IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 7, 2020, accepted December 13, 2020, date of publication December 28, 2020, date of current version January 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3047764

Tackling Small Data Challenges in Visual Fire Detection: A Deep Convolutional Generative Adversarial Network Approach

ZHAOYI XU, YANJIE GUO, AND JOSEPH HOMER SALEH, (Senior Member, IEEE)
School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

Corresponding author: Zhaoyi Xu (zxu328@gatech.edu)

This work was supported by the Space Technology Research Institute through the National Aeronautics and Space Administration's (NASA's) Space Technology Research Grants Program.

ABSTRACT Fire detection technologies remain a critical component of building automation. With the recent significant advances in computer vision, visual fire detection methods have been developed and integrated into building surveillance systems. Overfitting and accuracy challenges remain in fire detection when training datasets are limited. In this work, we tackle these challenges by developing a deep convolutional generative adversarial network (DCGAN) for highly accurate visual fire detection when training images are limited. Our model addresses three types of errors in visual fire detection with small training datasets: model overfitting, fire probability overestimation, and fire probability underestimation. The DCGAN includes a generator of fake fire images for self-supervised learning (SSL) and a discriminator for image classification. We designed computational experiments with high-quality datasets to test and validate our model against other supervised learning approaches. We also benchmarked the performance of the DCGAN against a best-in-class deep visual fire detection model. The results show that our model significantly outperforms other fire detection models on all performance metrics when trained with the same small dataset. The results demonstrate that the DCGAN effectively mitigates the three types of error when the training dataset is limited.

INDEX TERMS Deep convolutional generative adversarial network, self-supervised learning, visual fire detection.

I. INTRODUCTION

Fire detection technologies remain a critical component of building automation and information systems. They are essential for monitoring both indoors and outdoors for fire signatures such as smoke, heat, and radiation, and to identify early signs of fires to trigger appropriate responses. Significant progress has been made with these technologies in the last decades in part due to advances in sensor design and related technology [1]–[8]. Nonetheless, important challenges with fire detection remain, and these can roughly be subsumed under two broad headings, insufficient sensitivity on the one hand, and elevated false alarm rates on the other hand [9]. These overlap with and are related to challenges with the reliability, accuracy, and specificity of these technologies.

With the recent significant advances in computational power, computer vision [10], [11], [12], and machine learning (ML), visual fire detection methods have been developed [13]–[15] and integrated into buildings and outdoors camera surveillance systems. Deep learning (DL) is a key enabler of these applications. It uses deep artificial neural architecture to fulfill complex regression, classification, and some unsupervised functions. DL can be prior assumption-free and has significant potential to improve model accuracy and tease out more latent information from the raw data compared with shallow ML. One of the most common DL technologies is convolutional neural network (CNN), which uses convolutional and pooling layers to analyze high-dimensional data, such as pictures and videos. Advanced CNN, such as AlexNet [16], GoogLeNet [17], and DenseNet [18], are used for image classification, and they provide excellent performance in a host of applications. Advanced CNNs have the potential to be an effective solution for accurate visual fire detection. In this work, we develop a

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad Ayoub Khan.

novel, highly accurate visual fire detection method, based on state-of-the-art deep convolutional neural network with a generative adversarial network, which significantly outperforms existing methods when training data are limited.

Some background is required in order to understand the context and the challenges our method overcomes. We briefly discuss those next.

For DL and CNN visual fire detection, the quality of the training data can significantly affect the accuracy of the model. A poor-quality training set can degrade the model accuracy and result in errors, such as overfitting and group bias [19], [20]. In visual fire detection applications, one critical problem is the quality of the training images. Although fire images are abundant, high-quality training images are limited, and in some cases rarely available (e.g., fire images in micro-gravity environment). This limited availability of high-quality training images can cause overfitting, and it degrades the model accuracy. Three types of errors are recognized in visual fire detection when the training images are limited: (1) model overfitting, (2) fire probability overestimating, and (3) fire probability underestimating. First, model overfitting is caused by insufficient high-quality training dataset. Overfitting leads to and is reflected by a significant gap between training and testing accuracy. In this situation, one can achieve perfect training accuracy, but the testing accuracy (fire detection on new data not seen in the training) will be degraded. Second, for the fire probability overestimation error, the model has better accuracy of fire than non-fire image detection. When the training size is limited, these models are not trained by images with diverse fire scenarios. The image discrimination model would consider images with only partial characteristics of the flame, such as red color, as a fire image. Images with environmental disturbances, such as a red board in the background, can have these partial characteristics. Consequently, the discriminator can wrongly classify them into the fire category. This fire probability overestimation error leads to high false alarm rates in real-world applications. Third, for the fire probability underestimation error, this can be caused by a disparity between fire and non-fire image classification accuracies. This can be the result of, as noted previously, a limited (in size and diversity) training dataset. The image discriminator in this case would consider images without the full set of features of fires as non-fires, thus wrongly classifying them. This fire probability underestimation error leads to high missed detection rates in real-world applications.

In this work, we propose to overcome these three types of errors in visual fire detection when training images are limited. To accomplish this task, we leverage self-supervised learning (SSL), which is a subcategory of ML designed for applications when the training data is insufficient. More specifically, we use generative learning to produce fake fire images to improve the training efficiency and mitigate the overfitting problem. Generative learning originally aims at generating realistic fake images in computer vision. Goodfellow *et al.* [21] developed generative adversarial network (GAN) using two neural networks architecture (generator and discriminator) for a min-max optimization. GAN and its variants, such as deep convolutional generative adversarial network (DCGAN) [22] and BigGAN [23], provide excellent performance in image generative tasks. Furthermore, GAN can also be used with the image discriminator training to prevent overfitting and improve discrimination accuracy when high-quality labeled data is limited. For example, Ravanbakhsh *et al.* [24] used GAN to improve the training of the discriminator for the identification of abnormal crowd behavior when the truth data for supervised learning is lacking or insufficient. They compared their results with standard benchmarks data and showed that the GAN discriminator can outperform previous state-of-the-art models in abnormal crowd behavior detection. We follow a somewhat related path, and instead of abnormal crowed behavior, we are concerned with visual fire detection. The objective of this work is to develop an SSL architecture with DCGAN for visual fire detection, with an advanced discriminator network that overcomes the three types of errors in visual fire detection noted previously when training images are limited. We discuss the details in the next subsections.

The main contribution of this work is the development and validation of a highly accurate visual fire detection DL method when the training dataset is limited. Our method leverages Generative Adversarial Network, and it overcomes the three common types of errors found in visual fire detection discussed previously within a ‘‘small data’’ context.

The remainder of the article is as follows. A brief literature review is provided in Section II. Our DCGAN architecture along with its details is discussed in Section III. The computational experiments and results of DCGAN are discussed in Section IV, along with a comparative performance analysis with other visual fire detection methods. Finally, Section V concludes this work.

II. BRIEF LITERATURE REVIEW

Visual fire detection, as noted previously, is becoming increasingly popular and gradually integrated into surveillance systems. Initially, researchers sought to develop handcrafted techniques for fire detection by focusing on the color and motion properties of the flame. For example, Chen *et al.* [25] leveraged both the chromatic and dynamic properties of fire and smoke in their visual fire detection scheme. Similarly, Çelik *et al.* [26] sought to distinguish fires from (non-fire) environmental disturbances by leveraging two different color spaces, RGB and YCbCr, to devise a more accurate classification model. Flame motion is another fire visual signature that has been used as the criterion to detect fires. For example, Rafiee *et al.* [27] and Rinsurongkawong *et al.* [28] used the properties of the flame and smoke to visually detect fires. Although computationally efficient and can be easily deployed on readily-available hardware, these approaches to visual fire detection suffer from two thorny drawbacks: low detection accuracy and elevated false alarm rates when (non-fire) environmental disturbances are present.

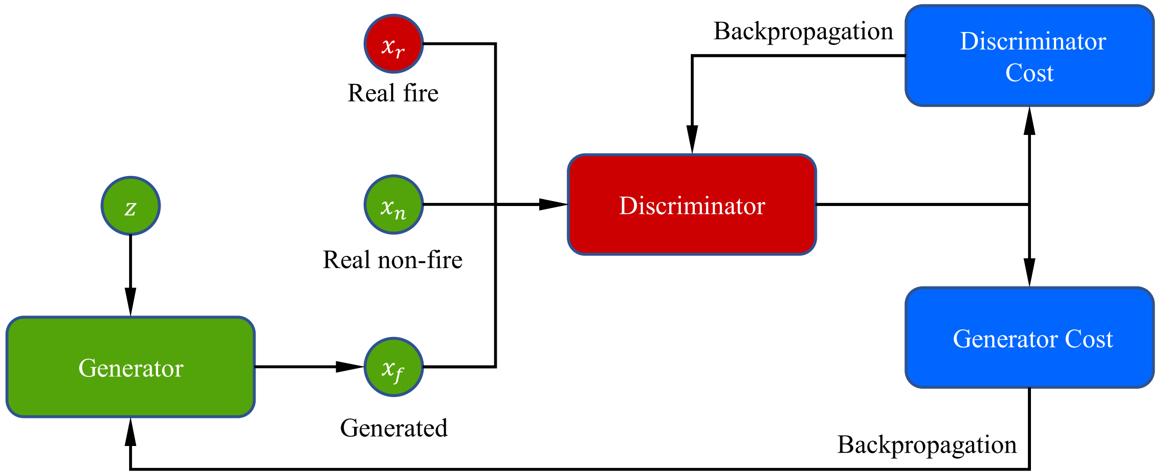

FIGURE 1. Overview of the architecture and training of the DCGAN.

To overcome these drawbacks, some researchers have leveraged DL models for visual fire detection, and different approaches have been proposed for this task. For example, Zhang *et al.* [29] utilized fire patches detection with a fine-tuned pre-trained AlexNet [16] for forest fire detection. Sharma *et al.* [30] proposed a DL fire detection approach based on VGG16 [31] and Resnet50 [32] architectures. Muhammad *et al.* fine-tuned different variants of CNNs, such as AlexNet [33], SqueezeNet [34], GoogleNet [35], and MobileNetV2 [36]. They reported excellent visual fire detection accuracy using these advanced DL models for both fire and non-fire images.

These DL visual fire detection models can effectively solve the problems of limited accuracy and elevated false alarm rate of the traditional models. However, to achieve this level of performance, all these DL models require significantly large datasets of fire and non-fire images for training. Jadon *et al.* [15] noted that the datasets for visual fire detection training can be low-quality and have a lack of balance between fire and non-fire images, which in turn can lead to inaccuracies and the three types of errors discussed previously.

III. DCGAN FOR VISUAL FIRE DETECTION: ARCHITECTURE AND ANALYTICS

This section introduces our proposed architecture, the DCGAN, and its underlying analytics for overcoming the challenges discussed previously in visual fire detection when the number (or data size) of training images is limited. We first present a high-level overview of the DCGAN for visual fire detection, which includes a discriminator and a generator of fire images. Next, we present the details of the fire image discriminator. Finally, we discuss the details of the generator and its training, as well as the datasets used in this work. In the next sections, we examine the computational experiments, and the results obtained with this DCGAN and other methods for visual fire detection for a comparative performance analysis.

A. DCGAN ARCHITECTURE: AN OVERVIEW

Recall our primary objective is to develop an accurate visual fire detection model when training images are limited. The DCGAN, which implements a form of self-supervised learning or SSL, is our proposed approach. Its architecture is shown in Fig. 1. The details are discussed in the next subsections.

The DCGAN consists of two major parts: (1) a discriminator network, and (2) a generator network. The functions of these networks can be fulfilled by different models.

For the discriminator, we use two different models, first a naïve CNN, and second a more advanced deep CNN termed fire detection SqueezeNet. The purpose of having two options is to assess the relative performance advantage of the fire detection method with either a naïve CNN or a fire detection SqueezeNet for the discriminator, if any.

For the generator, we use a transposed convolutional network, or TCNN, to produce fake fire images. The generator only captures partial characteristics of real fire images, such as light spots but without detailed flame shapes for example. These fake images, in turn, assist in the training of the discriminator and enable it to improve its performance when the training dataset is limited. The use of this generator is meant to improve the accuracy of the classification task and to reduce overfitting. The details of the generator and discriminator are discussed in the next subsections, and the computational experiments and results in section V.

As shown in Fig. 1, in each iteration of the training process, we generate a random latent vector (z) and use it as an input to the generator. In our model, z has a dimension of 50 for a large latent space. This is similar to the optimal latent size recommended in Ref. [37]. Next, the generator uses the latent vector (z) to generate fake fire images (x_f). Following this, three different types of images are used as input to the discriminator, real-world fire image (x_r), non-fire image (x_n), and fake fire images (x_f). The discriminator then classifies these images and provides estimated labels ($\hat{y}$) as fire, non-fire, or fake. Finally, this label is compared with the ground truth, and we calculate loss functions for both the generator and discriminator. We use the binary cross-entropy (BCE) loss function, as shown in Eq. 1, for the backpropagation and training of the generator and discriminator. The generator loss is calculated as shown in Eq. 2, where $\hat{y}_f$ is the estimated label of for fake images. We penalize the generator when the fake fire images are correctly identified by the discriminator. The discriminator loss is calculated as shown in Eq. 3, where $\hat{y}_r$ and $\hat{y}_n$ are the estimated labels for fire and non-fire real images. We penalize the discriminator when (1) it incorrectly classifies the real fire images as non-fire or fake images, (2) it incorrectly classifies non-fire as fire images, and (3) it incorrectly classifies fake images as real fire images. We conduct backpropagation with the ADAM optimizer on the loss functions [38] and train the generator and discriminator simultaneously. We terminate the overall DCGAN training when the loss function of the discriminator stops decreasing (no further improvement in the discriminator output). During the training process, we supervise the generator error, which can become unstable. This potential instability in the generator training is mitigated by the reinforcement mechanism and adding image noise, discussed shortly.

$$BCE(y,\hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log\left(\hat{y}_i\right)+\left(1-y_i\right)\log\left(1-\hat{y}_i\right)\right] \quad (1)$$

$$loss_g = BCE(1,\hat{y}_f) \quad (2)$$

$$loss_d = BCE\left(1,\hat{y}_r\right)+BCE\left(0,\hat{y}_n\right)+BCE(0,\hat{y}_f) \quad (3)$$

B. TWO MODELS FOR THE DISCRIMINATOR: NAÏVE CNN AND FIRE DETECTION SQUEEZENET

In this subsection, we develop the two different networks for the discriminator noted previously: first, a naïve CNN for a simple fire image discrimination model (with small hardware memory requirement); second a more advanced fire detection SqueezeNet (with larger memory requirement).

1) NAÏVE CNN FOR THE FIRE IMAGE DISTRIMINATOR

Our simple fire image discriminator with small model complexity is a naïve CNN shown in Fig. 2, where ''Conv'' stands for convolutional layer and ''Den'' for the fully connected dense layer.

This naïve CNN discriminator operates as follows. First, we input the 3 channels RBG images with a resolution of 64×64 into the network. We use two convolutional layers to map the original images to hidden states with a size of $4\times4\times64$. Next, we use two dense layers to transfer the hidden states (the input of Den1 and Den2) to one latent variable (the input of Sigmoid). We use the sigmoid function to map this latent variable to the probability of fire image as the output of this naïve CNN. We use the Leaky-ReLU activation function [39] as shown in Eq. 4 to add nonlinearity to the discrimination model, with $\alpha = 0.2$ as the slope when $x < 0$. We discuss the details of the weight function of each layer of this naïve CNN in Appendix A. The weight of the CNN parameters takes the memory of 0.260Mb and the processing memory per image is 0.067Mb.

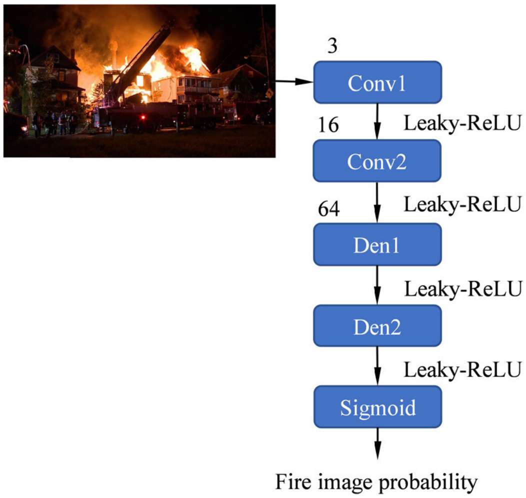

FIGURE 2. Structure of the fire detection naïve CNN discrimination model in the DCGAN for visual fire detection. The numbers 3, 16, and 64 indicate the number of channels of the input or the output of the convolutional layer.

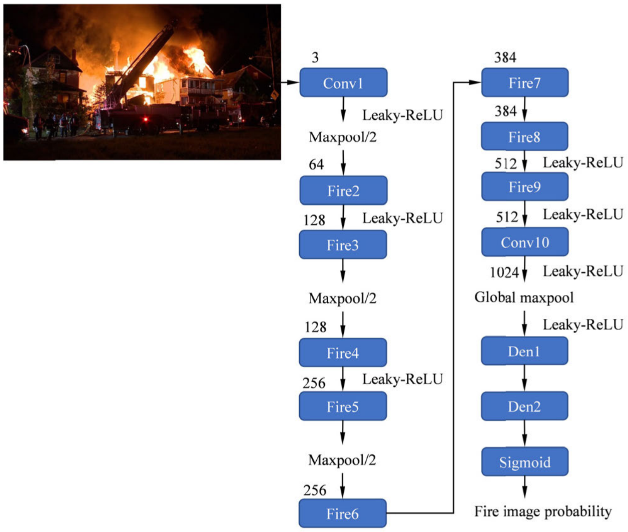

FIGURE 3. Structure of the fire detection SqueezeNet discrimination model in the DCGAN for visual fire detection.

$$f(x) = \begin{cases} x & \text{when } x > 0 \\ -\alpha x & \text{when } x < 0 \end{cases} \quad (4)$$

2) SQUEEZENET FOR THE FIRE IMAGE DISCRIMINATOR

Iandola *et al.* [40] developed SqueezeNet and noted that it provided excellent classification accuracy, as good as that of the award-winning AlexNet [16] while requiring 50 times fewer parameters than AlexNet. In order to tailor SqueezeNet to our fire detection and further improve the accuracy of this general model in our specific application, we modified the original model and developed a dedicated fire detection SqueezeNet. The structure of our fire detection SqueezeNet is shown in Fig. 3.

This fire detection SqueezeNet image discrimination operates as follows. First, we input the 3 channels RGB images with a size of 64 × 64 into the network. Then, we use convolutional layers, maxpoolings, and fire models (details in Appendix B) to map the input to a hidden vector of size 1024. Second, we use two dense layers to transfer the output of the global maxpool to a latent variable and use the sigmoid activation function to map this latent variable to the probability of fire. We discuss the weight function of each layer and the overall memory size of this fire detection SquzzeNet in Appendix C. The weight size of this SqueezeNet fire detection model is 5.24Mb, and the overall processing memory per image is 0.82Mb, roughly an order of magnitude larger than the memory requirements with the previous naïve CNN discriminator.

To improve its fire detection accuracy, we implemented three important modifications to the original SqueezeNet model. First, we modified the classification problem with 1000 outputs in the original SqueezeNet to a single output ranging from 0 to 1 as the probability of fire. This is achieved by using two fully connected dense layers. Second, we added Leaky-ReLU activation function to better model nonlinearities and to avoid the vanishing gradient problem in model training. Third, we modified the global average pool in the original model to the global maxpool (after Conv10). The reason for this modification is as follows: light spots in images can be used as possible fire signatures, and they can be represented as maximum values in the output of the Conv10 layer. The global average pool in the original SqueezeNet can smooth out or eliminate this useful information for fire image discrimination. We mitigate this effect by using the global maxpool. These three modifications can effectively improve the accuracy of the fire detection SqueezeNet in our computational experiments. We examine these implications in section IV.

C. GENERATOR: TRANSPOSED CONVOLUTIONAL NEURAL NETWORK AND TRAINING REINFORCEMENT

In this subsection, we introduce the transposed convolutional neural network (TCNN) generator model and discuss its training reinforcement mechanism.

1) TRANSPOSED CONVOLUTIONAL NEURAL NETWORK

The TCNN consists of transposed convolutional layers and fully connected dense layers. The transpose convolutional layer (TCL), also known as the deconvolutional layer, is a widely popular upsampling method used to generate an output feature map with a dimension larger than that of the input feature map. In our work, we use TCL to enlarge the input spatial size and decrease the channel number from 128 to 3 for the RBG information of the generated images. More details on TCL and its applications can be found in Ref. [41].

The structure of our TCNN generator is shown in Fig. 4, where "T-conv" stands for the TCL. This generator consists of three dense layers and nine TCLs to convert the input hidden variable z to a 64 × 64 × 3 RBG fake fire images. We use the Leaky-ReLU activation function in the TCNN as well for nonlinearity in the image generation. The parameters and model complexity in this TCNN are discussed in Appendix D.

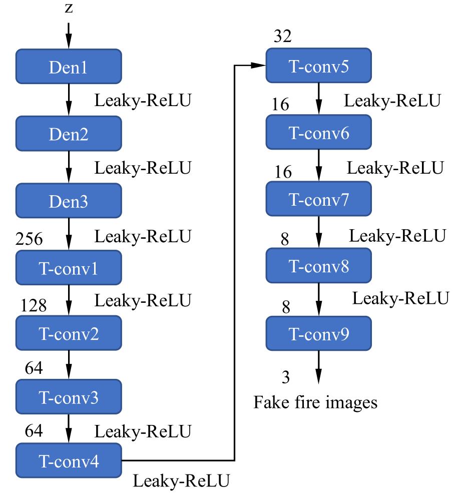

FIGURE 4. Structure of the TCNN generator in the DCGAN for fire detection.

In our overall approach to the DCGAN, we focus more on training the discriminator to achieve better classification accuracy than training a more complex generator to generate more realistic fake images. We obtain a relatively low-resolution generator that captures some characteristics of fire but without the full details of the flame. The generated fake fire images are used to prevent the discriminator from considering the images with partial flame characteristics as real fire images. This helps mitigate the model overfitting and the overestimation of fire probability when the training images are insufficient.

2) GENERATOR TRAINING REINFORCEMENT MECHANISM

In the standard DCGAN scheme, the generator is trained to generate highly realistic fake images [22]. Our approach is different for a number of reasons. First, we note that the training of a generator in a DCGAN is always a delicate matter, and it requires a careful balancing of art, heuristics, and trial and errors. We briefly share here our approach to this balancing act, which ultimately provided excellent results as discussed in section V.

In our DCGAN, we train the generator to produce partial characteristics of flames to mitigate the overfitting of the discriminator. This requirement for capturing some characteristics of flames increases the difficulty and the likelihood of instability in the TCNN training. Because this instability can "discourage" the generator from improving its performance, and it may lead to the production of useless or nonsense images, it is an important problem to tackle in all DCGAN training in general, and ours in particular. To solve or pre-empt this issue from occurring, we developed the following reinforcement mechanism for the training of the generator.

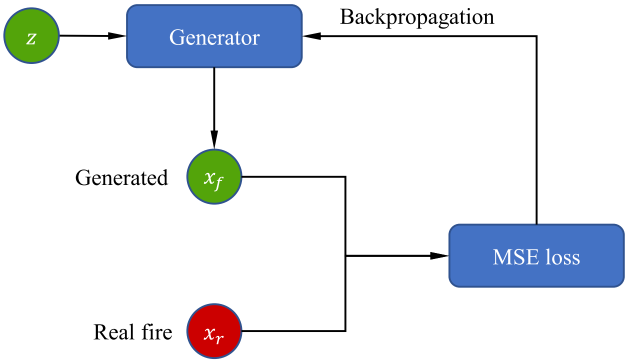

FIGURE 5. Generator training reinforcement mechanism.

Our approach is inspired by the imitation learning model in reinforcement learning [31], and it consists in supervised learning by real fire images to "teach" the generator to generate some characteristics of flames, as shown in Fig. 5.

First, we set the reinforcement mechanism parameter n and use $z = I$ to $n \times I$, where I is a 50 dimensions vector of all-ones, as the input of the generator network. In our computational experiments, we use n equals to the training size of real fire images. The generator converts the input z to generated images x_f. Second, we randomly select n real fire images from a training set, and we use them for the reinforcement mechanism in the training of the generator. We compare the generated and selected real fire images, and we calculate the pixel-based mean square error (MSE) loss. Finally, we conduct backpropagation of the MSE loss through the generator network with ADAM optimizer [38]. The limited model complexity of this TCNN generator constrains the generated fake image to capture only partial characteristics of the fire without much flame details, even with the assistance of the reinforcement mechanism. We use this reinforcement mechanism to initialize the generator and in each training iteration of the DCGAN. An additional, complementary approach to solving or pre-empting the likelihood of instability in the TCNN training is discussed next.

D. IMAGE NOISING

As noted previously, the generator in a DCGAN is difficult to train. The reinforcement mechanism just discussed is meant to facilitate this task. To further address or mitigate this problem, we apply an additional popular method that adds artificial noise to the real fire images. In the training process, we gradually decay the noise level for more resolved flame details to improve the accuracy and precision of the discriminator network. More specifically, we add pixel-based Gaussian random noise to the real fire images of the discriminator input, as shown in Eq. 5, where ϵ is the random noise, e is a random variable sampled from the standard Gaussian distribution, and m is the noise magnitude. The noise magnitude gradually decays during the training process, as shown in Eq. 6, where m_0 is the initial noise magnitude, t the training epoch number, and p the decay period. This magnitude of pixel-based random noise decays 100 times after every p epochs, and finally, it approaches zero. We set $m_0 = 10^{-2}$ in our computational experiments based on trial and error (higher values tended to degrade the discriminator training, and lower values lead to instabilities in the generator training in our computational experiments).

$$\epsilon = m \times e \tag{5}$$

$$m = m_0 \times 100^{-t/p} \tag{6}$$

E. DATASETS FOR TRAINING AND TESTING THE DCGAN

Jadon et al. [15] discussed the importance of high quality and diverse set of images for fire detection model training and testing. The authors pointed out that the widely used training datasets are not diverse enough, even though they are vast. They created a diverse dataset by recording fire and non-fire images from various challenging environments, for example, non-fire images with fire like objects in the background. They collected 1,124 fire and 1,301 non-fire images for training their FireNet visual fire detection model. They also collected 593 diverse fire and 278 non-fire images for testing. Figure 6 shows some sample images from this dataset. Readers can download this dataset from https://drive.google.com/drive/folders/1HznoBFEd6yjaLFlSmkUGARwCUzzG4whq [15]. For more details of the dataset, the reader is referred to [15].

In this work, we combine the training and testing fire images provided in ref. [15], which results in 1,717 fire images and 1,579 non-fire images overall [15]. We then randomly split this dataset into Set 1 (20%) and Set 2 (80%), the former for training and the latter for testing our DCGAN. We vary the size of the training set from 30 to 300 images in our computational experiments to assess the performance of the DCGAN and other visual fire detection methods when training images are limited (our primary objective).

IV. VISUAL FIRE DETECTION PERFORMANCE: RESULTS AND DISCUSSION

In this section, we discuss the performance results of our DCGAN, and we compare them with those obtained with other traditional supervised learning methods. We then compare the performance of our model with a current best-in-class network for visual fire detection, FireNet [15].

The computational experiments are conducted on a Pytorch platform on Windows 10 machine with AMD Ryzen 7 8 cores CPU processor and 31.9 GB system memory. The system is equipped with an NVIDIA GeForce RTX 2070 graphical card.

A. FIRE AND NON-FIRE IMAGES DISCRIMINATION ACCURACY

In our computational experiments, we examine the performance of visual fire detection methods when training images are limited. We conduct the training with datasets of varying sizes shown in Table 1.

The training and testing accuracy of different DL models, and with different training dataset sizes, are provided

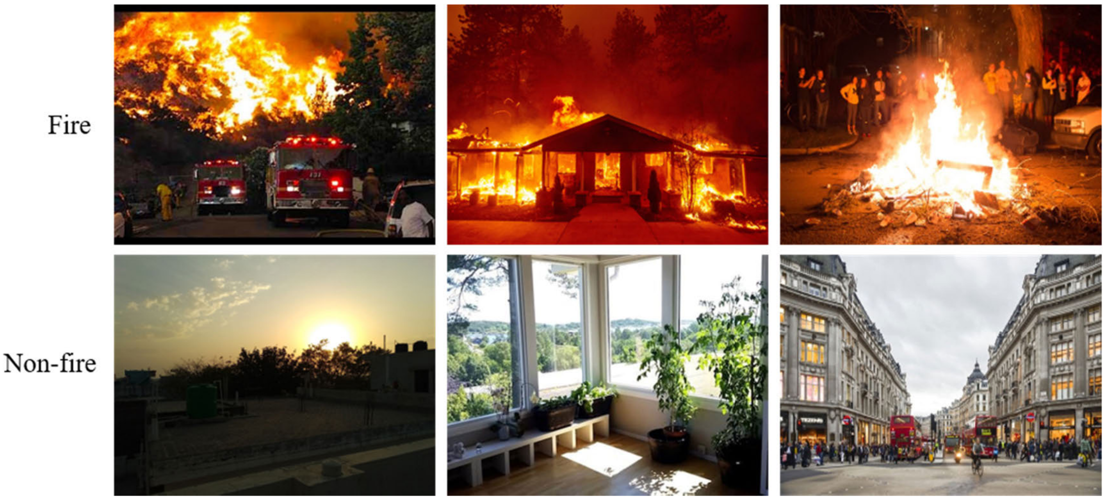

FIGURE 6. Samples of the fire and non-fire image dataset [15].

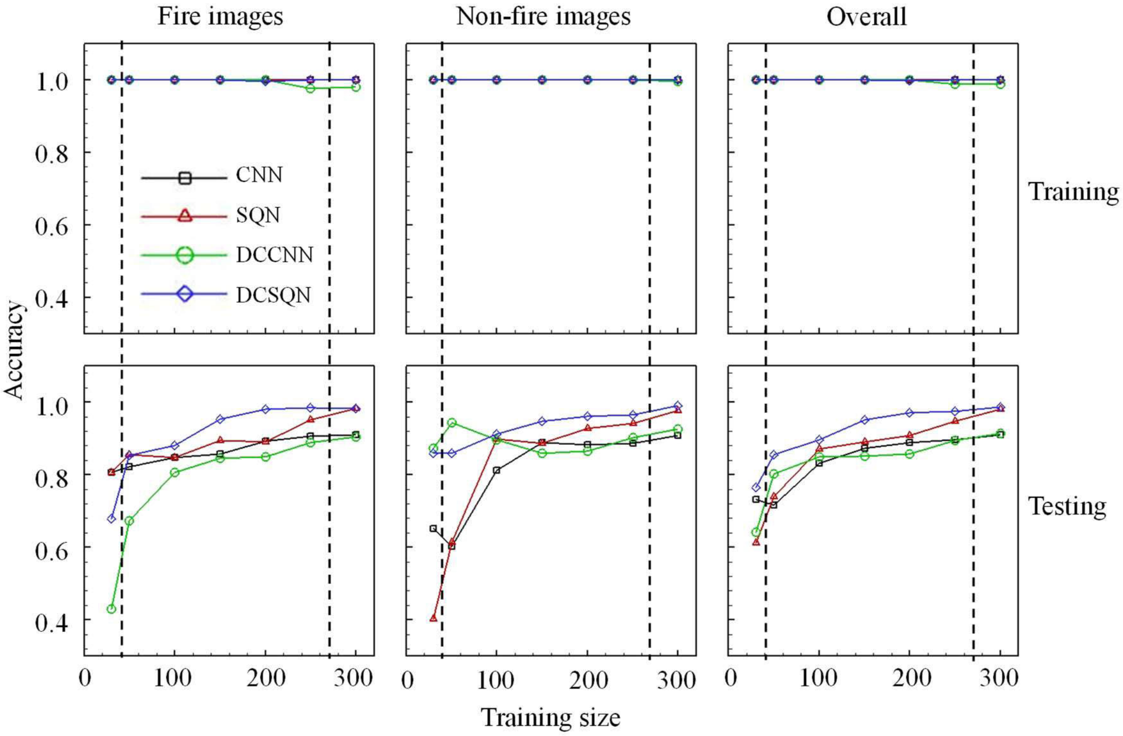

FIGURE 7. Training and testing accuracy of different DL models, and with different training dataset sizes.

in Fig. 7. We label the (standalone) supervised naïve CNN fire detection as CNN, the (standalone) supervised fire detection SqueezeNet as SQN, the DCGAN with the naïve CNN for discriminator as DCCNN, and the DCGAN with the SqueezeNet for a discriminator as DCSQN in Fig. 7.

We first note that different DL models have different testing performance characteristics in three different regions: when the training dataset is insufficient ($\leq$ 30 *images*); when the training dataset is small (50 to 250 images); and when the training dataset is adequate ($\geq$ 300 *images*). We label the boundaries of these regions with vertical dashed lines in Fig. 7 and Fig. 8. The most salient results are discussed next by region:

1) Within the insufficient (training dataset) region, all visual fire detection methods tested have poor or degraded performance. This exceedingly small training dataset causes significant overfitting, and as a result, while the training accuracy is (near) perfect, the testing accuracy (i.e., on images not seen during the training) is significantly degraded. The overall testing accuracy varies between 0.6 and 0.8 for all methods (bottom right panel in Fig. 7). The accuracy on fires and

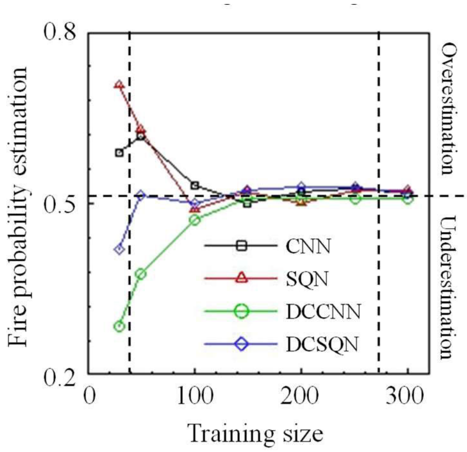

FIGURE 8. Testing fire probability estimation of different DL models, and with different training dataset sizes. The ground truth of fire image rate is 0.516 labeled by the horizontal dashed line.

TABLE 1. Number of training images for different computational experiments.

Training group	Image number (fire, non-fire, and fake)
1	30
2	50
3	100
4	150
5	200
6	250
7	300

non-fire images vary between the different detection methods: for example, the SQN performs better on fire images (accuracy 0.8) than on non-fire images (accuracy 0.4); whereas the DCCNN performs better on the non-fire images (accuracy 0.9) than on the fire images (accuracy 0.4). Figure 8 provides additional nuance to these observations. For example, the naïve CNN and SqueezeNet (CNN, SQN) overestimate the testing probability of fire, as seen in the insufficient (training dataset) region in Fig. 8 (left portion of the figure). In contrast, the DCGAN with a naïve CNN as a discriminator underestimates the probability of fire. The overall testing accuracy of the DCSQN is the best among all methods considered here within this insufficient (training dataset) region, as seen in the bottom right panel in Fig. 7. The fire probability underestimation of the DCSQN is less pronounced than that of DCCNN because of the increased model complexity of the SqueezeNet over the naïve CNN as discriminators. This increased model complexity enables the SqueezeNet and the DCSQN to resolve more details of the images provided. This higher model resolution is useful to recognize fires within a small range of pixels in the image. This feature of the fire detection SqueezeNet mitigates the underestimation problem when the training dataset is exceedingly small as in this insufficient (training dataset) region.

2) In the small (training dataset) region, the gap between the training and testing accuracy shrinks with increasing size of the training dataset. Overfitting is reduced when the training dataset increases from 30 to 250. This is expected for all detection methods. For the naïve CNN and the fire detection SqueezeNet with supervised learning, they both overestimate the testing fire probability when the training dataset includes fewer than 100 images. Although they both achieve good accuracy with fire image detection, their ability to accurately identify non-fire images remains rather poor. For the DCCNN, the fire image discrimination accuracy is degraded due to the insufficient model complexity and resolution. In contrast, the testing accuracy of the DCSQN with both fire and non-fires images (and overall accuracy) is the best among the methods considered here within this small (training dataset) region. The use of the DCSQN model effectively mitigates the model overfitting error, and the fire detection SqueezeNet discriminator has sufficient model complexity to accurately resolve the details of images to prevent the underestimation error.
3) In the adequate (training dataset) region, the overfitting error is significantly reduced for all four models. The firedetection SqueezeNet (SQN) and our DCSQN models achieve excellent accuracy (> 0.9) for both fire and non-fire images. The fire detection SqueezeNet is more complex than the naïve CNN, and this leads to its superior testing accuracy. Additionally, in this region, the advantages of DCSQN over the SQN are less significant than in the two previous regions because the size of the training dataset is no longer the driving factor for model accuracy.

As a side note, these results can serve as rules-of-thumb or heuristics for informing the selection of the training scheme and the discriminator model under the following circumstances: (1) if sufficient memory is available to meet the SqueezeNet requirement, and the training dataset is limited ($\leq$ 300 *images*), we recommend using the DCSQN; (2) if memory is significantly constrained, and the training dataset is plentiful, we recommend the naïve CNN with supervised learning.

Overall, Fig. 7 and 8 demonstrate that our DCGAN with the adapted fire detection SqueezeNet for a discriminator (DCSQN) achieves the best testing accuracy in all three regions of training dataset sizes (Insufficient, small, and adequate). In the next subsection, we compare the performance of our DCSQN with a current best-in-class network for visual fire detection, FireNet [15].

B. DCSQN VERSUS FIRENET: BEST-IN-CLASS DEEP LEARNING MODELS FOR VISUAL FIRE DETECTION

Here, we benchmark the performance of the DCSQN against FireNet in terms of the overall fire detection accuracy, false positives, false negatives, recall, precision, and F-score.

TABLE 2. Test Performance of DCSQN Versus FireNet.

	Our DCSQN (%)	**FireNet (%)**	FireNet in [15] (%)
Overall Accuracy	98.6	90.3	93.9
False Positives	0.95	5.50	1.95
False Negatives	2.70	4.25	4.13
Recall	98.2	91	94
Precision	99.1	89	97
F-score	98.6	90	95
Training Size (fire; non-fire)	(300; 300)	(300; 300)	(1,124; 1,301)
Testing Size (fire; non-fire)	(1,374; 1,263)	(1,374; 1,263)	(593; 278)

TABLE 3. The Memory Requirement of Naïve CNN.

Layer	Output size	Memory/image	Weights number
Input	$[64 \times 64 \times 3]$	49,152	N/A
Conv1	$[16 \times 16 \times 16]$	16,384	768
Conv2	$[4 \times 4 \times 64]$	4,096	16,384
Den1	$[1 \times 1 \times 50]$	200	51,200
Den2	$[1 \times 1 \times 1]$	4	50
Overall	N/A	$0.067Mb$	$0.065M$ ($0.260Mb$ on memory)

We compare the performance of both models with the same fire and non-fire image dataset. The results are provided in Table 2.

There are several ways of reading and interpreting these results. The most important are the following:

1) Within the shaded area (same size of the training and testing datasets), the DCSQN significantly outperforms FireNet on all performance metrics when trained with the same dataset size (300; 300). For example, overall accuracy and the F-score of the DCSQN are improved by about 8 percentage points, and precision by 10 percentage points over those of FireNet. False positives are reduced by about 4.5 percentage points, and false negatives by 1.5 percentage point. The DCSQN improves on FireNet's accuracy for both fire and non-fire classification;
2) When FireNet is trained with the original larger dataset (1,124; 1,301) [15], the DCSQN trained with the smaller dataset (300; 300) still outperforms its rival. While the false negatives with FireNet remain around 4% when the training dataset is reduced from (1,124; 1,301) to (300; 300), the false positives are significantly degraded (from $\sim 2\%$ to 5.5%). This indicates that overfitting occurs with FireNet when the training dataset size is reduced (more non-fires are falsely identified as fires).

Collectively, the results in this section indicate that our DCSQN effectively addresses the three types of errors in visual fire detection noted in the Introduction when the dataset size of training images is limited.

V. CONCLUSION

In this work, we developed a deep convolutional generative adversarial network (DCGAN) for highly accurate visual fire detection when training images are limited. Our model addressed three types of error in visual fire detection when training data is limited, namely model overfitting, fire probability overestimation, and fire probability underestimation errors. Our DCGAN includes a generator of fake fire images for self-supervised learning (SSL), and a discriminator to classify the images as fire, non-fire, and fake. We designed computational experiments with a diverse and high-quality fire detection image dataset to validate our model against other supervised learning approaches. We examined the accuracy of four models, the supervised naïve CNN, the supervised fire detection SQN, our DCGAN with a CNN for discriminator (DCCNN), and our DCGAN with a fire detection SqueezeNet for discriminator (DCSQN), with training size ranging from 30 to 300 images. The DCSQN achieved the best testing accuracy over all training dataset sizes. We then benchmarked the performance of our DCSQN against a best-in-class deep visual fire detection model, FireNet. The results of our computational experiments showed that: (1) the DCSQN significantly outperforms FireNet on all performance metrics (accuracy, false positive, false negative, precision, recall, and F-score) when trained with the same dataset size; (2) the DCSQN model effectively mitigates overfitting when the training dataset is limited; (3) and more generally, the results indicate that the DCSQN effectively addresses the three types of errors in visual fire detection when training images are limited.

This work should be considered in light of its limitations, and these constitute fruitful venues for future work. First, we only considered a naïve CNN and a fire detection SqueezeNet for the DCGAN discriminator. There is a broader range of options for the discriminator, such as AlexNet [16], GoogleNet [17], and ResNet [32]. In future work, we propose to examine the performance of our DCGAN with these deep learning networks for discriminators. Second, we used DCGAN to mitigate the overfitting error when the training dataset size is limited. We propose to leverage other machine learning generative models, such as variational autoencoder (VAE) [42] to explore further mitigation of the overfitting problem. Third, we trained and tested our model with 2D RGB colored pictures. In real-world applications, fire detection video can be available from surveillance systems. We propose to upgrade our system with 3D convolutional layer to enable it to handle visual fire detection with video. Fourth, another important future work of our proposed visual fire detection is a detailed training computational cost analysis. We plan to conduct a trade-off analysis for accuracy versus training complexity for different types of generator and discriminator in DCGAN. Finally, we propose to adapt our visual fire detection system for aerospace applications. In particular, we plan to create a high-quality fire detection image dataset in micro-gravity condition and apply our models to this situation.

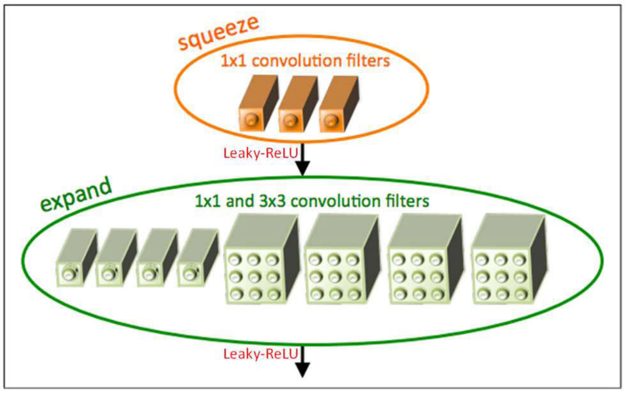

FIGURE 9. Organization of convolution filters in the fire model. In this example, $s_{1\times1} = 3$, $e_{1\times1} = 4$, and $e_{3\times3} = 4$. We illustrate the convolution filters but not the activations [40].

TABLE 4. The Memory Requirement of Fire Detection SqueezeNet.

Layer	Output size	Memory/image	Weights number
Input	[64 × 64 × 3]	49,152	N/A
Conv1	[30× 30 × 64]	230,400	1,728
Pool	[15× 15 × 64]	57,600	0
Fire2	[15× 15 × 128]	129,600	11,264
Fore3	[15× 15 × 128]	129,600	12,288
Pool	[7 × 7 × 128]	25,088	0
Fire4	[7× 7 × 256]	56,448	45,056
Fire5	[7× 7 × 256]	56,448	49,152
Pool	[3 × 3 × 256]	9,216	0
Fire6	[3 × 3 × 384]	15,552	104,448
Fire7	[3 × 3 × 384]	15,552	110,592
Fire8	[3 × 3 × 512]	20,736	188,416
Fire9	[3 × 3 × 512]	20,736	196,608
Conv10	[3 × 3 × 1024]	36,864	524,288
Pool	[1 × 1 × 1024]	4,096	0
Den1	[1 × 1 × 128]	512	131,072
Den2	[1 × 1 × 1]	4	1
Overall	N/A	0.82Mb	1.31M (5.24Mb on memory)

TABLE 5. The Memory Requirement of TCNN Generator.

Layer	Output size	Weight detail	Weight number
Input	[1 × 1 × 50]	N/A	N/A
Den1	[1 × 1 × 125]	(50, 125)	6,250
Den2	[1 × 1 × 250]	(125, 250)	31,500
Den3	[2× 2 × 256]	(250, 1,024)	256,000
T-conv1	[5× 5 × 128]	(256, 128, 4, 1, 0)	524,288
T-conv2	[11 × 11 × 64]	(128, 64, 3, 2, 0)	73,728
T-conv3	[13× 13 × 64]	(64, 64, 3, 1, 0)	36,864
T-conv4	[27× 27 × 32]	(64, 32, 3, 2, 0)	18,432
T-conv5	[29× 29 × 16]	(32, 16, 3, 1, 0)	4,608
T-conv6	[59× 59 × 16]	(16, 16, 3, 2, 0)	2,304
T-conv7	[61× 61 × 8]	(16, 8, 3, 1, 0)	1,152
T-conv8	[63× 63 × 8]	(8, 8, 3, 1, 0)	576
T-conv9	[64× 64 × 3]	(8, 3, 2, 1, 0)	96
Overall	N/A	N/A	0.91M

APPENDIX

A. WEIGHT PARAMETERS OF NAÏVE CNN

Here, we introduce the details of the output size, memory requirement per image, and the number of the weight of the naïve CNN discriminator of fire detection as shown in Table 3.

B. FIRE MODEL IN THE FIRE DETECTION SqueezeNet

We introduced the fire model as shown in Fig. 9 since it is extensively used in the SqueezeNet, where $s_{1\times1}$, $e_{1\times1}$, and $e_{3\times3}$ stand for the number of squeeze layers, the number of 1 × 1 expand layer, and the number of 3 × 3 expand layers, respectively.

In our fire detection SqueezeNet, we set $s_{1\times1}$, $e_{1\times1}$, and $e_{3\times3}$ as 1. We switch the activation function of the fire model from ReLU in the original model to LeakyReLU for more nonlinearity and preventing vanishing gradient problem for negative input.

C. WEIGHT PARAMETERS OF FIRE DETECTION SqueezeNet

Here, we introduce the details of the weight of fire detection SqueezeNet as shown in Table 4.

D. WEIGHT PARAMETERS OF TCNN GENERATOR

Here, we introduce the weight parameters of the generator TCNN network used in the SSL DCGAN framework as shown in Table 5.

ACKNOWLEDGMENT

The authors are indebted to the incredibly talented Ashling Devins for creating the graphical abstract.

REFERENCES

[1] C. Chaoxia, W. Shang, and F. Zhang, "Information-guided flame detection based on faster R-CNN," *IEEE Access*, vol. 8, pp. 58923–58932, 2020.

[2] X. Huang and L. Du, "Fire detection and recognition optimization based on virtual reality video image," *IEEE Access*, vol. 8, pp. 77951–77961, 2020.

[3] J.-H. Kim and J. Lee, "Fire sprite animation using fire-flake texture and artificial motion blur," *IEEE Access*, vol. 7, pp. 110002–110011, 2019.

[4] G. Roque and V. S. Padilla, "LPWAN based IoT surveillance system for outdoor fire detection," *IEEE Access*, vol. 8, pp. 114900–114909, 2020.

[5] Z. Wu, Y. Hu, J. X. Wen, F. Zhou, and X. Ye, "A review for solar panel fire accident prevention in large-scale PV applications," *IEEE Access*, vol. 8, pp. 132466–132480, 2020.

[6] Y. Xie, J. Zhu, Y. Cao, Y. Zhang, D. Feng, Y. Zhang, and M. Chen, "Efficient video fire detection exploiting motion-flicker-based dynamic features and deep static features," *IEEE Access*, vol. 8, pp. 81904–81917, 2020.

[7] G. Xu, Q. Zhang, D. Liu, G. Lin, J. Wang, and Y. Zhang, "Adversarial adaptation from synthesis to reality in fast detector for smoke detection," *IEEE Access*, vol. 7, pp. 29471–29483, 2019.

[8] H. Yang, H. Jang, T. Kim, and B. Lee, "Non-temporal lightweight fire detection network for intelligent surveillance systems," *IEEE Access*, vol. 7, pp. 169257–169266, 2019.

[9] Z. Liu, and A. K. Kim, "Review of recent developments in fire detection technologies," *J. Fire Protection Eng.*, vol. 13, no. 2, pp. 129–151, May 2003.

[10] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.

[11] R. Szeliski, *Computer Vision: Algorithms and Applications*. London, U.K.: Springer, 2010.

[12] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Comput. Intell. Neurosci.*, vol. 2018, pp. 1–13, Feb. 2018.

[13] D. Shen, X. Chen, M. Nguyen, and W. Q. Yan, "Flame detection using deep learning," in *Proc. 4th Int. Conf. Control, Automat. Robot. (ICCAR)*, Apr. 2018, pp. 416–420.

[14] W. Mao, W. Wang, Z. Dou, and Y. Li, "Fire recognition based on multi-channel convolutional neural network," *Fire Technol.*, vol. 54, no. 2, pp. 531–554, Mar. 2018.

[15] A. Jadon, M. Omama, A. Varshney, M. S. Ansari, and R. Sharma, "FireNet: A specialized lightweight fire & smoke detection model for real-time IoT applications," 2019, *arXiv:1905.11922*. [Online]. Available: http://arxiv.org/abs/1905.11922

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[17] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[18] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.

[19] V. Gudivada, A. Apon, and J. Ding, "Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations," *Int. J. Adv. Softw.*, vol. 10, no. 1, pp. 1–20, 2017.

[20] E. Breck, N. Polyzotis, S. Roy, S. E. Whang, and M. Zinkevich, "Data infrastructure for machine learning," in *Proc. SysML Conf.*, 2018, pp. 1–3.

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[22] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*. [Online]. Available: http://arxiv.org/abs/1511.06434

[23] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," 2018, *arXiv:1809.11096*. [Online]. Available: http://arxiv.org/abs/1809.11096

[24] M. Ravanbakhsh, E. Sangineto, M. Nabi, and N. Sebe, "Training adversarial discriminators for cross-channel abnormal event detection in crowds," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 1896–1904.

[25] T.-H. Chen, P.-H. Wu, and Y.-C. Chiou, "An early fire-detection method based on image processing," in *Proc. Int. Conf. Image Process. (ICIP)*, vol. 4, 2004, pp. 1707–1710.

[26] T. Çelik, H. Özkaramanlı, and H. Demirel, "Fire and smoke detection without sensors: Image processing based approach," in *Proc. 15th Eur. Signal Process. Conf.*, 2007, pp. 1794–1798.

[27] A. Rafiee, R. Dianat, M. Jamshidi, R. Tavakoli, and S. Abbaspour, "Fire and smoke detection using wavelet analysis and disorder characteristics," in *Proc. 3rd Int. Conf. Comput. Res. Develop.*, Mar. 2011, pp. 262–265.

[28] S. Rinsurongkawong, M. Ekpanyapong, and M. N. Dailey, "Fire detection for early fire alarm based on optical flow video processing," in *Proc. 9th Int. Conf. Electr. Eng./Electron., Comput., Telecommun. Inf. Technol.*, May 2012, pp. 1–4.

[29] Q. Zhang, J. Xu, L. Xu, and H. Guo, "Deep convolutional neural networks for forest fire detection," in *Proc. Int. Forum Manage., Educ. Inf. Technol. Appl.*, 2016, pp. 1–8.

[30] J. Sharma, O.-C. Granmo, M. Goodwin, and J. T. Fidje, "Deep convolutional neural networks for fire detection in images," in *Proc. Int. Conf. Eng. Appl. Neural Netw.*, 2017, pp. 183–193.

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[33] K. Muhammad, J. Ahmad, and S. W. Baik, "Early fire detection using convolutional neural networks during surveillance for effective disaster management," *Neurocomputing*, vol. 288, pp. 30–42, May 2018.

[34] K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang, and S. W. Baik, "Efficient deep CNN-based fire detection and localization in video surveillance applications," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 49, no. 7, pp. 1419–1434, Jul. 2019.

[35] K. Muhammad, J. Ahmad, I. Mehmood, S. Rho, and S. W. Baik, "Convolutional neural networks based fire detection in surveillance videos," *IEEE Access*, vol. 6, pp. 18174–18183, 2018.

[36] K. Muhammad, S. Khan, M. Elhoseny, S. Hassan Ahmed, and S. Wook Baik, "Efficient fire detection for uncertain surveillance environment," *IEEE Trans. Ind. Informat.*, vol. 15, no. 5, pp. 3113–3122, May 2019.

[37] P. Bojanowski, A. Joulin, D. Lopez-Paz, and A. Szlam, "Optimizing the latent space of generative networks," 2017, *arXiv:1707.05776*. [Online]. Available: http://arxiv.org/abs/1707.05776

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[39] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, 2013, vol. 30, no. 1, p. 3.

[40] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size," 2016, *arXiv:1602.07360*. [Online]. Available: http://arxiv.org/abs/1602.07360

[41] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *Nat. Sci. Rev.*, vol. 5, no. 1, pp. 44–53, Jan. 2018.

[42] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," 2019, *arXiv:1906.02691*. [Online]. Available: http://arxiv.org/abs/1906.02691

ZHAOYI XU received the B.S. degree in aerospace engineering from Northwestern Polytechnical University, China, in 2015, and the M.S. degree in aerospace engineering from the Georgia Institute of Technology, where he is currently pursuing the Ph.D. degree with the Guggenheim School of Aerospace Engineering. His research focuses on machine learning (ML) in engineering applications, with an emphasis on ML for reliability and system safety and design optimization. He is also proficient in computational fluid dynamics. His work on supercritical airfoil optimization, using the genetic algorithm with ML surrogate was recently published in the AIAA journal. He received an outstanding undergraduate thesis award based on his work on transonic unsteady flow from Northwestern Polytechnical University.

YANJIE GUO received the B.S. degree in aeronautical engineering from the Rensselaer Polytechnic Institute and the M.S. degree in attitude determination and control system for a CubeSat mission. He is currently pursuing the Ph.D. degree in aerospace engineering with the Georgia Institute of Technology. His current focus is on active sensing systems that exploit available actuators to improve system state estimation.

JOSEPH HOMER SALEH (Senior Member, IEEE) received the master's degree from Harvard and the Ph.D. degree from MIT in 2002. He served as the Executive Director of the Ford-MIT research partnership. He is an Associate Professor with the School of Aerospace Engineering, Georgia Institute of Technology. He has authored two books and 78 journal articles. His research cover three themes—spacecraft reliability and multi-state analysis, analytical systems engineering, with a focus on its temporal dimension, and multidisciplinary accident causation and system safety. He is an Associate Fellow of AIAA. He has received several awards at Georgia Institute of Technology, including the Outstanding Faculty Award, the Lockheed Dean's Excellence in Teaching Award, and the Class of 1940 Roanne Beard Outstanding Teaching Award. He served as an Associate Editor of *Reliability Engineering & System Safety* and the IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS.