applied sciences

MDPI

*Article*

# Cascaded Vehicle Matching and Short-Term Spatial-Temporal Network for Smoky Vehicle Detection

**Xiaojiang Peng [1], Xiaomao Fan [1,*], Qingyang Wu [1], Jieyan Zhao [2], Pan Gao [3]**

[1] College of Big Data and Internet, Shenzhen Technology University, Shenzhen 518118, China; pengxiaojiang@sztu.edu.cn (X.P.)
[2] Business School, Central South University, Changsha 410083, China
[3] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China; pan.gao@nuaa.edu.cn
* Correspondence: fanxiaomao@sztu.edu.cn

**Abstract:** Vehicle exhaust is the main source of air pollution with the rapid increase of fuel vehicles. Automatic smoky vehicle detection in videos is a superior solution to traditional expensive remote sensing with ultraviolet-infrared light devices for environmental protection agencies. However, it is challenging to distinguish vehicle smoke from shadow and wet regions in cluttered roads, and could be worse due to limited annotated data. In this paper, we first introduce a real-world large-scale smoky vehicle dataset with 75,000 annotated smoky vehicle images, facilitating the effective training of advanced deep learning models. To enable a fair algorithm comparison, we also built a smoky vehicle video dataset including 163 long videos with segment-level annotations. Second, we present a novel efficient cascaded framework for smoky vehicle detection which largely integrates prior knowledge and advanced deep networks. Specifically, it starts from an improved frame-based smoke detector with a high recall rate, and then applies a vehicle matching strategy to fast eliminate non-vehicle smoke proposals, and finally refines the detection with an elaborately-designed short-term spatial-temporal network in consecutive frames. Extensive experiments in four metrics demonstrated that our framework is significantly superior to hand-crafted feature based methods and recent advanced methods.

**Keywords:** smoky vehicle detection; smoke recognition; deep learning; convolutional neural networks

check for updates

**Citation:** Peng, X.; Fan, X.; Wu, Q.; Zhao, J.; Gao, P. Cascaded Vehicle Matching and Short-Term Spatial-Temporal Network for Smoky Vehicle Detection. *Appl. Sci.* **2023**, *13*, 4841. https://doi.org/10.3390/app13084841

Academic Editor: João M. F. Rodrigues

Received: 9 March 2023
Revised: 3 April 2023
Accepted: 10 April 2023
Published: 12 April 2023

**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

## 1. Introduction

Motor vehicles are the main source of urban carbon monoxide, hydrocarbons, and oxides of nitrogen, that are responsible for the formation of photochemical smog [1]. In the world, especially in developing countries, fuel vehicles produce a large number of emissions which affect our health [2]. Smoky vehicles refer to those vehicles with visible black smoke emitting from the vehicle exhaust hole. Automatic smoky vehicle detection together with license plate recognition systems is essential for environmental protection agencies to penalize over-emitting vehicles.

Generally, there are several techniques for smoky vehicle detection in the industry. An old type of smoky vehicle detection method works manually and includes public reporting, regular road inspection, night inspection by law enforcement workers, and manual video monitoring. Recent advanced methods including remote sensing vehicle exhaust monitoring with ultraviolet-infrared light, and automatic detection in surveillance videos [3]. Due to the cost of remote sensing with ultraviolet-infrared light being extremely expensive, detecting smoky vehicles in surveillance videos has recently attracted increasing attention [4–11].

In the past two decades, smoke detection and recognition in videos has been studied extensively [12–23], and most work has mainly focused on wildfire scenarios, while little attention has been paid to the road vehicle scenario. For a long time, textural features

dominated smoke recognition. For example, wavelet features were used in [14] and local binary patterns were used in [17,18,20,22,24]. Considering the motion information of smoke, motion orientation features [16], dynamic texture features [18], and 3D local difference features [19] are also widely used. After feature extraction for regions, a classifier such as support vector machine (SVM) [25] is used to categorize each region as smoke and non-smoke regions. Recently, with the development of deep learning [26], convolutional neural networks have been extensively used for smoke recognition [21,27–33]. Reference [27] evaluated different convolutional neural networks for image-based smoke classification including AlexNet [26], ResNet [34], VGGNet [35], etc. Three-dimensional (3D) convolutional neural networks were applied for smoke sequence recognition in [28,29,32]. Recurrent networks were also explored for sequence-based smoke recognition in [31]. Almost all the mentioned methods focus on general wildfire smoke or forest wildfire smoke as shown in the left and middle of Figure 1.



**Figure 1.** Smoke detection in different environments. (**Left**): general wildfire smoke [18]. (**Middle**): forest wildfire smoke [17]. (**Right**): vehicle smoke on road (ours).

Intuitively, the above smoke recognition methods can be used for smoke vehicle detection without considering vehicles. Along this direction, Tao et al. [7] leveraged a codebook model for background modeling and applied volume local binary count patterns, which achieved impressive smoke detection results in strong smoky traffic videos. However, there exist essential differences between wildfire smoke and vehicle smoke as shown in Figure 1. First, the smoke emitted from vehicles is mostly slighter than wildfire smoke, and occupies only a small region in video frames. Second, due to smoke being emitted from the bottom of vehicles, algorithms are apt to confuse smoke, shadows, and rainy road surfaces. Though much progress has been made in recent years [4–11], the false alarm rate over frames is still high (e.g., around 13% in [3]), which makes it hard to meet practical applications. Worse still, unlike wildfire smoke recognition, there is no publicly available vehicle smoke data in the community. The above issues make smoky vehicle detection more challenging and it is still an open problem.

In this paper, to address these challenges, we first introduce a real-world Large-Scale Smoky Vehicle image dataset, termed LaSSoV, which is the largest vehicle smoke dataset to date. In total, the LaSSoV contains 75,000 smoky vehicle images with bounding box annotations, which is expected to facilitate the effective training of deep learning models in smoky vehicle detection. To simulate practical scenarios, we also provide a smoky vehicle video dataset, termed LaSSoV-video, which includes 163 segment-annotated long videos in diverse environments.

Second, we propose a novel efficient cascaded framework which fully makes use of prior knowledge and deep neural networks. In the first step of this framework, we improved the famous YOLOv5 object detector with minimum width and depth in its architecture, and also replaced most of the convolutional operations with depth-wise and point-wise convolutional operations inspired by MobileNet [36] and GhostNet [37]. In the second step, with the prior knowledge that smoke must be emitted from vehicles, we designed a vehicle matching module which computes the Intersection-over-Union (IoU) ratios or pair-wise distances between smoke proposals and vehicle regions. The smoke-

vehicle matching module can easily remove these false positives, such as non-road and other non-vehicle regions. As found in practice, the detected regions of static images from a smoke detector are difficult for human beings to further check if they are cropped, while we can easily distinguish them by watching the spatial-temporal extent of each region. In the last step, we considered the spatial-temporal extent of a smoke region, and developed a short-term spatial-temporal network ($(ST)^2$Net) to classify the spatial-temporal extent for refinement. Specifically, we explored two kinds of $(ST)^2$Net based on ResNet18: one is called $(ST)^2$Net-prefix, which replaces the early 2D convolutional layers of ResNet18 with 3D ones, and the other is called $(ST)^2$Net-suffix, which performs 3D convolution in the last feature maps of ResNet18 over several frames.

We extensively evaluated our method and other recent advanced methods on four frame-wise metrics including detection rate, false alarm rate, precision, and F1 score. Almost all previous works only provide one or two of them for performance evaluation, which is not fair for a comprehensive comparison. We conducted experiments on our LaSSoV and LaSSoV-video since there is no public smoky vehicle data, and demonstrated that our framework is significantly superior to those hand-crafted feature-based methods and recent advanced methods.

In the remainder of this paper, we briefly review the related works on smoke detection and smoky vehicle detection in Section 2. We then introduce our datasets and compare them to existing datasets in Section 3. We further describe our methods in Section 4 and present experimental results in Section 5. Finally, we present a conclusion and discuss future work in Section 6.

## 2. Related Work

Smoky vehicle detection is a challenging scientific and engineering problem. We briefly review the related works on wildfire visual smoke detection and recent smoky vehicle detection in this section.

### 2.1. Visual Smoke Detection

Visual smoke detection aims to detect fire by recognizing early smoke or fire smoke anywhere within the field of the camera at a distance. In the past two decades, video-based wildfire smoke detection has been extensively studied. Here, we roughly categorize the video-based smoke detection methods into shallow feature-based methods and deep learning-based methods.

Shallow feature based methods. These methods mainly extract structural and statistical features from visual signatures such as the motion, color, edge, obscurity, geometry, texture, and energy of smoke regions. Guillemant and Philippe [38] implemented a real-time automatic smoke detection system for forest surveillance stations based on the assumption that the energy of the velocity distribution of a smoke plume is higher than that of other natural occurrences. Gomez-Rodriguez et al. [39] used an optical flow and wavelet decomposition algorithm for wildfire smoke detection and monitoring. Töreyin et al. [14] presented a close range (in 100 m) smoke detection system, which is mainly comprised of moving region detection and spatial-temporal wavelet transform. Yuan [16] proposed an accumulative motion orientation feature-based method for video smoke detection. Lin et al. [18] applied volume local binary patterns for dynamic texture modelling and smoke detection. Yuan et al. [20] considered the scale invariance of local binary patterns and utilized Hamming distances for local binary patterns to address image-based smoke recognition. Yuan et al. [23] leveraged multi-layer Gabor filters for image-based smoke recognition. Overall, all these shallow feature-based methods focused on improving the robustness of hand-crafted texture or motion features, which may be limited in their representation abilities.

Deep learning based methods. With the development of deep learning [26], convolutional neural networks (CNNs) have recently been extensively used for visual image classification and object detection [34,40]. Visual smoke detection can be viewed as an

object classification or detection task. From the characteristic of deep networks, we can roughly divide these deep networks into 2D appearance networks and spatial-temporal networks.

For 2D networks, Frizzi et al. [41] built their CNNs very similarly to the well-known LeNet-5 [42] with an increased number of feature maps in convolution layers, and used real smoke scenarios and achieved 97.9% accuracy, which is higher than the performance of traditional machine learning approaches. Tao et al. [43] applied the well-known AlexNet [26] for visual smoke detection and achieved a high performance on open their smoke image dataset. Filonenko et al. provided a comparative study of modern CNN models. To address image-based smoke detection, Gu et al. [30] designed a new deep dual-channel neural network for high-level texture and contour information extraction.

For spatial-temporal networks, Lin et al. [29] designed a 3D CNN to extract the spatial-temporal features of smoke proposals. Tao et al. [9] built a professional spatial-temporal model comprised of three CNNs on different orthogonal planes. Li et al. [28] proposed a 3D parallel fully convolutional network for real-time video-based smoke detection. In the other direction, recurrent neural networks have also been applied for video-based smoke detection [31].

Overall, almost all the outlined methods focus on general wildfire smoke recognition or detection.

### 2.2. Smoky Vehicle Detection

Smoky vehicle detection includes smoke detection and vehicle assignment, aiming to capture those vehicles that are emitting visible smoke. The obvious difference between general wildfire smoke and vehicle smoke is that the backgrounds of vehicle smoke are varied roads, which contain vehicle shadows, dirty regions, etc.

One direction of smoky vehicle detection only considers detecting vehicle smoke regions as accurately as possible, for which general smoke detection methods can be resorted to. Tao et al. [7] leveraged a codebook model for background modeling and applied volume local binary count patterns, which achieved impressive smoke detection results in strong smoky traffic videos. Tao et al. [4] also extracted the multi-scale block Tamura features for smoke proposals and classification. To avoid reshaping smoke regions and keep the shape of smoke plume, Cao et al. developed a spatial pyramid pooling convolutional neural network (SPPCNN) for smoky vehicle detection.

The other direction of smoky vehicle detection jointly considers detecting smoke regions and vehicles. Tao et al. [6] used hand-crafted features to jointly detect the rears of vehicles and smoke. Cao and Lu [8] first leveraged background subtraction to detect motion rear vehicles, then applied Inception-V3 [44] and a temporal multi-layer perception (MLP) for smoke classification. In practice, a background modelling method such as the Gaussian mixed model (GMM) can be more expensive than an optimized YOLOv5s model in terms of computational cost. Wang et al. [11] resorted to an improved YOLOv5 model for fast smoky vehicle detection, which actually only detects smoke regions.

## 3. Smoky Vehicle Datasets

In the computer vision community, there exist several widely-used wildfire smoke datasets which have promoted the progress of fire and smoke recognition, e.g., KMU Fire-Smoke [45], Mivia Fire-Smoke [46], and VSD [47]. However, to date, there exist no publicly-available smoky vehicle datasets. In the deep learning era, a good qualitative dataset is crucial for algorithm performance. To this end, we collected a large-scale smoky vehicle image dataset for model training and another video dataset for algorithm comparison.

### 3.1. Dataset Collection and Annotation

To manually collect a large-scale smoky vehicle image dataset from scratch is pretty hard since there may only exist a few smoky frames in one day of surveillance footage. To this end, we created a cold boot process for data collection. First, we rented several smoky

vehicles and drove them in the view of different surveillance cameras. Then, we captured all the smoky frames and used the LabelMe toolbox (https://github.com/CSAILVision/LabelMeAnnotationTool, accessed on 1 July 2022) to annotate smoke bounding boxes. Specifically, we obtained 1000 smoke frames at this stage. After that, with the annotated smoke images, we managed to train a rough YOLOv5s smoke detector. Finally, we deployed this detector in several surveillance cameras and gathered these frames with a smoke detection threshold of 0.05.

With the cold boot process, we managed to collect thousands of samples every day. To mine real smoky vehicle frames, we hired several people to carefully check whether the collected frames had a smoke region or not. To make the verification easy, the screen was adapted to a resolution of 1920 × 1080 since all images were 1920 × 1080. If a frame contained smoke, the people were asked to annotate the smoke bounding box using LabelMe. Within several months, we had annotated 75,000 smoke frames in total. We call this the LArge-Scale SmOky Vehicle image dataset, or LaSSoV. LaSSoV includes light smoke frames and strong smoke frames, as well as crossroad smoke frames. Figure 2 illustrates several samples in LaSSoV.



**Figure 2.** Some samples in our LaSSoV dataset. Note that our dataset includes light smoke frames, strong smoke frames, as well as crossroad smoke frames.

LaSSoV-video. Since temporal information is important for smoke recognition and existing private smoky vehicle datasets are comprised of videos, we further collected long-term smoke videos for temporal model exploration and performance comparison. We called this dataset LaSSoV-video. In total, we collected 163 one-minute videos with vehicle smoke, and used PotPlayer (http://potplayer.daum.net/, accessed on 1 July 2020) to annotate the initial smoke frame and the final smoke frame. Specifically, since we may use videos for temporal model training, we randomly split LaSSoV-video into two sub sets, i.e., a training set which includes 82 videos and a testing set which contains 81 videos.

### *3.2. Dataset Comparison*

We compared our datasets with several existing datasets in Table 1. The public KMU Fire-Smoke dataset [45] was collected by the KMU CVPR Lab and includes indoor and outdoor flame, indoor and outdoor smoke, wildfire smoke, and smoke or flame-like moving objects. Figure 3a shows some frames in the KMU Fire-Smoke dataset. The Mivia Fire-Smoke dataset [46] contains 149 long-term smoke videos which were mainly captured in a wildfire environment with the occurrence of fog, cloud, etc. The smoke video dataset in [18] (see Figure 3c) is comprised of 25 short distance smoke videos. Its block annotations can be used for local feature based algorithms. The VSD(Set4) [47] was built from similar videos in [18] while all the smoke patches and random non-smoke patches were extracted for smoke recognition, see Figure 3b. Though these datasets have promoted the progress of smoke recognition and detection, they are quite different from on-road smoke emitted from vehicles.

**Table 1.** The comparison between existing datasets and ours.

| Database | Type | Total Samples | Annotation | Available |
|---|---|---|---|---|
| KMU Fire-Smoke [45] | video | 10 | category | public |
| Mivia Fire-Smoke [46] | video | 149 | category | public |
| SmokeVideos in [18] | video | 25 | block labels | public |
| VSD(Set4) [47] | patches | 10,617 | category | public |
| SEU-SmokeVeh [7] | videos | 102 | temporal segments | private |
| **LaSSoV** (Ours) | image | 75,000 | bounding boxes | public |
| **LaSSoV-video** (Ours) | video | 163 | temporal segments | public |



(**a**) KMU Fire-Smoke (**b**) VSD-Set4 (**c**) SmokeVideos (**d**) SEU-SmokeVeh

**Figure 3.** Some examples in existing datasets [7,18,45,47].

Perhaps the most similar dataset to ours is the private SEU-SmokeVeh [7] dataset, which contains 98 short video and four long videos. Each short video has one smoke vehicle, and each long video has many non-smoke vehicles and one to three smoke vehicles. The videos in total have 5937 smoke frames and 151,613 non-smoke frames. All frames in SEU-SmokeVeh have been downsampled to the size of 864 × 480 pixels. Our LaSSoV-video includes 163 high-resolution videos with 12,287 smoky vehicle frames and more than 100,000 non-smoke frames. Moreover, the LaSSoV-video covers four scenes with different weather. In addition, since in the deep learning era training data are the devil of good performance, we also provided LaSSoV, which is the largest image smoky vehicle dataset so far. It is expected to promote the progress of smoky vehicle detection.

## 4. Approach

In this section, we first provide an overview of our cascaded smoky vehicle detection framework, and then present the details of its modules.

### *4.1. Overview of Our Framework*

Due to the lack of smoky vehicle data, there are not any pure deep learning algorithms for smoky vehicle detection. Instead, background modelling and proposal detection are normally used in existing methods. Here, we propose a novel cascaded smoky vehicle detection pipeline without traditional pre-processing steps, illustrated in Figure 4.

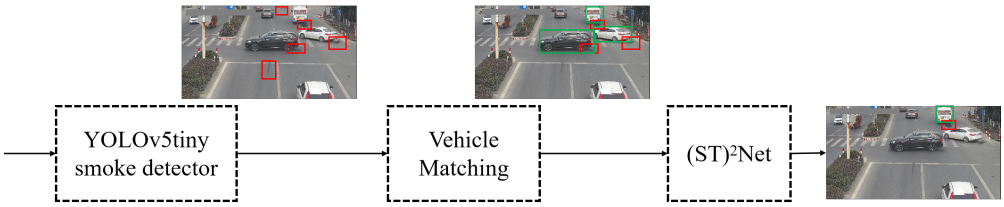



**Figure 4.** The pipeline of our cascaded smoky vehicle detection framework. The output of each module is also illustrated.

Our cascaded framework was comprised of three modules, namely a light-weight smoke detector, a vehicle matching module, and a short-term spatial-temporal network. The light-weight smoke detector, i.e., YOLOv5tiny, was re-designed from the famous YOLOv5 pipeline, which has a minimum width and depth in the architecture and largely uses depth-wise and point-wise convolutional operations to replace traditional convolutional operations. The light-weight smoke detector detects smoke regions in each video frame and may be confused by shadow and non-road regions since their appearances are very similar to vehicle smoke, see Figure 4. The vehicle matching module is activated once a smoke region is detected in a frame. This module first runs a vehicle detector to obtain vehicle regions, which can be optimized together in YOLOv5tiny. We roughly used the off-the-shelf (on COCO dataset) YOLOv5n, which can detect three vehicle categories, i.e., car, bus, and truck. With the assumption that each smoke region corresponds to a vehicle, the vehicle matching module then computes the Intersection-over-Union (IoU) ratios and pair-wise distances between smoke regions and vehicle regions. If the minimum distance was smaller than a given threshold or the IoU was larger than a threshold, we made the given smoke region and the certain vehicle a smoke–vehicle pair. The smoke–vehicle matching module can remove these false positive regions easily, such as non-road and other non-vehicle regions, see Figure 4. To further remove hard false positives, another short-term spatial-temporal model was applied on the spatial-temporal extent of the remaining regions. Overall, our framework is accurate and effective and largely leverages prior knowledge and deep learning technology.

### *4.2. YOLOv5tiny Smoke Detector*

As the first stage of our framework, the smoke detector should make a trade off between speed and accuracy. It is well-known that the YOLO framework is one of the most famous object detection algorithms due to its speed and accuracy. Thus, we chose YOLO as the frame-based smoke detector. Since the basic YOLOv1 [40] framework, there have been several versions and the recent popular one is YOLOv5. In YOLOv5, there exist several model structures which have different computation costs and parameters, and YOLOv5n (Nano) is the smallest one with 1.9 M parameters and about 4.5 GFLOPs, which can be run in real-time with CPU solutions.

Our light-weight YOLOv5tiny smoke detector was transferred from YOLOv5n with even less computation cost. Figure 5 illustrates its structure. Specifically, YOLOv5tiny first decreases the number of each *Conv.* block in the YOLOv5n backbone to one, and then replaces the C3 module with a so-called C3Ghost module inspired by MobileNet [36] and GhostNet [37]. Figure 6 shows the comparison between C3 and C3Ghost. As it is known in MobileNet, splitting a traditional convolution operation into a depth-wise and a point-wise convolution can save computation cost and parameters. The GhostNet demonstrates that the redundancy in feature maps (i.e., ghost features) can be obtained by a series of cheap linear transformations, resulting in a new GhostConv module. Supposing we set $n$ output feature maps, the standard GhostConv module first uses $m(m < n)$ traditional convolutional kernels with a non-linear activation function to generate $m$ feature maps, and then utilizes depth-wise convolution without non-linear operations to obtain other $s$ ghost feature maps, where $m + s = n$. We replaced the traditional convolutional layers of the C3 module with the plug-and-play GhostConv module, and obtained the C3Ghost module.

With these changes, our smoke detector only had 1.2M parameters and 2.8 GFLOPs, which reduces the computation cost of YOLOv5n significantly.

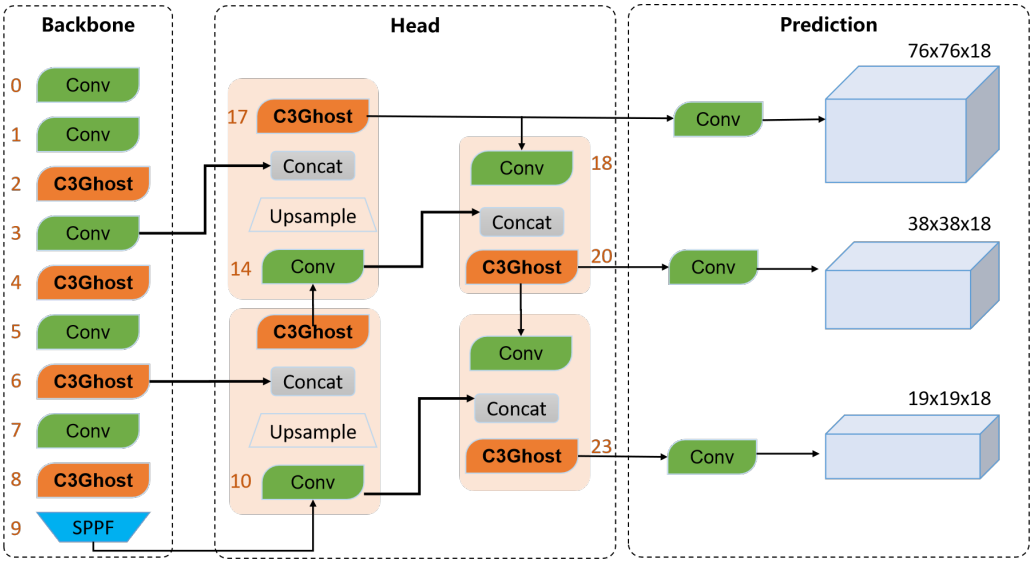
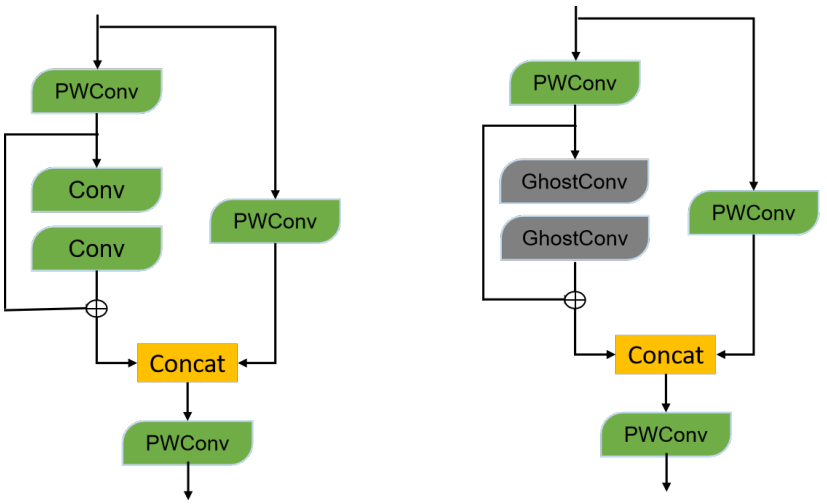



**Figure 5.** Our re-designed YOLOv5 vehicle smoke detector, termed YOLOv5tiny. It is modified from the YOLOv5 yet with fewer layers and GhostConv modules.




(a) The C3 module (b) The modified C3 module (C3Ghost)

**Figure 6.** Our light-weight YOLOv5 vehicle smoke detector.

### *4.3. Vehicle Matching*

For the pure YOLOv5 smoke detector, false positives are inevitable since it may return all the smoke-like regions. These false positives are serious, especially on wet roads, shadow roads, cluttered backgrounds, etc. Worse, these false positives are difficult to remove through frame-wise smoke detection. Considering that each instance of vehicle smoke is emitted from a certain vehicle, we introduced a smoke-vehicle matching strategy to refine.

Figure 7 depicts our smoke–vehicle matching strategy. Once a smoke region was detected, we resorted to using the YOLOv5n model pretrained on COCO data to detect vehicles. The vehicles included car, bus, and truck in the COCO dataset. We also tried to train a vehicle detector on our own vehicle data but found that it was less robust than the pretrained one. After vehicle detection, we obtained both the vehicle regions and smoke regions. As shown in Figure 7, there would be two cases that we needed to deal with

separately. For both cases, considering the smoke region usually comes from the back of vehicles, we only considered the front vehicles of a certain smoke region as possible smoky vehicles, i.e., the vehicle with its center above the smoke center. The first case is that the smoke region is overlapped with vehicle regions. For this case, our strategy regarded the largest overlapped vehicle as the smoky vehicle, see the green regions on the left of Figure 7. For the second case where there are no overlapped vehicles, we first computed the Euclidean distances between the bottom-middle points of the vehicles and the top-left (also top-middle and top-right) points of smoke regions, and then we set a threshold $L_{dist}$ for the mean distance. A smoke–vehicle pair keeps the smallest mean distance that is less than $L_{dist}$, see the green regions on the right of Figure 7.

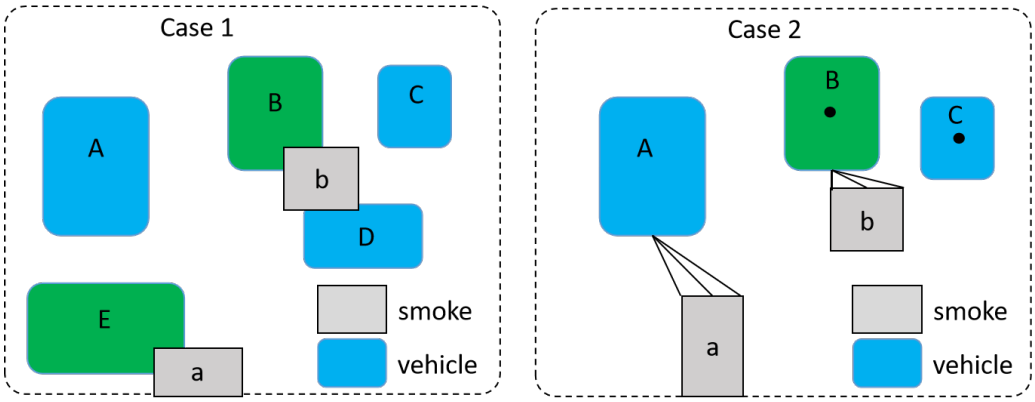



**Figure 7.** Two common cases in our smoke–vehicle matching strategy. Case 1 refers to smoke regions overlapping with vehicles, while case 2 refers to there being no overlap between smoke regions and vehicles. Green boxes denote smoky vehicles.

One could argue for the first case that it would make mistakes if there is a traffic jam. This is the truth in practice but still we have captured vehicle smoke, and maybe a further check from a human is needed. For the second case, the reason we did not just use the center points between vehicles and smoke regions for distance computation is that smoky vehicles are usually those large trucks whose center points can be far from smoke regions.

### *4.4. Short-Term Spatial-Temporal Network*

After the vehicle matching stage, though detections that cannot connect to a certain vehicle were removed, those false positives from vehicle shadow regions could not be filtered effectively. We found that those vehicle shadow regions are difficult even for human beings if only the region of a frame is available. However, when we checked the original video stream, we found that it was possible for non-expert people to filter these false positives. We argue that the reason could be that more information is available in space time, see the top of Figure 8.

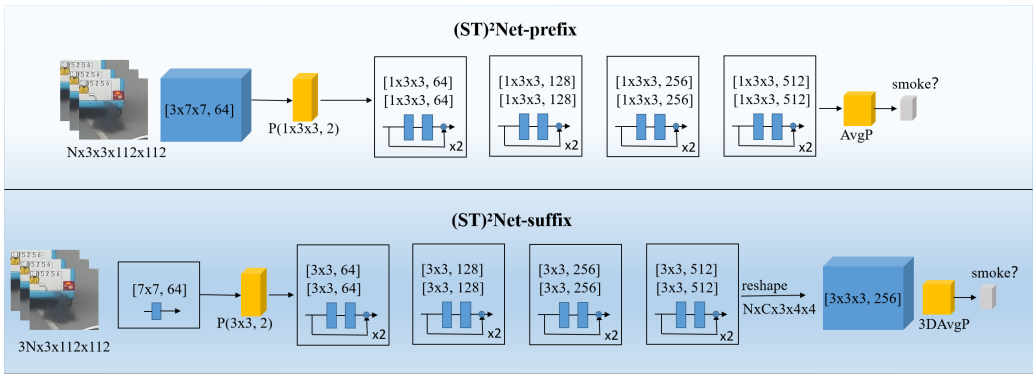



**Figure 8.** Our (ST)$^2$Net alternatives for spatial-temporal smoke recognition.

Inspired by the above fact, we extended the detected regions to larger square ones and designed two kinds of short-term spatial-temporal neural networks (ST)$^2$Net to handle those shadow regions. Specifically, if the largest side of a detected region was less than 112, we extended it to an 112 × 112 large region sharing the same center. Otherwise, we extended a detected region to a square region with the side as the largest side of the detected region. We also extended a detected frame region in time space, i.e., cropping the same region in $K$ (e.g., 3, 5, 9, etc) frames. For the first (ST)$^2$Net model, which is called (ST)$^2$Net-prefix, we took the ResNet18 as a template and adjusted the first several layers to 3D convolutional layers according to the $K$, see the left of Figure 8 when $K = 3$. For the second (ST)$^2$Net model in the right of Figure 8, we added a 3D convolutional layer after the last 2D convolutional layer according to the $K$, and then conducted adaptive pooling. We call this architecture (ST)$^2$Net-suffix. Intuitively, (ST)$^2$Net-prefix may be better at capturing low-level smoke motion features, while (ST)$^2$Net-suffix keeps static high-level abstract features for further motion analysis. An evaluation will be presented in the next section.

## 5. Experiments

### *5.1. Data Splits and Metrics*

Our LaSSoV image dataset contains 75,000 annotated smoke images, and we randomly split it into a training set and a testing set with 70,000 images and 5000 images, respectively. We compared our YOLOv5tiny smoke detector to others on the testing set. For our LaSSoV-video dataset, we also randomly split it into a training set and a testing set with 82 videos and 81 videos. The testing set contains 6727 smoky frames, which is slightly more than the private SEU-SmokeVeh [7]. Since existing methods do not mention the accuracy of smoke bounding boxes (i.e., IoU), we compared our method to others on the testing set of LaSSoV-video in terms of frame level performance.

For smoke detector evaluation, following popular detection metrics, we used mAP@0.5 IoU (Intersection over Union) and mAP@0.5:0.95 IoU for performance evaluation. For smoky vehicle detection, according to existing works, we can evaluate a method in terms of frame level, vehicle level, and segment (or video) level. Following [3,7], we used detection rate (DR) and false alarm rate (FAR) in the frame level. Specifically, their formulas are as follows:

$$DR = \frac{TP}{N_{pos}} \quad FAR = \frac{FP}{N_{neg}}, \tag{1}$$

where $TP$ denotes true positive (i.e., smoky frame) and FP is false positive; $N_{pos}$ and $N_{neg}$ are the numbers of positives and negatives. It is worth noting that DR is called the probability of correct classification on smoky frames, and also that DR is identical to the Recall conception. In addition, we also used Precision and F1 score for frame-level performance evaluation, which is also important for algorithm comparison. Their formulas are as follows:

$$Precision = \frac{TP}{TP + FP} \quad F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}. \tag{2}$$

### *5.2. Implementation Details*

We used the PyTorch toolbox (https://pytorch.org/, accessed on 1 July 2022) to implement our method on a Linux server with two GeForce RTX 3090 GPUs. Specifically, YOLO based smoke detectors were implemented with the open source code (https://github.com/ultralytics/yolov5, accessed on 1 July 2022), and were trained from scratch. We set the initial learning rate to 0.01, batch size to 64, and training epochs to 30. All other parameters were set by default. The threshold $L_{dist}$ of the smoke–vehicle matching module was set by default to 50 which meant that the distance between a smoke region and a vehicle should be less than 50 pixels.

For the training data of (ST)$^2$Net models, we first applied our YOLOv5tiny smoke detector (with a score threshold of 0.2) and smoke–vehicle matching module to the LaSSoV-video training set to collect sequence samples. The temporal extent $K$ was set by default

to 3, and 10,033 sequences were finally obtained to train the (ST)$^2$Net models. For data augmentation, we resized the crop regions in sequences to 128 × 128 and then randomly cropped 112 × 112 regions for training. We set the initial learning rate to 0.01, and decreased it by a factor of 0.1 after every four epochs, and stopped training at the 10th epoch.

### *5.3. Detector Evaluation on LaSSoV*

We evaluated several variations of YOLOv5, including YOLOv5s, YOLOv5n, and MobileNetv2 [36] backbone-based YOLOv5s. Table 2 compares their mean average precision and computation cost. For the computation cost, we ran the model three times on three smoke samples and reported the average cost. From Table 2, several observations can be made as follows. First, replacing the default YOLOv5s backbone with MobileNetv2 achieves the worst results and is not efficient. It is worth noting that the recent YOLOv5 architecture has already introduced efficient modules and is better optimized than MobileNetv2. Second, YOLOv5s obtained the best accuracy while needing more than twice the inference cost compared to YOLOv5n and our YOLOv5tiny. Finally, compared to YOLOv5n, our model obtained a similar performance but had fewer FLOPs, and was more efficient in CPU due to the Ghost module.

**Table 2.** Evaluation of different deep learning detectors on LaSSoV. All the detectors are trained on the training set of LaSSoV and the results are reported on the test set of LaSSoV with image size 640. The speeds are reported on three randomly-selected smoke images with average inference time.

| Detector | mAP@0.5 | mAP@0.5:0.95 | Params (M) | FLOPs | Speed-CPU (ms) | Speed-GPU (ms) |
|---|---|---|---|---|---|---|
| MobileNetv2-YOLOv5s | 64.52 | 22.96 | 4.6 | 36.7 | 470.6 | 13.1 |
| YOLOv5s | 87.3 | 39.8 | 7.2 | 16.5 | 134.5 | 18.5 |
| YOLOv5n | 80.9 | 33.1 | 1.9 | 4.5 | 61.5 | 13.2 |
| Our YOLOv5tiny | 79.8 | 32.1 | 1.2 | 2.8 | 48.9 | 12.8 |

### *5.4. Comparison to the State of the Art on LaSSoV-Video*

As mentioned above, our LaSSoV-video contains diverse scenes which could be challenging for existing methods. We evaluated the pure YOLOv5s, our YOLOv5tiny, and our cascaded framework on the test set (four scenes) of LaSSoV-video and reported the average performance. The detection threshold was set to 0.2 for our YOLOv5tiny. For the pure YOLOv5s and YOLOv5tiny, we reported the results with detection thresholds of both 0.2 and 0.5. We re-implemented the R-VLBC [7] on our LaSSoV-video and also compared it to other methods on the private SEU-SmokeVeh (results were copied from the reference papers). Unlike existing works, which only use one or two metrics, we applied four popular frame-wise metrics for evaluation.

From Table 3, several observations can be made as follows. First, from the first and the third rows, we find that YOLOv5s obtains the highest detection ratio (or recall ratio) while the performance gap between YOLOv5s and our YOLOv5tiny is not as large as the one in the LaSSoV image data. This suggests that the lightweight YOLOv5tiny is enough to learn the main features of obvious smoke regions while YOLOv5s can struggle with hard cases. Second, increasing the detection threshold to 0.5 for YOLOv5s and YOLOv5tiny largely improves precision and degrades detection rates, while still boosting F1 scores. Third, (ST)$^2$Net-suffix is consistently superior to (ST)$^2$Net-prefix for our cascaded framework, which may be explained by the motion information in high-level layers being better for classification. We fixed (ST)$^2$Net-suffix as the (ST)$^2$Net module in the remaining content unless otherwise specified. Fourth, our cascaded method significantly outperforms the others in Precision and F1, while still keeping reasonable detection rates. Specifically, based on the results of YOLOv5tiny(0.2), our method obtains around a 25% gain in F1 score. Last but not least, the R-VLBC [7] method that leverages hand-crafted local pattern features

achieves the worst performance in all metrics, which indicates the limitations of these features.

**Table 3.** Comparison between our method and state-of-the-art methods with varied metrics. The first four methods are evaluated on our LaSSoV-video, and the others are reported in existing papers on the private SEU-SmokeVeh (Dataset 5) [7], which is similar to ours in data scale. The values in parentheses denotes the detection thresholds of YOLOv5. The time is reported on CPU for one frame.

| Method | DR | FAR (↓) | Precision | F1 | Time/ms |
|---|---|---|---|---|---|
| YOLOv5s(0.2) | 0.9368 | 0.1349 | 0.4064 | 0.5580 | 134.5 |
| YOLOv5s(0.5) | 0.8471 | 0.0453 | 0.6379 | 0.7206 | 134.5 |
| YOLOv5tiny(0.2) | 0.9087 | 0.1436 | 0.3967 | 0.5370 | 48.9 |
| YOLOv5tiny(0.5) | 0.7168 | 0.0287 | 0.7000 | 0.6949 | 48.9 |
| Ours($(ST)^2$Net-prefix) | 0.7381 | 0.0297 | 0.7045 | 0.7200 | 68.2 + 61.5 |
| Ours($(ST)^2$Net-suffix) | 0.7713 | 0.0192 | 0.7936 | 0.7801 | 72.2 + 61.5 |
| R-VLBC [7] | 0.8453 | 0.1893 | 0.3171 | 0.4623 | 160.56 |
| R-VLBC [7] | 0.9253 | 0.0893 | – | – | |
| LBP [3] | 0.8674 | 0.3103 | – | – | – |
| NR-RLBP [3] | 0.9125 | 0.1610 | – | – | 140.34 |
| RF-TOP [48] | 0.8757 | 0.1236 | – | – | – |

To further investigate their performance in different scenes, Figure 9 illustrates the comparison of F1 score in each scene. The traditional R-VLBC method is consistently worse than deep learning methods especially in scene #3 (cross roads). There exist many varied vehicle shadows in scene #3, which largely confuse those static smoke detectors. YOLOv5s(0.2) and YOLOv5tiny(0.2) perform well and similarly in simple scenes (i.e., scene #1 and #2). Scene #4 shows a partially wet road and the wet regions can be easily mistaken as smoke regions by YOLOv5s and YOLOv5tiny, leading to low F1 scores. Our method is significantly superior to other methods in all scenes thanks to the $(ST)^2$Net classifier. Specifically, our method improves the F1 of YOLOv5s by around 43% in scene #3, indicating that $(ST)^2$Net is robust to different roads.

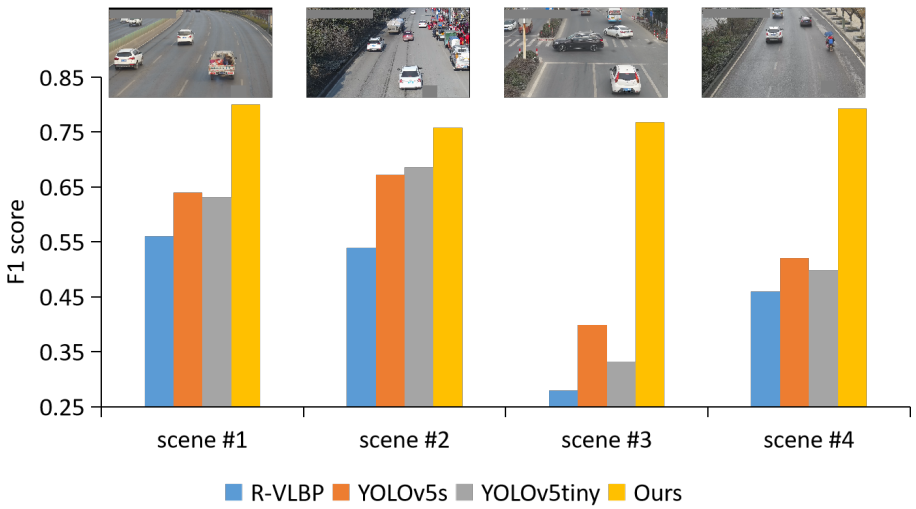



**Figure 9.** The F1 scores on four scenes. To clarify, we illustrate a sample frame for each scene at the top of the figure.

Figure 10 further illustrates some detection samples in our LaSSoV-video test set. The yellow bounding boxes denote those false positives of YOLOv5tiny with scores larger than 0.4. These false positives are even difficult for human beings in static images, which are

either classified as the non-smoke category in our (ST)$^2$Net-suffix model or filtered out by our vehicle matching module (the top-middle one).



**Figure 10.** Detection samples in LaSSoV-video test set. Yellow bounding boxes denote those false positives of YOLOv5tiny, and green bounding boxes refer to those results of our cascaded method. In other words, the yellow boxes are filtered by the vehicle matching module and (ST)$^2$Net.

In Table 3, we also report the computational cost on the CPU (Intel Xeon E5-2678 v3) for processing one frame. Specifically, the cost of our method depends on the YOLOv5tiny, the (ST)$^2$Net, and the extra YOLOv5n for vehicle detection (61.5 ms). It is worth noting that all other methods did not provide vehicle regions yet it is necessary for practical application. In general, our method is efficient on the CPU platform and obtains a 78.01% F1 score with a speed of around 8 FPS (frames per second).

### *5.5. Ablation Study*

Since our framework included two auxiliary modules compared to pure YOLO detectors, we conducted an evaluation of these two modules, presented in Table 4. From Table 4, we can see that the vehicle matching module slightly improves YOLOv5tiny in almost all metrics except for the detection rate (a slight degradation), and the (ST)$^2$Net module achieves larger performance gains than the vehicle matching module. Specifically, for the F1 scores, the vehicle matching module contributes a 2.5% improvement while the (ST)$^2$Net module contributes 22%.

**Table 4.** Evaluation of the individual modules of our method on LaSSoV-video test set.

| YOLOv5tiny | Vehicle Matching | (ST)$^2$Net | DR | FAR (↓) | Precision | F1 |
|---|---|---|---|---|---|---|
| √ | | | 0.9087 | 0.1436 | 0.3967 | 0.5370 |
| √ | √ | | 0.8815 | 0.1211 | 0.4255 | 0.5616 |
| √ | √ | √ | 0.7713 | 0.0192 | 0.7936 | 0.7801 |

We evaluated several other common spatial-temporal video classification models including long-short term memory with CNN features (CNN-LSTM) [49], temporal segment network [50] (i.e., temporal average with CNN features, CNN-Avg), and temporal concatenation with CNN features (CNN-Cat). We replaced our (ST)$^2$Net-suffix as one of them for the evaluation. All their backbones were the same as in our (ST)$^2$Net model (i.e., ResNet-18); the CNN features are the output of the last pooling layer in ResNet-18. The results are shown in Figure 11. Several observations can be made as follows. First, our (ST)$^2$Net-suffix obtains the best F1 score and Precision which suggests that the abstract motion information may be captured by 3D convolutions. Second, CNN-Avg and CNN-Cat perform similarly in all metrics since they actually use the same static CNN features of three frames. Last but not least, an LSTM model upon CNN features keeps high recall but is not effective in distinguishing smoke regions.

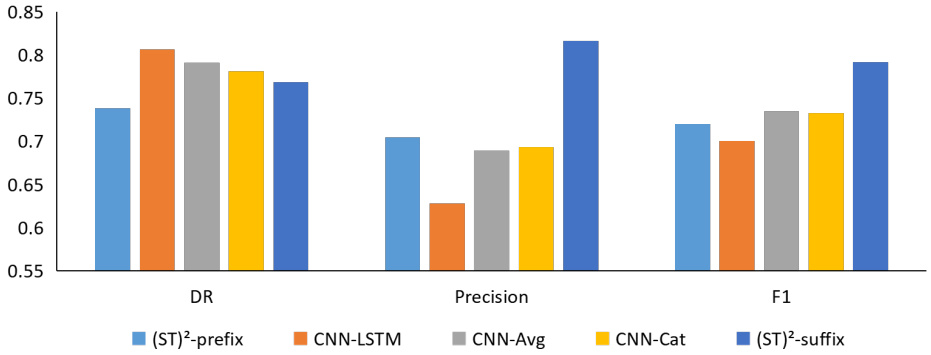



**Figure 11.** The comparison of different short-term spatial-temporal models in metrics of DR, precision, and F1 score.

As mentioned above, the default temporal input length is 3 for our (ST)$^2$Net modules. In fact, this short temporal extent is unusual in other similar tasks such as video action recognition [49–51]. Thus, we evaluated the temporal length as shown in Figure 12. We do not show the false alarm rates since all of these models obtain less than 0.02 in FAR. We took ResNet-18 to fine tune an image classification model for length 1. As can be seen in Figure 12, the F1 score and precision are improved when increasing the length but are saturated at length 3. The detection rate is slightly degraded with increasing input length. We show some temporal extent samples in Figure 13 to investigate why more frames are not helpful for performance. We find that most of the extented frame regions are not related to smoke, which is mainly caused by the high-speed vehicles and low frame rate cameras.

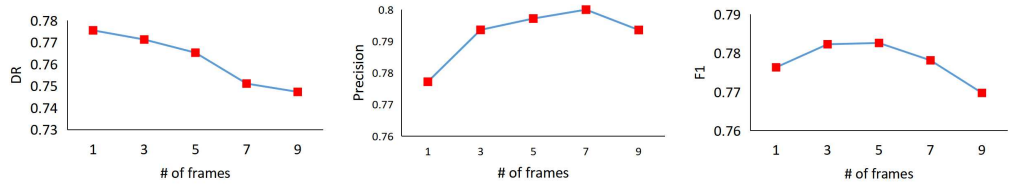



**Figure 12.** Evaluation of temporal length for our (ST)$^2$Net-suffix.

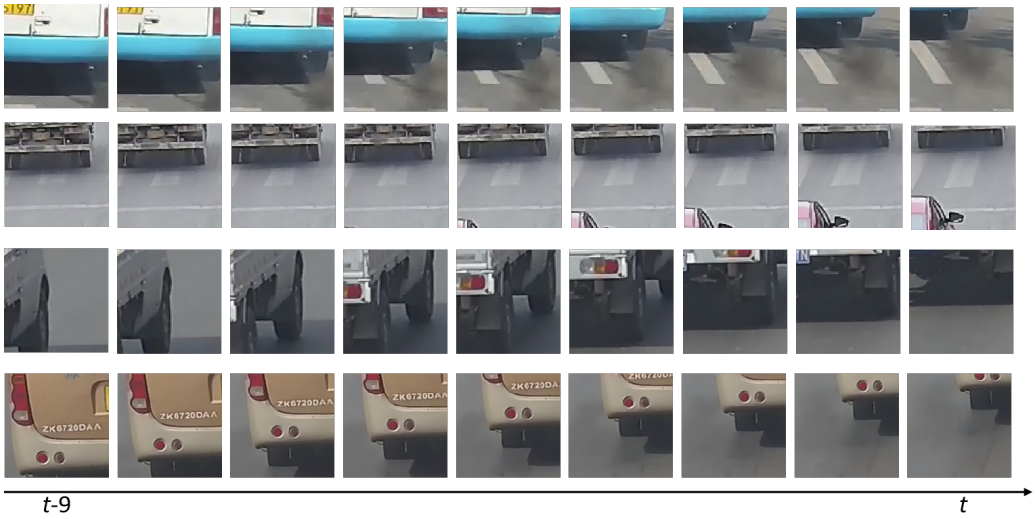



**Figure 13.** Temporal extent examples for detected smoke regions in time *t*.

As known in image classification, a large input size is helpful for performance. To verify this, we rescaled the input to 256 × 256, and randomly cropped 224 × 224 regions to train our (ST)$^2$Net-suffix model with temporal length 3. Replacing the default (ST)$^2$Net-suffix with this one, we achieved 0.7552, 0.0176, 0.8037, and 0.7763, in DR, FAR, Precision,

and F1, respectively. Compared to the results in Table 3, we find that only Precision is slightly improved while there is an obvious degradation in DR or F1. This may be explained by most of the cropping 'smoke' regions being smaller than 224 since smoky vehicles are far away from the cameras, and zoom-in regions can disturb the smoke characteristics.

## 6. Conclusions and Future Work

In this paper, we explored the issue of smoky vehicle detection. We first built a large-scale smoky vehicle image dataset, LaSSoV, which was used for training deep learning models. As there are no public smoky vehicle test videos, we also provided a video dataset called LaSSoV-video for algorithm comparison. Moreover, we presented a novel cascaded smoky vehicle detection framework, which was demonstrated to be efficient on the CPU platform and is superior to previous advanced methods. In addition, we comprehensively evaluated our method and others in four metrics which is more fair than only in one or two. The limit of our work is that we cannot continuously improve the accuracy by adding more frames. As found in our $(ST)^2$Net module, there are many irrelevant regions when increasing the temporal length; we think a tracking process after detection can be better, which is the focus of our future work.

**Author Contributions:** Conceptualization, X.P. and J.Z.; methodology, X.P. and X.F.; software, X.P. and P.G.; data curation, Q.W. and J.Z.; writing—original draft preparation, X.P. and X.F.; writing—review and editing, Q.W. and X.F. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is partially supported by the National Natural Science Foundation of China (62176165), the Stable Support Projects for Shenzhen Higher Education Institutions (SZWD2021011), the Natural Science Foundation of Top Talent of SZTU (GDRC202131), the Basic and Applied Basic Research Project of Guangdong Province (2022B1515130009), and the Special subject on Agriculture and Social Development, Key Research and Development Plan in Guangzhou (2023B03J0172).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data available on request due to restrictions.

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

## References

1. Beaton, S.P.; Bishop, G.A.; Zhang, Y.; Stedman, D.H.; Ashbaugh, L.L.; Lawson, D.R. On-Road Vehicle Emissions: Regulations, Costs, and Benefits. *Science* **1995**, *268*, 991–993. [CrossRef]
2. Ropkins, K.; Beebe, J.; Li, H.; Daham, B.; Tate, J.; Bell, M.; Andrews, G. Real-World Vehicle Exhaust Emissions Monitoring: Review and Critical Discussion. *Crit. Rev. Environ. Sci. Technol.* **2009**, *39*, 79–152. [CrossRef]
3. Tao, H.; Zheng, P.; Xie, C.; Lu, X. A three-stage framework for smoky vehicle detection in traffic surveillance videos. *Inf. Sci.* **2020**, *522*, 17–34. [CrossRef]
4. Tao, H.; Lu, X. Smoky vehicle detection based on multi-scale block Tamura features. *Signal Image Video Process.* **2018**, *12*, 1061–1068. [CrossRef]
5. Cao, Y.; Lu, C.; Lu, X.; Xia, X. A Spatial Pyramid Pooling Convolutional Neural Network for Smoky Vehicle Detection. In Proceedings of the 2018 37th Chinese Control Conference (CCC), Wuhan, China, 25–27 July 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 9170–9175.
6. Tao, H.; Lu, X. Automatic smoky vehicle detection from traffic surveillance video based on vehicle rear detection and multi-feature fusion. *IET Intell. Transp. Syst.* **2019**, *13*, 252–259. [CrossRef]
7. Tao, H.; Lu, X. Smoke vehicle detection based on robust codebook model and robust volume local binary count patterns. *Image Vis. Comput.* **2019**, *86*, 17–27. [CrossRef]
8. Cao, Y.; Lu, X. Learning spatial-temporal representation for smoke vehicle detection. *Multimed. Tools Appl.* **2019**, *78*, 27871–27889. [CrossRef]
9. Tao, H.; Lu, X. Smoke Vehicle Detection Based on Spatiotemporal Bag-Of-Features and Professional Convolutional Neural Network. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 3301–3316. [CrossRef]
10. Tao, H.; Lu, X. Smoke vehicle detection based on multi-feature fusion and hidden Markov model. *J. Real-Time Image Process.* **2020**, *17*, 745–758. [CrossRef]

11. Wang, C.; Wang, H.; Yu, F.; Xia, W. A High-Precision Fast Smoky Vehicle Detection Method Based on Improved Yolov5 Network. In Proceedings of the 2021 IEEE International Conference on Artificial Intelligence and Industrial Design (AIID), Guangzhou, China, 28–30 May 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 255–259.
12. Hashemzadeh, M.; Farajzadeh, N.; Heydari, M. Smoke detection in video using convolutional neural networks and efficient spatio-temporal features. *Appl. Soft Comput.* **2022**, *128*, 109496. [CrossRef]
13. Sun, B.; Xu, Z.D. A multi-neural network fusion algorithm for fire warning in tunnels. *Appl. Soft Comput.* **2022**, *131*, 109799. [CrossRef]
14. Töreyin, B.U.; Dedeoğlu, Y.; Cetin, A.E. Wavelet based real-time smoke detection in video. In Proceedings of the 2005 13th European Signal Processing Conference, Antalya, Turkey, 4–8 September 2005; IEEE: Piscataway, NJ, USA, 2005; pp. 1–4.
15. Xiong, Z.; Caballero, R.; Wang, H.; Finn, A.M.; Lelic, M.A.; Peng, P.Y. Video-based smoke detection: Possibilities, techniques, and challenges. In Proceedings of the IFPA, Fire Suppression and Detection Research and Applications—A Technical Working Conference (SUPDET), Orlando, FL, USA, 11–13 March 2007.
16. Yuan, F. A fast accumulative motion orientation model based on integral image for video smoke detection. *Pattern Recognit. Lett.* **2008**, *29*, 925–932. [CrossRef]
17. Zhou, Z.; Shi, Y.; Gao, Z.; Li, S. Wildfire smoke detection based on local extremal region segmentation and surveillance. *Fire Saf. J.* **2016**, *85*, 50–58. [CrossRef]
18. Lin, G.; Zhang, Y.; Zhang, Q.; Jia, Y.; Wang, J. Smoke detection in video sequences based on dynamic texture using volume local binary patterns. *Ksii Trans. Internet Inf. Syst.* **2017**, *11*, 5522–5536.
19. Yuan, F.; Xia, X.; Shi, J.; Zhang, L.; Huang, J. Learning multi-scale and multi-order features from 3D local differences for visual smoke recognition. *Inf. Sci.* **2018**, *468*, 193–212. [CrossRef]
20. Yuan, F.; Shi, J.; Xia, X.; Zhang, L.; Li, S. Encoding pairwise Hamming distances of Local Binary Patterns for visual smoke recognition. *Comput. Vis. Image Underst.* **2019**, *178*, 43–53. [CrossRef]
21. Yuan, F.; Zhang, L.; Wan, B.; Xia, X.; Shi, J. Convolutional neural networks based on multi-scale additive merging layers for visual smoke recognition. *Mach. Vis. Appl.* **2019**, *30*, 345–358. [CrossRef]
22. Yuan, F.; Li, G.; Xia, X.; Lei, B.; Shi, J. Fusing texture, edge and line features for smoke recognition. *IET Image Process.* **2019**, *13*, 2805–2812. [CrossRef]
23. Yuan, F.; Li, G.; Xia, X.; Shi, J.; Zhang, L. Encoding features from multi-layer Gabor filtering for visual smoke recognition. *Pattern Anal. Appl.* **2020**, *23*, 1117–1131. [CrossRef]
24. Yuan, F. A double mapping framework for extraction of shape-invariant features based on multi-scale partitions with AdaBoost for video smoke detection. *Pattern Recognit.* **2012**, *45*, 4326–4336. [CrossRef]
25. Vapnik, V.N. Statistical Learning Theory. *Encycl. Sci. Learn.* **1998**, *41*, 3185.
26. Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the NIPS, Carson, NV, USA, 3–6 December 2012.
27. Filonenko, A.; Kurnianggoro, L.; Jo, K.H. Comparative study of modern convolutional neural networks for smoke detection on image data. In Proceedings of the International Conference on Human System Interactions, Portsmouth, UK, 6–8 July 2017.
28. Li, X.; Chen, Z.; Wu, Q.; Liu, C. 3D Parallel Fully Convolutional Networks for Real-time Video Wildfire Smoke Detection. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *30*, 89–103. [CrossRef]
29. Lin, G.; Zhang, Y.; Xu, G.; Zhang, Q. Smoke Detection on Video Sequences Using 3D Convolutional Neural Networks. *Fire Technol.* **2019**, *55*, 1827–1847. [CrossRef]
30. Gu, K.; Xia, Z.; Qiao, J.; Lin, W. Deep Dual-Channel Neural Network for Image-Based Smoke Detection. *IEEE Trans. Multimed.* **2019**, *22*, 311–323. [CrossRef]
31. Yin, M.; Lang, C.; Li, Z.; Feng, S.; Wang, T. Recurrent convolutional network for video-based smoke detection. *Multimed. Tools Appl.* **2019**, *78*, 237–256. [CrossRef]
32. Hu, Y.; Lu, X. Real-time video fire smoke detection by utilizing spatial-temporal ConvNet features. *Multimed. Tools Appl.* **2018**, *77*, 29283–29301. [CrossRef]
33. Luo, Y.; Zhao, L.; Liu, P.; Huang, D. Fire smoke detection algorithm based on motion characteristic and convolutional neural networks. *Multimed. Tools Appl.* **2018**, *77*, 15075–15092. [CrossRef]
34. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NA, USA, 27–30 June 2016.
35. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
36. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
37. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C. GhostNet: More Features From Cheap Operations. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
38. Guillemant, P.; Vicente, J.R.M. Real-time identification of smoke images by clustering motions on a fractal curve with a temporal embedding method. *Opt. Eng.* **2001**, *40*, 554–563. [CrossRef]

39. Gomez-Rodriguez, F.; Pascual-Pena, S.; Arrue, B.; Ollero, A. Smoke detection using image processing. In Proceedings of the International Conference on Forest Fire Research &17th International Wildland Fire Safety Summit (ICFFR), Coimbra, Portugal, 11–18 November 2002.
40. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the Computer Vision & Pattern Recognition, Las Vegas, NA, USA, 27–30 June 2016.
41. Frizzi, S.; Kaabi, R.; Bouchouicha, M.; Ginoux, J.M.; Moreau, E.; Fnaiech, F. Convolutional neural network for video fire and smoke detection. In Proceedings of the Conference of the IEEE Industrial Electronics Society, Florence, Italy, 24–27 October 2016; pp. 877–882.
42. Lecun, Y.; Bottou, L. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
43. Tao, C.; Jian, Z.; Pan, W. Smoke Detection Based on Deep Convolutional Neural Networks. In Proceedings of the 2016 International Conference on Industrial Informatics—Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII), Wuhan, China, 3–4 December 2016.
44. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Wuhan, China, 3–4 December 2016; pp. 2818–2826.
45. Ko, B.; Kwak, J.Y.; Nam, J.Y. Wildfire smoke detection using temporospatial features and random forest classifiers. *Opt. Eng.* **2012**, *51*, 7208.
46. 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]
47. Yuan, F.; Shi, J.; Xia, X.; Fang, Y.; Fang, Z.; Mei, T. High-order local ternary patterns with locality preserving projection for smoke detection and image classification. *Inf. Sci.* **2016**, *372*, 225–240. [CrossRef]
48. Tao, H.; Lu, X. Smoky vehicle detection based on range filtering on three orthogonal planes and motion orientation histogram. *IEEE Access* **2018**, *6*, 57180–57190. [CrossRef]
49. Ng, Y.H.; Hausknecht, M.; Vijayanarasimhan, S.; Vinyals, O.; Toderici, G. Beyond short snippets: Deep networks for video classification. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
50. Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; Gool, L.V. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016.
51. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning Spatiotemporal Features with 3D Convolutional Networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015.


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