fire

MDPI

Article

An Efficient Forest Fire Target Detection Model Based on Improved YOLOv5

Long Zhang , Jiaming Li and Fuquan Zhang *

College of Information Science and Technology, Nanjing Forestry University, Nanjing 210037, China
* Correspondence: zfq@njfu.edu.cn

Abstract: To tackle the problem of missed detections in long-range detection scenarios caused by the small size of forest fire targets, initiatives have been undertaken to enhance the feature extraction and detection precision of models designed for forest fire imagery. In this study, two algorithms, DenseM-YOLOv5 and SimAM-YOLOv5, were proposed by modifying the backbone network of You Only Look Once version 5 (YOLOv5). From the perspective of lightweight models, compared to YOLOv5, SimAM-YOLOv5 reduced the parameter size by 28.57%. Additionally, although SimAM-YOLOv5 showed a slight decrease in recall rate, it achieved improvements in precision and average precision (AP) to varying degrees. The DenseM-YOLOv5 algorithm achieved a 2.24% increase in precision, as well as improvements of 1.2% in recall rate and 1.52% in AP compared to the YOLOv5 algorithm. Despite having a higher parameter size, the DenseM-YOLOv5 algorithm outperformed the SimAM-YOLOv5 algorithm in terms of precision and AP for forest fire detection.

Keywords: forest fire detection; DenseM-YOLOv5; SimAM-YOLOv5

check for updates

Citation: Zhang, L.; Li, J.; Zhang, F. An Efficient Forest Fire Target Detection Model Based on Improved YOLOv5. *Fire* **2023**, *6*, 291. https://doi.org/10.3390/fire6080291

Academic Editor: Alistair M. S. Smith

Received: 30 June 2023
Revised: 25 July 2023
Accepted: 27 July 2023
Published: 31 July 2023

1. Introduction

Forests serve as a vital natural resource within the Earth's ecosystem, contributing significantly to the preservation of biodiversity, climate regulation, and water resource management [1,2]. The emergence of forest fires inflicts considerable ecological harm, causing widespread loss of vegetation and devastation of habitats, which in turn adversely affects biodiversity and the overall ecological equilibrium [3–5].

Forest fires, as a severe disaster, cause significant harm to the ecosystem and human society [6]. In order to promptly detect and respond to forest fires, scientists and rescue personnel are continuously exploring and developing various fire detection methods [7,8]. Common methods for forest fire detection include manual patrols, aerial remote sensing, and fire detectors. Manual patrols rely on human resources for surveillance and reporting, while aerial remote sensing utilizes remote-sensing technology to monitor vast areas. Fire detectors, on the other hand, employ sensors and detection techniques to monitor environmental parameters in real-time [9–11]. While manual patrols continue to play a crucial role in fire monitoring, the deployment of advanced technologies, including remote-sensing technology and fire detectors, can markedly improve the effectiveness and efficiency of fire monitoring efforts. In addition, the integration of traditional machine learning or deep learning algorithms with remote-sensing technology and fire detectors has emerged as a trend in recent years [12]. Advanced systems are capable of learning from and examining diverse data sources, such as remote-sensing data, environmental factors, and visual media such as images and videos. These systems can estimate the likelihood of a fire starting or detect fires in their early stages. Such capabilities can greatly diminish response times, potentially protecting lives and property from harm. Effective forest fire detection enables early detection and rapid response to fires, allowing for timely measures to control the spread of fire, protect human lives and property, and mitigate ecological damage and economic losses caused by fires [13–15]. Efficient fire detection helps to improve fire warning capabilities, and prompt timely fire response and rescue operations, ensuring that fires are brought under control and extinguished as early as possible, thereby safeguarding the stability and sustainable development of forest ecosystems [16,17].

The swift advancement of deep learning has introduced novel approaches to forest fire detection. Deep-learning-based object-detection approaches exhibit distinct benefits when applied to identifying forest fires [18–20]. These advantages include high accuracy, fast detection speed, flexible installation and the ability to adapt to various fire features [21–23]. Mohnish et al. (2022) [24] preprocessed the images in the dataset and input them into a convolutional neural network (CNN) for feature extraction and detection. The detection accuracy on the training and testing datasets was 93% and 92%, respectively. Chen et al. (2023) [25] introduced an enhanced multi-scale forest fire detection model called YOLOv5s-CCAB, which is based on the YOLOv5s architecture. This model aims to tackle the issue of low detection accuracy resulting from the multi-scale attributes and variable morphology of forest fires. Experimental results reveal that YOLOv5s-CCAB, tested on a multi-scale forest fire dataset, boosts AP@0.5 by 6.2% to 87.7% and achieves an FPS (frames per second) of 36.6, demonstrating its high detection accuracy and speed. Yar et al. (2023) [26] put forward an enhanced YOLOv5s model that incorporates a stem module in the backbone, substitutes the larger kernel with a smaller one in the spatial pyramid pooling (SPP, neck), and includes a P6 module in the head. This model yields promising outcomes with low complexity and a compact model size, and it is capable of detecting both small and large fire areas within images. Ghali et al. (2022) [27] presented a new deep ensemble learning method that merges EfficientNet-B5 and DenseNet-201 models to identify and categorize wildfires using aerial imagery. Their suggested wildfire classification model attains an accuracy of 85.12%, surpassing numerous leading research outcomes in the field. Zhou et al. (2023) [28] utilized semi-supervised knowledge extraction (SSLD) during training to enhance the convergence speed and accuracy of the model. This was achieved by incorporating the overall structure of YOLOv5 and MobileNetV3 as the backbone network. Dilli et al. (2022) [29] employed a deep-learning-based YOLO model from the target detection library to perform early wildfire detection using thermal images from unmanned aerial vehicles (UAVs). To address the limitations of using thermal images, they integrated a significance graph with the thermal imagery. Their method is believed to offer valuable technical support for nighttime monitoring, potentially mitigating the devastating impact on forest resources, human lives, and wildlife during the initial stages of wildfires.

At present, YOLOv5 has demonstrated commendable performance in the field of object detection, as evidenced by various studies [30,31]. However, when it comes to long-range detection, particularly in the context of forest fires, the model encounters certain limitations. At times, the targets for detection in forest fires are typically quite small, and this becomes even more challenging during the initial stages of a fire. In certain scenarios, the flames and smoke plumes at the onset of a fire may be hidden by dense foliage and are often not large enough to be easily detected by the model. The intrinsic nature of forest fires presents a considerable obstacle for the detection performance of the YOLOv5 model. As a result, there can be instances of missed detections, where the model fails to identify the early signs of a fire. This not only compromises the timeliness of fire detection but also impacts the overall accuracy of the network. Therefore, while YOLOv5 has shown promise in object detection, its effectiveness in the early and long-range detection of forest fires remains an area that requires further improvement and exploration. Furthermore, the Cross Stage Partial (CSP) module in the YOLOv5 model is primarily intended for multi-class detection tasks, which may not be optimal for feature extraction in forest fire recognition, given the relatively low complexity of such scenarios. To overcome this limitation and improve the network's ability to capture features of small targets, this study proposes targeted enhancement strategies with two new network models: DenseM-YOLOv5 and SimAM-YOLOv5, both of which are built upon the backbone network of the YOLOv5 algorithm. The DenseM-YOLOv5 model incorporates Densely Connected Convolutional Networks (DenseNet), a network that improves feature propagation and reduces parameters. Additionally, a DenseM module was designed specifically for detecting small targets in forest fire recognition [32]. The SimAM module, equipped with an optimized energy function, takes into account both spatial and channel dimensions, enabling the network to acquire more discerning neurons. This attention module infers 3D weights for feature maps without introducing additional parameters. By emphasizing important features and suppressing background interference, the SimAM-YOLOv5 model achieves improved detection accuracy compared with YOLOv5. These strategies aim to enhance the network's representation capability and optimize feature extraction specifically for forest fire detection.

2. Materials and Methods

2.1. Hyperparameter Settings and Dataset

2.1.1. Hyperparameter Settings

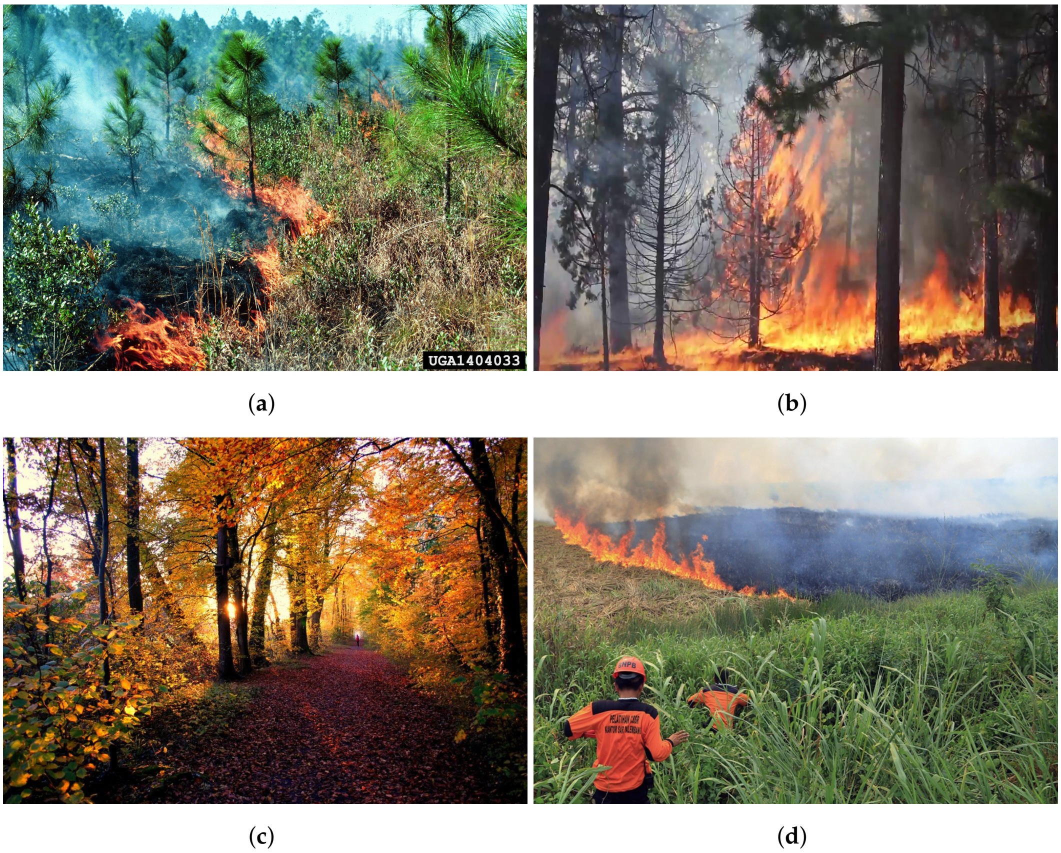
The hyperparameter settings used for experimentation in this study are displayed in Table 1. The settings for these hyperparameters are consistent across all three tested models. These settings play a crucial role in determining the performance of the proposed approach. The selection of hyperparameters is determined by considering a balance between system processing capabilities (demands for computation and memory) and accuracy, in conjunction with the results of multiple experiments and empirical evaluations. By carefully selecting and tuning these hyperparameters, we aim to achieve optimal results in the forest fire recognition task. The table provides a comprehensive overview of the specific values chosen for each hyperparameter, including image size, epochs, batch size, initial learning rate (Lr0), and optimizer. In the realm of deep learning, the image size, typically measured in pixels, determines the volume of input data for the neural network model, as well as the computational complexity and resource requirements. The term epochs represents the number of times the model learns from the entire dataset, with an excessive or insufficient number of epochs potentially affecting the outcome of model training. Batch size refers to the number of samples used each time the model weights are updated, directly impacting the stability of model training and the utilization of computational resources. The initial learning rate (Lr0) sets the pace of learning at the onset of model training. It requires meticulous adjustment based on the specifics of the model to ensure that the model can effectively and swiftly converge to the optimal solution. Furthermore, SGD (stochastic gradient descent) is an optimization algorithm employed to discover the local minimum of the loss function. In comparison to gradient descent (GD), SGD estimates the gradients using only a subset of samples during each iteration. As a result, SGD requires significantly less time for each update, leading to faster convergence. It possesses the characteristics of high computational efficiency and excellent scalability [33]. The chosen settings, derived from initial experiments and empirical evaluation, were established to guarantee the effectiveness of the modified approach. Hyperparameter tuning is an iterative process, and it is worth noting that the optimal values may vary depending on the dataset, the problem at hand, and the model architecture. As such, it is crucial to regularly reassess and fine-tune the hyperparameters as new data becomes available or as the problem being solved evolves. Continuous monitoring and adjustment of hyperparameters can help ensure that the model remains optimized and performs well over time.

Table 1. Hyperparameter settings.

Type	Value
Image Size	640 × 640
Epochs	200
Batch Size	8
Lr0	0.01
Optimizer	SGD

2.1.2. Dataset

To obtain the necessary forest fire images for model training, a mix of conventional forest fire photos and photos without forest fire were gathered through web scraping methods [34]. Additionally, a series of forest fire images were extracted from downloaded forest fire videos by capturing frames. This comprehensive approach ensured a diverse and representative dataset for training the model. In addition, some authors of research papers have also made certain fire datasets publicly available, such as the BoWFireDataset [35]. In the end, we gathered a total of 2328 images. Among these, 716 images captured instances of forest fires, while the remaining 1612 images were of forest environments without fire. These images were then used to create a comprehensive forest fire dataset. To ensure reliable evaluation, we partitioned 80% of the dataset as the training set, and 20% as the validation set [36]. This division allows for comprehensive training and effective validation of the dataset's performance. Importantly, the partitioning was carried out randomly to maintain the diversity and representativeness of both the training and validation sets [37]. Figure 1 showcases a collection of fire and non-fire images included in the acquired dataset.



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

(**c**) (**d**)

Figure 1. Dateset of forest fires: (**a**,**b**,**d**) fire images; and (**c**) non-fire image.

2.1.3. Model Performance Evaluation Index

The main task of our research is treated as a binary classification problem, where the outcome is either fire or non-fire. Within the forest fire classification, fire represents a positive example, while non-fire constitutes a negative example. In the binary classification problem of forest fire detection, there are four possible situations: true positive (*TP*), when the model accurately predicts a fire; true negative (*TN*), when the model correctly predicts no fire; false positive (*FP*), when the model mistakenly identifies a non-fire as a fire; and false negative (*FN*), when the model incorrectly classifies a fire as a non-fire [38].

One of the crucial metrics for assessing a model's performance is precision, which quantifies the model's prediction accuracy by determining the ratio of true positive instances to the total number of instances predicted as positive by the model. It is an important measure, especially in situations where false positives can have significant consequences [39]. The calculation method is shown as Equation (1) [40].

$$\text{Precision} = \frac{TP}{TP+FP} \tag{1}$$

Recall, also known as sensitivity or true positive rate (TPR), is an essential metric that evaluates the percentage of true positive instances correctly identified as positive by the model, relative to the total number of true positive instances [39]. In the context of forest fire detection, a high recall rate signifies the algorithm's effectiveness in detecting and precisely locating fire points within images. This metric is particularly valuable as it ensures that potential fire incidents are not overlooked, minimizing the risk of undetected fires in forest areas. The calculation method is shown as Equation (2) [41].

$$\text{Recall} = \frac{TP}{TP+FN} \tag{2}$$

Average precision (AP) is also a fundamental metric used to evaluate the performance of object-detection algorithms. It provides a comprehensive assessment of the trade-off between precision and recall by considering the entire precision–recall (P–R) curve. The AP is determined by calculating the area beneath the curve. By considering the complete P–R curve, AP provides a more informative evaluation of the algorithm's performance compared to using a single point on the curve. It takes into account the precision–recall trade-off at different operating points, capturing the algorithm's ability to balance accurate positive predictions (precision) with the ability to identify all positive instances (recall). Higher AP values indicate superior performance, demonstrating the algorithm's capability to achieve both high precision and recall simultaneously [42]. The method for computation is represented by Equation (3) [41].

$$\text{AP} = \int_0^1 P(R)dR \tag{3}$$

While computing AP, the average precision values across various classes are weighted and averaged to derive the mean average precision (mAP). Equation (4) [41] illustrates the calculation formula, where n denotes the total number of classes and APi signifies the average precision value for the th class.

$$\text{mAP} = \frac{\sum_{i=1}^{n} AP_i}{n} \tag{4}$$

mAP is a standard metric frequently employed to assess the effectiveness of object-detection algorithms across multiple object classes. It provides a reliable measure for different-sized and difficulty-level objects and is widely used in performance comparison and optimization of object-detection algorithms. The primary emphasis of this study lies in the detection of forest fires, which pertains to a single class exclusively. Therefore, under the same Intersection Over Union (IOU) threshold, the values of mAP and AP are the same [43]. Hence, this study adopts the AP metric and sets the IOU threshold to 50%, denoted as AP50.

2.2. *YOLOv5 Algorithm Structure*

YOLOv5 is an object-detection architecture known for its high performance and efficiency, demonstrating rapid, precise, and adaptable features [44]. It has broad applications in various computer vision tasks and provides powerful tools and technical support for real-time object detection. Its unique design and optimization make YOLOv5 perform exceptionally well in handling large-scale data with less computational resources [45]. This advanced architecture enables efficient processing of vast amounts of information without compromising performance. YOLOv5's strength lies in its ability to adapt to various target detection tasks. This versatility and flexibility empowers researchers and developers to explore new possibilities and drive innovation in the field.

YOLOv5 utilizes a single-stage detection technique in which the whole image is fed into the network, allowing it to directly determine the locations and classifications of objects within the images. This end-to-end detection approach allows YOLOv5 to quickly and precisely identify multiple objects, making it well-suited for real-time tasks. The complete network architecture of YOLOv5 is depicted in the Figure 2. YOLOv5 mainly consists of two parts: the backbone and the neck. In the backbone, the cross-stage partial network (CSP, or C3 as shown in the figure) structure uses two 1 × 1 convolutions for the transformation of input features [46]. The first CSP is used for feature extraction in the backbone section. The second CSP is used for feature fusion in the neck section. In addition to the CSP module, the backbone also has a spatial pyramid pooling fast (SPPF) module, whose function is to extract the global information of the detection target [47]. The neck of YOLOv5 uses the path aggregation network (PANet) structure to aggregate the features. The neck is primarily used for generating feature pyramids in order to enhance the model's detection of objects at different scales, thereby enabling the model to recognize the same object at different sizes and scales [48].

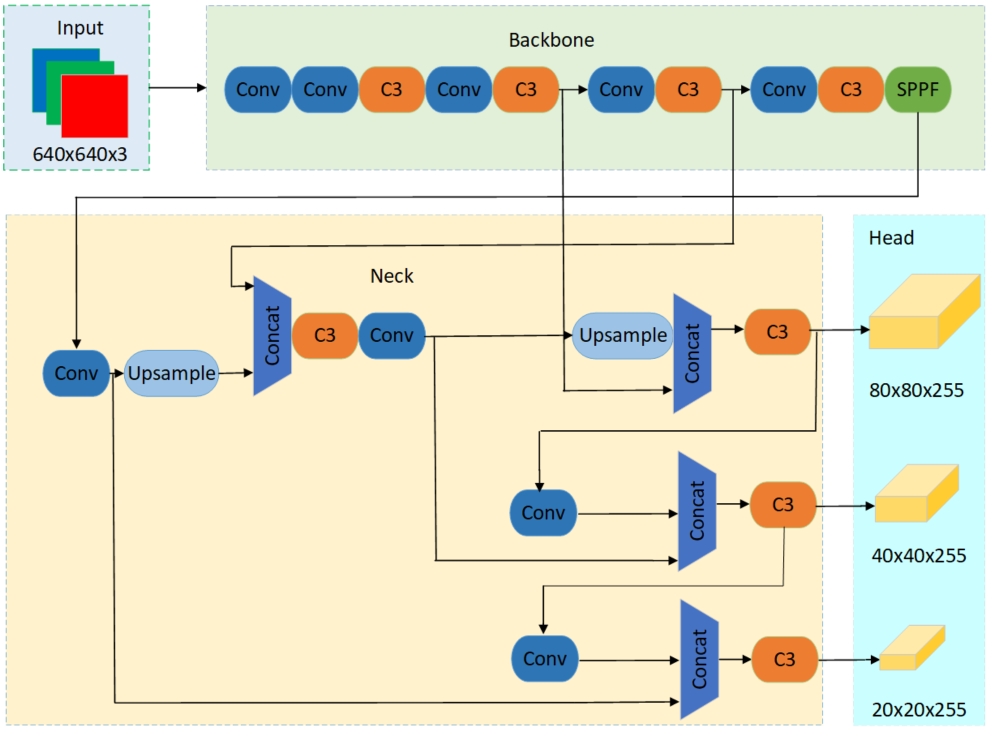


Figure 2. YOLOv5 Model Architecture.

2.3. *Improving the Network Used by the YOLO5 Algorithm*

Improvements to the Backbone Network

In the YOLOv5 model, the CSP structure in the backbone network is primarily designed for multi-class object-detection tasks and has shown good performance on datasets such as COCO and PASCAL VOC. However, for the specific problem of forest fire recognition, which has relatively lower complexity, the CSP structure may not be optimal for extracting relevant features. Therefore, to improve feature extraction and enhance the detection accuracy of forest fires, it is proposed to replace all CSP structures in the YOLOv5 backbone network with two new network models: DenseM-YOLOv5 and SimAM-YOLOv5.

Densely Connected Convolutional Networks (DenseNet), published in 2017 at CVPR, introduced a dense convolutional network that leverages short connections between layers close to the input and those close to the output. This approach enables more accurate training and higher efficiency in convolutional networks. DenseNet effectively alleviates the problem of gradient vanishing, significantly reduces the number of parameters, improves computational efficiency, and enhances feature propagation. When stacking multiple convolutional and pooling layers, network models can suffer from degradation and gradient vanishing issues. In ResNet, residual connections were introduced to address this problem and ensure better information and gradient flow. To maximize information flow between layers, DenseNet combines all layers using direct connections. Each layer receives additional inputs from all preceding layers, allowing the feature maps from earlier layers to be passed to all subsequent layers [49]. Unlike ResNet, feature combination in DenseNet does not involve summation; instead, connections are used to achieve combination. For feature reuse, DenseNet connects the current layer to all previous layers in the network and designs each layer to be narrow, enabling each layer to learn only a small number of features and reducing redundancy in feature learning [50].

In order to enhance the detection of targets which are small in the context of forest fire situations, a DenseM module was specifically devised, as illustrated in Figure 3. The DenseM module divides the input feature map into two separate branches. One branch undergoes a CBS operation. In this context, CBS denotes an operational sequence comprised of Convolution (Conv), Batch Normalization (BN), and the Sigmoid Linear Unit (SiLU) activation function. The other branch first goes through CBS, then passes through DenseNet, and undergoes another CBS operation. The results from the two branches are then merged and subjected to another CBS operation.

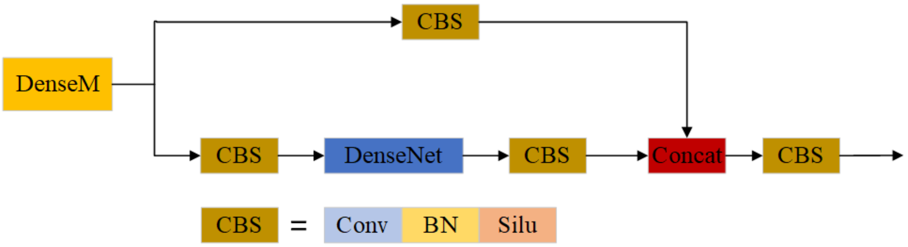


Figure 3. DenseM module.

To enhance the ability of the model to capture forest fire features and improve detection accuracy, all CSP structures in the backbone network of the YOLOv5 model were substituted with DenseM. This modification led to the creation of the DenseM-YOLOv5 model. The structure of the backbone network, incorporating the DenseM modules, is depicted in Figure 4.

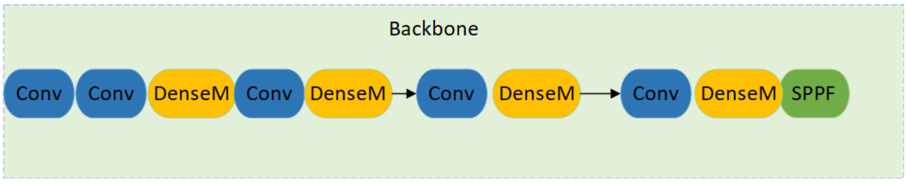


Figure 4. Backbone network of DenseM-YOLOv5 model.

At present, attention mechanisms generally face two primary constraints. First, they are only capable of enhancing features across either the channel or spatial dimensions, limiting their ability to learn flexible attention weights across both channel and spatial variations. Additionally, their pooling and other components require a series of complex elements. To address these issues based on neural science theories, the SimAM module is proposed. By considering both spatial and channel dimensions, the SimAM module infers 3D weights for the current neuron, enabling the network to learn more discriminative neurons. SimAM serves as a conceptually simple yet remarkably effective attention module [51]. Compared to common spatial and channel attention modules, SimAM has the ability to deduce 3D weights for the feature maps within a layer without adding extra parameters. Furthermore, an optimized energy function is proposed to determine the importance of each neuron. In terms of accuracy, the SimAM module surpasses commonly utilized attention modules such as Squeeze-and-Excitation (SE), Convolutional Block Attention Module (CBAM) and Efficient Channel Attention (ECA) on datasets such as CIFAR-10 and CIFAR-100 [51,52].

In order to emphasize important features and suppress irrelevant background interference, thereby improving detection accuracy, the SimAM module is applied to replace all CSP modules in the YOLOv5 backbone network, resulting in the SimAM-YOLOv5 model. Figure 5 depicts the SimAM-YOLOv5's backbone network.

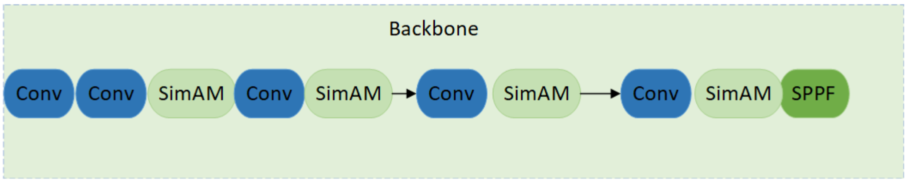


Figure 5. Backbone network of SimAM-YOLOv5 model.

3. Results

Comparison of Multiple Model Results

To verify the feasibility of the SimAM-YOLOv5 and DenseM-YOLOv5 algorithms, experimental results were compared among YOLOv5, SimAM-YOLOv5, and DenseM-YOLOv5. The experimental results are shown in Table 2. In order to assess the complexity of the models, the number of parameters for each network was calculated. By iterating through each layer and component of the model, the parameter information of each submodule, including weight matrices and bias vectors, can be obtained. These are returned in the form of a generator. These parameter tensors are then iterated over, the number of their elements is calculated, and these quantities are aggregated to derive the total number of parameters in the entire model. This process can be achieved by using the 'model.parameters()' function of PyTorch and operations on the parameter tensors. From the perspective of lightweight models, compared to the YOLOv5 algorithm, the parameter count of the SimAM-YOLOv5 algorithm decreased by 28.57%. SimAM-YOLOv5 achieved a detection accuracy of 81.95%, an improvement of 2.07% compared to YOLOv5, with a slight decrease in recall rate. The average precision (AP) of the two models differed only by 0.02%. Furthermore, from Table 2, it can be observed that compared to YOLOv5 and SimAM-YOLOv5, the DenseM-YOLOv5 algorithm showed effective improvements in precision, recall rate, and AP, despite having a higher parameter count. Specifically, compared to YOLOv5, DenseM-YOLOv5 achieved a precision of 82.12%, an improvement of 2.24%, and a recall rate and AP of 82.15% and 87.19%, respectively, representing increases of 1.2% and 1.52% each.

Table 2. The comparison experiment of DenseM-YOLOv5 and SimAM-YOLOv5 models.

Model	P/%	R/%	AP/%	Parameter/M
YOLOv5	79.88	80.55	85.67	7.0
SimAM-YOLOv5	81.95	80.32	85.69	5.0
DenseM-YOLOv5	82.12	81.75	87.19	8.6

The training results of the DenseM-YOLOv5 algorithm and YOLOv5, as they vary with the number of iterations, are shown in Figure 6. It can be observed that after reaching 50 iterations, DenseM-YOLOv5 shows a higher accuracy compared to YOLOv5. However, there is no significant improvement in terms of recall rate compared to the YOLOv5 model. In terms of AP, after 75 iterations, DenseM-YOLOv5 consistently outperforms YOLOv5 in terms of AP value. Given that SimAM-YOLOv5 is a lightweight model requiring no additional parameters, its performance is slightly inferior to that of DenseM-YOLOv5. Its accuracy improves after 150 iterations, with the recall reaching a similar level to that of

YOLOv5 after 200 iterations. Additionally, the AP value of SimAM-YOLOv5 outperforms YOLOv5 subsequent to 150 iterations.

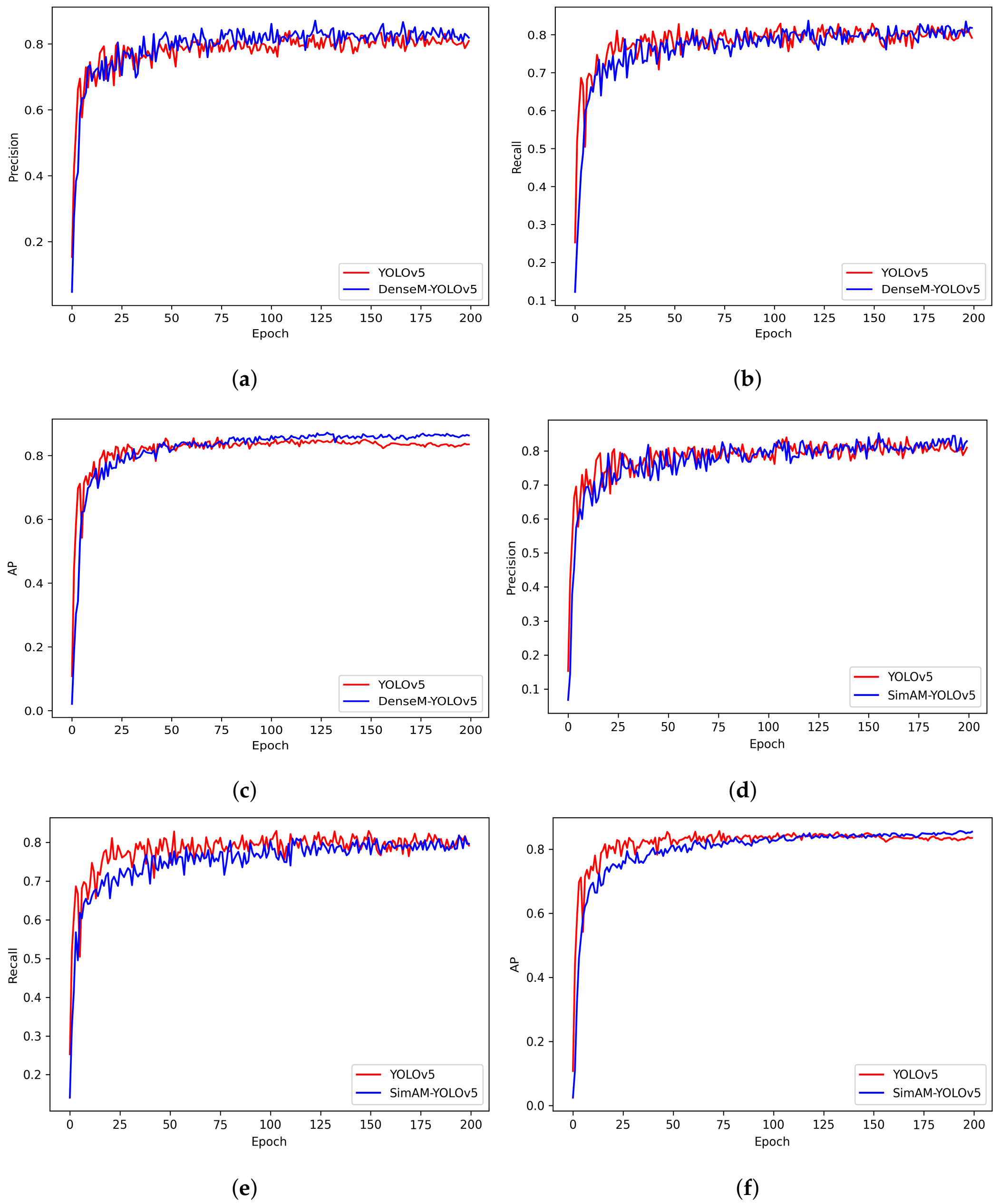


Figure 6. (**a**–**c**) Comparison of training results between YOLOv5 and DenseM-YOLOv5. (**d**–**f**) Comparison of training results between YOLOv5 and SimAM-YOLOv5.

The experiment conducted tests on an image dataset from a drone's perspective using the YOLOv5 algorithm as well as the DenseM-YOLOv5 algorithm, which has higher detection accuracy and AP. Figure 7 presents a visual representation of several test results. The test results belonging to the YOLOv5 algorithm are depicted in Figure 7a,c,e, while Figure 7b,d,f illustrate the test results of the DenseM-YOLOv5 algorithm. From Figure 7a,b, it can be observed that in images with a single fire point, the DenseM-YOLOv5 algorithm performs slightly better in terms of detection compared to the YOLOv5 algorithm. Figure 7c,d show that when there are two fire points in the forest fire image, YOLOv5 only detects one of them, while the DenseM-YOLOv5 algorithm is able to detect both fire points. As shown in Figure 7e,f, when detecting multiple fire points in the target image, the YOLOv5 algorithm is prone to missing detections, whereas the DenseM-YOLOv5 algorithm can detect multiple fire targets in the image. As a result, the utilization of the detection method based on DenseM-YOLOv5 can reduce the occurrence of missed detections when dealing with small targets in forest fire images.

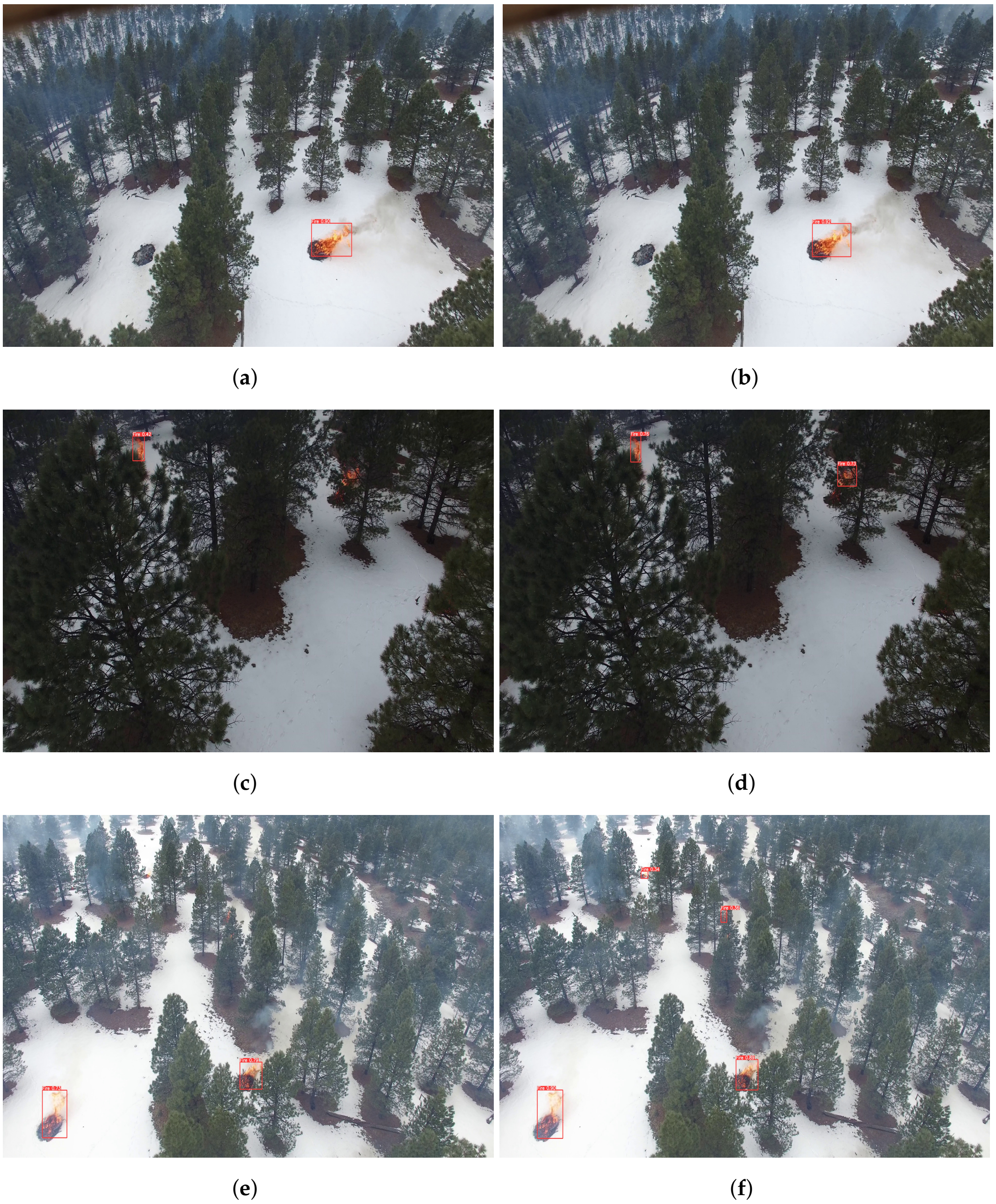

(**a**) (**b**)
(**c**) (**d**)
(**e**) (**f**)

Figure 7. Partial test results of YOLOv5 and DenseM-YOLOv5 models: (**a**,**c**,**e**) YOLOv5 algorithm; and (**b**,**d**,**f**) DenseM-YOLOv5 algorithm

4. Discussion

In this study, we proposed two modified YOLOv5 models, DenseM-YOLOv5 and SimAM-YOLOv5, to improve forest fire detection accuracy, particularly when dealing with small targets in forest fire images. By replacing the CSP modules in the YOLOv5 backbone network with DenseM modules, the DenseM-YOLOv5 algorithm achieves better feature extraction for small targets in forest fire recognition. The experimental results demonstrate that DenseM-YOLOv5 outperforms YOLOv5 in terms of precision, recall rate, and average precision (AP), despite having a higher parameter size. This suggests that the DenseM module effectively addresses the challenge of missed detections due to the small size of forest fire targets in long-range detection scenarios. On the other hand, the SimAM-YOLOv5 algorithm, while reducing the parameter size by 28.57% compared to YOLOv5, achieved a detection accuracy of 81.95%, an improvement of 2.07% compared to YOLOv5. Although there was a slight decrease in recall rate, the precision and average precision (AP) were improved. This indicates that the SimAM module, with its optimized energy function considering both spatial and channel dimensions, effectively emphasized important features and suppressed background interference, resulting in improved detection accuracy.

Future research directions can focus on optimizing the DenseM and SimAM modules. Further research could be conducted to optimize the design and implementation of the DenseM and SimAM modules, potentially improving their effectiveness in feature extraction and background suppression. The combination of DenseM and SimAM modules can be explored, which may yield superior performance in terms of accuracy and efficiency by harnessing the advantages of each module. Furthermore, adapting the models for real-time applications can be explored [53]. Optimizing the models for real-time detection of forest fires could lead to more practical applications, such as early-warning systems and real-time monitoring of forest fire progression. In addition, considering the significant computational and storage requirements of deep learning networks, which are detrimental to the susceptibility of wildfires, future work will focus on studying lightweight and efficient models, such as MobileNets and EfficientNets. Previous studies have shown that MobileNet and EfficientNet are lightweight convolutional neural network structures [54,55]. MobileNets use depthwise separable convolutions in place of standard convolutions, which significantly reduces computational cost [56]. EfficientNets employ the compound scaling method, which balances model complexity and computational efficiency, to simultaneously improve model accuracy and efficiency [57]. Researchers have utilized these architectures for forest fire detection, or incorporated them into other deep learning networks to achieve efficient and lightweight forest fire detection networks. Recently, the X-MobileNet was developed, with a key design focus on the efficient and swift detection of forest fires using UAV-sourced imagery. This model elegantly employs the scalable MobileNetV2 architecture and stands out due to its suitability for real-time deployment on drones, which often grapple with limited computational resources [58]. Another lightweight deep learning model has been developed, specifically tailored for wildfire detection using video cameras. This model achieves an impressive balance between accuracy and efficiency, accomplished through strategic modifications to the YOLOv5s backbone with MobileNet Version3 architectures, optimization of feature extraction and fusion processes, and the incorporation of attention mechanisms. It demonstrates promising performance for real-time wildfire detection on edge devices [59]. A lightweight encoder–decoder network, which is a modification of EfficientNet Version2, is proposed for real-time wildfire segmentation in UAV images. The proposed model utilizes efficient blocks such as depthwise convolutions and attention gating to reduce parameters while maintaining accuracy. The lightweight architecture enables deployment on drones with limited computing resources [55]. These approaches would enable more efficient and rapid identification of wildfires in images, consuming fewer computational resources while still achieving desirable accuracy and performance. This would be advantageous for real-time wildfire detection and the swift deployment of such systems.

In summary, the DenseM-YOLOv5 and SimAM-YOLOv5 models represent a promising step forward in forest fire detection. By exploring the suggested future research directions, we can continue to advance the field and develop increasingly effective solutions for forest fire monitoring and prevention.

Author Contributions: L.Z., J.L. and F.Z. have contributed equally in each and every stage of this research work. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Seidl, R.; Turner, M.G. Post-disturbance reorganization of forest ecosystems in a changing world. *Proc. Natl. Acad. Sci. USA* **2022**, *119*, e2202190119. [CrossRef] [PubMed]
2. Tiemann, A.; Ring, I. Towards ecosystem service assessment: Developing biophysical indicators for forest ecosystem services. *Ecol. Indic.* **2022**, *137*, 108704. [CrossRef]
3. Spicer, M.E.; Radhamoni, H.V.N.; Duguid, M.C.; Queenborough, S.A.; Comita, L.S. Herbaceous plant diversity in forest ecosystems: Patterns, mechanisms, and threats. *Plant Ecol.* **2022**, *223*, 117–129. [CrossRef]
4. Yadav, V.S.; Yadav, S.S.; Gupta, S.R.; Meena, R.S.; Lal, R.; Sheoran, N.S.; Jhariya, M.K. Carbon sequestration potential and CO_2 fluxes in a tropical forest ecosystem. *Ecol. Eng.* **2022**, *176*, 106541. [CrossRef]
5. Sorge, S.; Mann, C.; Schleyer, C.; Loft, L.; Spacek, M.; Hernández-Morcillo, M.; Kluvankova, T. Understanding dynamics of forest ecosystem services governance: A socio-ecological-technical-analytical framework. *Ecosyst. Serv.* **2022**, *55*, 101427. [CrossRef]
6. Wu, Z.; Wang, B.; Li, M.; Tian, Y.; Quan, Y.; Liu, J. Simulation of forest fire spread based on artificial intelligence. *Ecol. Indic.* **2022**, *136*, 108653. [CrossRef]
7. Agbeshie, A.A.; Abugre, S.; Atta-Darkwa, T.; Awuah, R. A review of the effects of forest fire on soil properties. *J. For. Res.* **2022**, *33*, 1419–1441. [CrossRef]
8. Morante-Carballo, F.; Bravo-Montero, L.; Carrión-Mero, P.; Velastegui-Montoya, A.; Berrezueta, E. Forest fire assessment using remote sensing to support the development of an action plan proposal in Ecuador. *Remote Sens.* **2022**, *14*, 1783. [CrossRef]
9. Yandouzi, M.; Grart, M.; Idrissi, I.; Moussaoui, O.; Azizi, M.; Ghoumid, K.; Elmiad, A.K. Review on forest fires detection and prediction using deep learning and drones. *J. Theor. Appl. Inf. Technol.* **2022**, *100*, 4565–4576.
10. Khan, F.; Xu, Z.; Sun, J.; Khan, F.M.; Ahmed, A.; Zhao, Y. Recent advances in sensors for fire detection. *Sensors* **2022**, *22*, 3310. [CrossRef]
11. Yang, X.; Wang, Y.; Liu, X.; Liu, Y. High-Precision Real-Time Forest Fire Video Detection Using One-Class Model. *Forests* **2022**, *13*, 1826. [CrossRef]
12. Qian, J.; Lin, J.; Bai, D.; Xu, R.; Lin, H. Omni-Dimensional Dynamic Convolution Meets Bottleneck Transformer: A Novel Improved High Accuracy Forest Fire Smoke Detection Model. *Forests* **2023**, *14*, 838. [CrossRef]
13. Huang, J.; He, Z.; Guan, Y.; Zhang, H. Real-time forest fire detection by ensemble lightweight YOLOX-L and defogging method. *Sensors* **2023**, *23*, 1894. [CrossRef] [PubMed]
14. Martynyuk, A.A.; Savchenkova, V.A.; Korshunov, N.A.; Kotelnikov, R.V. Methods for the use of the best Russian innovations in forest fire detection and suppression. *J. For. Res.* **2021**, *32*, 2255–2263. [CrossRef]
15. Tehseen, A.; Zafar, N.A.; Ali, T.; Jameel, F.; Alkhammash, E.H. Formal Modeling of IoT and Drone-Based Forest Fire Detection and Counteraction System. *Electronics* **2021**, *11*, 128. [CrossRef]
16. Sathishkumar, V.E.; Cho, J.; Subramanian, M.; Naren, O.S. Forest fire and smoke detection using deep learning-based learning without forgetting. *Fire Ecol.* **2023**, *19*, 1–17. [CrossRef]
17. Zheng, X.; Chen, F.; Lou, L.; Cheng, P.; Huang, Y. Real-Time Detection of Full-Scale Forest Fire Smoke Based on Deep Convolution Neural Network. *Remote Sens.* **2022**, *14*, 536 [CrossRef]
18. Kang, Y.; Jang, E.; Im, J.; Kwon, C. A deep learning model using geostationary satellite data for forest fire detection with reduced detection latency. *GISci. Remote Sens.* **2022**, *59*, 2019–2035. [CrossRef]
19. Abdusalomov, A.B.; Islam, B.M.S.; Nasimov, R.; Mukhiddinov, M.; Whangbo, T.K. An improved forest fire detection method based on the detectron2 model and a deep learning approach. *Sensors* **2023**, *23*, 1512. [CrossRef] [PubMed]
20. Jose, T.K.; Deepak, K.P.; Ezhilarasan, V.; Santhosh, K.M.; Suriya, S. A Survey on Fire Detection-Based Features Extraction Using Deep Learning. In *ICT with Intelligent Applications: Proceedings of ICTIS*; Springer Nature: Singapore, 2022; Volume 1, pp. 313–323.
21. Lin, J.; Lin, H.; Wang, F. STPM_SAHI: A Small-Target Forest Fire Detection Model Based on Swin Transformer and Slicing Aided Hyper Inference. *Forests* **2022**, *13*, 1603. [CrossRef]
22. Guan, Z.; Miao, X.; Mu, Y.; Sun, Q.; Ye, Q.; Gao, D. Forest Fire Segmentation from Aerial Imagery Data Using an Improved Instance Segmentation Model. *Remote Sens.* **2022**, *14*, 3159. [CrossRef]
23. Seydi, S.T.; Saeidi, V.; Kalantar, B.; Ueda, N.; Halin, A.A. Fire-Net: A deep learning framework for active forest fire detection. *J. Sensors* **2022**, *2022*, 8044390 . [CrossRef]
24. Mohnish, S.; Akshay, K.P.; Pavithra, P.; Ezhilarasi, S. Deep Learning based Forest Fire Detection and Alert System. In Proceedings of the 2022 International Conference on Communication, Computing and Internet of Things (IC3IoT), Chennai, India, 10–11 March 2022; pp. 1–5.
25. Chen, G.; Zhou, H.; Li, Z.; Gao, Y.; Bai, D.; Xu, R.; Lin, H. Multi-Scale Forest Fire Recognition Model Based on Improved YOLOv5s. *Forests* **2023**, *14*, 315. [CrossRef]

26. Yar, H.; Khan, Z.A.; Ullah, F.U.M.; Ullah, W.; Baik, S.W. A modified YOLOv5 architecture for efficient fire detection in smart cities. *Expert Syst. Appl.* **2023**, *231*, 120465. [CrossRef]
27. Ghali, R.; Akhloufi, M.A.; Mseddi, W.S. Deep Learning and Transformer Approaches for UAV-Based Wildfire Detection and Segmentation. *Sensors* **2022**, *22*, 1977. [CrossRef]
28. Zhou, M.; Wu, L.; Liu, S.; Li, J. UAV forest fire detection based on lightweight YOLOv5 model. *Multimed. Tools Appl.* **2023**, 1–12. [CrossRef]
29. 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), Greater Noida, India, 14–16 December 2022; pp. 1523–1528.
30. Jiang, W.; Jiang, Z. Research on early fire detection of Yolo V5 based on multiple transfer learning. *Fire Sci. Technol.* **2021**, *40*, 109–112.
31. Wu, Z.; Xue, R.; Li, H. Real-Time Video Fire Detection via Modified YOLOv5 Network Model. *Fire Technol.* **2022**, *58*, 2377–2403. [CrossRef]
32. Li, S.; Deng, M.; Lee, J.; Sinha, A.; Barbastathis, G. Imaging through glass diffusers using densely connected convolutional networks. *Optica* **2018**, *5*, 803–813. [CrossRef]
33. Wijnhoven, R.G.; de With, P.H.N. Fast training of object detection using stochastic gradient descent. In Proceedings of the 2015 20th International Conference on Pattern Recognition, Istanbul, Turkey, 23–26 August 2010; pp. 424–427.
34. Uzun, E. A novel web scraping approach using the additional information obtained from web pages. *IEEE Access* **2020**, *8*, 61726–61740. [CrossRef]
35. Chino, D.Y.; Avalhais, L.P.; Rodrigues, J.F.; Traina, A.J. Bowfire: Detection of fire in still images by integrating pixel color and texture analysis. In Proceedings of the 2015 28th SIBGRAPI Conference on Graphics, Patterns and Images, Salvador, Bahia, Brazil, 26–29 August 2015; pp. 95–102.
36. Nguyen, Q.H.; Ly, H.B.; Ho, L.S.; Al-Ansari, N.; Le, H.V.; Tran, V.Q.; Prakash, I.; Pham, B.T. Influence of data splitting on performance of machine learning models in prediction of shear strength of soil. *Math. Probl. Eng.* **2021**, *2021*, 1–15. [CrossRef]
37. Varoquaux, G.; Raamana, P.R.; Engemann, D.A.; Hoyos-Idrobo, A.; Schwartz, Y.; Thirion, B. Assessing and tuning brain decoders: Cross-validation, caveats, and guidelines. *NeuroImage* **2017**, *145*, 166–179. [CrossRef] [PubMed]
38. Yang, S.; Wang, Y.; Wang, P.; Mu, J.; Jiao, S.; Zhao, X.; Wang, Z.; Wang, K.; Zhu, Y. Automatic Identification of Landslides Based on Deep Learning. *Appl. Sci.* **2022**, *12*, 8153. [CrossRef]
39. Powers, D.M. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv* **2022**, arXiv: 2010.16061.
40. Everingham, M.; Eslami, S.A.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [CrossRef]
41. Xue, Q.; Lin, H.; Wang, F. FCDM: An Improved Forest Fire Classification and Detection Model Based on YOLOv5. *Forests* **2022**, *13*, 2129. [CrossRef]
42. Henderson, P.; Ferrari, V. End-to-end training of object class detectors for mean average precision. In Proceedings of the 13th Asian Conference on Computer Vision, Taipei, Taiwan, 20–24 November 2016; pp. 198–213.
43. Zaidi, S.S.A.; Ansari, M.S.; Aslam, A.; Kanwal, N.; Asghar, M.; Lee, B. A survey of modern deep learning based object detection models. *Digit. Signal Process.* **2022**, *126*, 103514. [CrossRef]
44. Lin, J.; Lin, H.; Wang, F. A Semi-Supervised Method for Real-Time Forest Fire Detection Algorithm Based on Adaptively Spatial Feature Fusion. *Forests* **2023**, *14*, 361. [CrossRef]
45. Xue, Z.; Lin, H.; Wang, F. A Small Target Forest Fire Detection Model Based on YOLOv5 Improvement. *Forests* **2022**, *13*, 1332. [CrossRef]
46. Zhang, Y.; Guo, Z.; Wu, J.; Tian, Y.; Tang, H.; Guo, X. Real-Time Vehicle Detection Based on Improved YOLO v5. *Sustainability* **2022**, *14*, 12274. [CrossRef]
47. Xue, Z.; Xu, R.; Bai, D.; Lin, H. YOLO-Tea: A Tea Disease Detection Model Improved by YOLOv5. *Forests* **2023**, *14*, 415. [CrossRef]
48. Yao, J.; Qi, J.; Zhang, J.; Shao, H.; Yang, J.; Li, X. A Real-Time Detection Algorithm for Kiwifruit Defects Based on YOLOv5. *Electronics* **2021**, *10*, 1711. [CrossRef]
49. Yuan, C.; Wu, Y.; Qin, X.; Qiao, S.; Pan, Y.; Huang, P.; Liu, D.; Han, N. An effective image classification method for shallow densely connected convolution networks through squeezing and splitting techniques. *Appl. Intell.* **2019**, *49*, 3570–3586. [CrossRef]
50. Huang, G.; Liu, Z.; Van, D.M.L.; Weinberger, K.Q. Densely connected convolutional networks.In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
51. Yang, L.; Zhang, R.Y.; Li, L.; Xie, X. Simam: A simple, parameter-free attention module for convolutional neural networks. In Proceedings of the 2021 38th International Conference on Machine Learning, Virtual, 18–24 July 2021; pp. 11863–11874.
52. You, H.; Lu, Y.; Tang, H. Plant Disease Classification and Adversarial Attack Using SimAM-EfficientNet and GP-MI-FGSM. *Sustainability* **2023**, *15*, 1233. [CrossRef]
53. Gao, D.; Liu, Y.; Hu, B.; Wang, L.; Chen, W.; Chen, Y.; He, T. Time Synchronization based on Cross-Technology Communication for IoT Networks. *IEEE Internet Things J.* **2023**. [CrossRef]
54. James, G.L.; Ansaf, R.B.; Al Samahi, S.S.; Parker, R.D.; Cutler, J.M.; Gachette, R.V.; Ansaf, B.I. An Efficient Wildfire Detection System for AI-Embedded Applications Using Satellite Imagery. *Fire* **2023**, *6*, 169. [CrossRef]

55. Muksimova, S.; Mardieva, S.; Cho, Y.-I. Deep Encoder–Decoder Network-Based Wildfire Segmentation Using Drone Images in Real-Time. *Remote Sens.* **2022**, *14*, 6302. [CrossRef]
56. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
57. Tan, M.; Le, Q.V. EfficientNetV2: Smaller Models and Faster Training. *arXiv* **2021**, arXiv:2104.00298.
58. Namburu, A.; Selvaraj, P.; Mohan, S.; Ragavanantham, S.; Eldin, E.T. Forest Fire Identification in UAV Imagery Using X-MobileNet. *Electronics* **2023**, *12*, 733. [CrossRef]
59. Wei, C.; Xu, J.; Li, Q.; Jiang, S. An Intelligent Wildfire Detection Approach through Cameras Based on Deep Learning. *Sustainability* **2022**, *14*, 15690. [CrossRef]

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.