fire

MDPI

Article

An Optimized Deep-Learning-Based Network with an Attention Module for Efficient Fire Detection

Muhammad Altaf [1], Muhammad Yasir [2], Naqqash Dilshad [3] and Wooseong Kim [2,*]

[1] Department of Semiconductor, Gachon University, Sujeong-Gu, Seongnam-si 13120, Republic of Korea; altafkhann@gachon.ac.kr
[2] Department of Computer Engineering, Gachon University, Sujeong-Gu, Seongnam-si 13120, Republic of Korea; yasir95@gachon.ac.kr
[3] Department of Computer Science and Engineering, Sejong University, Seoul 05006, Republic of Korea; dilshad.naqqash@sejong.ac.kr or dilshad.naqqash@gmail.com
* Correspondence: wooseong@gachon.ac.kr

Abstract: Globally, fire incidents cause significant social, economic, and environmental destruction, making early detection and rapid response essential for minimizing such devastation. While various traditional machine learning and deep learning techniques have been proposed, their detection performances remain poor, particularly due to low-resolution data and ineffective feature selection methods. Therefore, this study develops a novel framework for accurate fire detection, especially in challenging environments, focusing on two distinct phases: preprocessing and model initializing. In the preprocessing phase, super-resolution is applied to input data using LapSRN to effectively enhance the data quality, aiming to achieve optimal performance. In the subsequent phase, the proposed network utilizes an attention-based deep neural network (DNN) named Xception for detailed feature selection while reducing the computational cost, followed by adaptive spatial attention (ASA) to further enhance the model's focus on a relevant spatial feature in the training data. Additionally, we contribute a medium-scale custom fire dataset, comprising high-resolution, imbalanced, and visually similar fire/non-fire images. Moreover, this study conducts an extensive experiment by exploring various pretrained DNN networks with attention modules and compares the proposed network with several state-of-the-art techniques using both a custom dataset and a standard benchmark. The experimental results demonstrate that our network achieved optimal performance in terms of precision, recall, F1-score, and accuracy among different competitive techniques, proving its suitability for real-time deployment compared to edge devices.

Keywords: fire disaster; deep learning; machine learning; surveillance system

Academic Editors: Lizhong Yang, Giovanni Laneve and Darko Stipanicev

Received: 17 November 2024
Revised: 18 December 2024
Accepted: 27 December 2024
Published: 2 January 2025

Citation: Altaf, M.; Yasir, M.; Dilshad, N.; Kim, W. An Optimized Deep-Learning-Based Network with an Attention Module for Efficient Fire Detection. *Fire* **2025**, *8*, 15. https://doi.org/10.3390/fire8010015

1. Introduction

Globally, fire disasters are considered the most widespread destructive hazards, with severe consequences for properties, environments, and human life. Fire can ignite in both human-made and natural environments due to equipment failure, climate variation, human activities, and elevated temperatures, having substantial consequences in several fields such as forest zones, urban regions, factories, and residential buildings. Promptly overcoming a fire disaster is exceptionally challenging, particularly in the early stages, when fires can occur in regions filled with highly flammable materials, such as residential areas and forest woodlands. Overall, wildfires, such as bushfires and forest fires, cause significant devastation due to their rapid propagation, leading to substantial environmental destruction. For

instance, studies have reported that a bushfire that occurred in Australia in 2020 during January and March had substantial consequences, such as the deaths of 33 individuals and approximately 1.5 billion animals, and the destruction of over 3000 houses [1–6].

Similarly, another study claimed that fire in California caused significant loss of life and extensive damage to properties [7,8]. Fires in residential buildings and vehicles pose major risks to human safety and assets. A 2018 study reported that, from 1993 to 2016, approximately 4.5 million fire incidents occurred across more than 50 nations, leading to approximately 62,000 deaths [9]. Between 2009 and 2015 in China, the average number of vehicle fires per year was 20,000, imposing an annual financial toll reported to be around CNY 370 million [10]. Additionally, in the United States in 2019, there were approximately 189,500 highway vehicle fires, resulting in 550 fatalities [11]. Hence, utilizing smart technology, including various vision-sensor-based algorithms and leveraging edge devices, can significantly reduce fire risk in the initial stages, with the aim of mitigating the various consequences of fire disasters.

Fire detection approaches based on vision sensors can be broadly divided into two distinct categories: machine learning (ML) and deep learning (DL). ML-based techniques place significant emphasis on determining color, shape, and texture characteristics in the input data, as discussed in [12,13]. ML approaches heavily depend on manually crafted features, although choosing the relevant features is challenging, such as flame characteristics, burning materials, the effect of lighting, and airflow, which continuously change the shape of fires. Hence, attaining the optimal performance for effective fire detection in terms of accuracy, loss, and primary metrics like the false-positive rate (FPR) and false-negative rate (FNR) remains a persistent and unresolved challenge in traditional ML approaches.

To address the problems of using ML-based techniques, researchers have directly migrated to exploring DL-based techniques, which are considered the most popular approaches in various computer vision applications [12,14,15]. These techniques comparatively offer higher performance and significantly reduced false alarm rates in real-time decision making compared to ML-based methods. However, DL-based approaches still suffer from various limitations including fire localization and detection, particularly in complex environments such as conditions involving fire-colored lights, similar objects to fire, or sunlight that mimics fire. Additionally, various studies have [7,16–18] contributed datasets for experimental analysis that lack sufficient fire images for training and testing and are often small and lacking in diversity. This limitation hinders the development of robust and reliable models that are trained on effective data. To address this issue, we collected images from various sources including publicly available datasets, Google, and YouTube, particularly focusing on high-resolution images of both indoor and outdoor settings. Additionally, existing deep learning models continue to struggle to achieve both efficiency and accuracy, limiting their real-world applicability in fire detection systems. Overall, the major contributions of the study are given below:

- This study offers a custom fire dataset comprising two distinct classes, fire and non-fire, containing highly diverse and visually similar images in both categories. Our custom dataset tackles the challenge of distinguishing fire from non-fire objects, improving detection accuracy in complex real-world scenarios.
- This study introduces a novel super-resolution preprocessing technique applied to our custom dataset, focusing on enhancing image quality while preserving key information. This approach improves the dataset's usability for training deep learning models, ultimately leading to more accurate fire detection.
- We introduce a proposed network, an innovative framework for fire detection that addresses the limitations containing existing techniques. The proposed network is

built upon a pretrained optimized Xception architecture and incorporates an adaptive spatial attention (ASA) module to highlight relevant features and enhance the selection of dominant features from the training data, respectively.

- We conduct extensive experiments, comparing the proposed network with various deep-learning-based pretrained techniques, with and without attention modules and other state-of-the-art methods, and using the custom and benchmark datasets. The detailed analysis demonstrates that the proposed network outperforms the benchmarks in terms of precision, recall, F1-score, and accuracy, proving its suitability for real-time decision making.

2. Related Work

Currently, early fire detection solutions have become a critical area of research due to their potential to mitigate widespread social, economic, and environmental damage. Traditional fire detection systems, such as smoke or heat detectors, often suffer from delayed responses or inaccuracies in certain conditions or a wide range of areas, leading to a growing interest in using ML and DL approaches [19–21]. The various existing studies in the fire domain offer solutions with efficient and effective performances, especially in diverse and challenging environments. However, these existing studies are associated with certain limitations, including limited data availability, slow inference time, and suboptimal performance when it comes to complex scenarios. For instance, several authors employing traditional machine learning methods have proposed various techniques focused on spatial, temporal, and spectral analysis to enhance fire detection accuracy using input data. However, these approaches often rely on the assumption that fires display unusual shapes, which may not be reliable, as objects can change shape during movement [16]. Traditional machine learning techniques, such as the quick Fourier transform and wavelet analysis, have been implemented [22]. In another study, researchers utilized color features, shape variations, mobility analysis, and bag-of-words techniques to classify fires, particularly in their early stages [12]. Earlier approaches also employed a gray-level co-occurrence matrix and a histogram of oriented gradients alongside SVM [23]. Nonetheless, TFD-based methods involve complex and time-consuming manual feature extraction, which often leads to challenges in achieving high precision. Consequently, researchers have increasingly turned to deep learning techniques due to their superior performances.

In the recent literature for better fire scene classification, Khan et al. [24] proposed a UAV-based forest firefighting system that incorporates computer vision for continuous forest surveillance and fire detection. In this study, the authors used a VGG-19-based transfer learning technique for accurate fire detection using the DeepFire dataset, and they achieved a mean accuracy of 95%, with precision and recall rates of 95.7% and 94.2%, respectively. However, the system is limited by the potential for false alarms and the need for improved spatial resolution in the DeepFire dataset. Another study presented by Ayumi et al. [25] implemented a DL-based network for fire detection, particularly for outdoor fire detection, specifically focusing on fire in the forest; they utilized two distinct networks, including Xception and the MobileNet model Furthermore, to improve model accuracy, Contrast-Limited Adaptive Histogram Equalization (CLAHE) and data augmentation techniques were applied, resulting in optimal performance on the test data. Similarly, Nadeem et al. [26] presented a new lightweight network (FlameNet) for efficient real-time monitoring of fires in smart cities; the approach was shown to be particularly effective for classifying disasters into two different categories: fire and non-fire In their study, the authors proposed a framework for detecting fires in input data, followed by triggering an alert to fire and rescue departments. Additionally, the authors introduced a custom dataset to conduct comprehensive experiments, allowing for a thorough evaluation of the proposed

network in comparison to other techniques. While their proposed FlameNet demonstrated promising performance, further development is needed to accurately localize fire sources, such as those in cars, buildings, and ships.

In a study presented by Seydi et al. [27], the authors designed an innovative DL-based framework called Fire-Net for detecting fire in small forest areas. This framework was trained using Landsat-8 imagery to identify active fires; the experimental results demonstrated an overall accuracy of 97.35%, which shows that the model is able to detect even small active fires. Dilshad et al. [5] implemented a DL network named E-FireNet, considered to be a real-time, efficient fire detection framework; they accomplished the best validation accuracy with limited parameters on a custom dataset. Chetoui et al. [28] published the latest object detection models, YOLOv8 and YOLOv7, for fire and smoke detection. The models were trained on their newly created dataset that obtained an mAP50 of 92.6%, a precision score of 83.7%, and a recall of 95.2%, respectively. However, further improvement is needed to effectively discriminate smoke from similar elements like fog, haze, and clouds. As a follow-up to the research presented by Goncalves et al. [29], the YOLOv7x and YOLOv8s models were used to detect wildfires and smoke. The models were evaluated on the D-fire and WSDY datasets, where YOLOv7x showed the best detection performance, achieving an mAP of 80.40 on the D-Fire dataset, while YOLOv8s achieved a high performance with an mAP of 98.10% on the WSDY dataset. A study by Wahyono et al. [30] introduces a novel framework that combines color, motion, and shape features with machine learning techniques. The characteristics of the fires were not only extracted by their color but also by their irregular shapes and movements. An experiment was conducted on the VisiFire dataset and yielded 89.97% and 10.03% in the true positive rate and the false negative rate, respectively. Nevertheless, the physical installation of the camera poses a significant challenge. In a more recent investigation aiming to reduce the limitations of false positive rates, Xu et al. [31] implemented a new technique based on a hybrid structure, utilizing two individual learners, including Yolov5 and EfficientNet, which were integrated to accomplish the fire detection process and to be responsible for acquiring global information, respectively. For assessment, a custom dataset was used that improved detection performance by 2.5% to 10.9% and decreased false positives by 51.3%.

A study by Shamta et al. [32] developed a deep-learning-based surveillance system for early forest fire detection and monitoring using a UAV equipped with an NVIDIA Jetson Nano and a camera. Thereafter, the CNN-RCNN model was used for fire classification the YOLOv5 and YOLOv8 models were used and for fire detection; these achieved excellent results in the testing phase. For a more accurate and effective real-time monitoring system, further refinements and field implementations are necessary. In later research, to detect smoke in forest fire images, Kim et al. [33] proposed a modified version of the YOLOv7 model incorporated with a CBAM attention mechanism for improving the YOLOv7 feature extraction ability. Additionally, the model was then assessed on their smoke dataset, which outperformed the existing state-of-the-art and multistage object detection models. However, for advanced smoke detection in wildland scenarios, further improvements in the quality of smoke images are necessary. In another methodology, Luan et al. [34] designed an improved YOLOX network for the fast detection of forest fires in UAV images. A multi-level feature extraction structure model was employed to increase feature extraction capability in complex fire environments. Subsequently, a CBAM attention mechanism is embedded in the neck network to reduce interference caused by background noise and irrelevant information. The model was then assessed on test data, outperforming deep learning algorithms such as FasterRCNN, SSD, and YOLOv5, respectively. A thorough literature review highlights several critical gaps in the current fire detection research that must be addressed. These limitations include the following: (1) A lack of comprehensive

training data that adequately capture diverse weather conditions, real-world scenarios, and high-resolution images without compromising detail. (2) Many existing deep learning models suffer from high computational complexity, a high number of parameters, and slow inference times, making them impractical for real-time decision making. (3) Persistent challenges in fire scene classification and localization result in suboptimal performances, especially in complex environments, requiring significant improvements to achieve better performance across various evaluation metrics. The subsequent section provides an in-depth analysis of the proposed network and the advanced modules that are integrated into its architecture.

3. Methodology

This section presents the methodology of our proposed model for accurate fire detection. We start with advanced preprocessing techniques, focusing on super-resolution methods, to enhance low-resolution images into high-resolution formats while maintaining crucial information. Next, we employ a DL-based optimized network, Xception [35], for feature extraction over training data. Additionally, to enhance the model's focus on task-relevant features, we then integrated an attention mechanism known as the ASA attention module, aiming to enhance overall performance significantly in complex environments. Finally, fully connected layers are implemented for the final classification into either fire or non-fire categories. Overall, detailed discussions regarding the preprocessing techniques and the various modules integrated into the proposed network are provided in subsequent sections. The main diagram is shown in Figure 1.

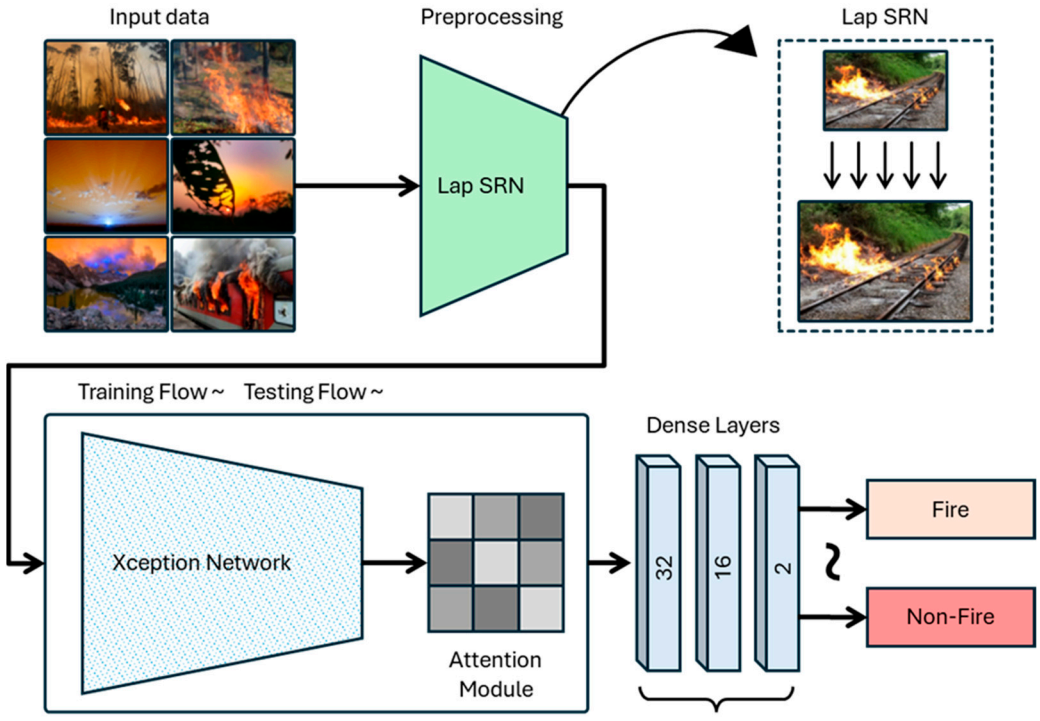

Figure 1. The proposed framework for fire detection using a custom dataset.

3.1. Data Preprocessing

The data preprocessing for our fire detection task focuses on increasing the data resolution from low quality to high quality without losing relevant information using the Laplacian Pyramid Super-Resolution Network (LapSRN) [36–38]. Our custom dataset contains images with limited resolution, resulting in the loss of fine details such as flame boundaries, smoke textures, and spark patterns. These subtle details are crucial for ac-

curately detecting fire-related features, and without resolution enhancement, the model might miss typical information. To address this, in this study, LapSRN is utilized as a preprocessing step to improve image quality. LapSRN works by progressively reconstructing high-resolution images from low-resolution inputs using a CNN with a coarse-to-fine approach. The network employs a pyramid structure that processes the image at multiple levels, each generating residuals that are used to refine the higher-resolution output.

At each level, l, the low-resolution input image is upsampled by a factor of 4, which can be mathematically represented as:

$$I_{l+1} = Upsample\ (I_l) + R_l \tag{1}$$

where I_l is the input image at level l, R_l is the residual image predicted by the network at that level, and Upsample (I_l) is the upscaled version of I_l.

The network minimizes the reconstruction error using L1 loss between the super-resolution image I_{SR} and the high-resolution ground truth I_{HR}, which is expressed as:

$$L(I_{SR},\ I_{HR}) = \frac{1}{N}\sum_{i=1}^{1}||I_{SR}(i) - I_{HR}(i)||_1 \tag{2}$$

Here, N is the total number of pixels in the image; $I_{SR}(i)$ refers to the pixel value at the $i-th$ position in the super-resolution image; $I_{HR}(i)$ refers to the pixel value at the $i-th$ position in the high-resolution ground truth image. By applying this loss, the network ensures the preservation of important structural details in the super-resolution images.

This preprocessing step using LapSRN enhances the overall quality of the fire images, allowing our detection model to extract more accurate and meaningful features from the dataset. The high-resolution images provide the model with the necessary fine details to improve its performance in identifying fire-related patterns, making it more reliable for real-world applications. The internal structure of the LapSRN network is shown in Figure 2.

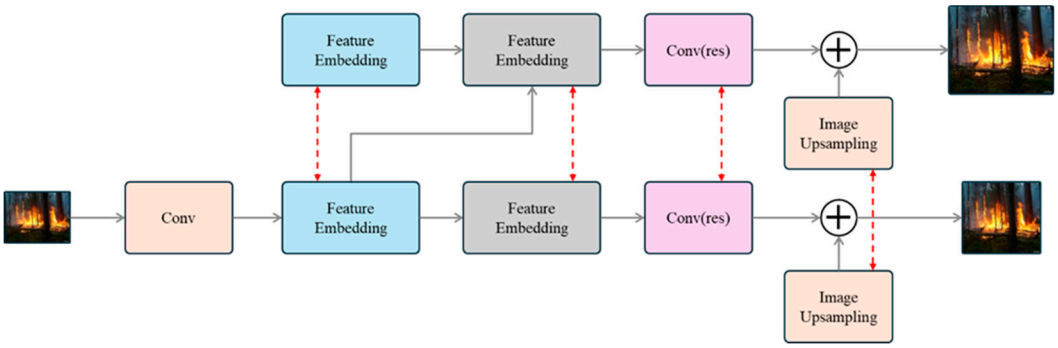

Figure 2. The main structure of LapSRN network.

3.2. Deep Feature Selection

In the computer vision domain, DL-based networks have been explored in various fields including video summarization, anomaly detection, medical image analysis, disease identification, object detection, and vehicle re-identification. DL-based algorithms are generally structured into several stages, including convolutional layers, pooling layers, and a final multi-layer perceptron. Deep convolutional neural networks (DCNNs) consist of an input layer, multiple hidden layers, and a fully connected layer with a SoftMax activation function [39]. In these networks, relevant features are extracted from the training data via convolutional layers, followed by downsampling through the mean, min, and max pooling layers to reduce dimensionality, ultimately aiding in the decision-making process.

Selecting efficient and effective algorithms in the computer vision domain—especially for deployment on edge devices—presents a significant challenge. The goal is to achieve optimal performance while maintaining low computational complexity [40]. Each CNN-based architecture possesses unique strengths and weaknesses. Notably, pretrained architectures such as VGG16 [41] and AlexNet [42] are popular due to their straightforward design and ease of implementation. AlexNet [42], in particular, set a benchmark for deep learning frameworks after its success in the ImageNet competition. Moreover, enhancing the model architecture, particularly by increasing the number of convolutional layers, is commonly believed to improve overall performance, a claim supported by the VGG16 [41] model. VGG16's internal architecture, featuring 16 convolutional layers with varying filter sizes, is recognized as a robust feature extractor. It excels in performance on large datasets, especially in scenarios with typical backgrounds. Additionally, VGG16 [41] significantly improves classification accuracy, making it a strong choice for such applications.

Existing studies indicate that deep architecture-based networks, such as as VGG19 and VGG16 [41], deliver significant performance across various tasks.

Although these networks suffer from computational complexity, they may not be applicable for real-time decision-making over edge devices. On other hand, CNN-based architectures, particularly NASNetMobile [43], EfficientNetB0 [44], Xception [35], and MobileNet, demonstrate superior performance with fewer parameters, especially in visual tasks. Among these networks, NASNetMobile, EfficientNetB0, and MobileNet are optimized for rapid and reliable response times. However, they tend to underperform in complex scenarios. In contrast, Xception [35] excels in various challenging visual tasks while maintaining a reduced parameter count, making it particularly suitable for applications that require fast processing capabilities. The Xception architecture provides significant advantages in computational efficiency, making it an appealing choice in many scientific contexts. It is crucial to consider the practical aspects of implementation, including the computational costs and the limitations of existing lightweight models for flam detection. We therefore introduce a novel framework that focuses on an optimized Xception architecture with an attention module named ASA for efficient fire classification and localization.

The Xception architecture, a deep learning-based model, is built using core components and depthwise separable convolution layers, offering a powerful and efficient structure for image and video visual tasks (see Equation (3)). In Xception, the internal architecture contains 71 distinct layers, integrating a special structure that is particularly focused on optimal feature selection while maintaining computational efficiency, as shown in Figure 3. Additionally, the initial layer mainly utilizes a standard convolution to process the training data effectively, followed by various building blocks, utilizing depthwise separable convolution layers and emphasizing two distinct operations including depthwise convolution; these implement a single filter per input channel. Pointwise convolution is also conducted, which uses 1×1 filters to combine the outputs from the depthwise layers. The procedure allows the networks to acquire typical relevant features while significantly reducing the number of parameters compared to traditional convolutional layers.

$$Y_d = D(X) = \sum_{i=1}^{C} W_d^i * X^i \tag{3}$$

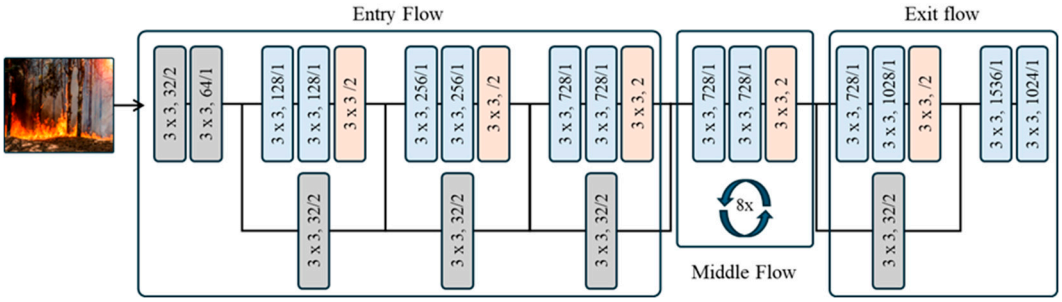

Figure 3. The internal mechanisms of the pretrained Xception deep learning network.

The standard input dimension for Xception [35] is generally configured at 299 × 299 × 3, indicating the width, height, and number of channels (RGB). Each convolutional layer is followed by batch normalization to stabilize and improve the training and convergence, respectively. Additionally, the networks utilized rectified linear unit (ReLU) activation functions throughout the network, aiming to improve the model's ability to learn complex representations. Additionally, the architecture employs global average pooling (GAP) to further emphasize dimensionality reduction and summarize overall features detailed, mainly calculating the average of each feature map to generate a compact representation of the extracted features. In the pretrained Xception [35] architecture, the dense layers are omitted, resulting in the extraction of a feature map with dimensions of 8 × 8 and 2048 channels. These extracted features can be mathematically represented as:

$$\Omega = (\Phi, \alpha)(x), \tag{4}$$

where Φ indicates the feature vectors, as (8 × 8), α represents the number of channels, and x is the training data (here, fire images). In addition, the feature vector Ω encapsulates a comprehensive range of information, including the object's configuration, boundary details, colors, shapes, and other primary characteristics.

To enhance the representation of the Ω feature map, Xception incorporates an ASA module before making a final decision, which effectively captures essential spatial patterns, allowing for improved performance in various visual tasks, significantly for fire classification and localization. Our proposed framework not only facilitates robust feature extraction but also promotes efficient computation, making it a preferred choice for a wide array of applications in the fire domain.

3.3. Adaptive Spatial Attention (ASA)

This study implements an attention module named ASA, shown in Figure 4, which is presented with the aim of further increasing the intermediate feature representations acquired from the backbone network [26,45]. The attention module specifically produces a spatial attention map by incorporating the inter-spatial relationships among features. In contrast to channel attention, which focuses features across distinct channels, the spatial attention module emphasizes specific regions that contain relevant elements within the vectors. To compute spatial attention, we begin by employing the min and max pooling operations. This is followed by the feature descriptor, mainly created by combining the features, which obtains the min and max operations. In this mechanism, the utilization of pooling operation has proven effective in emphasizing the informative area. Hence, the spatial module takes advantage of the inter-spatial connections between features. Unlike channel attention, the ASA mechanism is particularly introduced to identify the most essential region, thereby enhancing the quality of the intermediate features. The incorporation of the pooling operations along the axis serves as an effective technique for highlighting areas

with high dominance values. The combination of these two pooling operations, such as min and max, leads to the generation of enhanced features, as given below.

$$\Omega_{\Sigma} = Avg - P(f) \tag{5}$$

$$\Omega_{\lambda} = Max - P(f) \tag{6}$$

In these Equations, Ω_{Σ} denotes the output of the Avg average pooling operation applied to the feature map $P(f)$, while Ω_{λ} signifies the output of the Max maximum pooling operation applied to the feature map $P(f)$. After the feature maps are generated from the respective pooling operations, they are combined using an addition operation. This merged output is then passed through a convolutional layer to produce a two-dimensional spatial attention (SA) feature map.

Within the ASA module, we integrated two convolutional layers. The first employs a 1×1 convolution, and the second uses a 3×3 convolution. Both layers are followed by the ReLU activation function. Unlike previous research that applied dilated convolutions, our method uses standard convolutions, and this choice has been empirically validated to show its effectiveness.

$$SA(f) = \otimes(f_{1\times1}(\otimes(f_{3\times3}(\Omega_{\Sigma} + \Omega_{\lambda})))) \tag{7}$$

The f in the given Equation indicates the convolutional filter size, applied in the spatial attention mechanism. The spatial attention map, as demonstrated as SA, is generated by concatenation feature maps through following steps. First, average pooling (Ω_{Σ}) and max pooling (Ω_{λ}) are implemented in the feature maps. These two outputs are then combined together with the summed function, as mentioned ($\Omega_{\mathbf{\Sigma}} + \Omega_{\lambda}$) to form a combined feature representation.

Next, the combined feature map undergoes a sequence of convolutions. A 3×3 convolution ($f_{3\times3}$) is applied to extract spatial features, followed by a 1×1 convolution ($f_{1\times1}$) to reduce the number of channels while maintaining spatial information. The final output, spatial attention (f), is produced after applying these convolutional layers, which are represented by the convolution operation $\otimes$.

$$SA(f)_{GAP} = GAP(f) \tag{8}$$

The output from the GAP operation is concatenated with the filter function f, leading to the generation of the concatenated feature maps, denoted as f_{spa}:

$$f_{spa} = \Theta[SA(f)_{GAP}, f] \tag{9}$$

The concatenated feature maps f_{spa} undergoes batch normalization to stabilize and optimize the learning process. These normalized features are then integrated with the original feature set Ω, yielding the final feature representation f_{spaf} :

$$f_{spaf} = \otimes\ [BatchNorm,\ \Omega] \tag{10}$$

After obtaining f_{spaf}, the features are fed through three dense layers with 32, 16, and 2 neurons, respectively, for final classification. A $softmax$ function is subsequently applied to classify the input images into their designated categories, finalizing the output of the classification process.

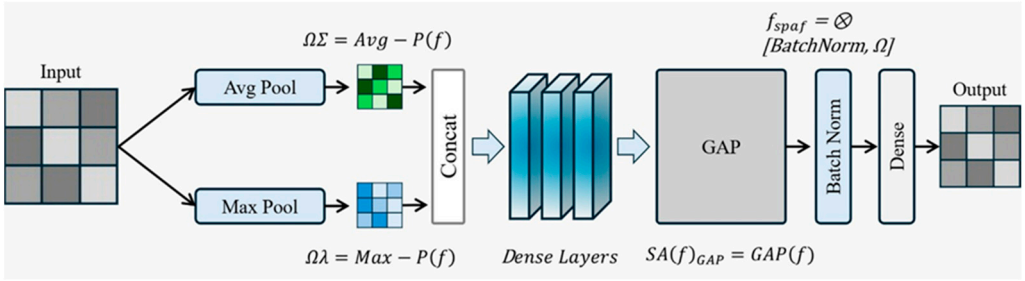

Figure 4. The adaptive spatial attention module used in this study.

4. Results and Discussion

This section offers a detailed overview of the experimental setup and parameter initialization, as well as evaluation metrics used to validate the model's performance, alongside a comprehensive explanation of the custom dataset and a standard benchmark. Additionally, we provide a thorough comparative analysis of the proposed network against various competitive techniques, measured using precision, recall, F1-score, and accuracy. In this work, all competitive networks, including the one proposed here, are trained at a low learning rate with over 30 epochs to improve the retention of learned information. We standardized the input size to 224 $\times$ 224 pixels and set a batch size of 32, using the Adam optimizer configured to 1×10^{-5}. In this study, the hyperparameters for the experimental analysis were carefully selected to optimize the proposed network and prevent underfitting or overfitting. The selection process involved a systematic approach, using techniques such as grid search to assess several combinations of hyperparameters, for example, determining different values for the number of epochs, learning rates, and batch sizes. The combination of parameters that demonstrated optimal results on the validation set was chosen for the final model. This technique ensured that the proposed network was well-tuned for the custom dataset, balancing performance and computational efficiency. Additionally, all the experimental analyses were conducted on a Windows 10 system equipped with an NVIDIA 2060 GPU that has 6 GB of memory; Kera's deep learning framework was used, with TensorFlow as a backend, and python version 3.9. Moreover, various performance indicators, including precision, F1-score, and accuracy, are used to assess the effectiveness of the proposed network and the other comparative models, as detailed in the next section.

4.1. Evaluation Metrics

In the following section, we present the comprehensive information of each evaluation metric as utilized in the study for comparative analysis. In the classification task, precision, as a well-known evaluation metric, is used to indicate the percentage of instances labeled as "Fire" that are correctly identified as actual fire within the training data, as described in Equation (11). Similarly, recall measures the ability of the network to correctly identify all relevant instances, focusing on the proportion of true positives out of the actual positives in the input data, as given in Equation (12).

$$Precision = \left(\frac{TP}{TP + FP} \right) \tag{11}$$

$$\text{Recall} = \left(\frac{TP}{TP + FN} \right) \tag{12}$$

Additionally, the F1-score illustrates a performance metric that concatenates both precision and recall into a single score, giving a balance between the two. The F1-score is

mathematically calculated as the harmonic mean of precision and recall, allowing for an assessment of a model's accuracy in classifying positive instances, as given in Equation (13).

$$F1 - score = 2 \times \left(\frac{Precision \times Recall}{Precision + Recall} \right) \tag{13}$$

Furthermore, accuracy, defined as the percentage of correct predictions relative to the total predictions across all categories, is calculated using the formula provided in Equation (14):

$$Accuracy = \left(\frac{TP + FN}{TP + TN + FP + FN} \right) \tag{14}$$

In the given equation, TP $(true\ positive)$, $TN(true\ negative)$, $FP(false\ positive)$, and FN $(false\ negative)$ indicate true positive, true negative, false positive, and false negative; a detailed explanation of these is available in [26,46]. Figure 5 presents the confusion matrix, used for classification performance.

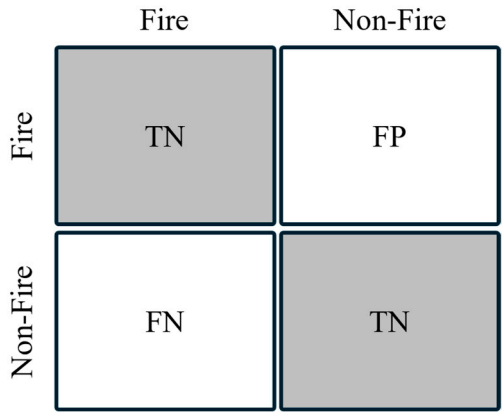

Figure 5. Confusion matrix for classification, highlighting the distribution of true and predicted labels across different classes.

4.2. Dataset Description

In the field of computer vision, particularly for fire detection, the availability of data is limited, which constrains the ability of deep learning models to achieve optimal performance. To address this, in this study, we contribute a custom dataset to validate the proposed network and compare it with various competitive techniques based on precision, recall, F1-score, and accuracy. Hence, acquiring suitable data in the fire domain for experimental analysis is often a challenging and time-consuming task; this is especially the case when the approach is focused on covering a wide range of scenarios and challenging scenes. In other words, the existing datasets contain limited numbers of samples and emphasize specific environmental conditions, such as indoor or outdoor spaces, which limits the DL-based network's ability to generalize across various situations. To tackle this limitation, we created a custom dataset that includes two distinct classes: fire and non-fire. The dataset comprises 1645 fire images and 1277 non-fire images, collected from various sources, including web searches and publicly available datasets. In this dataset, the collected samples for both fire and non-fire belong to various categories, for example, forest fire, urban fire, indoor scenes, and outdoor scenes. To provide a comprehensive overview of the dataset distributions, around 63% of the flame samples correspond to forest fires, while 37% belong to urban fires. Similarly, for non-fire images, roughly 40% of samples are captured in forest environments and 60% represent urban regions. Additionally, regarding the outdoor and indoor distributions, an estimated 30% of fire samples were collected from indoor scenes, while 70% of images indicated the outdoor environments. In addition, for non-fire images,

approximately 35% of images are taken in indoor scenarios, while 65% of samples belong to outdoor environments. This approach ensures that our dataset covers a diverse array of challenging fire and non-fire scenarios, making it better-suited for real-world applications. Additionally, we evaluated the performance of the proposed network using the standard BowFire dataset [47]. This dataset consists of two distinct classes, fire and non-fire, with 119 and 107 samples, respectively. The BowFire dataset is relatively small in size and exhibits class imbalance, containing a diverse range of challenging fire and non-fire images. For the experimental analysis, the datasets are categorized into three parts, for instance, 80% for training, 10% for validation, and 10% for testing. A few sample images from the dataset are presented in Figure 6, showing the complexity and variety of the data used in this study.

Figure 6. Various samples of our custom dataset, representing fire and non-fire images.

4.3. Comparative Analysis Before/After Preprocessing

This section presents a comprehensive comparative analysis of the proposed network among several competitive techniques, both with and without an attention module. These techniques include ResNet50 [48], VGG16 [41], ResNet101 [48], DenseNet121 [49], and Xception over different evaluation metrics such as precision, recall, F1-score, and accuracy, assessed before and after preprocessing. In this analysis, the proposed network demonstrates superior performance with both the preprocessed and original data, surpassing all state-of-the-art techniques. For example, ResNet50 [48] and VGG16 [41], when combined with an attention module, yield suboptimal results, achieving 67.30% and 65.20% accuracy on the original data, respectively. Conversely, ResNet101 [48] and DenseNet121 [49], when integrated with attention modules, gradually improve their overall performance, attaining accuracies of 69.30% and 69.60%, respectively. Similarly, the optimized Xception network with an attention module (the proposed network) offers the best results, outperforming other models by achieving a precision of 0.7447, recall of 0.7819, F1-score of 0.7628, and an accuracy of 73.70% on the original data, with a model size of 88 MB, as detailed in Table 1.

We further conducted extensive experiments on the preprocessed data, comparing different techniques against the proposed network using precision, recall, F1-score, and accuracy. In this evaluation, the proposed network exhibited even better performance, achieving precision, recall, F1-score, and accuracy of 0.8411, 0.8645, 0.8527, and 85.00%, respectively. DenseNet121 [49], with ASA being the second-best performer, attained a 77.10% accuracy with a model size of 33 MB. In addition, other networks, including ResNet101 [48], ResNet101 + ASA, VGG16 [41], VGG16 + ASA, ResNet50 [48], and ResNet50 + ASA, achieved accuracies of 74.00%, 76.50%, 69.78%, 71.00%, 73.00%, and 73.90%, respectively. This detailed analysis clearly highlights the proposed network's promising performance, proving its suitability for real-time decision making in resource-constrained environments such as edge devices. The network's ability to deliver high precision, recall, F1-score, and accuracy, even after preprocessing, demonstrates its robustness in handling complex fire

detection tasks. Its compact model size and optimized use of resources make it particularly well-suited for deployment in real-time applications, such as drones, surveillance systems, and embedded devices, where computational power and memory are often limited. Additionally, the integration of attention modules further enhances the network's focus on relevant features, ensuring efficient processing without compromising accuracy. This balance of performance and efficiency positions the proposed network as a practical solution for real-world fire detection scenarios. The visual performance results of the proposed network are illustrated in Figure 7.

Table 1. Comparative analysis of proposed network and various competitive techniques with and without attention module, representing strategies before and after preprocessing.

Model	Attention (ASA)	Before Preprocessing				After Preprocessing				Model Size
		Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score	Accuracy	
ResNet50 [48]	×	-	-	-	61.52	0.755	0.7303	0.7429	73.00	98 MB
ResNet50 [48]	✓	0.7189	0.6569	0.6865	67.30	0.7495	0.7393	0.7444	73.90	
VGG16 [41]	×	0.6269	-	-	64.08	-	-	-	69.78	528 MB
VGG16 [41]	✓	0.6316	0.7059	0.6667	65.20	0.7457	0.7054	0.7233	71.00	
ResNet101 [48]	×	-	-	-	-	0.7643	0.7473	0.7547	74.00	171 MB
ResNet101 [48]	✓	0.6729	0.7291	0.6999	69.30	0.7700	0.7624	0.7662	76.50	
DenseNet121 [49]	×	0.6833	0.7153	0.6984	69.38	0.7733	0.7375	0.7549	75.20	33 MB
DenseNet121 [49]	✓	0.7126	0.7072	0.7099	69.60	0.7600	0.7771	0.7685	77.10	
Xception [35]	×	-	-	0.7007	71.64	0.7840	0.7612	0.7724	76.90	88 MB
Proposed network	✓	**0.7447**	**0.7819**	**0.7628**	**73.70**	**0.8411**	**0.8645**	**0.8527**	**85.00**	

Figure 7. Visual representation of the proposed network's performance on the custom dataset.

4.4. Comparative Analysis Across State-of-the-Art Methods

In this section, we evaluate the performance of our proposed network on the standard BowFire dataset and compare it with existing techniques using precision, recall, F1-score, and accuracy. These techniques include FG-GCM [50], BowFire [47], EFD-IP [51], MS-CNN [52], EFDNet [16], and SE-EFFNet [53], as detailed in Table 2. In the comparative analysis, FG-GCM [50] achieved lower performances in precision, recall, and F1-score, achieving 55.0%, 54.0%, and 54.0%, respectively. In addition, the BowFire [47] indicated higher results for recall (65.0%) and F1-score (67.0%) as compared to FG-GCM. EFD-IP [51] achieved a higher result in terms of precision (75.0%), though it exhibited poor performance for recall (15.0%) and F1-score (25.0%). On the other side, EFDNet [16] offered further improvement, achieving a precision of 81.8%, a recall of 83.0%, an F1-score of 81.9%, and an accuracy of 83.3%. However, EFD-Net [16] required further improvement across precision, recall, F1-score, and accuracy. Similarly, MS-CNN [52] and SE-EFFNet [53] attained competitive accuracies of 93.0% and 94.4%, respectively, though their overall performance varied. In contrast, the proposed network significantly outperformed all the competitive networks, achieving superior results across all evaluation metrics: precision

of 95.74%, recall of 95.74%, F1-score of 95.74%, and accuracy of 95.74%. These findings clearly establish the proposed network as an optimal solution for effective fire detection in complex scenes.

Table 2. Comparative analysis of the proposed network and various competitive state-of-the-art methods over BowFire dataset.

Method	Precision	Recall	F1-Score	Accuracy
FG-GCM [50]	55.0	54.0	54.0	-
BowFire [47]	51.0	65.0	67.0	-
EFD-IP [51]	75.0	15.0	25.0	-
MS-CNN [52]	-	-	-	93.0
EFDNet [16]	81.8	83.0	81.9	83.3
SE-EFFNet [53]	-	-	-	94.4
Proposed Network	**95.74**	**94.78**	**95.26**	**95.29**

4.5. Comparison of Various Techniques Using K-Fold Cross Validation

The proposed model is assessed and compared to other networks, for example, ResNet50, VGG16, ResNet101, and DenseNet121 with an attention module, using a 5-fold cross-validation strategy using accuracy. These comparisons are conducted to analyze the capability of our proposed network for fire detection, whereas the results are structured in Table 3. In this analysis, the ResNet50 achieved average performance, achieving 72.20%, 70.00%, 72.32%, 71.56%, and 69.30% across folds, offering a distinct performance on each fold. Similarly, VGG16 offered 69.87%, 70.20%, 72.10%, 69.70%, and 71.21% accuracy for all folds. Additionally, ResNet101 exhibited an improved performance compared to ResNet50 and VGG16, achieving accuracies between 76.80% and 77.90%, demonstrating a more consistent performance across most folds. DenseNet121 further enhanced the performance, achieving accuracies between 77.70% and 80.10%, indicating its superior feature extraction. Overall, the proposed network achieved the highest performance among all compared methods, with accuracies varying between 80.45% and 85.0% across all the folds. In conclusion, this consistent improvement in accuracy highlights the network's effectiveness in leveraging attention mechanisms and optimized architecture for better feature learning and decision making. It is proven through a detailed comparative analysis that the proposed network outperformed all the competitive models, consistently achieving superior performance across the 5-fold cross-validation. This analysis indicates the robustness of our proposed network for fire detection.

Table 3. Performance evaluation of the proposed network across different models using a K-fold cross-validation over the custom dataset.

Method	Attention	Accuracy				
		1-Fold	2-Fold	3-Fold	4-Fold	5-Fold
ResNet50	✓	72.20%	70.00%	72.32%	71.56%	69.30%
VGG16		69.87%	70.20%	72.10%	69.70%	71.21%
ResNet101		77.70%	77.64%	76.80%	-	77.90%
DenseNet121		80.10%	79.34%	79.50%	78.05%	77.70%
Proposed Network		**83.90%**	**81.20%**	**83.24%**	**80.62%**	**85.00%**

4.6. Time Complexity

In this section, we conduct a comprehensive evaluation of the proposed network, comparing it with various DL-based networks integrated with attention modules. The experimental analysis significantly focuses on the FPS rate achieved on both CPU and GPU

settings. As presented in Table 4, the proposed network demonstrates a remarkable FPS rate of 8.56 on the CPU and 42.37 on the GPU, indicating superior efficiency relative to other DL-based networks.

Table 4. Comparison of the techniques in terms of frames per second (FPS) over CPU and GPU.

Method	Attention	Frame Per Second (FPS)	
		CPU	GPU
ResNet50 [48]	✓	3.01	26.87
VGG16 [41]	✓	2.98	21.02
ResNet101 [48]	✓	6.03	31.25
DenseNet121 [49]	✓	4.23	29.01
Proposed network	✓	**8.56**	**42.37**

Similarly, the ResNet50 [48] model recorded FPS values of 3.01 on the CPU and 26.87 on the GPU, while the VGG16 [41] model with attention yielded 2.98 FPS on the CPU and 21.02 FPS on the GPU. The ResNet101 [48] model, while showing an effective FPS of 6.03 on the CPU and 31.25 on the GPU, still did not match the efficiency of the proposed network. Meanwhile, DenseNet121 [49] achieved FPS values of 4.23 on the CPU and 29.01 on the GPU, demonstrating its capabilities but suggesting a need for further optimization to enhance performance and FPS rates overall.

As detailed in Sections 4.3 and 4.4, the experimental results highlight that the proposed network not only improves overall performance but also achieves a commendable FPS, as outlined in Table 4. The significant increase in FPS across both CPU and GPU settings underscore the model's robustness for real-time applications, offering its effectiveness in practical scenarios. Furthermore, the performance of the proposed network indicates its superior fire detection capabilities compared to various traditional algorithms, particularly under typical weather conditions both indoor and outdoor. Overall, our approach signifies a notable advancement in the efficiency of DL-based models, showcasing its potential for implementation in complex environments for fire detection and classification.

4.7. Limitations of the Proposed Network

The proposed network demonstrates an optimal performance in fire detection. However, it contains various limitations; for instance, the network may offer suboptimal performance in complex environmental conditions such as heavy smoke, extreme lighting, or dense fog, where significant noise can hinder its ability to accurately differentiate between fire and non-fire regions. Additionally, while the network achieved higher performance in binary classification tasks (fire and non-fire), its robustness for multi-class classification remains unexplored. Differentiating between specific fire types, such as bus fires, car fires, and bike fires, could be challenging due to variations in fire characteristics and contextual features. Furthermore, despite its lower number of parameters and faster inference speed compared to other competitive techniques, the network requires further optimization to enhance its efficiency for real-time decision-making applications.

5. Conclusions

We developed a novel framework for early fire detection, significantly focusing on two primary phases: preprocessing and model initialization. In the first phase, we applied super-resolution preprocessing to the input data, enhancing image quality without losing important details. In the second phase, we designed an attention-based DNN network, utilizing an optimized Xception network for efficient feature selection to achieve optimal performance with lower computational complexity. Additionally, we incorporated an

adaptive attention module named ASA, aiming to select further dominant features for final classification. In this study, we offer a custom dataset containing extremely diverse, high-resolution, and similar images of fire/non-fire, including two different classes: fire and non-fire. To validate the effectiveness of the proposed network, we conducted a comparative analysis against various competitive techniques with and without attention modules. The experimental results demonstrate that our network achieved superior performance, surpassing all networks in terms of precision, recall, F1-score, and accuracy. Thus, our proposed network offers a highly optimal and reliable solution for fire detection in challenging environments. Its effective analysis across challenging datasets ensures accurate fire detection, particularly in adverse circumstances, making it ideal for edge devices. In this study, the proposed network's high performance with low computational complexity enables real-time fire detection on devices with limited processing power, ensuring timely responses in typical conditions. This makes it particularly suitable for wide-scale use in areas where traditional, high-resource systems may not be feasible, significantly enhancing fire detection capabilities.

In the future, we aim to explore both indoor and outdoor fire detection, particularly in challenging environments, broadening the application scope of the proposed framework. Further, we plan to implement various model compression techniques including model quantization and pruning, to further reduce the model size while maintaining its performance. These techniques will improve the network's efficiency, making it easier to deploy edge devices with limited resources, without reducing the detection performance.

Author Contributions: M.A.: writing—original draft preparation; visualization; methodology; data curation; conceptualization. M.Y.: writing—review and editing; investigation; formal analysis. N.D.: writing—review and editing; validation; formal analysis; data curation. W.K.: supervision; funding acquisition; project administration; writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Gachon University Research Fund (GCU-202307760001) and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (2022R1F1A1074767).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: This study utilized a public dataset and a private dataset. The private dataset and analysis code are available upon formal request.

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

References

1. Filkov, A.I.; Ngo, T.; Matthews, S.; Telfer, S.; Penman, T.D. Impact of Australia's catastrophic 2019/20 bushfire season on communities and environment. Retrospective analysis and current trends. *J. Saf. Sci. Resil.* **2020**, *1*, 44–56.
2. Nosouhi, M.R.; Sood, K.; Kumar, N.; Wevill, T.; Thapa, C. Bushfire risk detection using Internet of Things: An application scenario. *IEEE Internet Things J.* **2021**, *9*, 5266–5274. [CrossRef]
3. Bushnaq, O.M.; Chaaban, A.; Al-Naffouri, T.Y. The role of UAV-IoT networks in future wildfire detection. *IEEE Internet Things J.* **2021**, *8*, 16984–16999. [CrossRef]
4. Muksimova, S.; Umirzakova, S.; Mardieva, S.; Abdullaev, M.; Cho, Y.I. Revolutionizing Wildfire Detection Through UAV-Driven Fire Monitoring with a Transformer-Based Approach. *Fire* **2024**, *7*, 443. [CrossRef]
5. Dilshad, N.; Khan, T.; Song, J. Efficient Deep Learning Framework for Fire Detection in Complex Surveillance Environment. *Comput. Syst. Sci. Eng.* **2023**, *46*, 749–764. [CrossRef]
6. Yar, H.; Khan, Z.A.; Hussain, T.; Baik, S.W. A modified vision transformer architecture with scratch learning capabilities for effective fire detection. *Expert Syst. Appl.* **2024**, *252*, 123935. [CrossRef]

7. Majid, S.; Alenezi, F.; Masood, S.; Ahmad, M.; Gündüz, E.S.; Polat, K. Attention based CNN model for fire detection and localization in real-world images. *Expert Syst. Appl.* **2022**, *189*, 116114. [CrossRef]
8. Muksimova, S.; Umirzakova, S.; Abdullaev, M.; Cho, Y.I. Optimizing Fire Scene Analysis: Hybrid Convolutional Neural Network Model Leveraging Multiscale Feature and Attention Mechanisms. *Fire* **2024**, *7*, 422. [CrossRef]
9. Bu, F.; Gharajeh, M.S. Intelligent and vision-based fire detection systems: A survey. *Image Vis. Comput.* **2019**, *91*, 103803. [CrossRef]
10. Zhang, D.; Xiao, L.Y.; Wang, Y.; Huang, G.Z. Study on vehicle fire safety: Statistic, investigation methods and experimental analysis. *Saf. Sci.* **2019**, *117*, 194–204. [CrossRef]
11. Csápaiová, N. The impact of vehicle fires on road safety. *Transp. Res. Procedia* **2021**, *55*, 1704–1711. [CrossRef]
12. Gaur, A.; Singh, A.; Kumar, A.; Kumar, A.; Kapoor, K. Video flame and smoke based fire detection algorithms: A literature review. *Fire Technol.* **2020**, *56*, 1943–1980. [CrossRef]
13. Celik, T.; Ozkaramanli, H.; Demirel, H. Fire Pixel Classification Using Fuzzy Logic and Statistical Color Model. In Proceedings of the 2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07, Honolulu, HI, USA, 15–20 April 2007.
14. Khan, T.; Aslan, H.İ. Performance evaluation of enhanced ConvNeXtTiny-based fire detection system in real-world scenarios. In Proceedings of the International Conference on Learning Representations (ICLR), Kigali, Rwanda, 2 March 2023.
15. 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]
16. Li, S.; Yan, Q.; Liu, P. An efficient fire detection method based on multiscale feature extraction, implicit deep supervision and channel attention mechanism. *IEEE Trans. Image Process.* **2020**, *29*, 8467–8475. [CrossRef] [PubMed]
17. Abdusalomov, A.; Umirzakova, S.; Safarov, F.; Mirzakhalilov, S.; Egamberdiev, N.; Cho, Y.I. A Multi-Scale Approach to Early Fire Detection in Smart Homes. *Electronics* **2024**, *13*, 4354. [CrossRef]
18. Yar, H.; Ullah, W.; Khan, Z.A.; Baik, S.W. An effective attention-based CNN model for fire detection in adverse weather conditions. *ISPRS J. Photogramm. Remote Sens.* **2023**, *206*, 335–346. [CrossRef]
19. Shakhnoza, M.; Sabina, U.; Sevara, M.; Cho, Y.I. Novel video surveillance-based fire and smoke classification using attentional feature map in capsule networks. *Sensors* **2021**, *22*, 98. [CrossRef]
20. Khan, T.; Khan, Z.A.; Choi, C. Enhancing real-time fire detection: An effective multi-attention network and a fire benchmark. *Neural Comput. Appl.* **2024**, 1–15. [CrossRef]
21. 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. [CrossRef]
22. Zhang, Z.; Zhao, J.; Zhang, D.; Qu, C.; Ke, Y.; Cai, B. Contour Based Forest Fire Detection Using Fft and Wavelet. In Proceedings of the 2008 International Conference on Computer Science and Software Engineering, Wuhan, China, 12–14 December 2008.
23. Wang, Z.; Wang, Z.; Zhang, H.; Guo, X. A Novel Fire Detection Approach Based on Cnn-Svm Using Tensorflow. In Proceedings of the Intelligent Computing Methodologies: 13th International Conference, ICIC 2017, Liverpool, UK, 7–10 August 2017; Springer: Berlin/Heidelberg, Germany, 2017. Part III 13.
24. Khan, A.; Hassan, B.; Khan, S.; Ahmed, R.; Abuassba, A. DeepFire: A novel dataset and deep transfer learning benchmark for forest fire detection. *Mob. Inf. Syst.* **2022**, *2022*, 5358359. [CrossRef]
25. Ayumi, V.; Noprisson, H.; Ani, N. Forest Fire Detection Using Transfer Learning Model with Contrast Enhancement and Data Augmentation. *J. Nas. Pendidik. Tek. Inform. JANAPATI* **2024**, *13*. [CrossRef]
26. Nadeem, M.; Dilshad, N.; Alghamdi, N.S.; Dang, L.M.; Song, H.K.; Nam, J.; Moon, H. Visual intelligence in smart cities: A lightweight deep learning model for fire detection in an IoT environment. *Smart Cities* **2023**, *6*, 2245–2259. [CrossRef]
27. Seydi, S.T.; Saeidi, V.; Kalantar, B.; Ueda, N.; Halin, A.A. Fire-Net: A Deep Learning Framework for Active Forest Fire Detection. *J. Sens.* **2022**, *2022*, 8044390. [CrossRef]
28. Chetoui, M.; Akhloufi, M.A. Fire and Smoke Detection Using Fine-Tuned YOLOv8 and YOLOv7 Deep Models. *Fire* **2024**, *7*, 135. [CrossRef]
29. Gonçalves, L.A.O.; Ghali, R.; Akhloufi, M.A. YOLO-Based Models for Smoke and Wildfire Detection in Ground and Aerial Images. *Fire* **2024**, *7*, 140. [CrossRef]
30. Wahyono; Harjoko, A.; Dharmawan, A.; Adhinata, F.D.; Kosala, G.; Jo, K.H. Real-time forest fire detection framework based on artificial intelligence using color probability model and motion feature analysis. *Fire* **2022**, *5*, 23. [CrossRef]
31. Xu, R.; Lin, H.; Lu, K.; Cao, L.; Liu, Y. A forest fire detection system based on ensemble learning. *Forests* **2021**, *12*, 217. [CrossRef]
32. Shamta, I.; Demir, B.E. Development of a deep learning-based surveillance system for forest fire detection and monitoring using UAV. *PLoS ONE* **2024**, *19*, e0299058. [CrossRef]
33. Kim, S.-Y.; Muminov, A. Forest fire smoke detection based on deep learning approaches and unmanned aerial vehicle images. *Sensors* **2023**, *23*, 5702. [CrossRef]
34. Luan, T.; Zhou, S.; Zhang, G.; Song, Z.; Wu, J.; Pan, W. Enhanced Lightweight YOLOX for Small Object Wildfire Detection in UAV Imagery. *Sensors* **2024**, *24*, 2710. [CrossRef]

35. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honohulu, HI, USA, 21–26 July 2017.
36. Lai, W.S.; Huang, J.B.; Ahuja, N.; Yang, M.H. Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honohulu, HI, USA, 21–26 July 2017.
37. Mardieva, S.; Ahmad, S.; Umirzakova, S.; Rasool, M.A.; Whangbo, T.K. Lightweight image super-resolution for IoT devices using deep residual feature distillation network. *Knowl.-Based Syst.* **2024**, *285*, 111343. [CrossRef]
38. Yasir, M.; Ullah, I.; Choi, C. Depthwise channel attention network (DWCAN): An efficient and lightweight model for single image super-resolution and metaverse gaming. *Expert Syst.* **2024**, *41*, e13516. [CrossRef]
39. Liu, W.; Wen, Y.; Yu, Z.; Yang, M. Large-margin softmax loss for convolutional neural networks. *arXiv* **2016**, arXiv:1612.02295.
40. Cheknane, M.; Bendouma, T.; Boudouh, S.S. Advancing fire detection: Two-stage deep learning with hybrid feature extraction using faster R-CNN approach. *Signal Image Video Process.* **2024**, *18*, 5503–5510. [CrossRef]
41. Theckedath, D.; Sedamkar, R. Detecting affect states using VGG16, ResNet50 and SE-ResNet50 networks. *SN Comput. Sci.* **2020**, *1*, 79. [CrossRef]
42. Alom, M.Z.; Taha, T.M.; Yakopcic, C.; Westberg, S.; Sidike, P.; Nasrin, M.S.; van Esesn, B.C.; Awwal, A.A.S.; Asari, V.K. The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv* **2018**, arXiv:1803.01164.
43. Saxen, F.; Werner, P.; Handrich, S.; Othman, E.; Dinges, L.; Al-Hamadi, A. Face Attribute Detection with Mobilenetv2 and Nasnet-Mobile. In Proceedings of the 2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA), Dubrovnik, Croatia, 23–25 September 2019.
44. Patel, C.H.; Undaviya, D.; Dave, H.; Degadwala, S.; Vyas, D. EfficientNetB0 for Brain Stroke Classification on Computed Tomography Scan. In Proceedings of the 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 4–6 May 2023.
45. Yar, H.; Hussain, T.; Agarwal, M.; Khan, Z.A.; Gupta, S.K.; Baik, S.W. Optimized dual fire attention network and medium-scale fire classification benchmark. *IEEE Trans. Image Process.* **2022**, *31*, 6331–6343. [CrossRef]
46. Khan, T.; Choi, G.; Lee, S. EFFNet-CA: An efficient driver distraction detection based on multiscale features extractions and channel attention mechanism. *Sensors* **2023**, *23*, 3835. [CrossRef]
47. Chino, D.Y.; Avalhais, L.P.; Rodrigues, J.F.; Traina, A.J. Bowfire: Detection of Fire in Still Images by Integrating Pixel Color and Texture Analysis. In Proceedings of the 2015 28th SIBGRAPI Conference on Graphics, Patterns and Images, NW Washington, DC, USA, 26–29 August 2015.
48. Targ, S.; Almeida, D.; Lyman, K. Resnet in resnet: Generalizing residual architectures. *arXiv* **2016**, arXiv:1603.08029.
49. Huang, G.; Liu, S.; van der Maaten, L.; Weinberger, K.Q. Condensenet: An efficient densenet using learned group convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
50. Celik, T.; Demirel, H. Fire detection in video sequences using a generic color model. *Fire Saf. J.* **2009**, *44*, 147–158. [CrossRef]
51. Chen, T.-H.; Wu, P.-H.; Chiou, Y.-C. An Early Fire-Detection Method Based on Image Processing. In Proceedings of the 2004 International Conference on Image Processing, 2004, ICIP'04, Singapore, 24–27 October 2004.
52. Jeon, M.; Choi, H.S.; Lee, J.; Kang, M. Multi-scale prediction for fire detection using convolutional neural network. *Fire Technol.* **2021**, *57*, 2533–2551. [CrossRef]
53. Khan, Z.A.; Hussain, T.; Ullah, F.U.M.; Gupta, S.K.; Lee, M.Y.; Baik, S.W. Randomly initialized CNN with densely connected stacked autoencoder for efficient fire detection. *Eng. Appl. Artif. Intell.* **2022**, *116*, 105403. [CrossRef]