electronics

MDPI

Article

Wildfire and Smoke Detection Using Staged YOLO Model and Ensemble CNN

Chayma Bahhar [1], Amel Ksibi [2], Manel Ayadi [2,*], Mona M. Jamjoom [3], Zahid Ullah [4], Ben Othman Soufiene [5] and Hedi Sakli [1,6]

1 MACS Research Laboratory RL16ES22, National Engineering School of Gabes, Gabes 6029, Tunisia
2 Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh 11671, Saudi Arabia
3 Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh 11671, Saudi Arabia
4 Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia
5 PRINCE Laboratory Research, ISITcom, Hammam Sousse, University of Sousse, Sousse 4000, Tunisia
6 EITA Consulting, 5 Rue du Chant des Oiseaux, 78360 Montesson, France
* Correspondence: mfayadi@pnu.edu.sa

Abstract: One of the most expensive and fatal natural disasters in the world is forest fires. For this reason, early discovery of forest fires helps minimize mortality and harm to ecosystems and forest life. The present research enriches the body of knowledge by evaluating the effectiveness of an efficient wildfire and smoke detection solution implementing ensembles of multiple convolutional neural network architectures tackling two different computer vision tasks in a stage format. The proposed architecture combines the YOLO architecture with two weights with a voting ensemble CNN architecture. The pipeline works in two stages. If the CNN detects the existence of abnormality in the frame, then the YOLO architecture localizes the smoke or fire. The addressed tasks are classification and detection in the presented method. The obtained model's weights achieve very decent results during training and testing. The classification model achieves a 0.95 F1-score, 0.99 accuracy, and 0.98e sensitivity. The model uses a transfer learning strategy for the classification task. The evaluation of the detector model reveals strong results by achieving a 0.85 mean average precision with 0.5 threshold (mAP@0.5) score for the smoke detection model and 0.76 mAP for the combined model. The smoke detection model also achieves a 0.93 F1-score. Overall, the presented deep learning pipeline shows some important experimental results with potential implementation capabilities despite some issues encountered during training, such as the lack of good-quality real-world unmanned aerial vehicle (UAV)-captured fire and smoke images.

Keywords: Fire detection; staged object detection; CNN; deep learning; computer vision

Citation: Bahhar, C.; Ksibi, A.; Ayadi, M.; Jamjoom, M.M.; Ullah, Z.; Soufiene, B.O.; Sakli, H. Wildfire and Smoke Detection Using Staged YOLO Model and Ensemble CNN. *Electronics* **2023**, *12*, 228. https://doi.org/10.3390/electronics12010228

Academic Editors: Giovanni Ramponi, Raffaela Cefalo and Žiga Kokalj

Received: 19 October 2022
Revised: 20 December 2022
Accepted: 21 December 2022
Published: 2 January 2023

1. Introduction

Forest fires are wild blasts of flames that start and spread in a mass of at least half a hectare in one piece, destroying at least part of the shrubby and/or tree-covered stages (high parts). A fire is a phenomenon that is beyond the control of humans, both in duration and in extent. Knowledge of the origins of fires is the foundation of any effective prevention policy. Indeed, when the causes of fire are known, it is then easier to eradicate them through the implementation of concrete actions, and therefore to limit the number of fires [1].

Generally speaking, summer is considered the most anticipated time of year for forest fires, because the combined effects of drought and low water content of plants are added to the high frequentation of these areas. However, the danger also exists in fall/winter and early spring, especially in massive lands or in mid-mountain areas. For ignition and combustion to occur, three factors must be combined, each in appropriate proportions: fuel,

which can be any material that can burn; an external source of heat (flame or spark); and oxygen, necessary to feed the fire. An uncontrolled forest fire may decimate everything within its path and spread over miles, crossing rivers and roads [2]. Every year, between 60 and 80 thousand wildfires occur and destroy 3 to 10 million miles of land. Forest fires have different environmental impacts depending on their importance, and their frequency causes may also be diverse [3].

Fires influence biological diversity in many ways. They are an important source of carbon emissions and contribute to global warming and changes in biodiversity [4]. Fires modify the volume of biomass and disrupt the hydrological cycle, affecting marine systems along with coral reefs and others [5]. Smoke from burning forests can dramatically minimize photosynthetic activity, involving the health of the general population, including animals [6]. For example, the Amazon has the best biodiversity sanctuary in the world. It is home to an abundance of species of plants, insects, animals, and countless more species never documented. Repetitive fires threaten this biodiversity. All this damage has introduced great challenges in firefighting [7].

Forest fires create more damage and increase the expense of fire suppression if they are not put out quickly enough [8]. The time spent between fire discovery and warning the relevant authorities is the most important aspect that could lessen wildfire dangers [9]. Currently, several terrestrial and spatial technologies are used to help official authorities in identifying wildfires at their early stages and the localization of their area [10]. However, these technologies have several drawbacks that may restrict their ability to detect fires. Furthermore, we need to develop new tools for monitoring wildfires and improve our fire control strategies to reduce the destruction of our forests and their riches. Recent advances in artificial intelligence and machine learning have increased the success of image-based modeling and analysis in a variety of applications [11]. In addition, machine learning is utilized in a variety of computer vision tasks, image classification [12], object detection, semantic segmentation [13], and a variety of other applications [14].

CNN has grown into the go-to approach for almost any image-or video-related problem. It outperforms the traditional approaches in terms of efficacy and mostly precision. It is also used in frame classification, object identification, semantic segmentation, and a range of other computer vision tasks [15]. CNN has a significant advantage over its predecessors in that it detects critical traits without the necessity for social interaction [16]. CNN is the most desirable and common deep learning model [17]. The feature of CNN is that it first splits the input image into multiple symmetric grids of regions, then classifies each distinct region into several classes [18]. Recurrent neural networks are neural networks used to process sequence data. Unlike traditional deep feedback networks, recurrent neurons in RNNs can feed back the output signal to the input. As a result, it may recall prior knowledge and acquire long-term reliance on consecutive inputs. Unfortunately, because of a gradient explosion or vanishing problem, classic RNN architecture is difficult to train. Several other works address scars in moderate-resolution imaging spectroradiometer (MODIS) images taken by satellites by implementing CNN architectures to improve scar detection [19,20].

The major contribution of the presented paper is to provide immersive assistance to the environment by reducing wildfire damages, which can be achieved through fire remediation and early anticipation. To deliver the described solution, the presented pipeline uses a deep-learning-based architecture that first combines three convolutional neural network architectures, namely, XceptionNet, MobileNetV2, and ResNet-50, as an ensemble. The listed CNN architectures are recognized with a low number of parameters compared and high accuracy, to achieve early fire prediction. The second contribution is linked to the implementation of the fire and smoke detection model by using the YOLO architecture, which is known to have low latency and higher detection frames per second (FPS). Overall, the model shows real-time accurate smoke and fire detection results; in addition, the present study contributes to the site as energy-efficient deep learning computer vision.

2. Related Work

Forest fires (or wildfires) are uncontrolled flames that naturally occur, causing severe damage to human and natural resources. The sooner the firefighter arrives at the location of the inferno, the more easily the conflagration may be controlled. The research addressing fire detection in the literature is increasing exponentially. Several researchers have developed fire detection applications also indicate the heavy focus on deep learning implementation within the internet of thing related devices with can massively increase precision which can lead to better automation [1–10], Table 1 summarizes the research that will be used to identify gaps and report on stated research motivations.

Table 1. Existing literature on deep-learning-based forest fire detection.

Ref	CNN Model	Dataset	Accuracy (%)	Early Detection	False Alarm Rate	Fire Intensity Estimation	Challenges
[21]	YOLOv3	UAV imagery	83	Yes	High	No	Low accuracy
[22]	Ssd_mobilenet_v1	UAV imagery	94	No	High	No	It takes more training time
[23]	Xception	UAV imagery	76	Yes	High	No	Inappropriately small dataset
	U-Net		91.99				
[24]	VGG-16	UAV imagery	99.74	Yes	High	No	High execution time The false alarm rate is very high
	ResNet-50		99.38				
	Inception v3		99.29				
	DenseNet		99.65				
	NASNetMobile		98.94				
	MobileNetV2		99.47				
[25]	DeepLabV3+	Aerial imagery	77.1	No	High	No	Low accuracy
[26]	Light-YOLOv4	Collected	95.24	No	High	No	Imbalanced data
[27]	Abi-LSTM	Real forest fire video	97.8	No	High	No	Overfitting issue
[28]	VGG19	DeepFire	95	No	High	No	High execution time

The authors in [21] present a proposal for a forest fire detection system that employs unmanned aerial vehicles to capture images of the forest, which are then analyzed using YOLOv3 and a small convolutional neural network. The experimental results indicate that the system exhibited high accuracy and speed, surpassing expectations and thereby demonstrating the efficacy and practicality of the UAV platform and deep learning-based fire detection system. The recognition rate of the algorithm, as determined by testing, was roughly 83%.

As part of the SFEDA project, Diyana et al. [22] have developed a platform for early forest fire detection (the Forest Monitoring System for Early Fire Detection and Assessment). Both fixed-wing and rotary-wing drones are used as unmanned aerial vehicles on this platform. The proposed platform for early forest fire detection employs two types of unmanned aerial vehicles equipped with optical, thermal, or both types of cameras. The fixed-wing drones operate at a medium altitude and survey the monitored region, while the rotary-wing drones operate at a lower altitude and verify suspected fire locations. Upon confirmation of a fire, an alarm is sounded to alert ground personnel and fire departments. The authors achieved a high accuracy rate of 94% through the use of a model based on the SSD with mobilenetv1 as the backbone and coco dataset weights.

Alireza et al. [23] have compiled a dataset featuring fire-related videos and images captured by drones in Northern Arizona, the dataset, titled FLAME (Fire Luminosity Airborne-based Machine learning Evaluation), utilizes normal and thermal cameras, drones were utilized to acquire aerial video footage and still images captured in four different color palettes: normal, Fusion, White-Hot, and Green-Hot. An ANN method was developed with

a 76% classification accuracy for frame-based fire classification. Moreover, the authors, to precisely determine fire borders, use segmentation methods. The FLAME solution reached 92% precision and recall of about 84%.

According to the authors of ref [24], six distinct convolutional neural network (CNN) architectures, including VGG16, DenseNet, Inception v3, MobileNet v2, and ResNet 50, were taken into account when building a wildfire inspection system and estimating its geolocation. The system uses an inexpensive commercial UAV. The suggested framework is divided into three major stages. In the first stage of the process, the operator communicates the region of interest using the tablet's graphical user interface (GUI). Once the coordinates of the location are transmitted to the drone, the drone independently navigates to and inspects the area, collecting additional data. The quadcopter then communicates the coordinates and video feed of the search area back to the operator's tablet after calculating the size and position of the fire using this information. The evaluation of the system's accuracy revealed that the VGG-16 model achieved 99.74%, the ResNet-50 model achieved 99.38%, the Inception v3 model achieved 99.29%, the DenseNet model achieved 99.65%, the NASNetMobile model achieved 98.9%, and the MobileNet v2 model achieved 99.47%.

Panagiotis et al. [25] have also presented a novel early fire detection system that integrates a UAV with a 360-degree aerial digital camera, allowing for infinite field-of-view recordings. The optical 360-degree camera is attached to an unmanned aerial vehicle. First, the photos in isosceles rectangle projection format were transformed to stereo graphic images. Following that, the authors used two DeepLab V3+ architecture to conduct flame and smoke segmentation experiment. The identified regions were merged and verified, considering the environmental appearance of the analyzed image. The authors developed a 360-degree fire detection dataset comprising 150 equirectangular photos. The proposed system had an F-score fire detection rate of 94.6%.

Furthermore, Yifan et al. [26] developed Light-YOLOv4, a lightweight detector for re-al-time detection of flames and smoke. They improved the YOLOv4 approach in three ways: replacing the YOLOv4 backbone network with a lightweight backbone network, using bidirectional cross-scale connections, and partitioning the convolution and separately computing the channel and spatial region. The Light-YOLOv4 detector had a flame detection accuracy of 86.43%, a smoke detection accuracy of 84.86%, a mAP@0.5 of 85.64%, a mAP%0.5 of 70.88%, and an FPS of 71 for flame and smoke detection tasks.

In reference [27], the authors developed an ABi-LSTM for detecting forest fire smoke. The ABi-LSTM consists of the spatial features Extraction network, the bidirectional LSTM network, and the temporal attention subnet. The ViBe background subtraction approach captures spatial features from candidate patches, which are extracted using the spatial features extraction network. The bidirectional LSTM network uses spatial features to learn long-term smoke-related information, and an attention network is used to focus on discriminative frames. The ABi-LSTM model achieved 97.8% accuracy, a 4.4% increase over the image-based deep learning model, using films from the forest fire monitoring system with a resolution of 1920×1080.

Ali Khan et al. [28] proposed DeepFire, a dataset and benchmark for detecting forest fires using UAVs. If a UAV detects a fire, it will communicate with nearby UAVs and send data to a remote forest fire disaster control center. The DeepFire dataset includes 1900 colored photos, 950 of which are in the fire and no-fire categories. The authors used the VGG19 architecture with transfer learning to improve prediction accuracy. The simulation results show that the proposed approach has an accuracy of 95%, precision of 95.7%, and recall of 94.2%.

In this work, we address two challenging tasks in computer vision by providing a solution to each of the problems, mainly image classification and object detection. The usage of CNN architecture allows us to achieve decent results in the classification task compared to the literature. As illustrated in the coming section, the usage of the YOLOv5 architecture elevates the detection performance and optimizes inference time. It knows the urge for the large dataset in the training phase of the deep learning approach to obtain good reliable

results; however, dataset creation, which includes data collection, data cleaning, and sanity check, is a very difficult and tedious operation due to the lack of raw data materials, which by itself could be a very challenging drawback and time-limiting issue.

This work is structured as follows: an introduction to related studies is provided in Section 2. The utilized models are presented in Section 3. Experimental results are presented in Section 4. Outcomes are discussed in Section 5. Finally, conclusions from this investigation, presented in video format for both classification and detection tasks, are presented in Section 6.

3. Materials and Methods

This study presents three applications based on the FLAME dataset that utilize deep learning solutions to address challenges related to fire detection. The first application in-volves the classification of fire versus non-fire using a deep neural network (DNN) ap-proach. The second application involves the detection of fire and smoke, which can be used for real-time monitoring and data labeling. The third application involves fire segmentation, which can be used to identify fire zones in video frames marked as containing fire in the first application.

3.1. Classification Task

This section describes the following process of building our classifier, discusses the performance of the classifier based on some evaluation procedures, and compares it to existing methodologies. Through this part, it is important to dictate the limits of the procedure and the challenges we experienced throughout the development.

3.1.1. Dataset

Unfortunately, until now, no open-sourced dataset has addressed forest fire as the first object. Deep learning approaches are considered data-driven approaches; hence, the word outlier in the database can deal with huge performance issues, from lowering performance to having type 1 errors. As we pointed out before, the data cleaning and processing stage is a critical step. In addition, DL approaches need a significantly large amount of data samples that are not easy to collect. Luckily, some research provides usable data collection. For the first task, we considered the FLAME dataset by Alireza et al. [23]. This dataset contains many palettes, including a normal-spectrum palette and thermal images (fusion, white-hot, and green) palettes. The provided FLAME dataset is captured from special cameras mounted under a UAV base, which means that the provided dataset is limited specifically to forest fire, so we could limit the performance. The generated dataset from the video of 29 FPS is 39,375, including 25,018 frames for the "fire" class and 14,357 for the "no-fire" class. The data was divided into a 70% training and 20% testing portion, with additional testing conducted in two stages. All samples are shuffled before being fed into the DL model. In order to overcome the problem of bias in the unequal number of samples in the "fire" and "non-fire" classes, augmentation techniques including horizontal flipping and random rotation were utilized to create fresh samples.

3.1.2. Method

In this study, the camera-captured frames were categorized using supervised machine learning techniques. This method involved training the model on labeled data to enable it to accurately identify different objects. For mixed photos that contain both fire and non-fire components, the frame was labeled as a fire frame and no fire was detected within the frame. CNN is one of the most advanced neural perceptron techniques for image classification. The study employs a binary classification model using binary cross-entropy (BCN) loss and some state-of-the-art CNNs, such as Google's Xception network [29], which is a deep convolutional neural network; MobileNetV2, proposed by Sandler et al. [30]; RestNet-50 [31]; and DenseNet121 [32]. Figure 1 illustrates the overall classifier architecture. We used these CNN as feature extractors, then stacked a global average pooling layer

(better to have a flatten layer or GAPGAP layer that stacks the prediction weights in smaller vectors). In addition, we added a dropout layer with 0.2 dropout rates as an overfitting prevention mechanism.

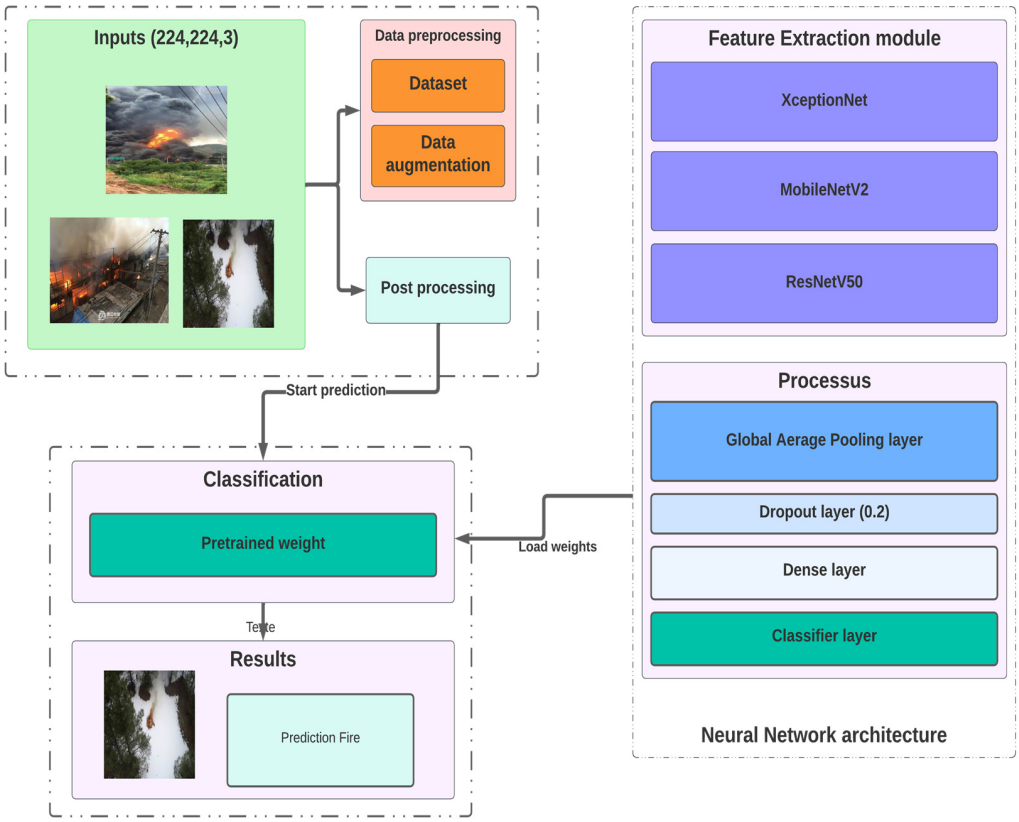


Figure 1. Classifier architecture: the classifier is divided into 3 stages/departments. The first stage is the data processing part, the second department is the neural net architecture, and the final one is the predictor part.

During the training phase, 25 epochs were sufficient, and the Adam optimizer's learning rate was fixed at 0.001. A batch size of 32 was used to fit the model. To evaluate the accuracy and loss of the model, the test dataset contained 8617 frames, including 5137 fire-labeled frames and 3480 non-fire-labeled frames, which were fed into the pre-trained networks. To better interpret the experimental results, medical classification metrics were used to address the importance and impact of detecting fires in forests. These metrics included accuracy, sensitivity, specificity, and negative predictive values (NPV). These metrics are summarized in Table 2.

Table 2. Model evaluation performance using classification metrics, namely, accuracy, sensitivity, specificity, and NPVs.

Metrics	Xception Baseline	MobileNetV2 Baseline	ResNet-50 Baseline
Accuracy	0.99	0.993	0.99
Sensitivity	0.992	0.996	0.996
Specificity	0.981	0.981	0.983
NPV	0.976	0.994	0.987

Overall, the evaluation metrics show very close results between the models, but we could say that MobileNetV2 is more balanced (with 3 metrics above 0.99) and more energy efficient due to the use of the inverted residual block (also called bottleneck architecture) that contains the depthwise/pointwise convolutional layers that help to obtain a smaller number of parameters and good results.

We could say that our proposed approach did achieve a good result in classifying the existence of fire from the input stream. Image classification is a foundational task in computer vision. However, to achieve fire and smoke real-time detection is more challenging which we will discuss in the next section.

3.2. Two-Stage Fire and Smoke Detection

For several reasons, object detection is the most relevant and difficult-to-achieve task in computer vision. As such, the dataset paradigm in deep learning models, as discussed in the previous section, is a data-dependent approach, which means to obtain a state-of-the-art result in object detection, we need to address some data dependencies: first and foremost, the data amount. In this section, we will address the object detection task due to the limitations of a dataset (availability and quality-wise). First, we will discuss the implemented dataset, then we will present our approach.

3.2.1. Dataset

Deep learning models cannot converge to minimal loss with a small dataset; second, in terms of data quality, we need to obtain clean data and, third, which is dedicated to the object detection task, data annotation. It is very difficult but cost-effective to perform data labeling. These tasks can differ in intensity degree (not for most medical image annotations). Furthermore, the severity level of these tasks may differ (this is not the case for medical image annotations).

There is unfortunately no public dataset that addresses the forest fire detection task; hence, we performed some web scraping to collect the dataset. In addition, we used some shared small datasets provided by the FLIR dataset presented in the previous section, the same as the smoke detection. On the other hand, some datasets were provided by [33,34], and the used dataset is presented in Figure 2.

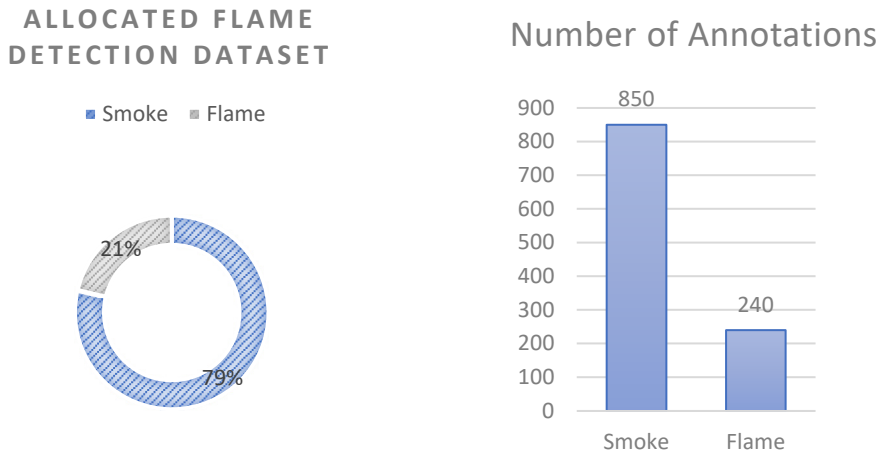


Figure 2. Dataset associated with the detection task: (**a**) number of samples in the dataset from the binary label (fire or smoke); (**b**) number of annotations in the images detected by the algorithm.

The allocated dataset for the classification task contains 7355 images with fire (non-forest fire) low-quality (non-fire local label). We started by cleaning the data, since the biggest class distribution was random fire images, so we annotated 200 wildfire images from the dataset of the first task. Then, we added 737 smoke images with good quality and good labeling, and the final dataset was 937 annotated images; however, the imbalance in the data distribution caused a massive drawback in the performance model. This problem presented is addressed in the following section.

3.2.2. Method

In this subsection, we will discuss our approach to achieving object detection tasks, including the data imbalance discussed in the previous subsection. Many methods, including data upsampling, are available to deal with data leakage and imbalance, but in our task object detection, the most amount of labeled data is available. Our approach addresses this issue by conducting two-stage object detection. There are many methods focused on two-stage object detection, such as Faster R-CNN [35], RetinaNet [36], etc.; however, these approaches are rather slower compared to one-stage object detection algorithms; hence, we implemented separate two-stage object detection algorithms. We chose YOLO for the 5th version with different architectures, and then we combined the object detection pipeline with the classifier. Figure 3 illustrates more.

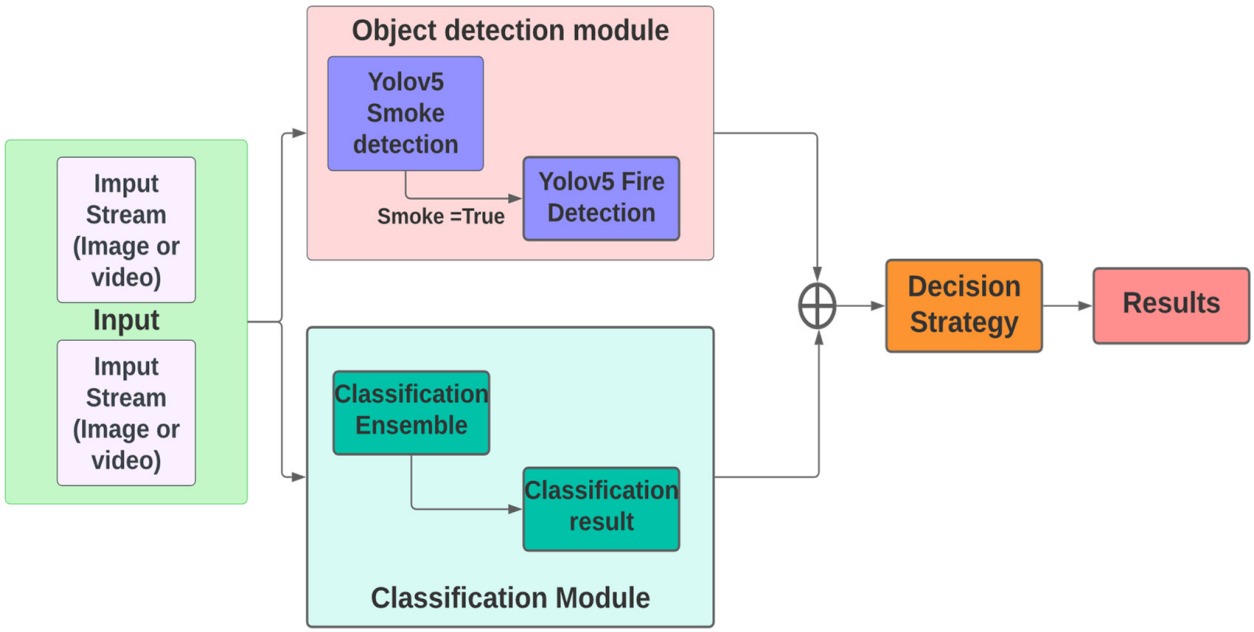


Figure 3. Our proposed method architecture: the method is divided into 4 blocks: input block, classification block, detection block, and lastly, the result block.

The method accepts image/video input, then passes it to the classification, which executes a binary classification operation, and at the same time, the object detection module, which is a two-stage YOLO detector. If the input stream contains smoke, it is passed to fire detection. The collected results from the two blocks are then combined to give the final decision. Our proposed two-stage detector uses two YOLOv5 architectures: YOLOv5s for smoke detection and the standard architecture for the detection.

3.2.3. Dataset Label

In this study, the Squeeze-and-Excitation (SE) mechanism was primarily employed with YOLOv5s conv layers. The SE channel attention mechanism focuses solely on internal channel information, disregarding the importance of location information in understanding the spatial composition of an item in the image, even though the spatial composition of the item in the visual is critical. This work employs SE to draw the network's attention to "what is" in a certain data input. Based on the SE attention mechanism, this study combined the average and maximum pooling characteristics. This study combined global pooling [37] and maximal pooling [38] techniques to collect feature mapping spatial information and facilitate efficient target region identification.

The YOLOv5s approach is improved in terms of real-time detection performance owing to its convolutional network architecture, small neural core size, and regression boundary box algorithm design. To further improve accuracy without increasing network

depth, the presented updated method replaces all existing network connections. Residual fusion is employed to retain information when screening transmission characteristics, resulting in more precise localization and classification.

3.2.4. Smoke Detector Architecture

When selecting a target detection method, the YOLO [39] network was used for target recognition and detection. Compared to other target detection algorithms, the YOLO method was found to be faster at recognizing targets and more consistent with the experiment's extraction procedure of Regions of Interest. The YOLOv5s architecture follows the same overall pattern of YOLOv3 and YOLOv4, consisting of four components: Input, Backbone, Neck, and Output, as shown in Figure 4.

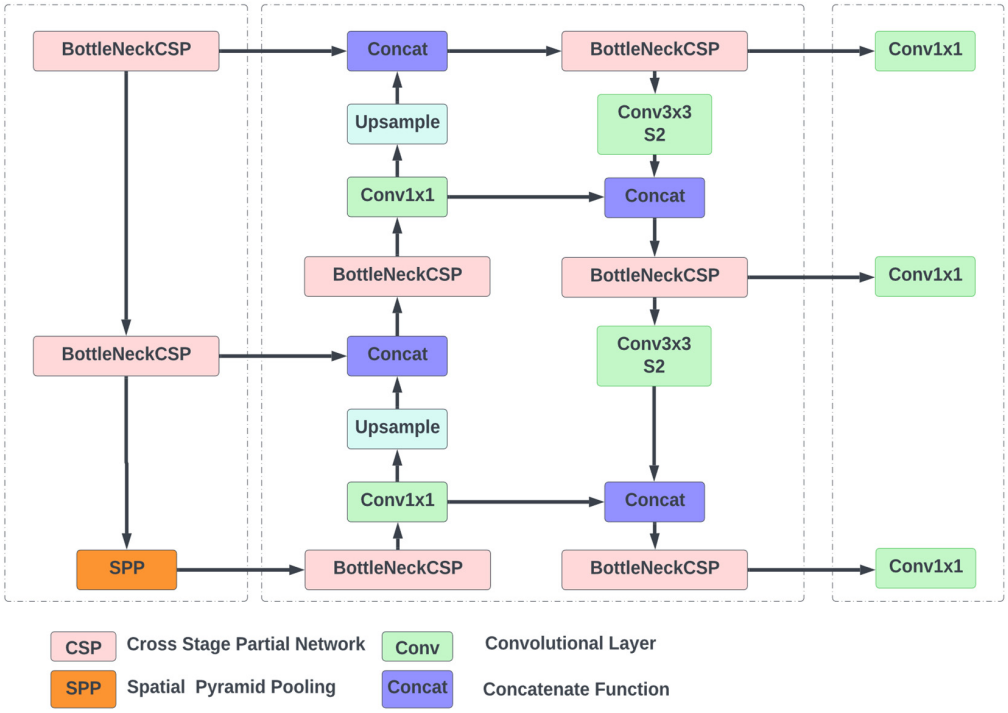


Figure 4. YOLOv5s network model.

Adaptive scaling images are used as inputs to automatically select the most appropriate anchor frame value for the dataset using the mosaic [40] data augmentation technique.

The Focus structure is combined with the CSP Net Cross Stage Partial Network [41], a cross-stage local fusion network, to form the Backbone. This Focus structure consists of four slice operations and one convolution operation involving 32 convolution cores, transforming the original 640 × 640 × 3 images into a feature map of 320 × 320 × 32. CSP Net implements the idea of a dense cross-layer jump-layer connection from DenseNet [32], performing local cross-layer fusion to create richer feature graphs by combining information from multiple layers.

The Yolov5 utilizes a feature pyramid network (PANet) [42] as its neck to increase information flow. PANet utilizes a unique FPN architecture with an enhanced bottom-up route to boost low-level feature propagation. Additionally, Adaptive feature pooling allows important information from each level of features to be efficiently transmitted to the next subnetwork, connecting all feature levels and the feature grid. PANet also maximizes the use of reliable localization signals in lower layers, improving object positioning accuracy. Lastly, Yolo layer, the core of Yolov5, generates feature maps with three different sizes (18, 36, 72) that enable multiscale (36) prediction, which allows the model to process small, medium, and large objects.

Instead of YOLOv3, GIOU Loss is utilized as the output layer. GIOU Loss is an improved version of IOU Loss and is used as a loss function to increase the scale of the intersection and address the issue that IOU Loss cannot maximize the disjoint of the two boxes.

3.2.5. Fire Detector Architecture

The fire detector is a second solution stage in the detection module that is triggered if smoke is detected. The fire detection block uses the YOLOv5l version, which has the same properties, illustrated in Figure 5.

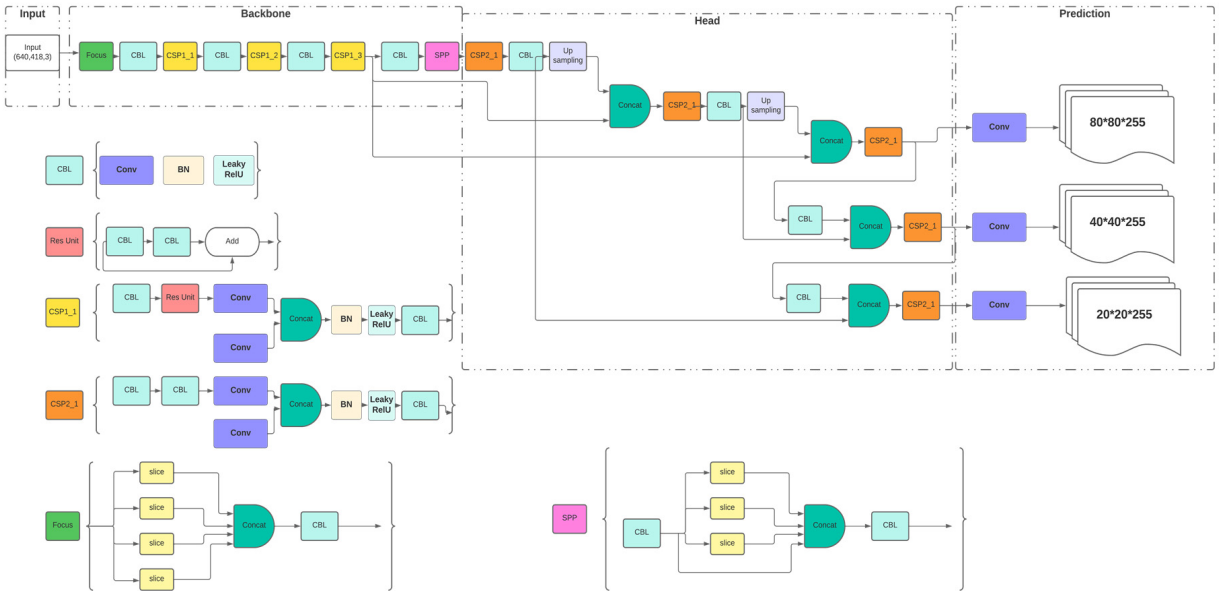


Figure 5. Yolov5L's network architecture consists of three components: the backbone (CSPDarknet), the neck (PANet), and the head (Yolo Layer). The data is first passed through CSP Darknet for feature ex-traction, then through PANet for feature fusion, and finally through the Yolo Layer for the detection results.

3.3. Model Evaluation

The constructed model's performance is evaluated using the following metrics: accuracy, precision, recall, and F1-score. Each metric is defined as follows:

- Accuracy: Is the proportion of correctly labeled samples out of the total number of samples. Accuracy is determined as:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

- Precision: The accuracy measure is the ratio of accurately predicted positive results (TP) to the total number of positive results ($TP + FP$) expected by the model. A substantial number of FPs leads to decreased accuracy [30]. The precise range is computed as follows and is between 0 and 1:

$$Precision = \frac{TP}{TP + FP}$$

- Recall: Is the proportion of true positives to the sum of true positives and false negatives. The recall is calculated using the following equation:

$$Recall = \frac{TP}{TP + FN}$$

❖ F1-score: Is the harmonic mean of Precision and Recall. The definition is as follows: the definition is as follows:

$$F1\ score = 2 \times \frac{Recall \times Precision}{Recall + Precision}$$

where *TP* refers to true positive, *FP* refers to false positive, *TN* refers to true negative, and FN refers to false negative.

4. Experimental Results

In this part, we will go through the experimental findings that were used to illustrate the performance of the suggested technique, as well as the detection results. As illustrated in Table 3, we present the performance of both YOLOv5s (smoke detection) and YOLOv5l (fire detection).

Table 3. Fire and smoke detection experimental results.

Models	Precision	Recall	mAP@0.5
Model 1: Smoke Detector	0.273	0.4	0.45
Model 1: Fire Detector	0.707	0.67	0.65
Model 2: Smoke Detector	0.801	0.99	0.85
Combined Models	0.8	0.87	0.76

Originally, the first model did not give good results for both fire and smoke due to the heavily unbalanced collected dataset. The results were good with regard to fire detection, but not with smoke. We applied a separate detector on the smoke dataset, which achieved very good results, with mean average precision of 0.85; hence, the final model is a prediction combination between the two models.

There are several problems we did face during the build of the presented model, referring to the previous section, data quality and computational power have been the biggest factor during the training, but still, we did manage to build high precision and recall model that can not only prediction the possibility initial wildfire sparkles but also to localize the smoke in a certain frame.

The limited result provided by model 1 is explained in the following results, starting with the dataset, the model is somehow biased to detect fire due to the quality of the user data, splitting the dataset into different meanings using the second detection, the smoke result improved a lot compared to the first model, the model's aggregation achieved better results than using the one model two objects. The smoke detection is the best-achieved task with 0.85 mAP, which is a good indicator as generally speaking, we intend to find smoke first.

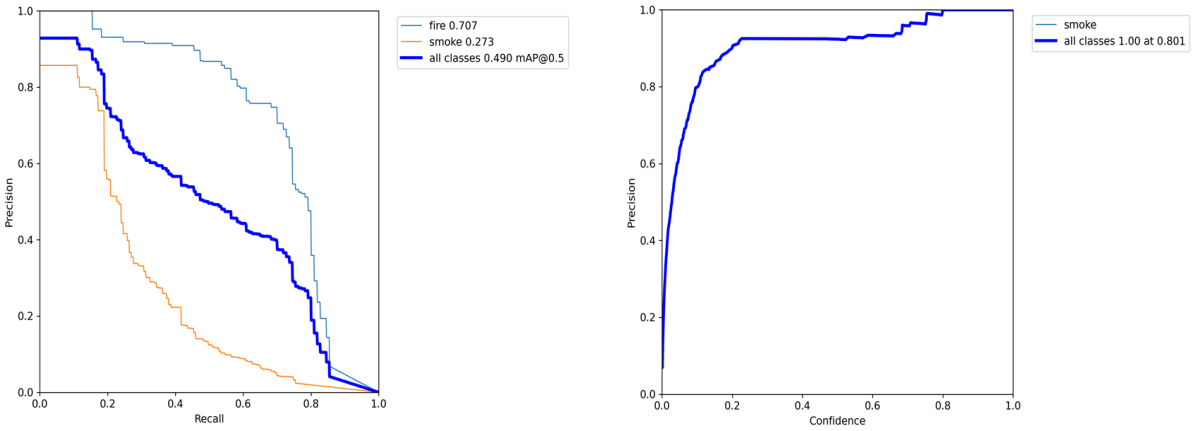
5. Discussion

We can see that the two detectors achieved acceptable results in terms of both fire detection and smoke detection. Due to the disparity in the quality of the dataset, the smoke detector achieved the most favorable results compared to the fire detector, detailed in Figures 6 and 7. Overall, the model achieves better performance compared to the literature in Table 4.

We can see some issues with the detection due to limited data. The model detects the smoke in the image, but also, one of the limitations of the YOLO architecture is the small region detection. The precision curve toward smoke detection presented in Figure 8, when combined prediction-wise with the first model, could not only obtain better localization of the fire and smoke, but also showed very acceptable results.

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

Figure 6. First-stage smoke detection and some experimental results: the model can detect specific colors of smoke (white smoke) due to the limitation of the dataset ((**a**) model 1 and (**b**) model 2).



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

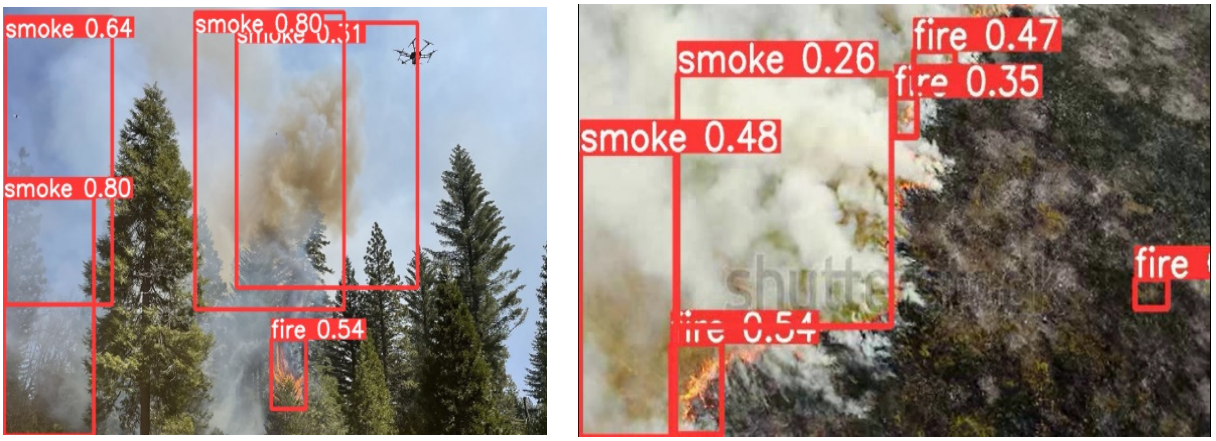
Figure 7. Model 1 and 2 precision evaluation: (**a**) first model, which is good at detecting fire; (**b**) smoke detection. According to the smoke and fire detector precision curve, the model jumps from 0.2 in the first model to 0.8.

Table 4. Comparison of the combined detection architecture with the literature review. Best results are in bold.

Models	F1 Score	mAP@0.5
DeepLabV3+ [25]	0.94	-
Light-YOLOv4 [26]	-	0.856
YOLOV3 [21]	0.81	0.79
Model 2 Smoke Detector	0.93	0.858

From the start, our goal was to build a good functional and usable model, and also, we think that detecting smoke is more important than fire due to the possibility to prevent fire from escalating. We can see better detection results with the two-stage detection with more coverage of the smoke.

Two-stage detection combined with the classifier ensemble has a promising result due to the fast response of the YOLOv5 model. In addition, we intend to implement a highly energy-efficient neural network for faster prediction, which can be extremely useful with edge devices.



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

Figure 8. Two-stage results: smoke detection combined with fire detection ((**a**) model 1 and (**b**) model 2).

In general, the model achieved the requested objective by detecting smoke in real time. However, due to data limitation factors and the YOLOv5s architecture, the model has some limitations, namely, the problem with big smoke/fire locations due to the limited data in both fire and smoke, noting that considering using artificial data cannot help in this situation.

Based on the obtained results, the model had potential applicability, and compared to the literature, it is possible to improve the model. It is much clearer that the limiting factor is the data quality and quantity.

6. Conclusions

Effective implementation of convolutional neural networks can significantly improve object detection performance. However, forest fires are dynamic events with no fixed shape that an individual object detector cannot handle. In addition, object detectors can be easily fooled by fire-like objects and produce false positives due to their limited field of view. This study presents a novel ensemble learning approach for real-time forest fire detection to address these challenges. The approach incorporates two strong object detectors (Yolov5s and Yolov5l) with varying levels of ability to make the overall model more robust in various forest fire situations.

Our model, which utilizes an ensemble classifier to guide the detection process and minimize false positives, has been found to outperform other common object classifiers in terms of precision, recall, frame accuracy, and F1 score according to experimental results. These improvements enable the model to effectively perform in real-world forestry applications.

Author Contributions: Conceptualization, methodology, C.B.; conceptualization, methodology, writing—original draft preparation, results in analysis, A.K.; data collection, data analysis, writing—review and editing, results in analysis, M.A.; methodology, writing—review and editing, M.M.J.; methodology, writing—review and editing, Z.U.; conceptualization, methodology, methodology, writing—review and editing, B.O.S.; conceptualization, methodology, methodology, writing—review and editing, H.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was financially supported by Princess Nourah bint Abdulrahman University Researchers Supporting Project (Number PNURSP2023R104), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Data Availability Statement: The datasets used during the current study are available from the corresponding author on reasonable request.

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

References

1. Almalki, F.; Soufiene, B.; Alsamhi, S.; Sakli, H. A Low-Cost Platform for Environmental Smart Farming Monitoring System Based on IoT and UAVs. *Sustainability* **2021**, *13*, 5908. [CrossRef]
2. Hu, Y.; Zhan, J.; Zhou, G.; Chen, A.; Cai, W.; Guo, K.; Hu, Y.; Li, L. Fast Forest fire smoke detection using MVMNet. *Knowl.-Based Syst.* **2022**, *241*, 108219. [CrossRef]
3. Wahyono; Harjoko, A.; Dharmawan, A.; Adhinata, F.D.; Kosala, G.; Jo, K.-H.G. Real-Time Forest Fire Detection Framework Based on Artificial Intelligence Using Color Probability Model and Motion Feature Analysis. *Fire* **2022**, *5*, 23. [CrossRef]
4. Guede-Fernández, F.; Martins, L.; de Almeida, R.V.; Gamboa, H.; Vieira, P. A Deep Learning Based Object Identification System for Forest Fire Detection. *Fire* **2021**, *4*, 75. [CrossRef]
5. Benzekri, W.; El Moussati, A.; Moussaoui, O.; Berrajaa, M. Early Forest Fire Detection System using Wireless Sensor Network and Deep Learning. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 5. [CrossRef]
6. Shahid, M.; Virtusio, J.J.; Wu, Y.H.; Chen, Y.Y.; Tanveer, M.; Muhammad, K.; Hua, K.L. Spatio-Temporal Self-Attention Network for Fire Detection and Segmentation in Video Surveillance. *IEEE Access* **2022**, *10*, 1259–1275. [CrossRef]
7. Muhammad, K.; Ahmad, J.; Lv, Z.; Bellavista, P.; Yang, P.; Baik, S.W. Efficient Deep CNN-Based Fire Detection and Localization in Video Surveillance Applications. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 1419–1434. [CrossRef]
8. Wu, C.; Shao, S.; Tunc, C.; Hariri, S. Video Anomaly Detection using Pre-Trained Deep Convolutional Neural Nets and Context Mining. In Proceedings of the IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA), Antalya, Turkey, 2–5 November 2020; pp. 1–8. [CrossRef]
9. Xu, R.; Lin, H.; Lu, K.; Cao, L.; Liu, Y. A Forest Fire Detection System Based on Ensemble Learning. *Forests* **2021**, *12*, 217. [CrossRef]
10. Pan, J.; Ou, X.; Xu, L. A Collaborative Region Detection and Grading Framework for Forest Fire Smoke Using Weakly Supervised Fine Segmentation and Lightweight Faster-RCNN. *Forests* **2021**, *12*, 768. [CrossRef]
11. Zhang, Q.X.; Lin, G.H.; Zhang, Y.M.; Xu, G.; Wang, J.J. Wildland forest fire smoke detection based on faster R-CNN using synthetic smoke images. *Procedia Eng.* **2018**, *211*, 441–446. [CrossRef]
12. Jeong, M.; Park, M.; Nam, J.; Ko, B.C. Light-Weight Student LSTM for Real-Time Wildfire Smoke Detection. *Sensors* **2020**, *20*, 5508. [CrossRef] [PubMed]
13. Xin, Z.; 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]
14. Mukhiddinov, M.; Abdusalomov, A.B.; Cho, J. A Wildfire Smoke Detection System Using Unmanned Aerial Vehicle Images Based on the Optimized YOLOv5. *Sensors* **2022**, *22*, 9384. [CrossRef]
15. Lu, K.; Xu, R.; Li, J.; Lv, Y.; Lin, H.; Liu, Y. A Vision-Based Detection and Spatial Localization Scheme for Forest Fire Inspection from UAV. *Forests* **2022**, *13*, 383. [CrossRef]
16. Gagliardi, A.; de Gioia, F.; Saponara, S. A real-time video smoke detection algorithm based on Kalman filter and CNN. *J. Real-Time Image Process.* **2021**, *18*, 2085–2095. [CrossRef]
17. He, L.; Gong, X.; Zhang, S.; Wang, L.; Li, F. Efficient attention based deep fusion CNN for smoke detection in fog environment. *Neurocomputing* **2021**, *434*, 224–238. [CrossRef]
18. Bouguettaya, A.; Zarzour, H.; Taberkit, A.M.; Kechida, A. A Review on Early Wildfire Detection from Unmanned Aerial Vehicles Using Deep Learning-Based Computer Vision Algorithms. *Signal Process.* **2022**, *190*, 108309. [CrossRef]
19. Dao, M.; Kwan, C.; Ayhan, B.; Tran, T.D. Burn Scar Detection Using Cloudy MODIS Images via Low-Rank and Sparsity-Based Models. In Proceedings of the IEEE Global Conference on Signal and Information Processing, Washington, DC, USA, 7–9 December 2016; pp. 177–181. [CrossRef]
20. Ayhan, B.; Kwan, C. On the Use of Radiance Domain for Burn Scar Detection under Varying Atmospheric Illumination Conditions and Viewing Geometry. *SIViP* **2017**, *11*, 605–612. [CrossRef]
21. Jiao, Z.; Zhang, Y.; Xin, J.; Mu, L.; Yi, Y.; Liu, H.; Liu, D. A Deep Learning Based Forest Fire Detection Approach Using UAV and YOLOv3. In Proceedings of the1st International Conference on Industrial Artificial Intelligence (IAI), Shenyang, China, 23–27 July 2019; pp. 1–5. [CrossRef]
22. Kinaneva, D.; Hristov, G.; Raychev, J.; Zahariev, P. Early Forest Fire Detection Using Drones and Artificial Intelligence. In Proceedings of the 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 20–24 May 2019; pp. 1060–1065. [CrossRef]
23. Shamsoshoara, A.; Afghah, F.; Razi, A.; Zheng, L.; Fulé, P.Z.; Blasch, E. Aerial imagery pile burn detection using deep learning: The FLAME dataset. *Comput. Networks.* **2021**, *193*, 108001. [CrossRef]
24. Novac, I.; Geipel, K.R.; Gil, J.E.D.; Paula, L.G.D.; Hyttel, K.; Chrysostomou, D. A Framework for Wildfire Inspection Using Deep Convolutional Neural Networks. In Proceedings of the IEEE/SICE International Symposium on System Integration (SII), Honolulu, HI, USA, 12–15 January 2020; pp. 867–872. [CrossRef]
25. Barmpoutis, P.; Stathaki, T.; Dimitropoulos, K.; Grammalidis, N. Early Fire Detection Based on Aerial 360-Degree Sensors, Deep Convolution Neural Networks, and Exploitation of Fire Dynamic Textures. *Remote Sens.* **2020**, *12*, 3177. [CrossRef]
26. Wang, Y.; Hua, C.; Ding, W.; Wu, R. Real-time detection of flame and smoke using an improved YOLOv4 network. *SIViP* **2022**, *16*, 1109–1116. [CrossRef]
27. Cao, Y.; Yang, F.; Tang, Q.; Lu, X. An Attention Enhanced Bidirectional LSTM for Early Forest Fire Smoke Recognition. *IEEE Access* **2019**, *7*, 154732–154742. [CrossRef]

28. Khan, A.; Hassan, B.; Khan, S.; Ahmed, R.; Abuassba, A. DeepFire: A Novel Dataset and Deep Transfer Learning Benchmark for Forest Fire Detection. *Mob. Inf. Systems.* **2022**, *2022*, 5358359. [CrossRef]
29. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
30. Sandler, M.; Howard, A.; Zhu, M. Mobilenetv2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520. [CrossRef]
31. 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, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
32. 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, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
33. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
34. Miao, J.; Zhao, G.; Gao, Y.; Wen, Y. Fire Detection Algorithm Based on Improved YOLOv5. In Proceedings of the International Conference on Control, Automation and Information Sciences, Jeju, Japan, 12–15 October 2021; pp. 776–781.
35. Ullah, Z.; Jamjoom, M. An Intelligent Approach for Arabic Handwritten Letter Recognition Using Convolutional Neural Network. *PeeJ Comput. Sci.* **2022**, *8*, e995. [CrossRef]
36. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2999–3007. [CrossRef]
37. Ahmad, H.; Ahmad, S.; Asif, M.; Rehman, M.; Alharbi, A. Evolution-Based Performance Prediction of Star Cricketers. *Comput. Mater. Contin.* **2021**, *69*, 1215–1232. [CrossRef]
38. Mehos, M.; Jorgenson, J.; Denholm, P.; Turchi, C. An Assessment of the Net Value of CSP Systems Integrated with Thermal Energy Storage. *Energy Procedia* **2015**, *69*, 2060–2071. [CrossRef]
39. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
40. Li, S.; Wang, X. YOLOv5-Based Defect Detection Model for Hot Rolled Strip Steel. *J. Phys. Conf. Ser.* **2022**, *2171*, 012040. [CrossRef]
41. Redmon, J.; Farhadi, A. YOLOv3: An Incremental improvement. In *Computer Vision and Pattern Recognition*; Springer: Berlin/Heidelberg, Germany, 2018.
42. Yan, J.; Wang, H.; Yan, M.; Diao, W.; Sun, X.; Li, H. IoU-Adaptive Deformable R-CNN: Make Full Use of IoU for Multi-Class Object Detection in Remote Sensing Imagery. *Remote Sens.* **2019**, *11*, 286. [CrossRef]