

Article

DCP-Net: An Efficient Image Segmentation Model for Forest Wildfires

Lei Qiao ¹, Wei Yuan ^{1,*}  and Liu Tang ^{2,3,*}¹ College of Computer Science, Chengdu University, Chengdu 610106, China² Sichuan Province Engineering Technology Research Center of Healthy Human Settlement, Chengdu 610225, China³ Sichuan University Engineering Design & Research Institute Co., Ltd., Chengdu 610225, China

* Correspondence: yuanwei@cdu.edu.cn (W.Y.); jkrjwk@163.com (L.T.)

Abstract: Wildfires usually lead to a large amount of property damage and threaten life safety. Image recognition for fire detection is now an important tool for intelligent fire protection, and the advancement of deep learning technologies has enabled an increasing number of cameras to possess functionalities for fire detection and automatic alarm triggering. To address the inaccuracies in extracting texture and positional information during intelligent fire recognition, we have developed a novel network called DCP-Net based on UNet, which excels at capturing flame features across multiple scales. We conducted experiments using the Corsican Fire Dataset produced by the “Environmental Science UMR CNRS 6134 SPE” laboratory at the University of Corsica and the BoWFire Dataset by Chino et al. Our algorithm was compared with networks such as SegNet, UNet, UNet++, and PSPNet, demonstrating superior performance across three metrics: mIoU, F1-score, and OA. Our proposed deep learning model achieves the best mIoU (78.9%), F1-score (76.1%), and OA (96.7%). These results underscore the robustness of our algorithm, which accurately identifies complex flames, thereby making a significant contribution to intelligent fire recognition. Therefore, the proposed DCP-Net model offers a viable solution to the challenges of wildfire monitoring using cameras, with hardware and software requirements typical of deep learning setups.

Keywords: fire detection; transformer; deep learning; semantic segmentation



Citation: Qiao, L.; Yuan, W.; Tang, L. DCP-Net: An Efficient Image Segmentation Model for Forest Wildfires. *Forests* **2024**, *15*, 947. <https://doi.org/10.3390/f15060947>

Academic Editor: Alistair M. S. Smith

Received: 17 April 2024

Revised: 15 May 2024

Accepted: 28 May 2024

Published: 30 May 2024



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

1. Introduction

Fire is currently one of the most common and widespread threats to social security and development. Given the escalating impact of global climate change, the increasingly frequent outbreaks of wildfires pose a significant global challenge [1]. Statistics from the European Forest Fire Information System show that in 2021, the area of forest fires in Spain reached 4260 hectares, while in Italy it exceeded 150,000 hectares and in Greece it reached 93,600 hectares [2–4]. These fires not only cause destruction to forest ecosystems [5] but also pose a severe threat to human living environments. Therefore, fire alarm systems have become an indispensable part of modern intelligent firefighting. In order to mitigate the harm caused by fires, many detection methods have been proposed to reduce the damage caused by such accidents. Fire detection methods can be divided into traditional fire alarm detection and visual sensor detection.

Traditional fire alarm detector systems commonly utilize sensors such as smoke, light, and heat detectors to detect fires [6]. These detectors typically require a certain level of fire intensity to be reached before they can function effectively. When the fire source is close enough, smoke and fire can be detected through ionized particles generated by the fire, which then trigger fire alarms and suppression systems. Although these systems are robust, they often fail to detect fires promptly and require manual intervention to confirm fire information when triggering alarms. Additionally, due to the strong destructive power and rapid spread of fires, detectors are prone to delays in detection and may miss the

optimal time for early fire suppression due to factors such as distance. To overcome the aforementioned drawbacks of traditional sensors, researchers have explored various detection methods based on visual sensors. However, vision-based detection methods encounter a significant obstacle in accurately identifying and analyzing the fire front in surface fires, specifically, delineating the boundary of the fire as it spreads across the ground [7]. This is because flames exhibit irregular shapes, scales, and are often interfered with by complex backgrounds [8]. The early stage framework for visual fire detection mainly consists of three stages. Firstly, different scales of sliding windows are used to traverse the input image to obtain potential fire regions. Secondly, manual feature extraction methods such as HOG [9], SIFT [10], LBP [11], etc., are utilized to extract features such as color, edges, and texture of the flames from candidate regions. These region images are then converted into feature vectors and passed to a classifier for training. Finally, classifiers such as SVM [12], Bayesian networks, random forests, BP neural networks, etc., are employed to compare these extracted features with a set of existing standard features to determine whether the image contains fire. Qiu et al. [13] proposed a new algorithm to clearly and continuously define the edges of flames and fire spots. Experimental results in the laboratory using different flame images and video frames demonstrated the effectiveness and robustness of this algorithm. However, further evaluation of the algorithm's performance in real-life fire detection scenarios is yet to be conducted. Celik et al. [14] proposed a real-time fire detector that integrates foreground object information and statistical information of fire-colored pixels. Subsequently, a general statistical model was used to refine the classification of fire pixels. The final correct detection rate reached 98.89%. However, such extraction methods involve significant redundant computation costs in the image preprocessing stage, which can affect the algorithm's speed and fail to extract deep image information.

As artificial intelligence advances within the realm of computer vision, deep learning [15] has emerged as the mainstream approach, leveraging its ability to automatically extract required features. Deep learning is a multi-layer neural network algorithm capable of automatically learning data features from datasets, and it has been applied to analyze and extract information from images captured by drones [16–20]. Lecun first proposed the use of convolutional neural networks (CNN) in 1998 with LeNet [21], which employed weight sharing to reduce the computational load of neural networks, greatly advancing the application of deep learning in image recognition. Although CNNs perform well in many tasks, they also have limitations, such as slow parameter updates during backpropagation, convergence to local optima, loss of information in pooling layers, and unclear interpretation of feature extraction. The Transformer model [22] was initially proposed by the Google team in 2017, replacing the convolutional neural network components of CNNs with self-attention modules. This model utilizes multiple attention heads to process and capture different input data features, thereby enhancing feature extraction capabilities. However, Transformers have high computational costs for image processing. Therefore, the Microsoft team proposed Swin Transformer [23], which divides images into multiple windows and limits Transformer calculations to within each window to reduce computational complexity, demonstrating excellent performance. With the increases in the depth and complexity of models, segmentation accuracy has comprehensively surpassed traditional machine learning methods, becoming the mainstream approach. Many scholars have also adopted deep learning methods in fire detection. Different domains have seen the application of various deep learning models for their respective tasks. Jadon et al. [24] constructed a lightweight neural network called FireNet, which can be deployed on a Raspberry Pi 3B [25] to replace conventional physical sensors while also providing remote verification functionality by offering real-time visual feedback in the form of alert messages using the Internet of Things (IoT). The disk space occupied by FireNet is only 7.45 MB, and it can run steadily at a frame rate of 24 frames per second, achieving over an accuracy of over 93% on experimental datasets. Jitendra Musale et al. [26] developed an efficient method based on transfer learning using the convolutional neural network Inception-v3, which divides the dataset into fire and non-fire images by training on satellite images. Zhang et al. [27],

in their work on forest fire detection, utilized fire patches detection with a fine-tuned pretrained CNN, 'AlexNet' [28], while Sharma et al. [29] also proposed a CNN-based fire detection approach using VGG16 [30] and Resnet50 [31] as baseline architectures.

Many researchers have also modified the YOLO series for fire detection. Xue et al. [32] modified the Spatial Pyramid Pooling-Fast (SPPF) module from YOLOv5 to develop the Spatial Pyramid Pooling-Fast-Plus (SPPFP) module specifically for fire detection, achieving a 10.1% improvement in mAP@0.5 on their dataset. Zhu et al. [33] used an improved YOLOv7-tiny model to detect cabin fires, resulting in a 2.6% increase in mAP@0.5 and a 10 fps increase in frame rate. Hojune Ann et al. [34] developed a fire risk detection system that detects fire sources and combustible materials simultaneously by object detection on images captured by surveillance cameras, comparing the performance of two deep learning models, YOLOv5 and EfficientDet.

Despite notable progress achieved in fire detection through deep learning technology, a considerable gap remains in wildfire image segmentation [16]. Compared to traditional fire detection methods, wildfire image segmentation techniques can provide more detailed information about fires, including fire scale, flame spread rate, and precise fire location. This information is essential for formulating efficient prevention and control strategies and for allocating firefighting resources sensibly. Wang et al. [35], based on Swin Transformer, combined adaptive multiscale attention mechanisms and a focal loss function to segment forest fire images, achieving an IoU of 86.73%. Compared to traditional models such as PSPNet [36], SegNet [37], DeepLabV3 [38], and FCN [39], their method demonstrated significant improvements. Muhammad et al. [40] proposed an original, energy-efficient, and computationally efficient CNN architecture. It uses smaller convolutional kernels and contains no dense fully connected layers, striking a balance between segmentation accuracy and efficiency.

Although many scholars have used various CNN networks or Transformer networks for fire detection work, CNNs can only capture local features, and capturing global features requires layer-by-layer stacking. In contrast, Transformer models perform well in handling long-range dependencies and global contexts, but they have a large computational overhead, resulting in slow processing speeds. Additionally, an excess of global features may overshadow certain local features, leading to suboptimal detection performance. To address this issue, this paper investigates a Dynamic Contextual Pooling (DCP) module and designs a network called DCP-Net, which effectively integrates multi-scale features and local features while considering contextual and global correlations. The model is trained using flame images captured by ground cameras. When performing inference on the images, it operates offline, classifying the images at the pixel level into two categories: flame and background, ultimately separating the flame from the background in the images. The model identifies only the flame. If there is smoke in the images but no visible flame, the model will not detect the presence of a fire. When using GPU for inference, a graphics card with approximately 4 GB or more of memory is required. It is also possible to use CPU for inference, but the speed will be extremely slow, roughly a fraction of the speed of GPU inference. The overall inference speed depends on the hardware performance. This model can segment flames more accurately, improving the precision of flame boundary segmentation.

1.1. Innovation and Contribution of This Paper

This paper mainly focuses on improving the accuracy of extracting features from fire images. The contributions of this paper are as follows:

- (1) We propose a module named DCP, which integrates features extracted by Dynamic Snake Convolution, Contextual Transformer, and partial convolution. This module comprehensively captures multi-scale feature information in context, enhancing the effectiveness of multi-level feature fusion in decoding.

(2) We propose a network that combines local features with multi-scale features. This network employs CNNs to capture local features and utilizes the DCP module to capture multi-scale image features integrated with contextual and global information, and effectively fuse them.

(3) We train the proposed network on a fire dataset and validate it on another dataset to analyze the generalization performance of each network.

1.2. Related Work

1.2.1. UNet

UNet [41] stands as a prevalent framework across various computer vision domains, including image classification and segmentation models. Its hierarchical feature map representation enables the capture of detailed multi-scaled contextual information. Furthermore, it enhances image reconstruction by leveraging residual connections between the folding and unfolding pathways. Notably, UNet boasts several advanced iterations, such as SegFormer, Swin-UNet, and TransUNet. Consequently, integrating a robust and adaptive feature extractor backbone into the UNet architecture could significantly elevate the model's overall performance.

1.2.2. Shift Pooling PSPNet

The essence of PSPNet lies in its pyramid pooling module, enabling it to effectively capture local features across various scales. Nonetheless, this module has notable drawbacks, particularly its fixed grid structure, which prevents pixels near the grid's edges from accessing complete local features. To overcome this limitation, an enhanced PSPNet architecture called Shift Pooling PSPNet [42] is introduced. This new approach replaces the traditional pyramid pooling module with a module known as shift pyramid pooling, allowing even edge pixels to access comprehensive local features.

1.2.3. NestedUNet

NestedUNet [43] is an advanced convolutional neural network architecture used for semantic segmentation tasks, notably in medical image analysis. It extends the U-Net model by nesting multiple U-Net modules within each other, enabling hierarchical feature extraction at various levels of abstraction. With its contracting and expansive paths, along with skip connections for information flow, NestedUNet excels in accurately delineating intricate structures in images, making it particularly valuable for tasks requiring precise segmentation, such as organ or tumor delineation in medical imaging. In addition to medical image analysis, NestedUNet can also be effectively utilized for semantic segmentation tasks in various domains such as satellite imagery, autonomous driving, and remote sensing.

2. Materials and Methods

2.1. Dynamic Snake Convolution

Yaolei Qi and their team introduced a method called Dynamic Snake Convolution [44], specifically designed for extracting features from tubular structures. This method possesses the unique ability to dynamically focus on the delicate and convoluted parts within such structures. Given that flames often display diverse shapes and their boundaries are subject to rapid changes, resembling to some extent the characteristics of tubular structures mentioned in the original text, Dynamic Snake Convolution (Figure 1) can be effectively employed for precise feature extraction of flame boundaries, thereby facilitating accurate segmentation.

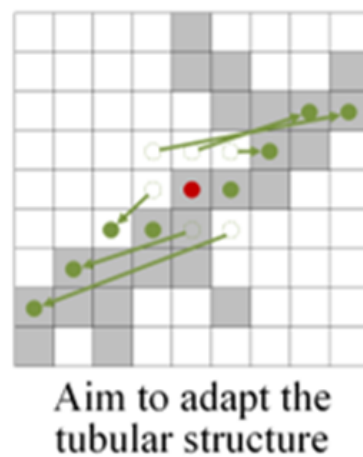


Figure 1. Dynamic Snake Convolution.

2.2. Partial Convolution

To design faster neural networks, much effort has been devoted to reducing the number of floating-point operations (FLOPs). However, merely reducing FLOPs does not necessarily result in faster computation speed, mainly because networks with lower FLOPs do not always have higher floating-point operations per second (FLOPS). Lower FLOPS values are often caused by frequent memory access, particularly evident when using depthwise convolution. Therefore, Jierun Chen et al. proposed a method called partial convolution [45] to reduce redundant computation and memory access. This method applies conventional convolution to a portion of input channels for spatial feature extraction while keeping the remaining channels unchanged (Figure 2). This approach achieves a balance between speed and accuracy.

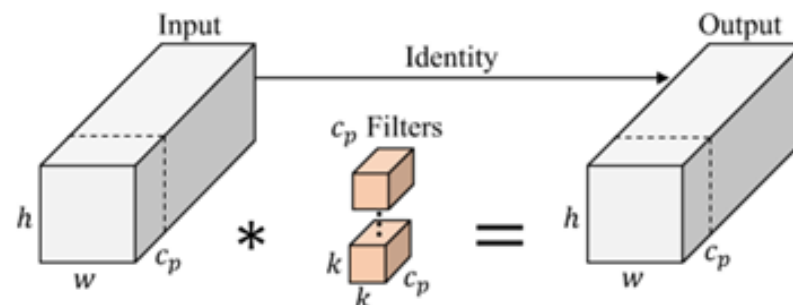


Figure 2. Partial convolution (PConv). It simply applies a regular convolution on only a part of the input channels for spatial feature extraction and leaves the remaining channels untouched.

2.3. Contextual Transformer

The Transformer emerged as a classic feature extraction algorithm following convolutional neural networks, gradually becoming mainstream due to its ability to address the limitations of CNNs in extracting only local features. However, most existing studies in the visual domain directly use self-attention on 2D feature maps to obtain attention matrices based on queries and keys at each spatial position, without fully exploiting the rich context between neighboring keys. Therefore, Yehao Li et al. designed the Contextual Transformer [46] for visual recognition tasks (Figure 3). This design fully utilizes contextual information between input keys to guide the learning of dynamic attention matrices, thereby enhancing the capability of visual recognition. The network can better capture spatial information in flame images. This structure can be utilized to construct a novel flame segmentation network, aimed at improving the accuracy and robustness of flame segmentation, particularly in complex scenarios such as low-light environments where visual confusion exists between flames and backgrounds, or during sunset where the bright

parts of flames may resemble the sunset glow, leading the network to mistakenly identify the sunset as flames.

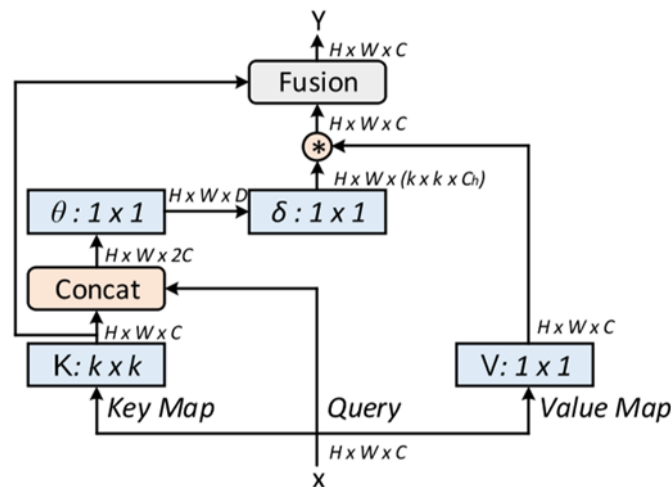


Figure 3. Contextual Transformer (CoT) block.

2.4. DCP Block

To effectively segment flame images, we propose the DCP (Dynamic Contextual Partial) block, based on the aforementioned three blocks. This module integrates the characteristics of the Dynamic Snake Convolution, Contextual Transformer, and partial convolution blocks. Dynamic Snake Convolution extracts subtle local features from irregular parts of the image. The Contextual Transformer captures global information of the image while considering contextual information between feature map pixels. The partial convolution efficiently extracts image information while reducing redundant computations. The DCP block effectively combines the features of these blocks, possessing the ability to comprehensively extract both local and global information. It can capture fine local features while considering contextual and global correlations between features, exhibiting good performance in flame segmentation tasks. The specific structure of this module is illustrated in the diagram below:

In Figure 4, the input feature map has a size of $H \times W \times C$. This feature map is duplicated three times to obtain feature maps A, B, and C. Then, feature maps A, B, and C are processed separately through the PC (partial convolution) block, CoT (Contextual Transformer) block, and DSC (Dynamic Snake Convolution) block, resulting in feature maps D, E, and F. Subsequently, the obtained feature maps are added together to generate an output image with different scale features, maintaining the size of $H \times W \times C$ as the input feature map. Therefore, the DCP (Dynamic Convolutional Pooling) module does not alter the size or the number of channels of the feature map. The pseudocode for the DCP module is depicted in Algorithm 1.

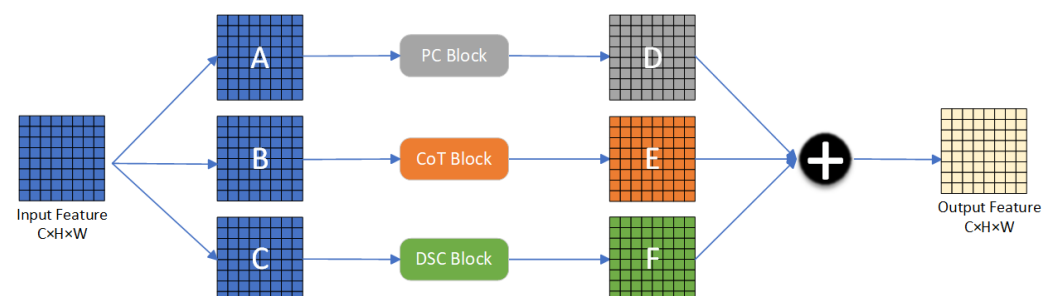


Figure 4. Architecture diagram of the DCP module.

Algorithm 1 DCP module**Input:** feature map**Output:** feature map

- 1: Copy the input feature map three times, obtaining feature maps A , B , C
- 2: Pass feature map A through PC block, obtaining feature map D
- 3: Pass feature map B through CoT block, obtaining feature map E
- 4: Pass feature map C through DSC block, obtaining feature map F
- 5: Add feature maps D , E , and F together to obtain the output feature map
- 6: **return** output feature map

2.5. DCP-Net

The Transformer module effectively captures global features. However, in the recognition of flame pixels, the importance of local features may outweigh that of global features. Excessive global feature information may obscure some critical local features. Therefore, we propose the DCP-Net network. This network utilizes convolutional neural networks and SCP modules to extract multi-scale image features for fusion. The SCP module utilizes the Contextual Transformer module to extract global features and context information. Finally, the network gradually upsamples to the original input image size. The architecture diagram of DCP-Net is shown in Figure 5:

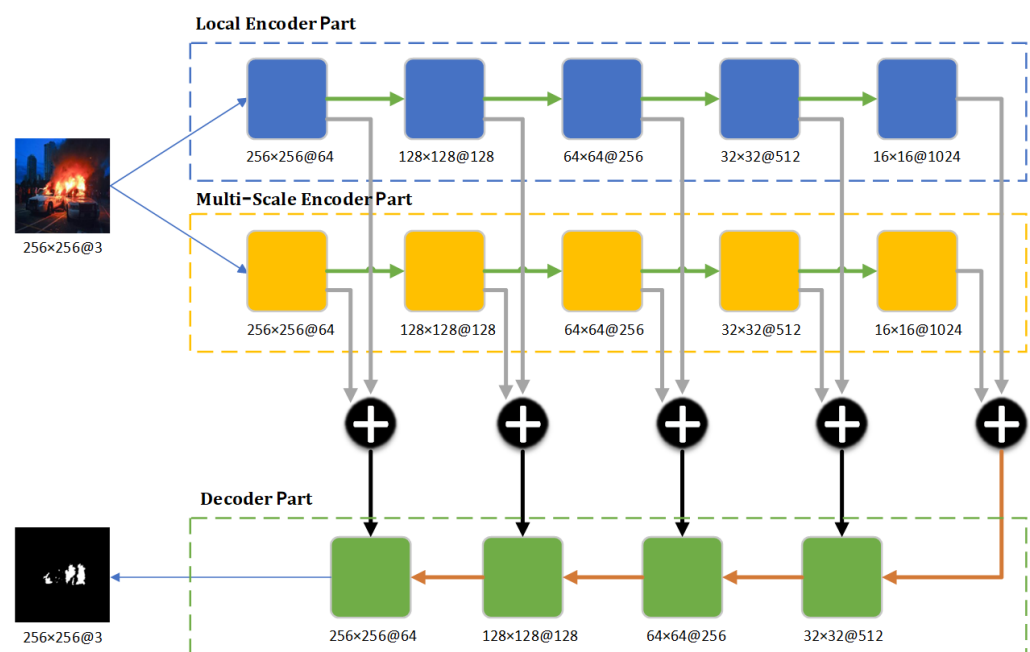


Figure 5. Network architecture of DCP-Net. The blue blocks represent two consecutive layers of Conv2D, BatchNorm2D, and ReLU, with a kernel size of 3 and a stride of 1. The yellow blocks represent the DCP module. The green blocks represent two consecutive layers of Conv2D, BatchNorm2D, and ReLU, with a kernel size of 3 and a stride of 1. The black circle represents the final extracted features obtained after combining local and global features. The green arrow pointing right represents Maxpooling2D with a kernel size of 2 and a stride of 2. The orange arrow pointing left represents ConvTranspose2D with a kernel size of 2 and a stride of 2. The gray arrow pointing downward represents addition. The black arrow pointing downward represents concatenation. The encoding part of DCP-Net adopts the same 5-layer structure as UNet. The encoding part consists of two parts: a CNN that captures local features and a DCP module that captures multi-scale features.

DCP-Net consists of three main parts in total. The topmost row of blue blocks represents the local feature extraction section, which includes five stages and outputs feature maps at five different scales. The original input image size is $256 \times 256 \times 3$. After passing through two convolutional modules with a kernel size of 3 and a stride of 1, followed by

BatchNorm and ReLU modules, the image size becomes $256 \times 256@64$, producing the first-stage feature map. Then, through a MaxPooling module with a kernel size of 2 and a stride of 2, the feature map size changes to $128 \times 128@64$. After passing through two convolutional modules with a kernel size of 3 and a stride of 1, followed by BatchNorm and ReLU modules, the size becomes $128 \times 128@128$, producing the second stage feature map. Next, after going through a MaxPooling module with a kernel size of 2 and a stride of 2, the feature map size changes to $64 \times 64@128$. Passing through two convolutional modules with a kernel size of 3 and a stride of 1, followed by BatchNorm and ReLU modules, produces the third stage feature map with a size of $64 \times 64@256$. Similarly, after passing through a MaxPooling module with a kernel size of 2 and a stride of 2, the feature map size changes to $32 \times 32@256$. Passing through two convolutional modules with a kernel size of 3 and a stride of 1 followed by BatchNorm and ReLU modules, produces the fourth stage feature map with a size of $32 \times 32@512$. Finally, the process is repeated, and after passing through a MaxPooling module with a kernel size of 2 and a stride of 2, the feature map size changes to $16 \times 16@512$. Passing through two convolutional modules with a kernel size of 3 and a stride of 1, followed by BatchNorm and ReLU modules, produces the fifth stage feature map with a size of $16 \times 16@1024$.

The yellow part in the middle of Figure 5 represents the multi-scale feature extraction section. Similar to the local feature extraction section, this section also consists of five stages and outputs feature maps at five different scales, with each feature map size matching the size of the feature maps in the local feature extraction section. The input image size is $256 \times 256@3$. Since our DCP module does not change the size of the image and can adjust the number of channels, there is no need for linear projection to stretch the image channels. The hidden layer size of the DCP module in the first stage is 32, with an input channel of 3 and an output channel of 64. After passing the original image through the first stage DCP module, a feature map size of $256 \times 256@64$ is obtained for the first stage. Then, after going through a MaxPooling module with a kernel size of 2 and a stride of 2, the feature map size changes to $128 \times 128@64$. The hidden layer size of the DCP module in the second stage is also 32, with an input channel of 64 and an output channel of 128, resulting in a feature map size of $128 \times 128@128$ for the second stage. Once again, after going through the same MaxPooling module, the feature map size changes to $64 \times 64@128$. The hidden layer size of the DCP module in the third stage is also 32, with an input channel of 128 and an output channel of 256, resulting in a feature map size of $64 \times 64@256$ for the third stage. After going through the same MaxPooling module again, the feature map size changes to $32 \times 32@256$. The hidden layer size of the DCP module in the fourth stage is also 32, with an input channel of 256 and an output channel of 512, resulting in a feature map size of $32 \times 32@512$ for the fourth stage. Finally, after going through the same MaxPooling module again, the feature map size changes to $16 \times 16@512$. The hidden layer size of the DCP module in the fifth stage is different from the previous stages and is set to 16. The input channel is 512 and the output channel is 1024, resulting in a feature map size of $16 \times 16@1024$ for the fifth stage. By adding the feature maps extracted from different scales in the local feature extraction section and the multi-scale feature extraction section at the corresponding stages, the final features extracted by the encoder are obtained. As shown in the black circles in Figure 2, the first-stage feature map has a size of $256 \times 256@64$, the second-stage feature map has a size of $128 \times 128@128$, the third-stage feature map has a size of $64 \times 64@256$, the fourth-stage feature map has a size of $32 \times 32@512$, and the fifth-stage feature map has a size of $16 \times 16@1024$.

The green part at the bottom of Figure 5 is the decoder module. This decoding process involves progressive upsampling and simultaneously fusing features of the same level. First, the feature map from the fifth stage is passed through a deconvolution layer, which doubles the height and width while halving the number of channels. This resulting feature map is then concatenated with the feature map from the fourth stage. After passing through two convolutional modules with a kernel size of 3 and a stride of 1, followed by BatchNorm and ReLU modules, a feature map of size $32 \times 32@512$ is obtained. The feature

map is then passed through a deconvolution layer and concatenated with subsequent feature maps. After passing through two convolutional modules, BatchNorm and ReLU modules, a feature map size of $64 \times 64@256$ is obtained. Continuing the upsampling process in the same way, subsequent feature maps are fused, resulting in a feature map size of $256 \times 256@64$. Finally, the feature map is passed through a convolutional layer with a kernel size of 1 and a stride of 1 to adjust the output size to $256 \times 256@2$.

The DCP-Net model utilizes an encoder-decoder architecture, where the encoder component extracts pertinent features from the input image, and the decoder component reconstructs the spatial intricacies of the image to facilitate accurate segmentation. This enables the model to accurately segment images, especially in complex scenarios. However, this also means that computational complexity increases, parameter adjustment becomes difficult, and the model itself becomes relatively more complex.

2.6. Evaluation Metrics

We use three metrics to evaluate each deep learning model: F1-score (which consists of precision and recall), mIoU (mean intersection over union), and OA (overall accuracy), three metrics to evaluate each deep learning model. They are calculated as shown in Equations (1)–(7). The formula of mIoU is:

$$mIoU = \frac{1}{N+1} \sum_{i=0}^N \frac{TP}{TP + FN + FP} \quad (1)$$

The formula of OA is:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

The formula of the F1-score is:

$$F1\text{-Score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (3)$$

where precision and recall are

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

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

In Formula (1), N represents the count of foreground pixels. TP denotes true positives, indicating the accurate prediction of foreground pixels. Conversely, FP signifies false positives, representing background pixels mistakenly identified as foreground. TN stands for true negatives, denoting correctly predicted background pixels. FN indicates false negatives, representing foreground pixels erroneously classified as background. Both fire and background are considered foreground to calculate the intersection over union (IoU), with the resulting average being the mIoU. In Equations (2)–(5), the foreground specifically refers to fire.

2.7. Data Preparation

The datasets used in the experiments are the BoWFireDataset [47] and the Corsican Fire Database [48]. The BoWFireDataset consists of 226 images with varying resolutions, divided into two categories: 119 images contain fire, and 107 images do not contain fire. The fire images include emergency situations involving different fire-related incidents, such as building fires, industrial fires, industrial fire accidents, and vehicle accidents, as well as disturbances like riots. The remaining images include emergency situation images without any apparent fire presence and images with regions that may resemble fire, such as sunsets

or red and yellow objects. These images have been manually cropped by professionals to create realistic images of fire regions. The fire regions on ground truth images are marked with white labels, while non-fire regions are marked with black labels. The Corsican Fire Database was collected as part of the “Fire” project conducted by the “Environmental Sciences UMR CNRS 6134 SPE” laboratory at the University of Corsica. The project focuses on modeling and experimenting with vegetation fires. The database contains 500 images and image sequences of wildfires captured under different conditions such as different shooting angles, types of burning vegetation, climatic conditions, brightness, and distances from the fire. These images were taken in the visible light and near-infrared regions, with the primary flame colors being red, orange, and yellow. Each image is accompanied by related data, including fire pixels manually selected by professionals (represented by white pixels), the dominant color of the fire, the percentage of fire pixels in the image, the percentage of fire pixels covered by smoke, and the texture level of the fire region. All of these parameters are associated with each image in the database.

The Corsican Fire Database and a subset of the BoWFire Database images were chosen for training due to the relatively small dataset size. The BoWFire Dataset contains masks for only 226 images, including 119 images with fire and 107 images without fire but with objects similar to fire, such as sunsets or neon lights. After removing the near-infrared captured images from the Corsican Fire Dataset, the dataset contains 1135 fire images with masks, all of which were used as the training set. Since the Corsican Fire Dataset consists entirely of fire images, it can be challenging for the trained model to distinguish between fire and objects similar to fire. Therefore, 36 images without fire from the BoWFire Dataset were included in the training set to improve the model’s performance. The remaining 71 non-fire images and 119 fire images from the BoWFire Dataset were combined to form the test set. As a result, the final training set comprises 1171 images, including both fire and non-fire images. The test set consists of 190 images, also including fire and non-fire images. All images and their corresponding masks were cropped to 256×256 pixels. The images were converted to JPG format, while the masks were converted to PNG format, with fire pixels assigned a value of 1 and non-fire pixels assigned a value of 0.

3. Experimental Analysis

Hardware and Software for Experiment

The hardware configuration of the computer used for the experiments is as follows: the CPU (Intel, USA) is an Intel i5-13600KF, RAM is SEIWHALE (China) DDR4 $16\text{G} \times 2$, and the GPU (Gigabyte, China) is an NVIDIA GeForce RTX 2080Ti 22G. The version of Python is 3.10.12, and Pytorch is used as the deep learning framework for model training and evaluation (Table 1).

Table 1. Hardware and software details.

Hardware and Software	Parameters
CPU	Intel i5-13600KF
GPU	NVIDIA GeForce RTX 2080Ti
Operating memory	32 GB
Total video memory	22 GB
Operating system	Ubuntu 22.04.4
Python	Python 3.10.12
IDE	PyCharm 2022.1.4
CUDA	CUDA 12.1
CUDNN	CUDNN 8.9.6
Deep learning architecture	PyTorch 2.0.1

The Adam optimizer was used for backpropagation, batchsize was set to 4, and the learning rate was set to 0.0001 during training. Because the default EPS is too small, which can cause some models to have a LOSS of NAN during training, we set the EPS to 0.003.

The total loss, comprising the sum of L2 regularization and binary cross-entropy, was employed to mitigate overfitting, as depicted in Formula (1). The training process was limited to a maximum of 300 epochs, with evaluation conducted on the validation dataset after each epoch. our stopping standard was that if the loss in the test dataset no longer reduced for 20 consecutive epochs, then training was stopped.

$$\begin{aligned} \text{TotalLoss} &= \text{BinaryCrossEntropy} + L2 \\ L2 &= \|w\|_2^2 = \sum_i |w_i|^2 \end{aligned} \quad (6)$$

4. Results

In order to impartially showcase the superiority of the proposed DCP-Net, we opted to use three objective evaluation metrics: mIoU, F1-score, and OA. We conducted training and validation on a dataset with SegNet, UNet, PSPNet, ShiftPoolingPSPNet, NestedUNet, and our DCP-Net, and assessed their performance on the test set. The results for each network on the test set are summarized in Table 2.

Table 2. Results of classic semantic segmentation on the test dataset.

Methods	mIoU (%)	F1-Score (%)	OA (%)
SegNet	73.5	68.1	95.6
UNet	76.9	73.4	96.2
PSPNet	76.8	72.8	96.5
ShiftPoolingPSPNet	77.1	73.7	96.1
NestedUNet	78.0	74.5	96.7
DCP-Net	78.9	76.1	96.7

As shown in Table 2, SegNet performed the worst among all models, with the lowest mIoU, F1-score, and OA, which are 73.5, 68.1, and 95.6, respectively. Compared to UNet, SegNet scored 3.4 points lower in mIoU, 5.3 points lower in F1-score, and 0.6 points lower in OA. Although SegNet was an improvement over UNet, the experimental comparison results indicate that SegNet's performance was far behind UNet on the dataset used in this study. This could be due to the significant differences between our test set and training set, as well as SegNet's relatively poor generalization capability. In our experiment, PSPNet used ResNet50 as its backbone. PSPNet's performance was similar to UNet, although PSPNet's mIoU was 0.1 points lower, and its F1-score was 0.6 points lower. However, its OA was 0.3 points higher. ShiftPoolingPSPNet performed slightly better than UNet and PSPNet, but the improvement was minimal, with mIoU, F1-score, and OA reaching 77.1, 73.7, and 96.1, respectively. NestedUNet showed significant improvement over the previously mentioned models, with mIoU, F1-score, and OA values reaching 78.0, 74.5, and 96.7, respectively. This suggests that NestedUNet exhibited superior generalization capabilities compared to the other models. The best performer was our proposed DCP-Net, which integrates both local and global features. It achieved the best results across all three metrics, with mIoU, F1-score, and OA values reaching 78.9, 76.1, and 96.7, respectively. This demonstrates that our proposed DCP-Net offers better segmentation capabilities, effectively captures image features, and has superior generalization abilities.

Figure 6 shows the predictions of various models on a fire image from the test set. In the first row, the first image on the left is the input fire image, the second image is the ground truth, the third image is the prediction from SegNet, and the fourth image is the prediction from the PSPNet network. In the second row, from left to right, are predictions from UNet, ShiftPoolingPSPNet, NestedUNet, and DCP-Net. In the figure, red pixels represent false negatives, which are pixels of fire that were missed. Green pixels represent false positives, which are background pixels mistakenly detected as fire. Black represents true negative pixels, and white represents true positive pixels.

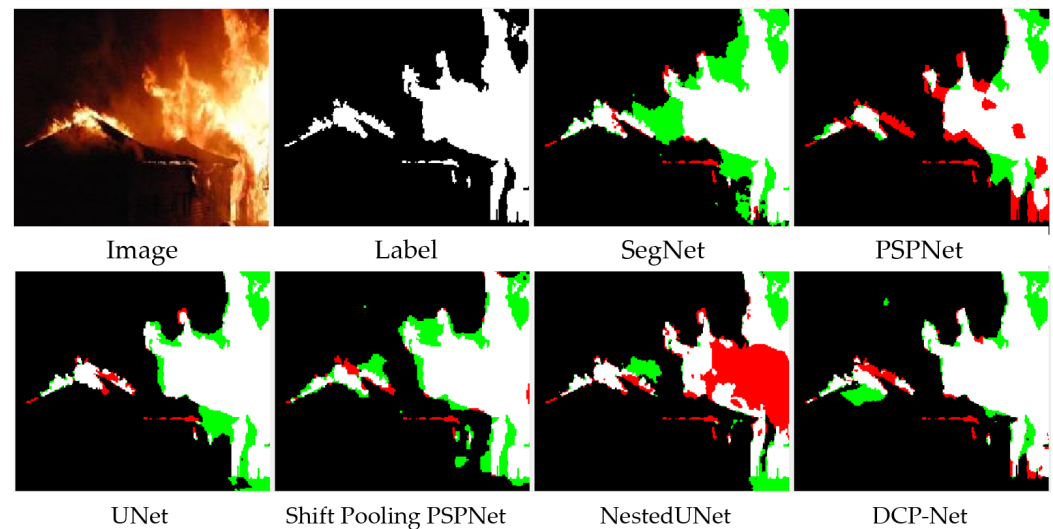


Figure 6. The output results of each model on the test set. Black represents true negative (TN) pixels, white represents true positive (TP) pixels, red represents false negative (FN) pixels, and green represents false positive (FP) pixels.

The predicted images reveal the segmentation performance of different models. SegNet has a significant number of green pixels, indicating a high false positive rate, which aligns with the lower metrics in Table 1. PSPNet has a considerable amount of both green and red pixels. UNet and ShiftPoolingPSPNet have similar outcomes, with a substantial number of green pixels, indicating a high false positive rate. PSPNet, UNet, and ShiftPoolingPSPNet are similar in pixel-level recognition accuracy, as they all have a similar quantity of red and green pixels but less than SegNet, matching the SCPn the experimental results table. NestedUNet provides clearer fire boundary delineation but contains significant areas of missed detection. Our proposed DCP-Net performs the best, leveraging its enhanced global feature capture and feature fusion capabilities. DCP-Net has the fewest red and green pixels, indicating the lowest false positive and false negative rates, and its fire boundary delineation is the clearest and most consistent with the ground truth.

5. Discussion

Our model exhibits excellent recognition performance in various scenarios such as red light, sunset, and red buildings. This slightly differs from previous research findings, indicating the strong generalization ability of our model. Unlike many prior studies conducted on the same dataset, our model's robust performance across diverse scenes suggests its potential for real-world applications in different environments and lighting conditions.

To investigate the contribution of each module in DCP-Net and whether the DCP module is more effective than other modules, we conducted ablation experiments, the experiments were conducted using the same test dataset as in previous experiments. The results are shown in Table 3. It can be seen that when using only CNN to capture global features, the mIoU is only 76.9, the F1-score is 73.4, and the accuracy is 96.2.

Table 3. Ablation experiment on test dataset.

Methods	mIoU	F1-Score	OA
CNN	76.9	73.4	96.2
CNN + Swin Transformer	78.1	74.8	96.6
CNN + DCP	78.9	76.1	96.7

After adding a Swin Transformer module to capture local features, there is an increase in all three metrics: mIoU, F1-score, and OA reach 78.1, 74.8, and 96.6, respectively. Replacing the Swin Transformer with the DCP module led to a further increase in all three

metrics, reaching 78.9, 76.1, and 96.7. This demonstrates that the proposed DCP module performs better than the Swin Transformer.

6. Conclusions

To address the issues of unclear and easily confusable boundaries in flame segmentation, we propose a block called DCP, which involves replicating the original input feature image and passing it through the PC block, CoT block, and DSC block to obtain corresponding different feature maps. These feature maps are then summed together. Additionally, to better integrate local features, the feature maps extracted by CNN and the DCP block are fused at each scale. DCP-Net adopts the same five-level structure as UNet, with decoding performed through incremental upsampling, resulting in a significant enhancement in flame recognition accuracy compared to existing methods. These technological advancements represent a substantial stride towards precise identification and analysis of fire fronts within intricate wildfire contexts. Furthermore, our validation on different datasets demonstrates the robust generalization capability of DCP-Net. Outperforming mainstream models such as SegNet, UNet, PSPNet, Shift Pooling PSPNet, and NestedUNet, our model consistently exhibits superior performance across multiple evaluation metrics. Notably, the low rates of false positives and false negatives, along with close alignment with ground truth in prediction images, underscore the reliability and efficacy of our approach.

In broader terms, the success of DCP-Net highlights the potential of advanced deep learning techniques in enhancing wildfire monitoring and management efforts. By improving the accuracy and efficiency of flame segmentation, our work contributes to the development of more effective tools for wildfire detection and mitigation, ultimately aiding in the protection of lives and ecosystems. However, it is essential to acknowledge the challenges associated with obtaining high-quality annotated data, especially for certain specific domains or tasks. A lack of sufficient annotated data may limit the model's generalization ability. This model is designed to segment forest fires, but its generalization for flames of other colors is limited. It is tailored for the orange flames commonly found in natural forest environments and may not be suitable for purple or blue flames typically encountered in industrial settings.

In future work, we plan to further explore how to optimize the model structure without compromising accuracy to reduce the computational load and thus increase processing speed. This research direction holds the potential to significantly enhance the practical applicability of our flame recognition model in various applications, including fire detection, monitoring, and emergency response. Subsequent research endeavors might consider enlarging the dataset to encompass a broader spectrum of fire categories and environmental circumstances.

Author Contributions: L.Q. designed the comparative experiments, coded the software, and wrote the manuscript; W.Y. revised the manuscript; L.T. prepared data. All authors have read and agreed to the published version of the manuscript.

Funding: This research was Funded by the Sichuan Province Engineering Technology Research Center of Healthy Human Settlement, No. JKOP202302.

Data Availability Statement: The data used in this study are from open datasets. The datasets can be downloaded from <https://bitbucket.org/gbdi/bowfire-dataset/downloads/> (accessed on 20 January 2024).

Acknowledgments: We would like to thank the anonymous reviewers for their constructive and valuable suggestions on the earlier drafts of this manuscript.

Conflicts of Interest: Author Prof. Dr. L.T. was employed by the company Sichuan University Engineering Design & Research Institute Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest. The authors declare that there are no conflicts of interest.

References

- Diffenbaugh, N.S.; Konings, A.G.; Field, C.B. Atmospheric variability contributes to increasing wildfire weather but not as much as global warming. *Proc. Natl. Acad. Sci. USA* **2021**, *118*, e2117876118. [CrossRef] [PubMed]
- Peñuelas, J.; Sardans, J. Global Change and Forest Disturbances in the Mediterranean Basin: Breakthroughs, Knowledge Gaps, and Recommendations. *Forests* **2021**, *12*, 603. [CrossRef]
- Davide, A.; Jose, V.; Marco, M.; Lorenzo, S. Land use change towards forests and wooded land correlates with large and frequent wildfires in Italy. *Ann. Silv. Res.* **2021**, *46*, 177–188.
- Sadowska, B.; Grzegorz, Z.; Stepnicka, N. Forest Fires and Losses Caused by Fires—An Economic Approach. *WSEAS Trans. Environ. Dev.* **2021**, *17*, 181–191. [CrossRef]
- Overpeck, J.T.; Breshears, D.D. The growing challenge of vegetation change. *Science* **2021**, *372*, 786–787. [CrossRef] [PubMed]
- Shahid, M.; Chen, S.F.; Hsu, Y.L.; Chen, Y.Y.; Chen, Y.L.; Hua, K.L. Forest Fire Segmentation via Temporal Transformer from Aerial Images. *Forests* **2023**, *14*, 563. [CrossRef]
- Ferreira, L.M.; Coimbra, A.P.; de Almeida, A.T. Autonomous System for Wildfire and Forest Fire Early Detection and Control. *Inventions* **2020**, *5*, 41. [CrossRef]
- Resco de Dios, V.; Nolan, R.H. Some Challenges for Forest Fire Risk Predictions in the 21st Century. *Forests* **2021**, *12*, 469. [CrossRef]
- Kuang, H.L.; Chan, L.L.H.; Yan, H. Multi-class fruit detection based on multiple color channels. In Proceedings of the 2015 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR), Guangzhou, China, 12–15 July 2015. [CrossRef]
- Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
- Lienhart, R.; Maydt, J. An extended set of Haar-like features for rapid object detection. In Proceedings of the International Conference on Image Processing, Agadir, Morocco, 25–27 June 2003. [CrossRef]
- Suyarto, R.; Diara, I.; Susila, K.; Saifulloh, M.; Wiyanti, W.; Kusmiyarti, T.; Sunarta, I. Landslide inventory mapping derived from multispectral imagery by Support Vector Machine (SVM) algorithm. *IOP Conf. Ser. Earth Environ. Sci.* **2023**, *1190*, 012012. [CrossRef]
- Qiu, T.; Yan, Y.; Lu, G. An Autoadaptive Edge-Detection Algorithm for Flame and Fire Image Processing. *IEEE Trans. Instrum. Meas.* **2012**, *61*, 1486–1493. [CrossRef]
- Celik, T.; Demirel, H.; Ozkaramanli, H.; Uyguroglu, M. Fire detection using statistical color model in video sequences. *J. Vis. Commun. Image Represent.* **2007**, *18*, 176–185. [CrossRef]
- LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
- 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]
- Vasconcelos Reinolds de Sousa, J.; Vieira Gamboa, P. Aerial Forest Fire Detection and Monitoring Using a Small UAV. *Kne Eng.* **2020**, *5*, 242–256. [CrossRef]
- Sudhakar, S.; Vijayakumar, V.; Kumar, C.; Priya, V.; Ravi, L.; Subramaniaswamy, V. Unmanned Aerial Vehicle (UAV) based Forest Fire Detection and monitoring for reducing false alarms in forest-fires. *Comput. Commun.* **2020**, *149*, 1–16. [CrossRef]
- Chen, Y.; Zhang, Y.; Xin, J.; Yi, Y.; Liu, D.; Liu, H. A UAV-based forest fire-detection algorithm using convolutional neural network. In Proceedings of the 2018 37th Chinese Control Conference (CCC), Wuhan, China, 25–27 July 2018; pp. 10305–10310.
- Zhang, L.; Wang, M.; Fu, Y.; Ding, Y. A Forest Fire Recognition Method Using UAV Images Based on Transfer Learning. *Forests* **2022**, *13*, 975. [CrossRef]
- Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
- Vaswani, A. Attention is all you need. In Proceedings of the Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017.
- 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 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2021. [CrossRef]
- Jadon, A.; Omama, M.; Varshney, A.; Ansari, M.; Sharma, R. Firenet: A specialized lightweight fire smoke detection model for real-time iot applications. *arXiv* **2019**, arXiv:1909.07981.
- Raspberry pi 3 Model b. Available online: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/> (accessed on 14 March 2019).
- Musale, J. Survey on Forest Wildfire Detection Using Deep Learning. *Int. J. Sci. Res. Eng. Manag.* **2023**, *8*. [CrossRef]
- Zhang, Q.; Xu, J.; Xu, L.; Guo, H. Deep Convolutional Neural Networks for Forest Fire Detection. In Proceedings of the 2016 International Forum on Management, Education and Information Technology Application, Guangzhou, China, 30–31 January 2016. [CrossRef]
- Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Communications of the ACM* **2017**, *60*, 84–90. [CrossRef]
- Sharma, J.; Granmo, O.C.; Goodwin, M.; Fidge, J.T. Deep Convolutional Neural Networks for Fire Detection in Images. In *Engineering Applications of Neural Networks, Communications in Computer and Information Science*; Springer: Cham, Switzerland, 2017; pp. 183–193. [CrossRef]

30. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2015**, arXiv:1409.1556.
31. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016. [[CrossRef](#)]
32. Xue, Z.; Lin, H.; Wang, F. A Small Target Forest Fire Detection Model Based on YOLOv5 Improvement. *Forests* **2022**, *13*, 1332. [[CrossRef](#)]
33. Zhu, J.; Zhang, J.; Wang, Y.; Ge, Y.; Zhang, Z.; Zhang, S. Fire Detection in Ship Engine Rooms Based on Deep Learning. *Sensors* **2023**, *23*, 6552. [[CrossRef](#)] [[PubMed](#)]
34. Ann, H.; Koo, K.Y. Deep Learning Based Fire Risk Detection on Construction Sites. *Sensors* **2023**, *23*, 9095. [[CrossRef](#)]
35. Wang, G.; Wang, F.; Zhou, H.; Lin, H. Fire in focus: Advancing wildfire image segmentation by focusing on fire edges. *Forests* **2024**, *15*, 217. [[CrossRef](#)]
36. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
37. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
38. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv* **2017**, arXiv:1706.05587.
39. Shelhamer, E.; Long, J.; Darrell, T. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 640–651. [[CrossRef](#)] [[PubMed](#)]
40. Muhammad, K.; Ahmad, J.; Lv, Z.; Bellavista, P.; Yang, P.; Baik, S.W. Efficient Deep CNN-Based Fire Detection and Localization in Video Surveillance Applications. *IEEE Trans. Syst. Man, Cybern. Syst.* **2019**, *49*, 1419–1434. [[CrossRef](#)]
41. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the Lecture Notes in Computer Science, Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015, Munich, Germany, 5–9 October 2015; Springer: Cham, Switzerland, 2015; pp. 234–241. [[CrossRef](#)]
42. Yuan, W.; Wang, J.; Xu, W. Shift Pooling PSPNet: Rethinking PSPNet for Building Extraction in Remote Sensing Images from Entire Local Feature Pooling. *Remote Sens.* **2022**, *14*, 4889. [[CrossRef](#)]
43. Zhou, Z.; Rahman Siddiquee, M.M.; Tajbakhsh, N.; Liang, J. UNet++: A Nested U-Net Architecture for Medical Image Segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, Lecture Notes in Computer Science*; Springer: Cham, Switzerland, 2018; pp. 3–11. [[CrossRef](#)]
44. Qi, Y.; He, Y.; Qi, X.; Zhang, Y.; Yang, G. Dynamic Snake Convolution based on Topological Geometric Constraints for Tubular Structure Segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2–3 October 2023.
45. Chen, J.; Kao, S.h.; He, H.; Zhuo, W.; Wen, S.; Lee, C.H.; Chan, S.H. Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023.
46. Li, Y.; Yao, T.; Pan, Y.; Mei, T. Contextual Transformer Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 1489–1500. [[CrossRef](#)] [[PubMed](#)]
47. Chino, D.Y.T.; Avalhais, L.P.S.; Rodrigues, J.F.; Traina, A.J.M. BoWFire: Detection of Fire in Still Images by Integrating Pixel Color and Texture Analysis. In Proceedings of the 2015 28th SIBGRAPI Conference on Graphics, Patterns and Images, Salvador, Brazil, 26–29 August 2015; pp. 95–102. [[CrossRef](#)]
48. Toulouse, T.; Rossi, L.; Campana, A.; Celik, T.; Akhloufi, M. Computer vision for wildfire research: An evolving image dataset for processing and analysis. *Fire Saf. J.* **2017**, *92*, 188–194. [[CrossRef](#)]

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