

Received November 10, 2021, accepted November 21, 2021, date of publication December 3, 2021, date of current version January 6, 2022.

Digital Object Identifier 10.1109/ACCESS.2021.3132787

Spatio-Temporal Self-Attention Network for Fire Detection and Segmentation in Video Surveillance

MOHAMMAD SHAHID¹, JOHN JETHRO VIRTUSIO¹, YU-HSIEN WU¹,
YUNG-YAO CHEN², (Member, IEEE), M. TANVEER³, (Senior Member, IEEE),
KHAN MUHAMMAD⁴, (Member, IEEE), AND KAI-LUNG HUA¹

¹Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei 10633, Taiwan

²Department of Electronic and Computer Engineering, National Taiwan University of Science and Technology, Taipei 10633, Taiwan

³Discipline of Mathematics, IIT Indore, Indore 453552, India

⁴Visual Analytics for Knowledge Laboratory (VIS2KNOW Lab), School of Convergence, College of Computing and Informatics, Sungkyunkwan University, Seoul 03063, Republic of Korea

Corresponding authors: Khan Muhammad (khan.muhammad@ieee.org) and Kai-Lung Hua (hua@mail.ntust.edu.tw)

This work was supported in part by the Center for Cyber-Physical System Innovation and the Center of Intelligent Robots from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan; in part by the Ministry of Science and Technology of Taiwan under Grant MOST109-2218-E-011-010, Grant MOST108-2221-E-011-116, and Grant MOST108-2622-E-011-016-CC3; and in part by the Wang Jhan-Yang Charitable Trust Fund under Contract WJY 2020-HR-01.

ABSTRACT Convolutional Neural Networks (CNNs) based approaches are popular for various image/video related tasks due to their state-of-the-art performance. However, for problems like object detection and segmentation, CNNs still suffer from objects with arbitrary shapes, sizes, occlusions, and varying viewpoints. This problem makes it mostly unsuitable for fire detection and segmentation since flames can have an unpredictable scale and shape. In this paper, we propose a method that detects and segments fire-regions with special considerations of their arbitrary sizes and shapes. Specifically, our approach uses a self-attention mechanism to augment spatial characteristics with temporal features, allowing the network to reduce its reliance on spatial factors like shape or size and take advantage of robust spatial-temporal dependencies. As a whole, our pipeline has two stages: In the first stage, we take out region proposals using Spatial-Temporal features, and in the second stage, we classify whether each region proposal is flame or not. Due to the scarcity of generous fire datasets, we adopt a transfer learning strategy to pre-train our classifier with the ImageNet dataset. Additionally, our Spatial-Temporal Network only requires semi-supervision, where it only needs one ground-truth segmentation mask per frame-sequence input. The experimental results of our proposed method significantly outperform the state-of-the-art fire detection with a 2 ~ 4% relative enhancement in F1-score for large scale fires and a nearly ~ 60% relative improvement for small fires at a very early stage.

INDEX TERMS Fire detection, early detection, disaster management, small-sized fire, video fire segmentation, semi-supervised.

I. INTRODUCTION

According to a National Fire Protection Association report [1], in 2018, approximately 1,318,500 fire disasters occurred in the United States, causing 3655 deaths, 15200 injuries, and damages worth \$25.6 billion. This problem motivated several works towards fire detection systems, categorized into two classes: sensor-based technologies and image-based approaches. Popular sensor-based fire detection technologies include smoke detectors, thermometers, or ultraviolet light sensors. At the same time, these cheap

and widely available technologies rely on particle sampling that makes their performance hypersensitive to its location and proximity to the fire. Typically, this limitation makes it only suitable in indoor environments [2]. On the flip side, image-based fire discovery systems are more flexible in terms of location and can be used in outdoor settings. In terms of early detection, image-based approaches also have the upper hand. Unlike sensor-based, it does not have to wait for enough samples and can immediately detect small combustion. Additionally, image-based methods also offer more information than sensor-based technologies. For instance, it can localize the fire, measure its intensity, and track its growth. These insights are critically helpful in combating fire disasters.

The associate editor coordinating the review of this manuscript and approving it for publication was Miaohui Wang.

Early image-based fire detection approaches often rely on handcrafted pixel features and heuristics involving different thresholds [3]–[7]. While it may work well in a controlled setting, these rule-based approaches require constant threshold tuning and may not work well in a real-world environment.

In the past few years, the advent of deep learning allowed automatic feature extraction. Advancements like Convolutional Neural Networks (CNNs) are now state-of-the-art in many video/image related tasks [8]–[10]. The shift away from handcrafted features allowed fire detection approaches to be more robust and adaptive to real-world settings. Some CNN-based examples include, [11]–[13]. While these methods have good results in fire classification, one limitation is that they suffer when the fire is still small—which should be detected to prevent more damage. Additionally, they only assume a single input image and do not take advantage of discriminative fire temporal features like flickering, luminosity, color and warmth changes.

We consider a distinct method to understand this enigma and perceive the fundamental differences of fire to common objects of interest in object detection/segmentation problems. Fire is incredibly unique because of its unpredictable spatial characteristics. It can be big or small, and it can have arbitrary shapes. This property makes it harder to learn by conventional CNNs and adds a level of complexity to our fire detection and segmentation problem. Additionally, there is a scarcity of generous video fire datasets with ground-truth segmentation masks, making supervised learning difficult.

In this paper, we propose a fire detection method that identifies fire regions which will enable the first responders to understand the intensity and growth of the fire over time. Additionally, our network is explicitly designed to handle small-sized fires, making it suitable as an early detection system. To be more conspicuous, our pipeline comprises of two-stage. In the first stage, we parallelized the data propagation through two streams, termed Spatio-Temporal Network, which treats the spatial and temporal information separately. We design a semi-supervised network for the temporal streams that segregates fire region features from the background in a video based on a given keyframe. We combine spatial and temporal features using self-attention, learn robust fire-distinctive dependencies, and extract quality segmentation masks used as region proposals. The second stage is an error-correcting mechanism to refine predictions. Additionally, it is designed to learn scale-invariant features to be more robust against arbitrary-sized fires. As one of our contributions, we constructed a fire video dataset with ground-truth segmentation masks that are manually created. Moreover, for evaluation, we also created a dataset containing videos of small-sized fires, some of which are synthetically generated. We performed several experiments to prove our method's effectiveness, and we show that it compares auspiciously in opposition to the state-of-the-art-including on small-sized fire scenarios.

To summarize, our main contributions are:

- We propose a novel two-stage fire-detection approach. In the first stage, we implement two streams, termed the spatial-temporal network. We design a semi-supervised network for the temporal stream that segregates fire region features from the background in a video based on a given keyframe. The spatio stream uses static features from a single frame, such as color and texture.
- Our proposed approach uses self-attention on Spatio-Temporal features that are discriminative of fire, enabling our network to produce superior segmentation masks to use as region proposals. CNN-based binary classifiers classify these region proposals in the second stage, which is essential because some objects are also similar to fire.
- We constructed a video dataset containing manually generated ground-truth segmentation masks. Additionally, since one of our goals is early fire detection, we created synthetic videos with small-sized fires for evaluation purposes.

The paper is organized as follows. Section II discusses some related works and the progression towards the state-of-the-art. In Section III, we explain the challenges related to this work and motivate our approach to justify the design choices we made to solve the problem. We explain our approach in Section IV and discuss evaluations in Section V before finally arriving at a conclusion in Section VI.

II. RELATED WORK

In the last few years, video surveillance has become nearly a defacto standard in various fields, including anomaly detection [14], pedestrian detection [15] and fire detection [11]. Moreover, multiple attempts have been made to find more effective and efficient methods for coding surveillance videos [11], [16], [17].

A. FIRE DETECTION

Early works on image-based fire detection rely on handcrafted features descriptive of fire. For instance, Töreyn *et al.* [3] propose a wavelet transform to extract temporal features and rule-based decisions, which rely on thresholds to identify fire regions. Chen *et al.* [6] used RGB and HSI color spaces to analyze fire behavior in multiple frames and proposed heuristics to detect fire-regions. Vipin [18] used YBbCr color space to separate luminance from chrominance and classify if pixels are fire regions or not. Recent works are CNN-based and stray away from handcrafted features and heuristics [11]–[13]. For instance, Sharma *et al.* [19] investigated fire detection by finetuning popular VGG16 and Resnet50. Muhammad *et al.* [20] applied a model similar to GoogleNet to extract features from the image for early-stage fire detection. They also explored in [11] lightweight SqueezeNet [21] for fire detection and localization. Dunning and Breckon [12] use super-pixels with CNN architectures based on Inceptionv1, AlexNet, and VGG16 for fire detection. As part of their effort, CNN models are simplified

by keeping only some convolution, pooling, and dense layers to decrease model complexity while preserving accuracy. Using Multiple Instance Learning, Aktas *et al.* [22] extend the current CNN-based fire detection method in video sequences. Xie *et al.* [23] used both deep static and motion flicker-based dynamic features for detecting fire. The researchers [24] used a multi-scale feature extraction mechanism based on AlexNet to gain spatial detail information of fire in an image. They apply channel attention to emphasize the contribution of different feature maps. Oh *et al.* [25] presented a method for detecting wildfires using a light-weight EfficientNet framework. As a means of resolving the classes imbalance problem, they utilized the focal loss. Wang *et al.* [26] proposed a suspicious region localization using the Cauchy-mixture model in a five-dimensional feature space. Moreover, they designed a light-weight Squeeze-and-Excitation shuffleNet for the classification of the suspicious region. Li *et al.* [27] designed a dilated convolutional network for fire localization and classification, even performing better than fine-tuned CNNs. Shen *et al.* [28] used a one-stage detector to detect flames, such as YOLO. Based on the spatial features, Kim and Lee [29] applied Faster R-CNN to detect suspected fire regions, which LSTM then used to interpret the dynamic fire behavior. Apart from detection, some CNNs also allowed for segmenting the fire in an image [30], [31]. One limitation of CNN-based approaches is that they suffer if fire regions are small, which is an inherent limitation of conventional CNNs due to their fixed-size receptive fields [32]. To mitigate this problem, we incorporate design decisions [33], [34] that better preserve localization information. Additionally, we also incorporate temporal features and show that it can further improve segmentation performance.

B. OBJECT DETECTION AND SEGMENTATION

There is a wide variety of possible applications for object detection [35]–[37] and segmentation [38]–[40], [40], [41], including remote sensing [42]–[44], object counting [45]–[47], and image editing [48]–[52]. In this work, we focus on fire detection and segmentation, which comes with unique challenges. For instance, objects found in popular datasets [53], [54], like cats, dogs, or cars, usually have a defined shape. On the other hand, fires have an unpredictable nature. It can have an arbitrary shape, size, and even location on the image, making it harder to learn. Additionally, there are no large datasets containing fire and ground-truth segmentation masks, adding another layer of complexity. To address these limitations, our method is trained in a semi-supervised manner and only requires the ground-truth mask of one frame. Additionally, we adopt a transfer learning strategy and pre-train our network on ImageNet [54] to learn background information.

C. TEMPORAL FEATURES

Early approaches on fire detection using handcrafted features harnessed the power of temporal features through wavelet transforms or frame differences [3], [6], [18]. One advan-

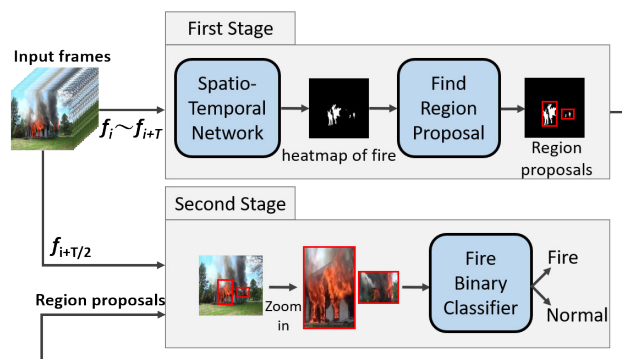


FIGURE 1. The overview of our approach containing two stages. The first stage takes in a sequence of frames $f_i - f_{i+T}$ to extract fire region proposals in frame $f_{i+T/2}$. The second stage classifies each proposal as either fire or normal.

tage of using temporal features for this problem is that fire behaves very distinctively across video frames. The presence of fire results in flickering luminosity, changes in color warmth, and rapid optical flow movements. Instead of relying on handcrafted features, we use convolution layers to learn temporal features. We augment spatial with temporal features using a self-attention mechanism, popular in NLP [55], to learn Spatio-temporal dependencies useful for segmenting fire regions.

D. SPATIO-TEMPORAL ATTENTION

Visual attention has been broadly applied in video-related tasks [56]–[59]. Liu *et al.* [56] enhanced the vanilla LSTM network's ability by appending Spatio-temporal attention for 3D action recognition, which selectively focuses on the action sequence's discriminative joints with the help of global contextual features from skeleton data. In a further study, Liu *et al.* [57] introduced a dynamic attention mechanism to progressively enhance recognition capability and improve network performance. Du *et al.* [58] presented a recurrent Spatio-temporal attention model that adaptively learns essential information from video context to intensify the ability of action representations. Wang *et al.* [60] introduced a non-local module to compute the spatial-temporal dependencies. In work for video captioning, Yan *et al.* [59] proposed an encoder-decoder architecture by embedding Spatio-temporal attention; thus, the decoder chooses essential regions from the most appropriate temporal segments for word prediction dynamically.

III. MOTIVATION

To reduce losses in fire disasters, we propose a method that can detect and segment fire regions in videos. Unlike traditional sensor-based technology, our approach can recognize small fires, enable early detection, and track its intensity progression through segmentation masks. We argue that since fires cause unexpected changes in size or shape, special design considerations should be made. In order to tackle the arbitrary characteristics of the fire's size and shape, we take

inspiration from popular network architectures like [34] and [33] and incorporate skip structure between the encoder and the decoder path. Additionally, to reduce our network’s reliance on spatial features like size or shape, we incorporate temporal features learned by 3D convolution layers. We use an attention mechanism to augment spatial with temporal features. As shown in our experiments, this strategy allows us to know Spatio-temporal dependencies that improve our network’s segmentation quality. Moreover, we apply a two-stage pipeline similar to existing object detection networks [35], [61]. The first stage extracts fire regions from the background based on a keyframe, and the second stage classifies the region. However, unlike object detection networks, our region proposals are segmentation masks, providing information about the fire’s size and intensity. This feature makes it especially useful as a fire detection system.

IV. PROPOSED APPROACH

We propose a fire detection approach sensitive to fires of varying sizes—from small to big. As shown in Fig. 1, our method has two stages: (1) region proposal and (2) classification. In the region proposal stage, we use a Spatio-temporal network that adopts self-attention to augment spatial with temporal features to extract high-quality segmentation maps. In the second stage, we utilize a classifier network to detect and verify fire regions accurately. This section is arranged as follows: in IV-A, we elaborate more about the first stage and discuss the Spatio-temporal network, followed by the extraction of region proposals in IV-B. Lastly, in IV-C, we discuss the fire classifier found in the second stage.

A. SPATIO-TEMPORAL NETWORK

Network Overview: As its name suggests, the Spatio-Temporal Network takes advantage of spatial and temporal features to extract fire segmentation masks. In Fig. 2, we show an overview of the network. It has 3 major parts: (1) TemporalNet, (2) SpatioNet, and (3) FuseNet. TemporalNet learns features related to the time component and takes in a sequence of frames f_i to f_{i+T} , where f_i is the initial frame and f_{i+T} is the final frame. On the other hand, SpatioNet only takes in a single frame $f_{i+T/2}$ as input. TemporalNet and SpatioNet would provide each output with a 64-channel feature map. In our implementation, TemporalNet takes in frames f_i to f_{i+14} , and SpatioNet takes in frame $f_{i+14/2}$. Inspired by [62], we concatenate the feature maps and pass them through a 1×1 convolution layer, which effectively learns how to shrink their size. Finally, FuseNet takes in the 1×1 conv layer output and learns spatial-temporal dependencies using a self-attention mechanism. Self-attention between spatial and temporal features extracts important relationships like how certain texture regions behave across time. These relationships are beneficial for fire detection and segmentation, as revealed in our experiments, which is discussed in Section V-D.

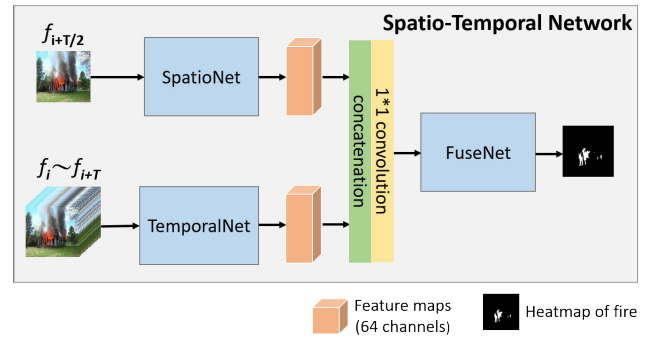


FIGURE 2. The architecture of our Spatio-Temporal Network. We use frames $f_{i+T/2}$ and f_i to f_{i+T} as inputs of SpatioNet and TemporalNet, respectively, then fuse their outputs using FuseNet to get the fire segmentation mask.

Training Overview: The network is trained in a multi-stage manner where we first train the SpatioNet and TemporalNet, then finally the FuseNet. SpatioNet and TemporalNet are trained independently to extract the fire segmentation mask of frame $f_{i+T/2}$. As shown in Fig. 2, these networks each output a 64-channel feature map. However, during the training stage, we augment these networks with another layer to output an $H \times W$ tensor corresponding to a segmentation mask. We phrase the segmentation problem as pixel classification and optimize the networks to reduce a binary cross-entropy also called as Log loss.

1) SpatioNet

Inspired by UNet++ [34], we use skip pathways structure to reduce the semantic gap between encoder and decoder feature maps. As shown in Fig. 3, our SpatioNet utilizes 2D VGG blocks, as depicted in Fig. 4, which concatenate the output of the previous block and the corresponding up-sampled output of the lower block. The dense skip connection enables the shallow layers to share information with deep layers easily. We use this design because localization information can be found in the shallow layers. By connecting it to deeper layers, we improve the network’s ability to detect small-size fires, which is critical in early fire detection systems.

In Eq. 1, we formally formulate the output of each block as $B^{i,j}$. The blocks are denoted as $L^{i,j}$, where i is the level of down-sampling, and j denotes the level of skip pathway. The function $V(\cdot)$ is a VGG convolutional operation, $C(\cdot)$ is a concatenation, and $U(\cdot)$ is an up-sampling layer. At level $j = 0$, nodes only receive one input from the previous layer. At level $j > 0$, nodes receive $j + 1$ inputs which j inputs are the outputs of previous nodes in the same skip pathway, and one input is the up-sampled output from the lower skip pathway. In our work, we use four layers of up-sampling and down-sampling.

$$B^{i,j} = \begin{cases} V(B^{i-1,j}), & j = 0 \\ V(C(C(B^{i,k})_{k=0}^{j-1}, U(B^{i+1,j-1}))), & j > 0 \end{cases} \quad (1)$$

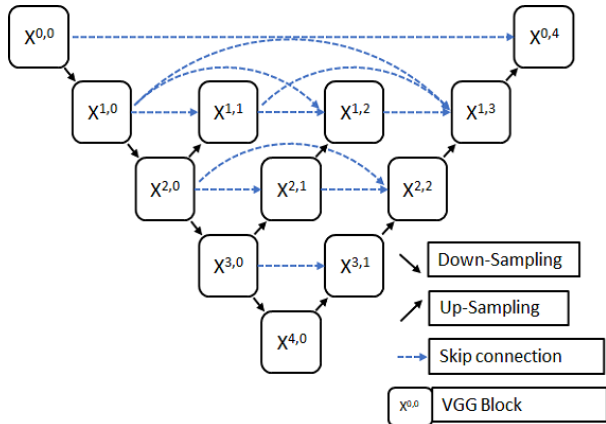


FIGURE 3. The architecture of the SpatioNet inspired by [34].

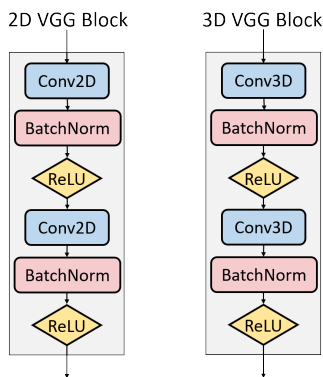


FIGURE 4. The architecture of VGG Block in 2D and 3D.

2) TemporalNet

This sub-network learns features from a series of frames f_i to f_{i+T} . These features are especially useful for our purpose because fire has specific temporal behavior. For instance, fire causes the luminosity of frames to flicker, or its color temperature to change. It may also exhibit rapid movements across frames.

As discussed previously (Section IV-A), we train the TemporalNet to output a segmentation mask. Because ground-truth labeling is expensive, we propose an architecture that takes in frames f_i to f_{i+T} but only requires the ground-truth segmentation mask of $f_{i+T/2}$. TemporalNet’s architecture is shown in Fig. 5. We use VGG blocks (shown in Fig. 4) with 3D for temporal behavior and max-pooling in the encoder to reduce the feature map’s dimension. In the decoder, 2D up-sampling recovers resolution’s spatial (height and width) dimension and ultimately outputs a segmentation mask of the middle frame $f_{i+T/2}$. This strategy allows for a semi-supervised learning approach that only needs one frame’s ground truth per input sequence. Nevertheless, this is not straightforward since the encoder primarily deals with 4-dimensional temporal information, and the decoder deals with 3-dimensional spatial information. To solve this problem, we utilize 1D max-pooling to reduce the feature map’s

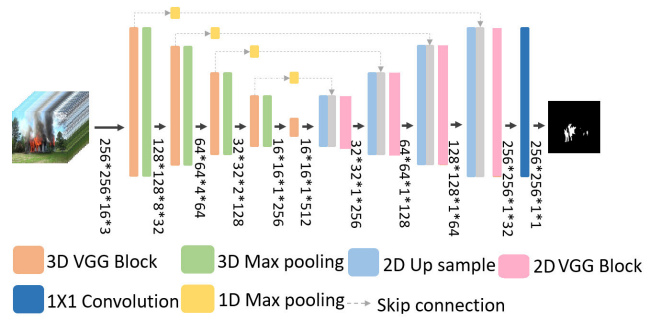


FIGURE 5. The architecture of TemporalNet.

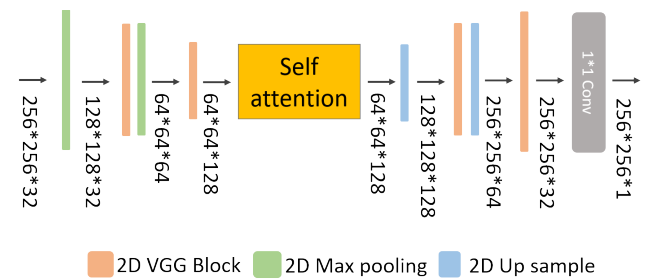


FIGURE 6. The architecture of FuseNet. To reduce the calculation, we down-sample the feature maps before sending them into the self-attention module.

temporal dimension from the contracting path of the encoder. The decoder uses 2D VGG blocks. We also incorporate skip-connections from the encoder to the decoder path, which is used to extract multi-scale features and retain detailed temporal information using 1D max-pooling. There are five convolutional layers and four max-pooling layers in the encoding path. In the decoding path, there are four upsampling layers and four convolutional layers.

3) FuseNet

As shown in Fig. 2, we concatenate features extracted from SpatioNet and TemporalNet using concatenation and a 1×1 convolution layer. Within the FuseNet, we use a self-attention mechanism inspired by [63] to extract dependencies of spatial and temporal features. The overview of our FuseNet is shown in Fig. 6. It has a Self-attention module between a down-sampling encoder and an up-sampling decoder. Before sending the feature maps into the Self-attention module, we down-sample the feature maps to reduce the calculation in the Self-attention module.

Fig. 7 shows an overview of the Self-attention module. Its goal is to get matrix $S \in \mathbb{R}^{N \times N}$, where each point of S_{ij} denotes i^{th} position’s impact on j^{th} position. This impact is regarded as self-attention, and it can learn pair-wise correlations of features. Since FuseNet takes in Spatial-Temporal features, it can effectively learn how each image region behaves in time with respect to the other areas. These features are especially critical for fire detection because a fire in one part of the image would always affect the surrounding areas. For instance, the surrounding area’s luminosity, color

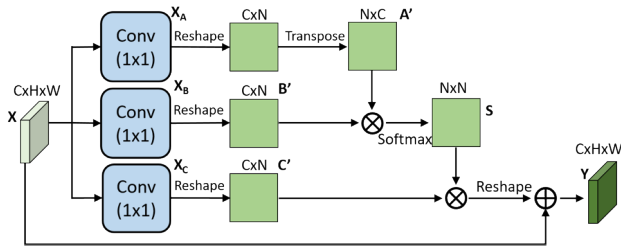


FIGURE 7. The architecture of the Self-Attention module.

temperature, and shadow movements are correlated with the fire’s intensity and behavior.

Specifically, to compute for the Self-attention matrix $S \in \mathbb{R}^{N \times N}$, we first use three 1×1 Convolution layers to transform the encoder output into three different feature spaces, X_A , X_B and X_C . We reshape the feature maps X_B and X_C to B' and C' , where $B', C' \in \mathbb{R}^{C \times N}$ and $N = H \times W$ and transpose it to A' , where $A' \in \mathbb{R}^{N \times C}$.

Using Softmax on A' and B' , we can get S , which is formally defined in the following equation:

$$S_{i,j} = \frac{\exp(A'_i \cdot B'_j)}{\sum_{i=1}^N \exp(A'_i \cdot B'_j)} \quad (2)$$

Next, we perform another matrix multiplication between C' and S , and then reshape the result to $\mathbb{R}^{C \times H \times W}$. Finally, we perform element-wise sum with the input feature maps X to get the final output Y , formally defined as follows:

$$Y_j = \alpha \sum_{i=1}^N (S_{i,j} \cdot C'_i) + X_j \quad (3)$$

Herein, C' denotes the reshaped output of X_C and dot (\cdot) denotes matrix multiplication. Inspired by [63], α is a learnable parameter.

Lastly, in the decoder shown in Fig. 6, we up-sample the feature maps back to the size of the input image and use a 1×1 convolution layer to get the final fire segmentation mask. In our experiments, we show that the Self-Attention module improves the performance of the network.

B. FINDING REGION PROPOSAL

After using the Spatio-Temporal Network, we want to extract the region proposals from the segmentation mask. To obtain this, as shown in Fig. 8, we convert the segmentation mask into binary and compute the bounding boxes for each connected component using component labelling of OpenCV, which is an algorithmic application of graph theory employed to determine the connectivity of “blob”-like areas in a binary image. Also, we extended each connected component’s region to find a single region of interest. Accordingly, dimensions of each region are enlarged from $[x, y, width, height]$ to $[x, y, x + width, y + height]$, then overlapping bounding boxes are merged into, and this process repeated iteratively. Finally, overlapping bounding boxes are consolidated into one region

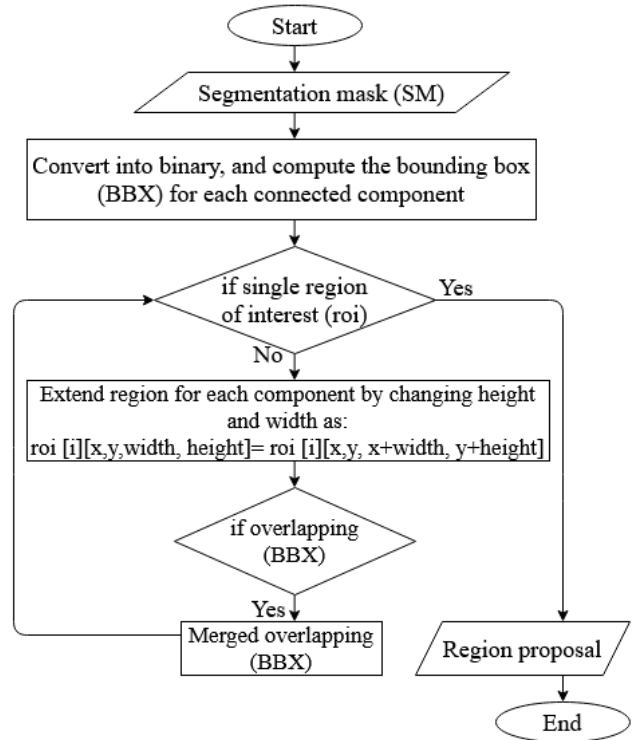


FIGURE 8. Flow diagram for the region proposal.

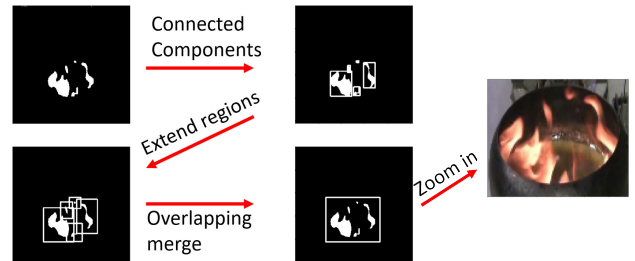


FIGURE 9. Computing the region proposal. Overlapping bounding boxes are merged as one region proposal.

proposal, as shown in Fig. 9. These region proposals will be classified in the next stage.

C. FIRE BINARY CLASSIFIER

The Fire Binary Classifier takes in region proposals from the first stage and identifies if it contains fire. Inspired by DenseNet [64], our classifier connects each layer to the other in a feed-forward way. For each layer, the feature maps of every previous layer are used as input, and its output feature maps are used as input for every layer behind. This strategy reduces gradient vanishing and enhances feature propagation. In this work, we call our classifier DenseFire, derived from the original DenseNet [64]. In our experiments, we also compare with different classifiers used in state-of-the-art fire detection approaches including, InceptionV1 [12] and SqueezeNet [11].

Because the fire dataset is small, we adopt a transfer learning strategy and train our classifier network for other tasks, indirectly enhancing fire classification performance. Specifically, we pre-train our classifier on ImageNet [54] so that it can learn useful features that can discriminate background objects from fire.

V. EXPERIMENTS

We will describe the experimental setting in detail in this section. First, we compare each sub-networks of our Spatio-Temporal Network and evaluate its segmentation quality. We replace our FuseNet with UNet to prove that the Self-attention module increases our network's segmentation performance. Then we perform several groups of experiments to prove the viability of our method. We compare our two-stage architecture with other state-of-the-art methods on publicly available and self-concreted fire datasets, including small-sized fires. We compare the computational cost of different state-of-the-art classifiers on the NTUST fire dataset. Finally, we test the robustness of the proposed framework.

A. IMPLEMENTATION DETAILS

All experiments are conducted on the machine (Intel(R) Core(TM) i7-7700K) with a RAM of sixty-four GIGabytes memory capacity and NVidia GTX 1080Ti graphics processing unit (GPU) of eleven GIGabytes. As for the software, all codes are implemented using the Pytorch deep learning framework on the Ubuntu system. We independently train SpatioNet and TemporalNet to optimize a binary cross-entropy loss. After these networks are trained, we freeze their weights, then train FuseNet. This multistage strategy allows us to train our model, despite memory constraints. Each network is trained using an Adam Optimizer with a learning rate of $3e-4$. We set the batch size to 4 and trained for 10000 epochs on the NTUST fire dataset.

B. DATASETS

Our approach aims to obtain the fire regions from a sequence of frames; therefore, we collected two datasets¹: NTUST fire dataset and small-sized fire dataset. We create one ground-truth mask per fire video to maximize our dataset's scenery variety because manually creating segmentation masks is a tedious task. To ensure fair evaluation and quantitatively appraise the achievement of our proposed method, we also used a publicly available dataset [7] and compared the results with other state-of-the-art techniques. Table 1 includes details about the datasets.

NTUST fire dataset We collected a total of 1033 videos, with 559 containing fires and 434 containing normal scenes. These videos contain diverse samples like scenes of burning wood, car, and trash. It also contains objects similar to fire, like sunsets and flashing lights. Fig. 10 shows some examples of our NTUST fire dataset. We used our NTUST dataset containing videos for training and testing.

¹Will be made available upon acceptance of manuscript.

TABLE 1. Details of datasets for the training and testing.

	Dataset source	Total no of videos	Total no of fire videos	Total no of normal videos
Train	NTUST dataset	706	401	305
Test		327	158	169
Test	small-size fire dataset	300	100	200
Test	Foggia dataset [7]	31	14	17

Small-sized fire dataset We define small-sized as occupying only 5% of the total pixels in the whole image. We gathered small-sized fires from the internet as the test set too. Additionally, we also generate synthetic videos by blending fire videos and normal videos frame by frame, as shown in Fig. 11, and use these images to augment our small-sized fire dataset. In total, we used 100 small-sized fire videos, with 200 normal videos sampled from the NTUST dataset. Fig. 12 shows some examples of small-sized fires from our dataset.

Foggia dataset [7] Provides 31 video clips with 62690 frames, which contains different situations; only 14 video clips hold the fire scene. Sample video clips from the Foggia dataset are shown in Fig. 13; the fire region of each video has a relatively substantial proportion of the images.

C. EVALUATION CRITERIA

The following metrics are used to examine the quantitative performance of the proposed approach.

The recall and precision are defined as:

$$recall = \frac{T_P}{T_P + F_N}, \quad precision = \frac{T_P}{T_P + F_P} \quad (4)$$

The F1-score is defined as:

$$F1 - score = 2 \times \frac{(precision \times recall)}{(precision + recall)} \quad (5)$$

The accuracy is defined as:

$$accuracy = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \quad (6)$$

T_P represents True-Positives, where the number of fires detected that ground truth are fires. F_P represents False-Positives, where the number of fires detected that ground truth are not fires. F_N represents False-Negative, those fires that have not yet been detected. T_N represents True-Negatives, where ground truth is not fire and predicted as False.

D. ABLATION STUDY

Segmentation Mask In the first stage, our pipeline outputs a segmentation mask using the proposed Spatio-Temporal Network (STNet), a fusion of sub-networks, SpatioNet, and TemporalNet. To analyze the individual contributions of SpatioNet and TemporalNet, we show the performance of each sub-network in terms of segmentation quality. As an evaluation metric, we use the dice coefficient (also known as F1-score) shown in Eq. (7), where H and W denote the height and width of the input image, X denotes the semantic ground-truth, and Y denotes the predicted segmentation mask. Dice

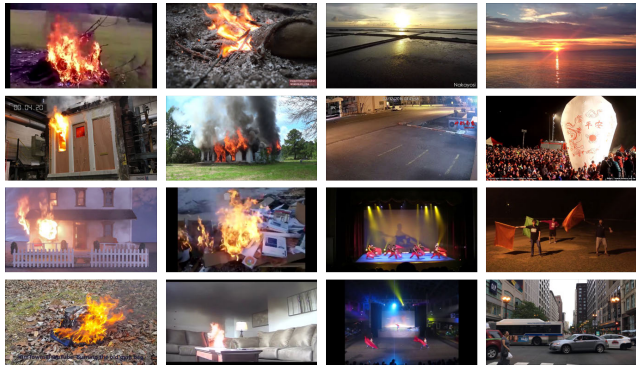


FIGURE 10. Some examples from our NTUST fire dataset. The two columns on the left are fire examples, and the two columns on the right are normal examples.



FIGURE 11. Generating synthetic small-sized fire videos through frame by frame blending.



FIGURE 12. Some examples from our small-sized fire dataset. The two rows on the top are real-world small fires, and the images in the last row are the synthesis small fire examples.

coefficient is a commonly used metric to evaluate segmentation quality [33], [34].

$$Dice = \frac{2 \times \sum_{i=1}^H \sum_{j=1}^W X_{ij}Y_{ij}}{\sum_{i=1}^H \sum_{j=1}^W (X_{ij}^2 + Y_{ij}^2)} \quad (7)$$

In Table 2, we show the dice coefficient of SpatioNet, TemporalNet, and Spatio-Temporal Network (STNet) on the

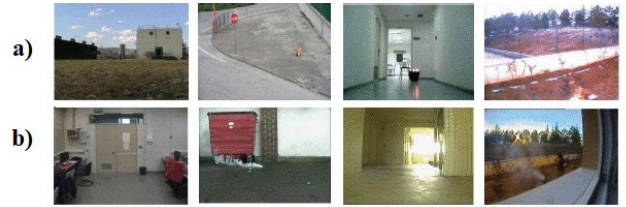


FIGURE 13. Some experimental images from the Foggia dataset [7]. Row a) video clips hold fire scenes, row b) video clips hold objects which look similar to fire.

TABLE 2. Dice coefficients of Spatio-Temporal Network (STNet) and its sub-networks.

Method	Dice
SpatioNet	0.771
TemporalNet	0.839
STNet	0.848

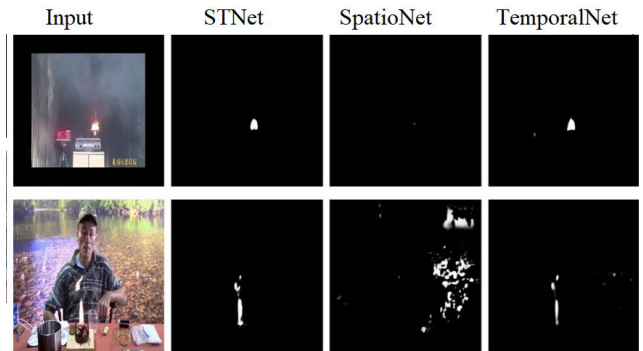


FIGURE 14. Output segmentation mask of Spatio-Temporal Network and its sub-networks, SpatioNet and TemporalNet.

test set of the NTUST dataset. Observe that the score of our TemporalNet is higher than the SpatioNet, highlighting the importance of temporal features in fire segmentation. Additionally, the result of our full network, Spatio-Temporal Network, proves that fusing the spatial and temporal features achieves the best results.

We also show the output segmentation masks of each network configuration in Fig. 14. In the first row, we show an example of a small-sized fire. It can be observed that only the SpatioNet failed to detect the fire, confirming our hypothesis that spatial features are not robust enough against arbitrarily-sized objects. In the second row, we show an input sample containing many objects that are brightly colored, similar to fire. The output of SpatioNet shows that it is sensitive to these objects that are not fire. On the other hand, it is harder to fool the TemporalNet because not many objects exhibit temporal features similar to fire. However, it could be observed that small patches on the right side of the image are still incorrectly labelled as fire. By combining spatial and temporal features, the Spatio-Temporal network shows the best segmentation masks.

Fusion Our FuseNet, as shown in Fig. 6, consists of a Self-attention module that learns global fire dependencies

TABLE 3. Comparing dice coefficients of Self-attention and UNet with FuseNet.

Method	Dice
FuseNet	0.804
FuseNet + UNet	0.839
FuseNet + Self-attention module	0.848

TABLE 4. Dice coefficients of Spatio-Temporal Network (STNet) and its sub-network with various self-attention (SA) mechanisms.

SpatioNet	TemporalNet	SA	Dice
✓		✓	0.775
	✓	✓	0.840
✓	✓	✓	0.848

from temporal and spatial features. In this experiment, we analyze the contribution of the Self-attention module in terms of improvements in segmentation quality. First, we obtain the dice coefficient of FuseNet alone and then compare the dice coefficient of FuseNet with the Self-Attention module and FuseNet with a UNet [33] structure. The resulting dice coefficient scores are shown in Table 3, and it can be observed that the Self-attention module achieves a better score than UNet, which justifies its use. Self-attention’s success is attributed to its ability to learn how each patch relates to the entire image. For fire segmentation, these relationships are critical because the presence of fire affects its surrounding regions.

Self Attention Each stream of the Spatio-Temporal Network provides specific information. And to further verify the significance of self-attention in the spatial and temporal streams in producing the output mask directly. We also examine the model by adding attention to the individual streams feature. We compare the dice coefficients for SpatioNet with self-attention, TemporalNet with self-attention, and the Spatio-Temporal Network (STNet) with self-attention on the NTUST dataset. The outcomes of the segmentation model are summarized in Table 4. It is noticed that the dice score of our SpatioNet is lesser than the TemporalNet (Table 2). It verifies that adding self-attention to individual streams does not make much significance (Table 4). SpatioNet with self-attention achieves a dice score of 0.775, whereas, without attention, it attains a dice score of 0.771. Similarly, TemporalNet with self-attention and without self-attention reach a dice score of 0.840, 0.839 respectively, which are almost similar. Additionally, the Spatio-Temporal Network result proves that fusing the spatial and temporal features with self-attention achieves the best results.

Two stage classifier For more comprehensive validation of two-stage classification, the ROC curve is added to estimate the fire detection of our network. The average values of area under the ROC curves for the NTUST Dataset is shown in Fig. 15.a. True positive rate is plotted against False positive rate in the ROC curve. It can be observed that the ROC curve of our two-stage classifier is quite close to the upper left corner, AUROC values of our STNet is 0.860, while for

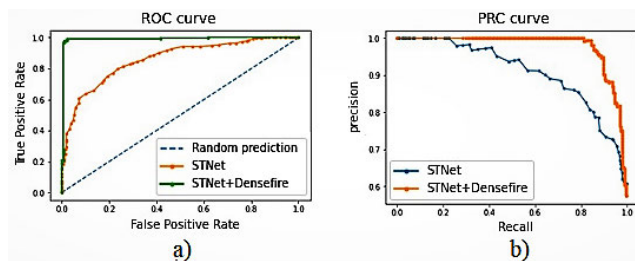


FIGURE 15. The ROC and PRC curves.

TABLE 5. Quantitative results fire segmentation on the NTUST dataset.

Method	Recall	Precision	F1-score
UNet [33]	0.727	0.541	0.645
UNet++ [34]	0.709	0.721	0.715
AttUNet [65]	0.748	0.569	0.646
R2UNet [66]	0.754	0.598	0.667
STNet (Ours)	1	0.736	0.8480

the two-stage (STNet+DenseFire) value is reached 0.991. Fig. 15.b shows the precision-recall curves. It can also be seen that the AUPRC values for two-stage (STNet+DenseFire) are relatively higher than STNet.

E. RESULTS ON THE NTUST DATASET

Segmentation Results on the NTUST Dataset: In this work, we utilize UNet [33] as a baseline. Moreover, to validate the proposed framework’s segmentation performance, we compare it against different deep learning-based models such as UNet++ [34], AttUNet [65] and R2UNet [66]. The qualitative results of our STNet with other deep CNN methods are shown in Fig. 16, which is based on the testing set of the NTUST Dataset. The binary mask outcomes indicate that our model is competent in capturing fire information. UNet++ shows good performance as compared to R2UNet and attention UNet. It can be noticed that the segmented fire areas using the conventional UNet model are worst among all. Furthermore, the quantitative evaluation score is listed in Table 5. We can see our model achieve higher recall and F1-scores (1, 0.848), respectively.

Comparing the Results of the Fire Binary Classifiers on the NTUST Dataset: We compare our fire binary classifier with other state-of-the-art fire detection methods. In the second stage of our pipeline, we use DenseFire to identify if the input contains fire or not. Usually, our DenseFire takes in region proposals from the first stage of our pipeline as shown in Fig. 17. However, in this experiment, first, we test the individual performance of our DenseFire without the first stage and see how it compares to other methods. We compare with InceptionOnFire [12] based on the Inception Network [67] and CNNFire [11] based on Squeeze Net [12]. We extracted a total of 13256 images from our NTUST dataset and used 80% for training and 20% for testing. We train each network as a binary classifier of fire or normal, and in Table 6, we show

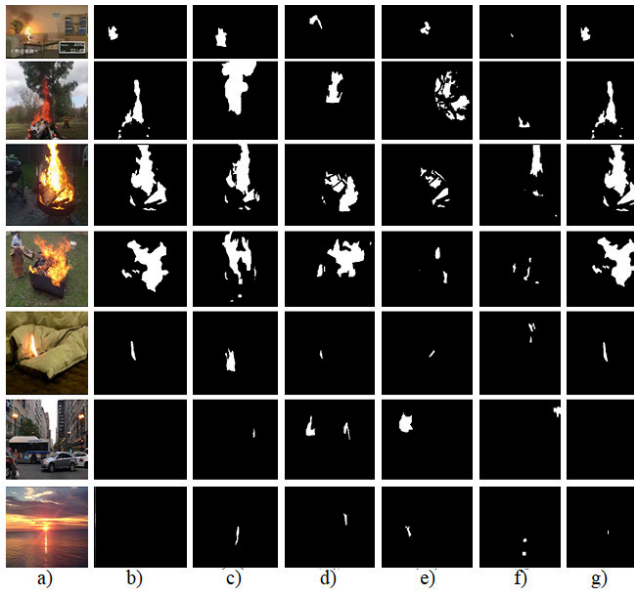


FIGURE 16. Qualitative comparison of fire segmentation. From left: a) Center frame from a sequence, b) ground truth, c) UNet++, d) AttUNet, e) R2UNet, f) UNet, and g) Ours STNet.

TABLE 6. Performance of single-stage approaches on the NTUST Dataset.

Method	Recall	Precision	F1-score
InceptionOnFire [12]	0.9870	0.9499	0.9681
EMNFire [13]	0.7422	0.9729	0.8420
CNNFire [11]	0.9902	0.9486	0.9690
DenseFire	0.9919	0.9607	0.9760

TABLE 7. Performance of two-stage approaches on the NTUST dataset.

Method	Recall	Precision	F1-score
STNet+ResNet	0.975	0.891	0.931
STNet+InceptionV1	0.963	0.915	0.938
STNet+SqueezeNet	0.998	0.898	0.945
STNet+DenseFire	1	0.984	0.992

each network’s performance in terms of recall, precision, and F1-score. We observe that DenseFire achieves the best F1-score.

Sometimes, it is difficult to distinguish between a real fire and an object that looks like a fire from a long distance by relying only on the above rules. Therefore, we considered a two-stage classifier. From the first stage obtained, the proposed region is re-classified by binary classifiers. Individual classifier’s success with STNet is measured in a recall, precision, and F1-score and presented in Table 7. We can observe that the performance is further enhanced. The classifiers discarded some of the region proposals that are identified as fire by STNet. It is apparent from the analysis that our method STN+Densefire is improved in various ways and achieved a recall of 100%, precision of 98.4% and F1-score of 99.2%, which indicates a more appropriate fire detection system in practice.

TABLE 8. Quantitative results fire segmentation on the small-sized fire dataset.

Method	Recall	Precision	F1-score
UNet [33]	0.631	0.569	0.598
UNet++ [34]	0.790	0.613	0.690
AttUNet [65]	0.739	0.584	0.652
STNet (Ours)	1	0.602	0.752

F. RESULTS ON THE SMALL-SIZED FIRE DATASET

Segmentation Results on the Small-Sized Fire Dataset: To test the versatility of our segmentation network, we compare it against various deep learning-based models such as UNet++ [34], AttUNet [65] and UNet [33] on the small-sized fire dataset. Fig. 18 shows visual comparisons with others. From row one, we can see that UNet cannot segment small fires, while UNet++ and AttUNet are partially able. Using the proposed approach, we can segment fire regions with excellent quality. From row two, we can observe that the proposed method and UNet++ correctly recognize the fire region while AttUNet over estimated fire region. UNet and AttUNet cannot accurately segment the fire in the third row while UNet++ exceeded the fire area. In comparison to UNet++, AttUNet, and UNet, STNet appears to be performing more salutary. Also, our quantitative results, shown in Table 8, confirm that our F1-score is the best among all methods, ensuring a high degree of specificity and sensitivity in identifying small fires.

Comparing the Results of the Fire Binary Classifiers on the Small Size Fire Dataset: In this experiment, we evaluate our model’s effectiveness in detecting small-sized fires, critical to early detection systems. In Table 9 and Table 10, we compare the performance of our Spatial-Temporal Network (STNet) against other state-of-the-art fire detection methods, including InceptionOnFire [12], CNNFire [11], and EMNFire [13]. Additionally, we also compare with ShuffleNet [68], a highly efficient classifier network.

These CNNs have some convolutional layers followed by a few fully connected layers. CNN, the image is converted into a vector which is primarily used in fire recognition. They are effective for fire recognition problems when fires are relatively large (Table 6). However, small fires are still giving them some trouble (Table 9). CNN layers reduce the images from high to low resolution, and a fully connected layer causes loss of spatial information. Consequently, small fire features they extract on the first layer (and a few of them to start with) disappear between the layers and are never actually used for classification. In Table 9, the very low recall value for InceptionOnFire, CNNFire, EMNFire and shuffleNet reveals that the classifier yields many results, with maximum results mislabeled for small fires. In contrast, the segmentation network does not have fully connected and only contains a convolutional layer. The image is converted into a vector and then converted back to an image using



FIGURE 17. Set of sample frames in video clips. a) The fire was increasing in a successive frame, b) the fire was almost consistent, c) video clip contained red color banner, and d) first, the fire was growing later shrinking.

TABLE 9. Performance of single-stage approaches on the small-sized fire dataset.

Method	Recall	Precision	F1-score
InceptionOnFire [12]	0.1	1	0.182
CNNFire [11]	0.04	0.667	0.075
EMNFire [13]	0.10	0.78	0.18
ShuffleNet [68]	0.54	0.26	0.35
DenseFire	0.04	1	0.077
STNet (Ours)	1	0.602	0.752

the exact mapping by preserving the original structure, also known as pixel-based classification. STNet provides us with a far more granular understanding of the fire in the video. It can be seen from Table 9, for STNet, the value of recall is the best, and the precision is low, which implies a high false positive. DenseFire, alone, achieves poor performance for recall and best for precision. It shows that detecting small fires is a nontrivial problem, and better performance can be achieved by using a 2-stage approach.

In Table 10, we compare the results of our STNet using different binary classifier architectures in its second stage. It could be observed that 2-staged approaches achieve significantly better results than the single staged approaches (as presented in Table 9). The classifier in the second stage discards region proposals, mistakenly identified as fire by STNet, which can be attributed to the success of two-staged approaches.

G. RESULTS ON THE FOGGIA DATASET

Segmentation Results on the Foggia Dataset: To validate the proposed framework’s fire segmentation performance,

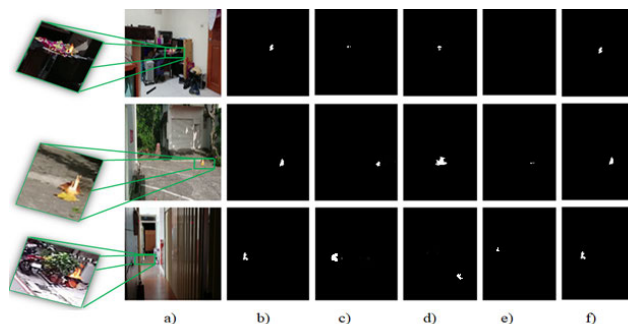


FIGURE 18. Qualitative comparison of fire segmentation on the small size fire dataset. From left: a) Center frame from a sequence, b) ground truth, c) UNet++, d) AttUNet, e) UNet, and f) Ours STNet.

TABLE 10. Performance of two-stage approaches on small-sized fire dataset.

Method	Recall	Precision	F1-score
STNet+ResNet	0.96	0.889	0.923
STNet+InceptionV1	0.96	0.905	0.932
STNet+SqueezeNet	1	0.892	0.943
STNet+DenseFire	0.96	0.923	0.941

we compare it against different deep learning-based models on the Foggia dataset such as UNet++ [34], AttUNet [65] and UNet [33]. The qualitative results of our STNet with other deep CNN methods are presented in Fig. 19. As we can observe in the first row, the AttUNet and UNet are incorrect in their segmentation based on the color of the fire. UNet++ overestimated the fire area. However, we can distinguish it

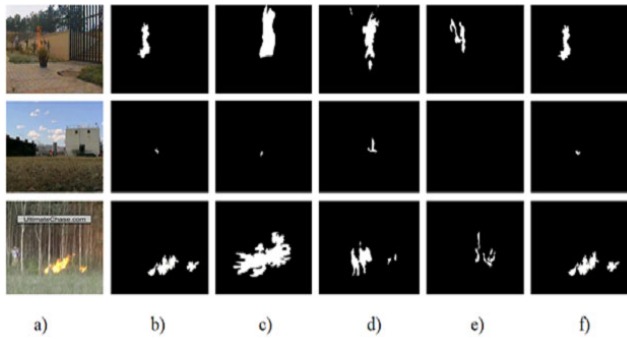


FIGURE 19. Qualitative comparison of fire segmentation on the Foggia Dataset. From left: a) Center frame from a sequence, b) ground truth, c) UNet++, d) AttUNet, e) UNet, and f) Ours STNet.

TABLE 11. Quantitative results fire segmentation on the Foggia dataset.

Method	Recall	Precision	F1-score
UNet [33]	0.838	0.725	0.769
UNet++ [34]	0.915	0.755	0.827
AttUNet [65]	0.890	0.727	0.800
STNet (Ours)	0.976	0.793	0.875

TABLE 12. Performance comparison on the Foggia dataset [7].

Method	Recall	Precision	F1-score
InceptionOnFire [12]	0.891	0.936	0.912
CNNFire [11]	0.957	0.895	0.924
EMNFire [13]	0.961	0.937	0.949
ShuffleNet [68]	0.845	0.928	0.884
DenseFire	0.982	0.964	0.973
STNet+DenseFire	0.989	0.995	0.992

from the proposed architecture. In the second row, UNet is unable to segment the fire while AttUNet exceeded the fire area. In the third row, the segmentation results, except the proposed, all other comparison methods used have a drawback. Table 11 shows the qualitative values of evaluation metrics received from the Foggia dataset, which indicate that the proposed method produces better results than other methods.

Comparing the Results of the Fire Binary Classifiers on the Foggia Dataset: We analyzed our results with other fire detection algorithms such as InceptionOnFire [12], CNNFire [11], EMNFire [13], and ShuffleNet [68] by considering a set of metrics such as recall, precision and F1-score. The experimental outcomes are shown in Table 12. We can see that ShuffleNet [68] reaches the recall of 0.845, which is worse than others. CNNFire [11] and EMNFire [13] perform similarly in terms of recall. However, the precision of EMNFire [13] is better than CNNFire [11]. DenseFire showed satisfactory performance in terms of precision. It is evident from Table 12, and our STNet+DenseFire has surpassed precision, recall, and F1-score values compared to others, indicating a more reliable fire detection ability.

H. ANALYSIS OF COMPUTATIONAL COST

The following section will see different deep learning model's performance in computational complexity, model complexity, and inference rate for fire detection.

To estimate the computational complexity of each deep learning model is based on floating-point operations

(FLOPs). For comparison, differences in computational complexity associated with various deep learning models for fire detection, CNNFire [11], EMNFire [13], GNetFire [20], ShuffleNet [68], DenseFire and UNet [33]+DenseFire are considered. As shown in Table 13, DenseFire requires 415×10^6 FLOPs counts. Densefire (96.9% accuracy for the NTUST dataset and 80.3% for small-sized fire dataset), with a 50% lower FLOPs count than CNNFire. Nevertheless, on both datasets, CNNFire performance is less than DenseFire. GNetFire requires 1500×10^6 FLOPs counts and performs well on the NTUST dataset in F1-score and accuracy (0.917, 90.2%) respectively, where its performance on the small-sized fire dataset is only hitting F1-score of 0.251 and accuracy of 32.5%. EMNFire has the lowest FLOPs counts and a 27.7% lower FLOPs count than DenseFire. Compared to EMNFire, DenseFire has improved the accuracy by 1.1% on the NTUST dataset and 31.5% on the small-sized fire dataset. ShuffleNet requires 542×10^6 FLOPs counts and a 27.7% Higher FLOPs count than DenseFire. The accuracy of DenseFire is higher by 8.1% on the NTUST dataset and 15.2% on the small-sized fire dataset compared to ShuffleNet. Furthermore, EMNFire, GNetFire and ShuffleNet gain F1-score (0.180, 0.251, 0.350) respectively, surpassing Densefire in F1-score on a small-sized fire dataset. Due to the poor robustness of the above classifier on challenging scenes, we also explore a two-stage classifier. The two-stage such as Unet+DenseFire classifier needs 1705×10^6 FLOPs, which is the highest. The performance as measured by the F1-score improved significantly on both datasets. The value of the F1-score on the NTUST dataset reached 0.895, while the accuracy reached 80.6%. On the small-sized fire dataset, its performance reached an F1-score (0.742) and accuracy (76.1%). Our two-stage STNet+DenseFire classifier requires 935×10^6 FLOPs counts. STNet+DenseFire obtain an accuracy of 99.5% for the NTUST dataset and 96.5% for the small-sized fire dataset. It implies that a two-stage classifier increases computational cost but also affect performance. Our STNet+DenseFire achieve F1-score of (0.992, 0.941) respectively on both datasets, which outperforms other methods given in Table 13.

Model complexity is also a standard metric for evaluating deep learning models. Counting the number of learnable parameters allows us to analyze the complexity of models. This information is quite helpful in determining how much GPU memory is needed for each model. We can also see in Table 13 the number of parameters for existing CNNs and our proposed network. The two-stage UNet+DenseFire classifier requires 36.7×10^6 parameters, while our STNet+DenseFire require 8.5×10^6 parameters. ShuffleNet introduces 5.4×10^6 parameters and achieves the F1-score (0.884) and accuracy (89.4%) for the NTUST dataset. In contrast, the small-size fire dataset had the F1-score (0.350) and accuracy (65.2%). Although CNNFire has the lowest parameter and lower parameter count than ours, it yields the worst performance on the small-sized fire dataset.

TABLE 13. Comparison between effectiveness and computational cost of fire detection models on different datasets.

Methods	#Parameter	FLOPs	NTUST Dataset		Small Size Dataset	
			F1-score	Accuracy	F1-score	Accuracy
CNNFire [11]	1.25×10^6	833×10^6	0.924	94.4	0.075	21.7
EMNFire [13]	3.4×10^6	300×10^6	0.945	95.8	0.180	38.8
GNetFire [20]	6.8×10^6	1500×10^6	0.917	90.2	0.251	32.3
ShuffleNet [68]	5.4×10^6	542×10^6	0.884	89.4	0.350	65.1
Our DenseFire	3.5×10^6	415×10^6	0.976	96.9	0.077	80.3
UNet [33]+DenseFire	36.7×10^6	1705×10^6	0.895	80.6	0.742	76.1
Our STNet+DenseFire	8.5×10^6	935×10^6	0.992	99.5	0.941	96.5

TABLE 14. Comparison between accuracy and inference rate (I.R.) of the different models.

		one-stage				two-stage		
		CNNFire [11]	EMNFire [13]	GNetFire [20]	ShuffleNet [68]	Our DenseFire	UNet [33] + DenseFire	Our STNet + DenseFire
Inference rate(fps)		47	65	22	38	40	11	32
Accuracy (%)	NTUST Dataset	94.4	95.8	90.2	89.4	96.9	80.6	99.5
	Small-sized Fire Dataset	21.7	38.8	32.5	65.1	80.3	76.1	95.5



FIGURE 20. Robustness of the proposed fire detection under various conditions, top row: NTUST, middle row: small-sized fire, and bottom row: Foggia dataset. a) No rotation, b) Clockwise 90, c) Clockwise 180 d) Clockwise 270, e) occlusion, and f) adding noise to video.

The frames per second (fps) unit is also a vital evaluation metric for the fire detection method. The comparison results are shown in Table 14 for fps, based on Nvidia GTX 1080Ti graphics processing unit (GPU) of eleven GIGabytes. We can observe that one-stage algorithms such as CNNFire [11], EMNFire [13], ShuffleNet [68], GNetFire [20] and DenseFire detect more quickly, which could detect more than

22 frames/s. CNNFire and EMNFire, operate faster than our approach. EMNFire attained an inference rate of 65 fps while maintaining an accuracy of 95.8% on the NTUST dataset. For the small-sized fire dataset, only reach an accuracy of 38.8% (Table 13). Similarly, CNNfire achieved an inference rate of 47 fps while maintaining an accuracy of 94.4% on the NTUST dataset. However, the small fire dataset had the

worst accuracy of 21.7% (Table 13). ShuffleNet has a similar inference rate to ours but is not competent in performance. As shown in Table 13, our two-stage approach is more reliable on both datasets than others. We reached an inference rate of 32 fps for our STNet+DenseFire model. Thus, our model is considerable enough for real-time fire detection, maintaining the F1-score of 0.992 on the NTUST dataset and 0.941 for the small-sized fire datasets. Although our two-stage method has a slower speed than EMNFire, both F1-score and accuracy are considerably higher. Our method achieves 96.5% accuracy on the small-sized dataset, which outperforms EMNFire by 57.7%. It is worth mentioning that the detection accuracy of our approach on the small-sized fire datasets outperforms that of the other methods by a large margin. In future work, we will further minimize model complexity to improve the inference rate for fire detection, providing a better balance between accuracy and inference.

I. MODEL ROBUSTNESS

Surveillance videos are primarily normal in real-world scenarios. A robust fire detection algorithm should have a minimum false-positive and false-negative on normal videos. Like false-negative, false alarm call-outs create a considerable drain on the fire and rescue service, also cause substantial disruption with loss of productivity to businesses. Moreover, firefighters diverted from real emergencies by answering false alarms may delay emergency response times, placing others at risk, such as children in schools, hospitals, and airports. Thus, in addition to analyzing computation costs with state-of-the-art methods, we also test the robustness of our networks to confirm detecting the fire in the video sequence. Fig. 20 shows three of the fire videos selected from the different datasets. Top row: NTUST dataset provides an indoor scene, middle row: small-sized fire dataset is considering, the camera may be far from the scene in some fire accidents, or fire is at an initial stage, and bottom row: Foggia dataset provides outdoor location. We examine various conditions such as a) no rotation, rotated in b) clockwise 90, c) clockwise 180, d) clockwise 270 degrees around the horizontal axis, e) fire entirely occluded by some object and f) adding noise to video to evaluate under possible attacks. From Fig. 20, we can see that the proposed method performs well in most cases. It also indicates that it is more effective at detecting fires in unknown conditions with varied atmospheres.

VI. CONCLUSION

In this paper, we proposed a two-stage architecture for early fire detection in videos, incorporating design strategies that can accurately detect small-sized fires. Precisely, we combined spatial features with temporal features in the first stage using a self-attention module to extract quality segmentation masks used as region proposals. Next, in the second stage, we classified the region proposal using a state-of-the-art classifier. Due to the lack of fire datasets, we employed semi-supervised learning, where we only needed a single ground-truth segmentation mask per frame-sequence input.

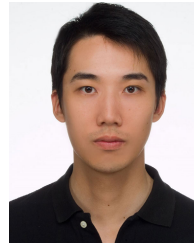
Additionally, we also adopted a transfer learning strategy and a pre-trained classifier on the ImageNet dataset. To train and evaluate our network, we constructed a fire video dataset with ground-truth segmentation masks. Since our goal is early detection, we also created a dataset of small-sized fires for evaluation. Using several evaluation metrics, we compared with other methods and showed that our approach performs best. Our proposed model's state-of-the-art performance can be attributed to the combination of learned temporal and spatial features, which allowed our model to detect fire based on its behaviour over time and its spatial features that can widely vary. Future work will be devoted to making a light-weight model to run on devices with computational or memory constraints.

REFERENCES

- [1] B. Evarts, "Fire loss in the United States during 2018," Nat. Fire Protection Assoc. (NFPA), Quincy, MA, USA, Tech. Rep., 2019. [Online]. Available: <https://www.nfpa.org/-/media/Files/News-and-Research/Fire-statistics-and-reports/US-Fire-Problem/FireLoss2019.aspx>
- [2] S. Khan, K. Muhammad, S. Mumtaz, S. W. Baik, and V. H. C. de Albuquerque, "Energy-efficient deep CNN for smoke detection in foggy IoT environment," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9237–9245, Dec. 2019.
- [3] B. U. Töreyn, Y. Dedeoğlu, U. Gündükbay, and A. E. Çetin, "Computer vision based method for real-time fire and flame detection," *Pattern Recognit. Lett.*, vol. 27, pp. 49–58, Jan. 2006.
- [4] G. Marbach, M. Loepfe, and T. Brupbacher, "An image processing technique for fire detection in video images," *Fire Saf. J.*, vol. 41, no. 4, pp. 285–289, 2006.
- [5] W. Phillips, M. Shah, and N. Da Vitoria Lobo, "Flame recognition in video," in *Proc. 5th IEEE Workshop Appl. Comput. Vis.*, Dec. 2000, pp. 224–229.
- [6] T.-H. Chen, P.-H. Wu, and Y.-C. Chiou, "An early fire-detection method based on image processing," in *Proc. Int. Conf. Image Process. (ICIP)*, vol. 3, Oct. 2004, pp. 1707–1710.
- [7] P. Foggia, A. Saggese, and M. Vento, "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.*, vol. 25, no. 9, pp. 1545–1556, Sep. 2015.
- [8] D. S. Tan, W.-Y. Chen, and K.-L. Hua, "DeepDemosaicking: Adaptive image demosaicking via multiple deep fully convolutional networks," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2408–2419, May 2018.
- [9] A. Talavera, D. S. Tan, A. Azcarraga, and K.-L. Hua, "Layout and context understanding for image synthesis with scene graphs," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 1905–1909.
- [10] J. J. M. Ople, D. S. Tan, A. Azcarraga, C.-L. Yang, and K.-L. Hua, "Super-resolution by image enhancement using texture transfer," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2020, pp. 953–957.
- [11] K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang, and S. W. Baik, "Efficient deep CNN-based fire detection and localization in video surveillance applications," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 7, pp. 1419–1434, Jul. 2019.
- [12] A. J. Dunning and T. P. Breckon, "Experimentally defined convolutional neural network architecture variants for non-temporal real-time fire detection," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 1558–1562.
- [13] K. Muhammad, S. Khan, M. Elhoseny, S. H. Ahmed, and S. W. Baik, "Efficient fire detection for uncertain surveillance environment," *IEEE Trans. Ind. Informat.*, vol. 15, no. 5, pp. 3113–3122, May 2019.
- [14] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6479–6488.
- [15] S. Alfasly, B. Liu, Y. Hu, Y. Wang, and C.-T. Li, "Auto-zooming CNN-based framework for real-time pedestrian detection in outdoor surveillance videos," *IEEE Access*, vol. 7, pp. 105816–105826, 2019.
- [16] J. Xiong, H. Gao, M. Wang, H. Li, and W. Lin, "Occupancy map guided fast video-based dynamic point cloud coding," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Mar. 2, 2021, doi: 10.1109/TCSVT.2021.3063501.

- [17] J. Xiong, X. Long, R. Shi, M. Wang, J. Yang, and G. Gui, "Background error propagation model based RDO in HEVC for surveillance and conference video coding," *IEEE Access*, vol. 6, pp. 67206–67216, 2018.
- [18] V. Vipin, "Image processing based forest fire detection," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 2, no. 2, pp. 87–95, 2012.
- [19] J. Sharma, O.-C. Granmo, M. Goodwin, and J. T. Fidge, "Deep convolutional neural networks for fire detection in images," in *Proc. Int. Conf. Eng. Appl. Neural Netw.* Athens, Greece: Springer, 2017, pp. 183–193.
- [20] K. Muhammad, J. Ahmad, I. Mehmood, S. Rho, and S. W. Baik, "Convolutional neural networks based fire detection in surveillance videos," *IEEE Access*, vol. 6, pp. 18174–18183, 2018.
- [21] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.
- [22] M. Aktas, A. Bayramcavus, and T. Akgun, "Multiple instance learning for CNN based fire detection and localization," in *Proc. 16th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Sep. 2019, pp. 1–8.
- [23] Y. Xie, J. Zhu, Y. Cao, Y. Zhang, D. Feng, Y. Zhang, and M. Chen, "Efficient video fire detection exploiting motion-flicker-based dynamic features and deep static features," *IEEE Access*, vol. 8, pp. 81904–81917, 2020.
- [24] S. Li, Q. Yan, and P. Liu, "An efficient fire detection method based on multiscale feature extraction, implicit deep supervision and channel attention mechanism," *IEEE Trans. Image Process.*, vol. 29, pp. 8467–8475, 2020.
- [25] S. H. Oh, S. W. Ghyme, S. K. Jung, and G.-W. Kim, "Early wildfire detection using convolutional neural network," in *Proc. Int. Workshop Frontiers Comput. Vis. Kagoshima, Japan: Springer*, 2020, pp. 18–30.
- [26] P. Wang, J. Zhang, and H. Zhu, "Fire detection in video surveillance using superpixel-based region proposal and ESE-ShuffleNet," *Multimedia Tools Appl.*, pp. 1–28, Sep. 2021, doi: [10.1007/s11042-021-11261-9](https://doi.org/10.1007/s11042-021-11261-9).
- [27] T. Li, E. Zhao, J. Zhang, and C. Hu, "Detection of wildfire smoke images based on a densely dilated convolutional network," *Electronics*, vol. 8, no. 10, p. 1131, Oct. 2019.
- [28] D. Shen, X. Chen, M. Nguyen, and W. Q. Yan, "Flame detection using deep learning," in *Proc. 4th Int. Conf. Control, Autom. Robot. (ICCAR)*, Apr. 2018, pp. 416–420.
- [29] B. Kim and J. Lee, "A video-based fire detection using deep learning models," *Appl. Sci.*, vol. 9, no. 14, p. 2862, Jul. 2019.
- [30] S. Frizzi, M. Bouchouicha, J. Ginoux, E. Moreau, and M. Sayadi, "Convolutional neural network for smoke and fire semantic segmentation," *IET Image Process.*, vol. 15, no. 3, pp. 634–647, Feb. 2021.
- [31] J. Mlich, K. Koplík, M. Hradis, and P. Zemcik, "Fire segmentation in still images," in *Proc. Int. Conf. Adv. Concepts Intell. Vis. Syst. Auckland, New Zealand: Springer*, 2020, pp. 27–37.
- [32] A. M. C. Antioquia, D. Stanley Tan, A. Azcarraga, and K.-L. Hua, "Single-fusion detector: Towards faster multi-scale object detection," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 76–80.
- [33] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. Munich, Germany: Springer*, 2015, pp. 234–241.
- [34] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: A nested U-Net architecture for medical image segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Granada, Spain: Springer, 2018, pp. 3–11.
- [35] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
- [36] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
- [37] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
- [38] J. Han, D. Zhang, X. Hu, L. Guo, J. Ren, and F. Wu, "Background prior-based salient object detection via deep reconstruction residual," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 8, pp. 1309–1321, Aug. 2015.
- [39] D. Zhang, D. Meng, and J. Han, "Co-saliency detection via a self-paced multiple-instance learning framework," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 5, pp. 865–878, May 2017.
- [40] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixe, D. Cremers, and L. Van Gool, "One-shot video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 221–230.
- [41] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille, "The secrets of salient object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 280–287.
- [42] J. Han, D. Zhang, G. Cheng, L. Guo, and J. Ren, "Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 6, pp. 3325–3337, Jun. 2015.
- [43] G. Cheng and J. Han, "A survey on object detection in optical remote sensing images," *ISPRS J. Photogramm. Remote Sens.*, vol. 117, pp. 11–28, Jul. 2016.
- [44] Z. Deng, H. Sun, S. Zhou, J. Zhao, L. Lei, and H. Zou, "Multi-scale object detection in remote sensing imagery with convolutional neural networks," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 3–22, Nov. 2018.
- [45] H. Cholakkal, G. Sun, F. Shahbaz Khan, and L. Shao, "Object counting and instance segmentation with image-level supervision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12397–12405.
- [46] M.-R. Hsieh, Y.-L. Lin, and W. H. Hsu, "Drone-based object counting by spatially regularized regional proposal network," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4145–4153.
- [47] C. Zhang, H. Li, X. Wang, and X. Lu, "Cross-scene crowd counting via deep convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 833–841.
- [48] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, "Universal style transfer via feature transforms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 386–396.
- [49] L. A. Gatys, A. S. Ecker, M. Bethge, A. Hertzmann, and E. Shechtman, "Controlling perceptual factors in neural style transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3985–3993.
- [50] Y. Jing, Y. Liu, Y. Yang, Z. Feng, Y. Yu, D. Tao, and M. Song, "Stroke controllable fast style transfer with adaptive receptive fields," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 238–254.
- [51] T. Park, J.-Y. Zhu, O. Wang, J. Lu, E. Shechtman, A. A. Efros, and R. Zhang, "Swapping autoencoder for deep image manipulation," 2020, *arXiv:2007.00653*.
- [52] O. Sbai, C. Couprie, and M. Aubry, "Unsupervised image decomposition in vector layers," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2020, pp. 1576–1580.
- [53] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis. Zürich, Switzerland: Springer*, 2014, pp. 740–755.
- [54] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
- [55] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
- [56] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot, "Global context-aware attention LSTM networks for 3D action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1647–1656.
- [57] J. Liu, G. Wang, L.-Y. Duan, K. Abdiyeva, and A. C. Kot, "Skeleton-based human action recognition with global context-aware attention LSTM networks," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 1586–1599, Apr. 2018.
- [58] W. Du, Y. Wang, and Y. Qiao, "Recurrent spatial-temporal attention network for action recognition in videos," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1347–1360, Mar. 2018.
- [59] C. Yan, Y. Tu, X. Wang, Y. Zhang, X. Hao, Y. Zhang, and Q. Dai, "Stat: Spatial-temporal attention mechanism for video captioning," *IEEE Trans. Multimedia*, vol. 22, no. 1, pp. 229–241, Feb. 2020.
- [60] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.
- [61] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
- [62] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1933–1941.
- [63] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 7354–7363.

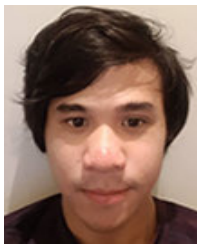
- [64] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.
- [65] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker, and D. Rueckert, "Attention U-net: Learning where to look for the pancreas," 2018, *arXiv:1804.03999*.
- [66] M. Zahangir Alom, M. Hasan, C. Yakopcic, T. M. Taha, and V. K. Asari, "Recurrent residual convolutional neural network based on U-Net (R2U-Net) for medical image segmentation," 2018, *arXiv:1802.06955*.
- [67] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
- [68] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.



YUNG-YAO CHEN (Member, IEEE) received the B.S. and M.S. degrees in electrical and control engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2004 and 2006, respectively, and the Ph.D. degree in electrical engineering from Purdue University, USA, in 2013. Before being a Faculty, he has worked with HP Labs—Printing and Content Delivery Lab (HPL-PCDL), for about one year. He is currently an Associate Professor with the Department of Electronic and Computer Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan. His current research interests include vision-based automation, automated/wisdom factory, self-driving car, and human–computer interaction. He is a member of the Golden Key International Honor Society and the Phi Tau Phi. He was a recipient of the Best Paper Award of the International Conference on Advanced Robotics and Intelligent Systems in 2015 and 2020, the Rotary Foundational Scholarship, and the Ta-Yu Wu Memorial Award from the Taiwan’s Ministry of Science and Technology (MOST).



MOHAMMAD SHAHID received the M.S. degree in computer science from Aligarh Muslim University, Aligarh, India. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology. His research interests include computer vision and image/video processing.



JOHN JETHRO VIRTUSIO received the B.S. and M.S. degrees in computer science from De La Salle University. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology. His current research interests include image/video processing and computer vision.



YU-HSIEN WU received the M.S. degree in computer science from the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology. His current research interests include object segmentation, deep learning, and image processing.



M. TANVEER (Senior Member, IEEE) received the M.Phil. degree in mathematics from Aligarh Muslim University, Aligarh, India, and the Ph.D. degree in computer science from Jawaharlal Nehru University, New Delhi, India. He is currently an Associate Professor and a Ramanujan Fellow with the Discipline of Mathematics, IIT Indore. Prior to that, he spent one year as a Postdoctoral Research Fellow with the Rolls-Royce@NTU Corporate Laboratory, Nanyang Technological University, Singapore. From 2012 to 2015, he was an Assistant Professor with the Department of Computer Science and Engineering, The LNM Institute of Information Technology (LNMIIT), Jaipur. His research interests include support vector machines, optimization, machine learning, deep learning, applications to Alzheimer’s disease and dementias, biomedical signal processing, and fixed point theory and applications. He has published over 40 refereed journal articles of international repute. He was a recipient of the 2017 SERB-Early Career Research Award in Engineering Sciences and the only recipient of the 2016 DST-Ramanujan Fellowship in Mathematical Sciences which are the prestigious awards of INDIA at early career level. He is currently a member of Editorial Board/a Guest Editor of several journals, including *ACM Transactions of Multimedia Computing, Communications, and Applications* (TOMM), *Applied Soft Computing* (Elsevier), *Applied Intelligence* (Springer), *Multimedia Tools and Applications* (Springer), and *Smart Science* (Taylor & Francis). He has also co-edited one book in Springer on machine intelligence and signal analysis. He has organized many international/national conferences/symposium/workshop as the general chair/the organizing chair/a coordinator, and delivered talks as a keynote speaker/a plenary speaker/an invited speaker in many international conferences and symposiums. He is the Co-Chair of the Special Session Proposal in 2018 IEEE SSCI, and organized several special sessions in top-ranked conferences, including WCCI, IJCNN, IEEE SMC, and IEEE SSCI. He is also a principal investigator (PI) or a co-PI of seven major research projects funded by the Government of India, including the Department of Science and Technology (DST), the Science and Engineering Research Board (SERB), the Council of Scientific and Industrial Research (CSIR), and MHRD-SPARC.



KHAN MUHAMMAD (Member, IEEE) received the Ph.D. degree in digital contents from Sejong University, Seoul, South Korea, in February 2019. He is currently an Assistant Professor with the School of Convergence, College of Computing and Informatics, Sungkyunkwan University, Seoul. He is currently a Professional Reviewer for over 120 well-reputed journals and conferences. He has registered eight patents and published over 170 articles in peer-reviewed international journals and conferences in his research areas. His research interests include medical image analysis (brain magnetic resonance imaging, diagnostic hysteroscopy, and wireless capsule endoscopy), information security (steganography, encryption, watermarking, and image hashing), video summarization, computer vision, fire/smoke scene analysis, and video surveillance. He is listed among the top cited researchers (top 1%) by Web of Science for the year 2021.



KAI-LUNG HUA received the B.S. degree in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 2000, the M.S. degree in communication engineering from National Chiao Tung University, Hsinchu, in 2002, and the Ph.D. degree from the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA, in 2010. Since 2010, he has been with the National Taiwan University of Science and Technology, Taipei, Taiwan, where he is currently a Professor with the Department of Computer Science and Information Engineering. Since 2021, he has also been the Dean of the Office of Industry-Academia Collaboration. He is also the Director of the Artificial Intelligence Research Center. His current research interests include digital image and video processing, computer vision, and machine learning. He is a member of the Eta Kappa Nu and the Phi Tau Phi. He was a recipient of the MediaTek Doctoral Fellowship. He was also a recipient of several research awards, including the 2019 Outstanding Research Award of Taiwan Tech, the 2018 Young Scholar Award of Taiwan Tech, the Top Performance Award of 2017 ACM Multimedia Grand Challenges, the Top 10% Paper Award of 2015 IEEE International Workshop on Multimedia Signal Processing, the Second Award of the 2014 ACM Multimedia Grand Challenge, the Best Paper Award of the 2013 IEEE International Symposium on Consumer Electronics, and the Best Poster Paper Award of the 2012 International Conference on 3D Systems and Applications.

...