sensors

MDPI

Article

Transformer-Based Fire Detection in Videos

Konstantina Mardani *, Nicholas Vretos and Petros Daras

Information Technologies Institute (ITI), Centre for Research and Technology Hellas (CERTH), 57001 Thessaloniki, Greece
* Correspondence: mardani@iti.gr

Abstract: Fire detection in videos forms a valuable feature in surveillance systems, as its utilization can prevent hazardous situations. The combination of an accurate and fast model is necessary for the effective confrontation of this significant task. In this work, a transformer-based network for the detection of fire in videos is proposed. It is an encoder–decoder architecture that consumes the current frame that is under examination, in order to compute attention scores. These scores denote which parts of the input frame are more relevant for the expected fire detection output. The model is capable of recognizing fire in video frames and specifying its exact location in the image plane in real-time, as can be seen in the experimental results, in the form of segmentation mask. The proposed methodology has been trained and evaluated for two computer vision tasks, the full-frame classification task (fire/no fire in frames) and the fire localization task. In comparison with the state-of-the-art models, the proposed method achieves outstanding results in both tasks, with 97% accuracy, 20.4 fps processing time, 0.02 false positive rate for fire localization, and 97% for f-score and recall metrics in the full-frame classification task.

Keywords: transformers; fire detection; segmentation; real-time; videos; fire localization; image classification

Citation: Mardani, K.; Vretos, N.; Daras, P. Transformer-Based Fire Detection in Videos. *Sensors* **2023**, *23*, 3035. https://doi.org/10.3390/s23063035

Academic Editor: Cosimo Distante

Received: 10 February 2023
Revised: 20 February 2023
Accepted: 7 March 2023
Published: 11 March 2023

1. Introduction

CCTV (closed-circuit television) control systems are always installed in industrial fields to protect the area from either illegal human activities or physical anomalies that could have catastrophic effects. Given the widespread use of these control systems, a number of automatic detection algorithms have been integrated into them to automatically alert in case of danger, such as fire [1]. Fire detection refers to the ability of control systems to identify and detect fire as quickly as possible. It can provide an early warning of fires, helping to minimize property damage and prevent the loss of life. Additionally, fire detection can be integrated into other smart building systems, such as fire suppression systems, to provide a comprehensive fire safety solution [2].

On the first steps of the research in the field, researchers suggested image processing techniques, using handcrafted features extracted from color, texture, contour, motion information, etc., either alone or in combination with machine learning approaches, by using classifiers such as SVMs, random forests, Bayes [3] for the fire detection task in videos. Recent research has made use of deep learning techniques, as they can automatically extract features and provide better results. Many approaches [4–7] in the literature for the fire detection in videos task use their own video datasets, which are not published for training and evaluation, or image datasets. Others although they attempt to solve the localization problem in videos, they evaluate their performance on the full-frame classification task (the presence of fire in frame or not), without evaluating their localization performance [8–11] and others prefer to present their results in fire localization task without comparing them with the results of prior works [7]. All these derive from the fact that there is not a complete fire detection in videos benchmark, annotated with segmentation masks in all frames, that contains an adequate amount of videos.

This paper introduces transformers for the fire detection task in videos. A transformer is a model that applies the self-attention mechanism, which measures the significance of each part of the input features (visual features, word embeddings, etc.), according to the problem one attempts to solve, in the form of weighting scores. Doing so, it utilizes the most relevant parts from the input [12]. It was initially applied to natural language processing (NLP) [13], but due to its success in that field, its use extended to computer vision tasks, such as image classification, object detection, and action recognition [12]. It is an encoder–decoder architecture, where the encoder consists of layers generating attention scores that indicate which parts of the input are relevant to each other. The decoder consists of layers that project the features in different and more suitable forms [12]. In NLP, the translation from one language to another is such an example.

The proposed approach is following the transformer from [14], but in a simplified form. This method removes the bipartite matching, class head, and bounding box head because they are unnecessary for the task and replaces the object queries, which are learnable embeddings, with the feature vector of the frame that is under process, in order to reduce model parameters, as this does not reduce the quality of the output. More specifically, both the encoder and the decoder consume the features from the current frame that is under process, in order to generate attention scores. For the fire localization task, these scores pass through a head, called mask head, that produces segmentation masks in the same way with [14]. Following [14], the mask head consists of a multi-head attention layer, in order to seek visual correlations between the features of the encoder–decoder outputs and a FPN architecture module for increasing the resolution of the predicted segmentation mask. For the full-frame classification task, the decoder attention scores pass through a linear layer, in order to predict the frame class (fire/no fire).

For the needs of the proposed method for the fire localization task, a segmentation mask dataset was created, based on the [15] dataset, by applying the SLIC [16] method, which divides the frame by superpixels. The superpixels that are totally enclosed in the ground truth bounding box rectangle were considered fire superpixels, and all the others were considered non-fire. This produces segmentation masks that specify the exact location and shape of the fire in the frames, according to the ground truth bounding boxes.

This study is divided into five sections. Section 1 highlights the need of integrated fire detection systems in surveillance systems, and the importance of accurate and fast models for their reliable performance. Additionally, the methodology is introduced and some prior works are referenced. Section 2 presents two categories of prior works in more detail: methodologies based on handcrafted features and deep learning-based methods. Section 3 describes, in detail, the proposed method, module by module. Section 4 presents the quantitative and qualitative results of the proposed method, in comparison with prior works, the implementation details, and discussion of the results. Finally, in Section 5, the study and its findings are summarized.

2. Related Work

As can be seen in the literature, the first efforts for fire detection in videos focused on the extraction of handcrafted features related to color, texture, motion information, etc, and their analysis, which provided remarkable results. However, deep learning-based approaches have gained field, due to their top performance.

2.1. Handcraft Features

In 2007, Celik et al. [17] used a generic color model in order to propose a fuzzy logic-based fire detection method. YCbCr color space was chosen for its deterministic characteristics in the Y, Cb, Cr color channels. For example, in a fire pixel, it is more possible that Y(x,y) will be greater than Cb(x,y) because the luminance information, which is related to the intensity of the pixel, is expected to be high. Three years later, Celik et al. proposed a combined color- and motion-based fire detection method, but this time in the CIE L*a*b* color space, in order to extract color related features. They used a threshold-based classifier

in order to subtract background from fire regions [18]. The same year, Zhou et al. [19] proposed a contour-based fire detection problem, which consists of three stages: (1) the candidate fire frame selection stage to select the most suspicious frames and remove the others, (2) the fire region selection stage to detect the fire pixels by fire-region selection rules, and (3) the contour-based fire decision, where it performs four operations (dilation, erosion, mini region erasing, and Canny edge detection) on all fire regions, in order to detect the exact fire contours, and finally, fire decision rules based on three characteristics (i.e., area, perimeter, and roundness of the flame contours) to determine whether a fire occurs in the video or not. The next year, Chenebert et al. [20] proposed a non-temporal texture-driven method for fire detection, where they used texture- and colour-based feature descriptors as input to train classifiers, such as neural networks and regression trees. For texture descriptors, they used hue and saturation from the HSV color space. This model provided competitive results, until now. A fire detection method based on shape variation, color, and motion was proposed by Foggia et al. [21], which combined with a multi-expert system and weighted voting, in order to manage the high dimensional feature vectors derived from the color and motion characteristics of fire. Additionally, they proposed a descriptor based on a bag-of-words approach for the motion representation. More recently, Kong et al. [11] proposed the use of logistic regression and temporal smoothing. More concretely, for the determination of a candidate fire region, they took into consideration the color component ratio, and they obtained the motion cue of fire from background subtraction. For size, motion, and color information, they used logistic regression. In order to differentiate fire, fire-like objects, and background, they used distribution and Chroma ratio (Cb/Cr). They concluded that temporal smoothing reduces false alarms. Han [10] proposed the use of motion and color features for their fire detection approach. They used the Gaussian mixture model for background subtraction and a combination of RGB, YUV, and HSI color spaces for the multi-color detection. With the help of these two steps they identify the fire areas in the image plane. Gong et al. [22] also proposed motion and color features for their fire detection method, but they combined a motion detector and a RGB color model to screen the candidate fire pixels. They proposed a fire centroid stabilization method based on spatio-temporal relation, in order to calculate the centroids of fire in all frames and leverage this temporal information for the fire localization task, after passing through a support vector machine.

2.2. Deep Learning Approaches

Although many deep learning models have been introduced for the detection and localization of fire regions in videos, there are few that specify this region in the pixel-level classification form, due to the lack of adequate datasets both for training and evaluation. In 2016, Zhang et al. [23] proposed a CNN-based method for forest fire detection by passing through the fire classifier image patches. The approach starts by checking for fire presence in the full image, in order to continue to the next step, which localizes the image patches in the image plane. They also built a fire detection dataset with patch-level annotations. In 2018, Zhao et al. [24] attempted to tackle the difficulties that arise when the frames are obtained from moving UAVs (Unmanned Aerial Vehicles). They proposed a saliency detection method combined with a logistic regression classifier. The saliency detector extracts the region of interest in which the presence of fire is possible, and then two logistic regression classifiers specify if the region of interest forms a fire region or smoke region using color and texture features. Their method prevents feature losses by using fixed image size for training. In 2019, Aslan et al. [5] proposed deep convolutional generative adversarial neural networks (DCGANs) for the fire detection in videos. Their approach demands two-stage training. In the first stage, they train spatio-temporal images (temporal slice images) and noise vectors, and in the second stage, they train the discriminator without the generator. The same year, Yu and Chen [6] proposed a method that consists of three stages: In the first stage, it preprocesses the video by combining motion and color feature detection, in order to extract fire regions. Therefore, the suspected region passes through

a spatial convolutional neural network, and the stacked optical flow of the fire region passes through a temporal convolutional neural network. Finally, their results were fused to give the ultimate prediction. Muhammad et al. [4] proposed a computational efficient CNN architecture inspired by the SqueezeNet, in order to be more easily applicable in real-life situations. They used smaller convolutional kernels and no dense or fully connected layers at all. However, they achieved comparable results with other architectures (AlexNet, VGG). In 2018, Dunnings and Breckon [25] experimented with three low-complexity CNN architectures (AlexNet, VGG-16, InceptionV1) providing, as input, superipixels extracted from frames to be classified as fire or non-fire. The InceptionV1 model provided the best results. Next year, Samarth et al. [9] proposed InceptionV3 and InceptionV4, inspired by InceptionV1, ResNet, and EfficientNet, with modifications in kernel sizes and the number of convolution layers. InceptionV4 produced the most accurate results. Another approach based on superpixels was proposed in 2020 by Thomson et al. [8], but this time, they experimented with NasNet-A-Mobile and ShuffleNetV2 architectures, but in a simplified form. ShuffleNetV2 gained performance compared to all other superpixel methods. Kim and Lee [7] introduced faster region-based convolutional neural networks (R-CNNs) for detecting possible fire regions based on their spatial features. The features included within the bounding boxes of sequential frames are aggregated and pass through a long short-term memory (LSTM) for classifying them as fire or not fire in a short-term period. The final decision arises from a majority voting after combining the obtained short-term decisions.

3. Materials and Methods

The proposed method introduces a fire detection method for videos based on transformers. The transformer encoder consumes the features of the frame that are under examination and outputs attention scores that indicate the correlations between the features of the same frame. The transformer decoder consumes features from the current frame, as well as the output of the encoder, in order to compute attention scores between the two representations. For the fire localization task, on top of the decoder, there is a mask head, which predicts a segmentation mask that localizes and describes the fire in the current frame, as can be seen in Figure 1. For the full-frame classification task, on top of the decoder, there is a class head with a linear layer for binary classification (fire/no fire), as can be seen in Figure 2. Therefore, the model consists of three components: a convolutional backbone, a transformer encoder–decoder, and a mask head for predicting segmentation masks or a linear layer for predicting the frame class.

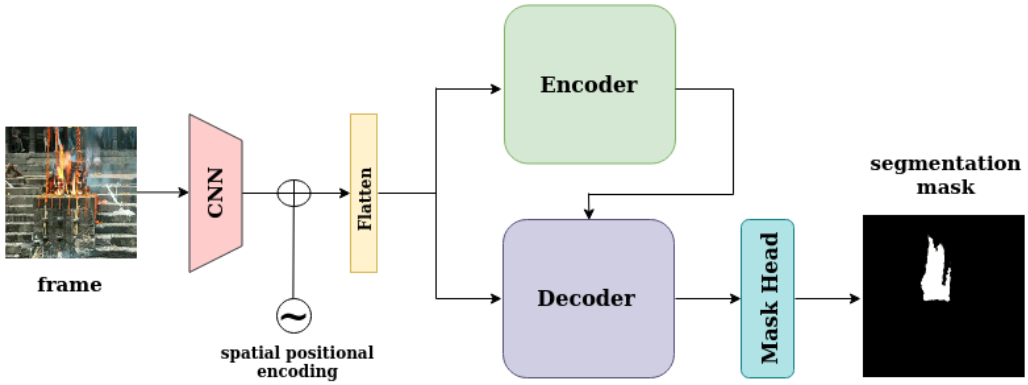

Figure 1. This figure depicts model architecture for the in-frame fire localization task. Both the encoder and the decoder consume the current frame that is under examination and output attention scores. These attention scores pass through the mask head in order to produce segmentation mask that localizes fire in the image plane.

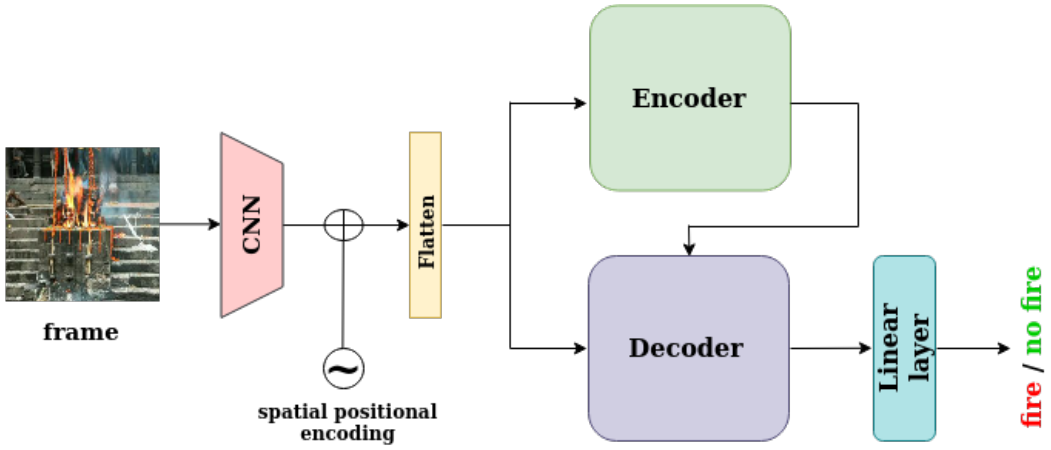

Figure 2. This figure depicts model architecture for the full-frame classification task. Both the encoder and the decoder consume the current frame that is under examination and output attention scores. The output of the decoder pass through a linear layer in order to predict the frame class (fire/no fire).

3.1. Encoder/Decoder Backbone

The model uses ResNet-101 to extract features from frames, in order to pass them through the encoder/decoder. It is fed with a frame $I \in \mathbb{R}^{3\times H\times W}$, and it generates a feature map $I_f \in \mathbb{R}^{C\times \frac{H}{32}\times \frac{W}{32}}$, where H and W are the height and width of the input frames, respectively. The output of the backbone then passes through an 1×1 convolutional layer to reduce the number of channels from C to d $(C = 2048, d = 128)$ for computational efficiency. As both the encoder and the decoder expect sequences for input, the feature map is flattened to a feature vector $V \in \mathbb{R}^{d\times \frac{H}{32}\times \frac{W}{32}}$ as can be seen in Figure 3.

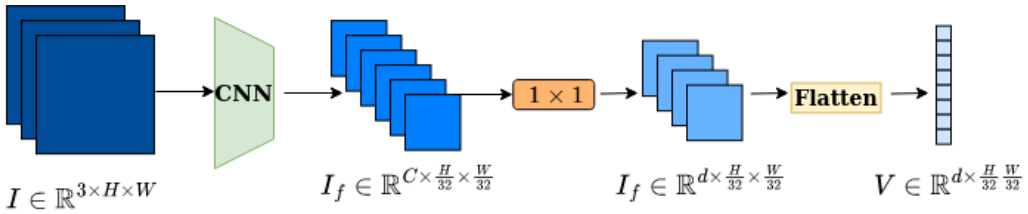

Figure 3. In this figure the encoder/decoder input pipeline is depicted. Both to the encoder and to the decoder, the frame passes through the same transformations until it is fed into the encoder/decoder blocks. Encoder and decoder inputs have the same dimensions.

3.2. Transformer Encoder

Similarly to the architecture of the transformer in [14], each encoder layer consists of a multi-head self-attention module with M heads, where $M = 8$, and a feed forward network (FFN). The encoder consists of N identical encoder layers, where $N = 4$, as can be seen in Figure 4. Due to the fact that the transformer architecture is permutation-invariant, a fixed positional encoding [26,27] is added both to the feature vector and to the input of each attention layer. The output of the encoder is attention scores with the same shape as its input, but in different representation.

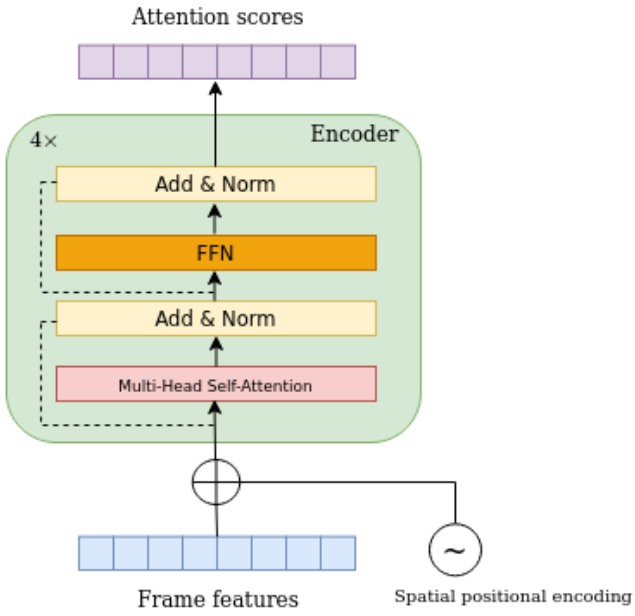

Figure 4. In this figure, the encoder layer architecture is depicted. The encoder of the model consists of four identical, consecutive layers.

3.3. Transformer Decoder

The same feature vector with the encoder passes through the decoder. Each decoder layer consists of two multi-head self-attention layers with M heads, where $M = 8$, and a feed forward network (FFN). The decoder contains four identical and consecutive decoder layers. Due to the fact that the transformer architecture is permutation-invariant, a fixed positional encoding [26,27] is added both to the feature vector and to the input of each attention layer. Additionally, the decoder consumes the obtained encoder attention scores, as can be seen in Figure 5, which pass through a multi-head self-attention layer fused with the decoder input, in order to obtain visual correlations between the two representations.

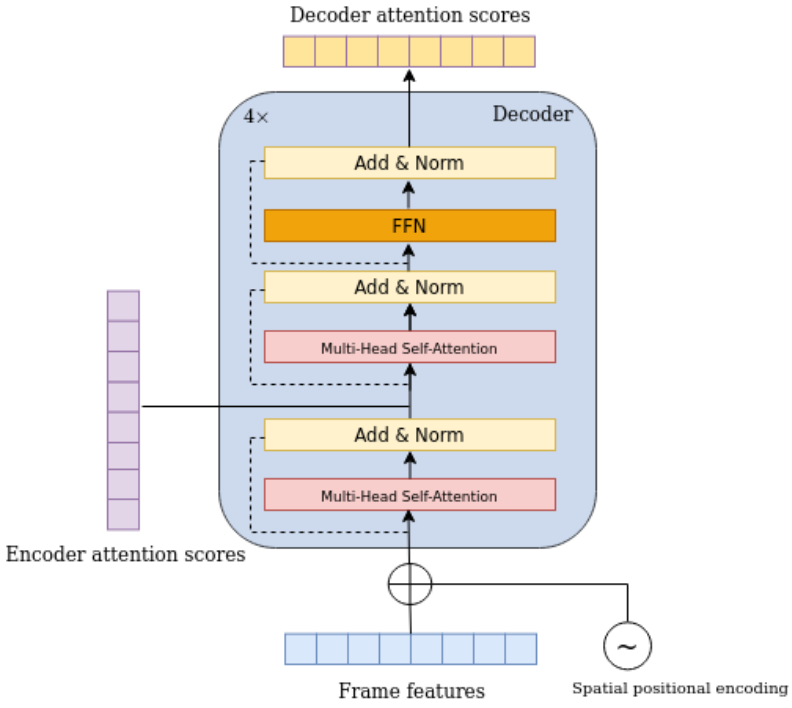

Figure 5. In this figure, the decoder layer architecture is depicted. The decoder of the model consists of four identical, consecutive layers.

3.4. Fixed Positional Encodings

Positional encodings in computer vision, are a method used in the transformer architecture to allow the model to understand the relative position of elements in an image. The fixed positional encodings are pre-calculated, numerical values that are added to the input representation of each element in the image. These values encode the relative position of the elements, allowing the model to distinguish between elements that are near or far from each other. The fixed positional encodings are learned during the training process and are used to enhance the representation of the input image, thus improving the accuracy and performance of the model. They are added to the input representation before the model begins processing the data, allowing the model to take into account the relative position of the elements in the image. In the proposed method, the sinusoidal positional encodings are used. This encoding method involves adding a sinusoidal function to the image elements, based on their position in the image [13,28].

3.5. Mask Head

For the fire localization task, on top of the decoder output, the class head and the bounding box head that DETR [14] uses for predicting bounding box coordinates and object classes are removed. These two heads are unnecessary in the proposed method for the localization task, as the model predicts only segmentation masks. For the prediction, the output of the decoder and the output of the encoder pass through a multi-head attention layer with M heads, where $M = 8$, to seek correlations between their features. This generates M attention heatmaps in low resolution. For the increase of the resolution for the final prediction an FPN (feature pyramid network) is used, as can be seen in Figure 6.

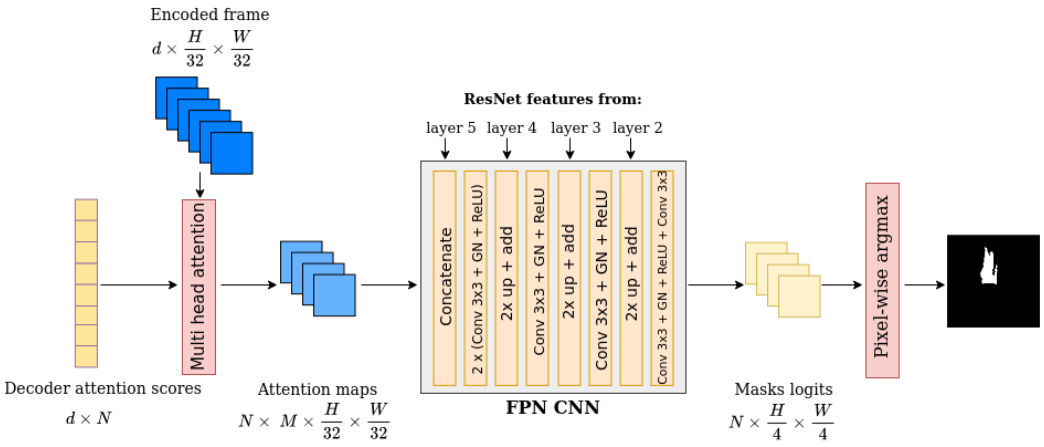

Figure 6. In this figure, the mask head architecture is depicted. All the steps from the decoder attention scores, until the final segmentation mask, are illustrated.

3.6. Class Head

For the full-frame classification task, on top of the decoder output, the mask head and the bounding box head that DETR [14] uses for predicting bounding box coordinates and panoptic segmentations are removed. These two heads are unnecessary in the proposed method for the classification task, as the model predicts only the frame class (fire/no fire). For the prediction, the output of the decoder pass through a linear layer, which is a fully connected layer (FCN), where every input neuron is connected to every output neuron. The number of the input neurons is 128, and the number of output neurons is 2, equal to the number of classes (fire/no fire).

3.7. Losses

3.7.1. Fire Localization

This proposed method for fire localization is trained using for loss function the combination of *DICE loss* and *focal loss*, similarly to the DETR [14] for panoptic segmentation. The loss function can be written as:

$$L = L_{DICE} + L_{focal} \tag{1}$$

DICE loss [29] is a measure of overlap between two sets. In computer vision, it is widely used to measure the similarity between two images. If p is the prediction binary mask and t is the target mask, L_{DICE} is defined as:

$$L_{DICE} = 1 - \frac{2t\sigma(p) + 1}{\sigma(p) + t + 1} \tag{2}$$

where σ is the sigmoid function.

Focal loss [30] on the other hand addresses the class imbalance, by downweighting the contribution of the easy examples during training, while focusing the model's attention on hard examples.

$$L_{focal} = -(1 - p_c)^{\gamma} log(p_c) \tag{3}$$

where p_c is the predicted class probability and γ is the focusing parameter ($\gamma = 0.25$).

3.7.2. Full-Frame Classification

The proposed method for full-frame classification is trained using for loss function the cross-entropy loss, the most commonly used loss function for categorical models.

$$L_{CE} = -\sum_{c=1}^{n} t_c log(p_c) \tag{4}$$

where n is the number of classes ($n = 2$), t_c is the ground-truth, and p_c is the probability of the cth class.

4. Experiments

In this section, the experimental results are presented. In the first subsection, the datasets and metrics that have been used for training and evaluation are detailed. Subsequently, the implementation details and the qualitative results are portrayed, and finally, the quantitative results are compared with the state-of-the-art methods.

4.1. Datasets

For the fire localization task, the model has been trained and evaluated on the furg fire dataset [15], the only video-based fire detection benchmark that consists of fire and non-fire image sequences and bounding box annotations for each frame. This dataset contains 24 videos published on the internet with 28,022 frames, under the Creative Commons 3.0 license. For the training and the evaluation of the proposed method, the dataset is split into training and testing sets, similarly to in [8,9,25]. The used dataset consists of 26,339 full-frame images with 14,266 fire images and 12,073 non-fire images. The training set consists of 18,590 images and the testing set consists of 2211 images.

Additionally, the proposed method has been trained and evaluated on the dataset from work [25] for the full-frame classification task. It is a combination of the [20] dataset, the [25] dataset, and videos from public sources (YouTube).

It contains 26,339 images, with 14,266 images of fire and 12,073 of non-fire images. The training set consists of 23,408 images, and the testing set consists of 2,931 images.

4.2. Evaluation Metrics

For evaluation purposes, the proposed method was compared to prior works based on their ability to address two different problems: (a) the full-frame classification problem

(if fire exists in the frame or not) and (b) the fire localization within the frame problem (predicting the exact location of fire in the image).

4.2.1. Full-Frame Classification

For the evaluation of the proposed method on the full-frame classification problem, precision (P/PPV), true positive rate or recall (TPR), F1-score (F1), false positive rate (FPR), accuracy (A), and frames per second (fps), according to [15], were used to compare the different methods on two different datasets.

4.2.2. Fire Localization

For the evaluation of the proposed method on the in-frame localization task, precision (P/PPV), true positive rate or recall (TPR), F1-score (F1), similarity (S), and frames per second (fps), according to [15], were used to compare the different methods.

4.3. Implementation Details

For the needs of the proposed method for the fire localization task, a segmentation mask dataset was created, based on [15]. A method called SLIC [16], which generates superpixels by clustering pixels based on their color similarity and their proximity in image plane, was applied at each frame, in order to divide it into superpixels. The superpixels that were entirely included into the ground truth bounding box rectangle (red box in the top right image of Figure 7) were labeled as fire (1), and all the other superpixels were labeled as non-fire (0). This pipeline is depicted in Figure 7. This produces segmentation masks that specify the exact location and shape of the fire in the frames, according to the ground truth bounding boxes, as can be seen in Figure 7.

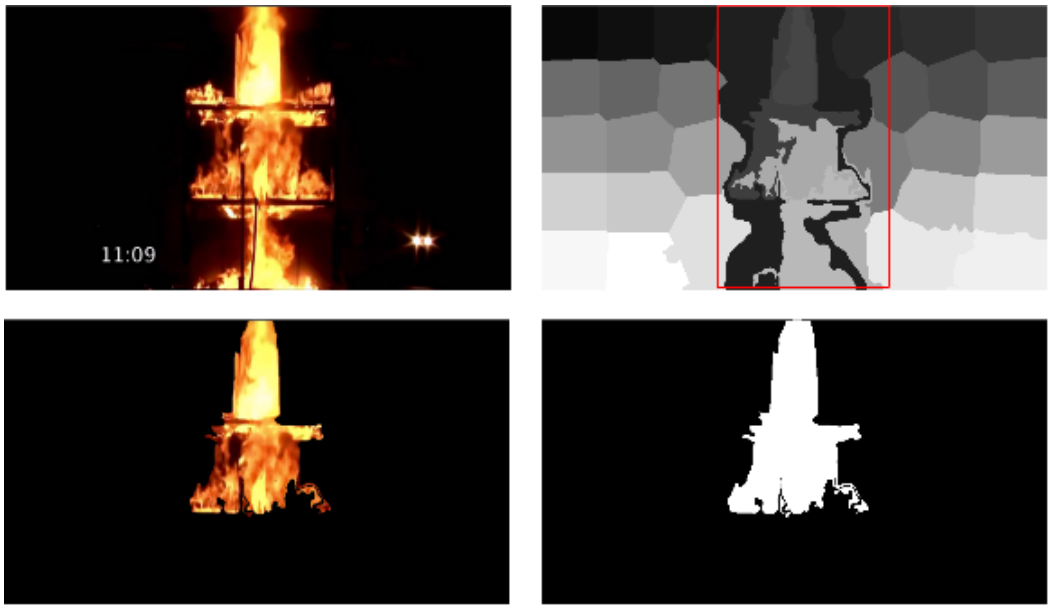

Figure 7. This figure illustrates the processing steps, in order to obtain the segmentation dataset from the bounding box annotations.

The proposed method was implemented in Pytorch 1.5.1 version [31] and trained using AdamW optimizer [32], with a weight decay at 0.0001, learning rate at 0.00001, and learning rate for backbone at 0.00001 too. The backbone was an ImageNet-pretrained ResNet-101 model [33] with frozen batchnorm layers. The input size was set to 480×854, and the batch size to 1. For the in-frame fire localization task, the probability threshold for producing the segmentation mask was set to 0.6. This means that, if a pixel had probability higher than 0.6, it was labeled as fire pixel, else as a non-fire pixel. The model was trained on GeForce RTX 2070 and achieved top results on epoch 6 for both tasks.

4.4. Evaluation Results and Discussions

4.4.1. Fire Localization Model

According to Tables 1 and 2, the proposed method outperforms all the prior works, either on speed or on accuracy on the furg fire dataset [15]. This model has been evaluated for both full-frame classification tasks by considering that the frame contains fire if pixels with value 1 existed in the binary segmentation mask and fire localization task. In Table 1, the full-frame fire classification results are presented. All the metric values given from the proposed method, except of TPR and fps, are top compared to the prior works. More concretely, the FPR metric drops about 66%, compared to the next lowest value of that achieved via the InceptionV4-OnFire. F, P, and A are as high as the top results of NasNet-A-OnFire at 0.98, 0.99, and 0.97, respectively. The proposed method achieves top performance, as regards the speed, with a value at 20.4 fps, which is 13% higher than the ShuffleNetV2-OnFire model. The TPR metric has not been surpassed since 2011. In Table 2, the in-frame fire localization results are presented. As it seems, the proposed method outperforms the prior works in precision and similarity metrics. In more detail, P and S are 0.95, and they achieve an increase of 2% and 19% from the prior top results, respectively. However, TPR and F values are lower than the prior methods and more concretely about 23% and 11%, respectively, from the top method. This low TPR value is connected with the training dataset. As mentioned above, the dataset is not human-annotated, and this adds an error to the ground truths. In order to avoid the misclassification of non-fire pixels to fire pixels, and consequentially, a high FPR value, some fire pixels close to the fire boundaries were labeled as non-fire. So, the model learns to miss some of the fire pixels that are close to these boundaries. For improving the TPR metric for the in-frame localization problem, dilation is applied to the predicted segmentation mask. However, this dilation causes a reduction of the precision and similarity metrics. Finally, the proposed method is much faster than the two prior works in the in-frame localization problem, as can be seen in Table 1. Additionaly, Figure 8, depicts the model's accurate localization performance on some sample frames from testing set.

Table 1. This table presents the quantitative results of the proposed method on the furg fire dataset [15], in comparison with prior works for the full-frame classification problem. The bolded values highlight each metric's top value

Models	TPR	FPR	F	P	A	fps
Chenebert, A., et al. [20]	**0.99**	0.28	0.92	0.86	0.89	0.16
InceptionV1-OnFire	0.92	0.17	0.90	0.88	0.89	8.4
InceptionV3-OnFire	0.94	0.07	0.94	0.93	0.94	13.8
InceptionV4-OnFire	0.94	0.06	0.94	0.94	0.94	12
NasNet-A-OnFire	0.98	0.15	0.98	0.99	0.97	5
ShuffleNetV2-OnFire	0.94	0.08	0.97	**0.99**	**0.97**	18
Ours	0.97	**0.02**	**0.98**	**0.99**	**0.97**	**20.4**

Table 2. This table presents the quantitative results of the proposed method on the furg fire dataset [15], in comparison with prior works for the in-frame fire localization problem. The bolded values highlight each metric's top value.

Models	TPR	F	P	S
Chenebert, A., et al. [20]	**0.98**	**0.90**	0.93	0.80
InceptionV1-OnFire	0.92	0.88	0.84	0.78
Ours	0.75	0.80	**0.95**	**0.95**
Ours+dilation(3×3, 4iter)	0.78	0.81	**0.93**	**0.94**
Ours+dilation(3×3, 5iter)	0.79	0.81	**0.93**	**0.94**
Ours+dilation(3×3, 6iter)	0.80	0.82	**0.93**	**0.94**
Ours+dilation(3×3, 7iter)	0.80	0.82	0.92	**0.94**

Figure 8. This figure illustrates qualitative results from the testing set. Red rectangle: prediction, blue rectangle: ground truth.

4.4.2. Full-Frame Classification Model

In Table 3, the quantitative results of the proposed method for the full-frame classification task and the state-of-the-art methods on the [25] dataset are presented. According to this, all the metrics given from the proposed method are surpassing the prior works, except the FPR metric. More concretely, the TPR and F metrics are 1% higher than the top prior works, the A and P metrics are equal to the top prior works, and the FPR metric is 33.3% higher than the top prior work. According to this, the proposed full-frame classification method is the only method that gives the highest values in almost all metrics.

Table 3. This table presents the quantitative results of the proposed method on the [25] dataset, in comparison with prior works for the full-frame fire classification problem. The bolded values highlight each metric's top value.

Models	TPR	FPR	F	P	A
InceptionV1-OnFire	0.96	0.10	0.94	0.93	0.93
InceptionV3-OnFire	0.95	0.07	0.95	0.95	0.94
InceptionV4-OnFire	0.95	0.04	0.96	0.97	0.96
NasNet-A-OnFire	0.92	**0.03**	0.94	0.96	0.95
ShuffleNetV2-OnFire	0.93	0.05	0.94	0.94	0.95
Ours	**0.97**	0.04	**0.97**	**0.97**	**0.96**

5. Conclusions

In this work, a video fire detection method based on transformers is presented. It is a simplified architecture of DETR [14], and it is capable of recognizing fire in frames fast and accurately with a very low false positive rate. The paper presents a full-frame classification model that achieves top performance in almost all metrics on the [25] dataset and an in-frame localization model that achieves top performance in almost all metrics for

the full-frame classification task on the [15] dataset. Moreover, it achieves top performance for precision and similarity metrics and top speed, compared to the prior works for the in-frame fire localization task on the [15] dataset. The creation of a human-annotated segmentation dataset with fire videos would increase the FPR metric in this work, but it would also be a big step for the research community in the field of fire detection in videos.

Author Contributions: Methodology, K.M.; Writing—original draft, K.M.; Supervision, N.V. and P.D. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the European Commission (INTREPID, intelligent toolkit for reconnaissance and assessment in perilous incidents), under grant No. 883345.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets that were used for training and evaluation [15,25] can be found at https://collections.durham.ac.uk/files/r2d217qp536 (accessed on 3 November 2022).

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

References

1. Geetha, S.; Abhishek, C.; Akshayanat, C. Machine vision based fire detection techniques: A survey. *Fire Technol.* **2021**, *57*, 591–623. [CrossRef]
2. Yuan, F. An integrated fire detection and suppression system based on widely available video surveillance. *Mach. Vis. Appl.* **2010**, *21*, 941–948. [CrossRef]
3. Gaur, A.; Singh, A.; Kumar, A.; Kumar, A.; Kapoor, K. Video Flame and Smoke Based Fire Detection Algorithms: A Literature Review. *Fire Technol.* **2020**, *56*, 1943–1980. [CrossRef]
4. Muhammad, K.; Ahmad, J.; Lv, Z.; Bellavista, P.; Yang, P.; Baik, S.W. Efficient deep CNN-based fire detection and localization in video surveillance applications. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *49*, 1419–1434. [CrossRef]
5. Aslan, S.; Güdükbay, U.; Töreyin, B.U.; Cetin, A.E. Deep convolutional generative adversarial networks based flame detection in video. *arXiv* **2019**, arXiv:1902.01824.
6. Yu, N.; Chen, Y. Video flame detection method based on TwoStream convolutional neural network. In Proceedings of the 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 24–26 May 2019; pp. 482–486.
7. Kim, B.; Lee, J. A video-based fire detection using deep learning models. *Appl. Sci.* **2019**, *9*, 2862. [CrossRef]
8. Thomson, W.; Bhowmik, N.; Breckon, T.P. Efficient and Compact Convolutional Neural Network Architectures for Non-temporal Real-time Fire Detection. In Proceedings of the 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), Miami, FL, USA, 14–17 December 2020; pp. 136–141.
9. Samarth, G.; Bhowmik, N.; Breckon, T.P. Experimental exploration of compact convolutional neural network architectures for non-temporal real-time fire detection. In Proceedings of the 2019 18th IEEE International Conference on Machine Learning and Applications (ICMLA), Boca Raton, FL, USA, 16–19 December 2019; pp. 653–658.
10. Han, X.F.; Jin, J.S.; Wang, M.J.; Jiang, W.; Gao, L.; Xiao, L.P. Video fire detection based on Gaussian Mixture Model and multi-color features. *Signal Image Video Process.* **2017**, *11*, 1419–1425. [CrossRef]
11. Kong, S.G.; Jin, D.; Li, S.; Kim, H. Fast fire flame detection in surveillance video using logistic regression and temporal smoothing. *Fire Saf. J.* **2016**, *79*, 37–43. [CrossRef]
12. Khan, S.; Naseer, M.; Hayat, M.; Zamir, S.W.; Khan, F.S.; Shah, M. Transformers in vision: A survey. *Acm Comput. Surv.* **2022**, *54*, 1–41. [CrossRef]
13. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; pp. 1–11.
14. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In *Computer Vision, Proceedings of the European Conference on Computer Vision (ECCV 2020), Glasgow, UK, 23–28 August 2020*; Springer: Cham, Switzerland, 2020; Volume 12346, pp. 213–229.
15. Steffens, C.R.; Rodrigues, R.N.; Silva da Costa Botelho, S. An Unconstrained Dataset for Non-Stationary Video Based Fire Detection. In Proceedings of the 2015 12th Latin American Robotics Symposium and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR), Uberlandia, Brazil, 29–31 October 2015; pp. 25–30. [CrossRef]
16. Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; Süsstrunk, S. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2274–2282. [CrossRef] [PubMed]
17. Çelik, T.; Özkaramanlı, H.; Demirel, H. Fire and smoke detection without sensors: Image processing based approach. In Proceedings of the 2007 15th European Signal Processing Conference, Poznan, Poland, 3–7 September 2007; pp. 1794–1798.

18. Celik, T. Fast and efficient method for fire detection using image processing. *ETRI J.* **2010**, *32*, 881–890. [CrossRef]
19. Zhou, X.L.; Yu, F.X.; Wen, Y.C.; Lu, Z.M.; Song, G.H. Early fire detection based on flame contours in video. *Inf. Technol. J.* **2010**, *9*, 899–908. [CrossRef]
20. Chenebert, A.; Breckon, T.P.; Gaszczak, A. A non-temporal texture driven approach to real-time fire detection. In Proceedings of the 2011 18th IEEE International Conference on Image Processing, Brussels, Belgium, 11–14 September 2011; pp. 1741–1744.
21. Foggia, P.; Saggese, A.; Vento, M. Real-time fire detection for video-surveillance applications using a combination of experts based on color, shape, and motion. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 1545–1556. [CrossRef]
22. Gong, F.; Li, C.; Gong, W.; Li, X.; Yuan, X.; Ma, Y.; Song, T. A real-time fire detection method from video with multifeature fusion. *Comput. Intell. Neurosci.* **2019**, *2019*, 1939171. [CrossRef] [PubMed]
23. Zhang, Q.; Xu, J.; Xu, L.; Guo, H. Deep convolutional neural networks for forest fire detection. In Proceedings of the 2016 International Forum on Management, Education and Information Technology Application, Guangzhou, China, 30–31 January 2016; Atlantis Press: Dordrecht, The Netherlands, 2016; pp. 568–575.
24. Zhao, Y.; Ma, J.; Li, X.; Zhang, J. Saliency detection and deep learning-based wildfire identification in UAV imagery. *Sensors* **2018**, *18*, 712. [CrossRef] [PubMed]
25. Dunnings, A.J.; Breckon, T.P. Experimentally Defined Convolutional Neural Network Architecture Variants for Non-Temporal Real-Time Fire Detection. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 1558–1562. [CrossRef]
26. Bello, I.; Zoph, B.; Vaswani, A.; Shlens, J.; Le, Q.V. Attention Augmented Convolutional Networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.
27. Parmar, N.; Vaswani, A.; Uszkoreit, J.; Kaiser, L.; Shazeer, N.; Ku, A.; Tran, D. Image Transformer. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; Volume 80, pp. 4055–4064.
28. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
29. Li, X.; Sun, X.; Meng, Y.; Liang, J.; Wu, F.; Li, J. Dice loss for data-imbalanced NLP tasks. *arXiv* **2019**, arXiv:1911.02855.
30. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
31. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. In Proceedings of the 32 Advances in Neural Information Processing Systems, Vancouver, BC, USA, 8–14 December 2019; pp. 1–12.
32. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv* **2017**, arXiv:1711.05101.
33. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.