EasyChair Preprint

№ 2581

Early Wildfire Detection Using Convolutional Neural Network

Seon Ho Oh, Sang Won Ghyme, Soon Ki Jung and Geon-Woo Kim

February 5, 2020

Early Wildfire Detection Using Convolutional Neural Network*

Seon Ho Oh[1][0000−0002−2927−7850], Sang Won Ghyme[1], Soon Ki Jung[2][0000−0003−0239−6785], and Geon-Woo Kim[1]

[1] Electronics and Telecommunications Research Institute, Daejon, Republic of Korea
{seonho, ghyme, kimgw}@etri.re.kr
[2] School of Computer Science and Engineering, Kyungpook National University, Daegu, Republic of Korea
skjung@knu.ac.kr

Abstract. Wildfires are one of the disasters that are difficult to detect early and cause significant damage to human life, ecological systems, and infrastructure. There have been several research attempts to detect wildfires based on convolutional neural networks (CNNs) in video surveillance systems. However, most of these methods only focus on flame detection, thus they are still not sufficient to prevent loss of life and reduce economic and material damage. To tackle this issue, we present a deep learning-based method for detecting wildfires at an early stage by identifying flames and smokes at once. To realize the proposed idea, a large dataset for wildfire is acquired from the web. A light-weight yet powerful architecture is adopted to balance efficiency and accuracy. And focal loss is utilized to deal with the imbalance issue between classes. Experimental results demonstrate the effectiveness of the proposed method and validate its suitability for early wildfire detection in a video surveillance system.

Keywords: Early wildfire detection · video surveillance · deep learning.

1 Introduction

Wildfire is a global problem causing devastating damage every year [24]. According to the National Interagency Fire Center (NIFC), 46,706 wildfires occurred between January 1 and November 22, 2019, burning about 4.6 million acres [8]. However, existing fire detection studies focused on flame detection can only work after the fire has spread over a large area. And it makes the control of the fire difficult or sometimes impossible to stop in time. As a result, wildfires can cause catastrophic damage to the atmosphere and the environment. Another problem caused by wildfires is a long-term disaster, such as impacts on local weather

* This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by Korea government (MSIT) (No. 2019-0-00203, Development of Predictive Visual Security Technology for Preemptive Threat Response)

patterns, global warming, and extinction of rare species of the flora and fauna. Therefore, developing an effective method to detect wildfire at an early stage is very important.

Most early studies attempted to explorer color, texture, shape, and motion features for fire detection. For instance, Chen et al. [1] proposed a decision rule-assisted fire detection method that examines the dynamic behavior and irregularity of flame in both RGB and HSI color spaces. Later works considered machine learning-based classification approaches such as support vector machine (SVM) [11] or neural networks [2, 26]. Chenebert et al. [2] presented a shallow neural network classifier using the color-texture feature. Recently, Foggia et al. [4] introduced a multi-expert framework that combines shape, color, and motion properties.

Thanks to the recent advances of deep learning, further developments based on the CNNs now perform more robustly compared to earlier works. For instance, Sharma et al [19] used well-known CNN architectures such as VGG16 [20] and ResNet [6]. And Muhammad et al. [14] adopted other architectures like AlexNet [12] and Inception [21] architectures to detect flame more efficiently and robustly. There was also an attempts to design dedicated architectures for fire detection. Namozov and Cho [15] presented a VGG16 based novel deep convolutional neural network, and an effective training strategy on a limited number of images by increasing the number of training images using a Generative Adversarial Networks (GAN) [5] and data augmentation techniques. Meanwhile, Jadon et al. [9] proposed a light-weight neural network that ensures real-time inference on low-powered devices such as Raspberry Pi.

In this work, we present a method for detecting wildfires at the early stage using deep CNNs. Due to the lack of data that meets the scale and diversity, we collect a large dataset including smokes, flames, etc. Also, we use a light-weight yet powerful architecture to balances efficiency and accuracy. And, to overcome the class imbalance issues, we also use Focal loss [13]. Finally, our experiments demonstrate that the effectiveness of the proposed method and suitability for early wildfire detection in a video surveillance system.

The rest of the paper is organized as follows. The next section describes the collection of dataset for detecting wildfires at an early stage. In Section 3, we present our method for early wildfire detection in surveillance video. Experimental results and discussion are given in Section 4. Conclusion and future work are given in Section 5.

2 Dataset Collection

Since the dataset previously used for fire detection only considers flame or smoke, there exists a potential to cause many errors in real-world video surveillance environments. Another problem is that the scale and diversity of the datasets previously used for wildfire detection are limited, which is insufficient to train deep CNNs. To address this issue, we collect a large set of data containing initial smoke or flame on the rural areas for early wildfire detection. Also, to take

complexities and ambiguities on real-world video surveillance environment into consideration when collecting data for wildfire, we include not only smokes and flames but also subjects having visual similarities such as clouds, fogs, snows, waves, waterfalls, etc. More specifically, in order to detect wildfire in an early stage, we first separated flame or smoke-like things having particles and others like solid, however, there are a variety of things like clouds, fog, waterfalls, waves, snow, and flock of birds, which has a flame or smoke-like feature, so it was a necessity to distinguish between them. Moreover, since fires can occur not only natural objects such as mountains adjacent to the coast, lakesides but also man-made structures such as ski slope, town border (rural), all of these places should be considered. Therefore, we summarized the dataset into three groups: solid (rural), smoke or flame, and smoke-like objects having particles. Then we categorized the dataset into six classes such as cloud, snow, rural, fire, wave, and waterfall. Further details to determine the dataset categories are discussed later.

Rest of this section, we present a multi-stage strategy to collect a large early wildfire detection dataset including: how candidate images were collected; and how the dataset was cleaned up both automatically and manually. Table 1 summarizes each stages and corresponding statistics. Individual stages are discussed in detail in the following paragraphs.

Table 1. Dataset statistics after each stage of processing.

Stage	Description	# images
1	Automatic image crawling	152,996
2	Automatic and manual cleanup	16,410
3	Final patch and class labeling	14,741

2.1 Automatic Image Crawling

Images could be easily collected from an image search site. An image crawling tool helps to collect images automatically from such sites. Here the image crawling tools were very useful, and most tools supported common image search engines such as Google, Bing, and Baidu. With such a tool, all we have to do is finding proper keywords. An image crawling tool collects only several hundred images for each keyword. Therefore a lot of keywords are required. For instance, keywords such as bushfire, forest fire, wildfire, etc. were selected for the 'fire' class. To obtain more images, we used keywords from various languages, including English, Korean, Chinese, and Japanese. A total of 101 keywords were selected for all classes and finally 152,996 images were collected.

2.2 Automatic and Manual Cleanup

Although a large number of images were collected at the first stage, most of the images were not related to our problem. For example, most images from the

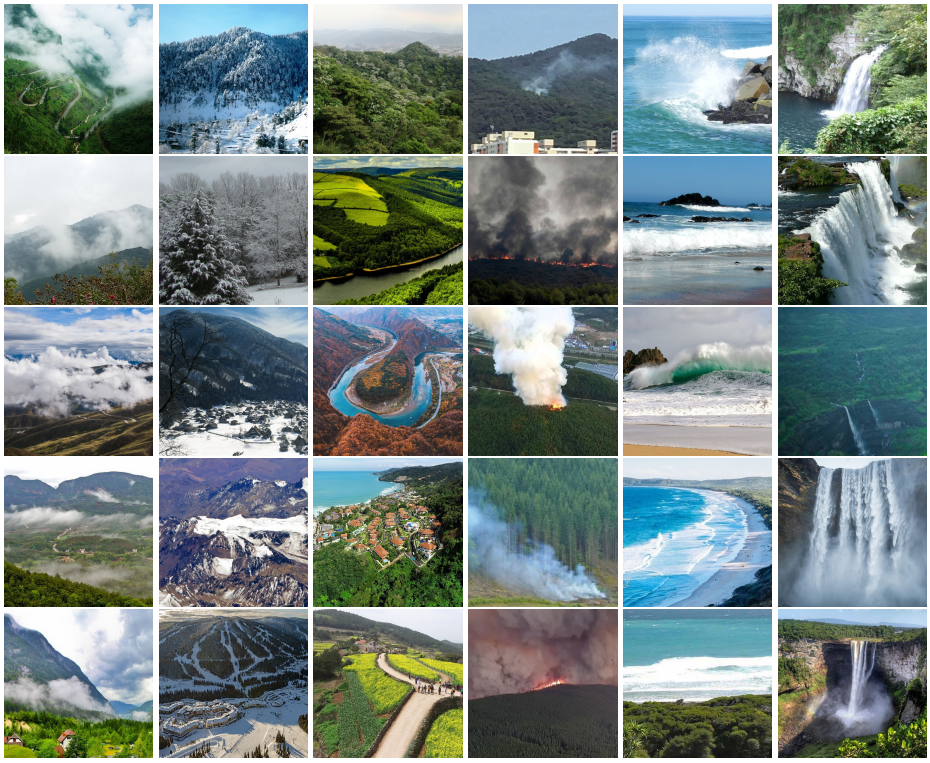

Fig. 1. Examples of images for test set. From the left, each column shows images corresponding to cloud, snow, rural, fire, wave, and waterfall.

keyword 'wave' were related to hair-style. Therefore it is required for removing irrelevant images manually. After removal, only 18,929 images were left alive.

Another problem was exact or near duplication between images. It has been observed that exact duplication due to the same images being found at different locations, or near duplications caused by geometric transformations such as mirror, rotation, cropping, or color adjustments and JPEG artifacts. Most image duplicates can be removed automatically with a duplicate removing tool, and finally, only 16,410 images were survived.

At this point, two types of issues may still remain; first, some images are too small to use; and second, some classes have an insufficient number of images compared to others. Thus, we filtered out all images having a smaller resolution than 300×300 pixels. And classes with fewer than 200 images were discarded. In addition, some classes like 'fog' that are difficult to distinguish from others were removed. As a result, a total of 13,051 images of six classes remained.

2.3 Final Patch and Class Labeling

In this stage, part of the image was manually labeled with a specific class. To this end, we used the image labeling tool. One image can have one or more patches, and patches can be mapped into different classes even if they are obtained from the same image. All patches cropped from images are resized with a regular size of 512×512 pixels. Finally, our dataset consists of a total of 14,504 patches in six classes including cloud, snow, rural, fire, wave, and waterfall.

Table 2 summarizes the number of images in each class. Figure 1 shows examples of images for each class of clouds, snows, rural, fires, waves, waterfalls. The 'rural' class contains images of mountains, seas, lakes, rivers, etc. And the 'fire' class consists of images of the flame and smoke in rural areas.

Table 2. Dataset statistics summary.

Class	Cloud	Snow	Rural	Fire	Wave	Waterfall	Total
# images	2,503	605	6,832	1,901	1,888	1,012	14,741

3 Proposed Framework

In this section, we will describe the selection of network architecture suitable for a video surveillance systems, and present our training strategy in detail.

3.1 Network Architecture Selection

Current state of the art CNN models, such as AlexNet [12], VGG16 [20], Inception [21], and ResNet [6] have been adjusted and demonstrated promising performance in numerous computer vision issues and applications, such as object detection, image segmentation, super-resolution, and classification. However, these deep CNN models are not suitable for real-time wildfire detection in a video surveillance system due to their computational costs and resource requirements.

Recently, Tan and Le introduced EfficientNet [23]. They systematically studied the impact of scaling different dimensions of the model and presented a family of models called EfficientNet which improve overall performance while balancing all dimensions of the network – width, depth, and image resolution. Surprisingly, these models offer significant gains for speed and resource efficiency. Especially, their baseline model, EfficientNet-B0, allows real-time inference on a low-powered device such as Raspberry Pi. Thus we adopt EfficientNet-B0 as our baseline network architecture. Table 3 illustrates EfficientNet-B0 as our baseline network structure. The mobile inverted bottleneck convolution called MBConv [17, 22] is the main building block for EfficientNet. In EfficientNet's MBConv, direct shortcuts between the bottlenecks that connect much fewer channels compared to the expansion layer of the residual block [6] and combined depthwise separable convolution effectively reduce computation by a factor of spatial resolution, and squeeze-and-excitation optimization [7] was also added.

Table 3. Network architecture: EfficientNet-B0 [23].

Stage	Operator	Input resolution	Output channels	# Layers
1	Conv(3x3)	224×224	32	1
2	MBConv(3x3)	112×112	16	1
3	MBConv(3x3)	112×112	24	2
4	MBConv(5x5)	56×56	40	2
5	MBConv(3x3)	28×28	80	3
6	MBConv(5x5)	14×14	112	3
7	MBConv(5x5)	14×14	192	4
8	MBConv(3x3)	7×7	320	1
9	Conv(1x1) & Pooling & FC	7×7	1280	1

3.2 Class Imbalance

The fire detection dataset has an inherent imbalance due to its rarity. Such class imbalance causes two problems: Training is inefficient as most samples are non-fire that contribute no useful learning signal; The non-fire samples can overwhelm training and lead to degenerate models.

To address the inherent class imbalance issue, we use Focal loss [13]. The focal loss is designed to resolve the class imbalance problem by weighting the contribution of easy example smaller even if their number is large. As a result, it focuses on training a sparse set of hard examples.

The original focal loss for binary classification defined as:

$$\mathrm{FL}(p_t) = -(1-p_t)^{\gamma}\log(p_t), \tag{1}$$

where

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise,} \end{cases} \tag{2}$$

and $p \in [0, 1]$ is model's estimated probability for the class with label $y = 1$. Based on this, we can rewrite the multi-class form as:

$$\mathrm{FL}(p_t) = -\sum_{i=1}^{C}(1-p_t(i))^{\gamma}\log(p_t(i)), \tag{3}$$

and

$$p_t(i) = \begin{cases} p & \text{if } y = i \\ 1-p & \text{otherwise,} \end{cases} \tag{4}$$

where C is the number of classes. The focusing parameter γ was set to 2.0 in our experiment.

3.3 Training

We used a $7:1:2$ split between the training, validation and test sets. And stratified sampling was used to partitioning the dataset so that each set had evenly

balanced classes. In order to increase the diversity in our train dataset, we used image augmentations such as random cropping, resizing, and flipping. Random cropping produces a patch having a random size $\in [0.08, 1.0]$ and random aspect ratio $\in [3/4, 4/3]$ of the original image. This patch is resized to 224×224 and finally flipped horizontally at random.

We train and fine-tuned ImageNet [3] pre-trained neural network model using a transfer learning strategy. Instead of fine-tuning only the last layer of the neural network, we trained all the layers using different learning rates. The learning rate for pre-trained convolutional layers was set to a factor of 0.1 than of the last dense layers. For regularization, we used dropout between convolutional layers and dense layers with probability of 0.2.

Our model was trained using the Adam optimizer [10] with β_1 0.9 and β_2 0.99; weight decay 1e-4; initial learning rate 0.001. The learning rate was decayed by a factor of 0.1 every 30 epochs. The batch size and total epochs were set to 256 and 90, respectively.

3.4 Implementation Details

The proposed method was implemented with PyTorch [16]. And the neural network trained and evaluated on a workstation equipped with Intel Core i7-6950X 3.0 GHz CPU, 128 GB memory, and two NVIDIA GeForce GTX 1080 GPUs.

4 Experimental Results and Discussion

In this section, we will first evaluate our model on multi-class classification and binary wildfire detection. Additionally, we also describe how to extend patch classification to frame-level detection.

4.1 Multi-class Classification

The primary multi-class classification accuracies on the validation and test sets are 99.46% and 98.58%, respectively. Figure 2 draws the normalized confusion matrix for each class. As can be seen in Figure 2, the model trained achieved higher and balanced accuracy for individual classes. This means that focal loss effectively alleviated class imbalance by increasing loss contribution from hard examples.

4.2 Binary Projection

For the binary wildfire detection, we projected multi-class classification into binary classification, either fire or non-fire. The validation set is composed of 190 fire images and 1,284 non-fire images, so a total of 1,474 images. And the test set consists of 381 positive images and 2,570 negative images, so 2,951 images in total.

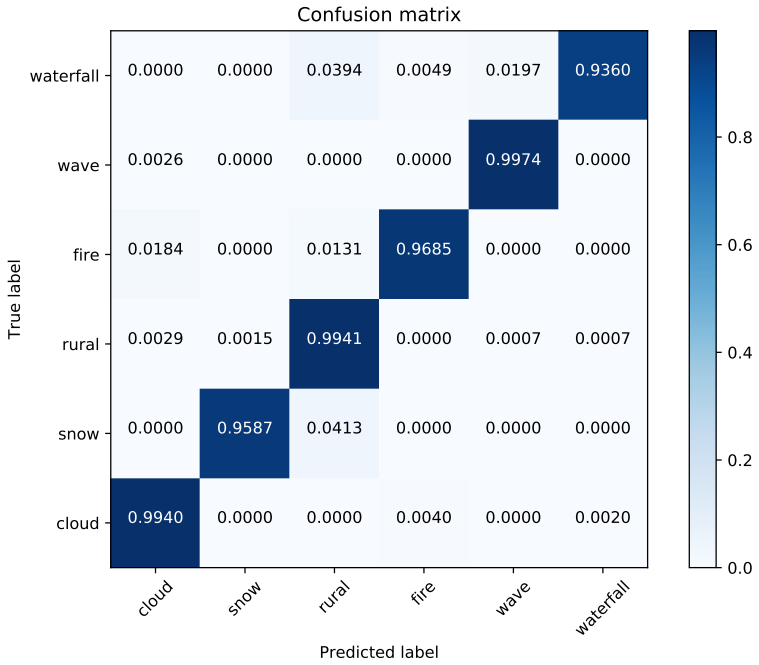

Fig. 2. Confusion matrix on test set.

To determine either fire or non-fire, a cut-off threshold obtained from receiver operating characteristic (ROC) analysis in the validation set is adopted. The details of finding the optimum cut-off point will be discussed next subsection.

Table 4 shows evaluation statistics for validation and test sets. We have used six different metrics (accuracy, precision, recall, F_1-score, FPR, and FNR) in order to present a complete and reliable analysis. The FPR and FNR denote false positive rate and false negative rate, respectively. As can be seen in Table 4, the results obtained for each metric is encouraging. It is also noted that the result gives balanced FPR and FNR since the cut-off threshold adopted from the Youden's index.

To investigate the effectiveness of the binary projection, we also compared binary fire vs non-fire classifier. The 'binary' and 'multi-proj.' represent simple binary classification and multi-class classification with binary projection, respectively. As shown in the test set metrics, although binary classification gives better validation set performance, however, our binary projection following the multi-class classification provided better generalization performance.

Table 4. Evaluation statistics on validation and test sets(%)

Method	Split	Accuracy	Precision	Recall	F_1-score	FPR	FNR
binary	Val.	99.53	96.92	99.47	98.18	0.47	0.53
	Test	98.78	93.89	**96.85**	95.35	0.93	**3.15**
mult-proj.	Val.	99.32	95.45	99.47	97.42	0.70	0.53
	Test	**99.05**	**96.57**	96.06	**96.32**	**0.51**	3.94

4.3 Optimal Cut-off Selection

The optimal cut-off point can be found from Youden's J statistic (also known as Youden's index) [25]. The index is defined for all points of a ROC curve, and the maximum value of the index is known as a good criterion for choosing the optimum cut-off point when a diagnostic test provides a numeric result [18].

Figure 3 illustrates the ROC curve and Youden's J statistic on the validation set. We used a threshold corresponding to the maximum value of the index from the ROC curve of the validation set as the cut-off value.

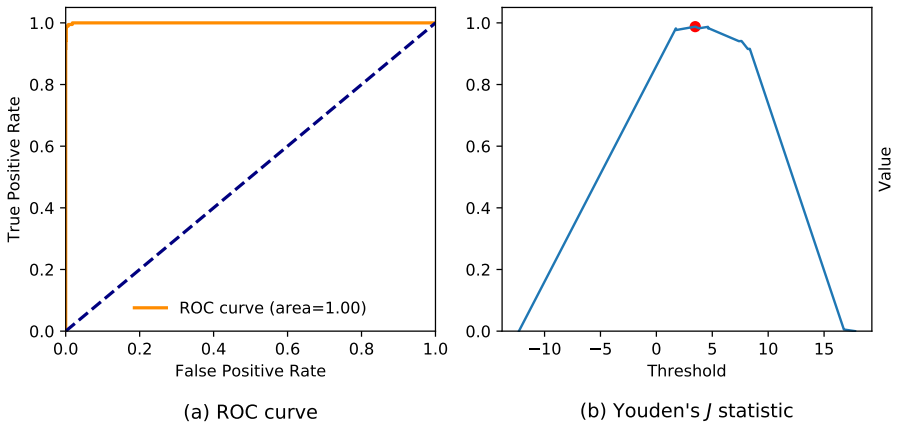

Fig. 3. Receiver operating characteristic curve (a) and Youden's J statistic on validation set (b). The red dot on (b) denotes the optimal cut-off threshold point for the validation set.

4.4 Patch Classification to Frame-Level Detection

Conventional CNNs only take a fixed-sized image patch, for instance, 224 × 224, as an input. In order to accept variable-sized images as input, therefore, additional modification is required. Inspired by Class Activation Map [27], thus,

we modified the last two layers of the trained model to handling the arbitrary size inputs.

In order to keep all responses from previous layers, first we removed the adaptive average pooling layer. Then we performed the following processes to handle arbitrary-sized tensor in the dense layer First, (B, C, H, W) tensor generated as result of the average pooling layer is reshaped to $(B, C, H \times W)$. Then, the last two axes of the tensor are swapped and reshaped as $(B \times H \times W, C)$. Finally, the dense layer is applied to $(B \times H \times W, 6)$ tensor and then, the output tensor is reshaped into $(B, H, W, 6)$. Here B, C, H, and W denote the batch size, channel, height, and width for feature map. Since the last dense layer reduces the number of the channel for the feature map to 6, the result tensor only has 6 channels.

4.5 Qualitative Wildfire Detection Results on Unseen Data

Figure 4 illustrates the frame-level wildfire detection results using Class Activation Map [27]. Note that the response for each input is normalized by subtracting the optimal cut-off threshold and then divided by the response range obtained from the validation set. As can be seen in the first and second rows of Figure 4, the trained model successfully detected the initial smoke and flame of wildfire.

5 Conclusions

In this work, we present an efficient and accurate CNN based early wildfire detection methods for video surveillance system. To realize our method, we collect a large dataset for early wildfire detection under the consideration of ambiguity in the real-world video surveillance environments. By adopting an efficient network architecture and sophisticate training strategy, we demonstrate that the learned model not only encouraging in performance in terms of accuracy, precision, recall, and F_1-score but also suitable for a video surveillance system. As future work, we plan to improve the performance of the model even a more diverse dataset. Furthermore, we plan to extend our work to early wildfire detection and localization as well.

References

1. Chen, T.H., Wu, P.H., Chiou, Y.C.: An early fire-detection method based on image processing. In: 2004 International Conference on Image Processing, 2004. ICIP'04. vol. 3, pp. 1707–1710. IEEE (2004)
2. Chenebert, A., Breckon, T.P., Gaszczak, A.: A non-temporal texture driven approach to real-time fire detection. In: 2011 18th IEEE International Conference on Image Processing. pp. 1741–1744. IEEE (2011)
3. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2009)

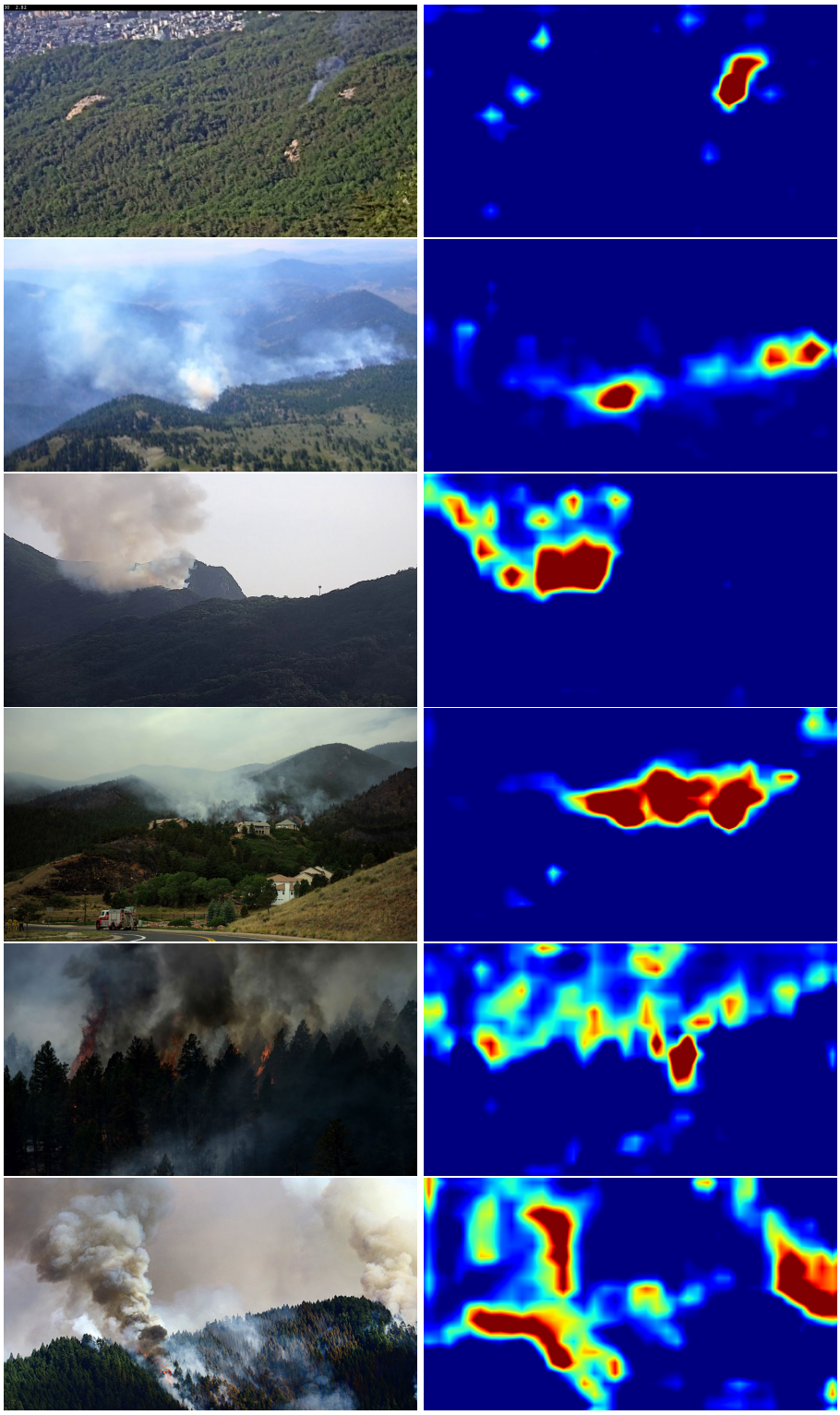

Fig. 4. The qualitative wildfire detection results on unseen data.

4. 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 Transactions on Circuits and Systems for Video Technology **25**(9), 1545–1556 (2015)
5. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems. pp. 2672–2680 (2014)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
7. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7132–7141 (2018)
8. Insurance Information Institute, I.: Facts + statistics: Wildfires (2019), `https://www.iii.org/fact-statistic/facts-statistics-wildfires`, [Online; accessed 9-December-2019]
9. Jadon, A., Omama, M., Varshney, A., Ansari, M.S., Sharma, R.: Firenet: A specialized lightweight fire & smoke detection model for real-time iot applications. arXiv preprint arXiv:1905.11922 (2019)
10. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
11. Ko, B.C., Cheong, K.H., Nam, J.Y.: Fire detection based on vision sensor and support vector machines. Fire Safety Journal **44**(3), 322–329 (2009)
12. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems. pp. 1097–1105 (2012)
13. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2980–2988 (2017)
14. Muhammad, K., Ahmad, J., Mehmood, I., Rho, S., Baik, S.W.: Convolutional neural networks based fire detection in surveillance videos. IEEE Access **6**, 18174–18183 (2018)
15. Namozov, A., Cho, Y.: An efficient deep learning algorithm for fire and smoke detection with limited data. Advances in Electrical and Computer Engineering **18**, 121–128 (11 2018). https://doi.org/10.4316/AECE.2018.04015
16. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019)
17. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4510–4520 (2018)
18. Schisterman, E.F., Perkins, N.J., Liu, A., Bondell, H.: Optimal cut-point and its corresponding youden index to discriminate individuals using pooled blood samples. Epidemiology pp. 73–81 (2005)
19. Sharma, J., Granmo, O.C., Goodwin, M., Fidje, J.T.: Deep convolutional neural networks for fire detection in images. In: International Conference on Engineering Applications of Neural Networks. pp. 183–193. Springer (2017)

20. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
21. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–9 (2015)
22. Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., Le, Q.V.: Mnasnet: Platform-aware neural architecture search for mobile. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2820–2828 (2019)
23. Tan, M., Le, Q.V.: Efficientnet: Rethinking model scaling for convolutional neural networks. arXiv preprint arXiv:1905.11946 (2019)
24. Wikipedia: List of wildfires — Wikipedia, the free encyclopedia (2019), `https://en.wikipedia.org/wiki/List_of_wildfires`, [Online; accessed 9-December-2019]
25. Youden, W.J.: Index for rating diagnostic tests. Cancer **3**(1), 32–35 (1950)
26. Zhang, D., Han, S., Zhao, J., Zhang, Z., Qu, C., Ke, Y., Chen, X.: Image based forest fire detection using dynamic characteristics with artificial neural networks. In: 2009 International Joint Conference on Artificial Intelligence. pp. 290–293. IEEE (2009)
27. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2921–2929 (2016)