2020 25th International Conference on Pattern Recognition (ICPR)
Milan, Italy, Jan 10-15, 2021

Early Wildfire Smoke Detection in Videos

Taanya Gupta, Hengyue Liu and Bir Bhanu
Department of Electrical and Computer Engineering
University of California, Riverside, CA 92521, USA
Email: tgupt001@ucr.edu, hliu087@ucr.edu, bhanu@vislab.ee.ucr.edu

***Abstract*—Recent advances in unmanned aerial vehicles and camera technology have proven useful for the detection of smoke that emerges above the trees during a forest fire. Automatic detection of smoke in videos is of great interest to Fire department. To date, in most parts of the world, the fire is not detected in its early stage and generally it turns catastrophic. This paper introduces a novel technique that integrates spatial and temporal features in a deep learning framework using semi-supervised spatio-temporal video object segmentation and dense optical flow. However, detecting this smoke in the presence of haze and without the labeled data is difficult. Considering the visibility of haze in the sky, a dark channel pre-processing method is used that reduces the amount of haze in video frames and consequently improves the detection results. Online training is performed on a video at the time of testing that reduces the need for ground-truth data. Tests using the publicly available video datasets show that the proposed algorithms outperform previous work and they are robust across different wildfire-threatened locations.**

I. Introduction

The unpredictable behavior and continuous expansion of wildfires are a threat to both humans and the environment. According to the year-to-date statistics of the National Interagency Fire Center (NIFC) [1], 2,511 fires burnt 43,219 acres of land in the United States in the first two months of 2020. The common approach to research in this area has been to develop solutions that detected fire [2]. But if a fire can be detected at an early stage, such as when smoke first appears, it can be easily extinguished and it will require the use of fewer resources; this can be both economically beneficial and reduce the impact on humans. Thus, research has been done on the early detection of smoke as it emerges above the crown of the trees in the early stages of a wildfire. This detection can be facilitated by the use of a camera mounted on a tower or an Unmanned Aerial Vehicle (UAV) that captures on-site evidence of smoke in a video [3]. Fortunately, the market has recently been flooded with high-end cameras that can easily capture distant smoke.

Traditional smoke detection techniques like [4] use expert analysis of extracted features. These methods are not able to differentiate smoke pixels from the surrounding pixels when the image is comprised of regions with similar semantics such as haze and clouds. Color has long been considered a unique feature for detection [5], but the color of smoke can be easily confused with that of clouds, fog, or other similar objects. An approach using deep learning can capture the ambiguity of features, reducing the number of false negative detections. This deep-learning approach could be enormously significant, as an unnoticed wildfire can have disastrous consequences.

The long-established detection methods for identifying smoke in an image did not consider pixel-level detection, which can give an exact location of the smoke. It is also important to note that the structure of smoke is not definitive: it changes in each frame of a video. To segment out smoke as an anomaly, a deeper segmentation network is required. There are many image segmentation work like U-Net [6], but they only perform image level segmentaions based on spatial features. For temporal features early efforts used optical flow to extract the motion features [7], and a few methods such as [8] also computed the spatio-temporal features given by a Histogram of Oriented Gradients (HOG) and a Histogram of Oriented Optical Flow (HOOF) that define the spatial and temporal features, respectively.

The proposed work is an amalgamation of video segmentation and dense optical flow used to identify spatio–temporal features. Unlike the traditional methods that rely on motion, color, or hand-picked features, the proposed work uses deep convolutional neural networks. This method extracts the temporal information from the dense optical flow and processes it through a semi-supervised segmentation network that is a Fully Convolutional Network (FCN) [9]. The lack of availability of labeled data for smoke detection presented a problem; to overcome this, the proposed method uses a minimum amount of labeled data by using an online training method. To improve the image quality, we adopted a de-hazing method that enhances the visibility of the smoke using the physical parameters behind it.

The remainder of the paper will be organized as follows: Section II discusses the previous work on wildfire detection, Section III presents the proposed technical approach, Section IV describes the experimental setup and results, and Section V presents the conclusions.

II. Related Work and Contributions

A. Related Work

1) State-of-the-Art of Wildfire Detection: Initially, sensors were the principal tools used in wildfire detection. As the field advanced to computer vision techniques, variations in color and geometry were used to identify fire. Numerous color spaces were used to extract the chromatic information [10], [11]. The red component was pivotal in some studies [12], and one previous work [13] used a combination of motion and color features. With the accessibility of high-resolution video cameras, video replaced static images. A few studies involved the characterization and detection of smoke observed

978-1-7281-8808-9/20/$31.00 ©2020 IEEE

2020 25th International Conference on Pattern Recognition (ICPR) | 978-1-7281-8808-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICPR48806.2021.9413231

in low-quality fixed video surveillance devices set at a specific distance from the prospective fire locations [14]. Many of the advanced alarming systems used video processing for fire detection [15]. A study by Shi [16] on fire due to power lines used segmentation techniques.

2) Smoke Detection for Wildfire Detection: The fact that early detection of a wildfire dramatically reduces the damage thus it is important to detect the smoke at the early stage. Insufficient data on smoke has led to the demand for data augmentation, synthetic data generation, or similar methods. Labati et al. [17] segmented smoke plumes using computational intelligence and synthetic data. Dung et al. [18] proposed an automated method for detecting smoke using camera surveillance and image processing procedures. The fundamental purpose of these studies is to perceive and track smoke as objects in motion and to differentiate smoke from non-smoke entities using a convolutional neural network for cascade classifications.

Video surveillance [19] can consider both spatial and temporal domains with a support vector machine-based classifier. Park [20] proposed using spatio-temporal features and classification involving a random forest classifier with the combination of a deep feature map and a saliency map to detect smoke [21]. Unlike traditional methods that deploy color or motion segmentation, Zhou et al. [22] used the maximally stable extremal region for detection. Focus has also been given to the histogram-based smoke segmentation of different color spaces for pixel-level analysis and to segment the smoke [23].

3) Machine Learning-based Wildfire Detection: In previous studies involving machine learning [24], the pixels in a frame of footage of fire were analyzed by integrating motion and Support Vector Machines (SVM) by employing wavelet coefficients. Background subtraction has long been used to evaluate the existence of motion, so Chunyu et al. [7] used it to identify candidate regions with optical flow for motion and back-propagation neural network for classification.

Instead of using just classical convolutional neural networks, Hu and Lu [25] also considers the temporal aspect. Shi [26] used binary patterns, optical flow and convolution neural network to detect smoke. Indicative of the trend towards deep learning, a faster region-based convolutional neural network was used in Kim and Lee [27] to find possible wildfire regions. Li et al. [28] focused on separating the background from the smoke plume and detecting using a back-propagation neural network. Yin et al. [29] proposed an approach that extracts and classifies smoke features using deep normalization and a convolutional neural network made up of 14 layers.

B. Contributions

Considering the available knowledge, the contributions of this paper are the following:

- The introduction of a novel approach that detects wildfire smoke at the pixel level from videos by integrating spatial and temporal features into a semi-supervised deep learning-based video object detection technique.
- A means of mitigating the paucity of data by adopting an online training method that focuses on a specific smoke video and transfers generic features to specific ones.
- An accounting of the physical parameter of haze, which impairs the visibility of smoke in the sky during a wildfire.

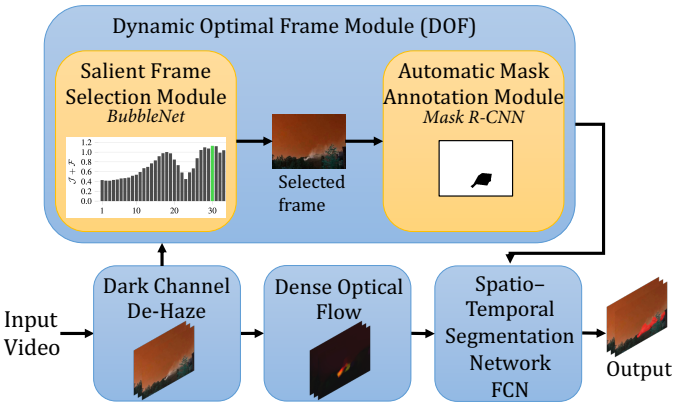

Fig. 1: Overall framework for the Smoke Wildfire Detection.

III. Technical Approach

The proposed work aims at segmenting the smoke plume in wildfire smoke videos as soon as the smoke is visible, which is right after the incipient stage.

The architecture of our approach is divided into four major components: (1) the dark channel de-haze pre-processor, (2) the Dynamic Optimal Frame (DOF) Module, (3) the dense optical flow module, and (4) the spatio-temporal segmentation network. The second component is further divided into the salient frame selection module and automatic mask annotation module. The fourth component integrates spatial and temporal features using the output from (2) and (3).

For smoke detection, sensors, color, and various state-of the-art techniques have been used in the analysis of sets of videos. Smoke is an anomaly that changes its shape in every frame of a video, thus deep learning techniques can be useful to extract features imperceptible to the human eye. Various deep-learning methods for the detection of wildfire smoke have been proposed, but each lacked either the ability to detect smoke in the presence of objects similar to the smoke, the use of temporal information, or the use of physical parameters surrounding the smoke and the camera proximity to the smoke itself. We developed a method that resolves these issues.

A. Dark Channel De-Haze Pre-Processor

Haze is an obscurity occurring due to smoke, dust, or other particles suspended in the atmosphere. Haze shadows smoke, making it inconspicuous. The video dataset in our research is comprised of the videos taken from a camera mounted on a tower; most of these videos include haze with a dense smoke plume. We eliminated the haze from the video frames employing a de-hazing technique (ϕ_D).

To get rid of the haze from the video frames, we adopted a state-of-the-art technique: single image haze removal using dark channel prior [30]. In outdoor statistical information, the regions in an image, excluding the sky, have low-intensity in one of the channels (RGB), resulting in a haze-free image. The air light in a hazy image alters the pixel intensity by compensating for the loss. For this study, we de-hazed the images using a filter on each of the video frames before feeding it into the network. From here on, all referrals to video are to haze-free videos.

The objective of the algorithm used here is to derive a haze-free image. The dark channel [30] for an arbitary image with x as its coordinates can be defined as:

$$J^{dark}(x) = min_{y\epsilon\Omega(x)}(min_{c\epsilon\{r,g,b\}}J^c(y)) \quad (1)$$

where J^c is the intensity for the RGB channel, and $\Omega(x)$ is the local patch. The dark channel is the minimum value for all RGB channels and all the pixels that exist in that patch. Mathematically, the hazy image can be defined as

$$I(x) = J(x)t(x) + A(1 - t(x)) \quad (2)$$

where I, J, and A represent the hazy image, haze-free image, and the atmospheric light, respectively. The $t(x)$ represents the transmission light, which depends on two factors — scene depth and scattering coefficient of the atmosphere.

To extract the re-contructed image we need to compute: dark channel prior, atmospheric light and, transmission map with refinement. We computer the dark channel prior centered at x on every 15 patch in the image. Transmission can be calculated as (1 - local patch) $\times$ the dark channel of normalized haze image. Here, local patch after experimentation is 0.95. For refined transmission map, we use soft mating which is calculated using a Laplacian matrix and a sparse linear system. The average of the pixels that constitute of the top 0.1 percent brightest pixels in the dark channel in the original frame are used to compute the atmospheric light. After computing these values, we use equation 2 to recover the scene radiance.

B. Dynamic Optimal Frame (DOF) Module

Our spatio–temporal segmentation network requires two annotated frames that help the network fine-tune the already-trained model for the specific video. This brings into focus two things: (1) the salient frame selection module (ϕ_S) and (2) the automatic mask annotation tool (ϕ_A).

For the salient frame selection module, we have two options: either we use the first two frames of the video or we specify particular frame numbers for annotating. We select one frame and adopt BubbleNet [31], which introduces the frame selection. There is a substantial difference in the performance between frames, and the appearance of the object varies significantly from frame-to-frame, so for better features, we must select a particular frame.

The deep learning-based bubble-sorting algorithm compares two frames and swaps them according to their performance. This prediction is not solely dependent on the performance: random frame references are used that impact the prediction every time they are altered. The pre-trained ResNet-50 [32] processes input feeds into four fully connected layers with descending neurons and prediction layers. The output gives the relative performance prediction and the frame for the next module to evaluate.

The label loss is defined for the performance label of a frame and includes the Intersection over Union (IOU) and the contour accuracy (C). IOU is defined as the area of intersection over the area of the union of the predicted smoke and its annotated ground truth, whereas closed set of contours of predicted smoke and its annotated ground truth. The loss is expressed as

$$y_i := \frac{1}{n}\sum_{k=1}^{n} IOU_k + C_k \quad (3)$$

After the annotation for the i^{th} frame, the performance on the k^{th} frame is given by $IOU_k + C_k$ for n-frames, and, y_i gives the label for the i^{th} frame.

Automatic Mask Annotation Tool: The spatio–temporal segmentation network requires two annotated frames to predict the segmentation of the whole video. We also need to select a particular frame that improves accuracy, as mentioned above. However, manually annotating a particular frame is a tedious task. We used Region Based Convolutional Neural Network (Mask-RCNN) [33] as our automatic mask annotation tool. Mask R-CNN is an extension to the framework Faster R-CNN and is a benchmark network for segmentation. In addition to the bounding box, Mask R-CNN provides pixel-level segmentation. Rather than by converting it to a vector, the output from each Region of Interest (ROI) provides a mask to preserve the spatial location. Thus, in ROIPool, the inputs from the region proposal network (RPN) were made to fit the feature map, resulting in misalignment. In Mask RCNN, ROIAlign takes the proposal and splits it into bins. The bilinear interpolation is used to find the values for the points sampled in each bin. The RPN aims at finding the objects in the image by scanning over the anchors for detection.

For each sampled region of interest at the time of training, a multi-task loss is given as

$$L = L_{cls} + L_{box} + L_{mask} \quad (4)$$

where L_{cls} is the classification loss and L_{box} is the bounding box loss. L_{mask} gives the average binary cross-entropy loss. The mask branch predicts the mask for every class without competition amongst classes.

We exploit this information by adopting it in our framework to automatically output binary segmented results. To annotate the frames for the segmentation network, we require two frames: we take the frame previously obtained from the salient frame selection module and the frame that comes just before it.

C. Dense Optical Flow Module

Optical flow provides the gradient of the vertical and horizontal axis. To compute the temporal features we used the dense optical flow estimation (ϕ_{DO}) that is beneficial to video object segmentation. The dense optical flow is less computationally expensive compared to other advanced flow techniques. We used dense flow because it provides the flow vectors of all the pixels in a frame. Since our intent is to deliver detection as precise as possible, we disregarded the little computational cost that comes with dense flow.

We were inspired by Farneback [34] on motion estimation techniques of interesting features. Quadratic polynomials are used to approximate the frame windows. To compute the displacement fields, the coefficients yielded by quadratic polynomials are used that after a series of refinement provide the dense optical flow vectors. To generate the flow vector, we input the frames corresponding to the frame numbers given by the annotated image module. This vector represents the motion between the two frames. This is further used as input for the segmentation network.

D. Spatio-Temporal Segmentation Network

To gain familiarity with the internal blocks, we have to first understand the flow of the framework. In this study, the Spatio-Temporal Segmentation Network (ϕ_{ST}) is a fully convolutional network that takes in a total of five inputs: the two frames, their corresponding segmentation annotations, and the corresponding flow images generated by the salient frame selection module, the automatic mask annotation tool, and the dense optical flow, respectively. The output comes out in the form of smoke masks generated for the whole video.

To detect the exact position of the smoke and to visualize its movement, we used a segmentation network to identify localized smoke pixel-by-pixel. While remarkable segmentation techniques exist, segmenting the foreground object in a video is a challenging task. Our work adopts a semi-supervised one-shot video object segmentation work OSVOS [35]. This technique was chosen in part for its simplicity and adaptability.

The next step is to segment the foreground object—the smoke—from the background given the condition that the annotated frames provide. Choosing two video frames instead of one allows us to exploit the temporal information. The CNN architecture is based on VGG [36], a five-stage network whose layers have been modified to extract the pixel level detection. It includes convolutional and Rectified Linear Unit (ReLU) layers with upscaling and downscaling wherever required. In the end, we get a single output by linearly fusing the feature maps with the size of the image, which, in our case, is 320×240. In cases where the salient frame selection module outputs the first and last video frame, we include padding. Focusing on the detection as a whole, bounding boxes are generated with the segmented smoke plumes. We use the cross-entropy loss for the binary classification in the segmentation network, [35], as follows:

$$Loss(Z) = -\sum_{j \epsilon i_p} logP(y_j = 1|X; Z) - \sum_{j \epsilon i_n} logP(y_j = 0|X; Z) \quad (5)$$

With X as the input, Z represents the trainable parameters,y_j represents the binary label, i_p , and i_n are the negative and positive labels, respectively.

One-shot video object segmentation uses offline and online training where the model moves from learning generic object segmentation to fine-tuning to a specific object. We apply it to our advantage in smoke detection by creating a model to learn general features for different smoke plumes, but can fine-tune its learning for a specific smoke video. The advantage of this is that smoke takes different shapes and can often look different in a different video; thus, the fine-tuning gives it an extra edge for detecting smoke.

Algorithm 1 Wildfire smoke detection in video

$WildfireSmokeDetectionModel(video)$:
- $V_d \leftarrow \phi_D(video)$
  - where, V_d = Dehazed video
- $I_t \leftarrow \phi_S(V_d)$
  - where, I_t = Optimal frame image
- $I_{t-1_{mask}}, I_{t_{mask}} \leftarrow \phi_A(I_{t-1}, I_t)$
  - where, $I_{t_{mask}}$ = Mask of frame I_t
- $IF_t \leftarrow \phi_{DO}(I_{t-1}, I_t)$
  - where, IF_t = Flow vectors of I_{t-1} and I_t

return $\phi_{ST}(I_{t-1}, I_t, I_{t-1_{mask}}, I_{t_{mask}}, IF_t)$

The inference path of the proposed method is shown in Algorithm 1 which takes a video as an input and outputs the smoke segmentation in the video frames. Each frame is de-hazed then ranked by salient frame selection module, and the mask of the selected frame is generated by the automatic mask annotation tool. This mask is used for online training the segmentation network. Finally, the spatio-temporal segmentation network returns the segmentation of the smoke.

IV. Experimental Setup and Results

A. Dataset

Fig. 2: Sample frames of the video dataset.

We required two sets of data: the smoke video data is used for training and testing the segmentation network and the

smoke image dataset is required to train the automatic mask annotation module, as explained in Section III. Numerous public datasets are available, some of which include synthetic smoke created inside buildings. But to provide results that approximate reality, we used mostly smoke video datasets from [37], [38], [39], [40]. These datasets were chosen because they all have a forest background and, the smoke varies from video to video in terms of density and shape. The dataset also includes videos at various distances, which helps create a solution that assists in the detection of various sizes of smokes. Sample frames of smoke video dataset is shown in Fig. 2. We also annotated a few videos using online annotation tools. Each video frame was 320×240. We divided the dataset so that it consisted of 1050 frames for training, and, to improve the results, we used scale and flip-data augmentation. This brought the total to 3500 frames. Each video has 150 frames. To determine the train–test division, we evaluated the size of wildfire smoke in order to include small and large smoke plumes in both the training and testing dataset.

TABLE I: Different ratio/sizes of smoke for each video. Ratio is computed by pixel area of the smoke over the area of the image; size is the dimension of the bounding box of the smoke segmentation mask.

Videos	Ratio of smoke	Size of smoke
Train video 1	0.116	104×86
Train video 2	0.035	73×37
Train video 3	0.209	102×158
Train video 4	0.193	146×102
Train video 5	0.039	38×79
Train video 6	0.081	69×91
Validation video 1	0.038	39×75
Test video 1	0.101	54×144
Test video 2	0.261	119×168
Test video 3	0.031	46×53
Test video 4	0.244	280×67
Test video 5	0.023	52×34

B. Evaluation

To quantitatively measure the performance of our framework, we evaluated the following metrics: True Positives (TP), True Negatives (TN), False negatives (FN), and False Positives (FP), IOU previously defined in Section III, and the average IOU over all test frames (Eqn 6), Pixel Accuracy (Acc) measuring the correctly classified pixels in an image (Eqn. 7), precision, recall, and F1 metrics. The metrics from the testing videos were averaged. It is important to remember that we perform per-pixel prediction of the smoke. This evaluation shows the variation of each pixel from the ground truth.

$$IOU = \frac{1}{h \times w \times m \times k} \sum_{l=1}^{m} \sum_{i=1}^{k} \frac{target \cap prediction}{target \cup prediction} \quad (6)$$

$$Acc = \frac{1}{h \times w \times m \times k} \sum_{l=1}^{m} \sum_{i=1}^{k} \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

There are *k* frames in the *m* videos. The *h* and *w* represent the height and width of the frame. To evaluate the precision, recall, and F1 score, we determined the closeness of the boundary of the foreground object and its ground truth.

TABLE II: 7-fold cross validation.

Different Models	Precision	Recall	Pixel Accuracy
Group 1	0.869	0.949	0.990
Group 2	0.996	0.876	0.986
Group 3	0.575	0.934	0.933
Group 4	0.962	0.933	0.994
Group 5	0.930	0.973	0.995
Group 6	0.630	0.750	0.970
Group 7	0.690	0.830	0.910

To evaluate if our proposed framework can consistently and accurately capture various smoke plumes across multiple frames, we used 7-fold cross-validation. It tests the effectiveness of our deep learning framework and makes it less biased, as all the smoke video datasets can appear in the test and train sets. The groups in Table II represent combinations of test videos. Group 2 had high precision, whereas group 5 showed high recall. For the pixel accuracy, almost all the groups gave good results. In our case, detection of the smoke plume is important, thus, recall is a better measurement than precision, which does not account for the false negatives.

C. Experimental Setup

For the segmentation network, we used pre-trained VGG-16, trainable end-to-end, on ImageNet [41] which is considered good for initialization. The parent network was a fully convolutional network trained using our dataset. We used a learning rate of 10^{-4} and trained for 15000 iterations. We experimented with different threshold values to obtain a precision-recall curve, as shown in Fig. 3 and chose 0.65 as the threshold. We took the batch size 6 on the NVIDIA GTX 1080Ti GPUs. For the online training, we used one video to train for 500 iterations, which takes approximately 55 seconds to process. We could have used more iterations to improve on the accuracy, but there is a trade-off between accuracy and time.

For the salient frame selection module, we used a pre-trained ResNet-50 on ImageNet. After experimenting with a different number of reference frames, three reference frames were used in addition to two comparison frames. To train our salient frame selection module, we chose a small batch of six. The automatic mask annotation tool exploits pre-trained weights from the COCO [42] dataset, which provides a broad-spectrum of features. For training, we chose batch size 1. In order to prevent it from over-fitting, we chose 45 epochs. To increase our dataset, we used data augmentation that alters

TABLE III: The quantitative metrics (mean ± standard deviation) defined for videos from the smoke video test dataset.

Video	True Positives	False Positives	IOU	Pixel accuracy	Precision	Recall	f1 score
video 1	118 ± 0.128	4 ± 0.257	0.974 ± 0.015	0.973 ± 0.014	0.872 ± 0.045	0.831 ± 0.044	0.850 ± 0.036
video 2	100 ± 0	0 ± 0	0.991 ± 0.004	0.991 ± 0.004	0.913 ± 0.034	0.908 ± 0.040	0.910 ± 0.032
video 3	142 ± 0.302	21 ± 0.591	0.942 ± 0.023	0.945 ± 0.025	0.743 ± 0.051	0.765 ± 0.045	0.753 ± 0.039
video 4	131 ± 0.496	25 ± 0.777	0.867 ± 0.025	0.870 ± 0.023	0.698 ± 0.034	0.706 ± 0.046	0.701 ± 0.024
video 5	114 ± 0.093	2 ± 0.186	0.990 ± 0.004	0.990 ± 0.004	0.869 ± 0.043	0.949 ± 0.018	0.907 ± 0.027

the input image according to our requirements by using affine transformations and adding Gaussian noise to alter the image. We used a 0.5 probability of flipping the images horizontally. The IOU threshold used was 0.5 to decrease the false positives. Initially, we chose the top 90 anchors and then amended it to 50 after refinement.

Fig. 3: Precision-recall curve

We tested our model on five different videos with various smoke plume sizes. Each video took 5.016 seconds on average, and each frame took approximately 4.16 milliseconds.

D. Results and Discussions

TABLE IV: Comparison of metrics among different split of the datasets.

Dataset	Acc (mean ± SD)	ratio$_{FP}$	ratio$_{FN}$
Training	0.981 ± 0.017	0.014	0.014
Validation	0.968 ± 0.031	0.017	0.019
Testing	0.953 ± 0.050	0.020	0.021

TABLE V: Comparisons of results over different segmentation frameworks.

Model	Mean IOU	Mean F1 score
Our Model	**0.95**	**0.82**
OSVOS [35]	0.88	0.74
Unet [6]	0.55	0.73
Fpn [43]	0.79	0.65
LinkNet [44]	0.39	0.55

We expected our proposed approach to work such that the false positives and false negatives are reduced. Moreover, detection at the pixel-level increases accuracy, which will help obtain the exact location of a wildfire. Table III shows the quantitative results over various metrics using our proposed method on our test set. In Table III, the metrics represent the average and standard deviation (SD) values for each video. These values are pixel-level except for true positives and false positives; i.e. for 118 true positives in a video, one corresponds to the average number of frames with smoke correctly detected in a video. The intersection over union gives good results. The results for the metrics may be improved, as our model works well with the regions in the smoke that are dense. However, for some videos where we manually labeled the ground-truth, we chose to include the faintest smoke to improve detection. Table IV represents the mean and standard deviation of pixel accuracy; the training value is more compared to the test and

Fig. 4: Visualizations of de-hazed and segmentation results of consecutive frames from 3 videos respectively. For each video, the first row shows the original frames; the second row is the de-hazed frame after pre-processing; the third row is the segmentation result (smoke regions are marked in red) obtained from the proposed method. Best viewed in color.

TABLE VI: Analysis of the effects of different modules on evaluation results. The modules used for each experiment are marked with ✔: Backbone - ResNet-101/VGG-16, De-haze - dark channel prior pre-processor, Temp - optical dense flow, and DOF - dynamic optimal frame module. All of the combinations include the FCN segmentation network. All metrics are the higher the better except FP.

		Add-on Modules			Experimental Results						
Model	Backbone	De-haze	Temp	DOF	IOU	Acc	TP	FP↓	Precision	Recall	F1
A	ResNet-101		✔	✔	86.8	89.2	97	45	68.5	79.40	71.74
B	ResNet-101	✔	✔	✔	94.6	92.7	104	31	70.3	81.60	75.27
C	VGG-16				88.0	93.0	46	23	74.8	63.16	74.90
D	VGG-16		✔	✔	95.1	94.0	108	29	71.7	82.14	76.34
E	VGG-16	✔	✔		95.0	94.0	109	27	71.7	82.14	76.30
Full	VGG-16	✔	✔	✔	**95.8**	**95.6**	**118**	**7**	**75.4**	**84.10**	**79.50**

validation, as expected.

Each module contributes to the improvement of the results. Our proposed work comprises of the 3 modules in addition to the spatio-temporal segmentation network where the dense optical flow provides the temporal information and improves the detection for the whole video. The Dark Channel De-Haze Pre-Processor contributes to the initial pre-processing to remove haze and emphasise smoke which helps in reducing the false positives. The Dynamic Optimal Frame (DOF) Module improves the proposed work as a whole and provides automatic annotation which cuts down the manual annotation work. A detailed study is shown in Section IV-E.

Figure 4 shows our model is able to precisely detect near or distant smoke plumes. We also compared our results with state-of-the-art segmentation frameworks that were trained with and evaluated using our smoke dataset. As seen from Table V, the state-of-the-art technique did not perform well. One of the reasons for this is that these techniques are not able to detect smoke when there is haze because they do not use any haze removal techniques. Another reason is that smoke is an anomaly and to detect it in a video, requires a network to understand the underlying features in that particular video which they fail to extract. Our proposed method outputs segmented smoke for the whole video which makes it difficult to compare with various smoke detection methods that aim at frame level detection rather than pixel level detection. Thus we used the various segmentation techniques for comparison.

Like any other approach based on assumptions, our proposed method has its limitations. Detecting wildfire smoke at the pixel level works well for most wildfire videos, but it may give confounding results in some extreme cases, such as when the smoke is adjacent to the sky and becomes indistinguishable from the background. Such cases, however, do not reflect early detection; early detection requires detecting the origin of the smoke from in between trees in a forest.

E. Ablation Study

Here, we show our analysis of different measures justifying the proposed work. To inspect each combination, we expect all the metrics to be high except the false positives. The incurred false positives reflect that the proposed framework alerted for false wildfire smoke. This is a waste of resources. Table VI gives a detailed analysis of the performances of different combinations of modules. The models A, B, C, D, E, and Full are various combinations where model C is OSVOS [35] and model Full is our proposed work.

We observed that the models C, D, E and Full are better as they use VGG as a backbone for the segmentation network rather than ResNet-101. When comparing the model B and model Full which constitute of the same modules, we observe the backbone plays an important role as the results improve with VGG-16. The number of false positives increased when the de-haze pre-processing was not included as seen in model A. Thus, for improved image quality, de-hazing works well. The experiments also evaluated the addition of the automatic mask annotation tool for our proposed work. Although the performances of the combinations without it are reasonable, there was a major decrease in false positives when it was used in the correct combination. Furthermore, in model C, eliminating the temporal features decreased the number of true positives, which defeats the sole purpose of detecting wildfire smoke. Finally, the results show that the model Full that is our proposed work outperforms the other combinations.

V. Conclusions

The proposed method aims to detect wildfires in their incipient stage. We proposed a novel method for pixel-level detection of wildfire smoke in a video by integrating spatial and temporal features across multiple frames and taking into account the physical parameter of haze surrounding the smoke. De-haze pre-processing improved the results, as the removal of haze makes the smoke more visible. Moreover, online training improved the results by helping to mitigate the paucity of the smoke data. We validated our proposed work on several public datasets pertaining to different forest backgrounds and smoke with different proximity from the camera using various metrics. The proposed work is feasible and can be easily switched to live smoke detection in video. Additionally, details on the exact location of the smoke can be helpful for extracting various smoke properties, such as the strengths of the wind and its direction that can further aid fire fighters.

Acknowledgments

This material is based on work supported in part by Bourns Endowment funds.

REFERENCES

[1] *Year-to-date statistics*, 2009 (accessed March 8, 2020). [Online]. Available: https://www.nifc.gov/fireInfo/nfn.htm

[2] O. Gunay, B. U. Toreyin, K. Kose, and A. E. Cetin, "Entropy-functional-based online adaptive decision fusion framework with application to wildfire detection in video," *IEEE Transactions on Image Processing*, vol. 21, no. 5, pp. 2853–2865, 2012.

[3] P. Ramanatha, U. R. Nelakuditi, S. Ravishankar, and V. Ranganathan, "Uav based smoke plume detection system controlled via the short message service through the gsm network," in *2016 International Conference on Inventive Computation Technologies (ICICT)*, vol. 2. IEEE, 2016, pp. 1–4.

[4] F. Yuan, "A double mapping framework for extraction of shape-invariant features based on multi-scale partitions with adaboost for video smoke detection," *Pattern Recognition*, vol. 45, no. 12, pp. 4326–4336, 2012.

[5] T. Çelik, H. Özkaramanlı, and H. Demirel, "Fire and smoke detection without sensors: Image processing based approach," in *2007 15th European Signal Processing Conference*. IEEE, 2007, pp. 1794–1798.

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[7] Y. Chunyu, F. Jun, W. Jinjun, and Z. Yongming, "Video fire smoke detection using motion and color features," *Fire technology*, vol. 46, no. 3, pp. 651–663, 2010.

[8] B. Ko, J. Park, and J.-Y. Nam, "Spatiotemporal bag-of-features for early wildfire smoke detection," *Image and Vision Computing*, vol. 31, no. 10, pp. 786–795, 2013.

[9] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[10] T. Celik and H. Demirel, "Fire detection in video sequences using a generic color model," *Fire safety journal*, vol. 44, no. 2, pp. 147–158, 2009.

[11] S. D. Thepade, J. H. Dewan, D. Pritam, and R. Chaturvedi, "Fire detection system using color and flickering behaviour of fire with kekre's luv color space," in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*. IEEE, 2018, pp. 1–6.

[12] T.-H. Chen, P.-H. Wu, and Y.-C. Chiou, "An early fire-detection method based on image processing," in *2004 International Conference on Image Processing, 2004. ICIP'04.*, vol. 3. IEEE, 2004, pp. 1707–1710.

[13] X.-F. Han, J. S. Jin, M.-J. Wang, W. Jiang, L. Gao, and L.-P. Xiao, "Video fire detection based on gaussian mixture model and multi-color features," *Signal, Image and Video Processing*, vol. 11, no. 8, pp. 1419–1425, 2017.

[14] J. Gubbi, S. Marusic, and M. Palaniswami, "Smoke detection in video using wavelets and support vector machines," *Fire Safety Journal*, vol. 44, no. 8, pp. 1110–1115, 2009.

[15] T.-H. Chen, Y.-H. Yin, S.-F. Huang, and Y.-T. Ye, "The smoke detection for early fire-alarming system base on video processing," in *2006 International Conference on Intelligent Information Hiding and Multimedia*. IEEE, 2006, pp. 427–430.

[16] J. Shi, W. Wang, Y. Gao, and N. Yu, "Detection and segmentation of power line fires in videos," in *2020 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. IEEE, 2020, pp. 1–5.

[17] R. D. Labati, A. Genovese, V. Piuri, and F. Scotti, "Wildfire smoke detection using computational intelligence techniques enhanced with synthetic smoke plume generation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 4, pp. 1003–1012, 2013.

[18] N. M. Dung, D. Kim, and S. Ro, "A video smoke detection algorithm based on cascade classification and deep learning." *KSII Transactions on Internet & Information Systems*, vol. 12, no. 12, 2018.

[19] C.-Y. Lee, C.-T. Lin, and C.-T. Hong, "Spatio-temporal analysis in smoke detection," in *2009 IEEE International conference on signal and Image processing applications*. IEEE, 2009, pp. 80–83.

[20] J. Park, B. Ko, J.-Y. Nam, and S. Kwak, "Wildfire smoke detection using spatiotemporal bag-of-features of smoke," in *2013 IEEE Workshop on Applications of Computer Vision (WACV)*. IEEE, 2013, pp. 200–205.

[21] G. Xu, Y. Zhang, Q. Zhang, G. Lin, Z. Wang, Y. Jia, and J. Wang, "Video smoke detection based on deep saliency network," *Fire Safety Journal*, vol. 105, pp. 277–285, 2019.

[22] Z. Zhou, Y. Shi, Z. Gao, and S. Li, "Wildfire smoke detection based on local extremal region segmentation and surveillance," *Fire Safety Journal*, vol. 85, pp. 50–58, 2016.

[23] D. Krstinić, D. Stipaničev, and T. Jakovčević, "Histogram-based smoke segmentation in forest fire detection system," *Information technology and control*, vol. 38, no. 3, 2009.

[24] B. C. Ko, K.-H. Cheong, and J.-Y. Nam, "Fire detection based on vision sensor and support vector machines," *Fire Safety Journal*, vol. 44, no. 3, pp. 322–329, 2009.

[25] Y. Hu and X. Lu, "Real-time video fire smoke detection by utilizing spatial-temporal convnet features," *Multimedia Tools and Applications*, vol. 77, no. 22, pp. 29 283–29 301, 2018.

[26] J. Shi, W. Wang, Y. Gao, and N. Yu, "Optimal placement and intelligent smoke detection algorithm for wildfire-monitoring cameras," *IEEE Access*, vol. 8, pp. 72 326–72 339, 2020.

[27] B. Kim and J. Lee, "A video-based fire detection using deep learning models," *Applied Sciences*, vol. 9, no. 14, p. 2862, 2019.

[28] X. Li, W. Song, L. Lian, and X. Wei, "Forest fire smoke detection using back-propagation neural network based on modis data," *Remote Sensing*, vol. 7, no. 4, pp. 4473–4498, 2015.

[29] Z. Yin, B. Wan, F. Yuan, X. Xia, and J. Shi, "A deep normalization and convolutional neural network for image smoke detection," *Ieee Access*, vol. 5, pp. 18 429–18 438, 2017.

[30] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 12, pp. 2341–2353, 2010.

[31] B. A. Griffin and J. J. Corso, "Bubblenets: Learning to select the guidance frame in video object segmentation by deep sorting frames," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8914–8923.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[33] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[34] G. Farneback, "Fast and accurate motion estimation using orientation tensors and parametric motion models," in *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, vol. 1. IEEE, 2000, pp. 135–139.

[35] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool, "One-shot video object segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 221–230.

[36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Pattern Recognition*, 2015.

[37] *Wildfire Observers and Smoke Recognition*, 2010 (accessed March 8, 2020). [Online]. Available: http://wildfire.fesb.hr

[38] *Computer Vision Based Fire Detection Software*, (accessed March 8, 2020). [Online]. Available: http://signal.ee.bilkent.edu.tr/VisiFire

[39] Soojung, *Fire-Detection-Model-Keras*, (accessed March 8, 2020). [Online]. Available: http://www.kaggle.com/csjcsj7477/firedetectionmodelkeras

[40] *CVPR Laboratory, Keimyung University, Korea*, (accessed March 8, 2020). [Online]. Available: https://cvpr.kmu.ac.kr

[41] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[42] 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 *European conference on computer vision*. Springer, 2014, pp. 740–755.

[43] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[44] A. Chaurasia and E. Culurciello, "Linknet: Exploiting encoder representations for efficient semantic segmentation," in *2017 IEEE Visual Communications and Image Processing (VCIP)*. IEEE, 2017, pp. 1–4.