electronics

Article

Sequential Learning of Flame Objects Sorted by Size for Early Fire Detection in Surveillance Videos

Widia A. Samosir [1], Duy B. Nguyen [2] and Seong G. Kong [1,*]

[1] Department of Computer Engineering, Sejong University, Seoul 05006, Republic of Korea; 23110148@sju.ac.kr
[2] Pintel Co., Yongin 16978, Republic of Korea; duy@pintel.co.kr
* Correspondence: skong@sejong.edu

Abstract: This paper presents a sequential learning method aimed at improving the performance of a lightweight deep learning model used for detecting fires at an early stage in surveillance video streams. The proposed approach involves a sequence of supervised learning steps, wherein the entire training dataset is partitioned into multiple sub-datasets based on the size of fire objects. The size of fire objects is measured by object size ratio, which is the ratio of the bounding box area of the detected fire flame object relative to the entire image area. The initial training sub-dataset contains the largest-sized fire objects, progressing to the final sub-dataset containing the smallest-sized fire objects. The objective is to employ sequential learning to enhance the detection of small-sized fire objects relative to the image area using a lightweight model suitable for edge computing devices. Experiment results demonstrate that a deep learning fire detection model trained sequentially with a descending order of object size can effectively detect small flame objects with an object size ratio less than 0.006, achieving an F1 score of 93.1%, representing a 27% improvement compared to traditional supervised learning with no sequential learning steps. Additionally, performance in detecting tiny flame objects with an object size ratio less than 0.0016 achieves an F1 score of 94.5%, showing a 17.5% increase compared to the baseline without sequential learning.

Keywords: sequential learning; early fire detection; small object detection; lightweight deep learning models; surveillance video; deep learning

check for updates

Citation: Samosir, W.A.; Nguyen, D.B.; Kong, S.G. Sequential Learning of Flame Objects Sorted by Size for Early Fire Detection in Surveillance Videos. *Electronics* **2024**, *13*, 2232. https://doi.org/10.3390/electronics13122232

Academic Editor: Beiwen Li

Received: 23 April 2024
Revised: 27 May 2024
Accepted: 6 June 2024
Published: 7 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

Fires in their early stages tend to be relatively small in size [1], posing a challenge for early fire detection in fire monitoring since its inception. Traditional fire monitoring methods suffer from several disadvantages, including inefficiency, high cost, labor intensiveness, and susceptibility to human error. Traditional methods of fire detection have relied heavily on sensor-based technologies, such as smoke detectors and heat sensors [2]. While IoT-based methods enhance these systems by leveraging interconnected sensors, smart cameras, and lightweight models implemented in embedded systems [3,4], they often face challenges such as high false positive rates and difficulties in detecting small-sized fires [5].

Recent advancements in deep learning technologies offer a promising solution to address these drawbacks. The utilization of deep learning models for fire detection from surveillance video cameras has garnered considerable attention in recent years. Deep learning algorithms have significantly improved the accuracy of fire detection by leveraging extensive annotated data, leading to the development of state-of-the-art techniques. For instance, a dilated convolutional neural network (CNN) [6] automatically extracts features for training the fire detection model. This approach enhances object detection speed by employing fewer layers, allowing the model to learn features of both large and small, complex objects.

Accurately identifying fires at an early stage is important as it can lead to improvement in efficiency and reductions in labor costs for fire mitigation and monitoring. Early

intervention will give firefighters and emergency responders a chance to extinguish the fire before it spreads and causes more damage. One of several challenges in developing efficient unmanned fire monitoring systems is designing a lightweight model that can be implemented in edge computing environments while providing high performance despite its small size. With these considerations in mind, we propose a detection method capable of identifying early-stage fires and detecting fire objects of all sizes.

In this paper, we present sequential learning of a detection model using fire objects sorted by size to detect early fires in surveillance videos. The key contributions are as follows: (1) Propose a sequential learning framework for training a deep learning model on a sequence of sub-datasets created by partitioning the entire training dataset, sorted in terms of object sizes; (2) Present a method to mitigate false alarms by reducing false positives over several learning cycles, targeting the detection of small fire flame objects in the early stages to facilitate faster mitigation of fire events; (3) Introduce an empirical formula for determining the number of sequential learning cycles to detect desired small objects.

2. Related Work

Early fire detection is crucial to prevent catastrophic losses and provide early mitigation of fire accidents. Various methods based on machine learning models and surveillance video analysis have been proposed to detect fires in their initial stages. One approach involves using satellites imaging and neural network models such as InceptionV3, VGG-16, ResNet-50, and DenseNet for forest fire detection [7]. Another method focuses on utilizing UAVs and the YOLO model to detect wildfires early [8]. Currently, transformer-based models are also commonly used in detecting small objects by leveraging the Vision Transformer (ViT) architecture [9,10].

Due to the latency involved in transferring large volumes of data such as images and videos through communication layers, many researchers are attempting to create lightweight models suitable for deployment on edge computing devices. Zhao et al. [11] proposed the improved Fire-YOLO algorithm to enhance the detection of small targets in fires and reduce the model size. Xu et al. [12] introduced the lightweight fire detection model called Light-YOLOv5, which increases the mean average precision (mAP) by 3.3% compared to the original YOLOv5 network. Tsalera et al. [13] enhanced the robustness and flexibility of the model using transfer learning, cross-dataset evaluation, and noise inclusion. Wang et al. [14] proposed the YOLOv5s architecture, which incorporates the mix-up data enhancement strategy to address blurred target boundaries and adds a feature pyramid module to enhance image feature extraction. This model, referred to as Feature-Enhanced YOLO (FE-YOLO), achieved a mAP result of approximately 72.53%, representing an improvement of around 3.42% compared to the YOLOv5s network. Another novel lightweight model for real-time fire detection was also introduced by Almeida et al. [15], utilizing a lightweight CNN called EdgeFireSmoke. This model attained an accuracy of around 98.97% and an F1 score of 95.77%. Xu et al. [16] present another method for lightweight object detection using High-Resolution Network (HRNet) as a backbone and propose a scale-aware squeeze-and-excitation (SASE) module that fully explores feature interactions without increasing network complexity. This model achieved a 3.7% improvement over Lightweight High-Resolution Network (Lite-HRNet). Xie et al. [17] introduced M-YOLOX, which utilizes MobileNetV3 with SPPBottleneck layer and Convolutional Block Attention Module (CBAM). This lightweight model achieved a mAP of around 88.99% in the fire class. Li et al. [18] also proposed using the same base network and introduced a transfer learning method applied to MobileNetV3. This method achieved an accuracy of around 92.1% through experiments involving collection of tunnel fire samples from the internet.

Sequential learning in object detection refers to the ability to continually learn new tasks without forgetting previously learned knowledge. In related literature, several approaches addressing similar methods as sequential learning have been explored, such as incremental learning [19–21], continuous learning [22–24], and application transfer learning [25,26]. Nenakhov et al. [19] incrementally introduce new object classes to the model

using a method referred to as random memory. These object classes are from the CORe50 dataset and include items such as scissors, plug adapters, mobile phones, and light bulbs. With each iteration, the number of instances stored in the random memory is systematically reduced, facilitating the model's adaptation to new data while managing computational resources effectively. Hasan et al. [22] employed a training methodology wherein the training set was divided into four batches and sequentially fed into the learning framework. Initially, the model exhibited misclassifications or low probability scores for activity classes. However, over sequential iterations, the model displayed continuous improvement, achieving higher probability scores and enhanced accuracy in activity recognition by a margin of 2–3% compared to other approaches in human activity recognition, as demonstrated across several human activity datasets in the state-of-the-art literature. These methods typically involve the addition of new classes in several iterations, allowing the model to adapt to evolving data.

In the proposed method, instead of training with the whole dataset in one cycle, we repeatedly train a detection model over multiple learning cycles using training sub-datasets sorted by object size to improve performance in detecting small fire flame objects in the scene. We enhance accuracy by mitigating false positives through sequential learning, which incrementally increases accuracy by incorporating false positive images into the training process in each cycle of sequential learning.

3. Sequential Learning to Enhance Performance in Small Object Detection

3.1. The Sequential Learning Process Pipeline

The proposed sequential learning involves iteratively training a deep learning model for fire detection across multiple training cycles. Given a training dataset, we partition it into multiple sub-datasets, each containing an approximately equal number of images arranged by the size of flame objects within the image. The initial sub-dataset consists of image samples containing the largest flame objects, followed by subsequent sub-datasets containing progressively smaller objects, until the smallest objects are encompassed in the final sub-dataset. During each training cycle of sequential learning, the fire detection model is trained on one sub-dataset, and its performance is sequentially evaluated on the validation dataset from the first to the last sub-dataset. Images detected as "false positive" by the model are incorporated into the subsequent sub-dataset to compose a training sub-dataset for training the subsequent cycle.

Figure 1 provides an overall diagram of the proposed sequential learning scheme for fire detection from a surveillance video stream. If we divide a training dataset into N sub-datasets, the proposed method will train the model N times. The first sub-dataset, DB_1, consists of image samples containing the largest flame objects. We then train our base model, FM_0, with DB_1 to build an updated fire detection model, FM_1. The model FM_1 is tested on the validation dataset to obtain true positives (TPs) as well as false positives (FPs). Only FP images are added to the second sub-dataset to form an updated sub-dataset, DB_2, for the second training cycle. We repeat this process for all the remaining sub-datasets in N cycles to complete the sequential learning. FM_k denotes the fire detection model trained with sub-dataset DB_k in training cycle k. The model FM_N, trained with DB_N in the last cycle, is used as the final fire detection model for real-time fire detection to monitor a surveillance video stream.

During learning cycle k, the performance of the fire detection model FM_k is evaluated on the validation dataset. False positive images are collected and added to the current sub-dataset DB_k to create the sub-dataset DB_{k+1} for the next sequential learning cycle ($k + 1$). The final fire detection model trained using this sequential learning approach can detect small flame objects with reduced false positives from a surveillance video stream, enabling early fire detection.

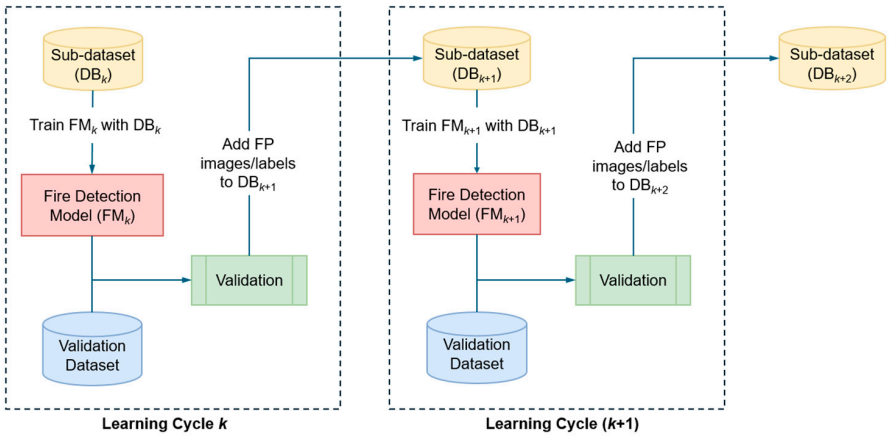


Figure 1. Visual description of sequential learning process: At learning cycle k, the fire detection model FM_k trained with sub-dataset DB_k is tested on the validation dataset. The resulting FP images by FM_k are added to DB_{k+1} for the next learning cycle.

Boosting techniques [27] attempt to improve the performance of a learning algorithm by combining multiple weak learners into a strong learner. The objective is to iteratively improve the overall model's performance by focusing on difficult-to-classify instances. In contrast, our sequential learning approach trains the model multiple times using multiple sub-datasets sorted by object size to improve performance in detecting smaller objects.

3.2. Definitions of Small Fire Flame Objects

There are several ways to define small objects. Chen et al. [28] use exact pixels as the threshold for defining small areas. In this paper, we measure the size of an object by object size ratio, which is the ratio of the bounding box area of the detected fire flame object relative to the size of the entire image area. We define the object size ratio (denoted A) as the ratio of the size of the bounding box of a fire flame object and the size of the entire image:

$$A = \frac{O_w O_h}{I_w I_h} \tag{1}$$

where O_h, O_w denote the height and width of the bounding box of the detected fire flame object, and I_h, I_w represent the height and width of the entire image. We sorted small-sized fire objects into three distinct categories: small, petite, and tiny, according to the object size ratios of $A < 0.016$, $A < 0.006$, and $A < 0.0016$, respectively. Since we conducted the experiments on the three categories separately, an object labeled as petite also falls under the category of small objects, but not of tiny objects. Consequently, when detecting tiny objects, petite objects will be excluded from consideration. Figure 2 shows a visual comparison of the three small-sized object definitions.

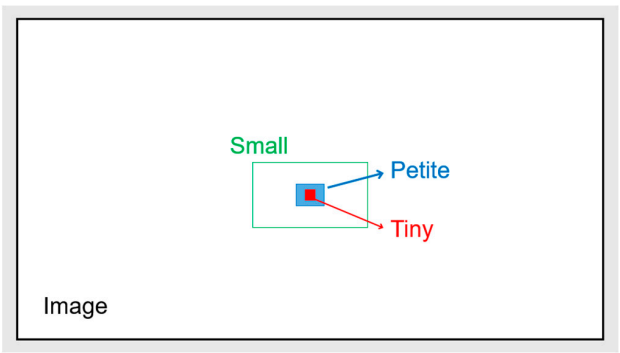


Figure 2. A visual comparison of the three definitions of small-sized objects: small (<0.016), petite (<0.006), and tiny (<0.0016).

3.3. Determining Number of Sequential Learning Cycles

We determine the number of sequential learning cycles (denoted N) based on the number of images with their corresponding object size ratios being less than a threshold. We set the size of the targeted small fire object as a threshold. For a desired size of small objects (denoted $A_{desired}$) to detect, the number of images with object size ratios below the threshold (denoted K_{less}) is given by:

$$K_{less} = \sum_{i=1}^{K} I(A_i < A_{desired}) \tag{2}$$

$$I(A_i < A_{desired}) = \begin{cases} 1 & if\ A_i < A_{desired} \\ 0 & if\ A_i \geq A_{desired} \end{cases} \tag{3}$$

where I denotes an indicator function, and A_i represents the object size ratio of the largest instance in image I_i. K_{less} counts the number of images whose object size ratios are below the threshold. Then, the number of sequential learning cycles is given by the following empirical formula:

$$N = \left\lceil \frac{K}{K_{less}} \right\rceil - 1 \tag{4}$$

where $\lceil x \rceil$ denotes the ceiling operator, which returns the smallest integer not less than the given value x. This empirical formula to determine the number of cycles (N) was derived from several experiments and observations by iterating through different values of the desired small area and observing the performance trend.

3.4. Partitioning a Dataset into Multiple Sub-Datasets Sorted by Object Size

We sort all images in a training dataset according to the object size ratio. For images containing multiple fire objects, we use the largest bounding box to determine the size of the flame object in the image. Given a set of K images in the training dataset, $D = \{I_{A_1}, I_{A_2}, \ldots, I_{A_K}\}$, where A_i represents the object size ratio of the largest instance in image I_i. We sort images in dataset D by their object size ratios in descending order to obtain dataset S, referred to as the sorted-by-size dataset, $S = \left\{I_{A_{largest}}, I_{A_{largest-1}}, \ldots, I_{A_{smallest}}\right\}$. After sorting all images in the training dataset by the size of largest instance of fire objects within the image, we divide the dataset S equally into N sub-datasets $\{S_1, S_2, \ldots, S_N\}$ of equal number of images.

3.5. Lightweight Fire Detection Model

The YOLO algorithm is based on a convolutional neural network for real-time object detection [29]. For fire detection, YOLO can be trained to recognize flames and smoke as distinct objects, enabling early detection of fires in surveillance video streams. In this paper, we employ YOLOv5n as our base lightweight model for early fire detection from a surveillance video stream. Given that our proposed fire detection model is intended to run on an embedded hardware board, we opt for the lightest network model of YOLO, namely YOLOv5n, from among several variants. The YOLO network structure comprises input, backbone, feature fusion, and prediction modules. YOLOv5n uses CSPDarknet with three convolutional layers in the backbone, as well as cross-stage partial networks, for feature extraction from an image, following the approach in [30,31]. YOLOv5n employs both binary cross-entropy and logistic loss functions in the loss function computation, resulting in higher accuracy and contributing to improved efficiency of the model in detection due to a reduced number of parameters in the neck network compared to YOLOv5s, a lightweight model version of YOLOv5 [32].

3.6. Model Evaluation

In this paper, we evaluate the model using the F1 score, chosen for its balance between precision and recall. While precision, recall, and mAP are also common metrics in object

detection, we focus solely on the F1 score due to its effectiveness in scenarios where false positives and false negatives are critical, such as fire and smoke detection. Precision measures the accuracy of positive class predictions, representing the proportion of correct positive predictions. Conversely, recall gauges the classifier's ability to detect positive instances by identifying the proportion of positive instances correctly identified. The F1 score, a composite metric combining precision and recall, offers a summary statistic for evaluating classifier performance.

The model performance was assessed using an IoU (Intersection over Union) threshold of 0.5 to measure precision and recall. The overlap between actual bounding boxes and predicted bounding boxes is evaluated using IoU. A higher value of IoU indicates a greater overlap between predicted and ground truth, demonstrating the model's ability to accurately detect or predict the object. A successful detection of an object is determined by the value of IoU for fire/flame and smoke exceeding the 0.5 threshold. To ensure the reliability of our findings, each experiment was run 10 times, and the results presented are the averages of these runs. This approach helps to mitigate the effect of random variability and provides a more robust measure of the model's performance.

4. Experiment Results and Discussion

4.1. Dataset

We collected fire images from three sources: the Internet, the AI Hub dataset [33], and the IEEE dataset [34]. Internet images were scraped from websites such as Kaggle, Google, and YouTube. The collected images varied in size and were of relatively low resolution. Both low-resolution and high-resolution images were necessary since the model needed to learn to detect fires in different environments. Low-resolution images were obtained from various conditions, such as fires inside and outside houses, in the forest, and during the day and night. High-resolution images contained small fire flames and smoke objects in surveillance videos. The AI Hub dataset was downloaded from the fire prediction video dataset on the AI Hub website. The dataset's original form was extracted from videos from the early stage of fire recorded by CCTV with a resolution of 1920 × 1080 pixels. The IEEE dataset contains wildfire images captured by unmanned aerial vehicles (UAVs) with a resolution of 1280 × 720 pixels. The videos in the IEEE dataset depict the behavior of a fire event in the middle of a forest. We initially used an annotation tool to label the images scraped from the Internet and AI Hub, followed by manual fine-tuning to ensure the bounding boxes of the fire objects were as tight as possible. Each image was carefully examined, and the bounding boxes were drawn around the fire object. Figure 3 shows image samples collected from the three sources.



(**a**) (**b**) (**c**)

Figure 3. Sample images in the dataset from three sources: (**a**) Internet; (**b**) AI Hub; (**c**) IEEE.

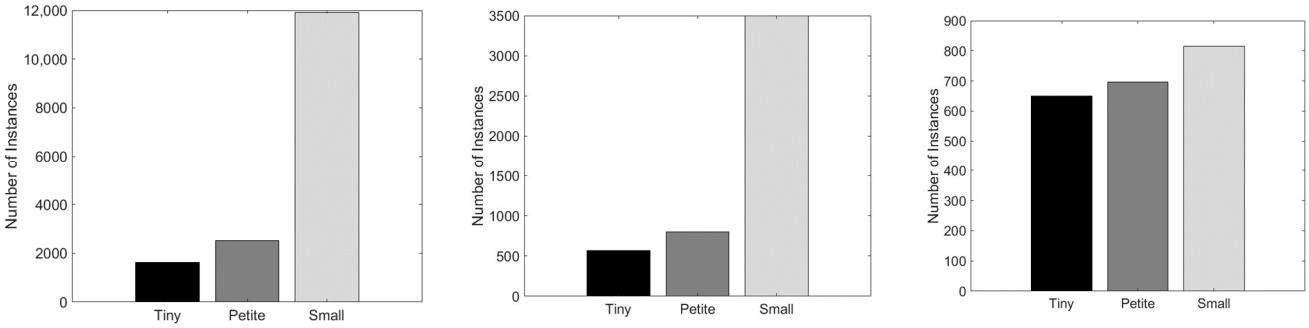
According to MS COCO's scale division for object detection [35], a small object is defined as data with a bounding box area size below 32 × 32 pixels, a medium object with a bounding box area size ranging from 32 × 32 pixels to 96 × 96 pixels, and a large object with a bounding box area size greater than 96 × 96 pixels. However, instead of relying on fixed pixel values proposed by MS COCO to set small-sized objects, we employ a more adaptable approach. An object is considered as a small object by its size relative to the image size, ensuring consistency across different resolutions. Figure 4 shows sample images with three types of small-sized fire flame objects.



(**a**) (**b**) (**c**)

Figure 4. Sample images with three types of small-sized flame objects from the AI Hub dataset: (**a**) small ($A = 0.012 < 0.016$); (**b**) petite ($A = 0.003 < 0.006$); (**c**) tiny ($A = 0.0007 < 0.0016$).

Figure 5 consists of three subplots, each representing the proportion of small-sized objects in each of the training, validation, and test datasets that were used in experiments. If small objects are underrepresented in the training data, the model may struggle to learn their features effectively. However, in the subsequent section, we aim to demonstrate the optimization of these instances of small objects through sequential learning.



(**a**) (**b**) (**c**)

Figure 5. Distribution of the number of small-sized objects in each of the training, validation, and test datasets used in the experiments. (**a**) Training, (**b**) validation, (**c**) test.

4.2. Number of Sequential Learning Cycles

We determine the number of sequential learning cycles using the process outlined in Section 3.3, utilizing the threshold $A_{desired}$ set to 0.016, 0.006, and 0.0016. The total number of images (K) in the training dataset is 17,812, and the values of K_{less} were 5889, 3372, and 1748 for the three types of small objects: small, petite, and tiny. Using Equation (3), we determined the number of sequential learning cycles (N) needed to successfully detect small objects as 3, 5, and 10 for each of the small, petite, and tiny objects, respectively. We validate the calculated number of sequential learning cycles for small, petite, and tiny objects using the validation image dataset. Table 1 displays F1 scores for the five chosen sequential learning cycles, 2, 3, 5, 7, and 10, for each of three values of $A_{desired}$ for the three small object categories. The numbers of sequential learning cycles corresponding to the highest F1 scores match the calculated number of sequential learning cycles, validating the formula in Equation (4) for determining the required number of sequential learning cycles.

Table 1. F1 scores for the five chosen sequential learning cycles, 2, 3, 5, 7, and 10, for each of three values of $A_{desired}$ for the three small object categories using the validation dataset.

$A_{desired}$	F1 Score (%)				
	$N = 2$	$N = 3$	$N = 5$	$N = 7$	$N = 10$
0.016 (Small)	83.4	**90.8**	85.5	86.1	85.1
0.006 (Petite)	84.1	90.3	**93.1**	86.2	85.8
0.0016 (Tiny)	76.6	52.2	75.1	79.6	**94.5**

Note: The numbers in bold indicate the best performance for each desired object size.

4.3. Training

The model is trained sequentially using the sub-datasets in the order of object size determined by the data selection process. The final model is then used to detect fires in a surveillance video stream. All experiments were conducted on a machine equipped with Intel Xeon E5-2620 processor, 128GB of RAM, and an NVIDIA TITAN XP GPU with 12GB of memory. We implemented the process using Python version 3.8.17 code on Windows 10. During the training phase, we set parameters for 100 epochs with a batch size of 16 for YOLOv5n. Table 2 presents the number of samples used in the training, validation, and testing phases. We split the dataset according to a 70:20:10 ratio, drawing from various sources including the Internet, AI hub, and IEEE. The number of images refers to the total number of images in the dataset, while the number of instances refers to the total number of individual fire objects present across all images in the dataset.

Table 2. Number of image samples in training/validating/test sets.

Dataset	Source	Number of Images	Number of Instances
Training (17,812)	Internet	12,999	24,845
	AI Hub	4305	4329
	IEEE	508	715
Validation (5441)	Internet	3722	6999
	AI Hub	1580	1571
	IEEE	139	170
Test (2618)	AI Hub	2514	2613
	IEEE	104	219
Total		25,871	41,461

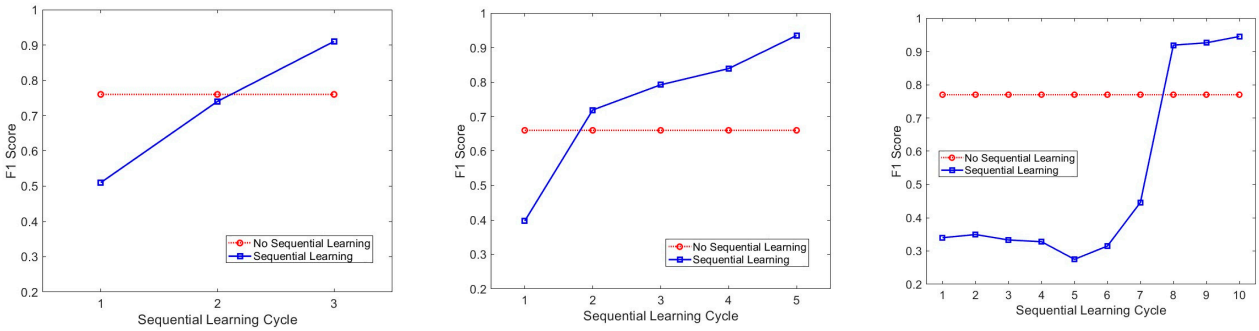
In each sequential learning cycle, image samples in the validation set corresponding to false positives are added to the sub-dataset, forming a sub-dataset for the next sequential learning cycle. Table 3 displays the object size ratio values in each sub-dataset used to partition the dataset equally into sub-datasets.

Table 3. Partitioning a dataset into sub-datasets for three, five, and ten sequential learning cycles.

N	Partitioning a Dataset into Multiple Sub-Datasets Based on Object Size Ratio									
	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
3	>0.07	0.07~0.01	<0.016	-	-	-	-	-	-	-
5	>0.15	0.051~0.15	0.021~0.051	0.006~0.021	<0.006	-	-	-	-	-
10	>0.34	0.15~0.34	0.08~0.15	0.05~0.08	0.03~0.05	0.02~0.03	0.01~0.02	0.006~0.01	0.0016~0.006	<0.0016

4.4. Small Fire Flame Object Detection Results

To verify the effectiveness of the proposed sequential learning scheme, we compared a model trained using sequential learning with one trained using traditional learning without sequential learning. Figure 6 presents the comparison between the results obtained from sequential learning and the baseline with no sequential learning. The significant results of sequential learning methods are evident in the increased performance of small object detection. Initially, the sequential learning approach exhibits low accuracy, indicating initial difficulties in handling smaller-sized objects. The model gradually improves its performance as sequential learning cycles progress, becoming more accurate in detecting small objects by the end of the sequential learning cycles.



(**a**) (**b**) (**c**)

Figure 6. Comparisons of small-sized object detection performances obtained from the proposed sequential learning scheme and the baseline with no sequential learning for three object size categories: (**a**) small; (**b**) petite; (**c**) tiny.

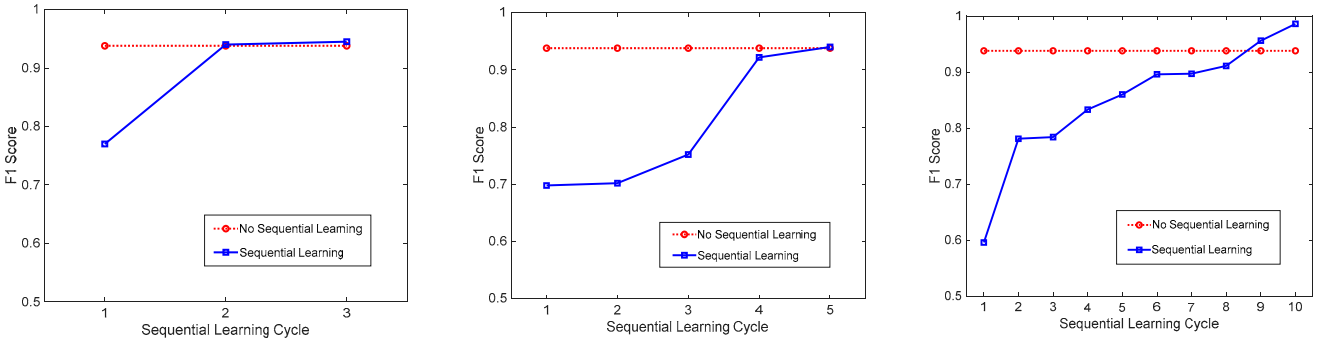
Table 4 displays the number of false positives at each sequential learning cycle for three experiments conducted with 3, 5, and 10 sequential learning cycles, respectively. We count the false positive objects within the test dataset sourced from both the AI Hub and IEEE repositories. This table demonstrates that the number of false positives gradually decreases with each sequential learning cycle.

Table 4. Numbers of false positives at each sequential learning cycle.

N	Sequential Learning Cycle									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
3	1954	1327	1131	-	-	-	-	-	-	-
5	1716	1116	920	867	793	-	-	-	-	-
10	2055	2034	1883	1554	1460	1352	1152	1091	857	811

4.5. Fire Detection Performances for Fire Objects Regardless of Their Sizes

We evaluate the effectiveness of our sequential learning scheme for detecting fire objects of any size across three cycles: 3, 5, and 10. Figure 7 compares the fire detection performances, specifically the F1 scores, between the model trained using sequential learning and the traditional learning approach with no sequential learning. Through this evaluation, we consistently observe that our proposed sequential learning approach outperforms the non-sequential learning method. This underscores not only the efficacy of sequential learning in enhancing the detection of small-sized objects but also its broader impact on fire detection overall.



(**a**) (**b**) (**c**)

Figure 7. Comparisons of fire detection performances obtained from the sequential learning scheme and the baseline with no sequential learning for all fire objects regardless of their sizes: (**a**) 3 cycles; (**b**) 5 cycles; (**c**) 10 cycles.

4.6. Ablation Experiments

In this study, we also aim to demonstrate the effectiveness of sequential learning in early fire detection and to provide a comparative analysis with non-sequential learning methods through various algorithmic experiments. To accomplish this goal, we conducted training using both sequential and non-sequential learning approaches. We employed two light-weight object detection models, YOLOv5n and Single Shot Detector (SSD), with MobileNet-v2 as the backbone and Feature Pyramid Network Lite (FPNLite) as the feature extractor (SSD + MobileNet + FPNLite). SSD + MobileNet + FPNLite is a combination of three elements, indicating that the object detection model is based on the SSD algorithm, uses the MobileNet-v2 architecture as its backbone, and incorporates the FPN structure to handle objects of different sizes and scales within images. This combination is often chosen for applications that require fast and efficient object detection, such as in mobile apps or embedded systems, compared to using only SSD + MobileNet-v2 [36,37].

The sequential learning scheme was evaluated on YOLOv5n and SSD + MobileNet + FPNLite models for detecting small-sized fire objects. Table 5 summarizes the evaluation results measured by F1 scores, showing that sequential learning outperforms non-sequential learning regardless of object detection models.

Table 5. A comparison of the performances of small-sized fire object detection using sequential learning, measured in terms of F1 scores (%), applied to two lightweight models, YOLOv5n and SSD + MobileNet + FPNLite, with and without sequential learning.

Model	*N*	Sequential Learning	No Sequential Learning
YOLOv5n	3	90.8	75.6
	5	93.1	66.1
	10	94.5	77.0
SSD + MobileNet + FPNLite	3	72.1	52.7
	5	60.4	54.7
	10	76.6	54.1

5. Conclusions

This paper presents a sequential learning method to enhance the performance of lightweight deep learning models in detecting small-sized fire objects in surveillance video streams. Our proposed approach involves a series of supervised learning steps, wherein the training dataset is divided into multiple sub-datasets based on the size of fire objects, measured by their object size ratio. Starting with the largest-sized fire objects and progressing to the smallest, the sequential learning aims to refine the model's ability to detect small-sized fire objects relative to the image area.

Experimental results showcase the efficacy of our sequential learning approach. By training a deep learning fire detection model sequentially with a descending order of object size, we achieve notable improvements in detecting small fire objects, with an object size ratio below 0.006 resulting in an F1 score of 93.1%. This represents a substantial 27% enhancement compared to traditional supervised learning methods that do not utilize sequential learning. Our method excels in detecting tiny objects, with an object size ratio below 0.0016 achieving an F1 score of 94.5%, indicating a 17.5% increase compared to the baseline without sequential learning. Our study underscores the effectiveness of employing sequential learning for early fire detection in surveillance videos, particularly focusing on improving small-sized fire object detection using lightweight models.

Author Contributions: Conceptualization, S.G.K. and W.A.S.; methodology, data curation, D.B.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Faculty Research Fund of Sejong University (2023).

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: Author Duy B. Nguyen was employed by the company Pintel, Co. The remaining authors 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. Carta, F.; Zidda, C.; Putzu, M.; Loru, D.; Anedda, M.; Giusto, D. Advancements in forest fire prevention: A comprehensive survey. *Sensors* **2023**, *23*, 6635. [CrossRef] [PubMed]
2. Kanakaraja, P.; Sundar, P.S.; Vaishnavi, N.; Reddy, S.G.K.; Manikanta, G.S. IoT enabled advanced forest fire detecting and monitoring on Ubidots platform. *Mater. Today Proc.* **2021**, *46*, 3907–3914. [CrossRef]
3. Khan, T. A Smart Fire Detector IoT System with Extinguisher Class Recommendation Using Deep Learning. *IoT* **2023**, *4*, 558–581. [CrossRef]
4. Jadon, A.; Omama, M.; Varshney, A.; Ansari, M.S.; Sharma, R. FireNet: A specialized lightweight fire & smoke detection model for real-time IoT applications. *arXiv* **2019**, arXiv:1905.11922.
5. Avazov, K.; Hyun, A.E.; Sami S, A.A.; Khaitov, A.; Abdusalomov, A.B.; Cho, Y.I. Forest Fire Detection and Notification Method Based on AI and IoT Approaches. *Future Internet* **2023**, *15*, 61. [CrossRef]
6. Valikhujaev, Y.; Abdusalomov, A.; Cho, Y.I. Automatic fire and smoke detection method for surveillance systems based on dilated CNNs. *Atmosphere* **2020**, *11*, 1241. [CrossRef]
7. Nallakaruppan, M.K.; Pillai, S.; Bharadwaj, G.; Balusamy, B. Early Detection of Forest Fire using Deep Image Neural Networks. In Proceedings of the 2023 IEEE IAS Global Conference on Emerging Technologies (GlobConET), London, UK, 19–21 May 2023; pp. 1–5.
8. Dilli, B.; Suguna, M. Early Thermal Forest Fire Detection using UAV and Saliency map. In Proceedings of the 2022 5th International Conference on Contemporary Computing and Informatics (IC3I), Uttar Pradesh, India, 14–16 December 2022; pp. 1523–1528.
9. Yang, C.; Pan, Y.; Cao, Y.; Lu, X. CNN-Transformer Hybrid Architecture for Early Fire Detection. In *International Conference on Artificial Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 570–581.
10. Mohnish, S.; Kannan, B.D.; Vasuhi, S. Vision Transformer based Forest Fire Detection for Smart Alert Systems. In Proceedings of the 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 4–6 May 2023; pp. 891–896.
11. Zhao, L.; Zhi, L.; Zhao, C.; Zheng, W. Fire-YOLO: A small target object detection method for fire inspection. *Sustainability* **2022**, *14*, 4930. [CrossRef]
12. Xu, H.; Li, B.; Zhong, F. Light-YOLOv5: A lightweight algorithm for improved YOLOv5 in complex fire scenarios. *Appl. Sci.* **2022**, *12*, 12312. [CrossRef]
13. Tsalera, E.; Papadakis, A.; Voyiatzis, I.; Samarakou, M. CNN-based, contextualized, real-time fire detection in computational resource-constrained environments. *Energy Rep.* **2023**, *9*, 247–257. [CrossRef]
14. Wang, T.; Cao, R.; Wang, L. FE-YOLO: An Efficient and Lightweight Feature-Enhanced Fire Detection Method. In Proceedings of the 2022 3rd International Conference on Electronics, Communications and Information Technology (CECIT), Sanya, China, 23–25 December 2022.
15. Almeida, J.S.; Huang, C.; Nogueira, F.G.; Bhatia, S.; de Albuquerque, V.H.C. EdgeFireSmoke: A Novel Lightweight CNN Model for Real-Time Video Fire–Smoke Detection. *IEEE Trans. Ind. Inform.* **2022**, *18*, 7889–7898. [CrossRef]
16. Xu, Z.; Hong, X.; Chen, T.; Yang, Z.; Shi, Y. Scale-Aware Squeeze-and-Excitation for Lightweight Object Detection. *IEEE Robot. Autom. Lett.* **2022**, *8*, 49–56. [CrossRef]
17. Xie, F.; Li, J.; Wang, Y.; Yang, J. Smoke/Fire Detection with an Improved YOLOX Model. In Proceedings of the 2023 42nd Chinese Control Conference (CCC), Tianjin, China, 24–26 July 2023.
18. Li, L.; Yi, J. Real-time Fire Detection for Urban Tunnels Based on Multi-Source Data and Transfer Learning. In Proceedings of the 4th International Symposium on Computer Engineering and Intelligent Communications (ISCEIC), Nanjing, China, 18–20 August 2023.
19. Nenakhov, I.; Mazhitov, R.; Artemov, K.; Zabihifar, S.H.; Semochkin, A.; Kolyubin, S. Continuous Learning with Random Memory for Object Detection in Robotic Applications. In Proceedings of the 2021 International Conference "Nonlinearity, Information and Robotics" (NIR), Innopolis, Russia, 26–29 August 2021; pp. 1–6.
20. Luo, Y.; Yin, L.; Bai, W.; Mao, K. An Appraisal of Incremental Learning Methods. *Entropy* **2020**, *22*, 1190. [CrossRef] [PubMed]
21. Wu, Y.; Chen, Y.; Wang, L.; Ye, Y.; Liu, Z.; Guo, Y.; Fu, Y. Large scale incremental learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
22. Hasan, M.; Roy-Chowdhury, A.K. A Continuous Learning Framework for Activity Recognition Using Deep Hybrid Feature Models. *IEEE Trans. Multimed.* **2015**, *17*, 1909–1922. [CrossRef]
23. Yang, B.; Deng, X.; Shi, H.; Li, C.; Zhang, G.; Xu, H.; Zhao, S.; Lin, L.; Liang, X. Continual object detection via prototypical task correlation guided gating mechanism. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
24. Menezes, A.; de Moura, G.; Alves, C.; de Carvalho, A. Continual object detection: A review of definitions, strategies, and challenges. *Neural Netw.* **2023**, *161*, 476–493. [CrossRef] [PubMed]

25. Khan, A.; Hassan, B.; Khan, S.; Ahmed, R.; Abuassba, A. DeepFire: A novel dataset and deep transfer learning benchmark for forest fire detection. *Mob. Inf. Syst.* **2022**, *2022*, 5358359. [CrossRef]
26. Reis, H.C.; Turk, V. Detection of forest fire using deep convolutional neural networks with transfer learning approach. *Appl. Soft Comput.* **2023**, *143*, 110362. [CrossRef]
27. Freund, Y.; Schapire, R.E. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139. [CrossRef]
28. Chen, G.; Wang, H.; Chen, K.; Li, Z.; Song, Z.; Liu, Y.; Chen, W.; Knoll, A. A survey of the four pillars for small object detection: Multiscale representation, contextual information, super-resolution, and region proposal. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *52*, 936–953. [CrossRef]
29. 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, 26 June–1 July 2016.
30. Wang, Y.; Yang, D.; Chen, H.; Wang, L.; Gao, Y. Pig Counting Algorithm Based on Improved YOLOv5n Model with Multiscene and Fewer Number of Parameters. *Animals* **2023**, *13*, 3411. [CrossRef] [PubMed]
31. Wang, C.; Wang, C.; Wang, L.; Wang, J.; Liao, J.; Li, Y.; Lan, Y. A Lightweight Cherry Tomato Maturity Real-Time Detection Algorithm Based on Improved YOLOV5n. *Agronomy* **2023**, *13*, 2106. [CrossRef]
32. Wang, S.; Zhang, C.; Xiao, Z. Object Detection in Security Inspection Scenarios Based on YOLOv5s: Exploring Experiments. *J. Phys. Conf. Ser.* **2023**, *2560*, 012018. [CrossRef]
33. AI Hub. Fire Prediction Video. AI Hub. 2021. Available online: https://aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=realm&dataSetSn=176 (accessed on 20 July 2023).
34. Shamsoshoara, A.; Afghah, F.; Razi, A.; Zheng, L.; Fulé, P.; Blasch, E. The FLAME dataset: Aerial Imagery Pile burn detection using drones (UAVs). *Comput. Netw.* **2020**, *193*, 108001. [CrossRef]
35. Wei, S.; Zeng, X.; Qu, Q.; Wang, M.; Su, H.; Shi, J. HRSID: A high-resolution SAR images dataset for ship detection and instance segmentation. *IEEE Access* **2020**, *8*, 120234–120254. [CrossRef]
36. Balaji, K.; Gowri, S. A Real-Time Face Mask Detection Using SSD and MobileNetV2. In Proceedings of the 4th International Conference on Computing and Communications Technologies (ICCCT), Chennai, India, 16–17 December 2021.
37. An, Y.; Tang, J.; Li, Y. A Mobilenet SSDLite model with improved FPN for forest fire detection. In *Chinese Conference on Image and Graphics Technologies*; Springer Nature: Singapore, 2022.

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.