mathematics

MDPI

Article

Automatic Recognition of Indoor Fire and Combustible Material with Material-Auxiliary Fire Dataset

Feifei Hou, Wenqing Zhao and Xinyu Fan *

School of Automation, Central South University, Changsha 410083, China; houfeifei@csu.edu.cn (F.H.); csuzwq@csu.edu.cn (W.Z.)
* Correspondence: auxyfan@csu.edu.cn

Abstract: Early and timely fire detection within enclosed spaces notably diminishes the response time for emergency aid. Previous methods have mostly focused on singularly detecting either fire or combustible materials, rarely integrating both aspects, leading to a lack of a comprehensive understanding of indoor fire scenarios. Moreover, traditional fire load assessment methods such as empirical formula-based assessment are time-consuming and face challenges in diverse scenarios. In this paper, we collected a novel dataset of fire and materials, the Material-Auxiliary Fire Dataset (MAFD), and combined this dataset with deep learning to achieve both fire and material recognition and segmentation in the indoor scene. A sophisticated deep learning model, Dual Attention Network (DANet), was specifically designed for image semantic segmentation to recognize fire and combustible material. The experimental analysis of our MAFD database demonstrated that our approach achieved an accuracy of 84.26% and outperformed the prevalent methods (e.g., PSPNet, CCNet, FCN, ISANet, OCRNet), making a significant contribution to fire safety technology and enhancing the capacity to identify potential hazards indoors.

Keywords: fire detection; combustible material recognition; deep learning; indoor fire scene; semantic segmentation

MSC: 68T45

Citation: Hou, F.; Zhao, W.; Fan, X. Automatic Recognition of Indoor Fire and Combustible Material with Material-Auxiliary Fire Dataset. *Mathematics* **2024**, *12*, 54. https://doi.org/10.3390/math12010054

Academic Editors: Jesús García-Herrero and Johan Debayle

Received: 15 November 2023
Revised: 14 December 2023
Accepted: 21 December 2023
Published: 23 December 2023

1. Introduction

The trend of urbanization has resulted in a growing population residing and working within large buildings [1]. Due to their densely populated and complex structures, these large buildings harbor numerous fire hazards. The most common sources of indoor fire are faulty appliances or equipment, aging electrical systems, careless disposal of cigarettes or matches, gas leaks, improper handling or storage of flammable liquids, and deliberate arson [2]. Indoor fires can spread rapidly and produce toxic smoke and gases, which can impair the visibility and health of the occupants and the firefighters. Indoor fires can also damage the structural integrity of the building and cause collapse or partial failure [3]. Therefore, substantial casualties and financial losses can result from indoor building fires [4]. Based on this, indoor fire detection has become increasingly challenging and plays a crucial aspect in effectively managing disasters [5].

One of the keys to indoor fire detection is to identify the combustible materials that are involved in the fire. Many combustible materials used indoors are one of the main causes of fires, and common flammable items such as sofas, mattresses, curtains, and wooden furniture can easily ignite and cause large-scale fires. After an indoor fire occurs, it is initially limited to the combustion of combustible materials at the ignition point, then spreads to adjacent rooms or areas as well as the entire floor, and finally spreads to the entire building. The degree of indoor fire spread is related to factors such as the combustion performance of indoor materials, substances, and the quantity of combustible materials.

Therefore, automating the assessment and identification of combustible materials in indoor fire scenarios becomes imperative to prevent fires and minimize property losses as much as possible. Firefighters can make an objective and rapid judgment of the fire situation, which is convenient for firefighters to rescue trapped personnel and will also play an important guiding role in fire prevention and control work.

There are three challenges in automating this process. The first challenge lies in the complexity of indoor scene images, surpassing that of outdoor images due to intricate backgrounds, diverse interior decorations, severe occlusions, variations in perspectives, etc. Therefore, the complex background will interfere with our fire detection task. The second challenge is that once a fire occurs, all factors such as the source, size, and degree of combustion of the fire will jointly determine the combustion level of indoor materials, leading to inconsistent damage levels, which will affect the classification and recognition of combustible materials in the future. Therefore, it is necessary to consider all constraints imposed by disaster management scenarios. The third challenge is that the benchmark fire datasets are not available, and the fire scene is severely damaged and chaotic, making it difficult to collect fire data. Moreover, the unavailability of on-site images due to privacy concerns often restricts further research in this field.

To address these challenges effectively, we propose this innovative research utilizing efficient CNNs for precise segmentation of both fires and combustible materials within real-world fire scenarios. Specifically, a novel fire-material recognition framework is proposed to revolutionize the current state of fire detection and rescue. Figure 1 presents the proposed framework. This work contributes in three main aspects:

(1) We present an efficient deep learning semantic segmentation framework based on a dual attention mechanism, which involves position attention and channel attention and assigns pixels with object class and attribute labels.
(2) We first simultaneously estimate the fire object and fire load in indoor scenes and explore a multi-task learning strategy to learn the correlations between fire burning degree and combustible material statistics. The segmentation accuracy levels of fire and combustible material can be significantly enhanced for detailed scene analysis.
(3) We introduce and collect a new database, the Material-Auxiliary Fire Dataset (MAFD), with attribute labels for combustible material and class labels for fire objects, which provides a benchmark to encourage automatic applications in indoor fire scenes.

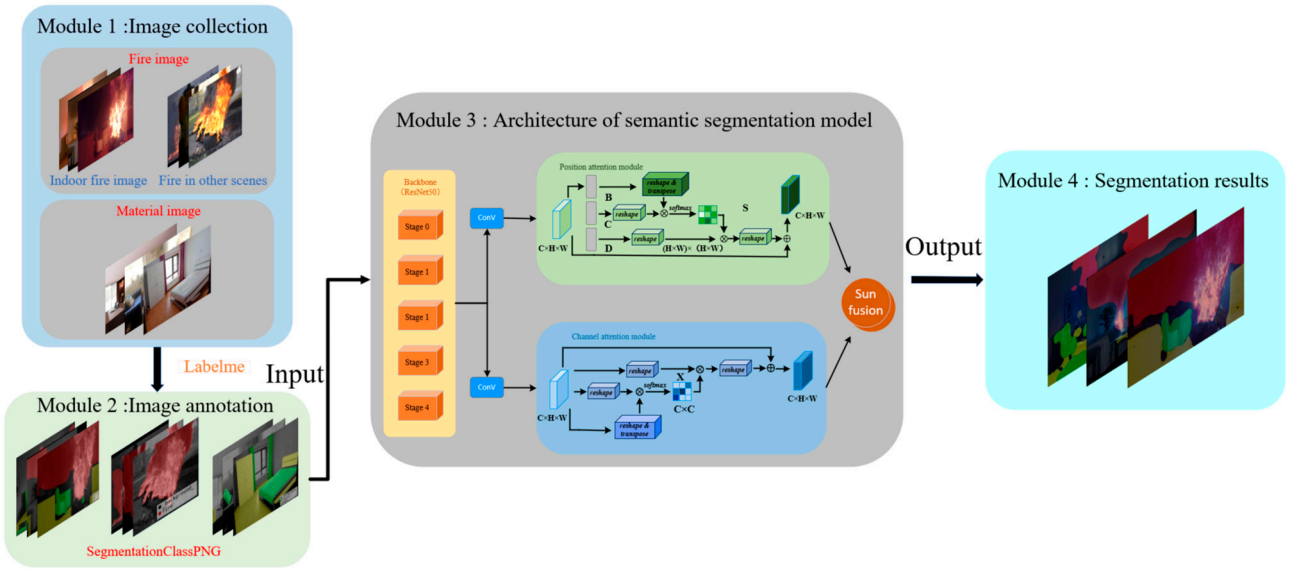


Figure 1. Indoor fire and combustible material recognition framework.

The subsequent sections of this paper are organized as follows. Section 2 provides an overview of the literature concerning fire detection and the recognition of material. Our

deep learning framework for fire and combustible material segmentation within indoor scenes is detailed in Section 3. Section 4 introduces the development of our dataset, the MAFD, and presents extensive experiments with it. Section 5 concludes the paper by summarizing key points and proposing future research directions.

2. Literature Review

With the continuous advancement of technology, there is growing attention toward developing efficient and reliable methods to identify fire and smoke. Numerous comprehensive reviews and surveys have been conducted within the realm of fire and smoke detection. Among them, the methods utilized can be categorized into two main groups: traditional methods and deep learning-based approaches.

Traditional methods are usually based on image processing algorithms such as edge detection, morphological processing, and threshold segmentation. Wu et al. [6] used camera sensors for fire smoke detection, extracting static and dynamic features, and achieving strong results with AdaBoost. Russo et al. [7] proposed a method for the smoke detection of surveillance cameras based on local binary pattern (LBP) and support vector machine (SVM). Wang et al. [8] proposed a rapid smoke detection method using slope fitting in video image histogram, addressing false alarms in early fire smoke detection. Cao et al. [9] proposed patchwise dictionary learning within the wavelet domain to detect smoke in forest fire videos. Their method aims to distinguish fire smoke from other objects in the forest that share a similar visual grayscale appearance. Fire smoke can be distinguished from other challenging objects in the forest with a similar visual grayscale appearance. Gagliardi et al. [10] introduced video-based smoke detection technology using techniques like the Kalman estimator, blob labeling, and decision-making processes. Hossain et al. [11] introduced a novel technique for forest fire detection that relied on fire-specific color features and the multi-color space local binary pattern to identify distinct attributes of flames and smoke. They also employed support vector machines as classifiers. Their results showed that this method has a higher performance compared to other color or texture-based methods. However, these traditional methods typically require manual parameter selection and adjustment and have poor adaptability to various smoke densities and color variations, which also suffer from several drawbacks including a high rate of false alarms (FAR), restricted accuracy, and a reduced detection range.

In contrast, deep learning methods, by utilizing learned features to identify and segment fire and smoke patterns and adapting to various smoke conditions, have introduced a novel research avenue for addressing early fire detection challenges. Jia et al. [12] utilized domain knowledge and transfer learning from deep convolutional neural networks (CNN) for video smoke detection and reduced the false positive rate of the video smoke detection (VSD) systems to some extent. However, low-level features were not utilized. Peng et al. [13] combined manually crafted features with deep learning features. They utilized an algorithm designed manually to extract areas suspected to contain smoke, which were then processed using an enhanced SqueezeNet deep neural network for smoke detection. Cheng et al. [14] employed Deeplabv3+ and conditional random fields for accurate segmentation, established smoke thickness heatmaps and predicted smoke trends with generative adversarial networks, contributing to fire protection and evacuation planning. To address issues in video-based smoke detection, Yuan et al. [15] introduced a deep smoke segmentation network designed to derive precise segmentation masks from unclear smoke images. Lin et al. [16] devised an integrated detection framework by combining a faster Region-CNN (RCNN) and 3D CNN, enhancing video smoke detection by maximizing the utilization of temporal information within video sequences. Li et al. [17] introduced an adaptive linear feature-reuse network (ALFRNet) for rapid forest fire smoke detection, effectively reducing information loss and interference caused by image blurring during the smoke image acquisition process. Liu et al. [18] introduced a smoke detection approach using an ensemble of simple deep CNNs by capturing diverse smoke aspects and aggregating subnetwork responses via majority voting, outperforming existing methods on newly

established noisy smoke image datasets. To meet the needs of complex aerial forest fire smoke detection tasks, Zhan et al. [19] proposed an adjacent layer composite network based on a recursive feature pyramid with deconvolution and dilated convolution and global optimal non-maximum suppression (ARGNet) for the high-accuracy detection of forest fire smoke. Hu et al. [20] proposed a novel method for early forest fire smoke detection called multi-oriented detection. This method integrated a value conversion-attention mechanism module and Mixed-Non-Maximum Suppression (Mixed-NMS) to overcome common misdetection and missed detection issues, elevating target detection accuracy. To change the fact that the majority of current computer vision-based fire detection methods can only identify either flames or smoke, Hosseini et al. [21] introduced a unified flame and smoke detection method, named UFS-Net, which can identify potential fire risks by categorizing video frames into eight distinct classes. Khan et al. [22] proposed an energy-efficient system based on VGG-16 architecture for early smoke detection in both normal and foggy IoT environments. He et al. [23] also proposed a method targeting foggy environments that combined attention mechanisms and feature-level and decision-level fusion modules. From various perspectives including overall, individual categories, small smoke, and challenging negative sample detection, their approach achieved higher detection accuracy, precision, recall, and F1 scores. To meet the requirements of smoke detection within an industrial environment, Muhammad et al. [24] proposed an energy-friendly edge intelligence-assisted method for smoke detection in foggy surveillance environments using deep CNN.

The concept of fire load is of utmost importance in fire safety and building resilience. Many combustible materials used indoors are one of the main causes of fires. Numerous studies have focused on material recognition. Strese et al. [25] proposed a tool-mediated surface classification method. This method combines the extracted feature information such as sound, image, friction, and acceleration with a naive Bayesian classifier to identify different materials. Zhang et al. [26] proposed a novel hierarchical multi-feature fusion (HMF2) model, aiming to gather essential information and employ a classifier for training a novel material recognition model. They tested the simplicity, effectiveness, robustness, and efficiency of the HMF2 model on two benchmark datasets. Lee et al. [27] proposed a material-type identification method using a deep CNN based on color and reflectance features. The proposed method was evaluated on public datasets, showing promising results for material type identification.

Although researchers have conducted extensive algorithmic research in the field of material recognition and have high-quality public datasets, there is currently no algorithmic research for complex indoor fire scenarios, nor is there a relevant public dataset. In addition, the main limitation of these methods is the lack of the simultaneous evaluation of fire objects and fire loads. Fortunately, our work has solved these problems. Specifically, Section 3 provides details of the proposed methodology, while Section 4 discusses the experimental validation.

3. Dual Attention Fire Recognition Methodology

3.1. Architecture of Semantic Segmentation Model

The Dual Attention Network (DANet) model [28] was adopted as our semantic segmentation model, and its overall architecture is depicted in Figure 2. Different from previous approaches that utilize multi-scale feature fusion to capture context, DANet addresses scene segmentation by a self-attention mechanism. This mechanism efficiently integrates intricate contextual dependencies, allowing the adaptive fusion of dispersed features along with their global dependencies. In order to improve the model accuracy, two types of attention modules were attached on top of the dilated FCN (Fully Convolutional Network), which simulated semantic interdependence in both the spatial and channel domains. The position attention module selectively aggregates the features of each position through a weighted sum of each position, and similar features will be related to each other regardless of the distance between them. At the same time, the channel attention module selectively emphasizes interdependent channel maps by integrating the relevant features

across the entirety of channel maps. Finally, the outputs of these two attention modules are summed to further improve the feature representation, which helps to obtain more accurate segmentation results. During the aggregation process of the two modules, according to the vector product theory, when the product of two vectors is larger, it means that the angle between the vectors is smaller, and the correlation between the two vectors is stronger. Detailed descriptions of the two modules are presented in the subsequent section.

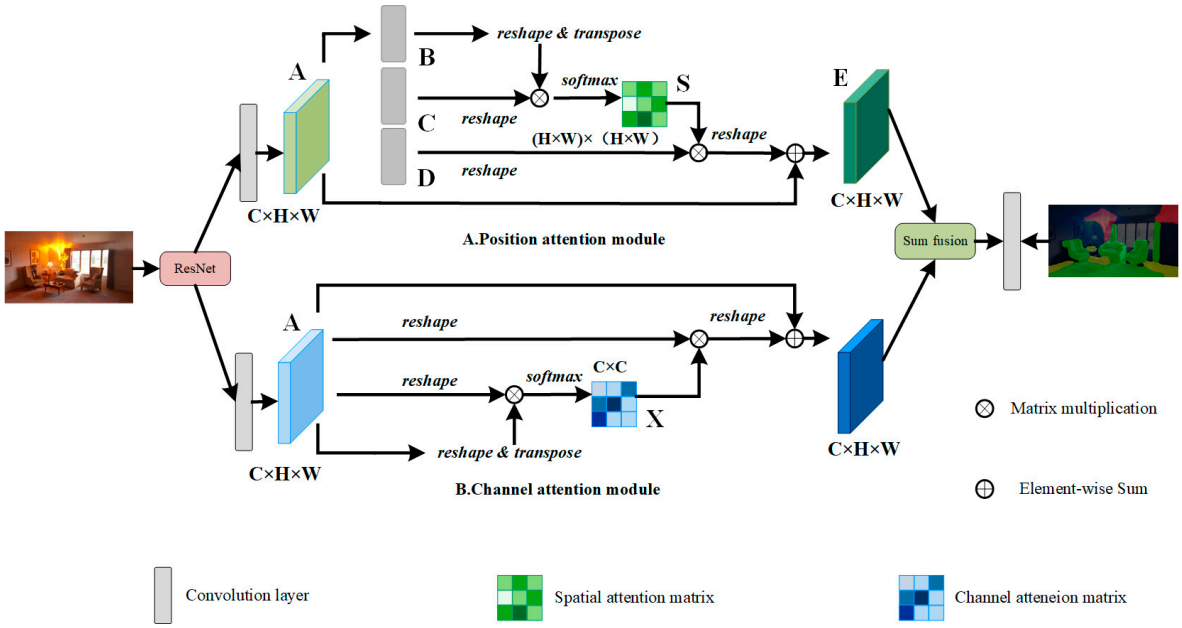


Figure 2. Dual Attention Network (DANet) framework with vector output distribution at each stage.

3.2. Attention Modules for Feature Representation

3.2.1. Position Attention Module

The input of the position attention module is a feature map A, expressed as $C \times H \times W$, where C represents the number of channels, while H represents the height of the feature map, and W represents the width of the feature map. The specific working principle of this module is shown in Figure 2, and the vector size obtained at each stage is also marked in Figure 2. Since the focus of the position attention module is to mine the similarity relationship between each pixel, in order to better use the attention module, the feature maps B, C, and D are reshaped to obtain three sizes of $C \times H$, where $N = H \times W$. These three matrices correspond to Q, K, and V of the self-attention mechanism. Each step is described in detail below:

(1) Calculating the similarity matrix between pixels, the process is to obtain a similarity matrix between pixels with a size of $N \times N$ through $QT \times K$, that is, the $(N \times C)$ matrix multiplied by the $(C \times N)$ matrix;
(2) Perform a softmax operation on the similarity matrix to obtain each relative factor that affects the pixel;
(3) Multiply the similarity matrix S after softmax with the V matrix, that is, multiply the $(C \times N)$ matrix by the $(N \times N)$ matrix, and finally obtain the recoded feature representation, and its size is also $C \times N$, where the generation formula of S is shown in Equation (1). The purpose of multiplying the original matrix by the similarity matrix is to amplify the influence of pixels that are similar to it and reduce the influence of pixels that are not similar to it, which can also be called a re-encoding operation;
(4) Perform the reshape operation on the finally obtained new feature matrix to obtain a recoded feature map with a size of $C \times H \times W$;
(5) Add the feature map to the features extracted from the upper network to obtain the output E of the final position attention module, whose size is still $C \times H \times W$, where

the generation formula of E is shown in Equation (2). The scaling factor α initially begins at 0 and gradually adjusts to attain higher weights.

$$S_{ji} = \frac{exp(B_i C_j)}{\sum_{i=1}^{N} exp(B_i C_j)} \tag{1}$$

$$E_j = \alpha \sum_{i=1}^{N} (s_{ji} D_i) + A_j \tag{2}$$

3.2.2. Channel Attention Module

The channel attention module is used to mine the similarity relationship between each channel in the image feature map, so that each channel has global semantic features. The input is also a feature map A, whose size is 1/8 of the original image. The specific process of the channel attention mechanism is displayed in Figure 2, and the specific process is as follows:

(1) Calculating the similarity matrix between pixels, the process is to obtain a similarity matrix between pixels with a size of $N \times N$ through $QT \times K$, that is, multiplying the $(C \times N)$ matrix by the $(N \times C)$ matrix;
(2) Perform a softmax operation on the similarity matrix to obtain each relative factor affecting the channel;
(3) Multiply the similarity matrix X and the V matrix after softmax, that is, the $(N \times C)$ matrix multiplied by the $(C \times C)$ matrix, and finally obtain the recoded feature representation, and its size is also $C \times N$, where the generation formula of X is shown in Equation (3). The purpose of multiplying the original matrix by the similarity matrix is to amplify the influence of similar channels and reduce the influence of dissimilar channels;
(4) Perform the reshape operation on the finally obtained new feature matrix to obtain a recoded feature map with a size of $C \times H \times W$;
(5) Add the feature map to the features extracted from the upper network to obtain the output E of the final channel attention module, whose size is still $C \times H \times W$, where the generation formula of E is shown in Equation (4). The initial value of the scaling factor β is set to 0 and incrementally adapts to gain higher weights.

$$x_{ji} = \frac{exp(A_i \cdot A_j)}{\sum_{i=1}^{C} exp(A_i \cdot A_j)} \tag{3}$$

$$E_j = \beta \sum_{i=1}^{C} (x_{ji} A_i) + A_j \tag{4}$$

3.3. BaseNet Selection

The DANet model is based on dilated-ResNet (Residual Network) [29] as the backbone network to extract features. Two CNN backbones were chosen as a basis for assessing the accuracy and time efficiency (denoted by network-depth-feature), forming two DANets: dilated-ResNet50 and dilated-ResNet101. ResNet and dilated-ResNet are state-of-the-art CNN models and are widely used backbones in semantic segmentation models, where dilated ResNet has some advantages over ResNet by introducing the dilated convolution to improve the resolution of the feature map. Typically, a deeper model is anticipated to achieve higher accuracy compared to a shallow model, while at the expense of increased computational time. The dilated-ResNet50 model architecture is shown in Figure 3. After the backbone network applies dilated convolution to replace the downsampling in the original model, the feature map of the image is obtained, and its size is 1/8 of the original image. Then, the feature map is input into two attention modules in parallel to obtain the

global features between pixels and the global features between channels, respectively, and finally integrate the output of the two attention modules to obtain a better expression.

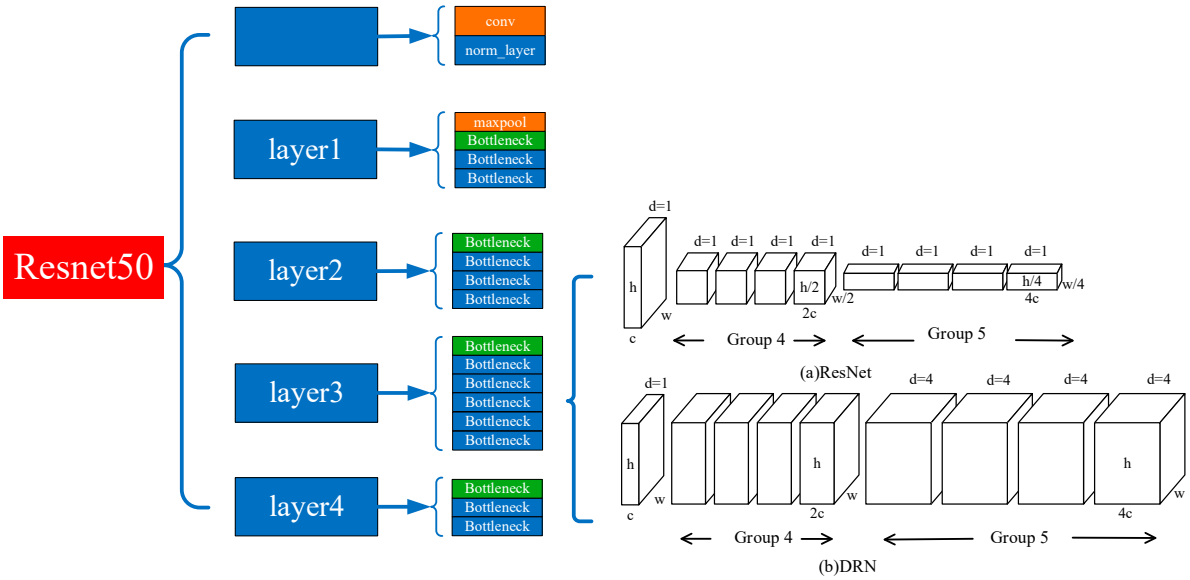


Figure 3. Dilated ResNet50 backbone.

Transfer learning aims to utilize previously acquired knowledge to efficiently solve new but similar problems. Unlike traditional machine learning methods, it capitalizes on knowledge gathered from auxiliary domains' data to enhance predictive modeling for disparate data patterns within the present domain. The fundamental idea of transfer learning is to extract the knowledge from a previous or source task and apply the extracted knowledge to a new/target task. A conceptual metaphor is that it will be easier for a child to learn how to recognize peaches if they have already learned how to recognize apples and pears.

We employed the transfer learning to reduce the training difficulty for our relatively small dataset as well as enhance performance. Transfer learning shows promise in minimizing the dependence on a large number of target domain data by transferring knowledge from diverse yet related source domains [30]. The deep learning model was first pretrained on a large general dataset VOC, and then trained and tested on our dataset.

4. Experiments and Results

4.1. Experimental Settings and Evaluation Metrics

The implementation was carried out in a Python environment (version 3.8.10, provided by the Python Software Foundation, Wilmington, DE, USA) using the PyTorch deep learning package (PyTorch:1.10.0 + cu111, TorchVision: 0.11.0 + cu111) and a single NVIDIA GeForce RTX 3060 GPU. MMSegmentation [31] is an open-source PyTorch-based toolbox specifically designed for semantic segmentation tasks. It decomposes the semantic segmentation framework into different components. By combining different modules, a customized semantic segmentation framework can be easily built. The toolbox provides direct support for prevalent and contemporary semantic segmentation frameworks, providing pre-trained semantic segmentation models on various mainstream datasets. In this paper, the semantic segmentation framework provided in MMSegmentation was utilized for both model training and verification, employing mmcv version 2.0.0rc4 and MMSegmentation version 1.1.2. DANet employs ResNet as the model backbone. The images in our dataset were randomly divided into training (90%) and validation (10%) sets with a 9:1 split ratio. The model was trained on the training dataset and tested on the validation dataset.

During training, images were resized to 512 × 512 pixels for input. The optimizer for the three models was stochastic gradient descent (SGD) with a learning rate of

5×10^{-4}, a momentum of 0.975 for L2 regularization, and a weight decay factor of 0.0004. SGD was selected instead of adaptive optimization methods (e.g., AdaGrad, RMSProp, or Adam) due to its potential to achieve a higher test accuracy, converge toward a flatter minimum, and consequently yield improved generalization [32]. The decode head predicts the segmentation map from the feature map using the decoding head of DAHead, and the loss function uses CrossEntropyLoss, and the loss weight is 0.3. Auxiliary_head encourages the backbone network to learn lower-level features that are not used for prediction. The decoding head of FCNHead is used. The loss function uses CrossEntropyLoss and the loss weight is 0.15. Data augmentation used during training included horizontal flipping and random cropping. The training used the PolyLR scheduler, which reduces the learning rate according to a polynomial function, the minimum learning rate is 1×10^{-6}, and is scheduled according to each iteration. The maximum number of iterations of the training loop was 40,000, and there was an interval of 10,000 loops for verification. Table 1 displays the experimental settings.

Table 1. Experimental settings.

Setting	Value
Batch size	1
Crop size	512×512
Momentum	0.975
Initial learning rate	0.0005
Weight decay	0.0004

The *Acc* (accuracy) [31] refers to the proportion of accurately classified pixels to the total number of pixels in the segmentation result. In image segmentation, we usually compare the predicted label for each pixel with the true label, and then calculate the accuracy rate between them. Specifically, we can count the number of pixels in the segmentation result that are the same as the real result, and then divide them by the total number of pixels to obtain the segmentation accuracy. The higher the *Acc*, the more pixels are correctly classified in the segmentation result, and the better the segmentation performance. The calculation formula of *Acc* is as follows:

$$Acc = \frac{\sum_{i=0}^{k} p_{ii}}{\sum_{i=0}^{k} \sum_{j=0}^{k} p_{ij}} \tag{5}$$

Among them, i represents the real value, j represents the predicted value, P_{ij} represents the number of pixels that predict i as j, and k represents the total number of categories.

The *mIoU* (mean intersection over union) refers to the average value of the intersection and union ratios between the segmentation results and the real segmentation results. In image segmentation, we usually compare the predicted value with the ground truth for each class, and then calculate the *IoU* between them. Specifically, for each category, we can count the number of pixels in the segmentation result and the ground truth result, and then calculate the intersection and union between them. We can then divide the intersection by the union to obtain the *IoU* for that category. Finally, we can average the *IoU* of all categories to obtain the *mIoU* of the whole image. The higher the *mIoU*, the closer the segmentation result is to the real result, and the better the segmentation performance. The calculation formula of *mIoU* is as follows:

$$mIoU = \frac{1}{k+1} \sum_{i=0}^{k} \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}} \tag{6}$$

Among them, i represents the real value, j represents the predicted value, P_{ij} represents the number of pixels that predict i as j, and k represents the total number of categories.

The *mAcc* (mean accuracy) refers to the average of the accuracy of each category. In image segmentation, we usually compare the predicted value of each class with the true value, and then calculate the accuracy between them. Specifically, for each category, we can count the number of pixels that are correctly classified in the segmentation result and the ground truth result, and count the number of pixels of that category in the ground truth result. We can then divide the number of correctly classified pixels by the number of pixels for that class to obtain the accuracy for that class. Finally, we can average the accuracies across all classes to obtain the mAcc for the entire image. The higher the *mAcc* means that each category is better recognized and distinguished, and the segmentation performance is better. The calculation formula of *mAcc* is as follows:

$$mAcc = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k} p_{ij}} \tag{7}$$

Among them, i represents the real value, j represents the predicted value, P_{ij} represents the number of pixels that predict i as j, and k represents the total number of categories.

4.2. Material-Auxiliary Fire Dataset

In this paper, we collected and developed a dataset containing indoor fire images and material segmentation annotations, named 'MAFD'. Indoor scenes typically contain a variety of materials with complicated layout relationships. Therefore, two material categories were identified as labels to represent each indoor object based on their primary composition. The selected categories included: (1) fabric and (2) wood. These materials are commonly used and are usually the main components of indoor scenes. They are combustible and also exhibit varying degrees of flammability. In addition, the dataset also included the fire category, which includes flames and smoke. Therefore, our dataset involves a total of three categories.

A total of 3899 images were collected from three parts. (1) A part of these images was gathered from the dataset used in Deep Learning-Based Instance Segmentation for Indoor Fire Load Recognition [33], available at https://github.com/Zhou-Yucheng/fire-load-detection (accessed on 5 November 2022). The dataset images contain at least one combustible object and have an image of appropriate resolution. There were 1015 images in total, distributed in five classes: bedroom, dining room, hospital, living room, and office. (2) The second part of these images was collected from online sources and other public datasets. This part encompasses outdoor scenes with fires including a variety of settings like buildings, streets, vehicles, forests, and farmland. We specifically targeted images with clearly distinguishable fire visual attributes and a rich variety of environmental features to make our dataset more representative. There was a total of 2466 images. (3) The rest were sourced from the Kaggle website (https://www.kaggle.com/ (accessed on 5 November 2022)), which is a platform for organizing machine learning competitions and hosting databases. Specifically, our selection focused on indoor images featuring both combustible materials and fire instances. Datasets need to be large enough to provide a balanced sample type, wide diversity, and multiple categories. This collection contains a variety of individual images from various indoor scenarios such as kitchen, factory, and living room. Finally, a total of 418 images were selected. These images were manually annotated into different classes based on their content, and 11,320 instance annotations were obtained. Table 2 details the number of annotations of each category in our MAFD.

Table 2. Number of annotated instances per category.

Category	Number of Instances
Fire	3071
Fabric	4055
Wood	4195

The type of annotation depends largely on the task attribute. Our task was to segment fire and material instances to obtain their complete boundary information. Therefore, in this paper, we focused on per-pixel segmentation and labeling for the fire category and three kinds of materials. Our MAFD contains polygon annotations that enclose same-material regions. Each style of annotation comes with a cost proportional to its complexity. Our images were annotated using the LabelMe [34] tool. The annotation results were saved in the JSON file, and then the format conversion toolkit provided by LabelMe was used to convert the JSON file into the standard Pascal VOC (Pattern Analysis Statistical Modeling and Computational Learning, Visual Object Classes) [35] format. Pascal VOC is a standardized dataset used for object detection, semantic segmentation, and more. Its annotations are stored in the form of XML files, each corresponding to an image, containing information about all objects in the image including their location and category. Sample images with annotated categories are visualized in Figure 4, where each labeled instance is indicated by a colored mask with its category name located in the bottom-right corner.

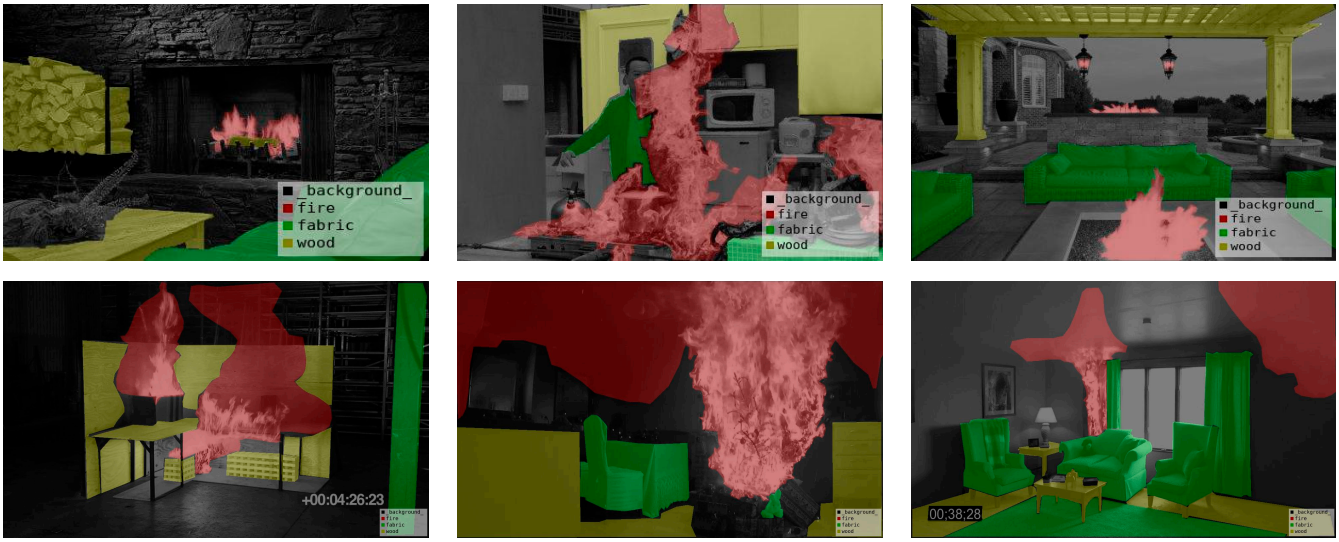


Figure 4. Sample images with annotated category and name.

4.3. Experiment 1: Selection of Optimal Model

We used DANet-50 with ResNet-50 as the backbone network and DANet-101 with ResNet-101 as the backbone network to train on our dataset, and compared the results when the training iterations were 20 k and 40 k. Although the mAcc of DANet-50–40 k was slightly higher than DANet-101, the average accuracy of DANet-101 was higher and the recognition accuracy in different categories was average, so we chose DANet-101 as the training model in this paper. Table 3 displays the aAcc, mIoU, and mAcc percentages of the DANet-50 and DANet-101 models after training for 20 k and 40 k iterations.

Table 3. The aAcc, mIoU, and mAcc (%) of the DANet-50 and DANet-101 models trained over 20 k and 40 k iterations.

Method	Total Number of Training Iterations	aAcc	mIoU	mAcc
DANet-50	20 k	82.15	59.46	71.27
DANet-50	40 k	81.84	60.63	73.98
DANet-101	20 k	82.99	60.95	72.25
DANet-101	40 k	83.19	61.73	73.33

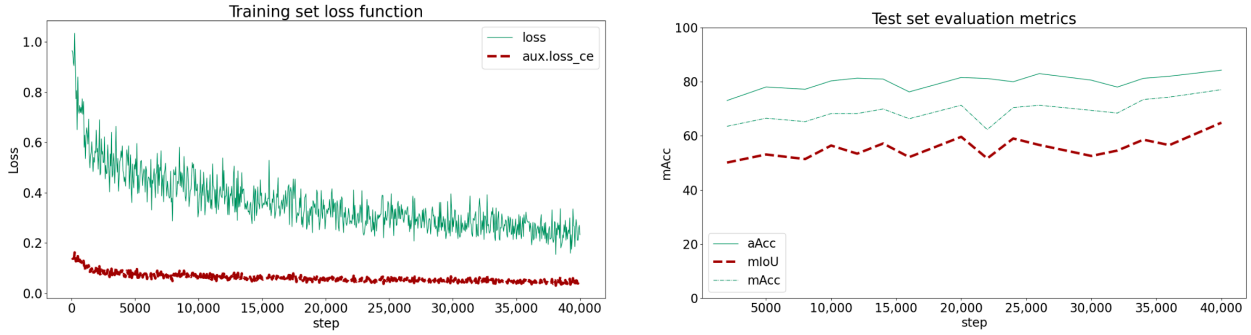
We then further optimized the parameters of the selected DANet-101 model and added training iterations. It was found that all indicators improved at first when the training iterations were set to 100 k. When the training iterations were set to 100 k, the trained

model performed best and its performance in the fire, wood, and fabric categories was relatively average. When the training iterations continued to increase, the aAcc, mIOU, and mAcc all decreased. Therefore, the corresponding model parameters when the training iterations were 100 k were finally selected. Table 4 illustrates how the aAcc, mIoU, and mAcc changed for the DANet-101 model across diverse training iterations.

Table 4. The aAcc, mIoU, and mAcc (%) of DANet-101 using the training iterations of 20 k, 40 k, 60 k, 80 k, 100 k, and 120 k.

Method	Total Number of Training Iterations	aAcc	mIoU	mAcc
DANet-101	20 k	82.99	60.95	72.25
DANet-101	40 k	83.19	61.73	73.33
DANet-101	60 k	82.50	61.11	73.73
DANet-101	80 k	82.71	61.59	73.74
DANet-101	100 k	84.26	64.85	77.05
DANet-101	120 k	83.04	60.03	70.53

Figure 5 showcases the performance curves over 100 k iterations. The loss function represents the difference between the predicted output and the actual target. As the training steps increase, the changes in the loss function will display the model's degree of fit to the training data during the training period as well as the optimization effect of the model. From Figure 5a, it can be observed that with an increase in training steps, the value of 'loss' gradually decreased to around 0.2, and 'aux.loss_ce' (cross-entropy loss) decreased to around 0.05. This indicates that the model progressively achieved a better fit to the training data. Figure 5b shows the changes in the test set classification evaluation indicators (mIoU, mAcc, and aAcc) with the step size. When the step reached 40,000, the aAcc, mIoU, and mAcc reached their maximum values of 84.26%, 64.85%, and 77.05%, respectively.



(**a**) (**b**)

Figure 5. Performance curves of 100 k iterations: (**a**) the variation of training set loss function with step size and (**b**) the changes in the test set classification evaluation indicators (mIoU, mAcc, and aAcc) with step size.

4.4. Experiment 2: Visualization Results of the Proposed Model

To verify the effectiveness of the proposed model, we provided visual results in various indoor scenes such as a living room, kitchen, restaurant, and office using the MAFD. As shown in Figure 6, the proposed scheme was able to segment fire objects well without any post-processing. The first and third rows in Figure 6 show the input fire images, while the corresponding output results segmented by the proposed model are shown in the second and fourth rows in Figure 6, where the fire objects are represented by red masks, wood objects are represented by a yellow mask, fabric objects are represented by a green mask, and other areas are darker. It can be seen that the scene in Figure 6a is the living room. The sofa in the living room is made of a flammable fabric material, and the tables and chairs are made of a flammable wood material. This complex indoor environment is a

highly flammable place. Figure 6c shows the kitchen scene, and the kitchen stove is also a dangerous place because of its obvious fire source. The fires in Figure 6b (restaurant) and Figure 6d (office) were intense and accompanied by a lot of smoke. The above visualization results can mark flame, fabric, and wood in various scenes in the MAFD, which verifies that the scheme can realize the segmentation of flame, fabric, and wood in indoor fire scenes. Therefore, it is feasible to use the proposed scheme to identify fire, fabric, and wood.

(**a**) Living room scene (**b**) Restaurant scene

(**c**) Kitchen scene (**d**) Office scene

Figure 6. Examples of the model prediction output.

4.5. Experiment 3: Comparison with State-of-the-Art Methods

To comprehensively demonstrate the performance of our proposed method, we conducted an evaluation comparing it with established state-of-the-art models such as PSPNet [36] (Pyramid Scene Parsing Network), CCNet [37] (Criss-Cross Attention Network), FCN [38], ISANet [39] (Interlaced Sparse Self-Attention Network), and OCRNet [40] (Object-Contextual Representations Network). They are all widely recognized models in the field of semantic segmentation. The basic experimental setup for other models remained consistent with our approach and all experiments were conducted exclusively on our MAFD.

For comparison, we selected fire and one material, fabric, to represent the evaluation. We first analyzed the comparative results based on aAcc, Acc.fire, and Acc.fabric. The results indicate that OCRNet performed relatively poorly in all of the evaluation metrics. Our proposed method exhibited the highest aAcc and Acc.fire. Although Acc.fabric was slightly lower by 0.94% compared to ISANet, considering the overall precision, the comprehensive performance of our method surpassed that of the existing models. Figure 7 visually displays the comparison results of accuracy (%) across different models, providing clear evidence of our model's performance.

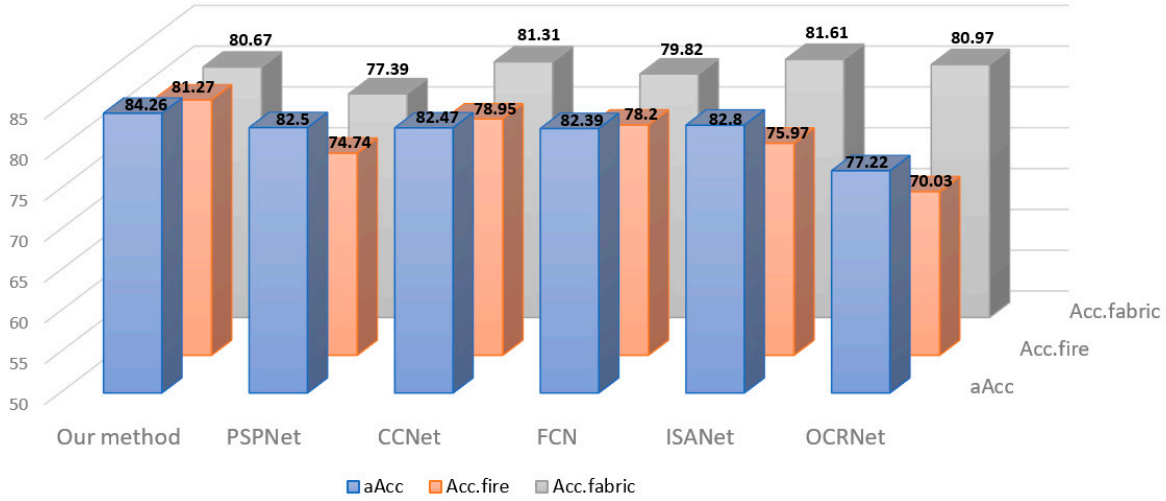


Figure 7. Comparison results of accuracy (%) across different models. (aAcc, Acc.fire, and Acc.fabric).

Table 5 shows an overall comparison of the mIoU, IoU.background, IoU.fire, and IoU.fabric across various models. Notably, OCRNet showed the poorest performance on the MAFD. In contrast, our proposed method outperformed other models across all evaluation metrics. Comparatively, it is reasonable to assume that the proposed method exhibits superior overall performance and higher IoU values among all existing models.

Table 5. Comparison results of IoU (%) across different models.

Model	mIoU	IoU.background	IoU.fire	IoU.fabric
PSPNet	60.20	78.07	65.73	62.54
CCNet	60.42	77.86	67.30	62.90
FCN	61.45	77.35	65.02	61.7
ISANet	61.37	78.18	65.42	65.33
OCRNet	53.48	73.48	63.96	39.14
The proposed method	64.85	79.43	70.61	64.53

To highlight the superiority of our method over state-of-the-art semantic segmentation models, the representative visual results obtained from our comparative experiments are depicted in Figure 8. In Figure 8a, we present the input images with fire and combustible materials. Figure 8b–g exhibit the segmentation results produced by PSPNet, CCNet, FCN, ISANet, OCRNet, and our proposed method, respectively.

From the images in the first column, it is evident that our method exhibited fewer incorrect pixels compared to other models and accurately identified both clothing and fire within the images. In the second column, concerning Figure 8b,c,e, our method demonstrated better recognition of smoke and some minor wooden items. Likewise, in the third column, our method performed exceptionally well in identification, even for minute objects.

The above comparison demonstrates the superior accuracy of our model compared to others when simultaneously detecting combustible materials and fire. Additionally, it exhibits exceptional performance in identifying even the most minor objects, highlighting the robustness of our method.

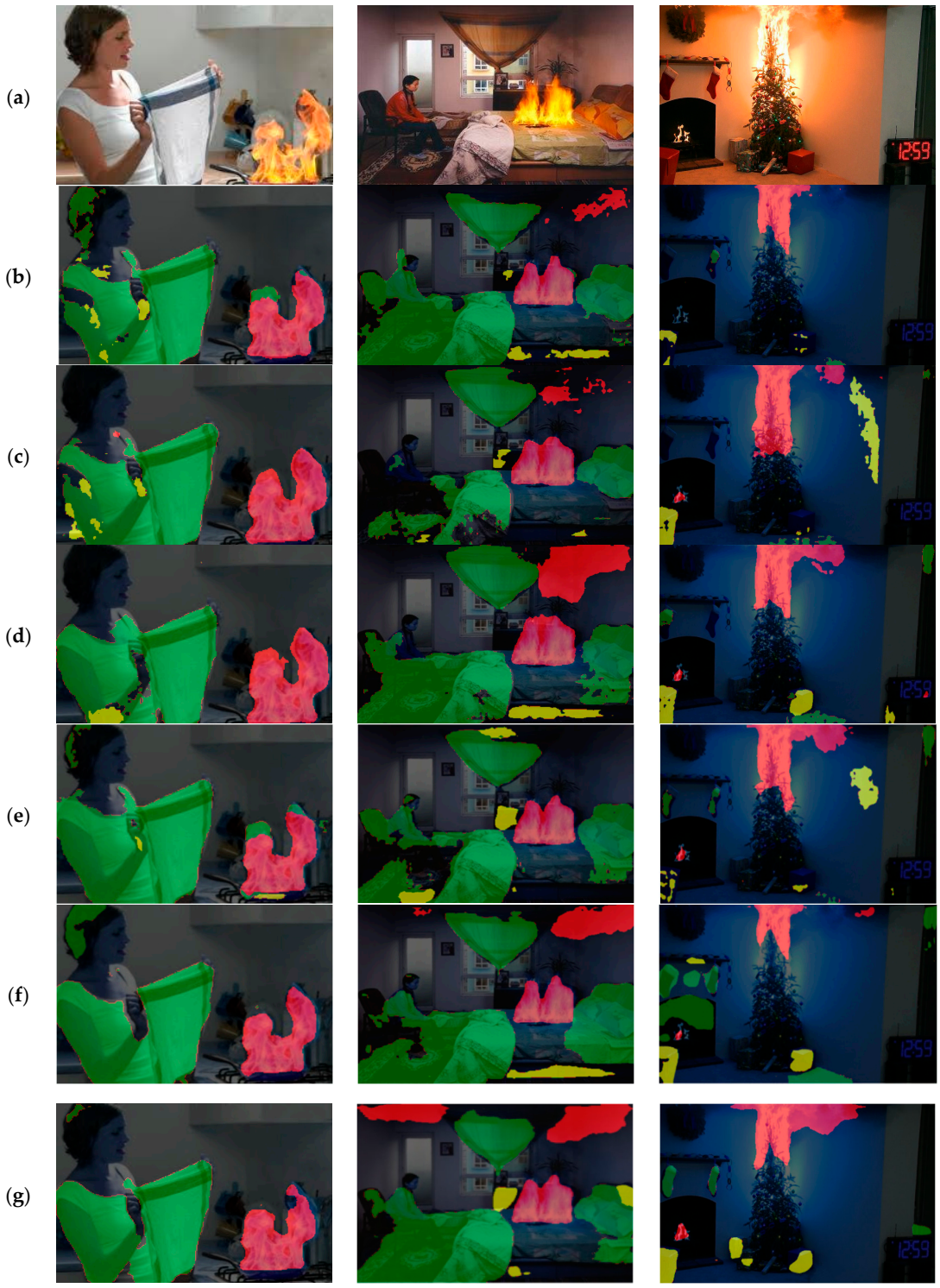


Figure 8. Exemplary visual results from various models: row (**a**) illustrates the input images, while (**b**–**f**) display the segmentation outcomes of PSPNet, CCNet, FCN, ISANet, OCRNet, and (**g**) displays the results of our approach.

5. Conclusions and Discussion

Mainstream fire detection techniques primarily concentrate on fire instances but often overlook the simultaneous detection of both fire instances and combustible materials. In this paper, we adopted the DANet as the main model to detect both fire and combustible materials. DANet is a deep learning network featuring a dual attention mechanism, aimed

at improving the accuracy of identifying minute details and complex relationships within indoor scenes. Additionally, a new database, MAFD, was tailored to collect a wide array of fire instances and potential combustible materials. Through meticulous annotation, this dataset aims to provide a comprehensive resource for training and evaluating models dedicated to fire detection and material recognition within indoor settings. Ultimately, the experimental results indicated that our model on the MAFD achieved an aAcc of 84.26% and mAcc of 77.05%. We pioneered the simultaneous estimation of fire instances and fire load in indoor scenarios, offering a novel strategy for fire safety protection and assessment.

To expand the applicability of our research, there are still more tasks to be undertaken in the future. Firstly, we will incorporate a wider array of combustible materials in the MAFD such as paper, plastic, etc., to ensure a richer representation of potential fire hazards within indoor environments. Secondly, we are refining the training parameters to augment model training efficiency and accuracy. Thirdly, our research can serve as a new reference for designing secure buildings and evaluating the fire resistance of structures.

Author Contributions: Conceptualization, F.H.; data curation, F.H.; methodology, F.H.; software, W.Z.; validation, W.Z.; visualization, W.Z.; writing—original draft, F.H. and W.Z.; writing—review and editing, X.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China (nos. 62203475) and Changsha Natural Science Foundation (nos. kq2208285).

Data Availability Statement: The data supporting the findings of this study are available upon request from the readers.

Conflicts of Interest: We hereby declare that we have no conflicts of interest that could be perceived as influencing the integrity or objectivity of our work.

References

1. Zhang, L.; Wang, G.X.; Yuan, T.; Peng, K.M. Research on Indoor Map. *Geom. Spat. Inf. Technol.* **2013**, 43–47. [CrossRef]
2. Kuti, R.; Zólyomi, G.; László, G.; Hajdu, C.; Környei, L.; Hajdu, F. Examination of Effects of Indoor Fires on Building Structures and People. *Heliyon* **2023**, *9*, e12720. [CrossRef] [PubMed]
3. Kodur, V.; Kumar, P.; Rafi, M.M. Fire Hazard in Buildings: Review, Assessment and Strategies for Improving Fire Safety. *PSU Res. Rev.* **2020**, *4*, 1–23. [CrossRef]
4. Li, S.; Yun, J.; Feng, C.; Gao, Y.; Yang, J.; Sun, G.; Zhang, D. An Indoor Autonomous Inspection and Firefighting Robot Based on SLAM and Flame Image Recognition. *Fire* **2023**, *6*, 93. [CrossRef]
5. Xie, Y.; Zhu, J.; Guo, Y.; You, J.; Feng, D.; Cao, Y. Early Indoor Occluded Fire Detection Based on Firelight Reflection Characteristics. *Fire Saf. J.* **2022**, *128*, 103542. [CrossRef]
6. Wu, X.; Lu, X.; Leung, H. A Video Based Fire Smoke Detection Using Robust AdaBoost. *Sensors* **2018**, *18*, 3780. [CrossRef] [PubMed]
7. Russo, A.U.; Deb, K.; Tista, S.C.; Islam, A. Smoke Detection Method Based on LBP and SVM from Surveillance Camera. In Proceedings of the 2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2), Rajshahi, Bangladesh, 8–9 February 2018.
8. Wang, H.; Zhang, Y.; Fan, X. Rapid Early Fire Smoke Detection System Using Slope Fitting in Video Image Histogram. *Fire Technol.* **2020**, *56*, 695–714. [CrossRef]
9. Wu, X.; Cao, Y.; Lu, X.; Leung, H. Patchwise Dictionary Learning for Video Forest Fire Smoke Detection in Wavelet Domain. *Neural Comput. Appl.* **2021**, *33*, 7965–7977. [CrossRef]
10. Gagliardi, A.; Saponara, S. AdViSED: Advanced Video SmokE Detection for Real-Time Measurements in Antifire Indoor and Outdoor Systems. *Energies* **2020**, *13*, 2098. [CrossRef]
11. Hossain, F.M.A.; Zhang, Y.M.; Tonima, M.A. Forest Fire Flame and Smoke Detection from UAV-Captured Images Using Fire-Specific Color Features and Multi-Color Space Local Binary Pattern. *J. Unmanned Veh. Syst.* **2020**, *8*, 285–309. [CrossRef]
12. Jia, Y.; Chen, W.; Yang, M.; Wang, L.; Liu, D.; Zhang, Q. Video Smoke Detection with Domain Knowledge and Transfer Learning from Deep Convolutional Neural Networks. *Optik* **2021**, *240*, 166947. [CrossRef]
13. Peng, Y.; Wang, Y. Real-Time Forest Smoke Detection Using Hand-Designed Features and Deep Learning. *Comput. Electron. Agric.* **2019**, *167*, 105029. [CrossRef]
14. Cheng, S.; Ma, J. Smoke Detection and Trend Prediction Method Based on Deeplabv3+ and Generative Adversarial Network. *J. Electron. Imaging* **2019**, *28*, 1. [CrossRef]
15. Yuan, F.; Zhang, L.; Xia, X.; Wan, B.; Huang, Q.; Li, X. Deep Smoke Segmentation. *Neurocomputing* **2019**, *357*, 248–260. [CrossRef]

16. Lin, G.; Zhang, Y.; Xu, G.; Zhang, Q. Smoke Detection on Video Sequences Using 3D Convolutional Neural Networks. *Fire Technol.* **2019**, *55*, 1827–1847. [CrossRef]
17. Li, J.; Zhou, G.; Chen, A.; Wang, Y.; Jiang, J.; Hu, Y.; Lu, C. Adaptive Linear Feature-Reuse Network for Rapid Forest Fire Smoke Detection Model. *Ecol. Inform.* **2022**, *68*, 101584. [CrossRef]
18. Liu, H.; Lei, F.; Tong, C.; Cui, C.; Wu, L. Visual Smoke Detection Based on Ensemble Deep CNNs. *Displays* **2021**, *69*, 102020. [CrossRef]
19. Zhan, J.; Hu, Y.; Zhou, G.; Wang, Y.; Cai, W.; Li, L. A High-Precision Forest Fire Smoke Detection Approach Based on ARGNet. *Comput. Electron. Agric.* **2022**, *196*, 106874. [CrossRef]
20. Hu, Y.; Zhan, J.; Zhou, G.; Chen, A.; Cai, W.; Guo, K.; Hu, Y.; Li, L. Fast Forest Fire Smoke Detection Using MVMNet. *Knowl.-Based Syst.* **2022**, *241*, 108219. [CrossRef]
21. Hosseini, A.; Hashemzadeh, M.; Farajzadeh, N. UFS-Net: A Unified Flame and Smoke Detection Method for Early Detection of Fire in Video Surveillance Applications Using CNNs. *J. Comput. Sci.* **2022**, *61*, 101638. [CrossRef]
22. Khan, S.; Muhammad, K.; Mumtaz, S.; Baik, S.W.; de Albuquerque, V.H.C. Energy-Efficient Deep CNN for Smoke Detection in Foggy IoT Environment. *IEEE Internet Things J.* **2019**, *6*, 9237–9245. [CrossRef]
23. He, L.; Gong, X.; Zhang, S.; Wang, L.; Li, F. Efficient Attention Based Deep Fusion CNN for Smoke Detection in Fog Environment. *Neurocomputing* **2021**, *434*, 224–238. [CrossRef]
24. Muhammad, K.; Khan, S.; Palade, V.; Mehmood, I.; de Albuquerque, V.H.C. Edge Intelligence-Assisted Smoke Detection in Foggy Surveillance Environments. *IEEE Trans. Industr. Inform.* **2020**, *16*, 1067–1075. [CrossRef]
25. Strese, M.; Schuwerk, C.; Iepure, A.; Steinbach, E. Multimodal Feature-Based Surface Material Classification. *IEEE Trans. Haptics* **2017**, *10*, 226–239. [CrossRef] [PubMed]
26. Zhang, H.; Jiang, Z.; Xiong, Q.; Wu, J.; Yuan, T.; Li, G.; Huang, Y.; Ji, D. Gathering Effective Information for Real-Time Material Recognition. *IEEE Access* **2020**, *8*, 159511–159529. [CrossRef]
27. Lee, S.; Lee, D.; Kim, H.-C.; Lee, S. Material Type Recognition of Indoor Scenes via Surface Reflectance Estimation. *IEEE Access* **2022**, *10*, 134–143. [CrossRef]
28. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual Attention Network for Scene Segmentation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
29. Yu, F.; Koltun, V.; Funkhouser, T. Dilated Residual Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
30. Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A Comprehensive Survey on Transfer Learning. *Proc. IEEE Inst. Electr. Electron. Eng.* **2021**, *109*, 43–76. [CrossRef]
31. GitHub—Open-Mmlab/Mmsegmentation: OpenMMLab Semantic Segmentation Toolbox and Benchmark. Available online: https://github.com/open-mmlab/mmsegmentation (accessed on 14 March 2023).
32. Wilson, A.C.; Roelofs, R.; Stern, M.; Srebro, N.; Recht, B. The Marginal Value of Adaptive Gradient Methods in Machine Learning. *arXiv* **2017**, arXiv:1705.08292.
33. Zhou, Y.-C.; Hu, Z.-Z.; Yan, K.-X.; Lin, J.-R. Deep Learning-Based Instance Segmentation for Indoor Fire Load Recognition. *IEEE Access* **2021**, *9*, 148771–148782. [CrossRef]
34. Torralba, A.; Russell, B.C.; Yuen, J. LabelMe: Online Image Annotation and Applications. *Proc. IEEE Inst. Electr. Electron. Eng.* **2010**, *98*, 1467–1484. [CrossRef]
35. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [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. Huang, Z.; Wang, X.; Huang, L.; Huang, C.; Wei, Y.; Liu, W. CCNet: Criss-Cross Attention for Semantic Segmentation. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.
38. Shelhamer, E.; Long, J.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 640–651. [CrossRef] [PubMed]
39. Huang, L.; Yuan, Y.; Guo, J.; Zhang, C.; Chen, X.; Wang, J. Interlaced Sparse Self-Attention for Semantic Segmentation. *arXiv* **2019**, arXiv:1907.12273.
40. Yuan, Y.; Chen, X.; Wang, J. Object-Contextual Representations for Semantic Segmentation. In *Computer Vision—ECCV 2020*; Lecture Notes in Computer Science; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; pp. 173–190, ISBN 9783030585389.