2020 54th Annual Conference on Information Sciences and Systems (CISS) | 978-1-7281-4085-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/CISS48834.2020.1570617429

Transfer Learning for Wildfire Identification in UAV Imagery

Haiyu Wu, Huayu Li, Alireza Shamsoshoara, Abolfazl Razi, Fatemeh Afghah
School of Informatics, Computing, and Cyber Systems, Northern Arizona University, Flagstaff, AZ

***Abstract*—Due to Wildfire's huge destructive impacts on agriculture and food production, wildlife habitat, climate, human life and ecosystem, timely discovery of fires enable swift response to fires before they go out of control, in order to minimize the resulting damage and impacts. One of the emerging technologies for fire monitoring is deploying Unmanned Aerial Vehicles, due to their high flexibility and maneuverability, less human risk, and on-demand high quality imaging capabilities. In order to realize a real-time system for fire detection and expansion analysis, fast and high-accuracy image-processing algorithms are required. Several studies have shown that deep learning methods can provide the most accurate response, however the training time can be prohibitively long, especially when using online learning for constant refinement of the developed model. Another challenge is the lack of large datasets for training a deep learning algorithm. In this respect, we propose to use a pre-trained mobileNetV2 architecture to implement transfer learning, which requires a smaller dataset and reduces the computational complexity while not compromising the accuracy. In addition, we conduct an effective data augmentation pipeline to simulate some extreme scenarios, which could promise the robustness of our approach. The testing results illustrate that our method maintains a high identification accuracy in different situations - original dataset (99.7%), adding Gaussian blurred (95.3%), and additive Gaussian noise (99.3%)[1] [2].**

***Index Terms*—Wildfire detection, deep learning, CNN for mobile device, unmanned aerial vehicles (UAVs)**

I. Introduction

Based on the statistic result from the National Interagency Fire Center (NIFC), wildfires exhaust 10 million acres of land in 2016 and brought $6 billion irrecoverable damages from 1995 to 2014 in the United States [1]. Wildfires not only impact the wildlife, but more importantly endanger human lives. Therefore, early detection of wildfires before they get out of control is an urgent requirement. Wildfires are often initiated in remote forest areas where the common fire detection methods such as lookout tower stations fail to detect such fires in a timely manner. Moreover, conventional detection approaches can barely provide sufficient fire information about the precise fire locations, the orientation of fire expansion, etc. To detect forest fires, there are two general approaches using satellite images, and sensor networks. However, the satellites cannot provide real-time video or images since the quality of their images is highly impacted by weather conditions. Fire detection using wireless sensor networks is costly and high-maintenance to cover wide forest areas [2]. Manned aircraft can precisely survey a wide area in a short amount of time, however, this solution is costly and will endanger the life of pilots due to the high-temperature airflow and thick smoke.

Unmanned Aerial Vehicles (UAV) have been recently utilized in wildfire detection and management as a low-cost and agile solution to collect data/imagery considering their unique features such as 3-dimensional movements, easy to fly, maneuverability and flexibility [3]–[7]. The UAV networks can offer several features in such operations including tracking the fire front line, fast mapping of wide areas and damage assessment, real-time video streaming, and search-and-rescue [8]–[11]. In this paper, we focus on early detection of wildfires from collected images using UAVs with the goal of developing a low-computational mechanism appropriate for resource-constrained UAVs. Therefore, we develop a deep learning-based fire detection mechanism for accurate detection of fires using small training datasets.

Machine learning algorithms have been recently utilized for fire detection using aerial images. For instance, the authors in [2] developed a Support Vector Machine (SVM)-based approach to achieve real-time wildfire detection. SVM, as a classic machine learning algorithm, can achieve a good accuracy for fire detection. In [2], they improve the average true detection rates, but the average speed decreases dramatically in complex situations and the accuracy of this method is much lower than the expected rate for real-world wildfire detection.

Deep learning algorithms can provide high accuracy for fire detection as long as a sufficiently large dataset is used for training. The key advantage of the deep learning-based fire detection methods over other techniques is their capability in automatically learning high-level features rather than relying on hand-crafted feature descriptors to define the shape and texture of smoke or flame, and the color of the fire. In [12], [13], Convolutional Neural Networks (CNN) are utilized. However, they did not use global average pooling [14] after the last convolution layer, which leads to unnecessarily high number of parameters in the fully connected layers and costs a large computation resource. To improve the accuracy of the CNN approach, the second paper implemented a deeper CNN and adopted saliency detection to support the identification work of CNN [13].

The objective of this paper is to develop a Deep Learning (DL)-based fire detection approach for aerial images with a focus on reducing the computational complexity while

[1]Huayu Li and Haiyu Wu equally contributed to this paper.

[2]This material was based upon the work supported by the National Science Foundation under Grant No. 1755984. This work is also partially supported by the Arizona Board of Regents (ABOR) under Grant No. 1003330

maintaining a high accuracy. One challenge for developing DL-based wildfire detection methods is the lack of large-scale aerial wildfire imaging datasets. To address this challenge, we utilized MobileNetV2, which is pre-trained on the Imagenet database as our wildfire classifier [15], [16]. This lightweight network structure uses depth-wise separable convolutions [17] which can reduce the calculation complexity suitable for UAVs. Instead of using the saliency detection algorithm for enhanced detection, we directly feed the images into our network. In detail, our network has 17 convolution layers with an average processing speed of 19.8 ms/frame in a 3GHz 7^{th} generation i5 CPU device while it achieves an overall accuracy of 99.3% on the dataset collected from [18]. All the images in this dataset are shuffled and re-slipped into train and validation set as further described in Section IV-A. Furthermore, to maintain this accuracy in extreme cases where the collected images by the UAVs are impacted by severe motions or other sources of image distortion, we augment our images by different artifacts such as flipping images, darkening images, brightening images, etc. The experimental results show the higher performance of the proposed method against FireNet [13] and AlexNet [19] on our dataset.

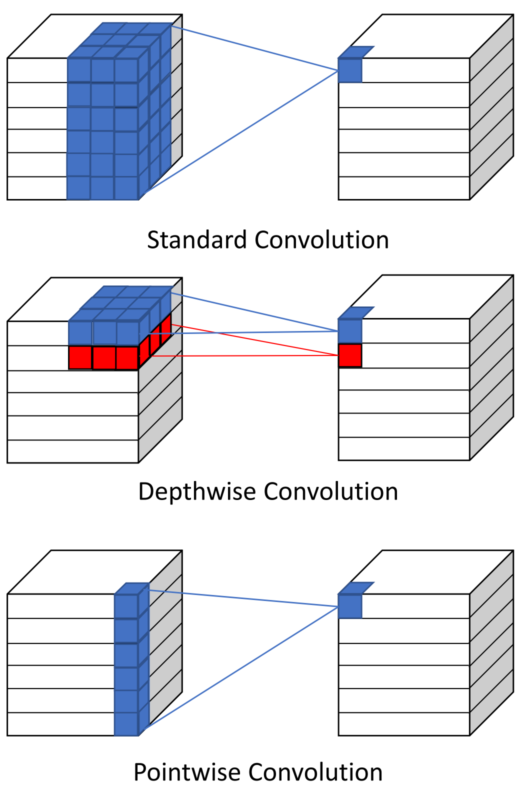

Fig. 1. The difference between the standard and Depth-wise convolution layers. From the graph, we observe that the depth-wise convolution involves fewer parameters and requires less computation. The depth-wise separable convolution can be viewed as a depth-wise convolution followed with a 1×1 standard convolution.

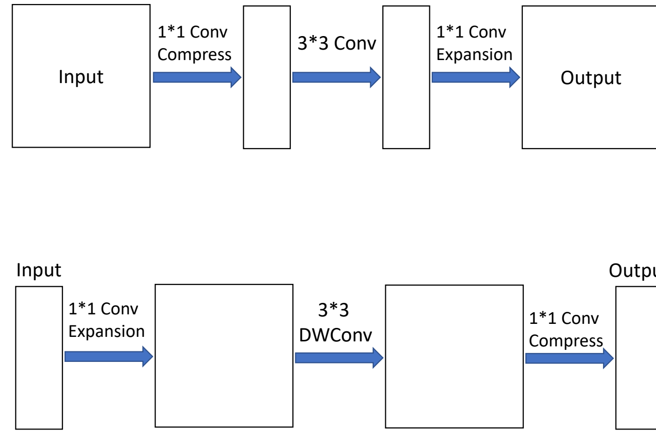

Fig. 2. The difference between standard residual block and inverted residual block. The standard residual block uses hourglass architecture, and the inverted residual block uses spindle architecture.

II. Related Works

Deep learning-based wildfire identification overcomes the spatio-temporal limitations of human observation [12], [13]. In [13], a CNN method assisted by saliency segmentation is proposed which utilizes the Region of Interest (ROI) proposal and Bayes-based saliency detection to localize a wildfire, then uses the banalization of the saliency map to support the CNN in identifying whether or not there is a wildfire. This saliency detection approach to localize the fire position with high probability before using CNN classifier can assure a high accuracy even with limited computing resources. However, it still suffers from a slow operation and cannot offer the expected real-time monitoring for time-sensitive fire detection missions. Lightweight CNN architectures have less parameters and provide almost the same performance as the traditional CNNs. Therefore, they can be appropriate candidates for fast fire detection using aerial images.

MobileNet [20] is a lightweight CNN architecture that focuses on deployment in mobile or embedded devices. MobileNet takes advantage of depth-wise separable convolution that can efficiently reduce the computational complexity and the number of model parameters. In MobileNetV2 [15], linear bottlenecks and inverted residuals are proposed to improve the network performance. Compared with the CNN model in [13], MobileNet architecture has a better learning capability while running faster and using fewer parameters. Most UAVs utilize embedded system-on-chip such as Arduino, Raspberry Pi, or ARM for light-weight computation tasks, therefore low-complexity architectures like MobileNet are preferred over the more complex DL structures [21]. Another advantage of such a low-complexity and real-time fire detection method is the possibility of fire detection by UAVs that operate in isolated and remote areas where the communication with the ground station is not available.

III. Proposed Method

A. Preliminaries

Before introducing the proposed method, we briefly introduce the key properties of the utilized MobileNet architecture [20].

1) MobileNet Series: MobileNet architecture is a lightweight neural network designed for embedded and mobile devices that has low execution time due to its few parameters [15], [20]. MobileNetV1 and MobileNetV2 replace the traditional convolution methods with depthwise separable convolutions [17] to achieve model compression which is inspired by inception [22], [23], and Xception (Extreme version of Inception) [17]. MobileNetV1 uses a simple and straight architecture similar to VGG [24] and AlexNet [19]. MobileNetV2 made further improvements by introducing inverted residual connections and linear bottlenecks into the original MobileNet architecture.

2) Depth-wise Separable Convolution: In this approach, the traditional convolution operation is divided into two steps. First, M 3×3 convolution kernels are used to convolve the M feature maps input one at a time, without summing them up. This operation is called depth-wise convolution that generates M separate output layers. Then, it uses 1×1 standard convolution kernels to aggregate the M previously generated channels into N output channels. This operation is called point-wise convolution. Splitting the convolution into two separate depth-wise and pointwise convolutions significantly reduces the number of parameters, and hence the training time. Fig. 1 illustrates the architectural difference between the standard convolution and depth-wise separable convolution operations.

3) Inverted Residual Block: MobileNetV1 [20] uses a very simple and straightforward structure similar to VGG [24] which is inefficient and hard to be trained as the depth of the network increases. The subsequent series of ResNet [25], DenseNet [26] and other structures have proven that residual connections usually have a positive effect on the performance by multi-scale features fusion. Since MobileNetV1 does not deploy residual connections which is a notable drawback. MobileNetV2 [15] solves this issue and makes a good use of residual connection. In MobileNetV2 [15], every bottleneck is switched to a inverted residual block. Inverted residual block is a convolutional module with residual connection, but a little different from standard residual block. To avoid a massive computation load, standard residual block uses 1×1 convolution to compress the channels before using 3×3 convolution to extract features that compose an hourglass structure. However, this structure leads to information loss in depth-wise separable convolution. To address this problem, MobileNetV2 uses an inverted residual block that expands the channels by 1×1 point convolution, then fed the expanded feature maps to 3×3 depth-wise convolution. Fig. 2 shows the difference between the standard residual block and the inverted residual block.

4) Linear bottlenecks: There is a simple but significant motivation behind linear bottlenecks, that is, Rectifier Linear Unit (ReLU) [27] that causes a large information loss for the tensor with a lower number of channels. Therefore, the 1×1 convolutional layer that performs the dimensionality reduction is not followed by a non-linear activation layer such as ReLU. The characteristics of non-linear activation, such as ReLU squash the input or just make the output of zero for negative input, while the dimension reduction itself is the process of feature compression, which makes the feature loss more serious.

B. Network Overview

In the present work, we proposed using MobileNetV2 which is pretrained on Imagenet dataset as our wildfire classifier. We replace the 1280×1000 fully connected layer with two fully connected layers with a size of 1280×256, and 256×2. We deployed *dropout* layers with a rate of 0.5 before each fully connected layer to counter overfitting. The whole architecture of the proposed method is summarized in Table I. In Table I, t stands for the expansion rate of the inverted residual block, c means the output channels, n is repeat number of each bottlenecks, and s is the step size of convolution. For step size greater than 1, the convolution operation performs a dowmsampling.

TABLE I
THE ARCHITECTURE OF MOBILENETV2, WE REPLACE THE 1280×1000 FULLY CONNECTED LAYER WITH 2 $1280 \times 256 \times 2$ FC LAYERS.

Input	operator	t	c	n	s
$224^2 \times 3$	**conv2d**	**-**	**32**	**1**	**2**
$112^2 \times 32$	**bottleneck**	**1**	**16**	**1**	**1**
$112^2 \times 16$	**bottleneck**	**6**	**24**	**2**	**2**
$56^2 \times 24$	**bottleneck**	**6**	**32**	**3**	**2**
$28^2 \times 32$	**bottleneck**	**6**	**64**	**4**	**2**
$14^2 \times 64$	**bottleneck**	**6**	**96**	**3**	**1**
$14^2 \times 96$	**bottleneck**	**6**	**160**	**3**	**2**
$7^2 \times 160$	**bottleneck**	**6**	**320**	**1**	**1**
$7^2 \times 320$	**conv2d 1x1**	**-**	**1280**	**1**	**1**
$7^2 \times 1280$	**avgpool 7x7**	**-**	**-**	**1**	**-**
$1 \times 1 \times 1280$	**conv2d 1x1**	**-**	**256**	**-**	
$1 \times 1 \times 256$	**conv2d 1x1**	**-**	**2**	**-**	

C. Data Augmentation

It is known that CNN is a data-driven deep learning algorithm that requires large datasets for training purpose. When large datasets are not available, using pre-trained models to implement transfer learning is a promising solution. Also, reasonable data augmentation methods can be adopted. To investigate the efficiency of the proposed method in real world situations. Besides using random shift, rotation, and flipping, we also deploy a data augmentation strategy as follows:

- Varying illumination intensity, especially extremely dark or bright conditions can negatively impact the classification accuracy. When UAVs operate in a dense jungle or at nights, the captured image may be too dark, which may cause the trained CNNs miss detecting smokes or small flames. To simulate darkness, we randomly scale each pixel by multiplying a rate between 0.75 to 0.9 during the pre-processing the training dataset. Likewise, to simulate sharp lights, we use the scaling rate of 1 to 1.25, and the pixels are clipped between 0 to 255.

- The UAVs may cruise in the wild for a long time causing the camera lens get dirty over time, which makes the captured images blurred or noisy. The most extreme case is when the camera lens is damaged and part of the image is missing. To simulate such conditions, we deploy random blurry kernels and additive noise to the training images. Also, a random zero masking is used to discard part of the images.

The pipeline of our data augmentation method is shown in Algorithm 1 for flipping, rotation, blurring, noisifying, and etc.

Algorithm 1: The algorithm of the data augmentation which is used in the proposed method

Load the Image:
Generate 6 uniformly distributed random variables $\alpha_1, \alpha_2, \ldots, \alpha_6 \sim \mathcal{U}(0, 1)$
if $\alpha_1 < 0.5$ **then**
  Flip the image
end
if $\alpha_2 < 0.5$ **then**
  Rotate the image
end
if $\alpha_3 < 0.5$ **then**
  Add noise to the image
end
if $\alpha_4 < 0.5$ **then**
  Make the image blurry
end
if $\alpha_5 < 0.5$ **then**
  if $\alpha_6 < 0.5$ **then**
    Make the image darker
  else
    Make the image brighter
  end
  Export the image to the output
else
  Export the image to the output
end

IV. Experimental results

A. Dataset

To build our dataset, we collect 1048 positive samples (images with flame or smoke) and 1048 negative samples (images with no fire nor smoke) from the Internet. We split this dataset into training set (888 images for each class and validation set (160 images for both positive and negative sample). We trained the model on training set, and evaluate the performance on validation set. We resize the images into 224×224 and then scale the values of pixels into $(0, 1)$. We noticed that some situations that may lead to wrong classification. Generally, sharp lights in no-fire images can be confused with fire. Smoke and fog also looks similar. Sample images of these situations are shown in Fig. 3. The red rectangle represents the confusion area in each image.

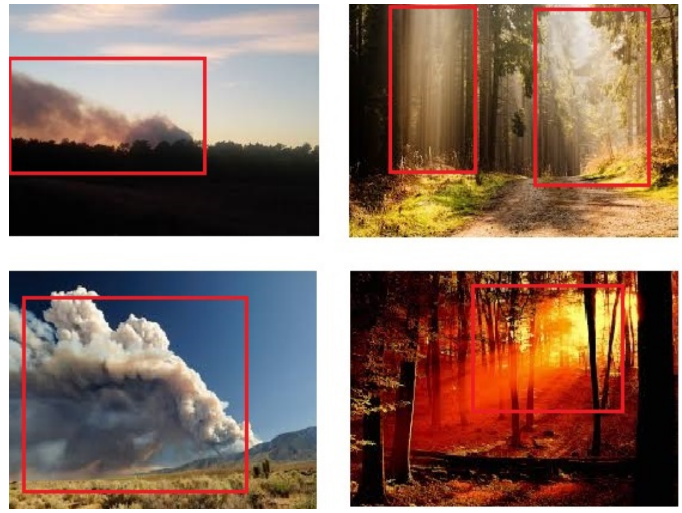

Fig. 3. Some samples from our dataset that may lead to miss-classification errors.

B. Training Details

We train our models using Keras with TensorFlow backend [28]. We use the Adam [29] as our optimizer with an initial learning rate of 0.001, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We trained the model for 4500 epochs, and then reduced the learning rate to 0.0005 and 0.0001 at 3000 and 4000 epochs. We used a GTX 1080 Ti GPU, and set up a batch size of 64.

TABLE II
COMPARISON FOR NUMBER OF PARAMETERS AND ACCURACY FOR CNN, FIRENET, ALEXNET, AND THE PROPOSED METHOD.

Methods	Accuracy	Million Parameters
CNN [12]	**85.6%**	**17**
FireNet [13]	**97.5%**	**2.94**
AlexNet [19]	**95%**	**43**
Proposed method	**99.3%**	**2.5**

C. Test results on original, noisy, and Blurry images

We re-implement the algorithms and follow all the hyper parameters in [12] [13], and use pre-trained AlexNet [19] with the same training strategy as our model. The comparison results on clean validation set are shown in Table II. We also test three alternative methods that are commonly used in wildfire detection using our original dataset. The results confirm that the proposed method achieves a classification accuracy of 99.3% on the original images that outperforms all the alternative methods. The two alternative methods (FireNet and AlexNet) maintain a high identification accuracy, but still lower than the proposed method. The performance of standard CNN [12], which has 3 convolution layers and 2 fully connected layer, is out of an acceptable error range. It is noteworthy that the proposed method overcomes other methods in terms of the modeling complexity as well. Our method utilizes only 2.5 million parameters that is significantly

Fig. 4. We add Gaussian blur function to blur our images (fire and no fire). (1): Original pictures, (2): $\sigma = 3$, (3): $\sigma = 5$, (4): $\sigma = 7$.

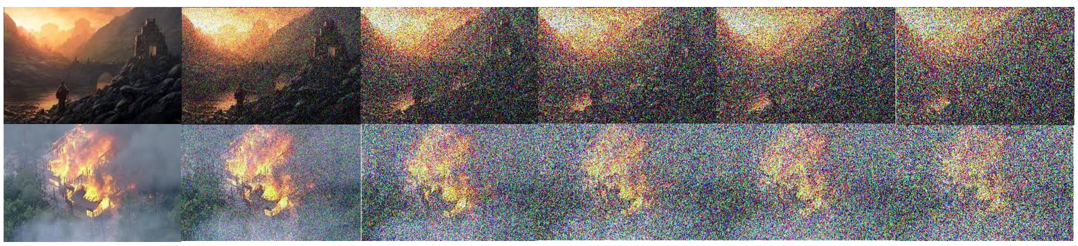

Fig. 5. Demonstration of original image (1) and noisy images with standard deviations (2): $\sigma = 0.15$, (3): $\sigma = 0.35$, (4): $\sigma = 0.45$, (5): $\sigma = 0.5$, and (6): $\sigma = 0.65$.

fewer than other methods. The second low-complexity model is FireNet with about 3 million parameters.

To examine the robustness of our model and data augmentation strategy in some extreme situations, we conduct two experiments on noisy and blurry datasets. In order to investigate the performance of the proposed method in classifying blurry images, we used a Gaussian kernel to blur the images in the test dataset. The standard deviation σ_B is set at 3, 5, 7 to evaluate the performance of the model under different blurring degrees. Figure 4 shows images under different blurring degrees. For testing our model on noisy validation data, we add Gaussian noise with different levels represented by $\sigma_N = 0.15, 0.35, 0.45, 0.5, 0.55, 0.65$ after normalizing the pixel values of the input images into $(0, 1)$. Figure 5 shows images with different noise levels. The test results are shown in Table III and Table IV, respectively. We can observe the outstanding performance of our method under extreme cases.

More specifically, Table III represents the results for blurred images with different σ_B values that represent the standard deviation of Gaussian kernel used for blurring the image. The results show that the propose algorithms achieves a classification accuracy of 95% at $\sigma_B = 3$ that exhibits a margin of about 7.5% with respect to the second best method of FireNet with accuracy 87.8%. This improvement is even more significant when the blurriness level increases (93.2% for $\sigma_B = 5$ and 86.5% for $\sigma_B = 7$).

Likewise, Table IV presents the identification accuracy results of all methods, when Gaussian noise is added. At noise level $\sigma_N = 0.15$, the identification accuracy of the proposed model is 99.3% significantly higher than other methods. The margin with respect to the second best (FireNet with a 96.3% accuracy) is about 2.8%. This margin improves with increasing the noise level and it is $2.8, 9.6, 23.1, 26.3, 31.8, 27.8\%$ for $\sigma = 0.15, 0.35, 0.45, 0.5, 0.5, 0.65$, respectively. At $\sigma_B = 0.65$ our method still maintain an acceptable accuracy of 87.8%, while others fail in providing accuracy above 60%.

V. Conclusion

In this paper, we considered an important problem of using aerial imaging for early fire detection noting the fact that current deep learning methods fail in providing accurate fire detection results with small datasets and low computational complexity appropriate for UAVs with limited onboard processing powers. To address this issues, we used a lightweight network, MobileNetV2 [15] to realize transfer learning. We also used an effective data augmentation pipeline by manipulating the original images by flipping, changing the brightness, blurring, and adding Gaussian noise to the images. This method not only alleviates the insufficient dataset size, but also helps to simulate real world conditions when the quality of captured

TABLE III
ACCURACY OF CNN, ALEXNET, FIRENET AND THE PROPOSED METHOD IN IDENTIFICATION FOR BLURRY IMAGES WITH THREE DIFFERENT VALUES OF GAUSSIAN KERNEL σ_B.

Gaussian blur	$\sigma_B = 3$	$\sigma_B = 5$	$\sigma_B = 7$
FireNet	**87.8%**	**70.3%**	**54.4%**
AlexNet	**83.4%**	**66.9%**	**51.9%**
CNN	**82.8%**	**57.8%**	**50.3%**
Proposed method (without data augmentation)	**93.8%**	**75.0%**	**65.9%**
Proposed method (with data augmentation)	**95.3%**	**93.2%**	**86.5%**

TABLE IV
IDENTIFICATION ACCURACY OF CNN, ALEXNET, FIRENET AND THE PROPOSED METHOD FOR DIFFERENT NOISE LEVELS.

Gaussian noise	$\sigma_N = 0.15$	$\sigma_N = 0.35$	$\sigma_N = 0.45$	$\sigma_N = 0.5$	$\sigma_N = 0.55$	$\sigma_N = 0.65$
FireNet	**96.3%**	**89.7%**	**74.1%**	**68.4%**	**61.6%**	**60%**
AlexNet	**92.5%**	**87.5%**	**69.1%**	**63.1%**	**54.4%**	**53.8%**
CNN	**81.6%**	**74.7%**	**63.1%**	**58.8%**	**44.1%**	**52.2%**
Proposed method (without data augmentation)	**98.1%**	**94.1%**	**83.8%**	**75.3%**	**68.1%**	**63.1%**
Proposed method (with data augmentation)	**99.3%**	**99.3%**	**97.2%**	**94.7%**	**93.4%**	**87.8%**

images are impacted by the environmental conditions, the camera quality, and the light intensity. Simulation results show that the proposed method performs significantly better than the alternative methods by a big margin ranging from 2.8% to 31.8% when the captured images are subject to noise and blurriness. The results confirm the suitability of the proposed method for wild-fire detection using aerial monitoring systems.

REFERENCES

[1] "National significant wildland fire potential outlook," https://www.nifc.gov/fireInfo/fireInfo_main.html, accessed: 2019-12-08.

[2] Y. H. Habiboglu, O. Gunay, and A. E. Cetin, "Real-time wildfire detection using correlation descriptors," in *2011 19th European Signal Processing Conference*. IEEE, 2011, pp. 894–898.

[3] D. Twidwell, C. Allen, and C. L. S. E. C. Detweiler, J. Higgins, "Smokey comes of age: unmanned aerial systems for fire management," *Frontiers in Ecology and the Environmentl*, August 2016.

[4] M. Półka, S. Ptak, and Łukasz Kuziora, "The use of uav's for search and rescue operations," *Procedia Engineering*, vol. 192, pp. 748 – 752, 2017, 12th international scientific conference of young scientists on sustainable, modern and safe transport. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877705817326759

[5] G. Hristov, J. Raychev, D. Kinaneva, and P. Zahariev, "Emerging methods for early detection of forest fires using unmanned aerial vehicles and lorawan sensor networks," in *2018 28th EAEEIE Annual Conference (EAEEIE)*, Sep. 2018, pp. 1–9.

[6] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz, "Help from the sky: Leveraging uavs for disaster management," *IEEE Pervasive Computing*, vol. 16, no. 1, pp. 24–32, Jan 2017.

[7] M. Erdelj and E. Natalizio, "Uav-assisted disaster management: Applications and open issues," in *2016 International Conference on Computing, Networking and Communications (ICNC)*, Feb 2016, pp. 1–5.

[8] F. Afghah, A. Razi, J. Chakareski, and J. Ashdown, "Wildfire monitoring in remote areas using autonomous unmanned aerial vehicles," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2019, pp. 835–840.

[9] S. Islam and A. Razi, "A path planning algorithm for collective monitoring using autonomous drones," in *2019 53rd Annual Conference on Information Sciences and Systems (CISS)*, March 2019, pp. 1–6.

[10] S. Islam, Q. Huang, F. Afghah, P. Fule, and A. Razi, "Fire frontline monitoring by enabling uav-based virtual reality with adaptive imaging rate'," in *Asilomar Conference on Signals, Systems and Computers*, November 2019, pp. 1–6.

[11] A. Shamsoshoara, M. Khaledi, F. Afghah, A. Razi, J. Ashdown, and K. Turck, "A solution for dynamic spectrum management in mission-critical uav networks," *arXiv preprint arXiv:1904.07380*, 2019.

[12] S. Kim, W. Lee, Y.-s. Park, H.-W. Lee, and Y.-T. Lee, "Forest fire monitoring system based on aerial image," in *2016 3rd International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*. IEEE, 2016, pp. 1–6.

[13] Y. Zhao, J. Ma, X. Li, and J. Zhang, "Saliency detection and deep learning-based wildfire identification in uav imagery," *Sensors*, vol. 18, no. 3, p. 712, 2018.

[14] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[16] 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.

[17] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[18] "Forest-fire-detection-through-uav-imagery-using-cnns," https://github.com/LeadingIndiaAI/Forest-Fire-Detection- through-UAV-imagery-using-CNNs/tree/master/data/train/Fire, accessed: 2018-06-26.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[20] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[21] S. Vaddi, C. Kumar, and A. Jannesari, "Efficient object detection model for real-time uav applications," *arXiv preprint arXiv:1906.00786*, 2019.

[22] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[23] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[25] 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.

[26] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[27] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[28] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.