Heliyon 8 (2022) e12375

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Instance segmentation using semi-supervised learning for fire recognition

Guangmin Sun*, Yuxuan Wen, Yu Li

Faculty of Information Technology, Beijing University of Technology, Beijing, 100124, China

ARTICLE INFO

Keywords:
Fire image recognition
Instance segmentation
Deep learning
Semi-supervised learning
Self-training

ABSTRACT

Fire disaster brings enormous danger to the safety of human life and property, and it is important to identify the fire situation in time through image processing technology. The current instance segmentation algorithms suffer from problems such as inadequate fire images and annotations, low recognition accuracy, and slow inference speed for fire recognition tasks. In this paper, we propose a semi-supervised learning-based fire instance segmentation method based on deep learning image processing technology. We used a lightweight version of the SOLOv2 network and optimized the network structure to improve accuracy. We propose a semi-supervised learning method based on fire features. To reduce the negative impact of error pseudo-labels on the model training, the pseudo-labels are matched by the color and morphological features of flames and smoke at the pseudo-label generation stage, and some images are screened for strong image enhancement before entering the next round of training for the student model. We further exploit the potential of the model with a limited dataset and improve the model accuracy without affecting the inference efficiency of the model. Experiments show that our proposed algorithm can successfully improve the accuracy of fire instance segmentation with good inference speed.

1. Introduction

Fire brings accidents and risks to human beings and seriously threatens the safety of people's lives and properties. According to NFPA data, fire departments reported about 1.3 million fires in 2019. These fires resulted in approximately 3,700 civilian fire deaths and $14.8 billion in property damage [1]. The method of fire detection has been based on the sensing capability of traditional sensors. This method has good applications in small confined spaces, but for large areas with non-confined spaces, this method is limited by environmental and cost constraints [2]. Image-based fire detection technology shows advantages in many aspects: better recognition efficiency, more intuitive information, and lower costs. This method can be deployed in a variety of platforms and scenarios at low cost, making it more suitable for widespread use at scale. With the widespread use of monitoring equipment in recent years, image-based fire detection systems become more convenient and faster.

Early research on fire image processing focused on extracting the color features of the flame and using the set threshold rules as the basis for the determination. Afterward, the researchers introduced complementary features to the color features of the flames, including morphological and textural features of suspicious regions, and motion features extracted from the video objects. In addition to this, various decision algorithms such as Bayesian algorithm, support vector machine, and expert system are introduced for improving the accuracy and recall of fire detection. Since the AlexNet [3] was proposed, Deep learning is rapidly evolving. Convolutional Neural Networks have replaced traditional manual feature extraction and become the mainstream method for image processing. In fire detection, traditional manual feature extraction relies excessively on human prior knowledge. Deep learning uses datasets to train neural networks instead of relying on the prior features. In the field of computer vision, instance segmentation is further refined in object detection, which separates the foreground and background of an object and achieves pixel-level object separation. The fire instance segmentation algorithm has some problems: lack of annotation and low speed. For this reason, how to combine the optical characteristics of fire with deep learning algorithms to design a high-precision fire instance segmentation algorithm has become a technical problem to be solved.

In the task of this paper, data labeling is difficult and there is no public dataset, while fire images are easily available on the Internet. In this case, semi-supervised learning is well suited to our task requirements. But too limited supervised datasets will lead to insufficient accuracy of the teacher model itself and generate partially wrong labels. The errors introduced by the semi-supervised approach may be amplified after several iterations of training.

* Corresponding author.
E-mail address: gmsun@bjut.edu.cn (G. Sun).

https://doi.org/10.1016/j.heliyon.2022.e12375
Received 20 March 2022; Received in revised form 20 June 2022; Accepted 7 December 2022

This research proposes a fire instance segmentation algorithm based on semi-supervised learning, and we apply the instance segmentation algorithm to the field of fire instance segmentation with optimized inference speed and accuracy. We introduce a semi-supervised learning strategy to solve the problem of difficult annotation of fire datasets. To address the problem that semi-supervised learning may introduce "dirty" data, we propose a semi-supervised learning strategy based on fire features for instance segmentation models. We introduce color and morphological features of flames and smoke to determine and filter the pseudo-labels generated in the semi-supervised learning and reduce the error caused by the mixture of wrong labels in the pseudo-labels. It improves the model accuracy and generalizability.

The remainder of this paper has the following sections: Section 2 presents previous research on image-based fire detection methods. Section 3 describes the network structure and optimization module. Section 4 describes the proposed semi-supervised learning method. Section 5 gives the experimental results and result analysis for the proposed method. Finally, Section 6 concludes the research of this paper.

2. Related work

Early fire image detection methods mostly used the color features of the image as the main reference quantity. Cruz [4] et al. designed a forest fire detection index (FFDI) by analyzing RGB channel characteristics, calculating index values based on RGB channels and setting thresholds to determine flame conditions. In the following studies, multi-feature fusion methods combining color features, texture features, and motion features have become mainstream. Wang [5] et al. used static features such as texture features and edge complexity of flames with dynamic features such as drift characteristics and region randomness input to a support vector machine (SVM) for smoke recognition. Chen [6] proposed a flame recognition algorithm incorporating dynamic features and Incremental Vector Support Vector Machine (IV-SVM) in the condition of integrating color information into the Scale Invariant Feature Transformation method (SIFT).

Since the rise of Deep Learning, Convolutional Neural Networks (CNN) have been applied as a method to extract features in Computer Vision. In object detection, Girshick [7] et al. proposed the R–CNN network in 2013, which extracts image features using a convolutional network and uses SVM to do classification and locate objects in a sliding window on the image from the top down. In 2015, Fast R–CNN [8] network was proposed with the addition of a candidate Region Proposal Network (RPN) instead of a sliding window based on R–CNN, which greatly improved the speed of object detection. In the field of instance segmentation, He [9] et al. proposed Mask R–CNN, which added a fully connected segmentation sub-network to Faster R–CNN [10], changed from two tasks (detection and classification) to three tasks (detection, classification, and segmentation), decomposed the instance segmentation task into a region-by-region segmentation task, and became a representative algorithm for two-stage instance segmentation. A new concept of instance category is introduced, Wang [11] takes a fresh perspective on the task of instance segmentation. He assigned categories to each pixel in an instance based on the instance location and size proposed a SOLO network framework that outperformed other single-stage instance segmentation algorithms, and later proposed a version of SOLOv2 [12] that optimized accuracy and running efficiency, and performed well in other tasks such as object detection and Panoptic Segmentation.

In the field of fire detection, Khan Muhammad [13] et al. proposed a neural network similar to Squeeze-Net architecture for flame recognition, containing three Convolutional layers with four Pooling layers, removing the final fully connected layer. The feature maps are extracted in the CNN, and binarized to segment the flame part, achieving 94.5% accuracy in indoor-specific scenes. Mao [14] proposed a new flame recognition method, it uses multiple convolutional layers and fully connected layers for automatic feature extraction. Secondly, for video-like samples, the inter-frame timing information optimization algorithm is considered. Li [15] proposed an image fire detection algorithm based on YOLOv3, which achieves 83.7% accuracy with 28 FPS on its own dataset. Moulay [16] proposed Deep-Fire, a deep convolutional neural network for fire pixel detection and fire segmentation, for forest fire scenes, and achieved good results in segmentation of fire regions in outdoor forest scene datasets. Sharma [17] compared the pre-trained VGG16 and Resnet50 by fine-tuning them on their dataset and testing them on the established unbalanced dataset, and the Resnet50 network achieved a good performance. Bochkov [18] divided the flame contours into red, yellow and orange regions to identify the hottest areas of the flame to help firefighters. The wUUNet model is proposed based on the U-net network architecture with VGG16 as the backbone network, which has achieved superior results on its dataset.

These research efforts have made great progress in the field, but there are still many shortcomings in the task of fire instance segmentation in real scenarios. Our study uses a dataset of real fire images obtained from the Internet, which contains many images that are challenging for computer vision, with varying image quality, high imbalance and lack of annotated data. In this paper, present instance segmentation algorithms are compared and an attempt is made to select and modify an instance segmentation algorithm that is more balanced in speed and accuracy, and a semi-supervised learning strategy based on fire features is proposed to solve these difficulties and improve the accuracy.

3. Network structure

3.1. Instance segmentation network

Instance segmentation is considered to be one of the more important and challenging areas of computer vision. In the field of instance segmentation, the algorithm based on the idea of detection before segmentation is represented by Mask-RCNN [9], which has the advantage of high accuracy, but also has certain limitations, such as high prediction latency, not real-time, instance segmentation results under the limitation of object detection frame. Instance segmentation models that can be called real-time are YOLACT and YOLACT++ [19], which divide the instance segmentation into two parallel subtasks and use a single-stage network structure to keep the network computation as small as possible. The core idea of the SOLO [11] (Segmenting Objects by Locations) algorithm, proposed by Wang in 2020, is transforms the segmentation task into a location classification task so that anchor and bounding boxes are not needed. The segmentation of instance objects is achieved by assigning a class to each instance pixel according to its position and size. As a representative of single-stage instance segmentation algorithms, the SOLO algorithm achieves good accuracy while being real-time, and is well suited for this study due to its excellent balance of speed and accuracy. In the subsequent improved SOLOv2 [12], some settings in SOLO are inherited, and the mask branch is decomposed into a mask kernel branch to predict the convolutional kernel and a mask feature branch to predict the convolutional features. Meanwhile, SOLOv2 uses Matrix NMS which is 9 times faster than the conventional NMS. the network structure of SOLOv2 is shown in Figure 1. In this study, to balance both computing efficiency and accuracy, we select a lightweight version of SOLOv2, use the ResNet50 network as the backbone feature extraction network, and reduce the size of the SOLOv2-head.

In Figure 1, the features of each FPN layer are output to two subnetworks, one for predicting instance classes and one for predicting instance masks. The arrows indicate convolution or interpolation. In the Category Branch, an input image is divided into a uniform grid S × S. C is the number of classes. In the Mask Branch, F is the input feature. For each grid, the kernel branches predict the convolutional kernel weights. E is the dimension of the mask feature. D is the number of parameters. After the last convolution layer, the size of the instance mask becomes H × W × S^2 [11, 12]

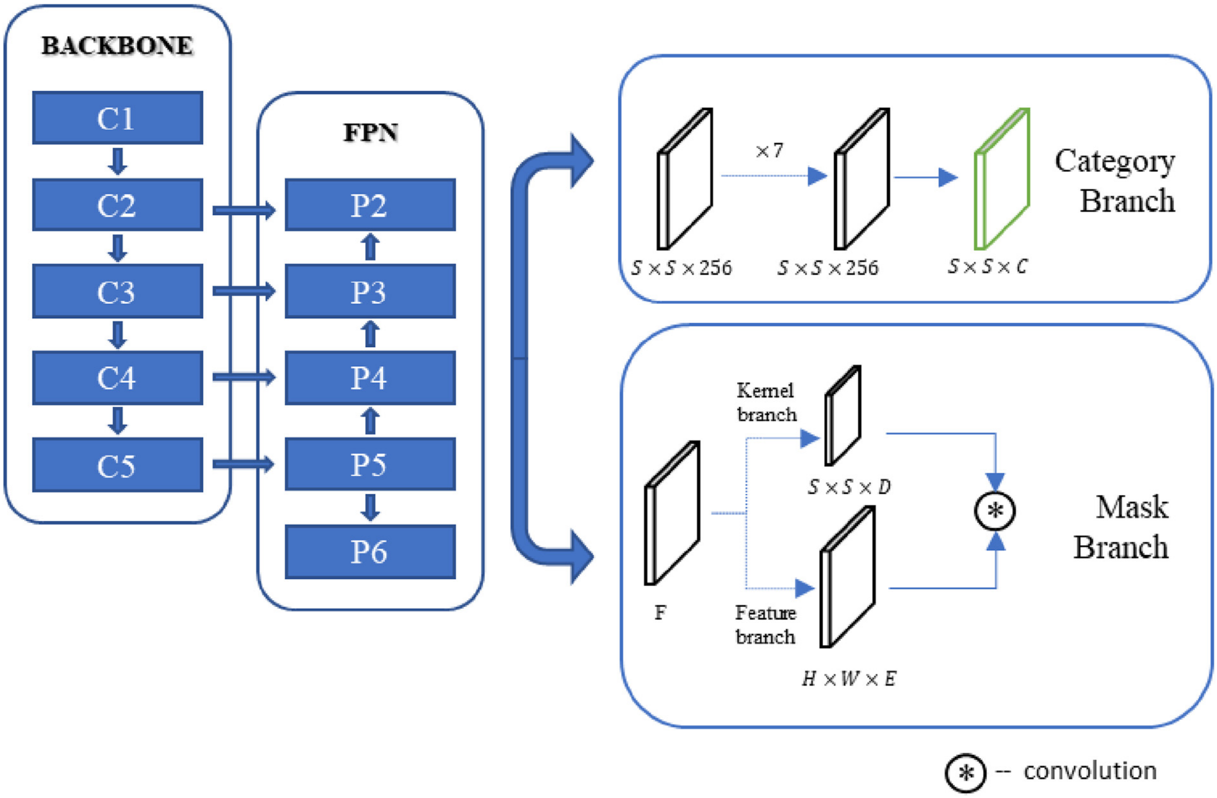


Figure 1. SOLOv2 network structure.

3.2. Deformable convolution

It is a challenge to make CNN effectively deal with the shape changes of the target for the complex shape of flame and smoke targets. In fire scenarios, flame and smoke deformations are unknown and unpredictable, and hand-designed features and algorithms are unable to handle such complex deformations. In this study, the regular convolutional kernels in layers C3–C5 of the backbone and SOLOv2-head in the network were replaced with deformable convolutional DCNv2 [20]. The Deformable Convolutional Network (DCN) [21] algorithm was proposed to enhance the ability of the model to learn complex target invariants. The core idea of DCN is that it believes that the convolutional kernel should not be a simple rectangle but may have its optimal convolutional kernel structure at different stages and on different feature maps. Therefore, DCN proposes to learn an offset for each point on the convolutional kernel, and the convolutional kernel can learn different convolutional kernel structures according to different data.

According to the study [22], not every pixel in the receptive field has an similar effect on the output. Since pixels closer to the center have a greater impact, only a small part of the theoretical receptive field is valid and obeys a Gaussian distribution. Deformable convolution can learn perceptual fields adaptively. Compared with standard convolution, deformable convolution has a powerful adaptive extraction ability for complex targets. When Deformable Convolutions are used, they adaptively adjust to the scale and shape of the object, enhancing localization especially for non-rigid targets such as flames and smoke.

3.3. Attention mechanism

The attention mechanism has become an important component of deep learning network architecture. This makes it and has many applications in natural language processing, computer vision. Convolutional Block Attention Module (CBAM) [23] is an attention mechanism module that combines spatial and channel. Compared with SEnet [24], an attention mechanism that focuses only on channels, CBAM can achieve better results. CBAM computes the attention map of the feature maps generated by the convolutional network from both channel and spatial dimensions. The attention mapping is then multiplied with the input feature mapping for adaptive learning of features. CBAM is a lightweight general-purpose module that can be incorporated into various convolutional neural networks for end-to-end training. In this paper, we add the CBAM attention mechanism to the backbone feature extraction network ResNet-50. The module is added after each ResNet-Block as shown in Figure 2.

In Figure 2, we add the CBAM attention module after layers C1–C4 in Backbone (ResNet).

3.4. Multiscale fusion feature pyramid

In the field of object detection and instance segmentation, small-scale targets are suitable for detection in the shallow feature layer because they have simpler semantic information, while large-scale targets have more complex semantic information and are therefore suitable for detection in the deep feature layer. In our research, the FPN [25] in the multiscale fusion part of the network is replaced by the PAFPN [26]. FPN is a top-down route that passes down strong semantic features at higher levels through lateral connections, enhancing only the semantic information of the feature pyramid and not localization information. Therefore, PAFPN adds a bottom-up path to shorten the information propagation path, using the precise localization information of low-level features.

The network structure of PAFPN is shown in Figure 3.

4. Semi-supervised learning strategy based on fire features

Semi-supervised learning is a learning paradigm between supervised and unsupervised learning and is often used under classification tasks. As we all know, in supervised learning, the category labels of the samples

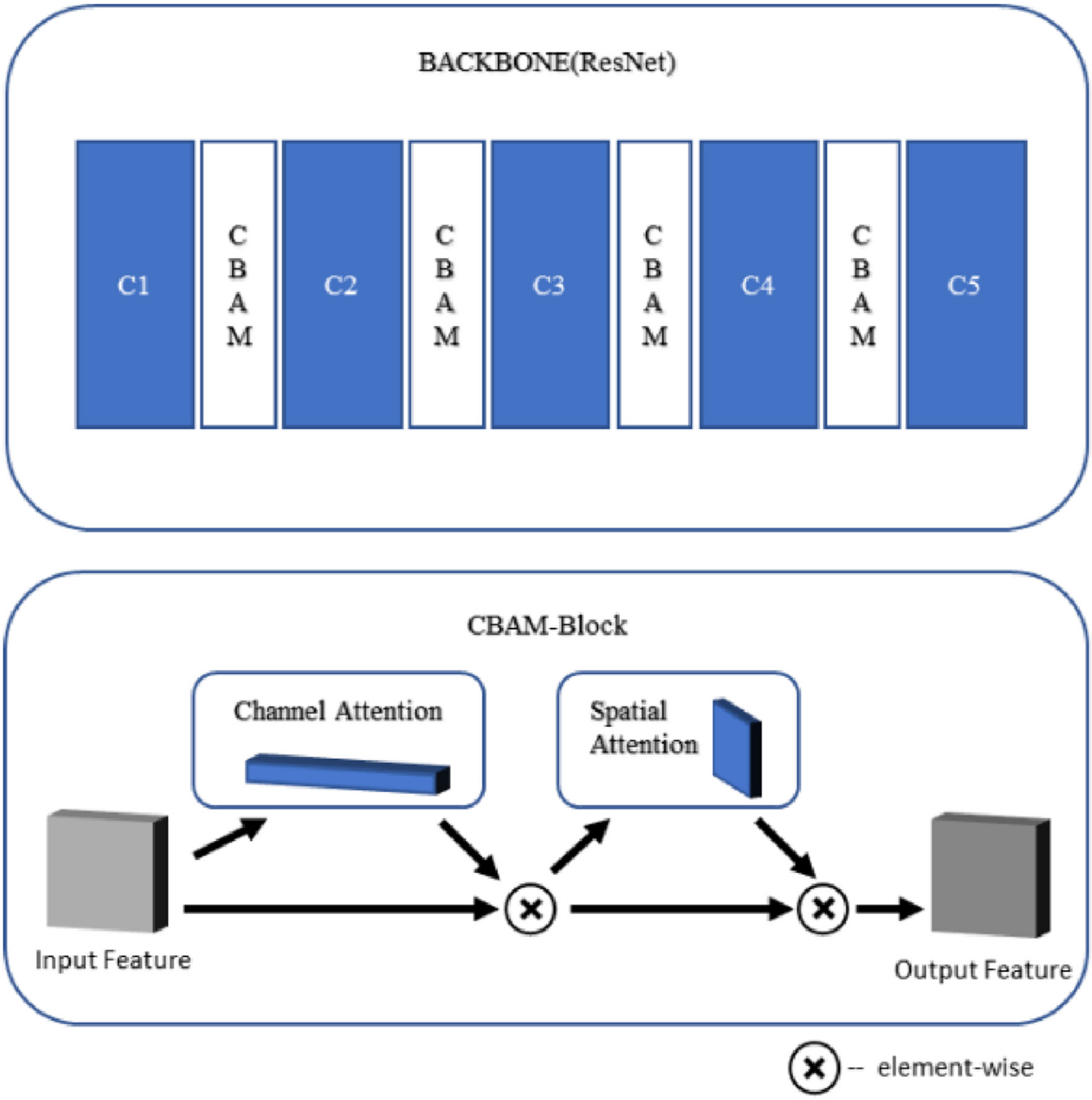


Figure 2. CBAM attention mechanism structure.

are known. The goal of learning is to find the connection between the features of the samples and the category labels. In general, the larger the number of training samples, the higher the classification accuracy and generalizability of the trained classifier. Semi-supervised learning uses many unlabeled samples and a small number of labeled samples to train the classifier, solving the problem of insufficient labeled samples.

However, in the fire instance segmentation task, the need for manual pixel-level labeling of non-rigid target samples such as flames and smoke is very costly, which leads to the rarity of labeled samples and datasets. Besides, unlabeled samples can be easily collected, and their number is often hundreds of times higher than that of labeled samples. This situation qualifies very well for semi-supervised learning. Although semi-supervised learning is more likely to be applied to classification tasks, we believe that the application of the idea of semi-supervised learning still has a great role in this study.

4.1. Self-Training

Self-training is one of the most common semi-supervised methods, where the main idea is to augment the labeled dataset with the unlabeled dataset. First, the labeled data is used to train a model, and then this model is used to label the unlabeled data. Since we know that not all predictions of a trained model on unlabeled data can be good. Therefore, for classical Self-training, it is common to filter some of the predictions using the confidence score to select a subset of the predicted pseudo-labels. Next, the generated pseudo-labels are combined with the original labeled data. The new student model is trained on the merged data. The whole process can be repeated several times until the accuracy is converged.

However, the Self-training algorithm is affected by the labeling accuracy of pseudo-labels. Many incorrect pseudo-labels can make the model move in the wrong direction. Our primary efforts below focus on how to reduce the impact of incorrect pseudo-labels on the model. We designed a semi-supervised learning method based on fire features. The method uses color features and morphological features of flames and smokes in fires to filter pseudo-labels. The pseudo-labels are subsequently subjected to a strong data enhancement operation. This method is effective in avoiding the negative impact of false pseudo-labels on the model. The Flowchart is shown in Figure 4.

As shown in Figure 4, the method is divided into Teacher Stage and Student Stage. In the Teacher Stage, the trained teacher model infers and generates Pseudo-labels on the unlabeled dataset. In Student Stage, this

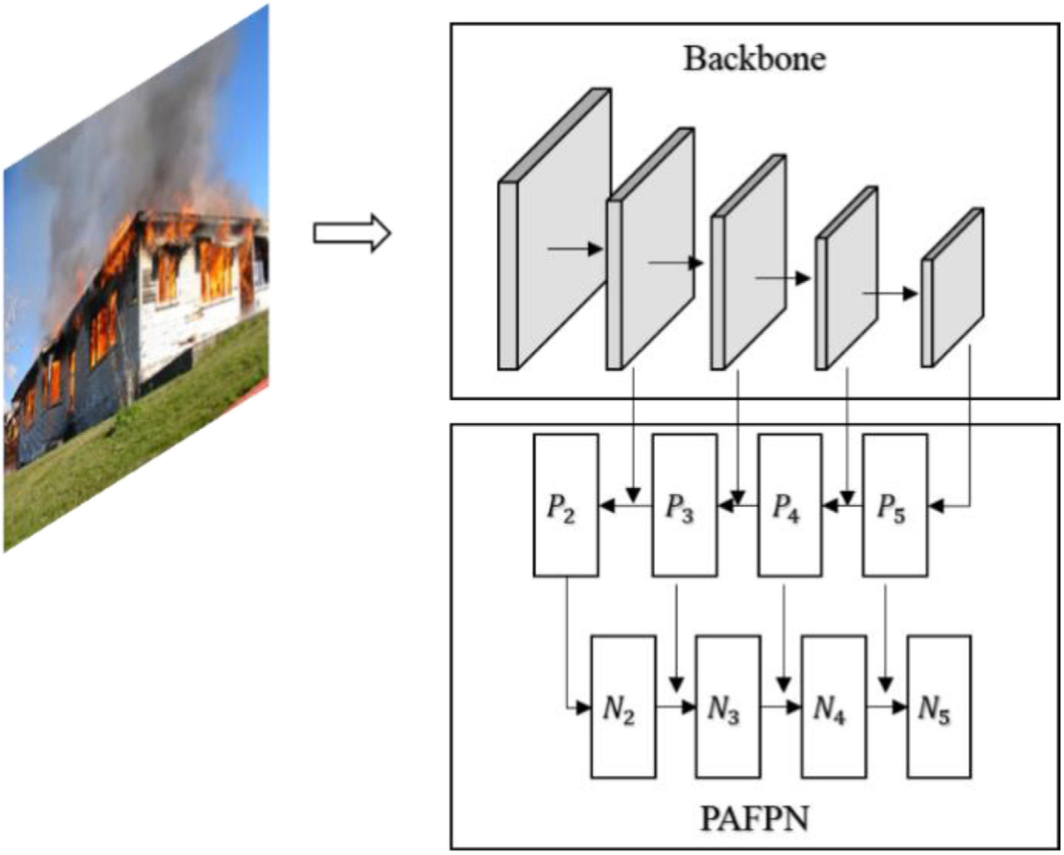


Figure 3. PAFPN network architecture.

pseudo-labeled dataset is filtered and merged with the labeled dataset. We perform hybrid data augmentation on the new dataset. The new training set is used to train the new teacher model.

4.2. Filtering of pseudo-labels based on fire features

Models trained using labeled datasets are limited by the lack of dataset size and diversity. This leads to a lack of model accuracy and generalization. Sun, lights, and other things can often be misidentified as fire. Improving the accuracy of the model without introducing new computing costs is what we want to achieve. The fire instance segmentation task is different from other tasks in that flames and smoke have their distinct color features and morphological characteristics, which is an important basis for us humans in observing and judging whether a fire has occurred. We introduce color features and morphological features evident in fires. After the pseudo-label generation in semi-supervised learning, the pseudo-label is scored and filtered by integrating the teacher model's score and feature matching degree. Feature degree

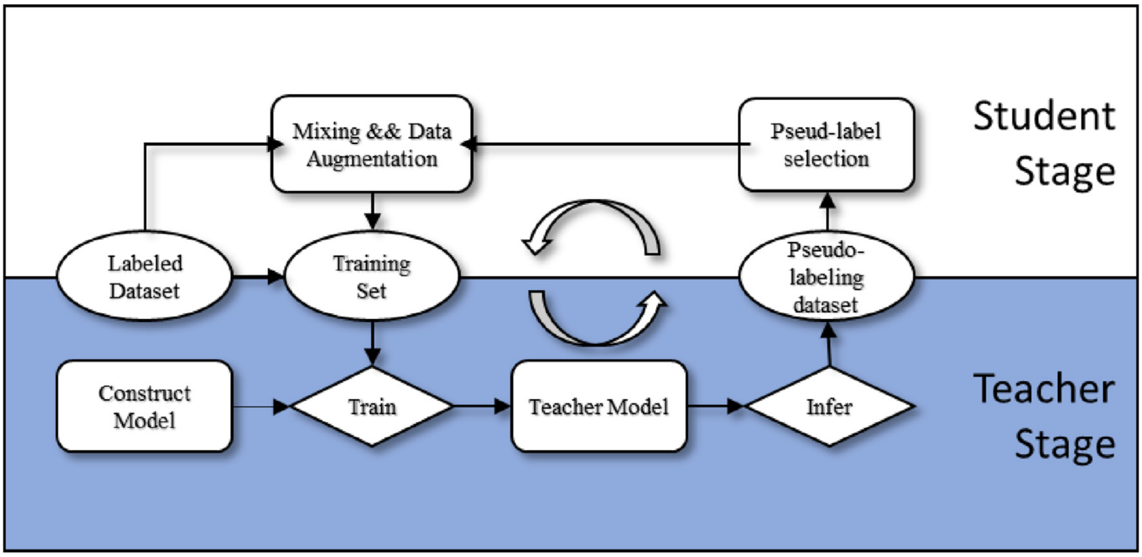


Figure 4. Flow Chart of Fire-based self-Training.

matching will remove samples that are wrong and reduce the error rate of pseudo-labeling, thus reducing the negative impact on the new round of student models. In the feature degree matching work, we aim to reduce the scores of incorrect samples and do not wish to reduce the recall of correct pseudo-labels, so the selection of thresholds is more generous.

The color features of the flame are mainly reflected in spectral characteristics, such as prominent brightness and chromaticity. In RGB images, the results of a large number of studies have shown that the flame image pixel point color is between red and yellow and the individual components satisfy the relationship R > G > B. The color feature criterion of the flame used in the RGB image flame detection model proposed by Chen [27] is shown in the following equation.

$$R > G > B \tag{1}$$

$$R > R_t \tag{2}$$

$$S > (255 - \mathrm{R}) \times R_t \,/\, S_t \tag{3}$$

$$S = (255 - 3 \times \min(\mathrm{R}, \mathrm{G}, \mathrm{B}))\,/\,(R + G + B) \tag{4}$$

In Eqs. (1), (2), (3), and (4), where R, G, and B denote the values of the three channels of red, green, and blue of a pixel. In Eqs. (2) and (3), R_t denotes a threshold value on the R channel, which takes values in the range of [55,65], and S_t in Eq. (3) denotes a threshold value of color saturation, which takes values in the range of [115,135], and the range of these two values is based on the empirical values obtained from experiments [27]. Eq. (4) is the formula for calculating the color saturation S. If a pixel in the image satisfies the above three conditions at the same time, it can be judged as a suspected flame pixel.

The smoke produced during a fire is a mixture of gases and soot. The smoke produced varies with the characteristics of the material being burned, the temperature and the availability of oxygen. In general, different periods of combustion in a fire can lead to different color states of smoke, and as the temperature rises, the smoke color can change from light gray to black. According to the smoke color model of Yuan et al [28], the difference between the three component values in the RGB color model is small for most of the smoke pixels. Using their proposed smoke color model criterion is shown in the following equations.

$$C_{min} = min(R, G, B) \tag{5}$$

$$C_{max} = max(R, G, B) \tag{6}$$

$$I = (R + G + B)\,/\,3 \tag{7}$$

```
Rule 1: $|C_{max} - C_{min}| < T_1$
Rule 2: $Rule\ 2 : T_2 < I < T_3$
Rule 3: $Rule3 : C_{max} = B$ AND $|C_{max} - C_{min}| < T_4$
If (Rule 1 AND Rule 2) OR (Rule3 AND Rule 2).
{
Smoke pixel
}
Else
{
Non-Smoke pixel
}
```

Equation (5) and Equation (6) represent the maximum and minimum values of RGB color model components. Eq. (7) indicates that I is the average of R, G and B. In the above rules, Rule 1 represents the gray characteristics, rule 2 limits the range of intensity variation, and rule 3 allows the pixel color to be whitish and bluish. The absolute value of the difference between the maximum and minimum values of R, G and B should be less than the predefined threshold T_1. And the intensity of the smoke pixel is between T_2 and T_3. The smoke color may turn white and blue in some special cases. Therefore, the value of component B is slightly larger than that of components R and G. T_4 is a predetermined threshold value slightly larger than T_1.

Since the form of flame and smoke is irregular and easy to confuse the object, such as lights, sun, and so on are more regular round, square and other shapes. Therefore, we introduced three morphological features, circularity, rectangularity and contour roughness, to match the characteristic degree of each mask contour.

The roundness parameter is used to measure how similar the outline of an object is compared to a circle. Compared with the shapes of objects such as daylight and light, flames and smoke are more irregular and less similar to circles. Therefore, the degree of circular approximation can be used to exclude interference sources. The formula for calculating roundness is as follows (8).

$$Circu = 4\pi \times S\big/L^2 \tag{8}$$

In Eq. (8), where S denotes the area of the mask contour and L denotes the circumference of the mask contour. The smaller the similarity between the shape of the mask and the circle, the smaller the value of circularity.

Rectangularity is a parameter that measures how similar the contours of a target area are to a rectangle. Areas of flame and smoke can be distinguished from near-rectangular sources of interference (e.g., interior lights, windows) and elongated objects (e.g., glow sticks) using rectangularity. The formula for rectangularity is as follows (9).

$$Rect = S/S_R \tag{9}$$

In Eq. (9), where S denotes the area of the mask contour and S_R denotes the area of the smallest outer rectangle that contains the mask. The closer the object is to the rectangle, the larger the value of the rectangularity.

Contour roughness [29] describes the roughness of an object's contour by calculating the ratio of the object's perimeter to the perimeter of its convex hull. Convex Hull is a concept in geometry and graphics. The edges of flames and smoke are often not normal convex polygons but have complex and rough edges. Therefore, the Contour roughness of flames and smoke is larger compared to normal shapes. Formula (10) shows the evaluation method of Contour roughness. L_{CH} is the perimeter of the convex hull and L is the object perimeter.

$$B_R = L/L_{CH} \tag{10}$$

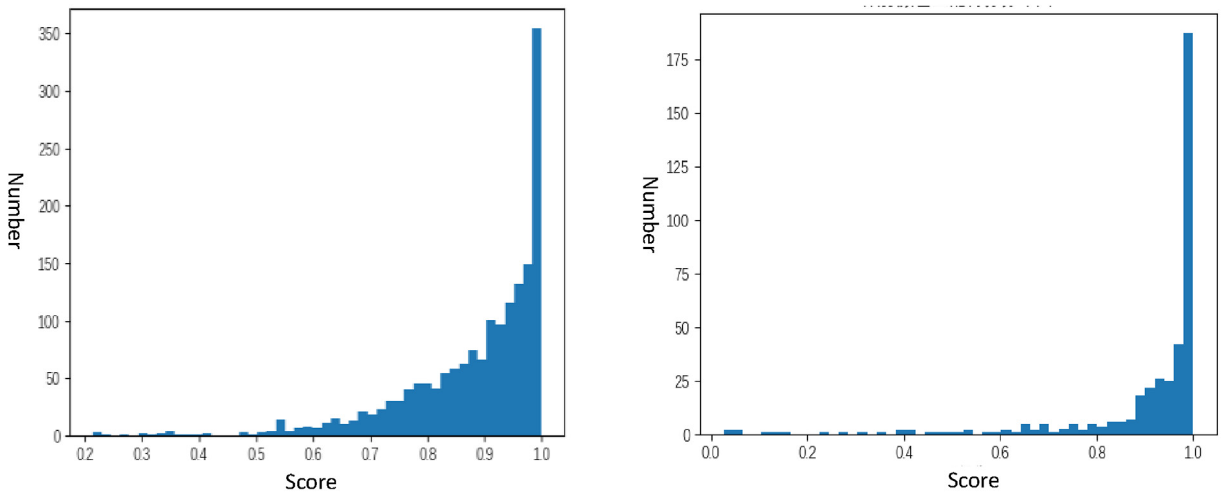
To verify the validity of the above method, we calculate all the data in the labeled dataset. For each image in the dataset, a matching score is calculated according to the method based on Eqs. (1), (2), (3), (4), (5), (6), (7), (8), (9), and (10) above. We use histograms to represent the statistical results of the matching scores in order to show more visually the statistical validity of our method.

The statistics of the results calculated by color matching of flames and smoke are shown in Figure 5.

From Figure 5(a), it is known that the flame color model has a good ability to recognize the pixels in the flame instances. According to the data, the average value of the matching score is 0.882, and 79.3% of the flame instances in the training set have a color matching score higher than 0.8. 96.2% of the flame instances have a color matching score higher than 0.6. This proves that the flame model we use can effectively distinguish the pixels that match the flame color characteristics effectively. As can be seen from Figure 5(b), only 5.8% of the smoke instances have a color match score below 0.6, and 86.4% of the smoke instances have a color match score above 0.8, which verifies that the smoke color model we use can effectively distinguish smoke pixels that match the a priori color features.

The results of the three morphological matching methods for flames are statistically shown in Figure 6.

As can be seen from Figure 5, most of the flame instances have roundness scores concentrated below 0.9, with 84% of the scores concentrated in the interval from 0.4 to 0.8. Only 0.77% of the flame instances in the figure have roundness scores greater than 0.9, verifying



(a) Flame Color

(b) Smoke Color

Figure 5. Flame and smoke color matching score chart. (a) The flame color matching score chart and (b) The smoke color matching score chart.

that flame instances have more significant differences from round objects. Most of the flame instances have rectangularity scores below 0.8. Only 0.2% of the flame instances have rectangularity scores greater than 0.8, and 80.1% of them have rectangularity scores between 0.3 and 0.6. This verifies that flame instances also have a large differentiation from rectangular objects. Only 0.18% of the flame instances have a boundary roughness score greater than 0.5, and 90.1% of the flame instances have a boundary roughness score concentrated below 0.3. This verifies that the flame has a lower boundary roughness score and the flame has a rougher boundary.

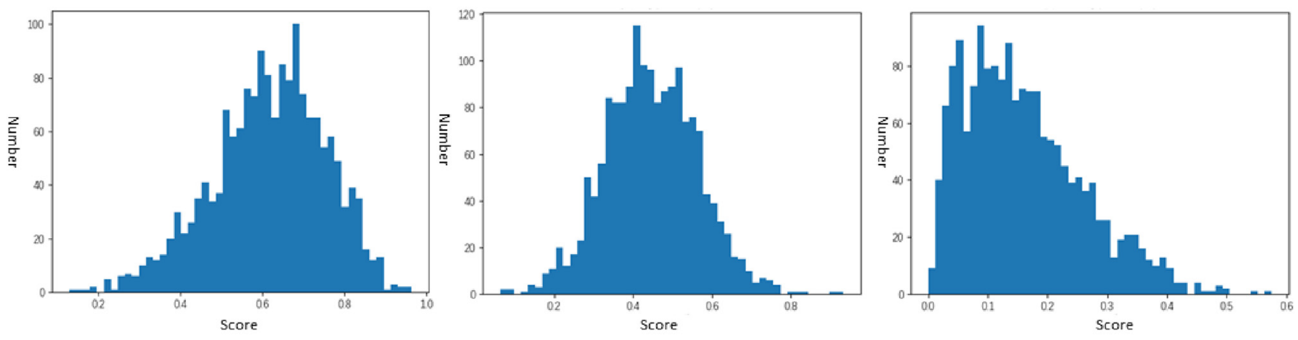
The results of the three morphological matching methods for flames are statistically shown in Figure 7.

In Figure 7, 91.2% of the smoke instances have roundness scores between 0.4 and 0.8, and only 3.77% of the smoke instances have roundness scores greater than 0.8. All of the smoke instances have rectangularity scores less than 0.7, and only 2.77% of the smoke instances have rectangularity scores greater than 0.6. 99.5% of the smoke instances have boundary roughness scores less than 0.5, and 89.7% of the smoke instances have boundary roughness scores greater than 0.3. These data verify that smoke instances are more distinct from circles and rectangles and have rougher boundaries.

In model training, the category distribution has a significant impact on the prediction accuracy of the model for different categories. And to avoid further amplification of this situation during semi-supervised training, we classify samples from different categories after pseudo-label generation. We divide the pseudo-labeled samples into three categories: smoke and fire both, smoke only, and flame only, and perform feature degree matching in each category. After feature degree matching is performed, the matching results are combined with the confidence level of the teacher model's output for each pseudo-label, and the pseudo-label data can be filtered to exclude those samples that are wrong. These three categories filtered roughly equal numbers of samples into the new round of student model training. In the end, the pseudo-labeled samples introduced into the next round of student model training do not result in large recognition errors for a particular class due to differences in the number of classes.

4.3. Data augmentation algorithm

In Noisy Student Self-Training [30], a new student model is trained based on this large dataset by merging the labeled and pseudo-labeled datasets and introducing strategies such as data augmentation and Dropout in the training. This step introduces a large amount of "noise". In the case when the same model is used for both teacher and student since the pseudo-label is generated using the same teacher model, a reasonable assumption is that in this case, the student model will have zero cross-entropy loss on the unlabeled data, and then the student model will eventually stop learning new things. Adding "noise" to the student model

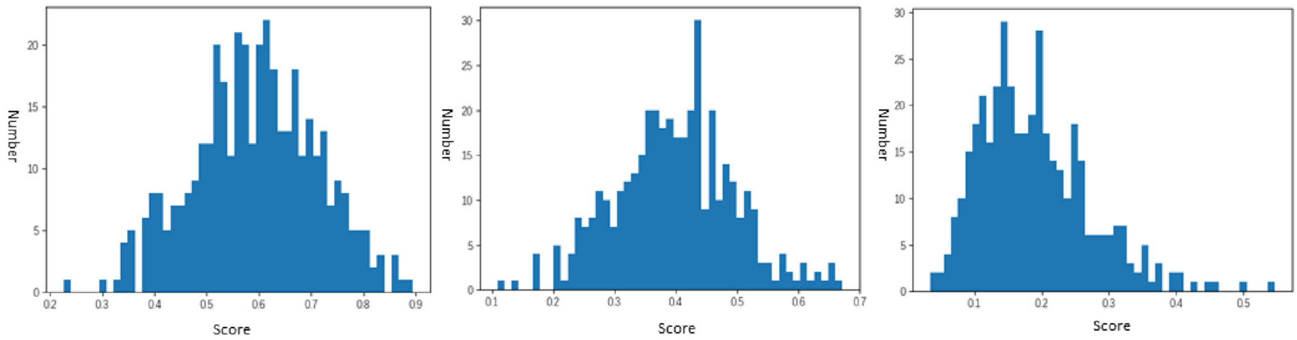


(a) Flame Roundness

(b) Flame Rectangularity

(c) Flame Contour roughness

Figure 6. Flame morphological matching score chart. (a) The flame roundness matching score chart, (b) The flame rectangularity matching score chart, (c) The flame contour roughness matching score chart.

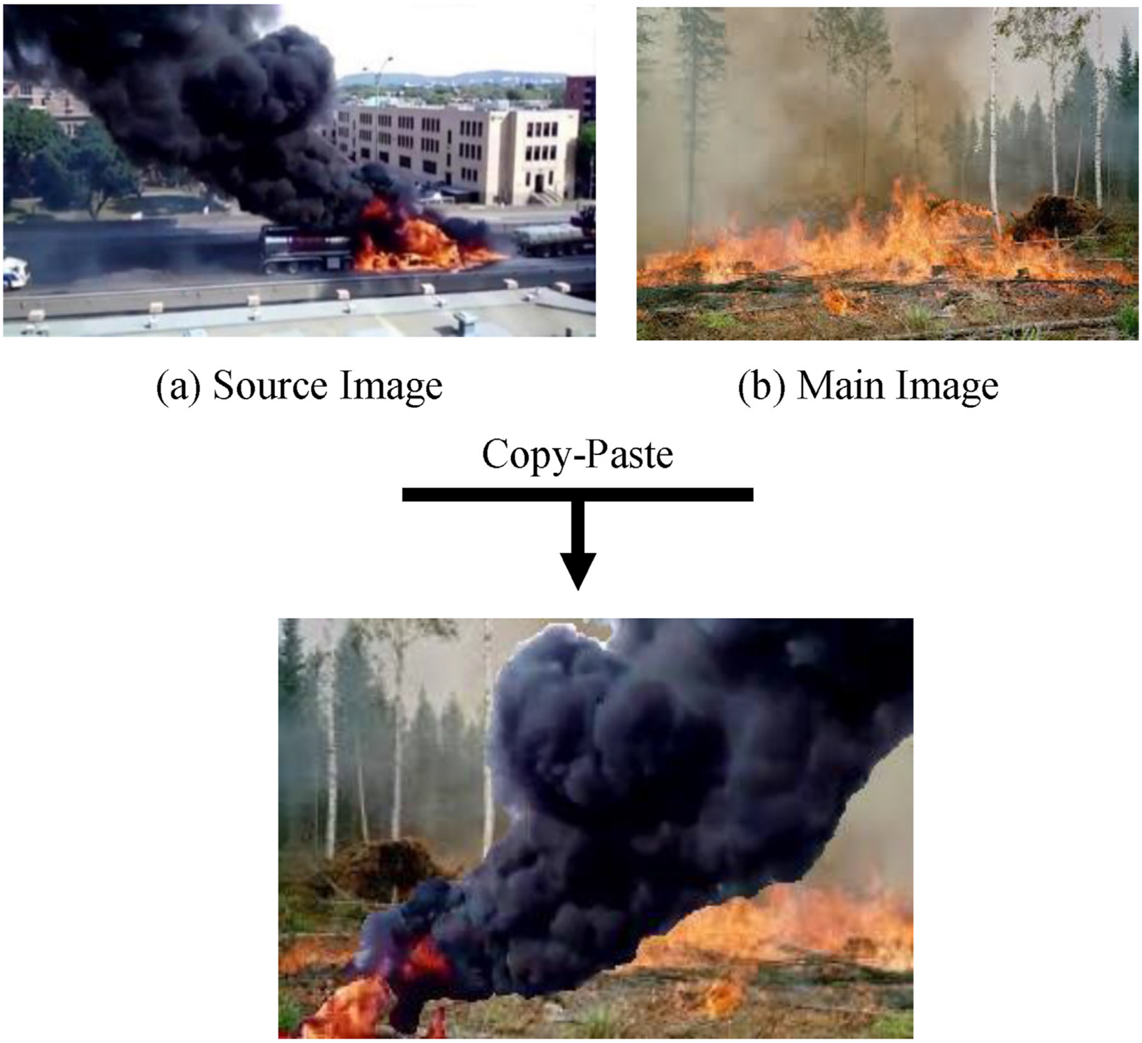


(a) Smoke Roundness (b) Smoke Rectangularity (c) Smoke Contour roughness

Figure 7. Smoke morphological matching score chart. (a) The smoke roundness matching score chart, (b) The smoke rectangularity matching score chart, (c) The smoke contour roughness matching score chart.

ensures that the student model learns more than just the teacher's model during training, which is an important reason for the huge improvement.

In this paper, a similar idea is adopted to introduce a more powerful data enhancement noise, and the "Copy-Paste" algorithm [31] with a balanced distribution of categories for labeled and pseudo-labeled datasets is used for image enhancement. As shown in Figure 8, The algorithm pastes in a new background image with different objects at different scales to obtain rich and novel training data. The object to be



(a) Source Image (b) Main Image

(c) Result

Figure 8. Copy-Paste algorithm example. (a) The source image with the target. (b) The main image and the target will be pasted onto the main image. (c) The result of the copy-paste process.

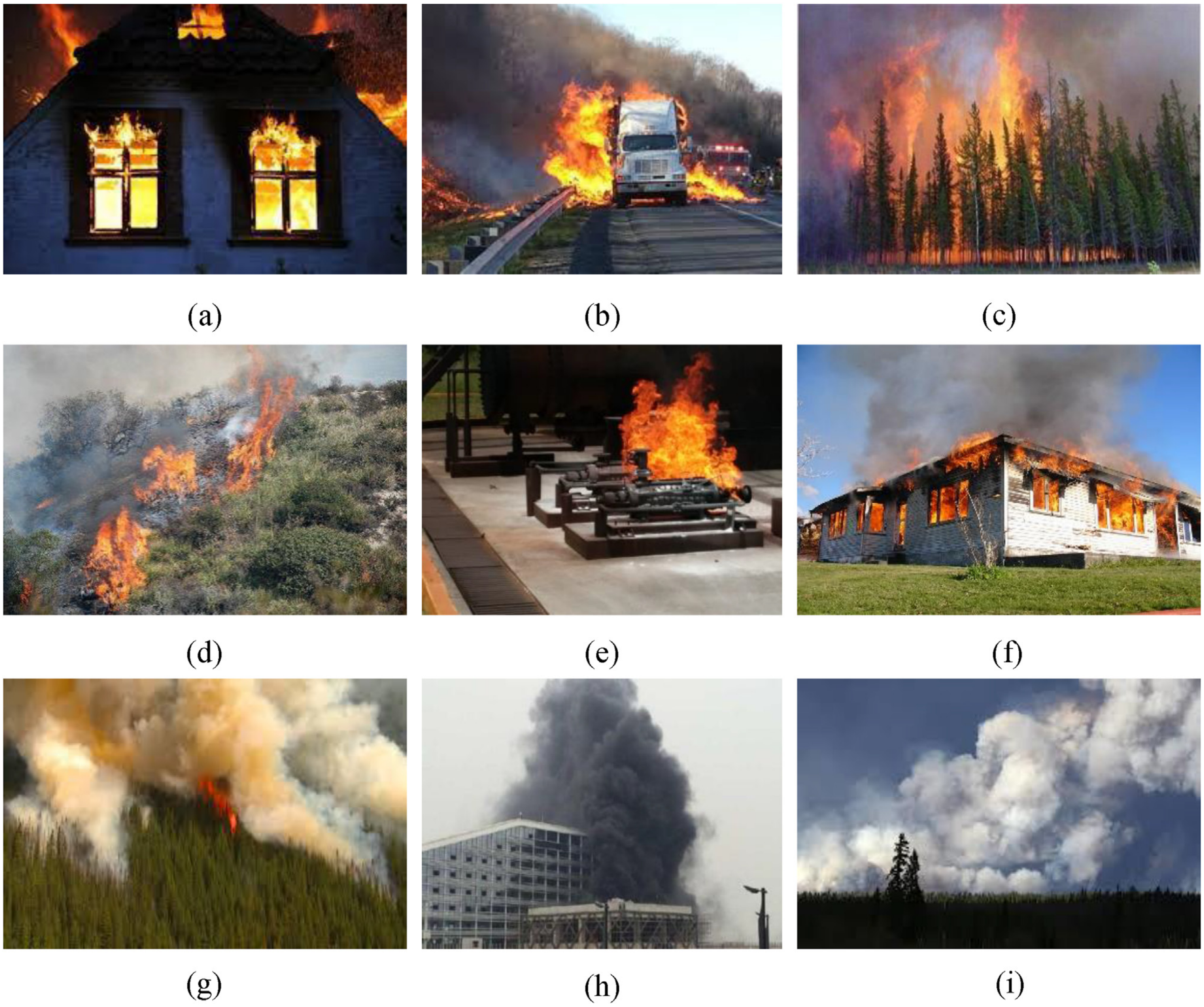


Figure 9. Examples of the collected fire images. (a)–(i) are examples of our collection of fire images. Contains a wide range of scenarios including urban, industrial and wilderness.

Table 1. Dataset image source table.

NAME	URL
Bowfire [33]	https://bitbucket.org/gbdi/bowfire-dataset/downloads/
Dunnings's [34]	https://collections.durham.ac.uk/files/r2d217qp536#.YuUZXHZBxPZ
fire-detection-image-dataset	https://github.com/cair/fire-detection-image-dataset
fire-smoke-detect-DATASET	https://github.com/gengyanlei/fire-smoke-detect-yolov4.

pasted is to pick the corresponding instance from one image and paste it randomly to another image. The objects copied and pasted are accurate to the pixel level. For each image in the pseudo-labeled dataset, a new image and corresponding label is generated by performing the hybrid paste method with the following formula once.

$$I_{new} = I_1 \times \alpha + I_2 \times (1 - \alpha) \tag{11}$$

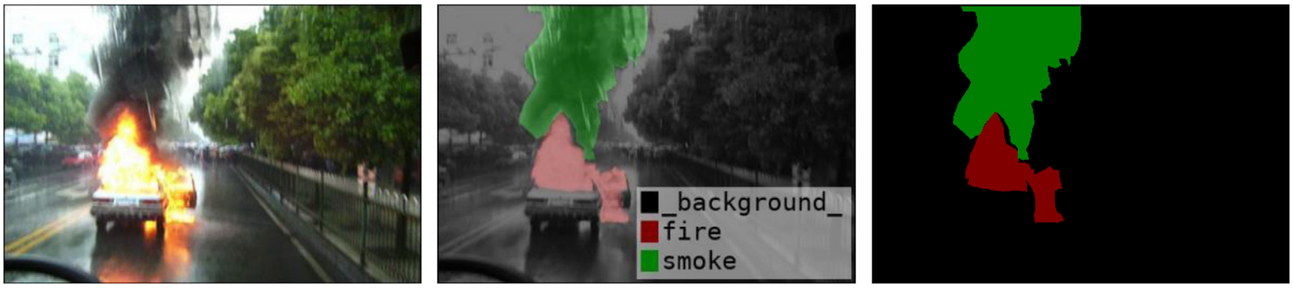
In the above Eq. (11), I_1 is the source image where the pasted object is located, I_2 is the main image, α is the mask corresponding to the target paste object in I_1, and the pixels of the mask part in I_1 is cropped out and pasted into I_2 to form a new image data. During the pasting process, a large-scale random scaling of 0.1–2.0 times is done for the cropped-out mask part and the main image.

Due to the large differences in the number of flame and smoke instances in the annotated dataset of this paper, to avoid the category differences being magnified during the paste process in the mixed paste, the category statistics of the annotated instances were first performed before selecting the source image I_1 and the target image I_2. Keep the alternative odds of flame and smoke constantly when randomly selecting the source image instances.

5. Experiment and results

5.1. Implementation and training details

We implemented the algorithm under the Pytorch deep learning framework in a Python environment, with training and testing done on a single NVIDIA RTX 2080Ti GPU. The model uses pre-trained Resnet weights from torchvision for the backbone. According to the training settings of the lightweight version of SOLOv2, the long side of the training input image is scaled to 512 and the short side is scaled randomly to [352,512], and the image scale of the source image is kept constant during the scaling process. The batch is set to 2 due to GPU memory limitations. Regular data enhancements such as random flip,



(a) Original image (b) Image annotation (c) Binarization annotation

Figure 10. Visualization of fire image annotation. (a) The original image. (b) The image with annotation. (c) The binarized annotation.

Table 2. Statistics of labeled dataset categories and instances.

Name	Number of Images	Number of Instances
Labeled Images	2238	3936
Flame Instance	2,164	3327
Smoke Instance	512	609

Table 3. Performance of different Instance Segmentation Network Models.

Name	mAP	AP_{50}	AP_{75}	FPS	Params
Mask RCNN [9]	50.3	75.6	61.7	11.5	44.17M
Mask-Scoring RCNN [35]	51.1	75.0	60.4	11.3	60.51M
YOLACT++ [19]	45.2	71.5	56.3	**28.3**	**35.29M**
SOLOv2 [12]	52.4	75.0	59.8	19.4	46.01M
SOLOv2-Light [12]	51.4	72.5	58.8	24.5	32.24M
Our Model*	53.8	77.6	61.1	22.6	39.18M
w/enhanced training*	**58.9**	**79.6**	**65.7**	22.6	39.18M

In Table 3, our Model* represents the performance of our enhanced SOLOv2 model on the test set. w/enhanced training* represents the performance of our model after a semi-supervised learning strategy based on fire features.

rotation, crop, and scale are synchronized with the image and label data during the training input. We used the Momentum-SGD optimizer with a learning rate of 0.001, which increases linearly to 0.001 within the initial 500 iterations of training for the warmup phase. Momentum is 0.9 and learning rate decay is 0.0001. The total number of training rounds was 120, and learning rate decay occurred at the 60th, 90th and 110th rounds.

5.2. Evaluation metrics

The Average Precision (AP) is a commonly used evaluation metric in target detection and instance segmentation tasks. First, based on calculating the Intersection over Union (IoU) between the true and predicted labels of each test image, we determine whether the target is the correct mask by whether the IoU indicator is greater than the Threshold (e.g., 0.5). TP (True Positive) and TN (True Negative) are the number of masks where the model prediction matches the true result, FP (False Positive) indicates the number of masks where the prediction is wrong, and FN (False Negative) indicates the number of masks not predicted.

$$Precision = TP / (TP + FP) \tag{12}$$

$$Recall = TP / (TP + FN) \tag{13}$$

$$AP = 1/11 \times \sum_{r\in 0,0.1,0.2,\dots,1} Precision(r) \tag{14}$$

$$mAP = 1/N \times \sum AP \tag{15}$$

Eqs. (12), (13), (14), and (15) are used to calculate the evaluation indicator mAP, The 11 in Eq. (14) represents 11 used recall values from 0, 0.1, 0.2 to 1. The N in Eq. (15) represents the number of defined category types. Use AP for mAP in MS COCO [32]. IoU Threshold = 0.5, 0.55, 0.6, ..., 0.95, and the AP values obtained under a total of ten IOU thresholds are averaged to obtain mAP for measuring model performance. where AP_{50} represents IoU Threshold = 0.5, AP_{75} represents IoU Threshold = 0.75.

5.3. Dataset

After researching various public datasets, we found most of the labels of current fire datasets are used for classification and detection, with few instances of segmentation labels. There are also a large number of publicly available fire images on the Internet. A total of 13,934 fire images were collected in this study, derived from other datasets for different purposes and public images on the Internet, and Figure 9 shows some of the images in the dataset. The dataset contains images of real fires occurring under different lighting conditions in different scenarios, such as forests, cities and factories, with complex background conditions and widely varying fire levels. This is a great challenge for our research work, and at the same time these complexities bring a wider variety of features to our model. The image sources in the dataset of this paper are shown in Table 1.

We performed manual pixel-level annotation on 2238 of the images. The annotated image labels were saved in JSON format and used as network training input, while the image and visualization labels are shown in Figure 10 below. The specific dataset annotation is shown in Table 2. There are 3936 annotated instances in 2238 images, of which 3327 are flame instances and only 609 are smoke instances. The distribution of the targets of the two types of instances is quite different, which poses a considerable challenge to our work. We hope to make full use of the features in the dataset in the task, while avoiding the effects of unbalanced distribution and using such a limited dataset to fully improve the model's capability. The fire instance segmentation dataset we constructed and a large amount of unlabeled data is publicly available on https://github.com/pomeloliv/Fire-Instance-Segmentation-Dataset.

5.4. Experimental results

The test set was derived from 500 images randomly segmented from the annotated dataset, and the test images were scaled to 512 pixels on

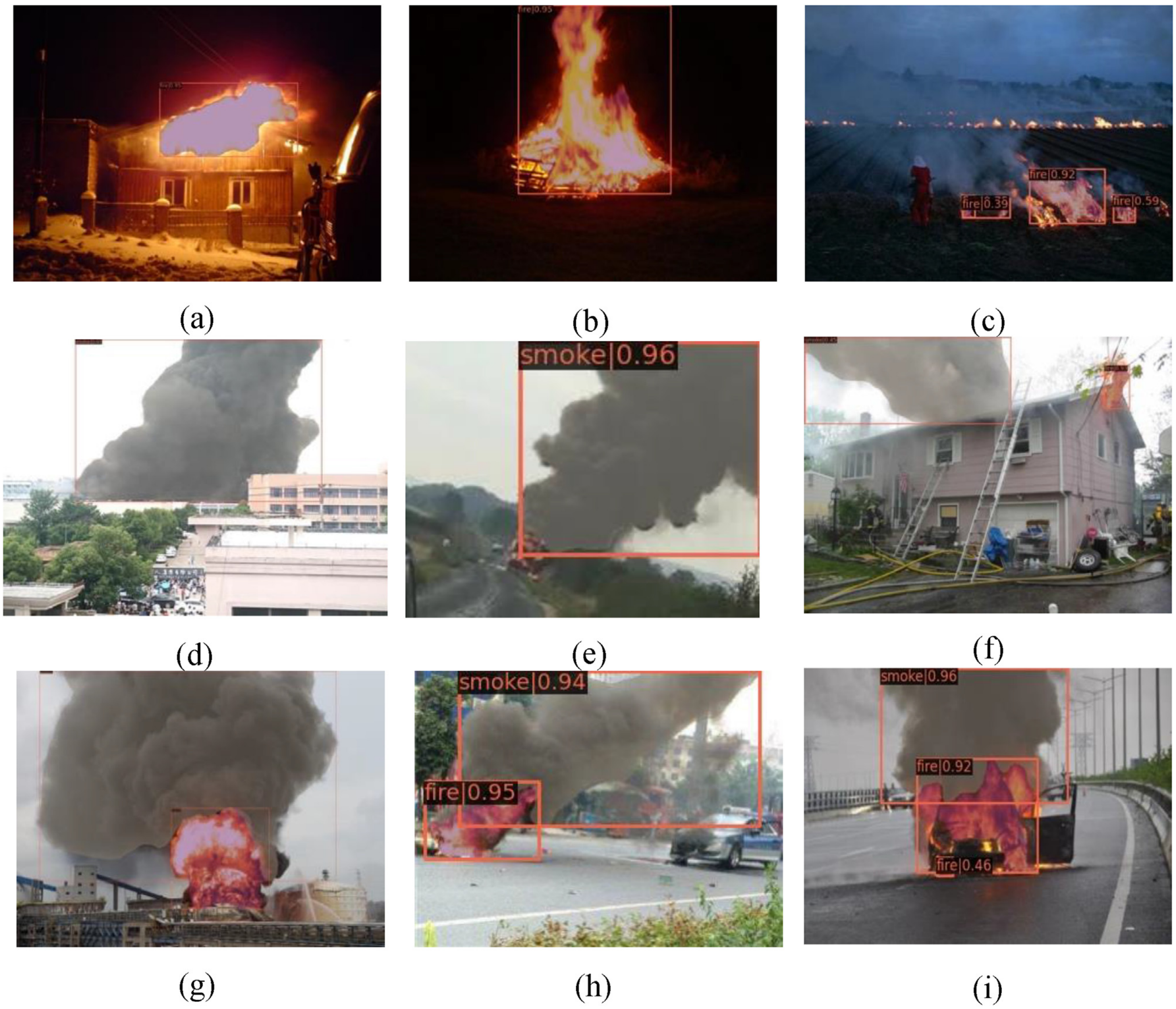


Figure 11. Examples of results. (a), (b), (c) are the results of a flame only fire scene. (d), (e) are the results of a smoke only fire scene. (f), (g), (h) and (i) are the results of fire scenes with both smoke and flames.

the short side. The test set contains a rich set of scenarios with different conditions of fire, which is a challenge for any model. Our comparison experiments between the improved SOLOv2 network proposed in this paper and other representative example segmentation networks yielded the results shown in Table 3, where the backbone feature extraction networks of all networks use ResNet50, where the computational speed (FPS) is the average value obtained from testing on 1000 images. In Table 3, Mask RCNN and Mask-Scoring RCNN are representative algorithms for two-stage instance segmentation. YOLACT++ is the real-time instance segmentation algorithm. SOLOv2 and SOLOv2-Light algorithms are the baseline models for the instance segmentation model in this paper.

Table 4. Enhanced SOLOv2 model ablation experiments.

DCNv2	CBAM	PAFPN	mAP	FPS	Params
			51.4	**24.4**	**32.24M**
✓			52.2	23.7	33.11M
	✓		52.5	23.4	34.77M
		✓	51.7	23.2	35.78M
✓	✓	✓	**53.8**	22.6	39.18M

In Table 4, the Baseline model for the experiments is SOLOv2-light. All improvements are based on this Baseline model.

Table 5. Comparison of semi-supervised training strategy and data enhancement algorithms.

Name	mAP	AP_{50}	AP_{75}
Flip + Rotation + Scale + Crop	53.8	77.6	61.1
Instaboost [36]	55.0	76.4	62.0
Mixup [37]	54.9	76.2	61.8
Copy-Paste [31]	55.1	76.6	62.2
Balanced Copy-Paste	55.3	77.0	62.3
Self-Training*	55.2	75.6	62.9
Noisy Student Training [30]	57.2	77.9	64.3
Noisy Student Training + Copy-Paste	58.3	78.2	65.5
Fire-based Self-Training* (ours)	57.8	78.3	65.2
Fire-based Self-Training + Balanced Copy-Paste (ours)	**58.9**	**79.6**	**65.7**

In Table 5, Self-Training* is the experiment where we follow the classical Self-Training paradigm with data enhancement removed compared to Noisy Student Training. Fire-based Self-Training* is our proposed fire-based feature-based semi-supervised training algorithm without data enhancement.

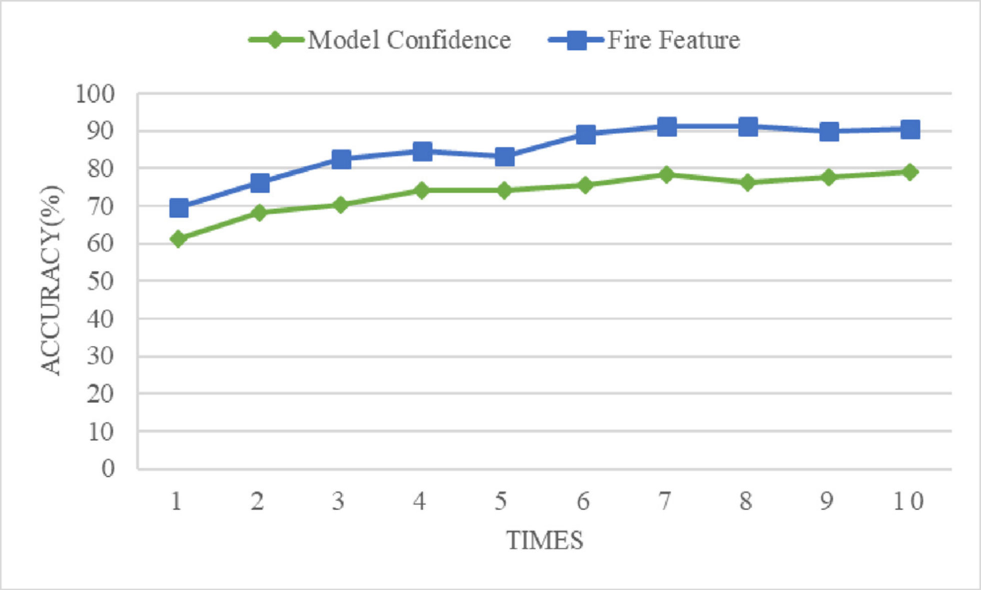


Figure 12. Line graph of pseudo-label accuracy for ten repetitions of training.

The enhanced SOLOv2 algorithm proposed in this study achieves 53.8% mAP accuracy and 22.1 FPS computing speed on the test set, combining the advantages of accuracy and efficiency. The enhanced SOLOv2 algorithm has a 1.4% mAP accuracy improvement and 2.7 FPS speed improvement compared to the original SOLOv2. Compared to the lightweight version SOLOv2-Light improves the accuracy by 2.4% mAP at the expense of 1.9 FPS speed. Compared with Mask RCNN and Mask-Scoring RCNN, the most representative two-stage algorithms, there are different degrees of improvement in speed and accuracy. Compared with the real-time instance segmentation algorithm YOLACT++, there is a speed difference, but the accuracy has a very significant improvement, with a difference of 8.6% mAP. Compared with the real-time instance segmentation algorithm YOLACT++, although it is slightly slower, the accuracy has a very significant improvement, with a difference of 8.6% mAP. After introducing the proposed enhanced training with semi-supervised learning based on fire features, the model improves the model accuracy to 58.9% mAP without introducing any inference computation, with AP50 and AP75 reaching 79.6% and 65.7%, respectively. Example results of the model output are shown in Figure 11.

To verify the performance improvement of the enhanced SOLOv2 model, we conducted an ablation experiment on the improvement points, as shown in Table 4. DCNv2, CBAM, and PAFPN make 0.8%, 1.1%, and 0.3% mAP improvements to the network, respectively. And the three together make a 2.4% mAP improvement to the network accuracy at the cost of losing 1.8 FPS. The experimental results effectively demonstrate that the enhanced SOLOv2 network can bring better network performance with less speed loss, which meets the requirements of this study.

In this paper, the proposed semi-supervised training strategy based on fire features and the copy-paste data enhancement algorithm after category equalization are compared with other methods, and all experiments are tested in the enhanced SOLOv2 network model proposed above. The evaluation metric still uses the Average Precision (mAP) and lists the AP when the more representative IoU thresholds are taken as 0.5 and 0.75. The obtained results are shown in Table 5.

According to the above table, the semi-supervised learning method based on fire features proposed in this study eventually achieved 58.9% mAP on the test set, with AP_{50} and AP_{75} reaching 79.6% and 65.7%, respectively. Our Copy-Paste algorithm for category equalization of the dataset delivers a performance gain of 0.2%–1.5% in a single model training compared to other image enhancement algorithms. Our proposed Fire-based Self-Training brings 0.5%–2.6% accuracy gain compared to classical Self-Training and Noisy Student Training algorithms. Fusing Noisy Student Training and Copy-Paste methods with our proposed semi-supervised training method based on fire features, our method finally achieves an increase of 0.6% mAP, with an increase of 1.3% mAP for AP_{50}.

To verify the effectiveness of our proposed pseudo-label selection method, we counted the accuracy of the pseudo-labels selected for the next training round during semi-supervised learning. We evaluate the pseudo-labels generated by ten rounds of training during semi-supervised learning. The training data for each round is obtained from the combination of the pseudo-labels generated from the previous round of training and the original training set. The statistical results in the ten repetitions of the training are shown in Figure 12.

In Figure 12, the green line represents that only the confidence level of the model output is used as the basis for judgment in the pseudo-label selection. The blue line represents the fire feature-based pseudo-label selection method proposed in this paper. The two methods differed in accuracy by 7.9%–15.3%. As can be seen from the figure, the method in this paper shows much better accuracy in the pseudo-label selection in training.

After the semi-supervised training, the training data grew by about 5 times compared to the initial one. Among them, 92.7% of the training data were correct. And after data augmentation, this gap will be further widened. This explains why the semi-supervised learning method proposed in this paper can be more effective.

6. Conclusion

We propose a fire instance segmentation method based on deep learning for pixel-level instance segmentation of flame and smoke targets. The method uses the improved SOLOv2 algorithm to segment instances of flame and smoke targets on an image or video. On the backbone network and SOLOv2-head, we replace some of the convolutional kernels with deformable convolutions. We add a CBAM attention mechanism between each block of the backbone network and replace the FPN with a bidirectional multiscale fusion PAFPN. These structural optimizations effectively enhance the model and improve accuracy at the expense of a small amount of computing efficiency. Since there is no publicly available dataset for fire instance segmentation, we collected a large number of unlabeled fire images from publicly available images on the Internet and other publicly available datasets. Limited by the huge workload required to label the data, we only accurately annotated some of them. We propose a semi-supervised learning method based on fire features for this type of dataset situation. To reduce the negative impact of erroneous pseudo-labels on the model training, in the pseudo-label generation phase of the teacher model during semi-supervised learning, the pseudo-labels are matched by the color and morphological features of flames and smoke for feature degree, and some images are screened for strong image enhancement before entering the next round of the training process of the student model. With the semi-supervised learning strategy, we further develop the potential of the model and improve the model accuracy under the condition of a limited dataset. In this research work, we constructed a fire dataset with rich features and information, which contains some images manually labeled with instance segmentation tags and a large number of unlabeled images, and we decided to make this dataset public in the hope of changing the current lack of data in the field of fire instance segmentation and providing assistance for future research work on fire instance segmentation.

Declarations

Author contribution statement

Guangmin Sun: Conceived and designed the experiments; Contributed reagents, materials, analysis tools or data.

Yuxuan Wen: Performed the experiments; wrote the paper.

Yu Li: Analyzed and interpreted the data.

Funding statement

This study was supported by the National Key R&D Program of China [2018YFF01012300] and National Natural Science Foundation of China [11527801, 41706201].

Data availability statement

The authors are unable or have chosen not to specify which data has been used.

Declaration of interest's statement

The authors declare no conflict of interest.

Additional information

No additional information is available for this paper.

References

[1] M. Ahrens, B. Evarts, Fire loss in the United States during 2019, National Fire Protect, Assoc (2020) 1–11.

[2] K. Bouabdellah, H. Noureddine, S. Larbi, Using wireless sensor networks for reliable forest fires detection[J], Procedia Comput. Sci. 19 (2013) 794–801.

[3] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks[J], Adv. Neural Inf. Process. Syst. 25 (2012) 1097–1105.

[4] H. Cruz, M. Eckert, J. Meneses, et al., Efficient forest fire detection index for application in unmanned aerial systems (UASs)[J], Sensors 16 (6) (2016) 893.

[5] W. Yuanbin. Smoke recognition based on machine vision[C]//international symposium on computer, IEEE, 2016, pp. 668–671.

[6] Y. Chen, W. Xu, J. Zuo, et al., The fire recognition algorithm using dynamic feature fusion and IV-SVM classifier[J], Cluster Comput. (10) (2018) 1–11.

[7] R. Girshick, J. Donahue, T. Darrell, et al., Rich feature hierarchies for accurat object detection and semantic segmentation [C]. Computer vision and pattern recognition, IEEE (2013) 580–587.

[8] R. Girshick, Fast R-CNN[C]//2015 IEEE international conference on computer vision (ICCV), IEEE (2015) 1440–1448.

[9] K. He, G. Gkioxari, P. Dollár, et al., Mask r-cnn[C], Proc. IEEE Int. Conf. Computer Vision (2017) 2961–2969.

[10] S. Ren, K. He, R. Girshick, et al., Faster r-cnn: towards real-time object detection with region proposal networks[J], Adv. Neural Inf. Process. Syst. 28 (2015) 91–99.

[11] X. Wang, T. Kong, C. Shen, et al., Solo: Segmenting Objects by locations[C]//European Conference on Computer Vision, Springer, Cham, 2020, pp. 649–665.

[12] X. Wang, R. Zhang, T. Kong, et al., SOLOv2: Dynamic and Fast Instance segmentation[J], 2020 arXiv preprint arXiv:2003.10152.

[13] K. Muhammad, J. Ahmad, Z. Lv, et al., Efficient deep CNN-based fire detection and localization in video surveillance applications[J], IEEE Tran. Sys. Man Cybernet.: Sys. 49 (7) (2019) 1419–1434.

[14] W. Mao, W. Wang, M. Jiang, et al., Fast flame recognition approach based on local feature filtering[J], J. Comput. Appl. 36 (10) (2016) 2907–2911.

[15] P. Li, W. Zhao, Image fire detection algorithms based on convolutional neural networks[J], Case Stud. Therm. Eng. 19 (2020), 100625.

[16] M.A. Akhloufi, R.B. Tokime, H. Elassady, Wildland Fires Detection and Segmentation Using Deep learning[C]//Pattern Recognition and Tracking Xxix, 10649, International Society for Optics and Photonics, 2018, p. 106490B.

[17] J. Sharma, O.C. Granmo, M. Goodwin, et al., Deep Convolutional Neural Networks for Fire Detection in images[C]//International Conference on Engineering Applications of Neural Networks, Springer, Cham, 2017, pp. 183–193.

[18] V.S. Bochkov, L.Y. Kataeva, wUUNET: advanced fully convolutional neural network for multiclass fire segmentation[J], Symmetry 13 (1) (2021) 98.

[19] D. Bolya, C. Zhou, F. Xiao, et al., Yolact++: better real-time instance segmentation [J], IEEE Trans. Pattern Anal. Mach. Intell. (2020) 1108–1121.

[20] X. Zhu, H. Hu, S. Lin, et al., Deformable convnets v2: more deformable, better results[C], Proc. IEEE/CVF Conf. Computer Vision Pattern Recogn. (2019) 9308–9316.

[21] J. Dai, H. Qi, Y. Xiong, et al., Deformable convolutional networks[C]//2017 IEEE international conference on computer vision (ICCV), IEEE Computer Society (2017) 764–773.

[22] W. Luo, Y. Li, R. Urtasun, et al., Understanding the effective receptive field in deep convolutional neural networks[C], Proc. 30th Int. Conf. Neural Info. Proc. Sys. (2016) 4905–4913.

[23] S. Woo, J. Park, J.Y. Lee, et al., Cbam: Convolutional Block Attention module[C]//Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 3–19.

[24] J. Hu, L. Shen, S. Albanie, et al., Squeeze-and-Excitation networks[J], IEEE Trans. Pattern Anal. Mach. Intell. 42 (8) (2019) 2011–2023.

[25] T.Y. Lin, P. Dollár, R. Girshick, et al., Feature pyramid networks for object detection [C]//2017 IEEE conference on computer vision and pattern recognition (CVPR), IEEE (2017) 936–944.

[26] S. Liu, L. Qi, H. Qin, et al., Path Aggregation Network for Instance segmentation [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8759–8768.

[27] T.H. Chen, Thou Ho, P.H. Wu, Y.C. Chiou, An early fire-detection method based on image processing[J], Int. Conf. Image Proc. IEEE 3 (2004) 1707–1710.

[28] F. Yuan, A fast accumulative motion orientation model based on integral image for video smoke detection[J], Pattern Recogn. Lett. 29 (7) (2008) 925–932.

[29] T.X. Truong, J.M. Kim, Fire flame detection in video sequences using multi-stage pattern recognition techniques[J], Eng. Appl. Artif. Intell. 25 (7) (2012) 1365–1372.

[30] Q. Xie, M.T. Luong, E. Hovy, et al., Self-training with Noisy student improves ImageNet classification[C]//2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR), IEEE (2020) 10684–10695.

[31] G. Ghiasi, Y. Cui, A. Srinivas, et al., Simple copy-paste is a strong data augmentation method for instance segmentation[C], Proc. IEEE/CVF Conf. Computer Vision Pattern Recognition (2021) 2918–2928.

[32] T.Y. Lin, M. Maire, S. Belongie, et al., Springer, Cham, 2014, pp. 740–755. Microsoft coco: Common objects in context[C]//European conference on computer vision.

[33] D.Y.T. Chino, L.P.S. Avalhais, J.F. Rodrigues, et al., Bowfire: detection of fire in still images by integrating pixel color and texture analysis[C]//2015 28th SIBGRAPI conference on graphics, patterns and images, IEEE (2015) 95–102.

[34] A.J. Dunnings, T.P. Breckon, Experimentally defined convolutional neural network architecture variants for non-temporal real-time fire detection[C]//2018 25th IEEE international conference on image processing (ICIP), IEEE (2018) 1558–1562.

[35] Z. Huang, L. Huang, Y. Gong, et al., Mask scoring R-CNN[C]//2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR), IEEE (2019) 6402–6411.

[36] Fang H S, Sun J, Wang R, et al. InstaBoost: boosting instance segmentation via probability map guided copy-pasting[C]//2019 IEEE/CVF international conference on computer vision (ICCV). IEEE, 682-691.

[37] H. Zhang, M. Cisse, Y.N. Dauphin, et al., Mixup: beyond Empirical Risk minimization[J], 2017 arXiv preprint arXiv:1710.09412.