forests

MDPI

Article

An Attention-Guided Deep-Learning-Based Network with Bayesian Optimization for Forest Fire Classification and Localization

Al Mohimanul Islam [1], Fatiha Binta Masud [1], Md. Rayhan Ahmed [1], Anam Ibn Jafar [1], Jeath Rahmat Ullah [1], Salekul Islam [1,2,*], Swakkhar Shatabda [1,2] and A. K. M. Muzahidul Islam [1]

[1] Department of Computer Science and Engineering, United International University, United City, Madani Avenue, Dhaka 1212, Bangladesh; aislam192054@bscse.uiu.ac.bd (A.M.I.); rayhan@cse.uiu.ac.bd (M.R.A.); jullah191036@bscse.uiu.ac.bd (J.R.U.); muzahid@cse.uiu.ac.bd (A.K.M.M.I.)

[2] Centre for Artificial Intelligence and Robotics (CAIR), United International University, United City, Madani Avenue, Dhaka 1212, Bangladesh

* Correspondence: salekul@cse.uiu.ac.bd

Abstract: Wildland fires, a natural calamity, pose a significant threat to both human lives and the environment while causing extensive economic damage. As the use of Unmanned Aerial Vehicles (UAVs) with computer vision in disaster management continues to grow, there is a rising need for effective wildfire classification and localization. We propose a multi-stream hybrid deep learning model with a dual-stream attention mechanism for classifying wildfires from aerial and territorial images. Our proposed method incorporates a pre-trained EfficientNetB7 and customized Attention Connected Network (ACNet). This approach demonstrates exceptional classification performance on two widely recognized benchmark datasets. Bayesian optimization is employed for the purpose of refining and optimizing the hyperparameters of the model. The proposed model attains 97.45%, 98.20%, 97.10%, and 97.12% as accuracy, precision, recall, and F1-score, respectively, on the FLAME dataset. Moreover, while evaluated on the DeepFire dataset, the model achieves accuracy, precision, recall, and F1-scores of 95.97%, 95.19%, 96.01%, and 95.54%, respectively. The proposed method achieved a TNR of 95.5% and a TPR of 99.3% on the FLAME dataset, as well as a TNR of 94.47% and a TPR of 96.82% on the DeepFire dataset. This performance surpasses numerous state-of-the-art methods. To demonstrate the interpretability of our model, we incorporated the GRAD-CAM technique, which enables us to precisely identify the fire location within the feature map. This finding illustrates the efficacy of the model in accurately categorizing wildfires, even in areas with less fire activity.

Keywords: forest fire classification; EfficientNetB7; attention mechanisms; localization; Bayesian optimization; computer vision; efficient channel attention; squeeze and excitation networks

check for updates

Citation: Islam, A.M.; Masud, F.B.; Ahmed, M.R.; Jafar, A.I.; Ullah, J.R.; Islam, S.; Shatabda, S.; Islam, A.K.M.M. An Attention-Guided Deep-Learning-Based Network with Bayesian Optimization for Forest Fire Classification and Localization. *Forests* **2023**, *14*, 2080. https://doi.org/10.3390/f14102080

Academic Editors: Luis A. Ruiz, Ting Yun, Yunfei Liu and Haifeng Lin

Received: 19 August 2023
Revised: 28 September 2023
Accepted: 2 October 2023
Published: 18 October 2023

1. Introduction

In recent years, wildfires have been a growing concern around the world. These wildland fires have devastated vast areas of forest and other plants, forcing tens of thousands of people to evacuate their homes and causing irreparable damage to the ecology. The increasing frequency of forest fires can be attributed to a variety of factors, such as climate change, drought, and human activities. Consequently, governments are confronted with substantial management expenses on an annual basis in order to address this pressing issue. Forest fires have become more severe and frequent in many parts of the world, including Australia, California, the Amazon rainforest, and the Mediterranean region [1]. Despite efforts to regulate and minimize them, forest fires continue to be a huge problem for governments, environmental groups, and people all over the world. Thus, the implementation of efficient forest fire monitoring holds significant importance in protecting forest resources and ensuring the well-being of human life and property [2].

Recent forest fire data confirms a long-standing concern: the increasing prevalence of forest fires, now consuming almost double the amount of tree cover as two decades ago. In 2021, which marked one of the worst years for forest fires since the beginning of the century, 9.3 million hectares of tree cover were lost worldwide—equivalent to more than a third of all tree cover loss for that year [3]. Over the past three years, statistics reveal a concerning trend in the scale and intensity of wildfires. In 2021, despite the relatively low number of fires, a significant area of 7.1 M acres was consumed, resulting in a relatively high average of 121.56 acres burned per fire in the US. Similarly, in 2022, while the number of fires remained relatively low, the burned area increased to 7.5 M acres, with an average of 113.72 acres burned per fire. The data for January to April 2023 indicate a continuation of this worrisome trend in the US, with a relatively low number of fires but a substantial average of 30.24 acres burned per fire [4]. Canada has also experienced a total of 5738 fires this year, resulting in the scorching of 13.7 million hectares (equivalent to 33.9 million acres) [5]. Moreover, recurring environmental problems in Southeast Asia include wildfires, notably linked to land and forest fires, mainly affecting nations such as Indonesia and Malaysia. The devastating 1997–1998 forest fires devoured roughly 8 million hectares of land, leading to an estimated economic loss of approximately USD 4.47 billion, with Indonesia bearing the largest share of this burden [6]. In 2019, intense forest fires in the Indonesian regions of Sumatra and Kalimantan burned over 930,000 hectares, leading to evacuations and the deployment of over 9000 personnel to combat the flames [7]. These figures underscore the urgent need for proactive measures to mitigate and prevent the devastating impact of forest fires on our environment and communities.

Various fire detection sensors, including those measuring smoke, temperature, gas, flame, etc., face limitations such as restricted coverage, delayed response, and challenges with public accessibility. The advancements in image processing and computer vision technology have made substantial contributions to the timely identification, surveillance, and control of forest wildfires. Consequently, the conventional methods for traditional fire detection, like flame-smoke sensors, are being substituted by vision-based models. These models offer numerous advantages over traditional sensors, including greater accuracy, reduced susceptibility to errors, environmental robustness, lower cost, and broader coverage [8].

Thoroughly observing fires can be achieved through the integration of data obtained from many sources, including infrared cameras, thermal sensors, and visible-light cameras. The utilization of image processing, computer vision, and deep learning techniques enables the fusion of these data streams, thereby increasing the precision of fire detection and analysis. Researchers have attempted to offer numerous unique strategies based on computer vision and image processing over the years to set up the most accurate, efficient, and optimized fire detection system conceivable. The color analysis method is commonly used to identify fire based on its color. This approach involves transforming the image into a different color space, such as YCbCr [9,10]. In this color space, the Y component represents the luma (brightness) or luminance, while the Cb and Cr components represent the blue and red components, respectively. While feature-based strategies have performed well in fire detection tasks, machine learning (ML) techniques [11,12] have surpassed them. Support Vector Machine (SVM), Markov models [13], Instance-Based Learning classifiers [14], and Bayesian classifiers [15] are popular fire classification algorithms that are specifically designed to predict the likelihood of a wildfire occurrence within an input image.

The primary challenge of the mentioned strategies lies in identifying relevant attributes that best describe the topic at present. As an alternative, a self-learning network can be employed to acquire relevant features autonomously. Deep learning (DL) techniques can deliver excellent accuracy for fire classification and detection if a sufficiently extensive dataset is utilized during the training process. The capacity of DL-based fire classification and detection algorithms to automatically learn high-level features provides a key advantage over conventional techniques. In the existing literature, it has been observed that to detect forest fires, multiple pre-trained DL algorithms have been incorporated such as ResNet50, AlexNet, GoogleNet, VGG16, and MobileNetV2 [16–18]. In recent studies, we have observed the extensive exploration of various attention mechanisms in the context of image classification and segmentation tasks. Forest fire classification utilizing attention mechanisms leverages advanced neural network techniques to efficiently identify and respond to critical patterns and features in imagery, enhancing the accuracy of fire detection and prevention [19,20]. Accordingly, we introduce an attention-guided multi-stream hybrid model for forest fire classification. The proposed approach is straightforward, employing two streams for effective feature extraction. One stream employs the pre-trained EfficientNetB7 [21] method, while the other utilizes a custom-built Attention Connected Network (ACNet). EfficientNet is chosen for its reliable scalability, achieved through uniform scaling of network dimensions. Specifically, EfficientNetB7 is utilized for its exceptional feature extraction capabilities from fire images. ACNet, on the other hand, enhances the model's ability to capture both low-level and high-level features, offering multiple perspectives on feature importance and interdependencies. We also employ the Bayesian optimization (BO) [22] method to optimize the model's hyperparameters. The objective is to optimize the key parameters of classifiers through the utilization of BO. Therefore, it is expected that the accuracy of the model is going to improve. Furthermore, we implement the GRAD-CAM [23] technique to enhance the model's interpretability. Our attention-guided dual-stream hybrid model not only enhances forest fire classification accuracy, but also holds the potential for real-world applications in forest and wildfire management. By enabling more precise and interpretable forest fire classification, our approach can play a pivotal role in early detection of fire, rapid response, and optimized resource allocation, ultimately contributing to the mitigation and control of forest fires. The major contributions of this study can be summarized as follows:

- We introduce a novel dual stream attention guided network for the classification of forest fires.
- In the first stream, we use EfficientNetB7 as a feature extractor to efficiently extract high-level features from images.
- In the second stream, we incorporate the newly proposed attention connected module, comprising a fusion of both Efficient Channel Attention and Squeeze-and-Excitation Network modules within the network architecture. This integration not only brought selective attention, but also featured enhancement, effectively optimizing the model's forest fire classification capabilities. Bayesian optimization was employed to fine-tune hyper-parameters, enhancing the model's performance.
- Our proposed architecture's effectiveness is being thoroughly demonstrated on two widely recognized benchmark datasets. Comprehensive analyses demonstrate its superiority over several state-of-the-art methods.
- To enhance model interpretability, we integrate the GRAD-CAM technique to understand which parts of the images were most important in guiding the model's decision-making process.

The rest of the paper is organized as follows. The related works Section 2 contains a full review of previous approaches for detecting forest fires. The Materials and Methods Section 3 provides a full explanation of the proposed model. The result analysis Section 4 gives an overview of the results obtained. Finally, in Section 6, the paper is brought to a close.

2. Related Works

Several studies have been undertaken in recent years on the topic of forest fire classification and detection systems. Researchers have explored a wide range of strategies to devise a precise and efficient approach for wild forest fire classification. The reviewed models are presented in a categorized format below.

2.1. Pre-Trained and Customized CNN

In the domain of forest fire classification, Convolutional Neural Network (CNN)-based pre-trained models have gained considerable attention, employing their capacity to comprehend complex features. A. Khan et al. [24] utilized the pre-trained VGG19 for feature extraction. Incorporating fully connected layers to enhance performance, the model achieved an impressive accuracy of 95% when evaluated on the DeepFire dataset. In their recent work, Namburu et al. [25] introduced X-MobileNet, a novel deep learning method that utilizes the pre-trained mobilenetV2 as a feature extractor, employing pre-trained weights for all layers to reduce computational expenses. Furthermore, they modified the classifier to enhance the performance of the model by using Global Average pooling (GAP). A fusion of pre-trained ImageNet weights with domain-specific modifications, their approach underscored the importance of feature extraction and classification tasks. Similarly, in [26], S. Khan et al. used pre-trained mobileNetv2 as the feature extractor along with additional dense layers. Serving as the backbone of the proposed architecture, MobileNetv2 proficiently captures relevant features from preprocessed images. This technique effectively aggregates spatial information and generates a concise fixed-length vector that facilitates precise fire classification. However, the authors did not mention details regarding the number of dense layers employed in the classifier, as well as the specific neuron unit counts within each dense layer. The lack of information could make it difficult for readers to understand the model. Treneska et al. [27] explored transfer learning on the FLAME dataset with five pre-trained models—VGG16, VGG19, RestNet50, InceptionV3, and Xception. While retaining pre-trained weights for the feature extractor, they refined the classifier with GAP and supplementary dense layers. The ResNet50 model achieved the highest accuracy score, reaching 88% through fine-tuning. An innovative approach to deep ensemble learning was introduced by Ghali et al. [28] combining EfficientNet-B5 and DenseNet-201 models to classify wildfires using aerial images. Combining EfficientNet and DenseNet can potentially utilize EfficientNet's scaling advantages with DenseNet's dense connectivity for improved feature extraction. The proposed wildfire classification model achieved an impressive accuracy of 85.12%, surpassing numerous state-of-the-art models and demonstrating its capability to accurately identify wildfires, including those in smaller fire areas.

Furthermore, apart from employing pre-trained CNN models, we noticed that multiple custom CNN models were used for the classification of forest fire. Shamsoshoara et al. [29] introduced the FLAME dataset that can be utilized in wildfire classification and segmentation. In order to justify the dataset's applicability in wildfire classification tasks, they utilized the pre-trained Xception network with additional dense layers. Moreover, they also tuned the model's hyperparameter to enhance the performance of the proposed model. The proposed model achieved an accuracy of 76% on the FLAME dataset. Vani et al. [30] employed the InceptionV3 based on transfer learning for classifying the fire images. InceptionNetv3 offers improved performance due to its novel inception modules that effectively capture and process features at multiple scales, which enhance its ability to recognize complex patterns. To mitigate overfitting, they utilized a single fully connected layer in the classifier. In their study [31], L. Kurasinski et al. explored how dataset variation influences model performance, training on two distinct datasets (FLAME and NASA) and conducting cross-validation on FLAME, NASA, and a GitHub dataset. The primary objective is to gain insights into how the choice of dataset impacts the performance of the Xception model, as referenced in [29]. Akagic et al. [32] make a substantial contribution to the field of wildfire image classification by introducing LightWeight wildFIRE (LW-FIRE). To validate its name, they employed a shallow CNN architecture to reduce the model's complexity and memory requirements during training. A shallow CNN consists of a concise number of layers, offering faster training and reduced computational complexity. As a result, the proposed model was more computationally efficient and faster to train, making it suitable for scenarios where computational resources are limited. According to the authors, LW-FIRE150, which was evaluated on Corsican Fire DataBase, attaining an accuracy of 97.25% is the most optimized version of LW-FIRE.

ResNet50 has served as the foundation for numerous models, with each variant and adaptation incorporating unique modifications and improvements. It offers improved training efficiency and accuracy by resolving vanishing gradient problems through residual connections. ForestResNet [33], a classification model based on the ResNet50 architecture with cross-entropy loss, was introduced for identifying fire. Through transfer learning, they improved the model's performance, attaining an accuracy of 92% on their proposed dataset.

In recent studies, multiple image processing techniques have been used alongside pre-trained and customized CNN models to amplify classification tasks by extracting relevant features, reducing noise, and improving data representation. Wang et al. [34] used a unique technique that involves segmenting the potential flame region using color features and applying AND operation between the segmented image and the original image. For classification, a CNN model inspired by AlexNet with adaptive pooling was developed to preserve local image information. Dutta et al. [35] in their proposed method combined separable CNN with various image processing techniques such as multi-channel binary thresholding, segmentation, and HSV color space filters were utilized to handle smoke and fog in fire images and accurately classify fires. Moreover, L2 regularization was used to address the over-fitting problem.

2.2. Attention-Based Model

In the field of forest fire classification, there has been limited exploration of the Attention mechanism. Guan et al. [19] proposed the Dual Semantic Attention (DSA) module, which utilizes the attention mechanism for convolutional kernels and enhances the model's ability to capture the relevant semantic information during the convolutional operations. This module dynamically selects and combines feature maps from various convolution kernel scales. The authors improved the resnet50 architecture by integrating the DSA module into the residual block, labeling it as the DSA-Residual module. Li et al. [20] proposed the Attention-Based Prototypical Network for forest fire smoke detection. This approach combines few-shot learning and attention mechanisms to effectively extract features and minimize false alarms in suspected smoke regions. To address limited smoke images and mitigate over-fitting, a meta-learning module is introduced, comparing class prototypes of support images with features from query images for accurate detection.

Table 1 presents a comprehensive overview of the advantages and shortcomings of existing forest fire classification models, highlighting the diverse strengths and limitations.

Table 1. Strength and shortcomings of different methodology of wildfire classification.

Authors	Methods	Strength	Shortcomings
Z. Guan et al. (2022) [19]	ResNet50 with DSA	Integration of an attention mechanism enables the extraction of more useful information.	ResNet50 requires huge computational resources that are not suitable for mobile devices and embedded systems
A. Namburu et al. (2023) [25]	X-MobileNet (pre-trained MobileNet with customized output layer, utilizing global average pooling)	MobileNet architecture uses depth-wise separable convolution, which is computationally efficient.	GAP may sacrifice fine-grained spatial details crucial for accurate fire classification.
S. Khan et al. (2022) [26]	Pretrained MobileNetV2	Useful in resource-constrained environment.	Limited evaluation on the ForestFire dataset, lacking performance assessment on other datasets
S. Treneska et al. (2021) [27]	Pretrained ResNet50 with GAP in classifier	Efficient memory usage and rapid prediction time, highly suitable for real-time applications.	A fixed 5-epoch fine-tuning duration may not be optimal for all models and datasets.

Table 1. *Cont.*

Authors	Methods	Strength	Shortcomings
R. Ghali et al. (2022) [28]	Ensemble model (EfficientNet-B5, DensNet201)	Ability to accurately detect and segment wildfires, even in small fire areas.	Computational resources required to train the model are high as two distinct models used for feature extraction.
A. Shamsoshoara et al. (2021) [29]	Xception	Depth-wise separable convolutions, which are computationally efficient.	76% accuracy is insufficient for real-life fire classification tasks.
K. Vani et al. (2019) [30]	InceptionV3 with addition dense layers in classifier	Efficient computational performance.	Small utilized dataset and evaluation on a limited number of datasets.
Y. Wang et al. (2019) [34]	AlexNet with adaptive pooling	Adaptive pooling helps preserve local information in the images.	Reliance on color features might be sensitive to variations in lighting conditions.
S. Dutta et al. (2021) [35]	FT-ResNet50	Focal Loss technique significantly enhanced models learning ability.	Performance does not meet current standards.

3. Materials and Methods

3.1. Datasets

In this paper, we used two distinct public datasets: the FLAME dataset (Fire Luminosity Airborne-based Machine Learning Evaluation) [29] and the DeepFire dataset [24]. The FLAME dataset consists of 39,375 images, with 25,018 representing fire and 14,357 representing non-fire images, all utilized for training. We split these images, dedicating 80% for training (31,500 images) and 20% for validation (7875 images). Furthermore, the dataset contains 8617 images for testing. The model was evaluated on randomly chosen 3000 images from the test set. Additionally, the DeepFire dataset was used, which consists of 1520 diverse wildfire images for training, which are subdivided into “fire” and “no-fire” folders, each containing 760 images. We first split the training images into a training set and a test set with a 80–20 split. The test set had 304 images and the training set had 1216 images. Then, we further split the training set into a validation set and a training set with an 80–20 split. The training set had 972 images and the validation set had 244 images. The validation set helps to identify overfitting by providing a way to measure the model’s performance on data that it has not seen before.

3.2. Parameters of the Experiment

The proposed model is trained using two datasets: FLAME and DeepFire. The loss function employed for both datasets is Binary Cross-entropy. The optimizer used for the FLAME dataset is AdamW, while for the DeepFire dataset, Adam optimizer is utilized. The learning rate (LR) for both datasets is set to 1×10^{-5}. The training process is carried out for 50 epochs, with a batch size of 16. Early stopping is utilized to prevent overfitting by stopping the training process once the model’s performance on a validation set starts to decrease. The input size for the model is 224 × 224 pixels, ensuring that the input images are standardized to this resolution during training.

3.3. Model Architecture

Our proposed method is a two-stream hybrid model designed for forest fire classification, as depicted in Figure 1. It leverages two streams to effectively extract features from the input images of wildfires or forest fires.

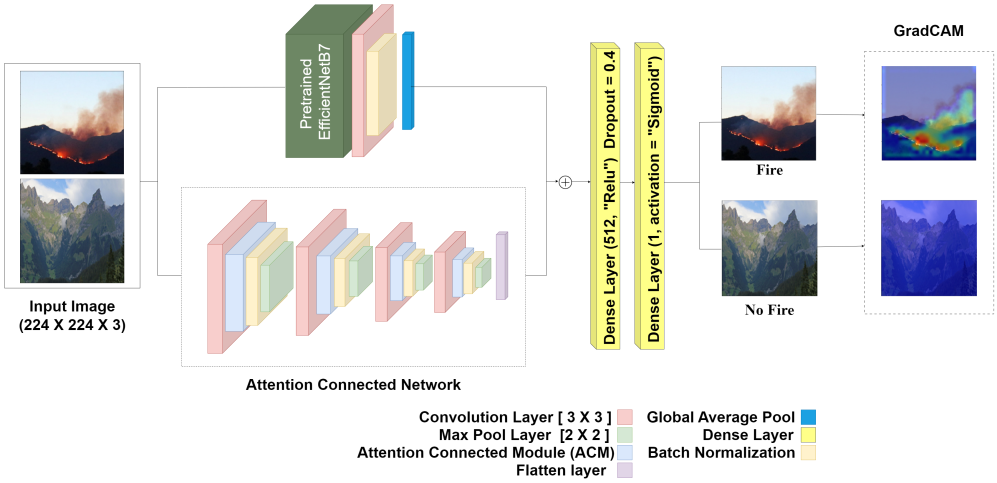

Pretrained EfficientNetB7
Input Image (224 X 224 X 3)
Attention Connected Network
Dense Layer (512, "Relu") Dropout = 0.4
Dense Layer (1, activation = "Sigmoid")
Fire
No Fire
GradCAM
Convolution Layer [3 X 3]
Max Pool Layer [2 X 2]
Attention Connected Module (ACM)
Flatten layer
Global Average Pool
Dense Layer
Batch Normalization

Figure 1. Architecture of the proposed two-stream model.

3.3.1. Pretrained EfficientNetB7

The first stream, pre-trained EfficientNetB7 [21] utilizes the transfer of knowledge from a CNN-based model pre-trained on a large ImageNet dataset. This architecture excels in tasks like image classification and object detection due to its efficiency in feature extraction and impressive performance. The practice of utilizing a pre-trained model to address a new problem can be referred to as transfer learning. Transfer learning offers several advantages, with its primary benefits encompassing the reduction of training time, enhancement of neural network performance, and the alleviation of data requirements [36]. CNNs are commonly up-scaled to enhance their accuracy when performing classification tasks on various benchmark datasets. However, the process of scaling convolutional models is often carried out randomly. Some models are scaled in terms of depth, while others are scaled in terms of width. The process of random scaling requires manual adjustment and demands a significant amount of time. In contrast, EfficientNet employs a technique known as the "compound coefficient" to scale models in a straightforward yet efficient manner. The main advantage of the EfficientNet method is its reliable scalability. Using the compound coefficient technique, EfficientNet uniformly scales all the dimensions of the network (width, depth, and resolution) using a constant ratio. The proposed model incorporates EfficientNetB7 as a feature extractor due to its remarkable ability to extract relevant and discriminative features from fire images.

The EfficientNetB7 model has 33 layers and employs the compound coefficient method to adjust model depth, resolution, and width. The implementation of this particular method leads to enhanced performance outcomes, but the associated computational costs remain relatively low. The inverted bottleneck block (MBConv), which is depicted in Figure 2, previously introduced in MobileNetV2, serves as the fundamental component in EfficientNet. The utilization of Depth-wise Separable Convolution is observed in these blocks. The initial step involves expanding the channels through a point-wise convolution (conv 1×1). Subsequently, a 3×3 depth-wise convolution is employed to significantly lower the parameter count. Finally, a 1×1 convolution is utilized to further decrease the number of channels, enabling the incorporation of the block's initial and final stages. EfficientNet applies the Squeeze and Excitation (SE) block alongside the MBConv block, resulting in the network dynamically assigning a high weight to the most important channels, thus generating effective features for the wildfire or forest-fire classification task on hand.

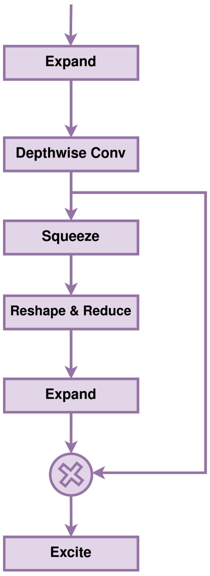


Figure 2. Inverted bottleneck block.

The incorporated EfficientNetB7 architecture was initially fine-tuned on the target datasets: FLAME and DeepFire. The base models were kept frozen (i.e., the parameters of the base model were unchanged). We further fine-tuned this model on forest fire datasets by unfreezing the final convolutional layer. In this specific case, the optimization process focused solely on the final convolutional layer and the classification layer, while the backbone network served the purpose of a pre-trained feature extractor. The collected features are then fed into a 3 × 3 convolutional block, followed by batch normalization. The use of this technique facilitates the enhancement of the training process for the specific stream and the overall network. This reduction in internal covariate shifts contributes to the stabilization of the training procedure. Consequently, the application of this technique enables faster and more efficient convergence of the network. After the convolutional layers, GAP is performed to derive a fixed-length feature vector, where each feature map contributes a single value computed as the mean of its values. This pooling operation reduces spatial dimensions while preserving channel information.

3.3.2. Attention Connected Network

As for the second stream of the proposed model, we utilize a customized Attention Connected Network (ACNet). The input fire image undergoes resizing and re-scaling in the pre-processing step before being passed through convolutional layers with varying numbers of filters and a 3 × 3 kernel size. An Attention Connected Module (ACM), which is depicted in Figure 3, batch normalization, and 2 × 2 max pooling operations are applied after each convolutional layer. This process is repeated three more times, resulting in a decrease in image dimensions due to max pooling. The filter sizes used are 32, 64, 128, and 256 maintaining a consistent kernel size. The output of the fourth max pooling layer is flattened and concatenated with the output of the GAP layer from the EfficientNetB7 backbone-based first stream of the model.

The proposed method takes RGB aerial images as input and utilizes both streams (i.e., the EfficientNetB7-backbone and ACNet) to extract re-weighted feature maps. After concatenating the feature maps, they are passed through a fully connected dense layer with 512 neurons, using ReLU activation. A dropout layer with a rate of 0.4 is applied to prevent the network from overfitting and to maintain generalizability. Finally, a sigmoid function is used for the final classification, determining whether the input image belongs to the Fire or No-Fire class.

To optimize the model's performance, hyperparameters such as the filter size in the convolutional layers, the number of neurons in the dense layer, kernel size in the Efficient

Channel Attention (ECA) block, the learning rate, and the dropout rate are optimized using the BO technique. By leveraging probabilistic models and acquisition functions, this method efficiently explores and exploits the hyperparameter space to find optimal configurations. In contrast to the existing literature, we employ optimization techniques to enhance the parameter-tuning process of the models.

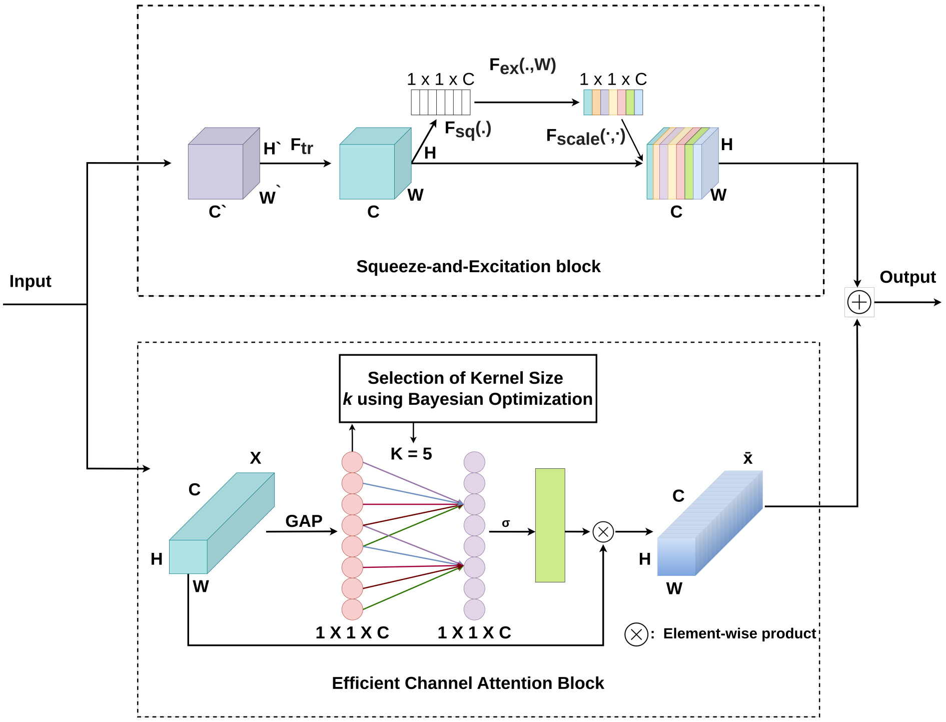


Figure 3. Attention connected module.

3.3.3. Attention Connected Module

The Attention Connected Module (ACM) employs an attention-based dual-stream architecture, where the Efficient Channel Attention (ECA) block is applied in one stream and the Squeeze-and-Excitation (SE) block is applied in the other stream. BO was utilized to determine the kernel size of 5 for the ECA mechanism. The reduction ratio was set to 16 for the SE block. The outputs of both streams are then concatenated together. Here, the SE block enhances representational power by re-calibrating feature maps to prioritize informative features, while the ECA block selectively weights channels to focus on relevant information. These components collectively improve the ACM's ability to capture essential information, leading to enhanced performance in various tasks. In addition, this combined attention can enhance the model's generalization ability in capturing both low-level and high-level features from the forest fire images. It can provide multiple perspectives on feature importance and interdependencies. It helps the model to adapt diverse input patterns and improves the performance of unseen data.

Squeeze and Excitation Block

The Squeeze and Excitation (SE) [37] module is a key component in DL models. It enhances the representational power of the model by adaptively re-calibrating the channel-wise feature responses. By selectively highlighting informative features while reducing the influence of less relevant ones, the SE module helps improve the discriminative capabilities of the model. Its ability to capture channel-wise dependencies and optimize feature representations makes it serve as an effective tool for improving the performance of DNN. It improves the model's resilience to noise and disturbances by enabling it to prioritize essential features while disregarding less significant ones.

The SE block initially utilizes GAP operation for each channel in an independent manner. Subsequently, two FC layers are employed, incorporating non-linearity. Finally, a Sigmoid function is applied to construct channel weights. The purpose of the two FC layers is to capture non-linear cross-channel interaction. This involves reducing dimensionality in order to control the complexity of the model [38].

Here, a series of convolutional transformations convert the provided image, as input with the dimension of (W', H', C'), then mapped to the feature map U. The SE block operates by squeezing the feature maps of a CNN into a lower-dimensional space, followed by applying an excitation function to re-weight the feature maps. This enables CNN to concentrate on the crucial features in an image, effectively ignoring the less relevant ones. The squeeze operation is accomplished by creating channel-wise statistics using GAP. The squeeze transform, denoted as F_{sq}, converts feature mappings U into global one-dimensional feature vectors, and it generates a statistic $z \in R^C$ by compressing U with spatial dimensions $H \times W$, which can be expressed as follows.

$$z_c = F_{sq}(u_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i,j) \tag{1}$$

An excitation operation is introduced to capture dependencies between channels to utilize the information gathered through the squeeze operation. Using the self-gating method with dual FC-layers, the excitation operation was developed for performing evaluation of weight on all channels for adaptive featured recalibrations. FC layers are employed to capture interdependence among channels and generate weights for each channel. The application of a Sigmoid activation function allows for acquiring weights within the interval of 0 to 1, which is used to signify the relative significance of each channel. The channel weights that have been obtained are multiplied with the original feature map in an element-wise manner, enabling the network to selectively enhance or reduce the influence of certain channels.

$$s = F_{ex}(z, W) = \sigma(g(z, W)) = \sigma(W_2 \delta(W_1 z)) \tag{2}$$

Here, δ means ReLU, $W_1 \in \mathbb{R}^{\frac{c}{r} \times c}$, $W_2 \in \mathbb{R}^{c \times \frac{c}{r}}$ and r represents a ratio of dimensionality reduction. The final outcome of the block is attained by adjusting the transformation's output through rescaling using the activations where F_{scale} refers to channel-wise multiplication between the feature map.

Efficient Channel Attention

ECA-Net [38], also known as Efficient Channel Attention Network, significantly improves the performance of CNNs by introducing the "Efficient Channel Attention" module. This module enables CNNs to efficiently capture interactions between channels by utilizing fast 1D convolution. The size of the convolutional kernel can be dynamically determined through a non-linear mapping of the channel dimension. This approach improves efficiency without sacrificing the dimensionality reduction characteristic of Squeeze and Excitation (SE) networks. It achieves this by adaptively re-calibrating feature responses through the application of GAP and fully connected layers. In the context of an input feature map X with shape (C, H, W), where C represents the number of channels, H represents the height, and W represents the width, a GAP operation is applied to the input feature map. This computes the average value for each channel, resulting in a channel-wise statistic with shape $(C, 1, 1)$, where,

$$Z(X) = \frac{1}{WH} \sum_{i=1,j=1}^{W,H} X_{ij} \tag{3}$$

denotes a channel-wise GAP. After that, the determination of the convolution kernel size K is based on the channel dimension C. This adaptive process ensures that the network performs a fast one-dimensional convolution operation of size K. Subsequently, a Sigmoid function is applied to learn channel attention, enabling the network to selectively emphasize important channels.

The convolution kernel size K plays a crucial role in determining the local cross-channel interaction coverage and is closely linked to the channel dimension C. Typically, the channel dimension is set to an integer power of 2, establishing a mapping relationship between K and C, which can be described as follows:

$$C = \phi(K) = 2^{\gamma * K - b} \tag{4}$$

The adaptive local convolution kernel size K is calculated using the following approach:

$$K = \varphi(C) = \psi \left| \frac{log_2(C)}{\gamma} + \frac{b}{\gamma} \right|_{odd} \tag{5}$$

where $|t|_{odd}$ denotes the nearest odd number t.

However, the authors did not provide a clear justification for the adaptive kernel size function $\psi(C)$, particularly regarding the underlying justification for the default values of γ and b. Additionally, the authors have not provided an explanation for their choice to set the kernel size to the default value of 3, rather than utilizing the adaptive function. As a result, we integrated BO to determine the kernel size in the ECA block, with a search space of K = 3 and K = 5. BO determined that K = 5 provided the optimal solution for our model and the integrated datasets.

The attention mechanism plays a vital role in focusing on important information. The ECA mechanism, as an efficient attention method with fewer parameters and superior performance, further enhances attention by emphasizing different feature representations and generating attention weights between channels. Incorporating the ECA mechanism enhances attention capability, leading to improved performance in various tasks. In our fire image classification task, we employed a combination of SE-Net and ECA-Net to tackle the challenge posed by highly diverse pixel values in fire-affected areas within the image datasets used. This approach was selected to ensure the production of dependable and efficient feature maps across all network channels. The SE block was particularly designed to capture intricate channel dependencies by integrating FC layers and bringing in adaptiveness and non-linear interactions. Moreover, the ECA-Net technique prioritizes the modeling of channel interactions by utilizing a 1D convolutional layer that imposes a lower computational cost.

3.3.4. Bayesian Optimization:

Bayesian optimization (BO) [22] is a technique that seeks to find the best possible solution for an objective function, $g(x)$, by incorporating prior knowledge and updating it based on observed data. This approach combines prior beliefs about the function with information gained from evaluations of $g(x)$ to refine the approximation of $g(x)$ and improve its accuracy. In addition, BO uses an acquisition function to guide the search process by selecting sampling points where there is a higher probability of finding improvements over the current best observation. As an example, suppose we have the objective function $g(x)$ and the estimated improvement (EI) based on the posterior distribution function Q. In this case, the expression for EI(x, Q) can be defined as follows:

$$EI(x, Q) = E_Q[max(0, \mu_Q(x_{best}) - g(x))] \tag{6}$$

where x_{best} is the location of the lowest posterior mean and $\mu_Q(x_{best})$ is the lowest value of the posterior mean.

In our research, we have integrated the BO algorithm to optimize various parameters, including the number of filters in the convolution block, kernel sizes, the number of neurons in the dense block, learning rates, and the dropout rate.

4. Result Analysis

This section focuses on evaluating the performance of the proposed model for classifying fire in various forest fire datasets. The proposed approach's performance was assessed, providing insights into its effectiveness in accurately classifying forest fire-related images on both FLAME and DeepFire datasets. The evaluation was conducted using the Kaggle virtual environment, utilizing a P100 GPU.

4.1. Hyper-Parameter Selection

Optimized hyperparameters can lead to significant improvements in the model's performance on both training and unseen data. By utilizing Bayesian optimization, we selected the optimized hyperparameters for our model. An overview of the model learning settings is shown in Table 2.

Table 2. Overview of model learning settings.

Dataset	Loss	Optimizer	LR	Epochs	Batch Size	Input Size
FLAME	Binary Cross-entropy	AdamW	1×10^{-5}	50	16	224 × 224
DeepFire	Binary Cross-entropy	Adam	1×10^{-5}	50	16	224 × 224

The ACNet in the proposed model is composed of four convolutional layers, each with varying numbers of filters, which are shown in Table 3. During the process of hyperparameter tuning using BO, we defined the filter size ranges for the convolution layers as follows: minimum values to 16, 32, 68, and 128, while the maximum values were set to 32, 68, 128, and 268, respectively, step sizes were 4, 8, 20, and 28. After extensive exploration and optimization, we determined that selecting 32, 64, 128, and 256 as the filter size values provided the best results for our objective. We experimented with kernel sizes 3, 5, and 7 in every convolution layer and 3, 5 in ECA block using BO and determined that a kernel size of 3 generates better results for convolution layer and a kernel size of 5 generates better results for ECA block. Each convolutional layer is subsequently followed by a max pooling operation with a pool size of 2.

Table 3. Attention connected network block optimization parameters.

No.	Convolutional Layer				Max Pooling	Kernel Size in ECA
	No. of Filters			Kernel Size		
	Max	Min	Step		(Pool Size)	
1st	32	16	4	3, 5, 7	2	3, 5
2nd	64	32	8	3, 5, 7	2	3, 5
3rd	128	64	20	3, 5, 7	2	3, 5
4th	256	128	28	3, 5, 7	2	3, 5

The dense block consists of two fully connected layers. In the initial dense layer, BO was utilized to determine the optimal number of neuron units. The search space for the dense layer was defined, ranging from a minimum of 128 units to a maximum of 512 units, in increments of 64 units as described in Table 4. BO demonstrated that utilizing 512 units provided the most optimal outcomes. For the final dense layer, we utilized the Sigmoid activation function to accurately classify the Fire and No-fire classes. The search space has been defined for BO regarding dropout rates, with a maximum dropout rate set at 0.4, a minimum rate of 0.25, and a step size of 0.05 shown in Table 5. Subsequently, BO provided the optimal dropout rate, which turned out to be 0.4. These values reflect the range of dropout rates employed in the model, allowing for regularization and reducing overfitting by randomly dropping out a fraction of input units. Additionally, the learning rate provides information about the learning rate used in the model's optimization process. This parameter represents the rate at which the model updates its internal parameters during training. The Table 6 states that two learning rates were considered in the proposed model which are 1 × 10^{-4} and 1 × 10^{-5}. This indicates that the model was trained using both learning rates, likely to explore the impact of different rates on the model's performance and convergence. After optimization using BO, a learning rate of 1 × 10^{-5} resulted in the most optimal outcomes and showed the best results for our objective. The search space for BO encompassed four optimizers: Adam, AdamW, SGD, and RMSprop as presented in Table 7. BO's analysis disclosed that AdamW provided superior outcomes for the FLAME dataset, whereas for the DeepFire dataset, Adam was found to be the more effective choice.

Table 4. Dense block optimization parameters.

Dense Layer	**Units**		
	Max	**Min**	**Step**
1st	512	128	64

Table 5. Dropout rate optimization parameters.

Parameter	**Units**		
	Max	**Min**	**Step**
Dropout	0.4	0.25	0.05

Table 6. Learning rate optimization parameters.

Parameter	**Units**
Learning Rate	1 × 10^{-4}, 1 × 10^{-5}

Table 7. Optimizer optimization parameters.

Parameter	**Name**
Optimizer	Adam, AdamW, SGD, RMSprop

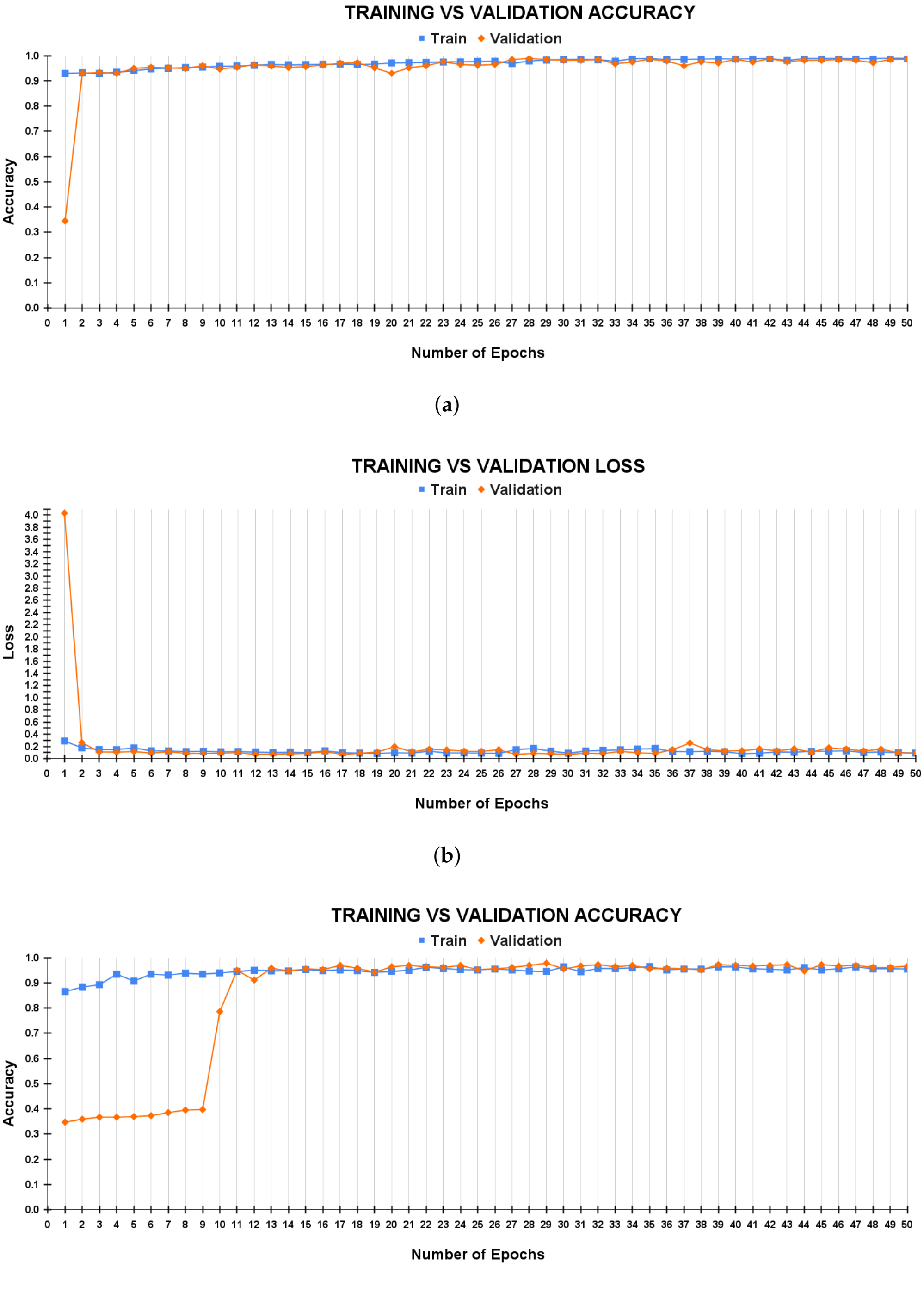
All the parameters shown in the previous tables were determined using BO which is a technique that combines probabilistic models and acquisition functions to efficiently search for optimal hyperparameters. This approach enables the model to adaptively and systematically explore the hyperparameter space, resulting in optimal parameter settings for enhanced model performance.

4.2. Comparative Evaluation of the Proposed Approach

The performance of the proposed architecture in accurately identifying forest fires was assessed using multiple metrics, including prediction accuracy, false positive rate, false negative rate, true negative rate, precision, recall, and F1 score. These metrics offer a thorough evaluation of the model's efficiency in classifying forest fires.

4.2.1. Analysis of Training and Testing Performance on Proposed Approach

The proposed model underwent training and testing on both the FLAME and DeepFire datasets, with a total of 50 epochs for each dataset. Figure 4a illustrates that using the FLAME dataset, the model demonstrated an impressive accuracy of 97.45%. The initial weights are initialized in a way that makes the model strongly biased towards the training data, it may quickly fit the training data during the first epoch, leading to a significant gap between training and validation performance, which is depicted in Figure 4a,b. In contrast, Figure 4c demonstrates that the model achieved a slightly lower accuracy of 95.97% using the DeepFire dataset. However, the accuracy graph itself might not immediately convey the exact accuracy value, it visually emphasizes the high level of accuracy attained by the model when working with the FLAME and DeepFire dataset. Randomness and noise in the training process can also impact early epoch performance, which is demonstrated in Figure 4c,d. At the start, the model's weights are initialized randomly, and this initial configuration can influence how quickly it learns to generalize well. Some random initialization may lead to faster convergence and better generalization in later epochs. These figures illustrate the trajectory of the model's loss curve. As training progresses, it gradually converges toward a better solution. It helps the model to improve the performance on both training and validation sets.

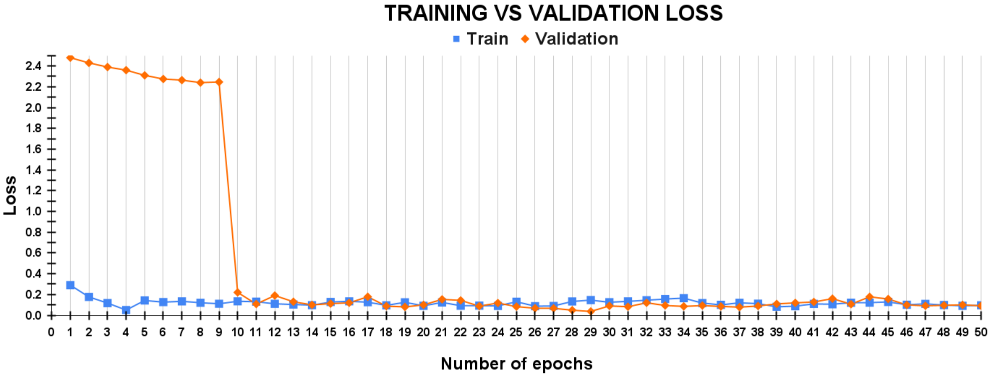


(**a**)

(**b**)

(**c**)

Figure 4. *Cont.*



(**d**)

Figure 4. Training vs. validation accuracy and loss curve for proposed architecture on FLAME dataset (**a**,**b**) and DeepFire dataset (**c**,**d**).

4.2.2. Performance Metrics

Different metrics are used to assess the performance of classification models by measuring different aspects of their predictive accuracy. Evaluating the performance of a model relies heavily on the metric of prediction accuracy in classifying a given problem. When employing a DNN approach, the primary goal is to attain a high level of prediction accuracy while simultaneously reducing the error rate. The equations below can be used to define the prediction accuracy and error rate.

$$PredictionAccuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

In the context of classification, TN (True Negative) refers to the correctly classified no-fire image number, while TP (True Positive) indicates the count of fire images accurately classified using the method. A false positive (FP) occurs when an image belonging to the no-fire class is incorrectly classified as fire, while a false negative (FN) happens when a fire image is mistakenly classified as no-fire.

Precision and recall are fundamental metrics for evaluating the performance of a classification model in a formal context. Precision quantifies the accuracy of positive predictions made by a model or classifier. It measures the proportion of true positive predictions (correctly identified positives) out of the total predicted positives. In other words, precision assesses the model's reliability and precision when predicting positive outcomes. On the other hand, recall, also known as sensitivity or true positive rate, measures the model's effectiveness in capturing positive instances by calculating the ratio of correctly predicted positive instances to the total actual positive instances. Also, the F1 score is an evaluation metric that effectively combines precision and recall to offer a well-rounded measure of a classification model's performance. By taking the harmonic mean of precision and recall, it provides a balanced measure.

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$Recall = TPR = \frac{TP}{TP + FN} \tag{9}$$

$$F1score = TPR = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{10}$$

4.2.3. K-Fold Cross-Validation

The k-fold cross-validation method is extensively employed in the field of ML for evaluating and validating the performance of a model. It helps in obtaining a more accurate understanding of how well the model is likely to perform on unseen data and aids in making informed decisions about model selection and hyperparameter tuning. This process entails dividing the dataset into K folds of equal size, where each fold is utilized alternately as a training set and a validation set. This process is repeated K times, with each fold being used as the validation set once. The importance of K-fold cross-validation lies in its ability to provide a more robust and reliable estimate of the model's performance. It helps to mitigate issues like overfitting and selection bias by ensuring that the model is evaluated on multiple different subsets of the data.

In our proposed model, we divided the dataset into four folds of equal size ensuring a comprehensive evaluation of the model's performance. By averaging the performance across the four iterations, we obtain a more representative and generalized assessment of the model's effectiveness, as it is evaluated on diverse subsets of the data. All data of K-fold validation are present in Table 8. After four iterations of K-fold cross-validation on the FLAME dataset, the model achieved an accuracy of 97.45%, precision of 98.20%, recall of 97.10%, and an F1-score of 97.12%. Similarly, on the DeepFire dataset, the model provided an accuracy of 95.97%, precision of 95.19%, recall of 96.01%, and an F1-score of 95.54%. These results demonstrate the strong performance and effectiveness of the model in accurately classifying and predicting the target variables for both datasets.

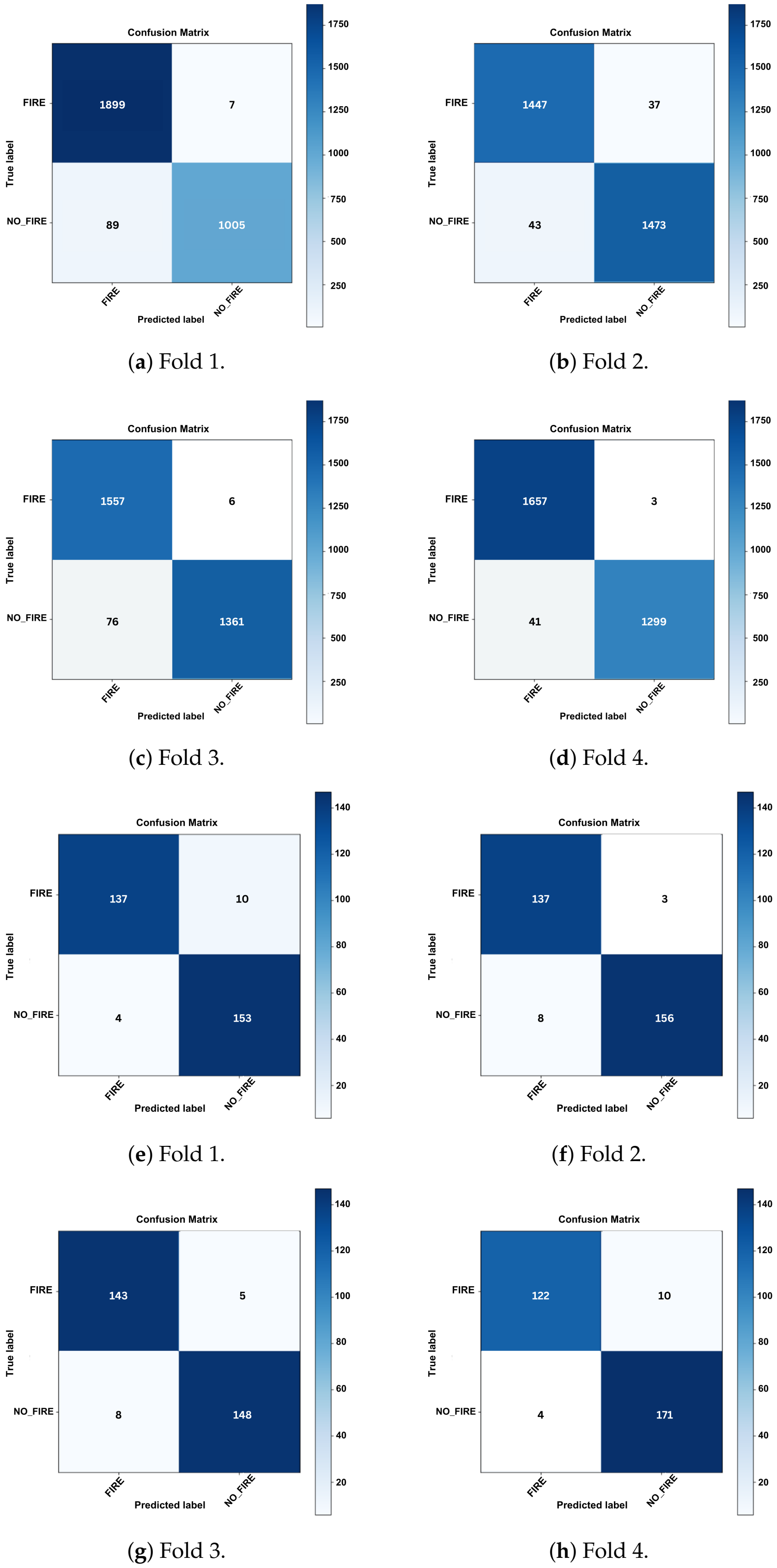
Table 8. K-fold.

Dataset	K-Fold	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
FLAME	1st	96.78	99.63	95.50	97.52
	2nd	97.29	97.48	97.17	97.29
	3rd	97.23	97.14	97.43	97.25
	4th	98.53	98.55	98.32	98.42
	Mean	97.45	98.20	97.10	97.12
DeepFire	1st	95.39	93.10	97.10	95.03
	2nd	96.38	97.80	95.52	96.13
	3rd	95.72	94.74	96.65	95.61
	4th	96.38	95.13	94.79	95.41
	Mean	95.97	95.19	96.01	95.54

4.2.4. Confusion Matrix

The confusion matrix serves as a tool for providing predictive analysis in forest fire classification, and provides a more comprehensive evaluation of the proposed method's performance, offering clarity in situations where accuracy alone may be ambiguous. In Figure 5, the confusion matrix represents the performance of the proposed model approach on both datasets. The diagonal elements of the matrix represent the number of correct predictions, while the off-diagonal elements represent the number of incorrect predictions. The larger values along the diagonal indicate that the model performed well, with few misclassifications.

The proposed model was evaluated using four-fold cross validation. This means that the data were split into four folds, and the model was trained on three folds and tested on the remaining fold. This process was repeated four times, and the results were averaged to obtain an estimate of the model's performance. The average accuracy of the model was 97.45 % on the FLAME dataset and 95.97 % on the DeepFire dataset.



(**a**) Fold 1. (**b**) Fold 2.

(**c**) Fold 3. (**d**) Fold 4.

(**e**) Fold 1. (**f**) Fold 2.

(**g**) Fold 3. (**h**) Fold 4.

Figure 5. Class-wise confusion matrix for proposed architecture on FLAME dataset (**a**–**d**) and DeepFire dataset (**e**–**h**).

4.2.5. ROC-AUC

The performance evaluation of the image classification method also incorporates additional metrics, namely the False Positive Rate (FPR), False Negative Rate (FNR), True Positive Rate (TPR), and True Negative Rate (TNR). These metrics provide valuable insights into the model's accuracy. The equations below define these metrics:

$$TNR = \frac{TN}{TN + FP} \tag{11}$$

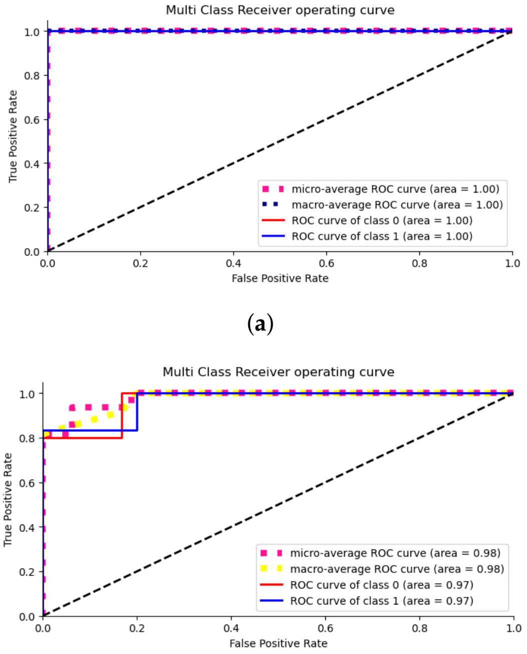
$$TPR = \frac{TP}{TP + FN} \tag{12}$$

$$FPR = 1 - TNR \tag{13}$$

$$FNR = 1 - TPR \tag{14}$$

The proposed method demonstrated strong performance on two different datasets. On the FLAME dataset, it achieved a True Negative Rate (TNR) of 95.5% and a True Positive Rate (TPR) of 99.3%. Additionally, on the DeepFire dataset, the method attained a TNR of 94.47% and a TPR of 96.82%. These metrics highlight the method's ability to accurately classify images.

The Receiver Operating Characteristic (ROC) curve is a graphical representation that showcases the predictive performance of a binary classifier as the prediction threshold is adjusted. The ROC curve is obtained by plotting TPR, also known as sensitivity or recall, against the FPR. The Area Under the Curve (AUC) is a metric that measures class separability, indicating the model's effectiveness in distinguishing between classes. A higher AUC score signifies superior predictive capabilities of the model. Figure 6 visually depicts the ROC curve generated by the proposed method. With an AUC value of 1.00, the model exhibits a perfect ability to correctly discriminate between positive and negative classes.



(**a**)

(**b**)

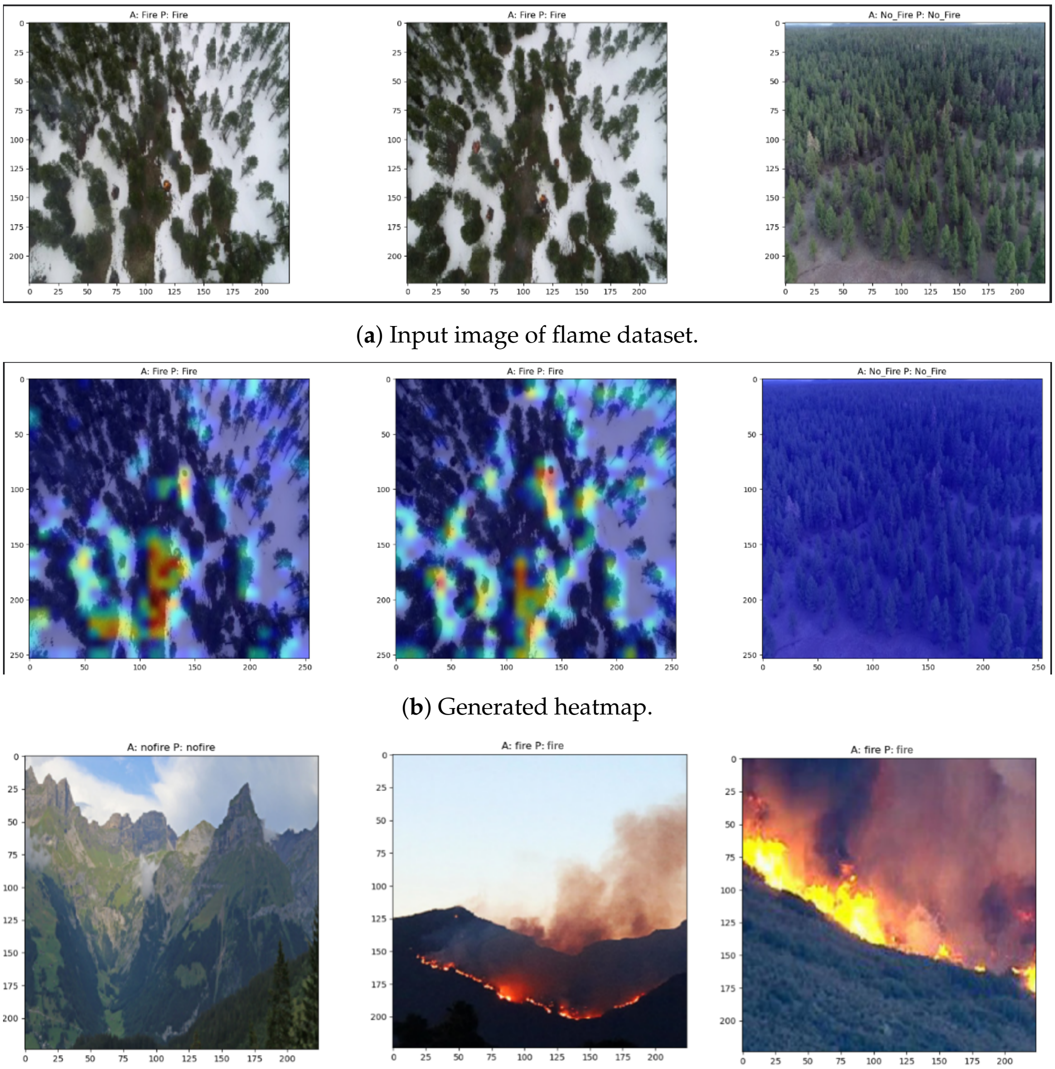
Figure 6. ROC curve on (**a**) FLAME dataset and (**b**) DeepFire dataset.

4.2.6. Heatmap Generation

To further facilitate visual interpretation, we utilized a visualization technique known as guided Gradient-weighted Class Activation Mapping (Grad-CAM) [23], which provides heatmaps with high resolution and distinct class discrimination. This technique allows us to visually interpret our model's predictions by generating a heat map representing the class activation map for a given input image.

The class activation map is a two-dimensional grid of feature scores associated with a specific output class. Each position on the grid indicates the importance of that class for the corresponding location in the image. When an image containing fire is input into the detection model, Grad-CAM generates a heat map that visually presents the level of similarity for each location within the image and the "fire" class. Darker colors on the heat map indicate a higher degree of similarity. This visualization technique enables us to be conscious of the specific local regions in the original image that played a significant role in the detection model's final classification decision.

In Figure 7, our work demonstrates the efficacy of the coordinated attention module by showcasing the regions of interest identified by our network. We observed that these regions closely corresponded to the areas depicting forest smoke and flames in the input images. This alignment between the identified regions of interest and the visual indicators of fire provides us with valuable insight into how our model focuses on the relevant areas when making predictions.

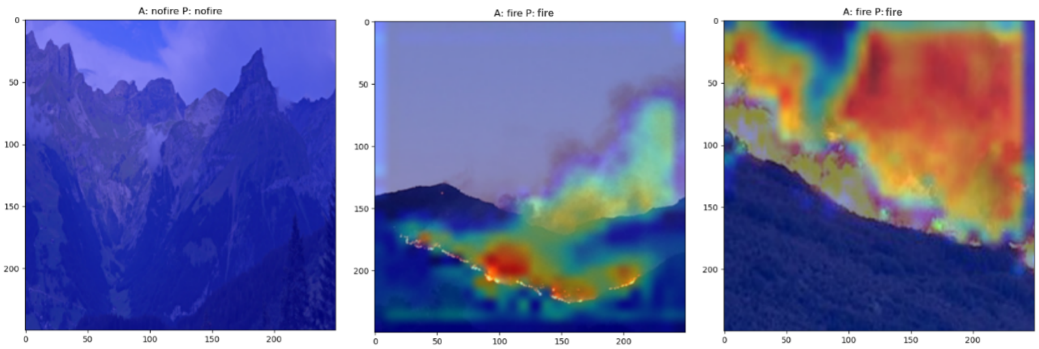


(**a**) Input image of flame dataset.

(**b**) Generated heatmap.

(**c**) Input image of DeepFire dataset.

Figure 7. *Cont.*



(**d**) Generated heatmap.

Figure 7. Showcasing the regions of interest on (**a**,**b**) FLAME dataset and (**c**,**d**) DeepFire dataset.

5. Discussion

Accurate wildfire classification is crucial, and our proposed dual-stream attention-guided approach has proven its exceptional performance on widely recognized two public datasets. It consistently outperforms most of the existing methods, thanks to its ability to capture both low-level and high-level dependencies within an attention-guided framework. This confirms its status as a novel wildfire classification solution, valuable for addressing wildfire classification and management challenges. For future enhancements, we plan to integrate transformers with larger datasets to boost accuracy and adaptability. We'll also work on reducing model parameters for improved efficiency and explore more effective attention mechanisms for even better classification precision. These improvements align with our commitment to advancing wildfire classification and real-world applications.

5.1. Performance Comparison with Other CNN Models

In this section, we present a performance comparison of our model with other CNN and attention-based models. While some models utilize the FLAME dataset, others employ the DeepFire dataset similar to ours. The comparison reveals that our model demonstrates the potential for superior performance compared to all the evaluated models, suggesting its effectiveness in forest fire classification tasks. Table 9 shows all the comparison data between the proposed and other mentioned CNN-based models. Though precision and recall serve as important metrics for evaluating all the ML and DL models, the majority of the reviewed papers did not explicitly mention these metrics. From Table 9, we can observe that our proposed method outperformed most of the notable existing methods in terms of all the metrics. However, the proposed method of S. Khan et al. [26] performed slightly better on the DeepFire dataset. However, the authors' lack of clarity regarding the model architecture is a significant limitation of their work. This lack of detail makes it difficult to understand and replicate their results, which is a major concern as it prevents other researchers from improving upon their work.

Table 9. Comparisons of the classification result for the proposed and conventional methods in all the employed datasets ('-' denotes not mentioned).

Catagory	**Methodology**	**Dataset**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
CNN Based	Pretrained VGG19 with customized classifier [24]	DeepFire	95.00	94.96	95.72	94.21
	FFireNet (Pretrained MobileNetV2 with additional dense layers) [26]	DeepFire	98.42	-	97.42	99.47
	Pretrained ResNet50 with customized fully connected layers [27]	FLAME	88.00	-	-	-
	Ensemble model (EfficientNet-B5 , DenseNet-201) [28]	FLAME	85.12	84.77	-	-
	Xception [29]	FLAME	76.23	-	-	-
Attention Based	ResNet50 with DSA (Dual Semantic Attention) [19]	FLAME	93.65	-	-	-
Proposed Model	-	FLAME	97.45	98.20	97.10	97.12
	-	DeepFire	95.97	95.19	96.01	95.54

5.2. Ablation Study

To assess the efficacy and interactivity of the components used in the proposed architecture, we conducted a comprehensive ablation study. The results of this study, showcasing the experimental outcomes, are outlined in Table 10. This ablation study allowed us to assess the individual contributions of each component and gain insights into their significance within the overall architecture. Through comparative analysis with methods mentioned in Table 10, we see that our architecture outperforms them in terms of overall performance. To assess the contributions and importance of specific components in our model, we conducted an ablation study where we systematically removed essential elements. Our proposed model comprises two important elements: a pre-trained EfficientNet B7 backbone and a customized ACM that incorporates SE and ECA mechanisms. To evaluate the effectiveness of these components, we performed the ablation study using the FLAME dataset exclusively.

Table 10. Experimental results from the ablation study.

Dataset	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
FLAME	Proposed Architecture w/o pre-trained EfficientNet B7	95.21	94.25	95.23	95.50
	Proposed Architecture w/o ACM	94.11	94.32	94.45	94.09
	Proposed Architecture w/o ECA in ACM	97.06	97.08	96.11	96.33
	Proposed Architecture w/o SE in ACM	96.54	96.68	96.39	96.52
	Proposed Architecture	97.45	98.20	97.10	97.12

The stepwise inclusion of each component in the proposed architecture resulted in a noticeable improvement in classification performance. The model achieved a remarkable accuracy score of 97.45%, a precision score of 98.20%, and a recall of 97.10% when all components were present, outperforming other models. Notably, the absence of the pre-trained backbone led to a lower accuracy of 95.21%, highlighting the substantial impact of the pre-trained backbone on the overall performance. Similarly, excluding the ACM resulted in decreased performance, with an accuracy of 94.11%, precision of 94.32%, recall of 94.45%, and an F1-score of 94.09%. This emphasized the crucial role played by the ACM in enhancing the model's predictive capabilities. The study revealed that the presence of both the ECA and SE components within ACM positively influenced the model's performance. Upon comparing the proposed model's performance with the model without the ECA component in ACM, we observed a decrease of 0.31% in accuracy. In contrast, when comparing the proposed model with the model without the SE component, there was a larger decrease of 0.83% in accuracy.

Figure 8 presents the ablation study results using heatmap generation techniques, which allow us to visualize how the model recognizes and highlights regions of interest. These results suggest that the absence of the SE component had a more significant impact on the model's performance, highlighting its influential role compared to other components. According to the findings of the ablation study, it is evident that every component incorporated in the proposed model carries significant importance.

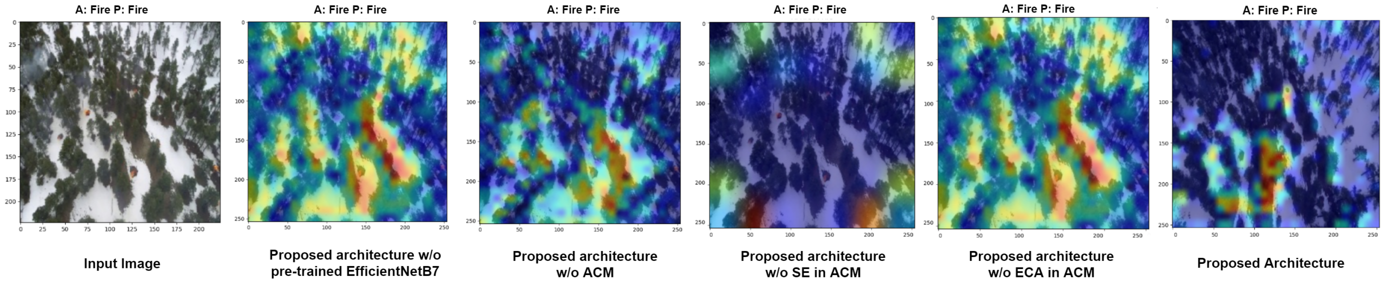

A: Fire P: Fire
A: Fire P: Fire
A: Fire P: Fire
A: Fire P: Fire
A: Fire P: Fire
A: Fire P: Fire
Input Image
Proposed architecture w/o pre-trained EfficientNetB7
Proposed architecture w/o ACM
Proposed architecture w/o SE in ACM
Proposed architecture w/o ECA in ACM
Proposed Architecture

Figure 8. *Cont.*

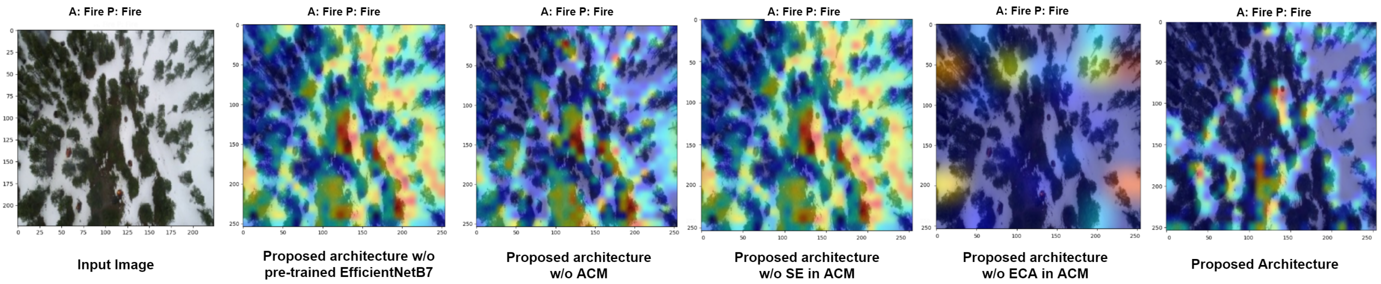

A: Fire P: Fire
A: Fire P: Fire
A: Fire P: Fire
A: Fire P: Fire
A: Fire P: Fire
A: Fire P: Fire
Input Image
Proposed architecture w/o pre-trained EfficientNetB7
Proposed architecture w/o ACM
Proposed architecture w/o SE in ACM
Proposed architecture w/o ECA in ACM
Proposed Architecture

Figure 8. Qualitative comparison of predictions of regions of interest among proposed model without different components.

6. Conclusions

In this paper, we present a novel model that incorporates a pre-trained EfficientNetB7, a customized Attention Guided Network called ACNet, and the BO technique. This model provides high accuracy for forest fire classification. To enhance the interpretability of this model, we implemented GRAD-CAM, which allows us to localize the fire within the feature map. This enables a deeper understanding of the model's decision-making process and provides valuable insights for fire detection. Furthermore, k-fold cross-validation was conducted to rigorously assess the model's performance. On the FLAME dataset, the model attained an accuracy of 97.45%, precision of 98.20%, recall of 97.10%, and an F1-score of 97.12%. Similarly, on the DeepFire dataset, the model demonstrated an accuracy of 95.97%, precision of 95.19%, recall of 96.01%, and an F1-score of 95.54%. The F1-score of the both dataset indicates that the model achieved a strong balance between precision and recall. This is a positive sign as it suggests that the model is effective at both correctly identifying positive cases (fire region) and minimizing false positives. The high accuracy also indicates overall strong performance. Additionally, the ablation study delves into the contributions of each individual component, providing a deeper understanding of how they impact the overall performance of the model. The ablation study showed that our proposed ACM plays a vital role in the model's performance. Without the ACM, the model's F1-score dropped to 94.09%, which is significantly lower than the model's F1-score with the ACM. In other words, the ACM is a key component of the model, and it is essential for achieving good performance. The numerical results along with interpretation through GRAD-CAM provides proof of our proposed model's efficiency in classifying forest fire. In our future work, we hope to optimize training time for larger network sizes, enabling the training of more powerful and accurate models. We also intend to enhance preprocessing techniques to improve classification outcomes, facilitating more effective learning and producing more precise results. Furthermore, we plan to reduce the model's computational cost, allowing for seamless integration into mobile devices.

Author Contributions: Conceptualization, M.R.A. and A.M.I.; methodology, A.M.I. and F.B.M.; validation, A.M.I. and F.B.M.; formal analysis, A.M.I. and M.R.A.; investigation, A.I.J. and J.R.U.; data curation, A.M.I., F.B.M., A.M.I. and J.R.U.; writing—original draft, A.M.I., F.B.M., A.I.J. and M.R.A.; writing—review & editing, S.I., S.S. and A.K.M.M.I.; supervision, M.R.A., S.I., S.S. and A.K.M.M.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by Institute for Advanced Research Publication Grant of United International University, Ref. No.: IAR-2023-Pub-029.

Data Availability Statement: The FLAME dataset can be accessed at: https://ieee-dataport.org/open-access/flame-dataset-aerial-imagery-pile-burn-detection-using-drones-uavs (accessed on 15 June 2023). The Deepfire dataset can be accessed at: https://www.kaggle.com/datasets/alik05/forest-fire-dataset (accessed on 15 June 2023).

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

References

1. Boer, M.M.; Resco de Dios, V.; Bradstock, R.A. Deep learning based forest fire classification and detection in satellite images. *Nat. Clim. Chang.* **2020**, *10*, 171–172. [CrossRef]
2. Zhang, L.; Wang, M.; Ding, Y.; Bu, X. MS-FRCNN: A Multi-Scale Faster RCNN Model for Small Target Forest Fire Detection. *Forests* **2023**, *14*, 616. [CrossRef]
3. MacCarthy, J.; Richter, J.; Tyukavina, S.; Weisse, M.; Harris, N. The Latest Data Confirms: Forest Fires Are Getting Worse. World Resources Institute. Available online: https://www.wri.org/insights/global-trends-forest-fires (accessed on 29 August 2023).
4. National Center for Environmental Information, Wildfire Report. April 2023. Available online: https://www.ncei.noaa.gov/access/monitoring/monthly-report/fire/202304 (accessed on 15 June 2023).
5. Reuters. Canada Wildfires: What Are the Causes and When Will It End? Available online: https://www.reuters.com/world/americas/canadas-record-wildfire-season-whats-behind-it-when-will-it-end-2023-08-17/ (accessed on 19 August 2023).
6. Glover, D. *Indonesia's Fires and Haze: The Cost of Catastrophe*; International Development Research Centre (IDRC): New Delhi, India, 2006.
7. Yeung, J. Indonesian Forests are Burning, and Malaysia and Singapore are Choking on the Fumes. Available online: https://edition.cnn.com/2019/09/11/asia/malaysia-singapore-pollution-intl-hnk/index.html (accessed on 19 September 2019).
8. Gaur, A.; Singh, A.; Kumar, A.; Kulkarni, K.S.; Lala, S.; Kapoor, K.; Srivastava, V.; Kumar, A.; Mukhopadhyay, S.C. Fire sensing technologies: A review. *IEEE Sens. J.* **2019**, *19*, 3191–3202. [CrossRef]
9. Celik, T.; Demirel, H. Fire detection in video sequences using a generic color model. *Fire Saf. J.* **2009**, *44*, 147–158. [CrossRef]
10. 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]
11. Ghali, R.; Jmal, M.; Mseddi, W.S.; Attia, R. Recent advances in fire detection and monitoring systems: A review. In Proceedings of the 8th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT'18), Genoa, Italy, 18–20 December 2018; Volume 1, pp. 332–340.
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. Hamme, D.V.; Veelaert, P.; Philips, W.; Teelen, K. Fire detection in color images using Markov random fields. In *Advanced Concepts for Intelligent Vision Systems: 12th International Conference, ACIVS 2010, Sydney, Australia, December 13–16. 2010, Proceedings, Part II 12*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 88–97.
14. Bedo, M.V.N.; de Oliveira, W.D.; Cazzolato, M.T.; Costa, A.F.; Blanco, G.; Rodrigues, J.F., Jr.; Traina, A.J.M.; Traina, C., Jr. Fire detection from social media images by means of instance-based learning. In *International Conference on Enterprise Information Systems*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 23–44.
15. Ko, B.; Cheong, K.-H.; Nam, J.-Y. Early fire detection algorithm based on irregular patterns of flames and hierarchical Bayesian Networks. *Fire Saf. J.* **2010**, *45*, 262–270. [CrossRef]
16. Lee, W.; Kim, S.; Lee, Y.-T.; Lee, H.-W.; Choi, M. Deep neural networks for wildfire detection with unmanned aerial vehicle. In Proceedings of the 2017 IEEE International Conference on Consumer Electronics (ICCE), Berlin, Germany, 3–6 September 2017; pp. 252–253.
17. Rahul, M.; Saketh, K.S.; Sanjeet, A.; Naik, N.S. Early detection of forest fire using deep learning. In Proceedings of the 2020 IEEE Region 10 Conference (TENCON), Osaka, Japan, 16–19 November 2020; pp. 1136–1140.
18. Wu, H.; Li, H.; Shamsoshoara, A.; Razi, A.; Afghah, F. Transfer learning for wildfire identification in UAV imagery. In Proceedings of the 2020 54th Annual Conference on Information Sciences and Systems (CISS), Princeton, NJ, USA, 8–20 March 2020; pp. 1–6.
19. Guan, Z.; Miao, X.; Mu, Y.; Sun, Q.; Ye, Q.; Gao, D. Forest fire segmentation from Aerial Imagery data Using an improved instance segmentation model. *Remote Sens.* **2022**, *14*, 3159. [CrossRef]
20. Li, T.; Zhu, H.; Hu, C.; Zhang, J. An attention-based prototypical network for forest fire smoke few-shot detection. *J. For. Res.* **2022**, *33*, 1493–1504. [CrossRef]
21. Tan, M.; Le, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; Chaudhuri, K., Salakhutdinov, R., Eds.; Volume 97, pp. 6105–6114.
22. Zhang, L.; Wang, M.; Fu, Y.; Ding, Y. Bayesian optimization with unknown constraints. *arXiv* **2014**, arXiv:1403.5607.
23. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Vancouver, BC, Canada, 7–14 July 2001.
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. Namburu, A.; Selvaraj, P.; Mohan, S.; Ragavanantham, S.; Eldin, E. Forest Fire Identification in UAV Imagery Using X-MobileNet. *Electronics* **2023**, *12*, 733. [CrossRef]
26. Khan, S.; Khan, A. FFireNet: Deep Learning Based Forest Fire Classification and Detection in Smart Cities. *Symmetry* **2022**, *14*, 2155. [CrossRef]
27. Treneska, S.; Stojkoska, B.R. Wildfire detection from UAV collected images using transfer learning. In Proceedings of the 18th International Conference on Informatics and Information Technologies, Skopje, North Macedonia, 18–23 September 2021; pp. 6–7.

28. Ghali, R.; Akhloufi, M.A.; Mseddi, W.S. Deep learning and transformer approaches for UAV-based wildfire detection and segmentation. *Sensors* **2022**, *22*, 1977. [CrossRef] [PubMed]
29. Shamsoshoara, A.; Afghah, F.; Razi, A.; Zheng, L.; Fulé, P.Z.; Blasch, E. Aerial imagery pile burn detection using deep learning: The FLAME dataset. *Comput. Netw.* **2021**, *193*, 108001. [CrossRef]
30. Priya, R.; Vani, K. Deep learning based forest fire classification and detection in satellite images. In Proceedings of the 2019 11th International Conference on Advanced Computing (ICoAC), Chennai, India, 18–20 December 2019; pp. 61–65.
31. Kurasinski, L.; Tan, J.; Malekian, R. Using neural networks to detect fire from overhead images. *Wirel. Pers. Commun.* **2023**, *130*, 1085–1105. [CrossRef]
32. Akagic, A.; Buza, E. LW-FIRE: A Lightweight Wildfire Image Classification with a Deep Convolutional Neural Network. *Appl. Sci.* **2022**, *12*, 2646. [CrossRef]
33. Tang, Y.; Feng, H.; Chen, J.; Chen, Y. ForestResNet: A deep learning algorithm for forest image classification. *J. Phys. Conf. Ser.* **2021**, *2024*, 012053. [CrossRef]
34. Wang, Y.; Dang, L.; Ren, J. Forest fire image recognition based on convolutional neural network. *J. Algorithms Comput. Technol.* **2019**, *13*, 1748302619887689. [CrossRef]
35. Dutta, S.; Ghosh, S. Forest fire detection using the combined architecture of separable convolution and image processing. In Proceedings of the 2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA), Riyadh, Saudi Arabia, 6–7 April 2021; pp. 36–41.
36. Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A comprehensive survey on transfer learning. *Proc. IEEE* **2020**, *109*, 43–76. [CrossRef]
37. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
38. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 11534–11542.

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.