forests

MDPI

Article

FireViT: An Adaptive Lightweight Backbone Network for Fire Detection

Pengfei Shen [1], Ning Sun [1,2,*], Kai Hu [1], Xiaoling Ye [1], Pingping Wang [3], Qingfeng Xia [2] and Chen Wei [1]

[1] Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology, Information and Systems Science Institute, Nanjing University of Information Science and Technology, Nanjing 210044, China; 20211249578@nuist.edu.cn (P.S.); 001600@nuist.edu.cn (K.H.); 000510@nuist.edu.cn (X.Y.); 202312490377@nuist.edu.cn (C.W.)
[2] School of Automation, Wuxi University, Wuxi 214105, China; xqf@cwxu.edu.cn
[3] Fire Research Institute, Shanghai 200030, China; wangpingping@shfri.cn
* Correspondence: 001764@cwxu.edu.cn

Abstract: Fire incidents pose a significant threat to human life and property security. Accurate fire detection plays a crucial role in promptly responding to fire outbreaks and ensuring the smooth execution of subsequent firefighting efforts. Fixed-size convolutions struggle to capture the irregular variations in smoke and flames that occur during fire incidents. In this paper, we introduce FireViT, an adaptive lightweight backbone network that combines a convolutional neural network (CNN) and transformer for fire detection. The FireViT we propose is an improved backbone network based on MobileViT. We name the lightweight module that combines deformable convolution with a transformer as th DeformViT block and compare multiple builds of this module. We introduce deformable convolution in order to better adapt to the irregularly varying smoke and flame in fire scenarios. In addition, we introduce an improved adaptive GELU activation function, AdaptGELU, to further enhance the performance of the network model. FireViT is compared with mainstream lightweight backbone networks in fire detection experiments on our self-made labeled fire natural light dataset and fire infrared dataset, and the experimental results show the advantages of FireViT as a backbone network for fire detection. On the fire natural light dataset, FireViT outperforms the PP-LCNet lightweight network backbone for fire target detection, with a 1.85% increase in mean Average Precision (mAP) and a 0.9 M reduction in the number of parameters. Additionally, compared to the lightweight network backbone MobileViT-XS, which similarly combines a CNN and transformer, FireViT achieves a 1.2% higher mAP while reducing the Giga-Floating Point Operations (GFLOPs) by 1.3. FireViT additionally demonstrates strong detection performance on the fire infrared dataset.

Keywords: CNN and transformer; lightweight; fire detection

Citation: Shen, P.; Sun, N.; Hu, K.; Ye, X.; Wang, P.; Xia, Q.; Wei, C. FireViT: An Adaptive Lightweight Backbone Network for Fire Detection. *Forests* **2023**, *14*, 2158. https://doi.org/10.3390/f14112158

Academic Editor: José Aranha

Received: 29 September 2023
Revised: 25 October 2023
Accepted: 27 October 2023
Published: 30 October 2023

1. Introduction

Hazards caused by fire are a serious threat to human life and property. According to data from the Global Disaster Database, from 2013 to 2022 the average number of deaths and missing persons due to forest and grassland fires alone reached 904,000 people. Data released by Global Forest Watch (GFW) and the World Resources Institute (WRI) indicate that on a global scale the forest area destroyed by wildfires is now double what it was at the beginning of this century. According to satellite data, the annual forest area destroyed by wildfires has increased by approximately 3 million hectares compared to the year 2001. In addition to forest fires, the increasing impact of other types of fires, such as electrical fires, is becoming more severe as society continues to progress and develop. The frequency of these incidents is on the rise. Real-time monitoring of fire-prone areas, timely fire alarms, and rapid localization of fire incidents are of paramount importance for safeguarding human life, property, and industrial safety.

Traditional fire detection methods primarily involve contact-based fire detectors, such as carbon monoxide sensors, temperature sensors, smoke detectors, etc. Rachman, F. et al. [1] proposed a fuzzy logic-based early fire detection system using KY-026 (fire detection), MQ-9 (smoke detection), and DS18b20 (temperature detection) sensors. Huang Ye et al. [2] proposed a wireless fire detection node design method based on multi-source sensor data fusion and provided a complete hardware selection and software data fusion processing method. Solorzano Soria, A.M. et al. [3] proposed a gas sensor-based array to speed up fire alarm response. Li Yafei et al. [4] developed a mid-infrared carbon monoxide (CO) and carbon dioxide (CO_2) dual gas sensor system for early fire detection. Liu Xiaojiang et al. [5] proposed a sensor optimization strategy and an intelligent fire detection method based on the combination of particle swarm optimization algorithm. Although contact fire detectors are commonly used in various public scenes, their detection range is limited to small indoor spaces, and it is difficult to apply them to large indoor spaces and outdoor open spaces where open flames are strictly prohibited. Moreover, traditional contact fire detectors are prone to age-related failures and require a lot of manpower and resources for maintenance and management.

Compared to contact fire detection using sensors, non-contact video fire detection technology has the advantages of no additional hardware, intuitive and comprehensive fire information, and large detection range. While real-time monitoring of fire appears to be a binary classification problem for images, in practice it requires further detection of fire images based on the classification. The use of fire image detection is justified due to the extensive coverage of video surveillance systems. Early fire incidents are often challenging to detect, and sometimes fires can escalate to an uncontrollable stage within a short timeframe. Therefore, relying solely on image classification is insufficient for accurately pinpointing the actual location of a fire. This limitation could undoubtedly hinder the timely response and effective management of fire incidents. Traditional fire target detection algorithms include region selection, feature extraction, and classifier design. Qiu, T. et al. [6] proposed an adaptive canny edge detection algorithm for fire image processing. Ji-neng, O. et al. [7] proposed an early flame detection method based on edge gradient features. Khalil, A. et al. [8] proposed a fire detection method based on multi-color space and background modeling. However, in traditional fire target detection algorithms, manually designed features lack strong generalization and exhibit limited robustness. The emergence of CNNs has gradually replaced traditional handcrafted feature methods, offering superior generalization and robustness compared to traditional fire detection approaches. Majid, S. et al. [9] proposed an attention-based CNN model for the detection and localization of fires. Chen, G. et al. [10] proposed a lightweight model for forest fire smoke detection based on YOLOv7. Dogan, S. et al. [11] proposed an automated accurate fire detection system using ensemble pretrained residual network. In recent years, a new generation of transformer-based deep learning network architectures has gradually started to shine. Li, A. et al. [12] proposed a combination of BiFPN and Swin transformer for the detection of smoke from forest fires. Huang, J. et al. [13] proposed a small target smoke detection method based on a deformable transformer. Although these transformer-based network architectures have achieved good results in fire detection, they tend to be more complex (i.e., the number of parameters reaches about 20 M or 30 M) and not very lightweight. In order to ensure a lightweight transformer architecture-based model, scholars have started to research solutions such as EfficientViT [14], EfficientFormerV2 [15], MobileViT [16], etc., although these network models have not become widely used in fire detection to date.

Most of the methods that utilize deep learning for fire target detection use rectangular convolution of fixed shapes for feature extraction of smoke and flame in fires; however, it is well known that smoke and flame features in a fire situation are scattered and irregular, which is undoubtedly a very challenging task in fire target detection. Therefore, in this paper we propose a lightweight adaptive backbone network called FireViT using deformable convolution [17] combined with transformer for fire detection to better adapt

to the irregularly varying smoke and flames in fire scenarios. The adaptive lightweight backbone network, FireViT that we propose here is capable of meeting the requirements of various other application scenarios.

The contributions of our paper are as follows:

- An adaptive lightweight backbone network consisting of deformable convolution combined with a transformer, which we name FireViT, is proposed for smoke and flame detection in fire. Our proposed DeformViT block is the main module in FireViT.
- An improved adaptive activation function, AdaptGELU, is proposed to increase the nonlinear representation of the model and further enhance the accuracy of the network.
- Considering the relatively small number of publicly available labeled fire datasets, we collected and built one of the richest labeled fire datasets with the largest number of fire scenes and fire images to evaluate our model. Our labeled fire dataset contains a fire natural light dataset and fire infrared dataset.

The rest of this paper is organized as follows. Section 2 presents previous works on the backbone network and the target detection header in the target detection algorithm that are related to this paper. Section 3 presents FireViT, a lightweight backbone network consisting of deformable convolution combined with a transformer that can be used for fire detection, along with AdaptGELU, an improved adaptive activation function. Section 4 discusses the process of determining the FireViT backbone network using the AdaptGELU activation function proposed in Section 3 and verifies the validity of the model by comparing it to other mainstream lightweight backbone networks for fire detection experiments on self-made labeled fire datasets. Finally, in Section 5, we summarize and conclude the paper.

2. Related Work

In this section, we present previous research work related to this paper on lightweight backbone networks and target detection headers in target detection algorithms.

2.1. MobileViT

The emergence of ViT [18] has led people to realize the tremendous potential of transformers in the field of computer vision. The transformer architecture has become a new neural network paradigm in the field of computer vision, following the advent of CNNs, with and more researchers starting to use networks with the transformer architecture. However, although transformers are powerful, they have a number of problems; the pure transformer model structures are usually bulky and not very lightweight; furthermore, inductive bias is a form of prior knowledge, and unlike CNNs, transformers do not have the same kind of induction bias, meaning that transformers require a substantial amount of data to learn such prior information.

The inductive bias of CNNs can generally be categorized into two types. The first is locality; CNNs convolve input feature maps using a sliding window approach, which means that objects that are closer together exhibit stronger correlations. Locality helps to control the complexity of the model. The second type is translation equivariance; regardless of whether object features in an image are first convolved and then translated, or first translated and then convolved, the resulting features are the same. Translation equivariance enhances the model's generalization capabilities.

However, CNNs are not without imperfections. The spatial features they extract are inherently local in nature, which to a certain extent constrains the model's performance, whereas transformers can obtain global information through their self-attention mechanism.

MobileViT is a lightweight network that combines the strengths of both CNN and ViT. MobileViT is available in three versions depending on model size: MobileViT-S, MobileViT-XS, and MobileViT-XXS. MobileViT primarily consists of standard convolutions, inverted residual blocks from MobileNetV2 (MV2), MobileViT blocks, global pooling, and fully connected layers. The network architecture is illustrated in Figure 1. In this paper, we have replaced the MobileViT block in the MobileViT-XS network with our proposed DeformViT block. Additionally, we have removed the Conv(1×1), global pooling, and fully connected

layers at the bottom of MobileViT-XS to create the FireViT network, which serves as the network backbone in our fire detection approach.

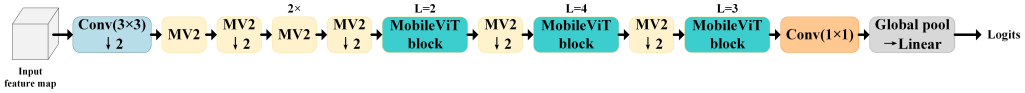

Figure 1. Structure of the MobileViT network; "(3×3)" and "(1×1)" denote the size of the convolution kernel, "MV2" denotes the inverted residual block in MobileNetV2, "↓2" denotes the downsampling operation, and "L" denotes the number of layers in the transformer block.

2.2. Prediction Head

Target detection is the ability to accurately localize objects of interest in an image; the prediction head (prediction part or output part) is required to perform classification and regression tasks. There are two types of detection heads, namely, coupled detection heads and decoupled detection heads. A coupled detection head means that the classification and regression tasks share a significant portion of their input parameters. In [19], the authors pointed out that the features of interest for classification and regression tasks during the learning of input parameters are different. These two subtasks are coupled, leading to spatial misalignment issues that can significantly impact network convergence speed. Decoupled detection heads address this issue. A decoupled detection head separately processes the input parameters for the classification and regression tasks, which enhances the detection accuracy and convergence speed of the detection network. The YOLOX [20] network uses decoupled heads for target detection, showing a 1.1% improvement in terms of mAP compared to YOLOv3 [21]. As a result, most target detection networks including PPYOLO-E [22], YOLOv6 [23], and YOLOv8 [24], have begun to adopt this paradigm. To ensure the fairness of our designed adaptive lightweight FireViT backbone feature extraction network in comparison with other backbone network models for fire target detection, all the models studied in this paper use three YOLOv8 decoupled heads at the bottom of their respective networks for fire target detection.

The prediction head of YOLOv8 uses decoupled classification and regression branches; the detailed structure is shown in Figure 2. The classification branch uses the Binary Cross-Entropy (BCE) loss. The regression branch employs the Complete Intersection over Union (CIoU) loss [25], and utilizes a novel loss function called the Distribution Focal Loss (DFL) [26]. The DFL is designed to optimize the probabilities of the left (y_i) and right (y_{i+1}) positions that are closest to the label y in a manner similar to cross-entropy. This allows the network to quickly focus on the distribution in the vicinity of the target location. The formula is represented as Equation (1):

$$DFL(S_i, S_{i+1}) = -((y_{i+1} - y)\log(S_i) + (y - y_i)\log(S_{i+1})). \quad (1)$$

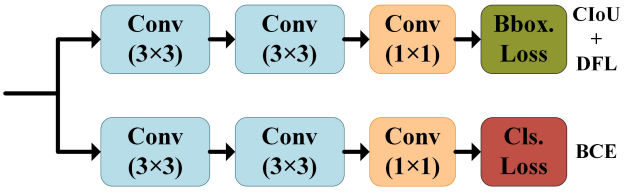

Figure 2. Structure of the prediction head of YOLOv8; CIOU is the complete intersection over union loss, DFL is the distribution focal loss, and BCE is the binary cross-entropy loss.

3. Methods

In this section, we discuss an adaptive lightweight backbone network called FireViT, which is an improvement based on the MobileViT-XS network. FireViT is designed for the detection of smoke and flame targets in fire scenarios; the overall structure is illustrated in Figure 3. The Deformable Vision Transformer (DeformViT) block is an adaptive lightweight

CNN combined with a transformer module, which plays a major role in FireViT for feature extraction. The DeformViT block is able to better extract the irregularly varying smoke and flame features in a fire situation, and can capture the flame and smoke in a fire situation locally and holistically better than the traditional convolutional module, thereby improving the accuracy of fire detection. The improved adaptive activation function proposed in our model, called AdaptGELU, can increase the model's nonlinear expressive power and further enhance the accuracy of fire detection.

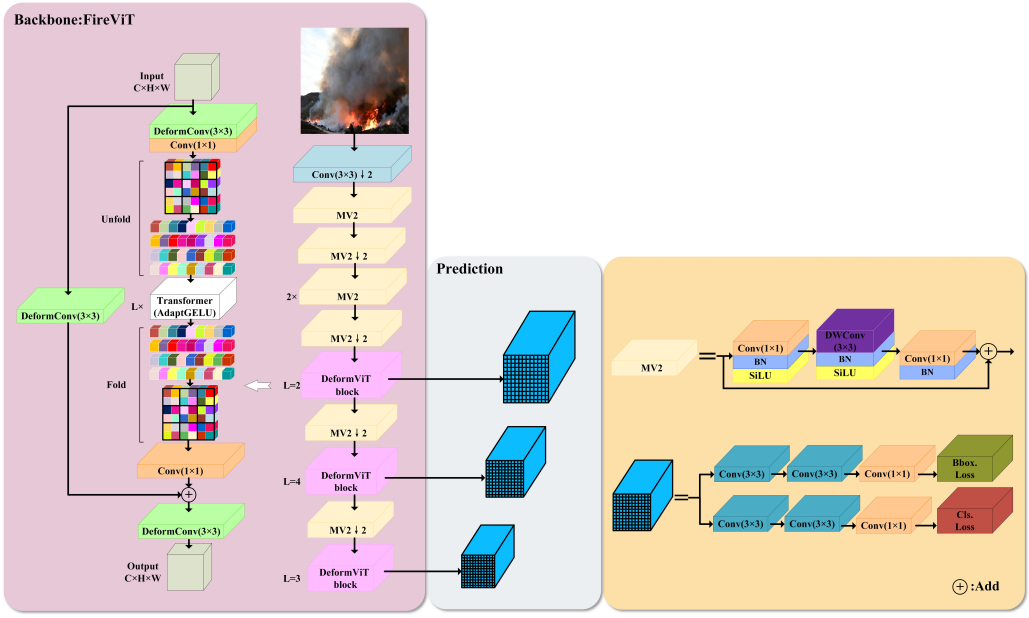

Figure 3. Detailed structure of the fire detection network using FireViT as the backbone network; *C* represents the number of channels of the feature map, *H* represents the height of the feature map, *W* represents the width of the feature map, DWConv stands for depthwise separable convolution, DeformConv stands for Deformable Convolution, (1×1) and (3×3) are the convolutional kernel sizes, BN stands for batch normalization, "↓2" indicates the downsampling operation, and "L" stands for the number of layers in the transformer block.

3.1. Adaptive Lightweight Backbone Network Module: DeformViT Block

The DeformViT block aims to combine the advantages of deformable convolutions for local feature extraction with those of transformers for global feature extraction and to perform feature extraction on the input tensor with fewer parameters. In [27], the authors proposed a Deformable Attention Transformer; however, the number of parameters in its minimal model reached 29 M. In addition to MobileViT, there are a number of network models [14,15] that attempt to combine the benefits of convolution and transformers; however, these use fixed-shaped convolutional modules, limiting feature extraction to the irregularly varying smoke and flames of a fire. Our proposed DeformViT block uses deformable convolution in combination with a transformer to enable better and more fine-grained extraction of the ever-changing features of flame and smoke in fire scenes.

Deformable convolution for capturing better local features. It is obvious that the standard fixed-size convolution kernel is not well suited for this task of adapting to irregular smoke and flames at a fire scene; thus, we use deformable convolution, which adapts the structure of the convolution kernel by learning the offsets, as shown in Figure 4.

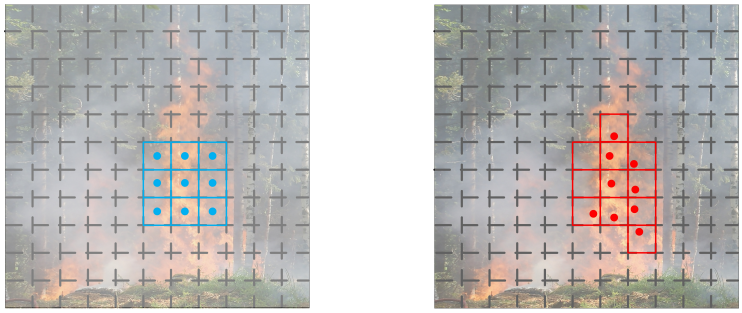

Figure 4. Illustration of sampling for the 3 × 3 standard fixed-size convolution kernel and the deformable convolution kernel. The (**left**) side of Figure 4 shows the sampling points and grid for the 3 × 3 fixed convolution kernel (blue dots and boxes), while the (**right**) side of Figure 4 shows the sampling points and grid for the deformable convolution with positional offset (red dots and boxes).

A standard fixed-size convolution can be divided into the following two-step operation.

Step 1: Sample the pixel point gridding from the input feature map $X \in \mathbb{R}^{C \times H \times W}$ (where C is the number of feature map channels, H is the feature map height, and W is the feature map width). Assuming that a 2D $K \times K$ convolution is used for sampling, the position of the pixel point in the sensory field can be denoted as P_n, as shown in Equation (2):

$$P_n = \{P_{ij}\} \quad (0 \leqslant i, j \leqslant K-1; 0 \leqslant n \leqslant K^2-1; i, j, n \in N). \tag{2}$$

Step 2: Output each position P_{out} on the feature map Y after the convolution operation, as shown in Equation (3):

$$Y(P_{out}) = \sum_{n=0}^{P_n} \omega(P_n) \cdot x(P_{out} + P_n), \tag{3}$$

where $\omega(\cdot)$ denotes the value learned by the network in the convolution and $x(\cdot)$ denotes the value after grid sampling on the input feature map.

The essence of deformable convolution lies in the modification of the sampling results to achieve a variation in the convolutional effect, as illustrated in Figure 5. The right side of Figure 5 shows the value of the convolution kernel offset predicted by the convolution. In deformable convolution, ΔP_n is used to offset the position of the point P_n on the feature map, the weight coefficients $\Delta\gamma$ are added at each sampling point to reduce the interference of irrelevant information on the feature map; at this point, deformable convolution is computed as in Equation (4):

$$Y(P_{out}) = \sum_{n=0}^{P_n} \omega(P_n) \cdot x(P_{out} + P_n + \Delta P_n) \cdot \Delta\gamma_n. \tag{4}$$

The offset value predicted by the convolution operation is often a small number that cannot be sampled directly from the feature map X. Here, bilinear interpolation (Equation (5)) is used to ensure that the feature map can be sampled after the offset:

$$x(P) = \sum_{P_s} B(P_s, P) \cdot x(P_s), \tag{5}$$

where P is the position after offset; $P = P_{out} + P_n + \Delta P_n$; P_s is the site on the space of all integrals of the feature map X; and $B(\cdot, \cdot)$ is a 2D bilinear interpolation kernel, as shown in Equation (6):

$$B(P_s, P) = b(P_{sx}, P_x) \cdot b(P_{sy}, P_y), \tag{6}$$

where $b(e, f) = \max(0, 1 - |e - f|)$.

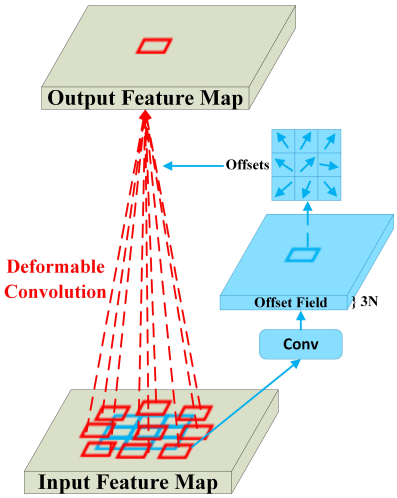

Figure 5. Diagram of the 3 × 3 deformable convolution structure; N is the number of pixels in the convolution kernel, and the figure shows a 3 × 3 convolution with N = 9.

Unfold–Transformer–Fold operation focusing on global features. The emergence of DETR [28] has opened the door to the use of transformers for computer vision target detection. After the emergence of ViT, transformers were identified as a neural network architecture that performs very well in the field of computer vision. While ViT requires attention for each token (high computational cost), our network uses deformable convolution prior to the transformer block to better grasp the local features; in this way, the computational cost can be reduced by dividing the feature map into multiple patches during global modeling of the feature map $X \in \mathbb{R}^{C \times H \times W}$ (where C is the number of feature map channels, H is the feature map height, and W is the feature map width), then self-attention can be performed for the pixels in the same position in each patch. We call this the Unfold–Transformer–Fold operation. The patch has dimensions of (h, w) (ignoring the number of channels C), where h is the height of the patch and w is its width. The unfold operation spreads the pixels at the same position in each patch in a sequence, then the attention of each sequence is calculated in parallel by the transformer and finally collapsed back to the size of the original feature map by the fold operation, as shown in Figure 6.

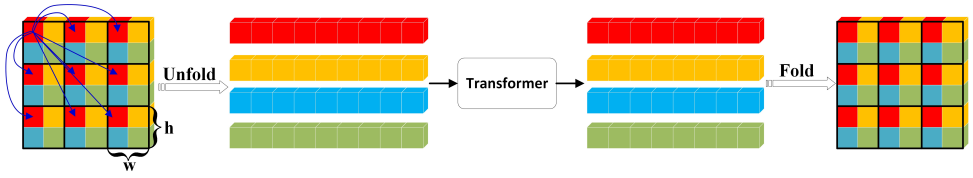

Figure 6. Schematic diagram of the Unfold–Transformer–Fold operation. The dimension (h, w) of each patch in the figure is $(2, 2)$, i.e., each patch consists of four pixels (shown in the figure in red, yellow, blue, and green). The token (pixel) in each patch only calculates attention with its own token of the same position (the color block of the same color in the figure), as indicated by the dark blue arrow. The feature map X dimensions are (C, H, W) and the cost according to self-attention alone is $O(CHW)$. According to the Unfold–Transformer–Fold calculation, at this time, $Patch = 2 \times 2 = 4$ and the calculation cost is $O(\frac{CHW}{4})$, amounting to $\frac{1}{4}$ of the original calculation cost. The unfold and fold operations reshape the data to satisfy the self-attention computation.

In the fire occurrence scenario, we use deformable convolution to better capture the local features and the Unfold–Transformer–Fold operation to capture the global features; we propose an adaptive lightweighting module called DeformViT that combines them to improve the accuracy of fire detection. We conducted a comparative experiment on the eight forms of DeformViT modules designed in Section 4.4, finally choosing scheme VIII, which is shown in Figure 7.

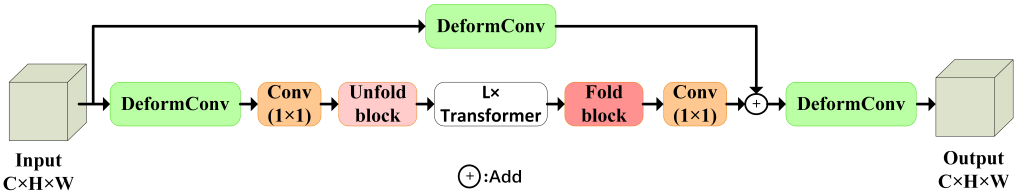

Figure 7. Structure of the DeformViT module, where L denotes the number of layers in the transformer block, C denotes the number of channels in the feature map, H denotes the height of the feature map, and W denotes its width.

3.2. Adaptive Activation Function: Adaptive GELU (AdaptGELU)

Activation functions are of paramount importance in deep learning, as they enhance the neural network's capacity for nonlinear expression. Gaussian Error Linear Units (GELUs) [29] are beginning to attract attention in applications such as Google's BERT [30] and OpenAI's GPT-2 [31]. A graphical representation of the GELU function is provided in Figure 8, while its mathematical expression is provided in Equation (7):

$$GELU(G) = x \cdot PR(G \leqslant x) = x \cdot \Phi(x), x(0,1), \tag{7}$$

where x is the input, G is a Gaussian random variable with zero mean and unit variance, $PR(G \leqslant x)$ is the probability of G being less than or equal to a given value of x, and $\Phi(x) = \frac{1}{2}\left[1 + erf\left(\frac{x}{\sqrt{2}}\right)\right]$ is the cumulative distribution function of the standard normal distribution.

The approximate solution of Equation (7) is calculated as shown in Equation (8):

$$GELU(G) = 0.5x\left(1 + \tanh\left[\sqrt{\frac{\pi}{2}}\left(x + 0.044715x^3\right)\right]\right). \tag{8}$$

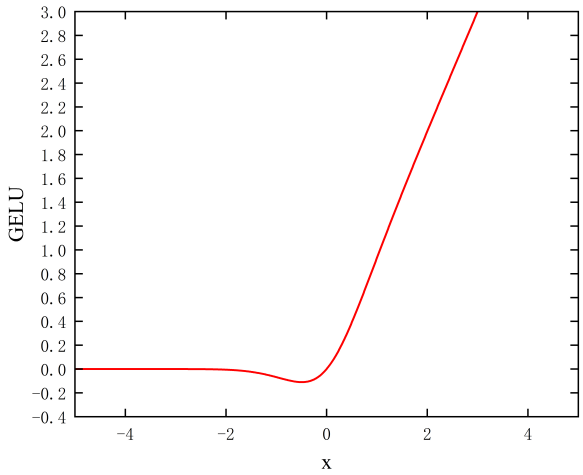

Figure 8. GELU activation function.

GELU is smoother compared to ReLU [32], and is better at mitigating vanishing gradients and supporting network training and optimization in deep neural network models. However, as GELU is not optimal in certain cases, we propose Adaptive GELU

(AdaptGELU) by introducing a trainable parameter a (with the initial value of a set to 1.0) to further improve the performance of the network. AdaptGELU replaces the original activation function in the feedforward network in the transformer. We propose two versions of AdaptGELU, which we respectively name AdaptGELUv1 and AdaptGELUv2. The morphology of the two versions of AdaptGELU with different values of a is shown in Figure 9.

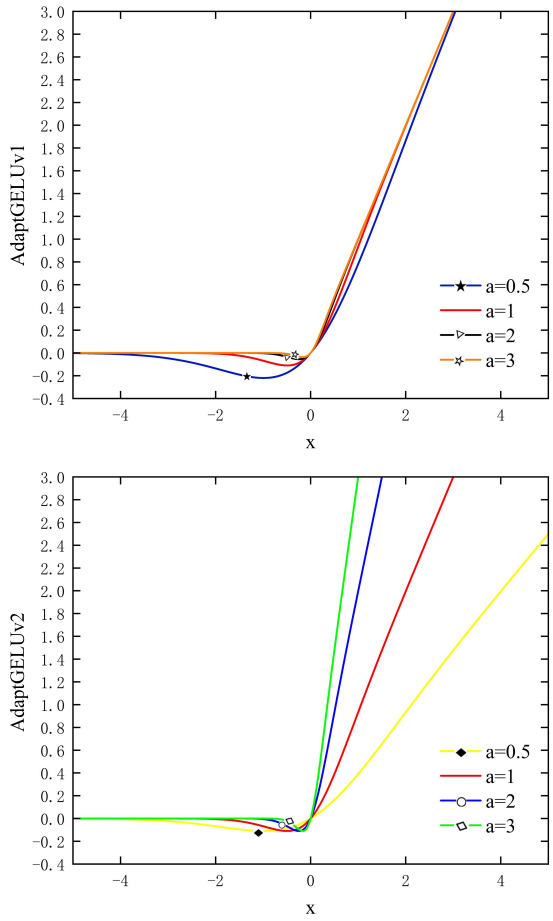

Figure 9. Morphological maps of the two versions of AdaptGELU with different values for the a parameter.

AdaptGELUv1 is represented by Equation (9):

$$AdaptGELUv1(G) = \frac{1}{2}x\left[1 + erf\left(\frac{ax}{\sqrt{2}}\right)\right]. \tag{9}$$

AdaptGELUv2 is represented by Equation (10):

$$AdapeGELUv2(G) = \frac{1}{2}ax\left[1 + erf\left(\frac{ax}{\sqrt{2}}\right)\right]. \tag{10}$$

Section 4.4 describes our comparative fire detection experiments, where we replaced the original SiLU [33] activation function in the transformer's feedforward network with Sigmoid, ReLU, GELU, AdaptGELUv1, and AdaptGELUv2 functions. Our comparative analysis of reveals that the network with AdaptGELUv2 achieves better accuracy; thus, we ultimately selected AdaptGELUv2 as the activation function for our model in the feedforward network of the transformer. Detailed experimental results and analysis are presented in Section 4.4.

4. Experiments and Results

This section first describes the labeled fire dataset we collected and produced, along with details of the implementation and evaluation metrics. Then, we describe the validation of our proposed model components through a series of comparative experiments. Finally, the effectiveness of our model is further demonstrated by comparing the experimental results and visualizations of different lightweight backbone networks used in fire detection experiments on the fire natural light and the fire infrared datasets.

4.1. Dataset

The FireViT backbone network we designed is used in this paper to study supervised fire detection. The overall fire detection model using FireViT as the backbone network is able to identify and localize smoke and flame in a fire. Therefore, preparing labeled public datasets of fires or constructing a self-made labeled fire dataset is an essential step. Due to the severe shortage of publicly available labeled fire datasets at present, we constructed a self-made labeled fire dataset by extensively reviewing the relevant literature and collecting fire-related data from various sources. The labeled fire dataset we collected and constructed is currently the richest labeled fire dataset containing fire occurrence scenarios, as well as the dataset containing the largest number of fire images. The fire dataset we collected and constructed is divided into a fire natural light dataset and fire infrared dataset.

The data in the fire natural light dataset come from: (A) the fire image and PNG still image dataset from the 2018 study of Dunnings and Breckton [34]; (B) VisiFire [35]; (C) the KMU fire and smoke database [36]; (D) video smoke detection [37]; (E) the flame dataset: aerial imagery pile burn detection using drones dataset [38]; and (F) the fire public welfare web platform. All of these data were unlabeled for fires, as shown in the top half of Figure 10, which depicts a partial demonstration of the collected fire natural light data. These data consist of both image format and video format data; as our focusing here is on fire detection from images, the collected video data were processed into images by exporting frames at 0.05 s intervals. In order to improve the robustness of the fire detection models and further enrich the fire data, we randomly selected three fire data sources from A, B, C, D, E, and F. Next, we randomly decided whether or not to carry out the corresponding operation with 50% probability through Rotate–Flip–Affine transformation (as shown in Figure 11) one by one for each of the three fire images in order to expand the dataset.

The data in the fire infrared dataset mainly consist of data captured by the infrared thermal imager in E and simulated fire data captured by an infrared structured light depth camera. The first three rows of the fire infrared data section shown in Figure 10, from top to bottom, show portions of the Fusion, GreenHot, and WhiteHot data captured by an infrared thermal imager, while the last row shows portions of the simulated fire data captured by an infrared structured light depth camera. We similarly first cut the frames of the captured video data at 0.05 s intervals in order to process them into images. Next, we randomly selected two fire data sources from Fusion, GreenHot, WhiteHot, and infrared structured light data, and finally selected each fire image from these two fire data sources one by one in all the modes of operation of the Rotate–Flip–Affine transformation with 50% probability to perform the corresponding operation to expand the dataset.

All of the above data were labeled using the LabelImage data labeling tool. We constructed datasets containing both fire natural light data (121,339 images) and fire infrared data (96,112 images) to fulfill different application requirements. Detailed information is presented in Table 1. The dataset was divided into training, validation, and test sets in a ratio of 8:1:1 to constructed the final labeled fire dataset in VOC data format. The fire natural light dataset contains the labels “fire” and “smoke”, while the fire infrared dataset uses only the “fire” label, as smoke is much harder to capture at night. All of the fire natural light data were used in the experiments, while only the Fusion data were used in the fire infrared data (all of the following are expressed as fire infrared data).

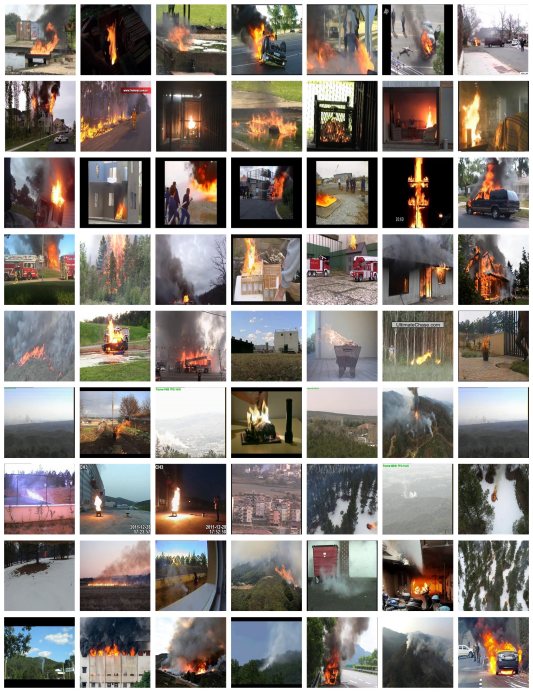

Figure 10. Partial presentation of the collected fire data.

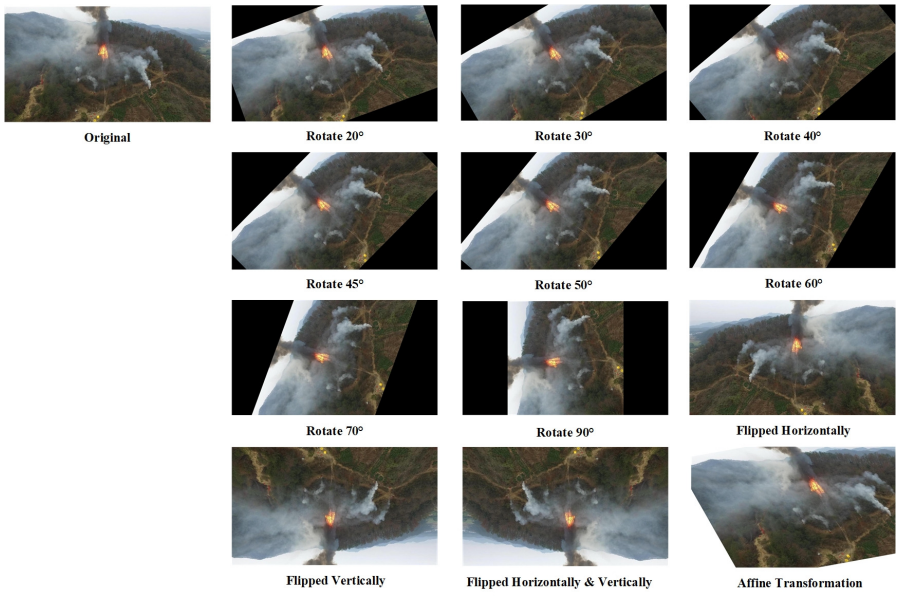

Figure 11. Rotate–Flip–Affine transformation.

Table 1. Fire dataset details: (a) fire natural light dataset information and (b) fire infrared dataset information. Here, "√" indicates that the Rotate-Flip-Affine transformation was used, while "×" indicates that it was not used. Data sources: (A) PNG still image fire image dataset from Dunnings and Breckton, 2018; (B) VisiFire; (C) KMU fire and smoke database; (D) video smoke detection; (E) the flame dataset: aerial imagery pile burn detection using drones; and (F) the fire public welfare web platform.

(a) Fire natural light dataset				
Data Sources	Rotate-Flip-Affine Transformation	Total Number	Number of "smoke" labels	Number of "fire" labels
A	×	10,048	27,221	19,103
B	√	54,968	61,324	20,297
C	√	29,222	25,345	17,711
D	√	6488	12,124	5453
E	×	12,201	23,615	72,660
F	×	8412	6064	34,159
(b) Fire infrared dataset				
Infrared data	Rotate-Flip-Affine Transformation	Total Number	Number of "fire" labels	
Fusion	√	75,219	214,005	
GreenHot	×	5011	26,386	
WhiteHot	×	5891	22,590	
Structured Infrared Light	√	9991	10,198	

4.2. Implementation Details

All models were trained and experimented with on a system running Ubuntu 20.04, Python 3.8, CUDA 11.3, PyTorch1.11.0, and an NVIDIA RTX4090 GPU. The models were trained with fixed random number seeds and without pretraining weights. The parameter settings used for model training are shown in Table 2.

Table 2. Parameter settings during model training.

Training Parameter Settings	Particulars
Initialization	MSRA initialization [39]
Input image dimensions	(640, 640, 3)
Optimizer	SGD
Momentum	0.937
Initial learning rate	0.01
Weight Decay	0.0005
Number of images per batch	8
Epochs	50

4.3. Evaluation Metrics

To evaluate the effectiveness of FireViT as a backbone network for fire detection, we used the following metrics: Precision, Recall, Parameters, Average Precision (AP), mAP, and Floating Point Operations (FLOPs). AP is the area under the Precision–Recall (PR) curve, where we used a value of 0.5 for the Intersection over Union (IoU = 0.5). mAP is the mean of AP calculated for each individual class. Higher AP and mAP values indicate better performance.

Precision is shown in Equation (11):

$$Precision = \frac{TP}{TP + FP}. \tag{11}$$

Recall is shown in Equation (12):

$$Recall = \frac{TP}{TP + FN}. \tag{12}$$

Above, TP (True Positive) represents the number of correctly identified positive samples, FP (False Positive) represents the number of samples that are actually negative and predicted as positive, and FN (False Negative) represents the number of samples that are actually positive and predicted as negative.

FLOPs are used to measure the complexity of the algorithm, as shown in Equation (13):

$$FLOPs = (2 \times C_{in} \times K^2 - 1) \times H \times W \times C_{out}, \tag{13}$$

where C_{in} represents the number of input channels, C_{out} represents the number of output channels, K represents the size of the convolutional kernel, and H and W respectively denote the height and width of the feature map.

4.4. Ablation Experiments on the Fire Natural Light Dataset

We designed eight build scenarios for the DeformViT module, as shown in Figure 12. We replaced the DeformViT block with the MobileViT block in the MobileViT network and used the SiLU activation function in the network. In addition, we replaced the MobileViT block in the MobileViT network with the DeformViT block and used the SiLU activation function in the network, at which point, forming a backbone network that we name FireViT-SiLU. Fire detection was performed on the bottom layer of the FireViT-SiLU network using the three decoupled detection heads mentioned in Section 2.2 The results of the ablation experiments are shown in Table 3.

Table 3. Comparative experimental results for fire detection using the FireViT-SiLU backbone feature extraction network composed of DeformViT blocks with different architectural configurations.

Options	mAP	Params	GFLOPs
MobileViT block	90.9%	1.9M	13.8
I	91.7%	2.1M	12.5
II	91.6%	1.8M	12.2
III	91.5%	1.8M	12.2
IV	91.1%	1.6M	11.9
V	91.2%	1.6M	11.9
VI	91.5%	1.8M	12.2
VII	91.5%	1.8M	12.2
VIII	91.8%	2.1M	12.5

From Figure 12 and Table 3, it can be seen that the fire detection network formed by our proposed building scheme on the Type I DeformViT block has an overall advantage over the fire detection network with MobileViT as the backbone network; the number of parameters improves by only 0.2 M, GFLOPs decrease by 1.3, and mAP improves by 0.8%. The comparison of the fire detection backbone network scheme consisting of the Type II DeformViT block to the construction scheme of the Type VII DeformViT block provided us with further ideas. Therefore, we replaced the copy operation in the construction scheme of the type I DeformViT block with deformable convolution to form the type VIII construction scheme of the DeformViT block. The mAP of the fire detection network formed using the type VIII building scheme for the DeformViT block was further improved with respect to the fire detection network based on the type I building scheme of the DeformViT block. Based on these results, we selected the construction scheme using the type VIII DeformViT block as the final build scheme in our network.

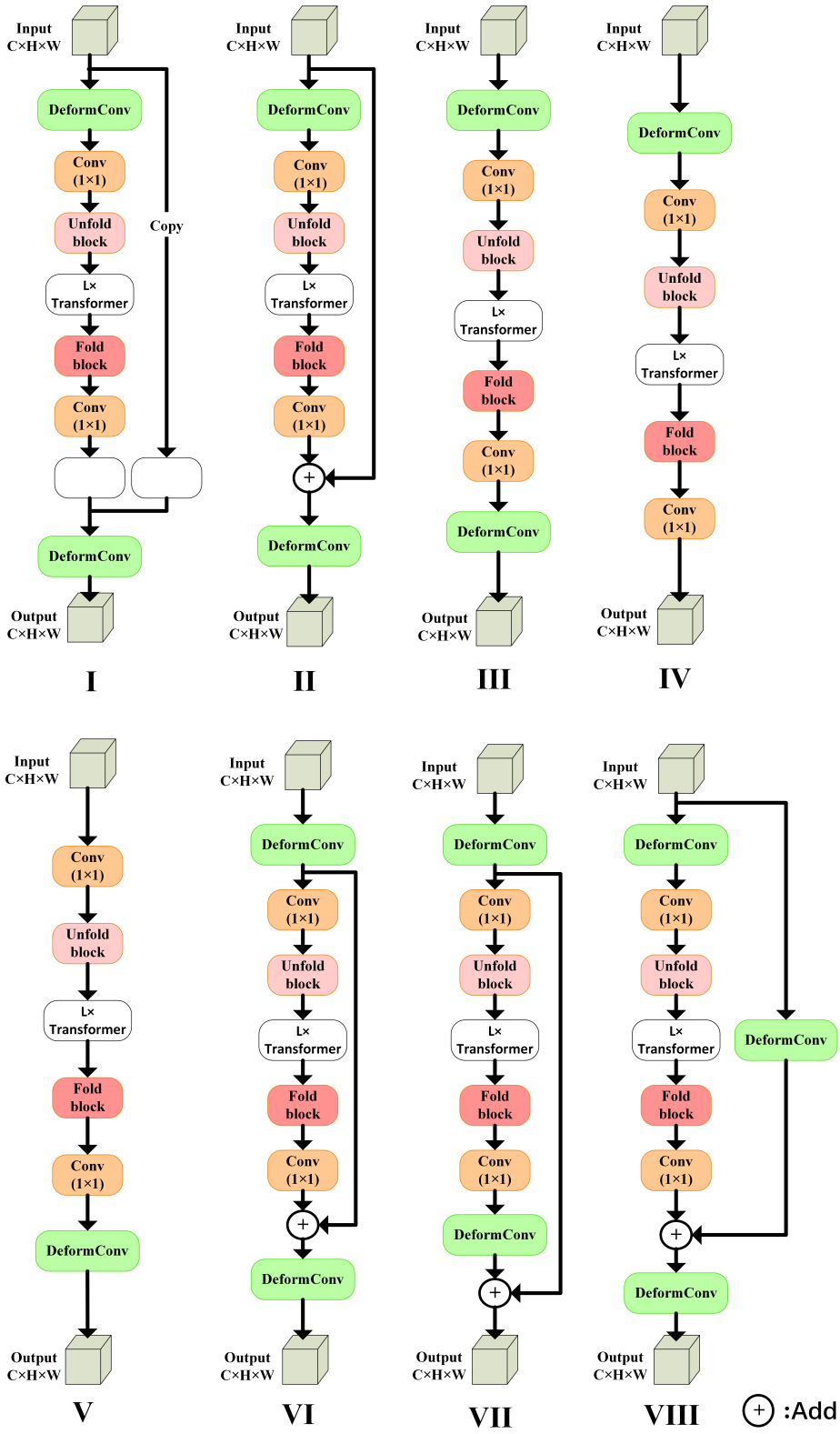

Figure 12. Different architectural configurations used for the DeformViT block.

FireViT-SiLU uses the SiLU global activation function. In this section, we used the Sigmoid, ReLU, GELU, and our improved adaptive AdaptGELUv1 and AdaptGELUv2 activation functions to replace the SiLU activation function in the feedforward network of FireViT-SiLU's transformer block. The three decoupled detection heads mentioned in Section 2.2 were used for fire detection at the bottom layer of the overall network. The

comparative results of the detection experiments are shown in Table 4. Because only the activation function in the transformer's feedforward network was replaced, the amount of variation in the number of parameters and GFLOPs in the overall network is almost negligible.

Table 4. Comparative experimental results of different activation functions replacing the SiLU activation function of the transformer's feedforward network in FireViT-SiLU for fire detection. Because only the activation function in the transformer's feedforward network was replaced, the amount of variation in the number of parameters and GFLOPs in the overall network is almost negligible.

Model	mAP	Params	GFLOPs
FireViT-SiLU	91.82%	2.1 M	12.5
FireViT-Sigmoid	91.38%	2.1 M	12.5
FireViT-ReLU	91.89%	2.1 M	12.5
FireViT-GELU	91.88%	2.1 M	12.5
FireViT-AdaptGELUv1	92.09%	2.1 M	12.5
FireViT-AdaptGELUv2	92.14%	2.1 M	12.5

From Table 4, it can be seen that our proposed improved GELU activation functions, AdaptGELUv1 and AdaptGELUv2, have advantages over other activation functions in the fire detection context of this paper. When FireViT-SiLU, FireViT-Sigmoid, FireViT-ReLU, and FireViT-GELU were applied to fire detection, FireViT-ReLU achieved the highest mAP at 91.89%, FireViT-AdaptGELUv1 a 0.2% higher mAP than FireViT-ReLU, and FireViT-AdaptGELUv2 a 0.25% higher mAP than FireViT-ReLU. After comparing the experimental results, we chose AdaptGELUv2 as the activation function for use in the transformer's feedforward networkm, and used FireViT-AdaptGELUv2 as the backbone network for the final FireViT fire detection model.

We conducted fire detection comparison experiments by comparing our proposed approach with several mainstream lightweight convolutional backbone network algorithms: GhostNetV2 [40], PP-LCNet [41], ShuffleNetV2 [42], MobileNetV3 [43], and EfficientNet [44]. The results of our experiments are shown in Table 5. The best fire detection among GhostNetV2, PP-LCNet, ShuffleNetV2, MobileNetV3, and EfficientNet as backbone networks was PP-LCNet, with an mAP of 90.25%. Although the GFLOPs of our proposed FireViT backbone network for fire target detection were 1.7 higher than PP-LCNet, FireViT had a 1.85% higher mAP and 0.9 M fewer parameters than PP-LCNet. Our proposed FireViT network backbone achieves a good balance between model complexity and detection accuracy for fire target detection.

Table 5. Experimental results comparing FireViT to mainstream lightweight convolutional backbone networks for fire target detection on our fire natural light dataset.

Model	AP_{fire}	AP_{smoke}	mAP	Params	GFLOPs
GhostNetV2	89.2%	91.0%	90.1%	3.8 M	6.3
PP-LCNet	89.6%	90.9%	90.25%	3.0 M	10.8
ShuffleNetV2	89.4%	90.1%	89.75%	3.0 M	10.7
MobileNetV3	89.0%	90.2%	89.6%	3.1 M	5.5
EfficientNet	89.0%	91.0%	90.0%	3.4 M	8.4
FireViT(ours)	91.3%	92.9%	92.1%	2.1 M	12.5

We next analyzed and compared FireViT with mainstream lightweight backbone network algorithms based on the transformer architecture: EfficientViT-M0 [14], SwinTransformer [45] (the model underwent depth compression of 0.33 and width compression of 0.25), EfficientFormerV2-S0 [15], and MobileViT-XS [16]. The results of these experiments are shown in Table 6. The results show that our proposed FireViT network backbone for fire target detection can improve detection accuracy while reducing the computational complexity of the model. In the case where the models have roughly the same number of

parameters, MobileViT-XS has the highest mAP among EfficientViT-M0, EfficientFormerV2-S0, SwinTransformer and MobileViT-XS when used as the backbone network for fire target detection. Our proposed FireViT network backbone has a 1.2% higher mAP and 1.3 fewer GFLOPs relative to MobileViT-XS.

In order to obtain a more intuitive understanding of the features learned by various networks for detecting smoke and flames in fire incidents, we conducted heatmap visualization for the following backbone networks: GhostNetV2, PP-LCNet, ShuffleNetV2, MobileNetV3, EfficientNet, EfficientViT-M0, SwinTransformer (the model underwent depth compression of 0.33 and width compression of 0.25), EfficientFormerV2-S0, MobileViT-XS, and our proposed FireViT. The visualization results are depicted in Figure 13. From the heatmap visualization results of each network model, it is clear that our proposed FireViT is better able to capture the characteristics of smoke and flame in fire detection scenarios.

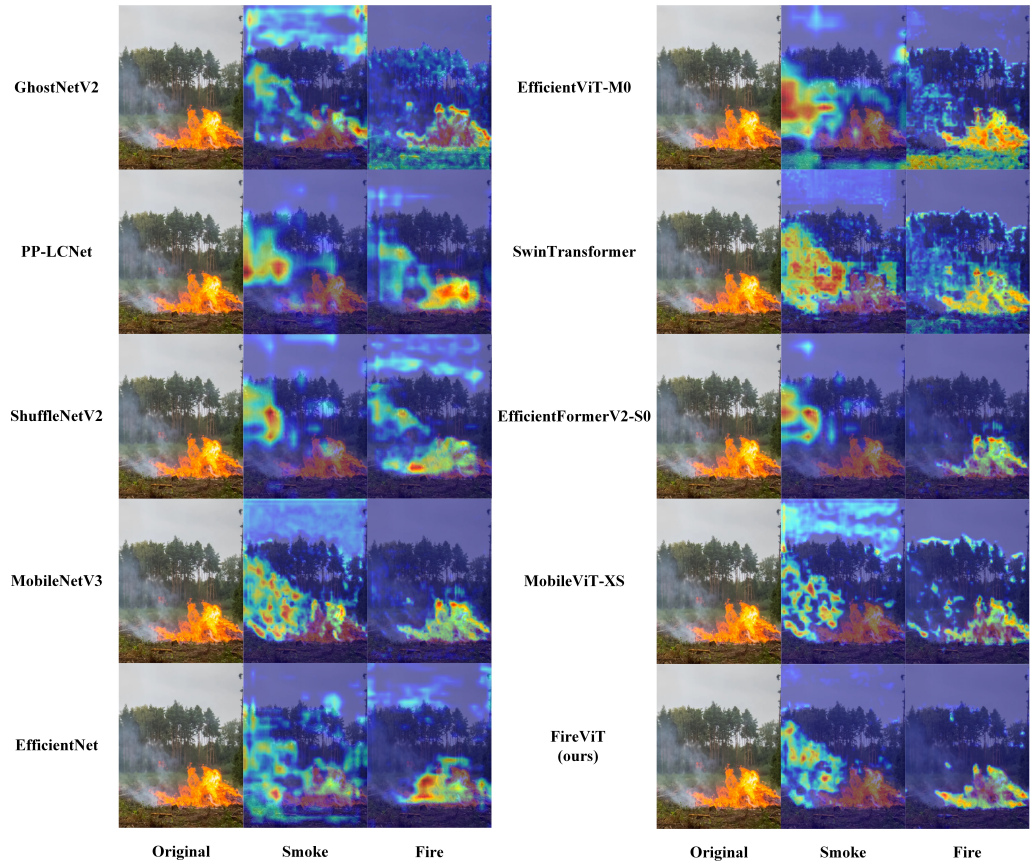

Figure 13. Heat map visualization results of each lightweight backbone network model for fire target detection.

The fire detection performance of FireViT as the network backbone is illustrated in Figure 14. From the first detection picture from top to bottom in the second column and the third picture from top to bottom in the fourth column of Figure 14, it can be seen that FireViT is able to achieve good detection and recognition performance even for smaller fire targets. In the third column of Figure 14, from top to bottom, the first fire picture contains an obvious "smoke" target, an obvious "fire" target, and another fuzzy "fire" target; FireViT can easily detect the two obvious targets, and is able to detect the fuzzy "fire" target as well. Overall, it can be seen from the figure that FireViT used as a network backbone for fire target detection is well suited for detection tasks in fire scenarios.

Table 6. Experimental results of comparison between FireViT and mainstream lightweight backbone networks based on the transformer architecture for fire target detection on our fire natural light dataset; the SwinTransformer model underwent depth and width compression, with 0.33 for depth and 0.25 for width.

Model	AP_{fire}	AP_{smoke}	mAP	Params	GFLOPs
EfficientViT-M0	89.0%	90.7%	89.85%	2.8 M	7.4
SwinTransformer	88.9%	90.1%	89.5%	2.2 M	6.9
EfficientFormerV2-S0	89.4%	91.2%	90.3%	3.8 M	9.0
MobileViT-XS	91.0%	90.8%	90.9%	1.9 M	13.8
FireViT(ours)	91.3%	92.9%	92.1%	2.1 M	12.5

Figure 14. Detection results of FireViT used as the backbone network for fire detection.

4.5. Comparison Experiments on Fire Infrared Datasets

Our previous fire detection comparison experiments used the fire natural light dataset. In order to further validate the generalization performance of FireViT, we used GhostNetV2, PP-LCNet, ShuffleNetV2, MobileNetV3, EfficientNet, EfficientViT-M0, SwinTransformer (the model underwent depth compression of 0.33 and width compression of 0.25), EfficientFormerV2-S0, MobileViT-XS, and FireViT as network backbones for fire detection on our fire infrared dataset. The results of these comparison experiments are shown in Table 7.

On the fire infrared dataset, other than our proposed FireViT lightweight network backbone model, MobileViT-XS was the most effective for fire detection, with its fire detection accuracy reaching 94.3%; PP-LCNet was second to MobileViT-XS, with a detection accuracy of 94.1%. Our proposed FireViT model for fire detection achieved 0.8% better detection accuracy on the infrared dataset than MobileViT-XS and 1% better detection accuracy than PP-LCNet. Overall, our proposed FireViT showed good generalization performance on the infrared fire dataset when used as the network backbone for fire detection.

Table 7. Comparison results of FireViT and mainstream lightweight backbone networks for fire detection on our fire infrared dataset.

Model	AP_{fire}	Params	GFLOPs
GhostNetV2-Infrared	93.7%	3.8 M	6.3
PP-LCNet-Infrared	94.1%	3.0 M	10.8
ShuffleNetV2-Infrared	94.0%	3.0 M	10.7
MobileNetV3-Infrared	93.5%	3.1 M	5.5
EfficientNet-Infrared	93.9%	3.4 M	8.4
EfficientViT-M0-Infrared	93.8%	2.8 M	7.4
SwinTransformer-Infrared	93.3%	2.2 M	6.9
EfficientFormerV2-S0-Infrared	93.9%	3.8 M	9.0
MobileViT-XS-Infrared	94.3%	1.9 M	13.8
FireViT-Infrared (ours)	95.1%	2.1 M	12.5

5. Conclusions

In this study, we have proposed an adaptive lightweight network backbone that can be used for fire detection, along with presentation of an improved adaptive activation function and a collection of labeled fire datasets containing the richest fire scenarios and the largest number of fire images built to date. First, in order to address the relatively small number of publicly available labeled fire datasets, as part of this research we collected and established a fire dataset that contains the richest fire scenes and the largest number of fire images currently available, for which we used the Rotate–Flip–Affine transformation operation. Our full fire dataset consists of a fire natural light dataset and fire infrared dataset, thereby meeting different application requirements. Second, in order to solve the problem of insufficient extraction of smoke and flame features that change irregularly in the fire scene, we propose the DeformViT block, a lightweight module that combines deformable convolution and a transformer to better grasp the features of smoke and flame in fire scenes both locally and holistically. Finally, we propose an improved adaptive activation function to further enhance the detection accuracy and nonlinear representation of the network. Our experimental results indicate that the FireViT adaptive lightweight network backbone proposed in this paper has high accuracy in fire detection scenarios. When used as the network backbone for fire detection, FireViT achieved an mAP of 92.1% on the fire natural light dataset and 95.1% on the fire infrared dataset with a model computational complexity of 12.5 GFLOPs. Based on these results, FireViT has important application value for fire warning, and can provide an effective solution for early warning in intelligent firefighting.

Author Contributions: Conceptualization, P.S., N.S. and K.H.; methodology, N.S., K.H. and X.Y.; software, P.S., N.S. and K.H.; validation, P.S. and N.S.; formal analysis, P.S., N.S. and P.W.; investigation, P.S., N.S. and Q.X.; resources, N.S., X.Y. and P.S.; data curation, P.S., K.H. and C.W.; writing—original draft preparation, P.S., Q.X. and C.W.; writing—review and editing, P.S., X.Y. and P.W.; supervision, N.S. and P.W. All authors have read and agreed to the published version of the manuscript.

Funding: The research in this article was supported by the National Natural Science Foundation of China (No. 42275156 and 42205150), Jiangsu Natural Science Foundation (No. BK20210661), Jiangsu Postgraduate Innovation Project (No. SJCX23_0386), and Qing Lan Project of Jiangsu Province.

Data Availability Statement: The data and code used to support the findings of this study are available from the corresponding author upon request (001764@cwxu.edu.cn).

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

References

1. Rachman, F.Z.; Yanti, N.; Hadiyanto, H.; Suhaedi, S.; Hidayati, Q.; Widagda, M.E.P.; Saputra, B.A. Design of the early fire detection based fuzzy logic using multisensor. *Conf. Ser. Mater. Sci. Eng.* **2020**, *732*, 012039. [CrossRef]
2. Ye, H.; Xiaogang, W.; Shuchuan, G. Design and Evaluation Method of Wireless Fire Detection Node Based on Multi-Source Sensor Data Fusion. *Int. J. Sens. Sens. Netw.* **2021**, *9*, 19. [CrossRef]
3. Solórzano, A.; Eichmann, J.; Fernández, L.; Ziems, B.; Jiménez-Soto, J. M.; Marco, S.; Fonollosa, J. Early fire detection based on gas sensor arrays: Multivariate calibration and validation. *Sens. Actuators B Chem.* **2022**, *352*, 130961. [CrossRef]
4. Li, Y.; Yu, L.; Zheng, C.; Ma, Z.; Yang, S.; Song, F.; Tittel, F.K. Development and field deployment of a mid-infrared CO and CO_2 dual-gas sensor system for early fire detection and location. *Spectrochim. Acta Part A Mol. Biomol. Spectrosc.* **2022**, *270*, 120834. [CrossRef]
5. Liu, X.; Sun, B.; Xu, Z. D.; Liu, X.; Xu, D. An intelligent fire detection algorithm and sensor optimization strategy for utility tunnel fires. *J. Pipeline Syst. Eng. Pract.* **2022**, *13*, 04022009. [CrossRef]
6. Qiu, T.; Yan, Y.; Lu, G. An autoadaptive edge-detection algorithm for flame and fire image processing. *IEEE Trans. Instrum. Meas.* **2011**, *61*, 1486–1493. [CrossRef]
7. Ji-neng, O.; Le-ping, B.; Zhi-kai, Y.; Teng, W. An early flame identification method based on edge gradient feature. In Proceedings of the 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Xi'an, China, 25–27 May 2018; pp. 642–646. [CrossRef]
8. Khalil, A.; Rahman, S. U.; Alam, F.; Ahmad, I.; Khalil, I. Fire detection using multi color space and background modeling. *Fire Technol.* **2021**, *57*, 1221–1239. [CrossRef]
9. 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]
10. Chen, G.; Cheng, R.; Lin, X.; Jiao, W.; Bai, D.; Lin, H. LMDFS: A Lightweight Model for Detecting Forest Fire Smoke in UAV Images Based on YOLOv7. *Remote Sens.* **2023**, *15*, 3790. [CrossRef]
11. Dogan, S.; Barua, P.D.; Kutlu, H.; Baygin, M.; Fujita, H.; Tuncer, T.; Acharya, U.R. Automated accurate fire detection system using ensemble pretrained residual network. *Expert Syst. Appl.* **2022**, *203*, 117407. [CrossRef]
12. Li, A.; Zhao, Y.; Zheng, Z. Novel Recursive BiFPN Combining with Swin Transformer for Wildland Fire Smoke Detection. *Forests* **2022**, *13*, 2032. [CrossRef]
13. Huang, J.; Zhou, J.; Yang, H.; Liu, Y.; Liu, H. A Small-Target Forest Fire Smoke Detection Model Based on Deformable Transformer for End-to-End Object Detection. *Forests* **2023**, *14*, 162. [CrossRef]
14. Liu, X.; Peng, H.; Zheng, N.; Yang, Y.; Hu, H.; Yuan, Y. EfficientViT: Memory Efficient Vision Transformer with Cascaded Group Attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 14420–14430. [CrossRef]
15. Li, Y.; Hu, J.; Wen, Y.; Evangelidis, G.; Salahi, K.; Wang, Y.; Ren, J. Rethinking vision transformers for mobilenet size and speed. *arXiv* **2022**, arXiv:2212.08059. [CrossRef]
16. Mehta, S.;Rastegari, M. Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv* **2021**, arXiv:2110.02178. [CrossRef]
17. Wang, R.; Shivanna, R.; Cheng, D.; Jain, S.; Lin, D.; Hong, L.; Chi, E. Dcn v2: Improved deep and cross network and practical lessons for web-scale learning to rank systems. *arXiv* **2020**, arXiv:2008.13535.
18. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929. [CrossRef]
19. Zhuang, J.; Qin, Z.; Yu, H.; Chen, X. Task-Specific Context Decoupling for Object Detection. *arXiv* **2023**, arXiv:2303.01047. [CrossRef]
20. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.[CrossRef]
21. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767. [CrossRef]
22. Xu, S.; Wang, X.; Lv, W.; Chang, Q.; Cui, C.; Deng, K.; Lai, B. PP-YOLOE: An evolved version of YOLO. *arXiv* **2022**, arXiv:2203.16250. [CrossRef]
23. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Wei, X. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976. [CrossRef]
24. Ultralytics-YOLOv8. Available online: https://github.com/ultralytics/ultralytics (accessed on 26 June 2023).
25. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI conference on artificial intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12993–13000. [CrossRef]
26. Li, X.; Wang, W.; Wu, L.; Chen, S.; Hu, X.; Li, J.; Yang, J. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *arXiv* **2020**, arXiv:2006.04388. [CrossRef]
27. Xia, Z.; Pan, X.; Song, S.; Li, L.E.; Huang, G. Vision transformer with deformable attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 4794–4803. [CrossRef]
28. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 213–229. [CrossRef]
29. Hendrycks, D.; Gimpel, K. Gaussian error linear units (gelus). *arXiv* **2016**, arXiv:1606.08415. [CrossRef]

30. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805. [CrossRef]
31. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI blog* **2019**, *1*, 9.
32. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 11–13 April 2011; pp. 315–323.
33. Elfwing, S.; Uchibe, E.; Doya, K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Netw.* **2018**, *107*, 3–11. [CrossRef]
34. Dunnings, A.; Breckon, T.P. *Fire Image Data Set for Dunnings 2018 Study-PNG Still Image Set*; Durham University: Durham, UK, 2018. [CrossRef]
35. Dedeoglu, N.; Toreyin, B.U.; Gudukbay, U.; Cetin, A.E. Real-time fire and flame detection in video. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05), Philadelphia, PA, USA, 18–23 March 2005; Volume 2, pp. 669–672. [CrossRef]
36. Ko, B.; Kwak, J.Y.; Nam, J.Y. Wildfire smoke detection using temporospatial features and random forest classifiers. *Opt. Eng.* **2012**, *51*, 017208. [CrossRef]
37. Zhang, Q.X.; Lin, G.H.; Zhang, Y.M.; Xu, G.; Wang, J.J. Wildland forest fire smoke detection based on faster R-CNN using synthetic smoke images. *Procedia Eng.* **2018**, *211*, 441–446. [CrossRef]
38. 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]
39. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1026–1034. [CrossRef]
40. Tang, Y.; Han, K.; Guo, J.; Xu, C.; Xu, C.; Wang, Y. GhostNetv2: Enhance cheap operation with long-range attention. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 9969–9982. [CrossRef]
41. Cui, C.; Gao, T.; Wei, S.; Du, Y.; Guo, R.; Dong, S.; Ma, Y. PP-LCNet: A lightweight CPU convolutional neural network. *arXiv* **2021**, arXiv:2109.15099. [CrossRef]
42. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 116–131. [CrossRef]
43. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Adam, H. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324. [CrossRef]
44. An, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114. [CrossRef]
45. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual, 11–17 October 2021; pp. 10012–10022. [CrossRef]