

Article

Robust Forest Fire Detection Method for Surveillance Systems Based on You Only Look Once Version 8 and Transfer Learning Approaches

Nodir Yunusov¹, Bappy MD Siful Islam¹, Akmalbek Abdusalomov^{1,2}  and Wooseong Kim^{1,*} 

¹ Department of Computer Engineering, Gachon University, Sujeong-gu, Seongnam-si 461-701, Gyeonggi-do, Republic of Korea; bappy.ac.bd@gmail.com (B.M.S.I.); a.abdusalomov@tsue.uz (A.A.)

² Department of Information Systems and Technologies, Tashkent State University of Economics, Tashkent 100066, Uzbekistan

* Correspondence: wooseong@gachon.ac.kr

Abstract: Forest fires have emerged as a significant global concern, exacerbated by both global warming and the expanding human population. Several adverse outcomes can result from this, including climatic shifts and greenhouse effects. The ramifications of fire incidents extend widely, impacting human communities, financial resources, the natural environment, and global warming. Therefore, timely fire detection is essential for quick and effective response and not to endanger forest resources, animal life, and the human economy. This study introduces a forest fire detection approach utilizing transfer learning with the YOLOv8 (You Only Look Once version 8) pretraining model and the TranSDet model, which integrates an improved deep learning algorithm. Transfer Learning based on pre-trained YoloV8 enhances a fast and accurate object detection aggregate with the TranSDet structure to detect small fires. Furthermore, to train the model, we collected 5200 images and performed augmentation techniques for data, such as rotation, scaling, and changing due and saturation. Small fires can be detected from a distance by our suggested model both during the day and at night. Objects with similarities can lead to false predictions. However, the dataset augmentation technique reduces the feasibility. The experimental results prove that our proposed model can successfully achieve 98% accuracy to minimize catastrophic incidents. In recent years, the advancement of deep learning techniques has enhanced safety and secure environments. Lastly, we conducted a comparative analysis of our method's performance based on widely used evaluation metrics to validate the achieved results.

Keywords: forest fire; fire detection; YOLOv8; deep learning; TranSDet; wildfire incidents; brushfire spread



Citation: Yunusov, N.; Islam, B.M.S.; Abdusalomov, A.; Kim, W. Robust Forest Fire Detection Method for Surveillance Systems Based on You Only Look Once Version 8 and Transfer Learning Approaches.

Processes **2024**, *12*, 1039. <https://doi.org/10.3390/pr12051039>

Academic Editors: Xiong Luo, Sheng Du, Li Jin, Xiongbo Wan and Zixin Huang

Received: 25 March 2024

Revised: 1 May 2024

Accepted: 15 May 2024

Published: 20 May 2024



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

1. Introduction

Forest fires are catastrophic events that result in widespread economic, ecological, and environmental damage all over the world. High temperatures can ignite dry fuels, such as sawdust, leaves, and lightning, or they can be sparked by human activities, such as unextinguished fires, arson, or improperly burned debris [1]. Between 2002 and 2016, an estimated 4,225,000 km³ of forest burned by fire [2]. Forest fires can arise from both natural phenomena and human activities. Natural causes include factors such as heat, dry weather, lightning strikes, volcanic eruptions, coal-seam fires, and smoking. On the other hand, human-induced causes encompass activities like cooking, accidental ignition, and deliberate fire lighting. Either natural or human-created fires have a severe impact on wild as well as human life. Human activity contributes 90%, and lightning provides the rest of the 10% of forest fire sources [3]. Both people and wildlife are affected by wildfire toxic gases in the troposphere [4].

Forest fires were previously identified by watchtowers, which are ineffective, and human surveillance is expensive [4,5]. Implementing automation offers a significantly improved and more precise approach to forest fire detection. Additionally, weather conditions, rainy days, and high temperatures constrain the fire detection process. Therefore, a real-time fire detection technique is much better and has a low cost [6].

To prevent expanding fire, commonly there are two methods used, and are vision-based fire detection (VFDs) and sensors that exhibit sensitivity to sound, flames, temperature, gases, or solid materials [7]. Sensors trigger chemical characteristics of smoke and the variance of the environment. Once smoke is in the range, the sensor alarm turns on. In certain scenarios, sensor-based detection systems [8] may not be feasible, particularly in expansive coverage areas, forested regions, and environments with elevated temperatures, as they may generate numerous false alarms [9]. Moreover, the operational range of sensors is constrained [10], leading to a reduced lifespan.

The inception of object detection leveraging AI (artificial intelligence) traces back to 1986 [11]. Nonetheless, the substantive contributions of AI and machine learning models were hindered during that period due to technological constraints. Furthermore, the introduction of Facebook and Google big data forced deep learning (DL) models to have technical advantages. Perceptron was the first stepping stone towards deep learning. Time-different deep learning techniques, such as Alexnet [12], VGG16 [13], Faster R-CNN [14], Detectron2 [15], and YOLOv1 [16], were introduced by the need for scaling tasks, speed, and so on. In this study, the following problems were identified in detecting forest fires.

1. Collection and labeling images of forest fires pose significant challenges, primarily attributed to the absence of readily available open-access datasets containing fire images.
2. Given the absence of standardized shapes or sizes of fires, detecting objects of varying dimensions in real-time poses a considerable challenge, particularly in achieving high levels of accuracy.
3. Fire and fire-like object detection as fire is a real problem in forest fire identification and classification.

The integration of artificial intelligence (AI) with mathematical models for fire detection and prediction has been a burgeoning area of research. In [17–19], researchers provide a comprehensive overview of various machine-learning techniques applied to fire detection and prediction. It discusses the integration of AI algorithms with mathematical models for more accurate predictions. These papers explore the application of deep learning techniques, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), for wildfire detection and prediction. They discuss how these AI methods can be integrated with mathematical models to improve prediction accuracy [20,21].

Some papers discuss wildfire probability modeling and resilience in wildland fire management, exploring wildfire risk assessment methodologies and resilience planning strategies [22]. The mentioned approaches introduced probabilistic risk modeling, scenario analysis, and community-based resilience planning. It also provides case studies illustrating the application of these methods in wildfire-prone regions. Scientists also present an integrated modeling framework that combines wildfire probability modeling with resilience assessment. It discusses how environmental factors, land use patterns, and social dynamics influence both wildfire occurrence and community resilience [23,24].

This paper is organized into the following sections: Section 2 presents an overview of the relevant literature. Section 3 details our dataset and explains our proposed fire detection method. Section 4 provides a comprehensive examination of experimental results, accompanied by a detailed analysis of performance, and Section 5 discusses limitations and outlines future work. In the end, Section 6 provides the findings and final summaries to overcome challenges.

2. Literature Review

Forest flame recognition techniques generally fall into the following two main categories: artificial intelligence/computer vision approaches and sensor-based techniques. There are some limitations to the sensor-based method. To overcome these limitations, we upgraded the deep learning method based on transfer learning on pre-trained Yolov8 and TranSDet. CNNs [25–29] (convolutional neural networks) and DNN (deep neural network) are the most popular methods in the field of object detection. As sensor-based systems have some limitations and object detection techniques cover these limitations, deep learning has gained more popularity [30,31].

2.1. Detection of Forest Fires Utilizing Machine Learning and Deep Learning Methodologies

With AI advancing day by day, numerous research techniques have been innovated in the field of deep learning on fire detection. Among all deep learning models, CNNs commonly use techniques in computer vision. Toulouse et al. [32] proposed a system to recognize geometrical observation of flame based on length and position. The model's foundation is the classification of pixels based on the non-refractory pictures' average intensity. Jian et al. [33] upgraded the model with a boundary detection operator with multistep operation. However, the model only obtains better performance in simple and stable fire and flame images. For the first time in the history of deep learning, using a combination of foreground and background color frames, Turgay [34] generated a real-time fire detector. Though the model produced better output for fire images only, for fire-like images, and in the presence of smoke and shadow, the model was outperformed. With the improvement in deep learning, LDS is capable of identifying dynamic smoke and flame textures [35].

With technological improvement, researchers introduced the YOLOFM algorithm based on YOLOv5n approach to classifying small objects, which makes forest fire detection more accurate [36]. Furthermore, in [37–41], more improved fire detection techniques were innovated. To improve fire detection precision accuracy, a DL-based approach named DTA (Detection and Temporal Accusations) was proposed in [42]. It tries to identify a person by imitating the identification process. This model increases the accuracy of flame identification while accurately interpreting the temporal SRoF characteristics. The authors of [43] designed an early flame recognition technique using a lightweight CNN.

2.2. Detection of Forest Fires Utilizing YOLO and Transformers Methodologies

In [44], the Yolov3 network was employed for small-scale object classification. This model leverages the K-means clustering technique to distinguish flames. In [45], the authors introduce a fire depth separable convolution to mitigate the computational costs of the method and enhance the perceptual feature layer using the cavity convolution method. Furthermore, in [46], the authors propose ELASTIC-YOLOv3 as an enhancement over YOLOv2 to enhance performance without increasing the number of parameters. Initially, fire detection algorithms encountered challenges, such as high light intensity, limited color information, and variations in flame shapes and sizes, which prompted the development of enhanced technologies for real-time flame classification and recognition, manifested in modulated YOLO networks (v4, v5, v6, v7, v8) as introduced in [47–53].

Far object detection using a deep learning model did not perform well before introducing transformer learning. Transformer-based learning shows superior object prediction performance in various advanced vision areas, including image/video analyzing [54], image super-resolution [55], object recognition [56], segmentation [57], and ViT [58] (image classification). This advancement is facilitated through the utilization of Vision Transformer [59], DeiT [60] (Data-Efficient Image Transformers), and MedT [61] (Medical Transformers). Vision Transformer (ViT) is a deep learning model that applies the transformer architecture, originally designed for sequence processing tasks like natural language processing, to image data. Instead of using convolutional layers like traditional convolutional neural networks (CNNs), ViT represents images as sequences of patches, which are then processed

by a transformer encoder. ViT divides the input image into fixed-size, non-overlapping patches and flattens them into sequences, which are then fed into the transformer encoder. By leveraging self-attention mechanisms, ViT captures long-range dependencies within the image and learns representations that are effective for image classification tasks. Data-Efficient Image Transformers (DeiT) is a variant of ViT designed to achieve better performance with smaller amounts of labeled training data. DeiT introduces techniques such as distillation, where a larger pre-trained model (teacher model) is used to distill knowledge into a smaller student model. Medical Transformers (MedT) are transformer-based models specifically tailored for medical imaging tasks. These models adapt the transformer architecture to handle medical image data, which often differ from natural images due to factors like different modalities (e.g., MRI, CT scans) and specialized structures (e.g., anatomical features). MedT models are trained in medical imaging datasets and learn to perform tasks, such as image classification, segmentation, and detection, within the medical domain. By leveraging the power of transformers, MedT models can capture complex patterns and relationships in medical images, leading to improved performance in various medical image analysis tasks [62–64].

3. Proposed Method and Model Architecture

3.1. Forest Fire Dataset

For the model to be trained, the first task begins with collecting a diverse range of datasets. We collected fire images and videos from the internet. Due to a wide range of datasets helping to generalize the model, our obtained images collected distinct angles, focal lengths, and brightening conditions. Additionally, on the internet, there are some popular platforms such as Roboflow, Bing, Flickr images, and Kaggle to arrange images. To achieve more accurate results, we divided our dataset into two classes: fire and non-fire images. To train our dataset, we standardized the dimensions of all images by resizing them to the same height and width, thus minimizing potential errors. Our dataset comprised 7000 images captured during both the daytime and nighttime, which were subsequently compressed, as shown in Table 1.

Table 1. Images of the forest fire scenes from the custom dataset.

Dataset	Google, Bing, Kaggle, Flickr Images	Video Frames	Total
Forest Fire Images	4136	2864	7000

After collecting the dataset, we applied custom image pre-processing, dropped 1000 images from the dataset and left 6000 images. Figure 1 shows how we increased our dataset by rotating each image at 90° angles to 270° using a computer vision algorithm. After applying the augmentation technique, our dataset increased four times, and the total number of images extended to 24,000 images. Our dataset is divided as follows: 70% for training images, 10% for test images, and 20% for validation images, as shown in Table 2.

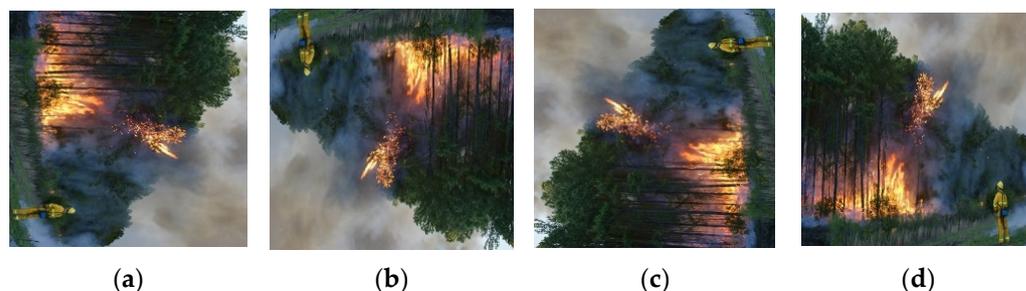
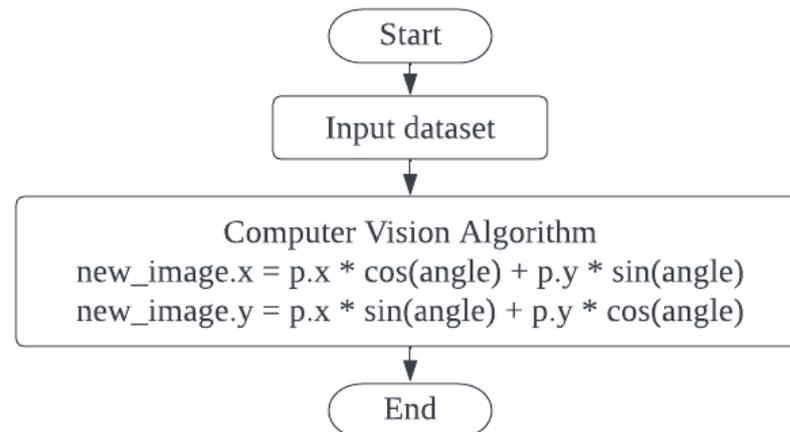


Figure 1. Sample images of forest fires rotated from various perspectives. (a) 90° rotation, (b) 180° rotation, (c) 270° rotation, and (d) the original image.

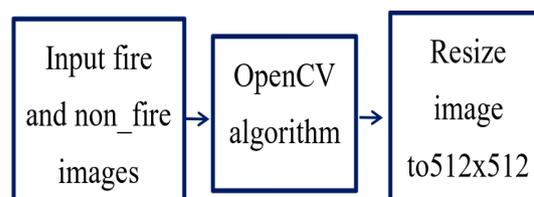
Table 2. Distribution of flame frames within the dataset.

Dataset	Training Images	Testing Images	Validation Images	Total
Fire	11,760	1680	3360	16,800
Non-Fire	5040	720	1440	7200

Scheme 1 shows the convergence of image sizes where x and y are the new image dimensions. P is the input image matrix, and using the multiplication of the angle, the expected image rotation takes place.

**Scheme 1.** Image processing (rotation).

After completing image augmentation, we labeled our images into two classes: fire and non-fire. According to image annotation, a JSON file was created for each image. Earlier in this section, we mentioned that we resized all the images to overcome unexpected errors. In our dataset, we resized our images to 512×512 using OpenCV2, as shown in Figure 2.

**Figure 2.** The overall process of resizing images.

3.2. Model Selection

YOLOv8 is the latest edition of the CNN dynamic object model with high accuracy in real-time. The entire image region is processed by a single neural network, after which it is divided into multiple components, and the potential bounding boxes and probability estimates are estimated for each component. The YOLOv8 network is the continuous improvement of YOLOv1 to YOLOv7. The YOLO model has a backbone that is a series of convolution layers pulling the pixels down at the different resolution sizes, and these features pass through a neck where they are pulled together and put into a head, which leads to an object detection process base loss matrix. The YOLOv8 model architecture is shown in Figure 3.

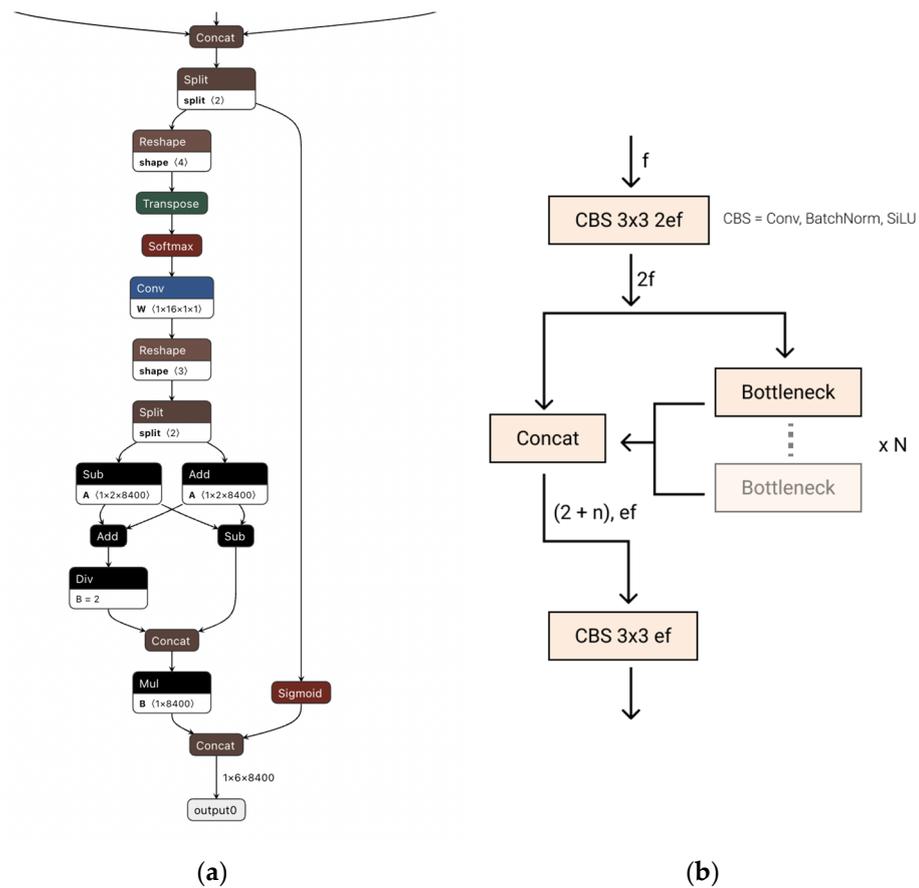


Figure 3. (a,b) are the overall architecture of the YOLOv8 model.

As shown in Figure 3, YOLOv8 is an anchor-free model. Instead of predicting the offset of an anchor box, it predicts the center of an object directly. With anchor-free detection, fewer box predictions are made, which speeds up Non-Maximum Suppression (NMS), which is a post-processing step used to sift through candidate detections. YOLOv8 uses RasNet as its head. The final detection stage used C2F; here, f is the number of features. CBS block consists of convolution, batch norm, and silu function. The final detection stage uses anchor boxes to show the probability of object detection.

The YOLOv8 network contains five different models. YOLOv8n is a good solution for mobile phone applications. YOLOv8s (small model) is compatible with the CPU. YOLOv8m is a medium-sized model that has 25.9 million parameters, which are balanced between speed and accuracy. YOLOv8l has 43.7 million parameters, and this model works best with numerous databases and training. Out of the five models, YOLOv8x is the largest and has the highest mean average precision; nevertheless, it performs slower than YOLOv8l. The relation between YOLOv8n and YOLOv8x is shown in Table 3.

Table 3. Relation between the YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x model [65].

Network	Size (Pixels)	aMP ^{val} (50–95)	Speed CPU (ms)	Speed T4 GPU (ms)	Params (M)	Flop _s (B)
YOLOv8n	640	37.3	-	-	3.2	8.7
YOLOv8s	640	44.9	-	-	11.2	28.6
YOLOv8m	640	50.2	-	-	25.9	78.9
YOLOv8l	640	52.9	-	-	43.7	165.2
YOLOv8x	640	53.9	-	-	68.2	257.8

3.3. Proposed Forest Fire Model

In this subsection, we elaborate on the methodologies employed in computer vision for forest detection, focusing on deep learning approaches, transfer learning techniques, and the aggregation of models.

3.3.1. Transfer Learning

Deep learning algorithms learn features from data, which provide additional support to a deep learning model to detect objects more accurately and fast. However, collecting data is expensive, may not be available, and annotation is a tedious task with high costs. Transfer learning allows one to learn from features and transfer model weight to another model for further learning.

First, we review traditional methods of object detection for transfer learning [66]. This paper proposes using the YOLO pre-trained weights and transfer learning features. As we mentioned earlier, we used 24,000 forest fire images in our dataset, and for the train, 16,800 images were used. We used the default YOLOv8 in all five models to train our dataset and show the result after 50,000 iterations in Table 4. In addition, image hue 0.1, saturation 1.5, and exposure 1.5 were used.

Table 4. Pre-trained weights obtained using a limited dataset.

Models	Input Size	Training Accuracy (AP50)	Testing Accuracy (AP50)	Weight Size	Iteration Number	Training Time
YOLOv8n		83.8%	81.8%			27 h
YOLOv8s		84.1%	82.9%			34 h
YOLOv8m	512 × 512	86.4%	84.8%	186 MB	50,000	38 h
YOLOv8l		91.7%	90.7%			43 h
YOLOv8x		87.1%	85.5%			48 h

Table 4 shows the results obtained for the training and testing accuracies with different indicators. YOLOv8l showed the highest training and test accuracy of 91.7% and 90.7% in 43 h, respectively. The following results were obtained from YOLOv8x with 87.1% and 85.5% accuracy. YOLOv8m and YOLOv8s had 86.4% and 84.1% training accuracy and 84.8% and 82.9% testing accuracy, respectively. YOLOv8n showed the lowest training accuracy at 83.8%; on the other hand, 81.8% was the testing accuracy with 27 h of training time. Moreover, our fire detection approach using YOLOv8 showed better accuracy with large-size forest fire detection but was not efficient with small-size forest fires. Images of forest fires, small-scale fires, and no fires may all be distinguished with the human eye. For deep learning, the method needs more information to improve prediction accuracy. Figure 4 shows the overall fire detection using the YOLOv8 object detection model. Here, the model shows insufficient accuracy in terms of small images but is compatible with large-size forest fire detection.



Figure 4. Big and small-size forest fire detection.

3.3.2. Detect Small-Size Image

From the previous section, detecting small fire images has some limitations in our model. To solve the accuracy factor, we drove forward the concept of the TranSDet [67] model. This model proposes a meta-learning-based dynamic resolution adaption transfer learning (DRAT) schema to adapt the pre-trained general model to detect small objects. The model includes an additional stage on the pre-trained model using DRAT and then transfers the model modified for the intended database. The pre-trained model only generates training on small objects, and to generalize for small objects, the data augmentation (resizing the input images to small images) technique is used to train the model. Fine-tune follow-up is used after the augmentation technique. Figure 5 shows the TranSDet architecture.

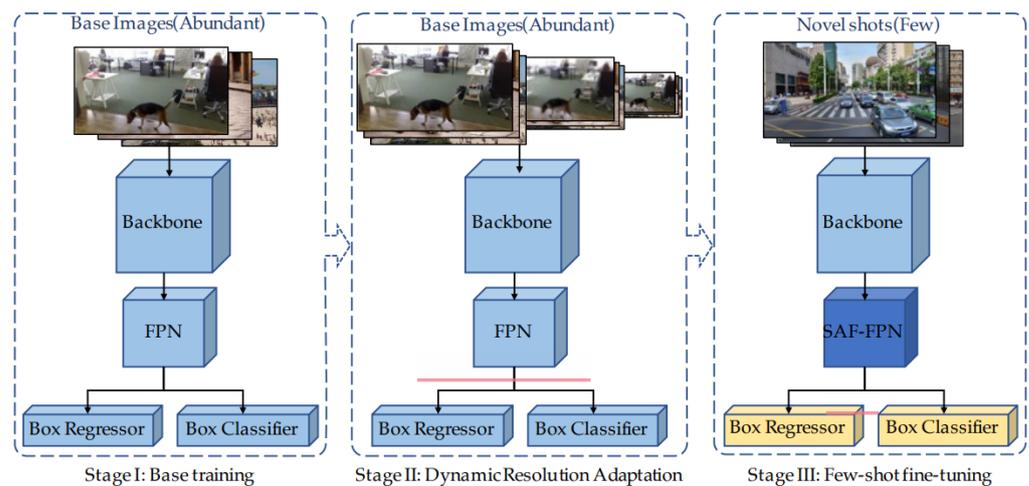


Figure 5. The overall architecture of the TranSDet model.

TranSDet model directly transfers conventional stage 1 and stage 3 pre-trained models using fine-tuning and implementing the dynamic resolution adaptation technique. Stage 2 adopts the pre-trained model to improve transfer learning to detect small objects.

$$\theta = \arg \min_{\theta} E_{R_i} L(D, R_i, M) \quad (1)$$

Here, R is a set of resolutions, and M is the model, and θ pre is the weight to generalize the dataset. In the equation, D represents the database, and L stands for the loss function. To address the meta-learning problem, we employed the widely used MAML (Model Agnostic Meta-Learning) model [68].

$$\theta'_i = \theta - \alpha \nabla_{\theta} L_{R_i}(M(\theta; X_i)) \quad (2)$$

$$\theta_{new} = \theta - \beta \nabla_{\theta} \sum_{R_i} L_{R_i}(M(\theta'_i; X_i)) \quad (3)$$

Here, Equation (2) θ'_i is updated by θ , X_i is the mini batch of images, R_i is the resolution of set R and α is the step size. From Equation (2), DRAT performs an inner update. In Equation (3), θ new is the next iteration. When the model is complete, the final epoch of the model training is complete.

After understanding the mathematical equation, we started building the model and obtained the final output, as shown in Figure 6. After implementing the TranSDet model, we obtained a maximum accuracy of 95%.



Figure 6. Prediction after implementing the TranSDet model.

3.3.3. Model Aggregation

Model aggregation in deep learning refers back to the method of combining the predictions or outputs of a couple of common neural networks to make a final prediction or choice. This technique is regularly used to enhance the overall performance and robustness of gadget learning. There are several techniques for model aggregation, such as boosting, bagging, stacking, averaging, voting, and so on. Our proposed model used the boosting technique (the amalgamation of multiple weak models, each exhibiting slightly superior performance to random guessing, culminating in the formation of a robust learner).

Boosting is an ensemble learning technique that merges multiple weak learners, often decision trees, to construct a formidable learner. Its fundamental concept revolves around assigning greater weight to misclassified instances from the preceding iteration, which helps the ensemble focus on the difficult instances. The equations involved in boosting include the following:

The weighted error (ϵ_t) for the t -th weak learner is calculated as follows:

$$\epsilon_t = \text{Sum of weights of misclassified examples} / \text{Total weight of all examples.}$$

The weight (α_t) for the t-th weak learner is calculated as follows:

$$\alpha_t = 0.5 * \ln((1 - \varepsilon_t) / \varepsilon_t)$$

The weights are updated based on whether they are correctly or incorrectly classified by the t-th weak learner:

$$\text{If correctly classified: } w_{i_t}(t + 1) = w_{i_t} * e^{-\alpha_t}$$

$$\text{If misclassified: } w_{i_t}(t + 1) = w_{i_t} * e^{\alpha_t}$$

The final prediction is a weighted combination of the weak learners' outputs as follows: Final Prediction = Sign($\sum(\alpha_t * \text{Output of Weak Learner}_t)$).

These equations are used iteratively in boosting algorithms like AdaBoost and Gradient Boosting to create a strong ensemble model from weak learners.

Figure 7 shows our proposed simple model diagram. As mentioned earlier, we prepared our dataset and used YOLOv8l pre-train model transfer model learning and TranSDet model learning transfer and aggregated both modes to detect small and large forest fires using the boosting technique.

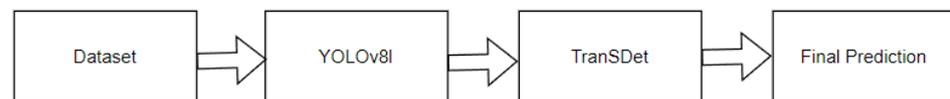


Figure 7. Our proposed model workflow.

After applying the boosting method to our dataset, the accuracy increased to 97% and a small fire to 96%, respectively. Furthermore, the inclusion of fire-like images enhanced our model's accuracy, as shown in Figure 8. Finally, our proposed model was implemented on Raspberry PI 3B+, as shown in Figure 9. The suggested approach employed two different models and used the transfer learning technique, and the model achieved 97% accuracy performance.



Figure 8. Cont.



Figure 8. Prediction after implementing boosting technique on the YOLOv8l and TranSDet model. The first row displays the result of detecting a small fire, the second row shows the fire-like image level to represent no fire, and the third row shows the detection of a large fire.

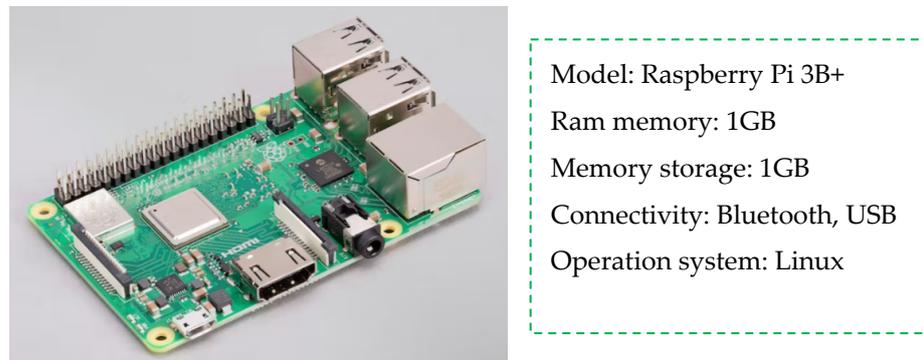


Figure 9. Characteristics of Raspberry Pi 3B+ [12].

To evaluate the performance of our suggested model, we juxtapose it with an established model in Table 5. However, the Results and Discussion sections show an explanation of these observations in detail.

Table 5. Comparison between different models.

Features	YOLOv8l	TranSDet	Our Method (Model Aggregation)
Test speed/s	2 s	2.3 s	4.5 s
Real-time implementation	Possible	Possible	Possible
Small object detection	Possible (but not sufficient)	possible(shows better output)	Possible (highly accurate)
Algorithm	Selective search	Selective search	Selective search

4. Results and Discussions from the Experimentations

Test with Fire and Non-Fire Image

We evaluated our model implementing the Visual Studio code on our MSI GS66 laptop (MSI, Taipei, Taiwan), equipped with a CPU speed of 5.3 GHz, 64 GB of RAM, and 6 GPUs. In the previous section, we discussed and implemented our proposed model and aggregated the YOLOv8l and TranSDet models. In this subsection, we review and discuss our model's advantages and drawbacks. Traditionally, the YOLO model is known for real-time fire detection with high accuracy. However, when we applied the model to our custom dataset, small fire images were provided with insufficient accuracy. To improve our model, we applied another model called TranSDet, which provides a high accuracy of up to 96%. In our proposed model, the boosting technique was used to aggregate both models,

and our model provided accuracy of up to 97% and 96% with large and small forest fires, respectively, as shown in Figures 10 and 11.

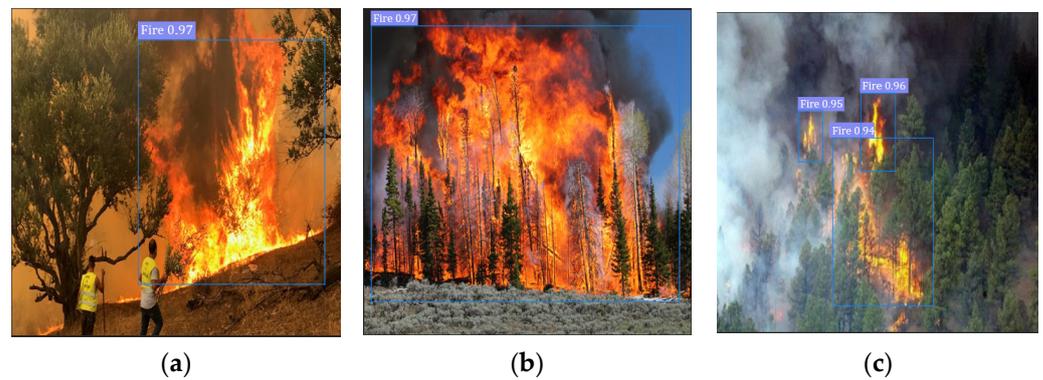


Figure 10. (a–c) Outcomes of daytime image detection accuracies of forest fires.

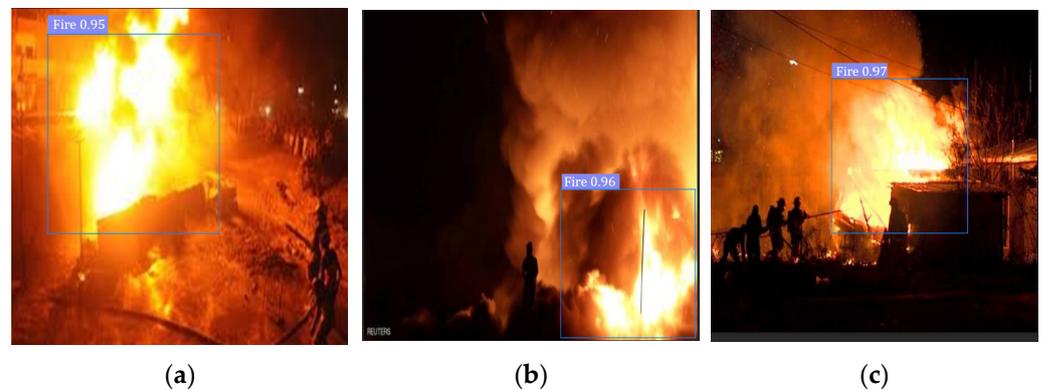


Figure 11. (a–c) Outcomes of nighttime image detection accuracies of forest fires.

Figure 12 shows the training and testing accuracy with loss throughout passing the epoch to train the model. At the beginning of training, our model loss was at its maximum at 0.5. However, after completing the model training process, the loss was minimal at 0.11. On the other hand, for initial testing, it was 0.9, and at the very end, it was 0.1. As we mentioned earlier, for training accuracy, the model reached 96.7% training accuracy and a testing accuracy of 97%.

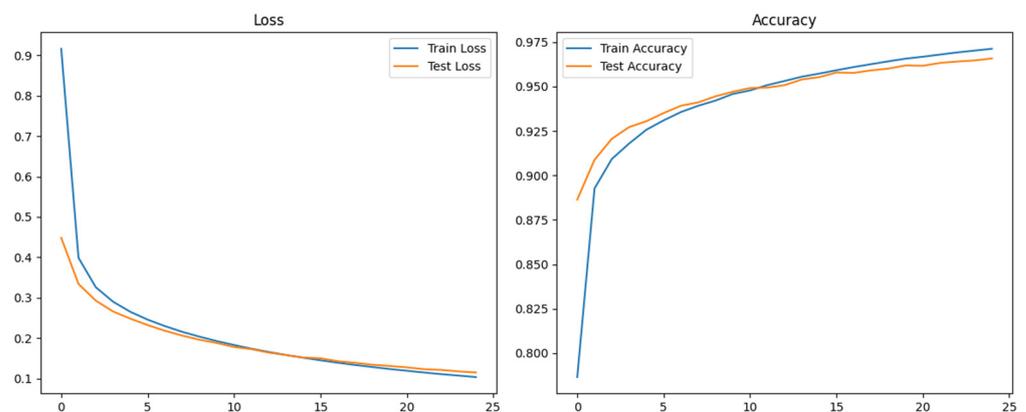


Figure 12. Model training and testing loss and accuracy visualization with epoch.

In this section, the compression of our proposed approach is discussed utilizing various parameters and models. Our model was developed in three stages. First, we used

pre-train YOLOv8l and then the TransDet model to detect small forest fires, and we reached 97% accuracy. F-measures (FM) were employed to compute the weighted average, which balances precision and recall. The true-positive and false-negative rates were taken into consideration when calculating this score. In order to detect an object, the FM parameter was most commonly used since measuring the accuracy rate was difficult. True positives and false negatives were more effective in a detection model with identical weights. If true positives and false negatives differ, precision and recall need to be taken into account. True positive observations are measured by precision.

Recall, on the other hand, is the ratio of false positives to true positives, as described in previous studies [69,70]. Our developed system had a precision of 97% and a false detection rate of 0.7%. As shown in Equations (4) and (5), our proposed model had an average precision and recall rate of 97% and 3%, respectively. TP refers to the accurate detection of a forest fire, while FP refers to a false negative detection (Figure 13).

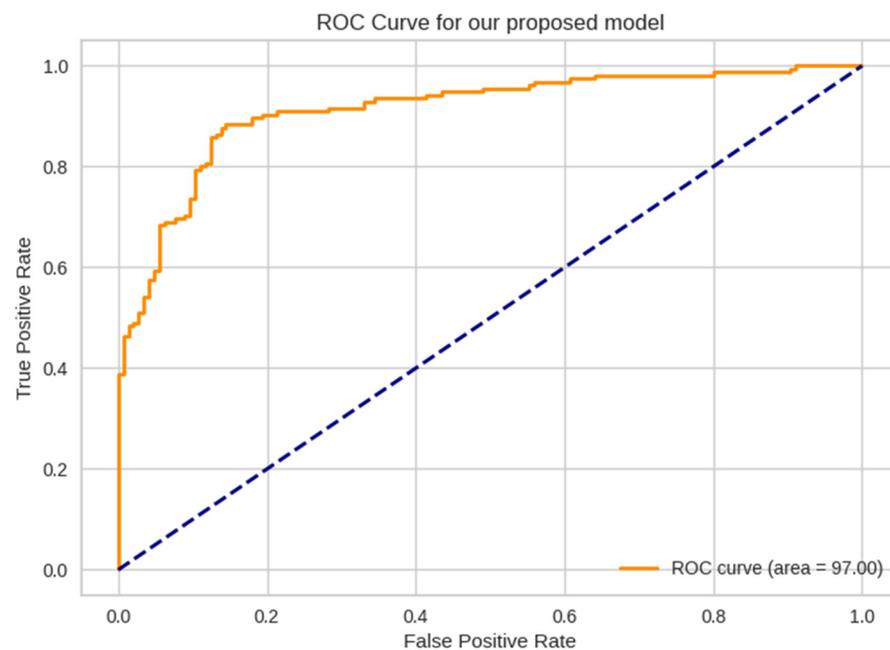


Figure 13. ROC curve of our proposed model.

Equation (6) shows the relation between precision and recall.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4)$$

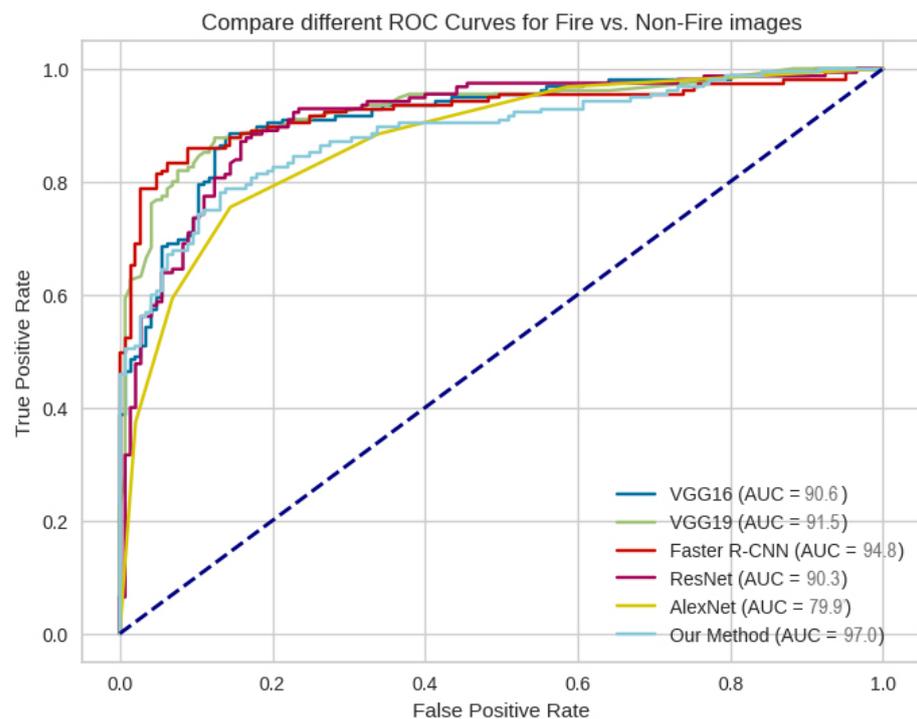
$$\text{Recall} = \frac{TP}{TP + FN} \quad (5)$$

$$\text{FM} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (6)$$

Forest fire detection is a complex task in the field of deep learning to achieve high accuracy. Table 6 shows recently published fire detection models with precision, recall, and FM. Here, our model reached the highest precision, recall, and FM with 97%, 96.1%, and 96.5, respectively. This was followed by VGG16 and RsNet with a precision of 92.5% and 90.8%, recall of 82.9% and 98.6%, and finally 90.8% and 90.2%, respectively. However, AlexNet was the poorest in terms of precision, recall, and FM at 73.3%, 61.3%, and 75.1%, respectively (Figure 14).

Table 6. Numerical outcomes for the detection of fire.

Algorithm	<i>P</i> (%)	<i>R</i> (%)	<i>FM</i> (%)	Average (%)
VGG16	92.5	82.9	90.8	90.6
VGG19	93.1	84.5	91.7	91.5
Faster R-CNN [55]	81.7	94.5	87.2	97.8
ResNet [56]	90.8	89.6	90.2	90.3
AlexNet	73.3	61.3	75.1	79.9
Our Method	97	96.1	96.3	96.5

**Figure 14.** Comparison of different models using ROC for the fire and non-fire images.

5. Discussion

As stated in Table 6, the categorization of a model as good or bad is contingent upon specific criteria rather than its overall performance. Furthermore, a model shows high accuracy in some specific tasks, but depending on the complexity model can have some limitations. Our proposed model has a couple of limitations. First, our dataset did not include smoke images. Therefore, if only smoke was visible at the initial stage of the forest fire, the model did not detect this as fire. Next, occasionally, the model considered sun and electric light as forest fires when testing the method in various scenes. In the future, we intend to enhance the developed system by incorporating a database encompassing additional classes from diverse environments pertinent to this challenge [71]. Our methodology involved utilizing extensive datasets, like JFT-300M [72], comprising 300 million annotated images.

Despite the aforementioned challenges, the main contributions of this study are as follows:

1. The pre-trained YOLOv8 model and transferring the learning can detect large-size forest fires. The YOLOv8 algorithm is known for its speed and ability to perform object detection in real-time.
2. To detect small-size fires, the TransDet model and transfer learning approaches can be applied. Utilizing deep learning to acquire fire-specific features, the presented

methodology has the potential to mitigate the prevalent issue of false alarms in conventional fire detection methods. Such an advancement stands to not only prevent unwarranted emergency responses but also to alleviate the financial burden attributed to false alarms.

- Both models can be aggregated with boosting techniques to detect forest fires. The goal of this research was to apply deep learning models in the field of forest fire prevention. Early detection with high accuracy is beneficial for environmental safety.
- In contrast to alternative approaches that rely on limited datasets, our method leverages a substantial dataset encompassing fire, fire-like, and standard scenes. This dataset comprises authentic imagery and videos sourced from diverse origins, thereby encapsulating a broad spectrum of fire scenarios. These scenarios encompass both day and night fire incidents, spanning variations in fire scale and accounting for varying lighting conditions, including low-light and high-light environments.

Future efforts will aim to overcome the model's limitation of yielding a high number of false positives, particularly in challenging scenarios like low-altitude cloud cover and haze. Enhancements could involve integrating historical fire record data on fire location, date, and weather conditions, as fires often occur in similar contexts during specific months, thereby improving prediction accuracy. Additionally, the current approach's incompatibility with edge devices presents a drawback. Nonetheless, we plan to address this issue in upcoming research by optimizing the model size while maintaining prediction accuracy. One potential avenue is to employ distillation techniques to train a smaller deep network, such as YOLOv9, which is capable of achieving comparable performance to our current model, thus making it more suitable for edge computing environments.

6. Conclusions

Day-to-day life unicorn tech companies work with big data. Aligning AI to mimic human operations can secure human life in such a way as preventing life-threatening tasks and better service. For object detection, numerous models have been developed based on deep-learning CNN models. Forest fire detection using YOLOv is nothing new and provides high accuracy. In the first proposed model, we collected the dataset and conducted some preprocessing tasks. Our model architecture is based on three stages. First, use the transfer technique on the YOLOv8 pre-train model to detect large-size forest fires. Next, to detect small-size forest fires in real-time, the TransDet model technique was transferred to the learning of our model. Lastly, both transfers were fed learning to the boosting algorithm to train weak learners and show high accuracy.

Several investigations have focused on enhancing forest flame identification and classification tasks through CNN-based AI networks. Nevertheless, the potential of the Detectron2 network remains unexplored in forest fire detection. The acquisition of ample image data for training forest fire detection models presents challenges, often resulting in issues such as data imbalance or overfitting, which can hinder the efficacy of the model. In this study, we introduce a method for forest fire detection utilizing the enhanced Detectron2 model and develop a dataset to address these challenges.

After achieving 97% tested accuracy, our model was implemented in Raspberry Pi 3B+ to run on GPU mode. Furthermore, after testing our model in the variant environment, we detected some limitations in real-time frameworks, such as no data related to smoke frames; therefore, fume detection was not included in our model. In the future, we aim to develop a model that can concentrate on healthcare using 3D CNN/U-Net [73–75].

Author Contributions: Conceptualization, N.Y. and B.M.S.I.; Formal analysis, W.K.; Algorithms: N.Y.; Funding acquisition, W.K.; Investigation, B.M.S.I. and N.Y.; Methodology, A.A. and N.Y.; Project administration, W.K.; Resources, A.A. and N.Y.; Software, B.M.S.I.; Supervision, W.K.; Validation, W.K.; Writing—original draft, N.Y. and B.M.S.I.; Writing—review and editing W.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Gachon University Research Fund (GCU-202300740001), the Ministry of Education of the Republic of Korea, and the National Research Foundation of Korea (NRF-2022S1A5C2A07090938).

Data Availability Statement: The data are contained within the article.

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

References

1. Nelson, R. Untamedscience.com. April 2019. Available online: <https://untamedscience.com/blog/the-environmental-impact-of-forest-fires/> (accessed on 30 December 2023).
2. Jain, P.; Coogan, S.C.; Subramanian, S.G.; Crowley, M.; Taylor, S.; Flannigan, M.D. A review of machine learning applications in wildfire science and management. *Environ. Rev.* **2020**, *28*, 478–505. [CrossRef]
3. Milne, M.; Clayton, H.; Dovers, S.; Cary, G.J. Evaluating benefits and costs of wildland fires: Critical review and future applications. *Environ. Hazards* **2014**, *13*, 114–132. [CrossRef]
4. Varma, S.; Sreeraj, M. Object detection and classification in surveillance system. In Proceedings of the 2013 IEEE Recent Advances in Intelligent Computational Systems (RAICS), Trivandrum, India, 19–21 December 2013; pp. 299–303. [CrossRef]
5. Terradas, J.; Pinol, J.; Lloret, F. Climate warming, wildfire hazard, and wildfire occurrence in coastal eastern Spain. *Clim. Chang.* **1998**, *38*, 345–357.
6. Alkhatib, A.A. A review on forest-fire detection techniques. *Int. J. Distrib. Sens. Netw.* **2014**, *10*, 597368. [CrossRef]
7. Xavier, K.L.B.L.; Nanayakkara, V.K. Development of an Early Fire Detection Technique Using a Passive Infrared Sensor and Deep Neural Networks. *Fire Technol.* **2022**, *58*, 3529–3552. [CrossRef]
8. 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]
9. Karthi, M.; Priscilla, R.; Subhashini, G.; Abijith, G.R.; Vinisha, J. Forest fire detection: A comparative analysis of deep learning algorithms. In Proceedings of the 2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF), Chennai, India, 5–7 January 2023.
10. Kaur, P.; Kaur, K.; Singh, K.; Kim, S. Early Forest Fire Detection Using a Protocol for Energy-Efficient Clustering with Weighted-Based Optimization in Wireless Sensor Networks. *Appl. Sci.* **2023**, *13*, 3048. [CrossRef]
11. Mijwil, M.M. *History of Artificial Intelligence*; 2015; Volume 3, pp. 1–8. [CrossRef]
12. Xiao, L.; Yan, Q.; Deng, S. Scene classification with improved AlexNet model. In Proceedings of the 2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), Nanjing, China, 24–26 November 2017; pp. 1–6. [CrossRef]
13. Tammina, S. Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images. *Int. J. Sci. Res. Publ. (IJSRP)* **2019**, *9*, 143–150. [CrossRef]
14. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448. [CrossRef]
15. Abdusalomov, A.B.; Islam, B.M.S.; Nasimov, R.; Mukhiddinov, M.; Whangbo, T.K. An Improved Forest Fire Detection Method Based on the Detectron2 Model and a Deep Learning Approach. *Sensors* **2023**, *23*, 1512. [CrossRef]
16. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *arXiv* **2015**, arXiv:1506.02640.
17. Alkhatib, R.; Sahwan, W.; Alkhatieb, A.; Schütt, B. A Brief Review of Machine Learning Algorithms in Forest Fires Science. *Appl. Sci.* **2023**, *13*, 8275. [CrossRef]
18. Jayasingh, S.K.; Swain, S.; Patra, K.J.; Gountia, D. An Experimental Approach to Detect Forest Fire Using Machine Learning Mathematical Models and IoT. *SN Comput. Sci.* **2024**, *5*, 148. [CrossRef]
19. Rehman, A.; Kim, D.; Paul, A. Convolutional neural network model for fire detection in real-time environment. *Comput. Mater. Contin.* **2023**, *77*, 2289–2307. [CrossRef]
20. Ghali, R.; Akhloufi, M.A. Deep Learning Approaches for Wildland Fires Using Satellite Remote Sensing Data: Detection, Mapping, and Prediction. *Fire* **2023**, *6*, 192. [CrossRef]
21. Keeping, T.; Harrison, S.P.; Prentice, I.C. Modelling the daily probability of wildfire occurrence in the contiguous United States. *Environ. Res. Lett.* **2024**, *19*, 024036. [CrossRef]
22. Li, Y.; Xu, S.; Fan, Z.; Zhang, X.; Yang, X.; Wen, S.; Shi, Z. Risk Factors and Prediction of the Probability of Wildfire Occurrence in the China–Mongolia–Russia Cross-Border Area. *Remote Sens.* **2023**, *15*, 42. [CrossRef]
23. Villaverde Canosa, I.; Ford, J.; Paavola, J.; Burnasheva, D. Community Risk and Resilience to Wildfires: Rethinking the Complex Human–Climate–Fire Relationship in High-Latitude Regions. *Sustainability* **2024**, *16*, 957. [CrossRef]
24. Marey-Perez, M.; Loureiro, X.; Corbelle-Rico, E.J.; Fernández-Filgueira, C. Different Strategies for Resilience to Wildfires: The Experience of Collective Land Ownership in Galicia (Northwest Spain). *Sustainability* **2021**, *13*, 4761. [CrossRef]
25. Myagmar-Ochir, Y.; Kim, W. A survey of Video Surveillance Systems in Smart City. *Electronics* **2023**, *12*, 3567. [CrossRef]
26. 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]

27. Giglio, L.; Boschetti, L.; Roy, D.P.; Humber, M.L.; Justice, C.O. The Collection 6 MODIS burned area mapping algorithm and product. *Remote Sens. Environ.* **2018**, *217*, 72–85. [[CrossRef](#)] [[PubMed](#)]
28. Ba, R.; Chen, C.; Yuan, J.; Song, W.; Lo, S. SmokeNet: Satellite Smoke Scene Detection Using Convolutional Neural Network with Spatial and Channel-Wise Attention. *Remote Sens.* **2019**, *11*, 1702. [[CrossRef](#)]
29. Larsen, A.; Hanigan, I.; Reich, B.J.; Qin, Y.; Cope, M.; Morgan, G.; Rappold, A.G. A deep learning approach to identify smoke plumes in satellite imagery in near-real time for health risk communication. *J. Expo. Sci. Environ. Epidemiol.* **2021**, *31*, 170–176. [[CrossRef](#)]
30. Avazov, K.; Mukhiddinov, M.; Makhmudov, F.; Cho, Y.I. Fire Detection Method in Smart City Environments Using a Deep-Learning-Based Approach. *Electronics* **2022**, *11*, 73. [[CrossRef](#)]
31. Mukhiddinov, M.; Cho, J. Smart Glass System Using Deep Learning for the Blind and Visually Impaired. *Electronics* **2021**, *10*, 2756. [[CrossRef](#)]
32. Toulouse, T.; Rossi, L.; Celik, T.; Akhloufi, M. Automatic fire pixel detection using image processing: A comparative analysis of rule-based and machine learning-based methods. *Signal Image Video Process.* **2016**, *10*, 647–654. [[CrossRef](#)]
33. Jiang, Q.; Wang, Q. Large space fire image processing of improving canny edge detector based on adaptive smoothing. In Proceedings of the 2010 International Conference on Innovative Computing and Communication and 2010 Asia-Pacific Conference on Information Technology and Ocean Engineering, Macao, China, 30–31 January 2010; pp. 264–267.
34. Celik, T.; Demirel, H.; Ozkaramanli, H.; Uyguroglu, M. Fire detection using statistical color model in video sequences. *J. Vis. Commun. Image Represent.* **2007**, *18*, 176–185. [[CrossRef](#)]
35. Dimitropoulos, K.; Barmpoutis, P.; Grammalidis, N. Spatio temporal flame modeling and dynamic texture analysis for automatic video-based fire detection. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 339–351. [[CrossRef](#)]
36. Geng, X.; Su, Y.; Cao, X.; Li, H.; Liu, L. YOLOFM: An improved fire and smoke object detection algorithm based on YOLOv5n. *Sci. Rep.* **2024**, *14*, 4543. [[CrossRef](#)]
37. Li, P.; Zhao, W. Image fire detection algorithms based on convolutional neural networks. *Case Stud. Therm. Eng.* **2020**, *19*, 100625. [[CrossRef](#)]
38. Valikhujaev, Y.; Abdusalomov, A.; Cho, Y.I. Automatic Fire and Smoke Detection Method for Surveillance Systems Based on Dilated CNNs. *Atmosphere* **2020**, *11*, 1241. [[CrossRef](#)]
39. Li, T.; Zhao, E.; Zhang, J.; Hu, C. Detection of Wildfire Smoke Images Based on a Densely Dilated Convolutional Network. *Electronics* **2019**, *8*, 1131. [[CrossRef](#)]
40. Kutlimuratov, A.; Khamzaev, J.; Kuchkorov, T.; Anwar, M.S.; Choi, A. Applying Enhanced Real-Time Monitoring and Counting Method for Effective Traffic Management in Tashkent. *Sensors* **2023**, *23*, 5007. [[CrossRef](#)] [[PubMed](#)]
41. 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 (SCID), Hangzhou, China, 8–9 December 2018; pp. 280–284.
42. Kim, B.; Lee, J. A video-based fire detection using deep learning models. *Appl. Sci.* **2019**, *9*, 2862. [[CrossRef](#)]
43. Zhao, L.; Liu, J.; Peters, S.; Li, J.; Oliver, S.; Mueller, N. Investigating the Impact of Using IR Bands on Early Fire Smoke Detection from Landsat Imagery with a Lightweight CNN Model. *Remote Sens.* **2022**, *14*, 3047. [[CrossRef](#)]
44. Zhao, Y.Y.; Zhu, J.; Xie, Y.K.; Li, W.L.; Guo, Y.K. Improved Yolo-v3 Video Image Flame Real-Time Detection Algorithm. *J. Wuhan Univ. Inf. Sci. Ed.* **2021**, *46*, 326–334.
45. Abdusalomov, A.; Baratov, N.; Kutlimuratov, A.; Whangbo, T.K. An Improvement of the Fire Detection and Classification Method Using YOLOv3 for Surveillance Systems. *Sensors* **2021**, *21*, 6519. [[CrossRef](#)]
46. Park, M.; Ko, B.C. Two-Step Real-Time Night-Time Fire Detection in an Urban Environment Using Static ELASTIC-YOLOv3 and Temporal Fire-Tube. *Sensors* **2020**, *20*, 2202. [[CrossRef](#)] [[PubMed](#)]
47. Mukhiddinov, M.; Abdusalomov, A.B.; Cho, J. Automatic Fire Detection and Notification System Based on Improved YOLOv4 for the Blind and Visually Impaired. *Sensors* **2022**, *22*, 3307. [[CrossRef](#)]
48. Talaat, F.M.; ZainEldin, H. An improved fire detection approach based on YOLO-v8 for smart cities. *Neural Comput. Appl.* **2023**, *35*, 20939–20954. [[CrossRef](#)]
49. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. *arXiv* **2022**, arXiv:2209.02976.
50. Wang, C.; Bochkovskiy, A.; Liao, H. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
51. Shi, P.; Lu, J.; Wang, Q.; Zhang, Y.; Kuang, L.; Kan, X. An Efficient Forest Fire Detection Algorithm Using Improved YOLOv5. *Forests* **2023**, *14*, 2440. [[CrossRef](#)]
52. Reis, D.; Kupec, J.; Hong, J.; Daoudi, A. Real-Time Flying Object Detection with YOLOv8. *arXiv* **2023**, arXiv:2305.09972.
53. Saydirasulovich, S.N.; Mukhiddinov, M.; Djuraev, O.; Abdusalomov, A.; Cho, Y.-I. An Improved Wildfire Smoke Detection Based on YOLOv8 and UAV Images. *Sensors* **2023**, *23*, 8374. [[CrossRef](#)]
54. Girdhar, R.; Carreira, J.; Doersch, C.; Zisserman, A. Video Action Transformer Network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 9–15 June 2019; pp. 244–253.
55. Yang, F.; Yang, H.; Fu, J.; Lu, H.; Guo, B. Learning Texture Transformer Network for Image Super-Resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 5791–5800.

56. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. In *Computer Vision—ECCV*; Springer International Publishing: Cham, Switzerland, 2020; pp. 213–229.
57. Ye, L.; Rochan, M.; Liu, Z.; Wang, Y. Cross-Modal Self-Attention Network for Referring Image Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 9–15 June 2019; pp. 10502–10511.
58. He, X.; Chen, Y.; Lin, Z. Spatial-Spectral Transformer for Hyperspectral Image Classification. *Remote Sens.* **2021**, *13*, 498. [[CrossRef](#)]
59. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.
60. Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; Jégou, H. Training data-efficient image transformers & distillation through attention. *arXiv* **2020**, arXiv:2012.12877.
61. Valanarasu, J.M.J.; Oza, P.; Hacihaliloglu, I.; Patel, V.M. Medical Transformer: Gated Axial-Attention for Medical Image Segmentation. *arXiv* **2021**, arXiv:2102.10662.
62. Abdusalomov, A.B.; Mukhiddinov, M.; Kutlimuratov, A.; Whangbo, T.K. Improved Real-Time Fire Warning System Based on Advanced Technologies for Visually Impaired People. *Sensors* **2022**, *22*, 7305. [[CrossRef](#)]
63. Pandey, B.; Pandey, D.K.; Mishra, B.P.; Rhmann, W. A comprehensive survey of deep learning in the field of medical imaging and medical natural language processing: Challenges and research directions. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *34*, 5083–5099. [[CrossRef](#)]
64. Mukhiddinov, M.; Djuraev, O.; Akhmedov, F.; Mukhamadiyev, A.; Cho, J. Masked Face Emotion Recognition Based on Facial Landmarks and Deep Learning Approaches for Visually Impaired People. *Sensors* **2023**, *23*, 1080. [[CrossRef](#)] [[PubMed](#)]
65. Jocher, G.; Chaurasia, A.; Qiu, J. *Ultralytics YOLO*; Version 8.0.0; Ultralytics: Los Angeles, CA, USA, 2023. Available online: <https://github.com/ultralytics/ultralytics> (accessed on 12 January 2024).
66. Wang, X.; Huang, T.; Gonzalez, J.; Darrell, T.; Yu, F. Frustratingly Simple Few-Shot Object Detection. In Proceedings of the 37th International Conference on Machine Learning, Virtual, 13–18 July 2020; pp. 9919–9928.
67. Xu, X.; Zhang, H.; Ma, Y.; Liu, K.; Bao, H.; Qian, X. TransSDet: Toward Effective Transfer Learning for Small-Object Detection. *Remote Sens.* **2023**, *15*, 3525. [[CrossRef](#)]
68. Finn, C.; Abbeel, P.; Levine, S. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1126–1135.
69. Avazov, K.; Jamil, M.K.; Muminov, B.; Abdusalomov, A.B.; Cho, Y.-I. Fire Detection and Notification Method in Ship Areas Using Deep Learning and Computer Vision Approaches. *Sensors* **2023**, *23*, 7078. [[CrossRef](#)] [[PubMed](#)]
70. Norkobil Saydirasulovich, S.; Abdusalomov, A.; Jamil, M.K.; Nasimov, R.; Kozhamzharova, D.; Cho, Y.-I. A YOLOv6-Based Improved Fire Detection Approach for Smart City Environments. *Sensors* **2023**, *23*, 3161. [[CrossRef](#)] [[PubMed](#)]
71. Ergasheva, A.; Akhmedov, F.; Abdusalomov, A.; Kim, W. Advancing Maritime Safety: Early Detection of Ship Fires through Computer Vision, Deep Learning Approaches, and Histogram Equalization Techniques. *Fire* **2024**, *7*, 84. [[CrossRef](#)]
72. Sun, C.; Shrivastava, A.; Singh, S.; Gupta, A. Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 843–852.
73. Chen, W.-F.; Ou, H.-Y.; Liu, K.-H.; Li, Z.-Y.; Liao, C.-C.; Wang, S.-Y.; Huang, W.; Cheng, Y.-F.; Pan, C.-T. In-Series U-Net Network to 3D Tumor Image Reconstruction for Liver Hepatocellular Carcinoma Recognition. *Diagnostics* **2021**, *11*, 11. [[CrossRef](#)] [[PubMed](#)]
74. Shah, S.M.; Sun, Z.; Zaman, K.; Hussain, A.; Ullah, I.; Ghadi, Y.Y.; Khan, M.A.; Nasimov, R. Advancements in Neighboring-Based Energy-Efficient Routing Protocol (NBEER) for Underwater Wireless Sensor Networks. *Sensors* **2023**, *23*, 6025. [[CrossRef](#)]
75. Aldughayfiq, B.; Ashfaq, F.; Jhanjhi, N.Z.; Humayun, M. YOLO-Based Deep Learning Model for Pressure Ulcer Detection and Classification. *Healthcare* **2023**, *11*, 1222. [[CrossRef](#)]

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