

Article

A Forest Fire Detection System Based on Ensemble Learning

Renjie Xu ¹, Haifeng Lin ¹, Kangjie Lu ¹, Lin Cao ²  and Yunfei Liu ^{1,*} ¹ College of Information Science and Technology, Nanjing Forestry University, Nanjing 210037, China; xurenjie@njfu.edu.cn (R.X.); haifeng.lin@njfu.edu.cn (H.L.); lukangjie@njfu.edu.cn (K.L.)² Co-Innovation Center for Sustainable Forestry in Southern China, Nanjing Forestry University, Nanjing 210037, China; lincao@njfu.edu.cn

* Correspondence: lyf@njfu.com.cn; Tel.: +86-139-1389-5117

Abstract: Due to the various shapes, textures, and colors of fires, forest fire detection is a challenging task. The traditional image processing method relies heavily on manmade features, which is not universally applicable to all forest scenarios. In order to solve this problem, the deep learning technology is applied to learn and extract features of forest fires adaptively. However, the limited learning and perception ability of individual learners is not sufficient to make them perform well in complex tasks. Furthermore, learners tend to focus too much on local information, namely ground truth, but ignore global information, which may lead to false positives. In this paper, a novel ensemble learning method is proposed to detect forest fires in different scenarios. Firstly, two individual learners Yolov5 and EfficientDet are integrated to accomplish fire detection process. Secondly, another individual learner EfficientNet is responsible for learning global information to avoid false positives. Finally, detection results are made based on the decisions of three learners. Experiments on our dataset show that the proposed method improves detection performance by 2.5% to 10.9%, and decreases false positives by 51.3%, without any extra latency.



Citation: Xu, R.; Lin, H.; Lu, K.; Cao, L.; Liu, Y. A Forest Fire Detection System Based on Ensemble Learning. *Forests* **2021**, *12*, 217. <https://doi.org/10.3390/f12020217>

Academic Editor: Stelian Alexandru Borz

Received: 4 January 2021

Accepted: 12 February 2021

Published: 13 February 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 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/>).

Keywords: forest fire detection; deep learning; ensemble learning; Yolov5; EfficientDet; EfficientNet

1. Introduction

With the change of the earth's climate, forest fires occur frequently all over the world, which not only cause serious economic losses and destroy the ecological environment, but also pose a great threat to the safety of human life.

Forest fires usually spread quickly and are difficult to control in a short time. Therefore, it is imperative to detect the early forest fire before it spreads out, but traditional detection methods have obvious drawbacks in detecting it in open forest areas. Sensors-based [1–3] detection systems have good performance in indoor space, but it is difficult to install them outdoors, considering high coverage cost [4,5]. In addition, they cannot provide important visual information which can help firefighters promptly grasp the situation of the fire scene. Infrared or ultraviolet detectors [6,7] are easy to be interfered by the environment, and considering their short detection distance, they are not suitable for large open areas. Satellite remote sensing [8] is good at detecting large-scale forest fires, but it cannot detect early regional fire.

Impressed by the rising computer vision technology, researchers start to seek an efficient and effective fire detection model based on image processing. Chen et al. [9] proposed an RGB (red, green, blue) model based chromatic and disorder measurement for extracting fire-pixels in the video. The color information is responsible for extracting fire-pixels, and dynamic information is used to verify if it is a real fire. Töreyin et al. [10] used 1D temporal wavelet transform to detect flame flicker, and applied 2D spatial wavelet transform to identify fire moving regions. This method, which integrated color and temporal variation information, reduced false alarms in real-world scenes. Çelik et al. [11] studied diverse video sequences and images, and proposed a fuzzy color model using statistical analysis.

Combined with motion analysis, the model achieves a good discrimination between fire and fire-like objects. Teng et al. [12] analyzed fire characteristics and proposed a real-time fire detection method based on hidden Markov models (HMMs), which extracted candidate fire-pixels using moving pixel detection, fire-color inspection, and pixel clustering. Chino et al. [13] found that most algorithms were designed for video, which had obvious limitations. To solve this problem, a novel fire detection method named BowFire was proposed. The method combined color features with superpixel texture discrimination to detect fire in still images. In conclusion, most traditional fire detection methods based on image processing focused on creating artificial features like color, motion, and texture to detect fires.

However, powerful deep learners begin to replace human intelligence. They are better at learning features than humans, and the features they extract contain much deeper semantic information than manmade ones. Recently, deep learning has outperformed traditional manmade features in many fields, and have been widely used in fire detection. Zhang et al. [14] created a forest fire benchmark, and used Faster R-CNN (region-based convolutional neural network) [15], Yolo (you only look once) [16–19], and SSD (single shot multibox detector) [20] to detect fire. They found that SSD was better regarding efficiency, detection accuracy, and early fire detection ability. Moreover, they proposed an improved tiny-Yolo by adjusting the network architecture. Kim et al. [21] employed faster R-CNN to detect fire and non-fire regions based on their spatial features. In addition, long short-term memory (LSTM) is used to verify the reliability of fire alarm. Lee et al. [22] proposed a video-based fire detection model, which used faster R-CNN to generate a fire candidate region for each frame. Then, structural similarity (SSIM) and mean square error (MSE) were calculated to determine similarity between adjacent frames. Final fire regions were determined based on spatial and temporal features. Pan et al. [23] proposed a camera-based wildfire detection system via transfer learning, in which block-based analysis strategy was used to improve fire detection accuracy. Redundant filters, which had low energy impulse response, were removed to ensure the model's efficiency on edge devices. Wu et al. [24] applied principal component analysis (PCA) to process forest fire images, and then fed them into the training network. The combination of two models was proved to enhance location results. In conclusion, faced with fire detection task, most researchers tend to only assign individual learners to perform object detection tasks, which is considered unreliable, since it may lead to false negatives.

In this paper, a novel method based on ensemble learning for forest fire detection is proposed. First, forest fire detection is a complicated and difficult task, making it highly impractical for individual learners to detect fires in diverse scenarios. Every individual learner has its own expertise, and can extract different features from the image, so integrating different individual learners can significantly improve the robustness of the model and enhance detection performance. Therefore, two individual object detectors Yolov5 [25] and EfficientDet [26] are integrated to detect the fire in parallel. These two learners work synergistically in detecting different types of forest fires, thereby improving the detection accuracy. Second, the object detectors only care about what fire is like, so they do not take the whole image into consideration. In this case, fire-like objects will absolutely affect the detection results. To solve this problem, the EfficientNet image classifier [27] is incorporated into our model, whose role is to enable the model to take full advantage of the global information. Final detection results will be made through the decision strategy according to results of these three learners, which will efficiently increase detection accuracy and decrease the false positives.

2. Materials and Methods

2.1. Datasets

To ensure our learners can handle different kinds of forest fires (ground fire, trunk fire, and canopy fire), we collected images from multiple public fire datasets: BowFire [28], FD-dataset [29], ForestryImages [30], VisiFire [31], etc. After manual filtration, we created

a single integrated forest fire dataset containing 10,581 images, with 2976 forest fire images and 7605 non-fire images. Representative samples of our dataset are shown in Figures 1–3.

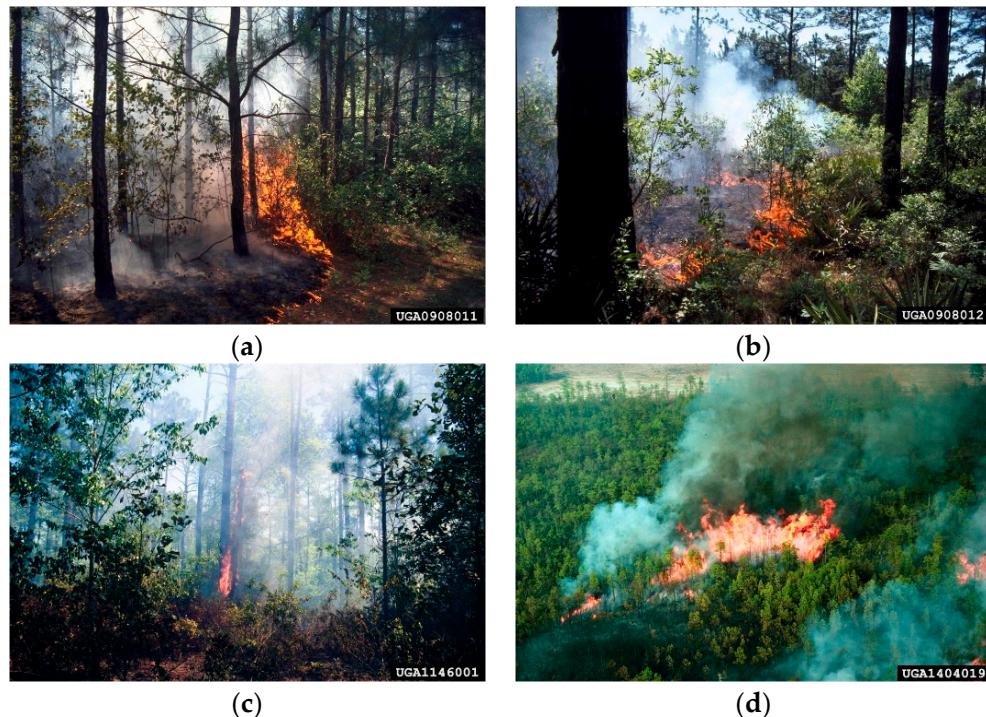


Figure 1. Representative forest fire images in the fire section of our dataset, including (a) ground fire 1, (b) ground fire 2, (c) trunk fire, and (d) canopy fire.

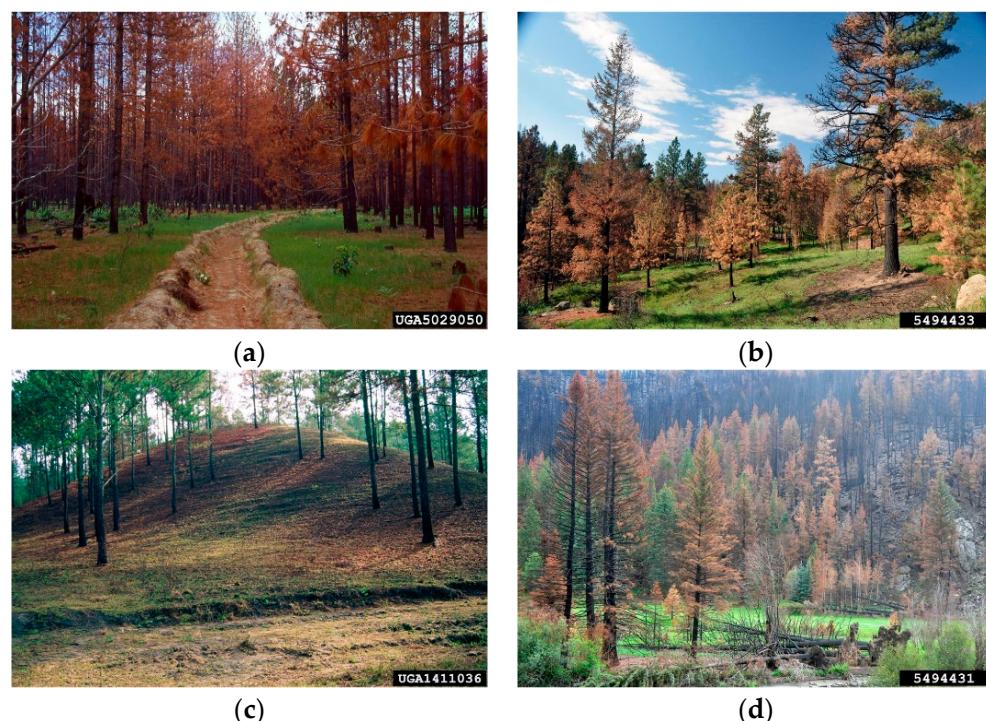


Figure 2. Representative normal forest images in the non-fire section of our dataset, including (a) normal forest scene 1, (b) normal forest scene 2, (c) normal forest scene 3, and (d) normal forest scene 4. (a–d) illustrate normal forest scenes without fire objects.

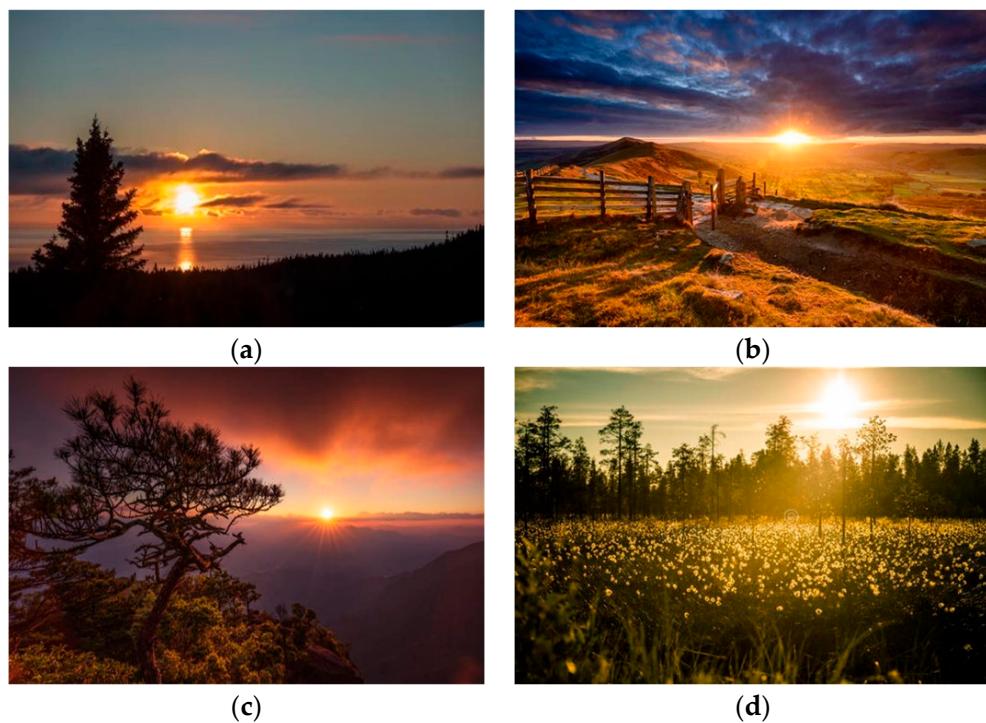


Figure 3. Representative images in the non-fire section of our dataset, including (a) wild scene with sun 1, (b) wild scene with sun 2, (c) wild scene with sun 3, and (d) wild scene with sun 4. (a–d) illustrate normal wild scenes containing fire-like object (e.g., sun).

2.2. Yolov5

Yolo is a state-of-the-art, real-time object detector, and Yolov5 is based on Yolov1-Yolov4. Continuous improvements have made it achieve top performances on two official object detection datasets: Pascal VOC (visual object classes) [32] and Microsoft COCO (common objects in context) [33].

The network architecture of Yolov5 is shown in Figure 4. There are three reasons why we choose Yolov5 as our first learner. Firstly, Yolov5 incorporated cross stage partial network (CSPNet) [34] into Darknet, creating CSPDarknet as its backbone. CSPNet solves the problems of repeated gradient information in large-scale backbones, and integrates the gradient changes into the feature map, thereby decreasing the parameters and FLOPS (floating-point operations per second) of model, which not only ensures the inference speed and accuracy, but also reduces the model size. In forest fire detection task, detection speed and accuracy is imperative, and compact model size also determines its inference efficiency on resource-poor edge devices. Secondly, the Yolov5 applied path aggregation network (PANet) [35] as its neck to boost information flow. PANet adopts a new feature pyramid network (FPN) structure with enhanced bottom-up path, which improves the propagation of low-level features. At the same time, adaptive feature pooling, which links feature grid and all feature levels, is used to make useful information in each feature level propagate directly to following subnetwork. PANet improves the utilization of accurate localization signals in lower layers, which can obviously enhance the location accuracy of the object. Thirdly, the head of Yolov5, namely the Yolo layer, generates 3 different sizes (18×18 , 36×36 , 72×72) of feature maps to achieve multi-scale [18] prediction, enabling the model to handle small, medium, and big objects. A forest fire usually develops from small-scale fire (ground fire) to medium-scale fire (trunk fire), then to big-scale fire (canopy fire). Multi-scale detection ensures that the model can follow size changes in the process of fire evolution.

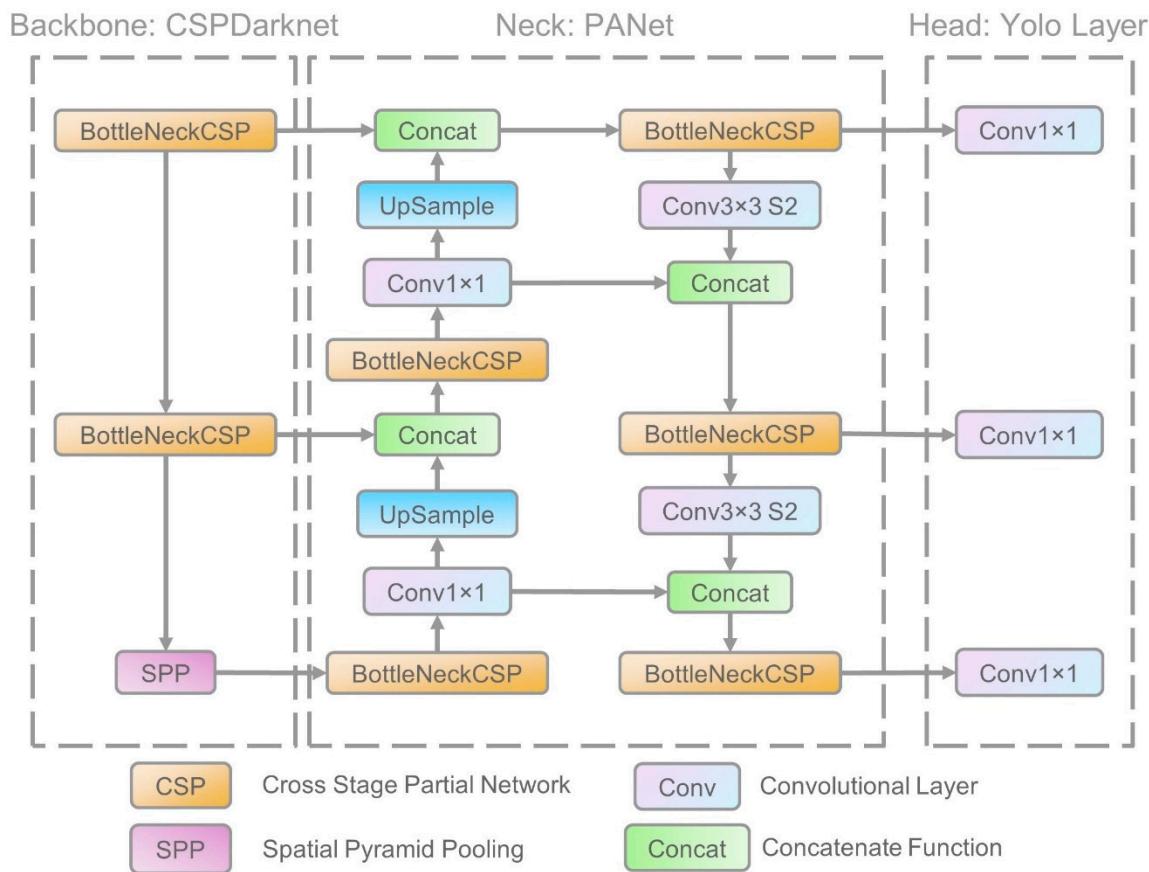


Figure 4. The network architecture of Yolov5. It consists of three parts: (1) Backbone: CSPDarknet, (2) Neck: PANet, and (3) Head: Yolo Layer. The data are first input to CSPDarknet for feature extraction, and then fed to PANet for feature fusion. Finally, Yolo Layer outputs detection results (class, score, location, size).

2.3. EfficientDet

EfficientDet is a new family of object detectors developed by Google, and it consistently achieves better efficiency than prior art across a wide spectrum of resource constraints. Similar to Yolov5, EfficientDet has also achieved remarkable performances in Pascal VOC and Microsoft COCO tasks, and is widely used in real-world applications.

The network architecture of EfficientDet is shown in Figure 5. There are three reasons why we choose EfficientDet as our second learner. Firstly, EfficientDet employed state-of-the-art network EfficientNet [27] as its backbone, making that the model has sufficient ability to learn the complex feature of diverse forest fires. Secondly, it applied an improved PANet, named bi-directional feature pyramid network (Bi-FPN) as its neck, to allow easy and fast multi-scale feature fusion. Bi-FPN introduces learnable weights, enabling network to learn the importance of different input features, and repeatedly applies top-down and bottom-up multi-scale feature fusion. Compared with Yolov5's neck PANet, Bi-FPN has better performances with less parameters and FLOPS. Meanwhile, different feature fusion strategy brings different semantic information, thereby bringing different detection results. Thirdly, similar to EfficientNet, it integrates a compound scaling method that uniformly scales the resolution, depth, and width for all backbone, feature network, and box/class prediction networks at the same time, which ensures the maximum accuracy and efficiency under the limited computing resources. With more available resources, accuracy will be consistently improved. Our second learner, EfficientDet, with different backbone, neck, and head, can learn different information that Yolov5 cannot.

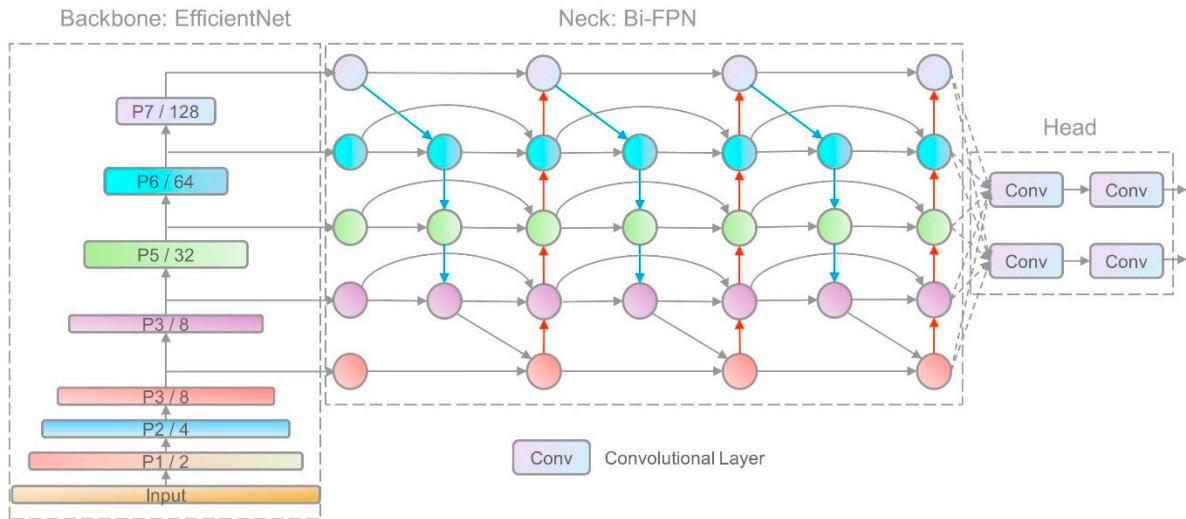


Figure 5. The network architecture of EfficientDet. It consists of three parts: (1) Backbone: EfficientNet, (2) Neck: Bi-FPN, (3) Head. Similar to Yolov5, the data are first input to EfficientNet for feature extraction, and then fed to Bi-FPN for feature fusion. Finally, head outputs detection results (class, score, location, size).

2.4. EfficientNet

EfficientNet is a new efficient network proposed by Google. It applied a novel model scaling strategy, namely compound scaling method, to balance network depth, network width, and image resolution for better accuracy at a fixed resource budget. With this, EfficientNet outperformed other hot networks like ResNet [36], DenseNet [37], ResNeXt [38] with the highest Top-1 accuracy in ImageNet image classification task.

The network architecture of EfficientNet is shown in Figure 6. The reason why we choose EfficientNet as our third learner is that it achieves a superior trade-off between accuracy and efficiency. In our model, the third learner plays the most important role. It is responsible for learning the whole image to guide the detection, meaning that its decisions directly determine the final results. Meanwhile, it must be highly efficient, otherwise it will slow down the speed of the entire model.

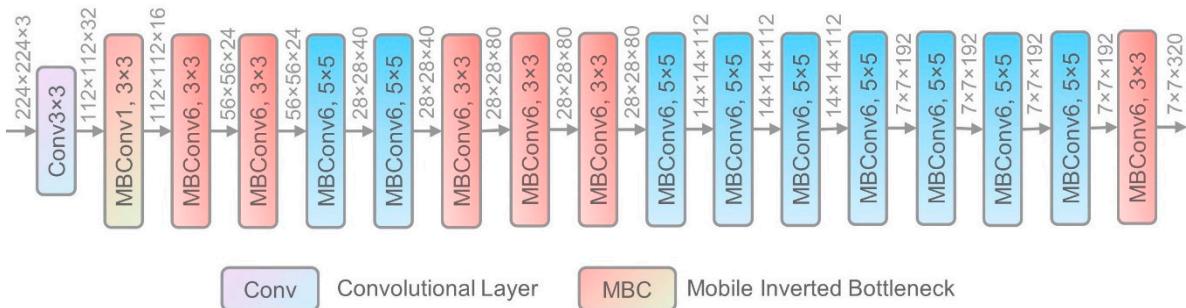


Figure 6. The network architecture of EfficientNet. It can output a feature map with deep semantic information after the input data flows through the multi-layer network.

2.5. Our Model

In real-world forest fire detection task, we need to handle different types of forest fires like ground fire, trunk fire, canopy fire. These fires, influenced by the environment, are diverse in shape, texture, or even color, bringing great difficulty for individual learner to extract effective features. By careful observations, we find that Yolov5 is better at learning long-area fires (Figure 7), but it sometimes misses objects (Figure 8). Meanwhile, even though EfficientDet is not sensitive to long-area fires (Figure 7), it is more careful than Yolov5, meaning that EfficientDet can make a complementary detection (Figure 8). There-

fore, we consider that integrating these two efficient learners with different specialties to make detection together can improve detection accuracy.

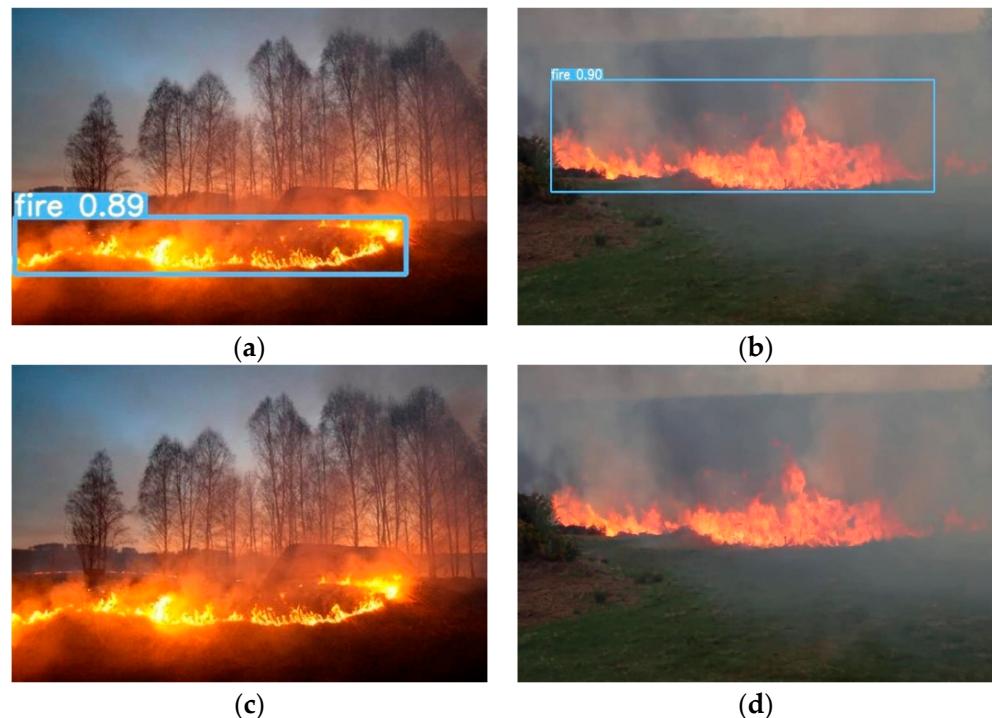


Figure 7. Yolov5 is better at detecting long-area fires than EfficientDet. (a) True positive of Yolov5; (b) true positive of Yolov5; (c) false negative of EfficientDet; (d) false negative of EfficientDet. (a,b) illustrate that Yolov5 detect long-area fires successfully, while (c,d) show that EfficientDet fails to detect them.

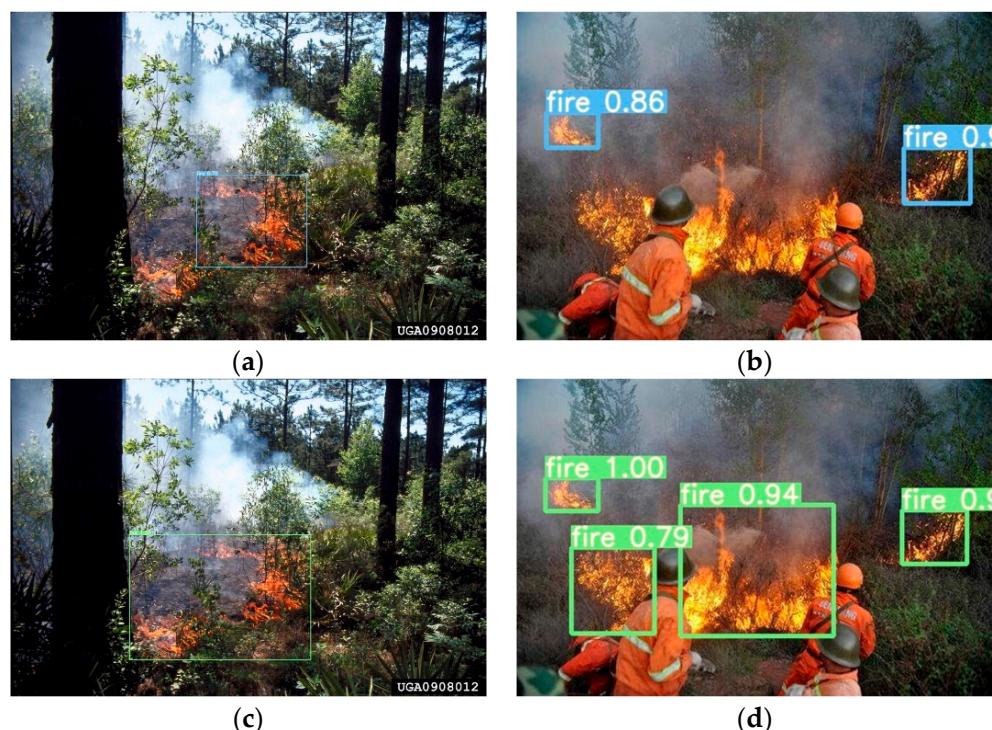


Figure 8. EfficientDet is a more careful object detector than Yolov5, meaning that it seldom losses potential objects easily. (a) Yolov5 fails to cover all fire areas; (b) Yolov5 misses two fire objects; (c) EfficientDet covers all fire areas; (d) EfficientDet detects four fire objects.

Another issue is that the ability of the object detector is limited. It only learns the fire region, which is just a local pattern of the whole image, but ignores the other information like background. As a result, the object detector may treat fire-like objects (e.g., sun) as fires (Figure 9), thereby making false alarms. Therefore, a good leader EfficientNet that has a full understanding of the whole image is needed to guide the detection process.

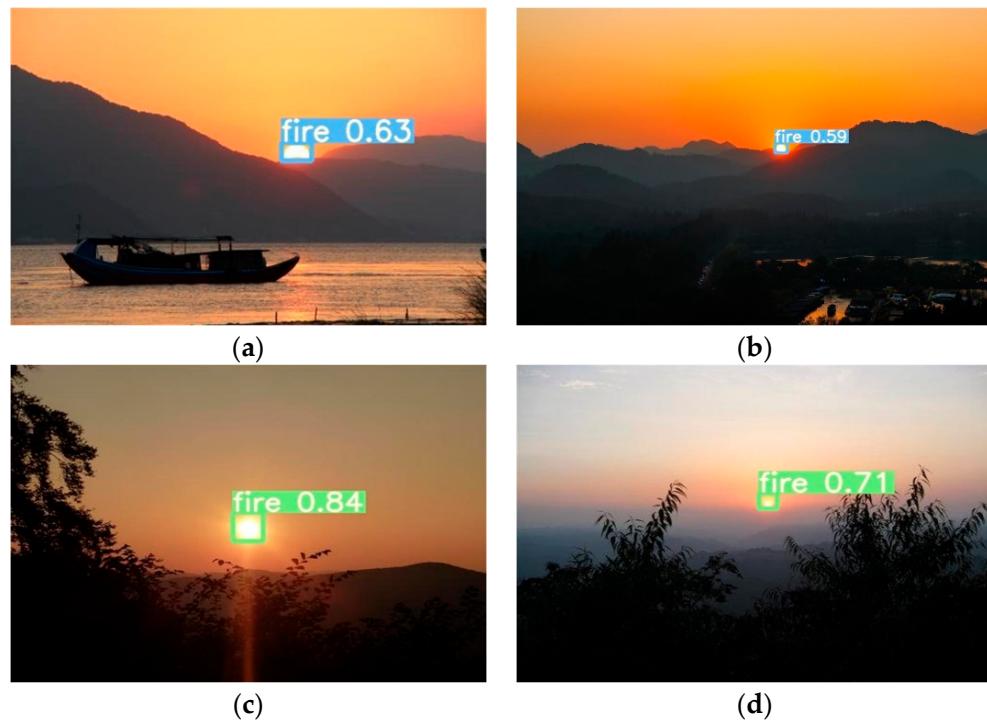


Figure 9. Object detectors Yolov5 and EfficientDet are easy to be deceived by fire-like objects (e.g., sun). (a) False positive of Yolov5 (confidence score: 0.63); (b) false positive of Yolov5 (confidence score: 0.59); (c) false positive of EfficientDet (confidence score: 0.84); (d) false positive of EfficientDet (confidence score: 0.71).

To address the above two issues and make sure our model is robust to diverse scenarios, three deep learners are integrated to make decisions together (Figure 10). The first and second learners Yolov5 and EfficientDet act as object detectors, to detect fire locations in images by generating candidate boxes, respectively. Then, the non-maximum suppression algorithm [39] (Algorithm 1) is employed to eliminate redundant boxes, preserving boxes with top confidence. The third learner EfficientNet acts as a binary classifier, responsible for learning the whole image to determine whether the image contains fire objects. Finally, the object detection results, and image classification results are sent into a decision strategy module, in which if the image is considered to contain fire objects, retaining object detection results, otherwise ignoring them.

In addition, integrating multiple learners will not affect the overall efficiency of model, because the three learners are structurally independent, and the whole model is executed by multi processes, meaning that each learner has a separate process responsible for it.

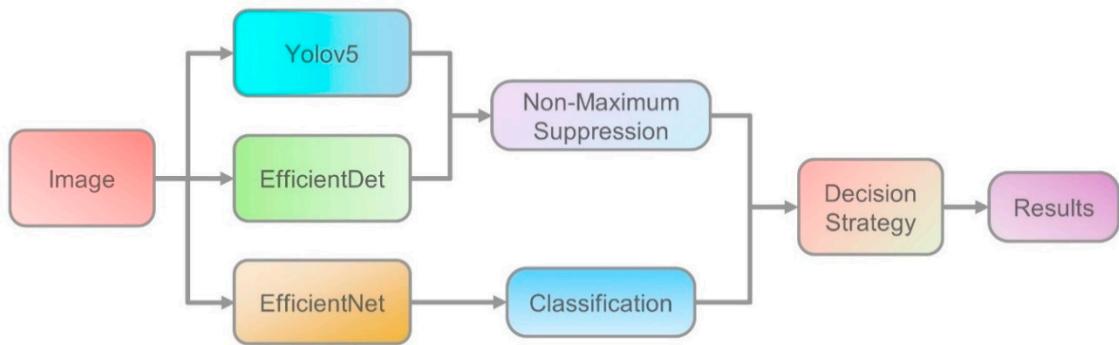


Figure 10. Structure of the proposed model in this paper. Three deep learners are ensembled in parallel. Two object detectors Yolov5 and EfficientDet are integrated to perform object detection task, and the classifier EfficientNet is in charge of discriminating whether the image contains fire objects. Final detection results are made based on the decisions of three learners.

Algorithm 1. Non-Maximum Suppression (NMS)

INPUT: $B = \{b_1, \dots, b_N\}, S = \{s_1, \dots, s_N\}, N_t$
 B is the list of initial detection boxes
 S contains corresponding detection scores
 N_t is the NMS threshold

Begin:
 $D \leftarrow \{\}$
while $B \neq \text{empty}$ **do**
 $m \leftarrow \text{argmax } S$
 $M \leftarrow b_m$
 $D \leftarrow D \cup M; B \leftarrow B - M$
 for b_i in B **do**
 if $\text{iou}(M, b_i) \geq N_t$ **then**
 $B \leftarrow B - b_i; S \leftarrow S - s_i$
 end
 end
 end
 Return D, S

End

2.6. Model Evaluation

We evaluate models using Microsoft COCO criteria (Table 1), which is widely used in object detection tasks. However, fire is a special object, which is diverse in shape, texture, and color. Bounding box generated by object detectors may slightly differ from ground truth (Figure 11), thereby influencing the calculation of average precision, but detectors do identify the fire areas successfully. Therefore, to evaluate models more comprehensively, we introduce two additional evaluation metrics, namely frame accuracy (FA) and false positive rate (FPR). For one image, if the detector misses any fire object, we call it a frame false (FF), otherwise frame true (FT). If the detector treats any fire-like object as fire, we call it a false positive (FP), otherwise true positive (TP). Note that FA is calculated on the test set containing 476 forest images, and FPR is calculated on our challenging non-fire dataset containing 641 images with fire-like objects (e.g., sun). The FA and FPR can be calculated as Equation (1) and Equation (2), respectively:

$$\text{FA} = \frac{\text{FT}}{\text{FT} + \text{FF}} \times 100, \quad (1)$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TP}} \times 100. \quad (2)$$

Table 1. Microsoft COCO criteria—commonly used in object detection task for evaluating the model precision and recall across multiple scales.

Average Precision (AP)		AP at IoU = 0.5
AP _{0.5}		
AP Across Scales:		
AP _S	AP _{0.5} for small objects: area < 32^2	
AP _M	AP _{0.5} for medium objects: 32^2 < area < 96^2	
AP _L	AP _{0.5} for big objects: area > 96^2	
Average Recall (AR)		AR at IoU = 0.5
AR _{0.5}		
AR Across Scales:		
AR _S	AR _{0.5} for small objects: area < 32^2	
AR _M	AR _{0.5} for medium objects: 32^2 < area < 96^2	
AR _L	AR _{0.5} for big objects: area > 96^2	

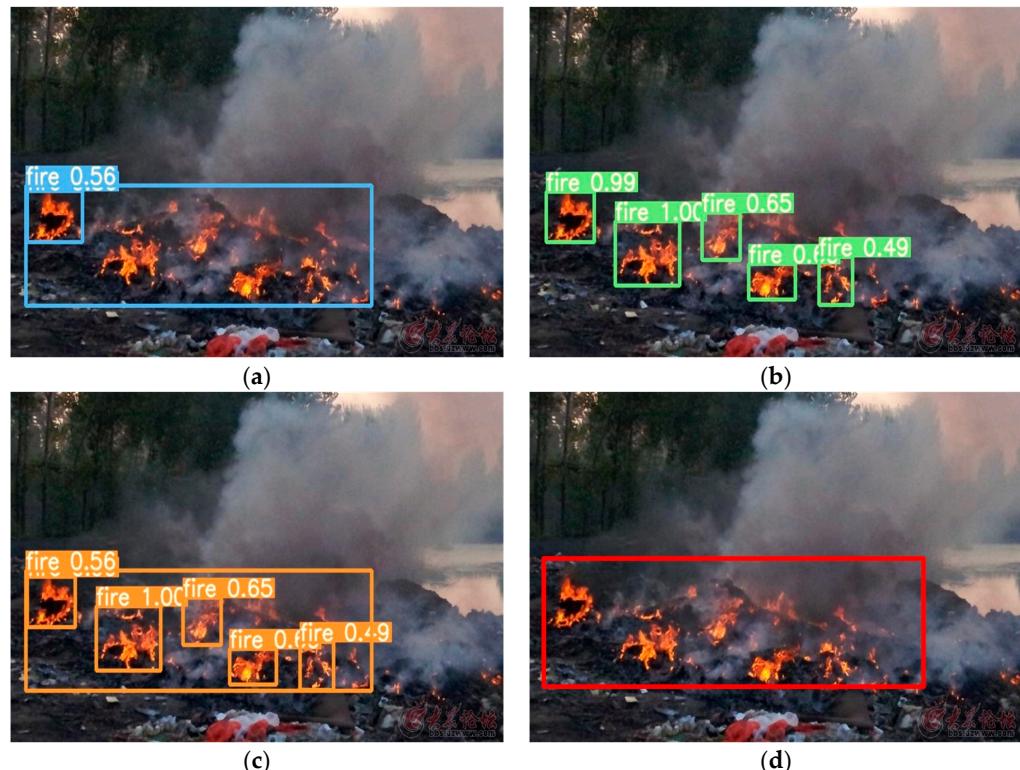


Figure 11. Bounding boxes generated by (a) Yolov5, (b) EfficientDet, and (c) our model (3 learners) are different from (d) ground truth, but still has good detection performance.

3. Results

3.1. Training

We applied different strategies to train our three learners: Yolov5, EfficientDet, and EfficientNet. Object detectors, namely Yolov5 and EfficientDet, are trained with 2381 forest fire images, and tested with 476 forest fire images. The image classifier, namely EfficientNet, is trained with 2381 forest fire images and 5804 non-fire images, and tested with 476 forest fire images and 1160 non-fire images. Note that non-fire images contain normal images, and images with fire-like objects (e.g., sun). Each model is built up by Pytorch [40] and trained on NVIDIA GTX 2080TI. The details of our training strategy are shown in Table 2.

Table 2. Detailed training strategies of models.

Model	Train	Test	Optimizer	LR	Batch Size	Epoch
Yolov5	2381	476	SGD [41,42]	1×10^{-2}	8	300
EfficientDet	2381	476	AdamW [43]	1×10^{-4}	4	300
EfficientNet	8185	1636	SGD	1×10^{-2}	8	300

LR: learning rate, SGD: stochastic gradient descent, AdamW: Adam with decoupled weight decay.

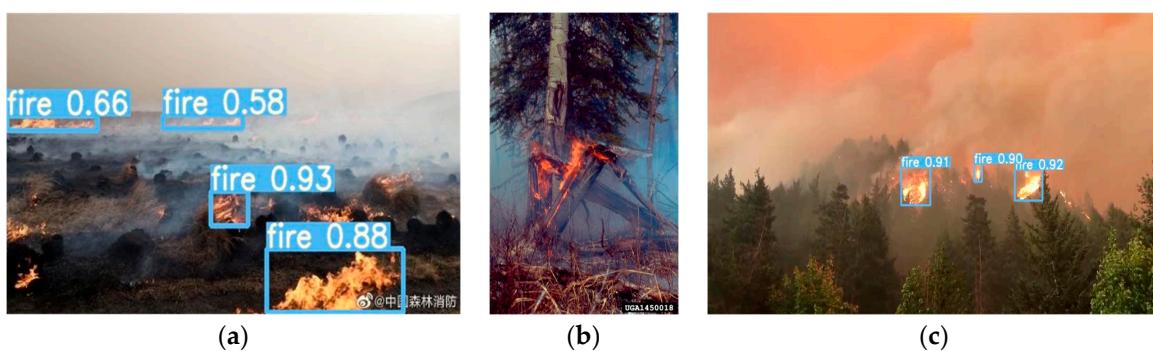
3.2. Comparison

We compare our model with typical one-stage object detectors. As is shown in Table 3, even though Yolov5 and EfficientDet are the most powerful detectors in this task, the high false positive rate and missing detections cannot be ignored. By integrating them (2 learners), all evaluation metrics are significantly improved, but the false positive rate is increased to 51.6%, since the false positives come from both Yolov5 and EfficientDet. Under the guide of our third learner EfficientNet, the false positive rate is reduced to 0.3%. What is also worth mentioning is that, after introducing the third learner, some metrics are slightly decreased. It is because that EfficientNet wrongly treats some fire images as non-fire ones, and then ignores the object detection results, but we consider it is worthwhile to sacrifice a tiny decrease in average precision and recall for substantial improvement in the false positive rate. To sum up, our model (3 learners) is superior in AP_{0.5}, AP_S, AP_M, AP_L, AR_{0.5}, AR_S, AR_M, AR_L, FPR, and FA compared with other typical object detectors. Comprehensive improvements make the model have better performance in detecting different types of forest fires: small-scale fires, medium-scale fires, big-scale fires, ground fires, trunk fires, canopy fires, and fires at night (Figures 12 and 13). Faced with fire-like objects (e.g., sun), our model will not be interfered. (Figure 14).

Table 3. Experiments on our dataset—evaluating models using Microsoft COCO criteria, FPR, FA, and latency.

Model	AP _{0.5}	AP _S	AP _M	AP _L	AR _{0.5}	AR _S	AR _M	AR _L	FPR	FA	Latency (ms)
SSD	66.8	37.8	42.4	78.6	70.1	39.1	45.7	82.7	45.6	92.6	88.8
Yolov3	66.4	26.0	44.6	78.1	71.1	26.1	52.5	82.5	22.9	88.0	15.6
Yolov3-SPP	68.3	56.3	49.9	76.7	73.9	60.9	56.6	81.9	30.7	93.3	15.6
Yolov4	69.6	53.7	48.9	78.4	75.5	60.9	57.5	83.9	61.9	94.1	20.5
Yolov5	70.5	51.9	53.7	79.2	75.6	56.5	61.2	83.0	22.6	94.7	28.0
EfficientDet	75.7	63.7	58.5	83.0	79.2	65.2	63.9	86.5	41.8	95.5	65.6
Ours (2 learners)	79.7	72.2	65.6	85.5	84.1	76.1	73.1	89.3	51.6	99.4	66.8
Ours (3 learners)	79.0	72.2	64.9	84.7	83.8	76.1	72.6	88.9	0.3	98.9	66.8

Note that AP_{0.5}, AP_S, AP_M, AP_L, AR_{0.5}, AR_S, AR_M, AR_L, FPR, and FA are all percentages. The best figure of each metric are highlighted in bold.

**Figure 12. Cont.**

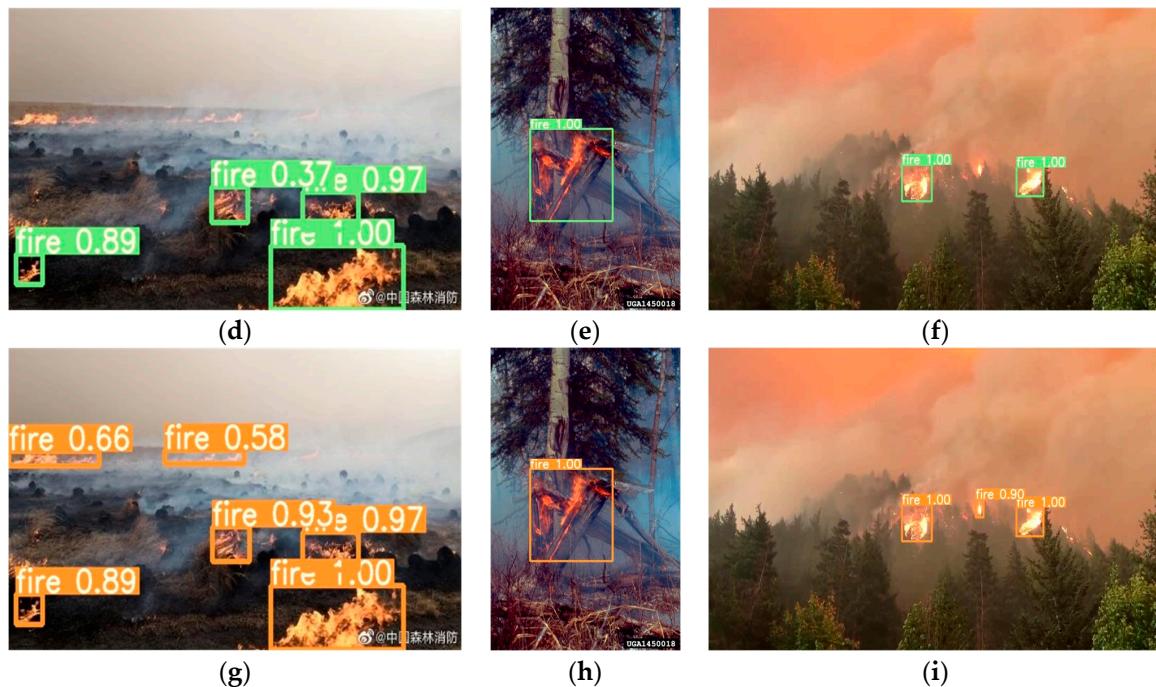


Figure 12. Our ensemble model (3 learners) has better performance on ground fires, trunk fires, and canopy fires. (a) Four ground fires detected by Yolov5; (b) Yolov5 fails to detect the trunk fire; (c) three canopy fires detected by Yolov5; (d) four ground fires detected by EfficientDet; (e) the trunk fire detected by EfficientDet; (f) two canopy fires detected by EfficientDet; (g) six ground fires detected by our model; (h) the trunk fire detected by our model; (i) three canopy fires detected by our model.

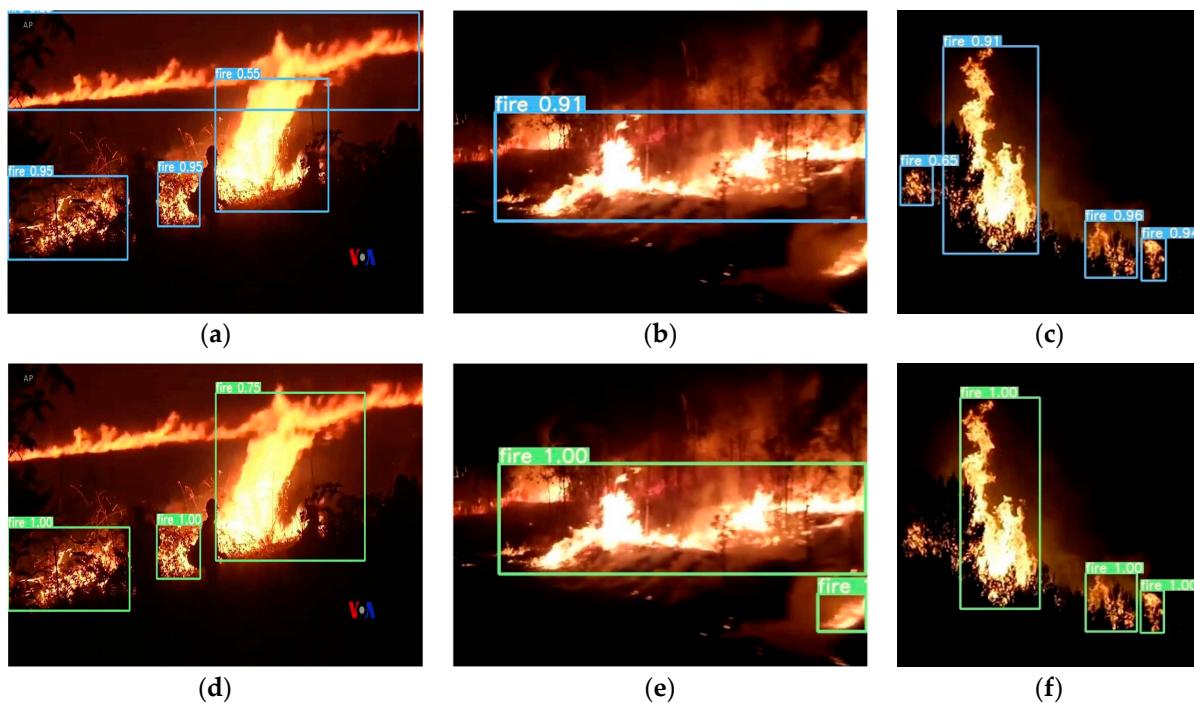


Figure 13. Cont.

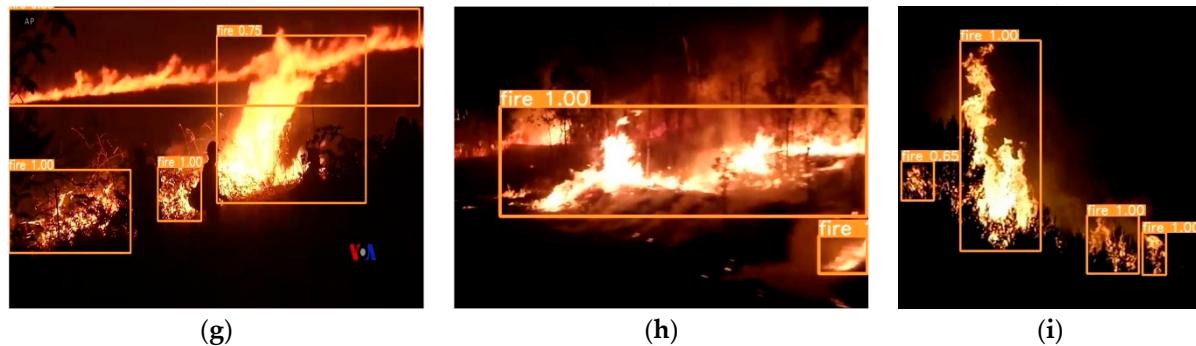


Figure 13. Our improved model has better performance on small-scale, medium-scale, and big-scale fires at night. (a) Medium-scale and big-scale fires detected by Yolov5; (b) medium-scale and big scale fires detected by Yolov5; (c) small-scale, medium-scale and big-scale fires detected by Yolov5; (d) medium-scale and big-scale fires detected by EfficientDet; (e) medium-scale and big scale fires detected by EfficientDet; (f) small-scale, medium-scale, and big-scale fires detected by EfficientDet; (g) medium-scale and big-scale fires detected by our model; (h) medium-scale and big scale fires detected by our model; (i) small-scale, medium-scale, and big-scale fires detected by our model.



Figure 14. Under the guide of EfficientNet, our ensemble model has a good discriminability between fire and fire-like objects (e.g., sun). (a) True negative of Yolov5; (b) false positive of Yolov5 (confidence score: 0.59); (c) false positive of EfficientDet (confidence score 0.71); (d) true negative of EfficientDet; (e) true negative of our model; (f) true negative of our model.

4. Discussion

Compared with other common objects that have fixed form, forest fire is a dynamic object [44]. In the real-world scenario, a forest fire usually starts from small-scale fire, develops to medium-scale fire, and then to big-scale fire [45]. In terms of types, it starts from ground fire, then spreads to the trunk, and finally to the canopy [46]. The various shapes, sizes, textures, and colors of forest fires make the fire evolution a complex process, and bring great difficulty in fire detection.

Therefore, it is highly imperative for detectors to be sensitive to different types of fires. Through careful experimental comparisons, we find that no single detector that can handle all kinds of fires. They have respective advantages and disadvantages. Yolov5 is excellent at detecting long-area fires (Figure 7), but it frequently misses objects (Figure 8). EfficientDet is a more careful detector, compared to Yolov5; even though it has a bad performance on long-area fires (Figure 7), it can detect fires that Yolov5 cannot (Figure 8), meaning that it is a good partner for Yolov5. Our model, which efficiently integrates decisions of these two powerful learners, boost detection performance by 2.5–10.9%, in terms of $AP_{0.5}$, AP_S , AP_M , AP_L , $AR_{0.5}$, AR_S , AR_M , AR_L . The significant improvements of average precision and average recall for small, medium, and big objects make the model more sensitive to the size changes of fires, thereby enhancing detection performance on different types of forest fires: ground fire, trunk fire, canopy fire, and fires at night in the fire evolution (Figures 12 and 13).

Another problem is that the false positive rate of the improved model (2 learners) becomes higher: 22.6% to 51.6% since the model also integrates wrong detection results from both learners. To address this issue, we use 8185 images containing 2381 forest fire images and 5804 non-fire images (containing fire-like images and normal forest images) to train our third learner EfficientNet. Sufficient training sets enabled EfficientNet to show a good discriminability between fire objects and fire-like objects, with 99.6% accuracy on 476 fire images, and 99.7% accuracy on 676 fire-like images. With the help of the leader learner EfficientNet, wrong detection results are eliminated, and the false positive rate is significantly decreased to 0.3% (Figure 14). Noticeably, the join of EfficientNet reduces $AP_{0.5}$, AP_M , AP_L , $AR_{0.5}$, AR_M , AR_L by roughly 1%, which is because that EfficientNet wrongly ignores 2 fire images containing medium-scale and big-scale fire objects.

In terms of latency, the Yolo family is superior compared to EfficientDet and SSD. Excellent inference speed makes Yolo family widely used in real-world applications, but experimental results show that they are not able to have a satisfactory performance on forest fire detection tasks. The latency of EfficientDet is 65.6 ms, which is over twice that of Yolov5 (28.0 ms), but EfficientDet outperforms Yolov5 by over 5% regarding detection performance. We ensemble these three learners Yolov5 (28.0 ms), EfficientDet (65.6 ms), EfficientNet (31.3 ms) in parallel to make sure that our model can achieve the best performance without any extra latency. The final latency of our model (3 learners) is 66.8 ms, which shows that an excellent trade-off between detection performance and efficiency has been achieved, and the model is applicable for real-time detection task.

For further improvement, we plan to study the labeling strategy for forest fires, since the quality of training data directly determines the detection performance. Another interesting extension is to investigate the network architecture of backbones, and modify them to make sure that they are specially designed for forest fire detection task. Additionally, we will work on developing a forest fire tracking system, which can classify different types of forest fires: ground fire, trunk fire and canopy fire, to track the evolution and spread of forest fires.

5. Conclusions

The successful application of convolutional neural networks significantly improves the performance of object detection. However, forest fire is a dynamic object with no fixed form, which the individual object detector cannot handle. In addition, object detectors are easy to be deceived by fire-like objects and generate false positives due to their limited

visual field. To address these problems, a novel ensemble learning method for real-time forest fire detection is proposed in this paper. Two powerful object detectors (Yolov5 and EfficientDet) with different expertise are integrated to make the whole model more robust to diverse forest fire scenarios. Then, a leader (EfficientNet) is introduced to guide the detection process to reduce false positives. Experimental results show that, compared with other popular object detectors, our model achieves a superior trade-off among average precision, average recall, false positive rate, frame accuracy, and latency. The significant improvements make it possible for the model to perform well in real-world forestry applications.

Author Contributions: R.X. devised the programs and drafted the initial manuscript. H.L. and K.L. helped with data collection, data analysis, and figures and tables. L.C. contributed to fund acquisition and writing embellishment. Y.L. designed the project and revised the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key R&D Program of China (grant number 2017YFD0600904) and the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD).

Data Availability Statement: Publicly available datasets were analyzed in this study. The data can be found here: BowFire [28], FD-dataset [29], ForestryImages [30], VisiFire [31].

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

References

1. Zhang, J.; Li, W.; Yin, Z.; Liu, S.; Guo, X. Forest fire detection system based on wireless sensor network. In Proceedings of the 4th IEEE Conference on Industrial Electronics and Applications (ICIEA 2009), Xi'an, China, 25–27 May 2009; pp. 520–523.
2. Yu, L.; Wang, N.; Meng, X. Real-time forest fire detection with wireless sensor networks. In Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2005), Wuhan, China, 26 September 2005; pp. 1214–1217.
3. Chen, S.J.; Hovde, D.C.; Peterson, K.A.; Marshall, A.W. Fire detection using smoke and gas sensors. *Fire Saf. J.* **2007**, *42*, 507–515. [[CrossRef](#)]
4. Zhang, F.; Zhao, P.; Xu, S.; Wu, Y.; Yang, X.; Zhang, Y. Integrating multiple factors to optimize watchtower deployment for wildfire detection. *Sci. Total Environ.* **2020**, *737*, 139561. [[CrossRef](#)] [[PubMed](#)]
5. Zhang, F.; Zhao, P.; Thiagarajam, J.; Kirubarajan, T. Terrain-influenced incremental watchtower expansion for wildfire detection. *Sci. Total Environ.* **2018**, *654*, 164–176. [[CrossRef](#)] [[PubMed](#)]
6. Lee, B.; Kwon, O.; Jung, C.; Park, S. The development of UV/IR combination flame detector. *J. KIIS* **2001**, *16*, 1–8.
7. Kang, D.; Kim, E.; Moon, P.; Sin, W.; Kang, M. Design and analysis of flame signal detection with the combination of UV/IR sensors. *J. Korean Soc. Int. Inf.* **2013**, *14*, 45–51. [[CrossRef](#)]
8. Fernandes, A.M.; Utkin, A.B.; Lavrov, A.V.; Vilar, R.M. Development of neural network committee machines for automatic forest fire detection using lidar. *Pattern Recognit.* **2004**, *37*, 2039–2047. [[CrossRef](#)]
9. Chen, T.H.; Wu, P.H.; Chiou, Y.C. An early fire-detection method based on image processing. In Proceedings of the IEEE International Conference on Image Processing (ICIP 2004), Singapore, 24–27 October 2004; pp. 1707–1710.
10. Töreyin, B.U.; Dedeoğlu, Y.; Güdükbay, U.; Cetin, A.E. Computer vision based method for real-time fire and flame detection. *Pattern Recognit. Lett.* **2006**, *27*, 49–58. [[CrossRef](#)]
11. Çelik, T.; Özkaramanlı, H.; Demirel, H. Fire and smoke detection without sensors: Image processing based approach. In Proceedings of the IEEE 15th European Signal Processing Conference (EUSIPCO 2007), Poznan, Poland, 3–7 September 2007; pp. 1794–1798.
12. Teng, Z.; Kim, J.H.; Kang, D.J. Fire detection based on hidden Markov models. *Int. J. Control Autom. Syst.* **2010**, *8*, 822–830. [[CrossRef](#)]
13. 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 28th SIBGRAPI Conference on Graphics, Patterns and Images, Salvador, Brazil, 26–29 August 2015; pp. 95–102.
14. Wu, S.; Zhang, L. Using popular object detection methods for real time forest fire detection. In Proceedings of the 11th International Symposium on Computational Intelligence and Design (ISCID 2018), Hangzhou, China, 8–9 December 2018; pp. 280–284.
15. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [[CrossRef](#)] [[PubMed](#)]
16. 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 (CVPR 2016), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.

17. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Honolulu, Hawaii, 21–26 July 2017; pp. 7263–7271.
18. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
19. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
20. 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 European Conference on Computer Vision (ECCV 2016), Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.
21. Kim, B.; Lee, J. A video-based fire detection using deep learning models. *Appl. Sci.* **2019**, *9*, 2862. [CrossRef]
22. Lee, Y.; Shim, J. False Positive Decremented Research for Fire and Smoke Detection in Surveillance Camera using Spatial and Temporal Features Based on Deep Learning. *Electronics* **2019**, *8*, 1167. [CrossRef]
23. Pan, H.; Badawi, D.; Cetin, A.E. Computationally Efficient Wildfire Detection Method Using a Deep Convolutional Network Pruned via Fourier Analysis. *Sensors* **2020**, *20*, 2891. [CrossRef] [PubMed]
24. Wu, S.; Guo, C.; Yang, J. Using PCAand one-stage detectors for real-time forest fire detection. *J. Eng.* **2020**, *2020*, 383–387. [CrossRef]
25. Ultralytics-Yolov5. Available online: <https://github.com/ultralytics/yolov5> (accessed on 1 January 2021).
26. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2020), Washington, DC, USA, 14–19 June 2020; pp. 10781–10790.
27. Tan, M.; Le, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the International Conference on Machine Learning (ICML 2019), Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
28. BoWFire Dataset. Available online: <https://bitbucket.org/gbdi/bowfire-dataset/downloads/> (accessed on 1 January 2021).
29. FD-Dataset. Available online: <http://www.nnmrl.cn/EFDNet/> (accessed on 1 January 2021).
30. ForestryImages. Available online: <https://www.forestryimages.org/browse/subthumb.cfm?sub=740> (accessed on 1 January 2021).
31. VisiFire. Available online: <http://signal.ee.bilkent.edu.tr/VisiFire/> (accessed on 1 January 2021).
32. 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]
33. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the 13th European Conference on Computer Cision (ECCV 2014), Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
34. Wang, C.Y.; Mark Liao, H.Y.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of cnn. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2020), Washington, DC, USA, 14–19 June 2020; pp. 390–391.
35. Wang, K.; Liew, J.H.; Zou, Y.; Zhou, D.; Feng, J. Panet: Few-shot image semantic segmentation with prototype alignment. In Proceedings of the IEEE International Conference on Computer Vision (ICCV 2019), Seoul, Korea, 20–26 October 2019; pp. 9197–9206.
36. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
37. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Honolulu, Hawaii, 21–26 July 2017; pp. 4700–4708.
38. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1492–1500.
39. Neubeck, A.; Van Gool, L. Efficient non-maximum suppression. In Proceedings of the 18th International Conference on Pattern Recognition (ICPR 2006), Hong Kong, China, 20–24 August 2006; pp. 850–855.
40. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L. Pytorch: An imperative style, high-performance deep learning library. In Proceedings of the Neural Information Processing Systems (NIPS 2019), Vancouver, BC, Canada, 8–14 December 2019; pp. 8026–8037.
41. Bottou, L. Large-scale machine learning with stochastic gradient descent. In Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT 2010), Paris, France, 22–27 August 2010; pp. 177–186.
42. Zinkevich, M.; Weimer, M.; Li, L.; Smola, A.J. Parallelized stochastic gradient descent. In Proceedings of the Neural Information Processing Systems (NIPS 2010), Vancouver, BC, Canada, 6–11 December 2010; pp. 2595–2603.
43. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv* **2017**, arXiv:1711.05101.
44. Merino, L.; Caballero, F.; Martínez-de-Dios, J.R.; Maza, I.; Ollero, A. An unmanned aircraft system for automatic forest fire monitoring and measurement. *J. Intell. Robot. Syst.* **2012**, *65*, 533–548. [CrossRef]
45. Serón, F.J.; Gutiérrez, D.; Magallón, J.; Ferragut, L.; Asensio, M.I. The evolution of a wildland forest fire front. *Vis. Comput.* **2005**, *21*, 152–169. [CrossRef]
46. Pimont, F.; Dupuy, J.L.; Linn, R.R.; Dupont, S. Impacts of tree canopy structure on wind flows and fire propagation simulated with FIRETEC. *Ann. For. Sci.* **2011**, *68*, 523–530. [CrossRef]