
28. Jiang, P.; Ergu, D.; Liu, F.; Cai, Y.; Ma, B. A Review of Yolo algorithm developments. *Procedia Comput. Sci.* **2022**, *199*, 1066–1073. [CrossRef]
29. Diwan, T.; Anirudh, G.; Tembhurne, J.V. Object detection using YOLO: Challenges, architectural successors, datasets and applications. *Multimed. Tools Appl.* **2023**, *82*, 9243–9275. [CrossRef] [PubMed]
30. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
31. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
32. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
33. Zhai, X.; Huang, Z.; Li, T.; Liu, H.; Wang, S. YOLO-Drone: An Optimized YOLOv8 Network for Tiny UAV Object Detection. *Electronics* **2023**, *12*, 3664. [CrossRef]
34. Chen, C.; Zheng, Z.; Xu, T.; Guo, S.; Feng, S.; Yao, W.; Lan, Y. Yolo-based uav technology: A review of the research and its applications. *Drones* **2023**, *7*, 190. [CrossRef]
35. Beauxis-Aussalet, E.; Hardman, L. Visualization of confusion matrix for non-expert users. In Proceedings of the IEEE Conference on Visual Analytics Science and Technology (VAST)-Poster Proceedings, Paris, France, 25–31 October 2014; pp. 1–2.
36. Narkhede, S. Understanding auc-roc curve. *Towards Data Sci.* **2018**, *26*, 220–227.
37. Fan, J.; Upadhye, S.; Worster, A. Understanding receiver operating characteristic (ROC) curves. *Can. J. Emerg. Med.* **2006**, *8*, 19–20. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.