Deep Dual-Channel Neural Network for Image-Based Smoke Detection

Ke Gu, *Member, IEEE,* Zhifang Xia, Junfei Qiao, *Member, IEEE,* and Weisi Lin, *Fellow, IEEE*

Abstract*—Smoke detection plays an important role in industrial safety warning systems and fire prevention. Due to the complicated changes in the shape, texture and colour of smoke, identifying the smoke from a given image still remains a substantial challenge, and this has accordingly aroused a considerable amount of research attention recently. To address the problem, we devise a new deep dual-channel neural network (DCNN) for smoke detection. In contrast to popular deep convolutional networks, e.g., Alex-Net, VGG-Net, Res-Net, and Dense-Net, and the DNCNN that is specifically devoted to detecting smoke, our proposed end-to-end network is mainly composed of dual channels of deep subnetworks. In the first subnetwork, we sequentially connect multiple convolutional layers and max-pooling layers. Then, we selectively append the batch normalization layer to each convolutional layer for over-fitting reduction and training acceleration. The first subnetwork is shown to be good at extracting the detail information of smoke, such as texture. In the second subnetwork, in addition to the convolutional, batch normalization and max-pooling layers, we further introduce two important components. One is the skip connection for avoiding the vanishing gradient and improving the feature propagation. The other is the global average pooling for reducing the number of parameters and mitigating the over-fitting issue. The second subnetwork can capture the base information of smoke, such as contours. We finally deploy a concatenation operation to combine the aforementioned two deep subnetworks to complement each other. Based on the augmented data obtained by rotating the training images, our proposed DCNN can promptly and stably converge to the perfect performance. Experimental results conducted on the publicly available smoke detection database verify that the proposed DCNN has attained a very high detection rate that exceeds 99.5% on average, superior to state-of-the-art relevant competitors. Furthermore, our DCNN only employs approximately one-third of the parameters needed by the comparatively tested deep neural networks. The source code of DCNN will be released at https://kegu.netlify.com/.

***Index Terms*—Smoke detection, deep learning, convolutional network, dual-channel network, classification**

I. Introduction

It is an urgent task to promptly and effectively detect the smoke for industrial automation and fire safety warning systems, such as torch black smoke detection in petrochemical fields and forest fire warnings. Existing approaches for torch black smoke detection and pyrotechnic detection mainly depend on manual observation or sensors. Because of limited human resources, popular manual observation-based methods cannot be used to monitor smoke rapidly and validly in the long term, particularly given intermittent interruptions or distractions. On the other hand, smoke sensors that are based on smoke particle sampling or relative humidity sampling are very likely to cause a severe time delay; moreover, they cannot simultaneously and completely cover the detection areas when applied to detecting smoke because of the influences of environmental variations. Overall, existing smoke detection methods meet with difficulties in satisfying the requirements of today's industrial processes and safety warnings.

During the last several years, image-based smoke detection methods have been broadly explored to solve such problems. In [1], motivated by an observation that smoke affects the high-frequency information of an image's background area, Toreyin *et al.* proposed a method that applied the spatial wavelet transform to measure the high-frequency energy loss of the scene for smoke detection. In [2], by introducing wavelet decompositions and a support vector machine (SVM), Gubbi *et al.* developed a smoke characterization-based technique to identify smoke from a video sequence. In [3], Yuan devised a fast accumulative motion orientation model based on an integral image for video smoke detection. In [4], Yuan devised a video-based smoke detection approach using histograms of the local binary pattern (LBP) and local binary pattern variance (LBPV). In [5], Yuan put forward a smoke detection method by learning shape invariant features on multi-scale partitions with AdaBoost. Note that this is the first time that the multi-scale strategy was used to amend the performance of smoke detection by a sizable margin. In [6], Yuan *et al.* proposed a classification algorithm for smoke images based on a high-order local ternary pattern (LTP) with local preservation projection, and this algorithm has led to a noticeable performance gain compared with the existing relevant methods. In [7], Yuan *et al.* incorporated the LBP, kernel principal component analysis (KPCA) and Gaussian process regression for detecting smoke. For the readers' convenience, we summarize the image-based smoke detection algorithms illustrated above in Table I. It is not difficult to determine that most of the existing smoke detection technologies are only based on the analysis

This work was supported in part by the National Science Foundation of China under Grant 61703009, the Young Elite Scientist Sponsorship Program by China Association for Science and Technology under Grant 2017QN-RC001, the Young Top-Notch Talents Team Program of Beijing Excellent Talents Funding under Grant 2017000026833ZK40, and the National Science and Technology Major Project of the Ministry of Science and Technology of China under Grant 2018ZX07111005. (Corresponding author: Ke Gu.)

K. Gu and J. Qiao are with the Beijing Advanced Innovation Center for Future Internet Technology, Beijing Key Laboratory of Computational Intelligence and Intelligent System, Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China (email: guke.doctor@gmail.com; junfeiq@bjut.edu.cn).

Z. Xia is with Beijing Advanced Innovation Center for Future Internet Technology, Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China, and also with the State Information Center of P.R.China, Beijing, China (email: spidergirl21@163.com).

W. Lin is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore, 639798 (email: wslin@ntu.edu.sg).

TABLE I: Summary of image-based smoke detection methods.

Reference	Description
[1]	Spatial wavelet transform + High-frequency energy loss
[4]	Multi-scale + LBP + LBPV + Histograms of pyramids
[6]	High-order LTP + Local preservation projection
[7]	LBP + KPCA + Gaussian process regression
[22]	CNN + Batch normalization + Full connection

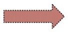
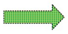
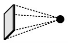
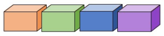
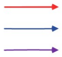
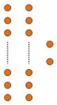
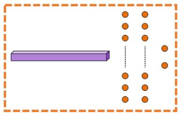
and synthesis of textural information.

To date, the majority of existing smoke detection models have been developed by using hand-crafted features for identifying smoke. Nonetheless, such studies might encounter a bottleneck because the manually extracted image features are still of insufficient ability to characterize the complicated variations in the smoke images [1]-[7]. In contrast, deep learning is very possibly a better solution for smoke detection since recent years have seen an unmatched performance attained by deep learning, particularly in pattern recognition applications such as image recognition and image classification [8]-[9]. Multiple prevailing deep convolutional networks participated in the well-known ILSVRC (ImageNet Large Scale Visual Recognition Challenge [10]) competition classification project and have made significant breakthroughs. For example, some typical networks, such as Alex-Net [11], ZF-Net [12], VGG-Net [13], GoogLe-Net [14], Xception [15], and Res-Net [16], illustrate that neural networks have become increasingly deeper, from a few layers to more than one hundred layers during the past several years. Moreover, deep learning has also been successfully applied to numerous multimedia applications recently [17]-[21].

Despite the great achievements obtained by deep learning, very limited effort has been devoted to the smoke detection task. To the best of our knowledge, only one deep learning-based model exists for detecting smoke. More specifically, in [22], Yin *et al.* developed a deep normalized convolutional neural network (DNCNN) for smoke detection from images. The DNCNN imposes two main improvements on a sequential convolutional neural network. One is that the DNCNN embeds the batch normalization (BN) [23] into the convolutional layer for alleviating the gradient dispersion and over-fitting problems when training the network. The other is that the DNCNN adopts the data augmentation technology to settle the problem of positive and negative sample imbalance and insufficiency that occurs in the training samples. These two improvements have promoted the performance of DNCNN to a high level beyond 97%[1]. However, such detection accuracy is unable to meet our requirements because the poisonous smoke emitted into the air due to the imperfect smoke detection technology is harmful to safeguarding life and the environment; in other words, achieving a perfect detection performance of 100% is our unique and never-ending pursuit, similar to the goal for autonomous vehicles [24].

To further enhance the performance and robustness of smoke detection, in this paper we put forward a novel deep dual-channel neural network, dubbed DCNN. For a given image, we first divide the input image into patches, and then separately identify each patch based on the proposed DCNN. By such processes, we can convert the task of detecting smoke into a two-category classification problem, i.e., smoke patch and smoke-free version. The proposed end-to-end DCNN is mainly established by employing dual channels of deep subnetworks. Multiple convolutional layers and max-pooling layers are sequentially connected to generate the first channel of the subnetwork. To alleviate the over-fitting problem and accelerate the training process, we introduce the BN operations. We selectively append the BN layer to each of the last four convolutional layers since it was found that the BN layer is very likely to restrict the freedom of extracted features [25]-[26]. The first channel of the subnetwork is shown to be good at extracting the detail information of smoke.

TABLE II: Important symbols and implications.

Symbol	Implication
	Convolution
	Normalization and convolution
	Transmission
	Max-pooling
⊕	Concatenation
	Global average pooling
	Feature Map
	Concatenated feature maps
● ●	Onehot type output
	Feature transmission
	Full connection
	Removed structure

Next, the second channel of the subnetwork is constructed by incorporating two new significant components with the convolutional, BN, and max-pooling layers. One component is the skip connection, which contributes to preventing the gradient vanishing and enhancing the feature propagation. The other component is the global average pooling, which is beneficial in decreasing the number of parameters and mitigating the problem of over-fitting. It was found that the second channel of the subnetwork is capable of capturing the base information of smoke. Eventually, we construct the DCNN by introducing a concatenation operation to fuse the features extracted using the aforementioned two deep subnetworks. By complementing each other, the concatenation operation can condense the extracted features and enable their stronger representation ability. Our DCNN is learned based on the augmented training data, which are generated by rotating

[1] We also include DNCNN as an image-based smoke detector in Table I.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2019.2929009, IEEE Transactions on Multimedia

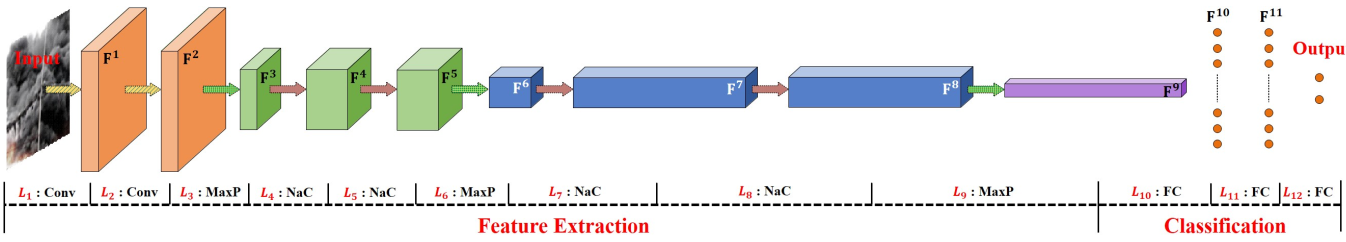


Fig. 1: The basic architecture of SBNN, including six convolutional layers, three max-pooling layers, four normalization layers, and three fully-connected layers. The implications of symbols can be found in Table II. ‘Conv’, ‘MaxP’, ‘NaC’, and ‘FC’ stand for convolution, max-pooling, normalization and convolution, and full connection operations, respectively.

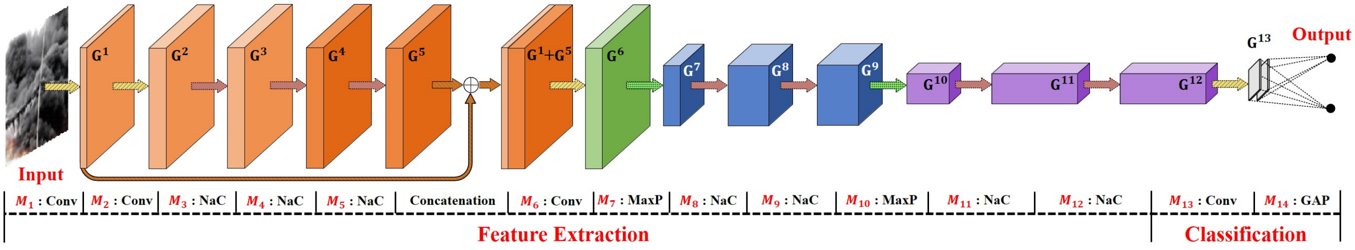


Fig. 2: The basic architecture of SCNN, including eleven convolutional layers, three max-pooling layers, seven normalization layers, and one global average pooling layer. The implications of symbols can be found in Table II. ‘Conv’, ‘MaxP’, ‘NaC’, and ‘GAP’ stand for convolution, max-pooling, normalization and convolution, and global average pooling operations, respectively.

training image patches. Experiments demonstrate that our proposed DCNN leads to noticeable improvement by boosting the performance of smoke detection and lowering the number of model parameters compared with recently proposed deep networks including Alex-Net [11], ZF-Net [12], VGG-Net [13], GoogLe-Net [14], Xception [15], Res-Net [16], Dense-Net [27], and DNCNN [22].

We highlight the main novelty and contribution of this work compared with existing image-based smoke detection methods as follows. First, from the viewpoint of the design principle, this paper is the first work that integrates the low-level local textural characteristics and the high-level global contour information for detecting smoke from images. Further, by using image decomposition, we straightforwardly illustrate the necessity of fusing the aforesaid two components in smoke detection. Second, from the aspect of network structure, this paper is the first work that designs a dual-channel deep neural network for effective and efficient smoke detection. Specifically, to improve the detection effectiveness, we insert the skip connection into a sequential convolutional network for capturing contour information and introduce the feature fusion layer for comprehensively synthesizing textural characteristics and contour information. We replace the fully-connected layers with a simple global average pooling to largely reduce the number of model parameters, and thus enhance the efficiency during training and testing. Last, from the perspective of detection performance, our proposed DCNN attains very high accuracy beyond 99.5% on average, resulting in a relative performance gain of approximately 1% compared with the second-rank model.

The structure of this paper is outlined as follows. Section II illustrates the details concerning the network architecture, parameter settings, etc. In Section III, the superiority of our proposed DCNN is validated by comparison with state-of-the-art deep learning models and recently proposed smoke detection methods. Furthermore, we specifically discuss how the two deep subnetworks complement each other. Section IV summarizes the whole paper.

II. Proposed Deep Neural Network

Recent years have seen a growing number of multimedia technologies that were applied for resolving environmental problems, e.g. smoke detection [22], PM2.5 monitoring [28]-[29], and air quality forecast [30]-[32]. The proposed DCNN, particularly devised for detecting smoke, will be described in detail. As illustrated earlier, our DCNN is composed of dual channels of deep subnetworks. We arrange the whole Section II by first introducing the two deep subnetworks, namely, the selective-based batch normalization network (SBNN) and skip connection-based neural network (SCNN). Then, we present how to reasonably combine the above subnetworks to build the DCNN. Last, we illustrate the details of network training. For the convenience of readers, we present important symbols and the associated implications in Table II.

A. SBNN’s Architecture

Built on the convolutional neural network and inspired by the recently proposed DNCNN, we establish the SBNN, as given in Fig. 1. First, we sequentially connect six convolutional layers and three max-pooling layers for feature extraction. Convolution is a commonly used operation for capturing local information to generate a tensor of outputs. The r-th convolutional layer consists of n^r feature maps, denoted as $\mathbf{F}_p^r(p = 1, 2, \cdots, n^r)$. Each feature map in the $(r-1)$-th convolutional layer, $\mathbf{F}_q^{r-1}(q = 1, 2, \cdot\cdot\cdot, n^{r-1})$, is convolved

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2019.2929009, IEEE Transactions on Multimedia

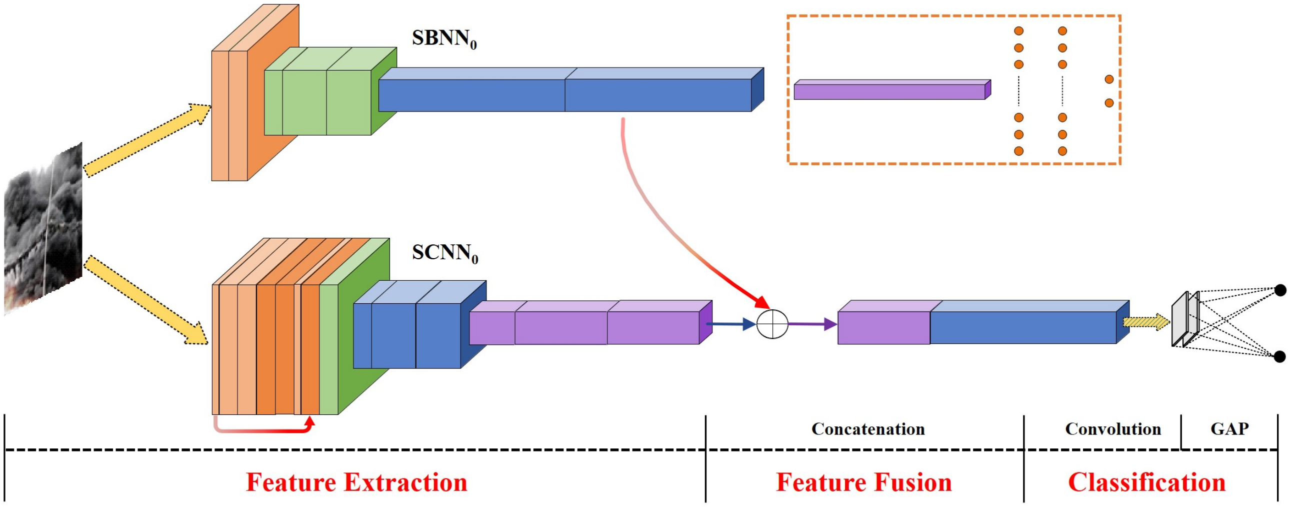


Fig. 3: The basic architecture of DCNN, including the dual channels of deep subnetworks for feature extraction, a concatenation operation for feature fusion, as well as a convolution layer and a global pooling layer for classification. ‘GAP’ stands for the global average pooling operation. For the sake of clarity, we remove the operation symbols from the original subnetworks and only keep the feature blocks. The top channel is the SBNN$_0$ and the bottom channel is the SCNN$_0$.

with the filter $\mathbf{W}_{qp}^r$ and added to the bias $\mathbf{b}_p^r$, followed by a non-linear activation function $\Lambda(\cdot)$:

$$\mathbf{F}_p^r = \Lambda\left(\sum_{q=1}^{n^{r-1}} \mathbf{F}_q^{r-1} \star \mathbf{W}_{qp}^r + \mathbf{b}_p^r\right), \quad p = 1, 2, \cdots, n^r \quad (1)$$

where “$\star$” indicates the convolution operation. The activation function leverages the rectified linear unit (ReLu) function since it is more consistent with the characteristics of biological neurons [33]. The max-pooling targets to learn biologically plausible features by activating the local maximum response. The main merits of max-pooing are the invariance of translation, rotation and scale, as well as the reduction in the number of network parameters. In our implementation, we select the maximum activation value over a small pooling region.

Then, we selectively append the BN layer to each of the last four convolutional layers. When training deep convolutional neural networks, the most commonly used optimization method is the mini-batch stochastic gradient descent (SGD). However, the internal covariate shift, namely, the variations of internal input distributions during training, usually reduces the training efficiency seriously. The BN was proposed to resolve such limitations of convolutional layers [23]. By transforming internal inputs with a scale and shift step prior to non-linear activation, the BN can validly speed up the network training and prevent parameter over-fitting. More specifically, based on the mini-batch mean and variance, each feature f_j is normalized as follows:

$$\hat{f}_j = \frac{f_j - \overline{\mu}_j}{\sqrt{\sigma_j^2 + \epsilon}} \quad (2)$$

where $\overline{\mu}_j = \frac{1}{n}\sum_{i=1}^n f_{j,i}$ and $\sigma_j^2 = \frac{1}{n}\sum_{i=1}^n (f_{j,i} - \overline{\mu}_j)^2$ are respectively the mini-batch mean and variance, with n being the size of a mini-batch and $f_{j,i}$ being the j-th feature of the i-th sample in the mini-batch. ϵ is a fixed small positive number used for promoting numerical stability. However, normalizing the input features might decrease their representation capability. Two free parameters α and β are thus introduced to settle such problems by transforming normalized features via a scale and shift step in the BN:

$$B(f_j) = \alpha \hat{f}_j + \beta. \quad (3)$$

The reason why we do not append the BN layers to the first two convolutional layers originated from insights in a recent study, which implied that the BN makes the extracted features freely constrained [25]. Therefore, we remove the BN in the first and second convolutional layers to facilitate better protection of the smoke characteristics of the image patch.

We extract the features from a given image patch based on the above-mentioned operations. Then, we append three fully-connected layers at the end of the last max-pooling layer L_9. Sufficient experiments indicated that the full connection operation can easily cause over-fitting since the fully-connected layers often contain a substantial number of learnable parameters. A typical solution to overcome such a problem is to introduce the dropout technique [34]. In our implementation, as shown in Fig. 1, the first fully-connected layer L_{10} receives all feature maps of $\mathbf{F}^9$ as the input neurons to yield the feature maps of $\mathbf{F}^{10}$:

$$\mathbf{F}^{10} = \mathbf{W}^{10} \star \mathbf{F}^9 + \mathbf{b}^{10}. \quad (4)$$

Likewise, we can derive the output of the second and third fully-connected layers, namely, the feature maps $\mathbf{F}^{11}$ and $\mathbf{F}^{12}$. The output layer L_{12} consisting of two neurons produces two classes of probabilities, $\hat{\mathbf{x}} = [\hat{x}_1, \hat{x}_2]^T$. The output probability of the u-th neuron for the u-th class is calculated using the softmax function:

$$\hat{x}_u = \frac{\exp(\mathbf{F}_u^{12})}{\sum_{i=1}^2 \exp(\mathbf{F}_i^{12})}, \quad u = 1, 2. \quad (5)$$

Furthermore, it is noteworthy that, compared with the DNCNN, our proposed SBNN has two dominant improvements:

1) a more compact structure, and 2) a selective-based batch normalization.

B. SCNN's Architecture

On the basis of SBNN, the SCNN in the second channel further introduces the skip connection and global average pooling, as shown in Fig. 2. First, we connect eleven convolutional layers, seven BN layers, and two max-pooling layers to construct a sequential network for extracting features. Note that, akin to the SBNN, the BN layer is not appended to the former two convolutional layers for better feature protection. The kernel size of the first convolution layer M_1 is assigned as nine for extracting richer image features without seriously increasing network parameters and that of the second convolution layer M_2 is set as one for the purpose of merging the features extracted from the front layer without changing the features' structure. Further, the BN layer is also not appended to the sixth and eleventh convolutional layers (i.e., M_6 and M_{13}) since these two layers are used to lower the dimensionality by properly fusing the feature maps. Apart from the operations mentioned above, the SCNN hops and connects the first feature map $\mathbf{G}_1$ to the fifth feature map $\mathbf{G}_5$ (before the max-pooling) through the skip connection. These two feature maps are merged together with a concatenation operation followed by a convolutional layer with its kernel size of one:

$$\mathbf{G}^6 = \max(0, \mathbf{V}^6 \star [\mathbf{G}^1, \mathbf{G}^5] + \hat{\mathbf{b}}^6) \tag{6}$$

where the operation $[\mathbf{G}^1, \mathbf{G}^5]$ concatenates the two feature maps $\mathbf{G}^1$ and $\mathbf{G}^5$ together. $\mathbf{V}^6$ and $\hat{\mathbf{b}}^6$ represent the weights and biases of the sixth layer for convolution. It is noted that the merged feature map contain the initial simple features and the complicated features after multiple convolution layers. Next, by the following convolution, BN and max-pooling operations, the redundant features in the above merged feature map can be removed.

Compared with the SBNN, the second change in SCNN is the global average pooling, which replaces the three cumbersome fully-connected layers. Specifically, instead of the fully-connected layer, this paper adopts one convolution layer M_{13} (with kernel size of one and kernel number of two) followed by a simple global average pooling layer M_{14} to yield a pair of average numbers. The global average pooling is computed by the following:

$$\mathbf{G}_t^{14} = \frac{1}{W_t \cdot H_t} \sum_{s=1}^{W_t \cdot H_t} g_{s,t} \tag{7}$$

where $g_{s,t}$ stands for the s-th pixel value in the t-th feature map $\mathbf{G}_t^{13}$; W_t and H_t are the width and height of $\mathbf{G}_t^{13}$. Based on the softmax function, we can derive the output probability of the u-th neuron for the u-th class to be as follows:

$$\hat{y}_u = \frac{\exp(\mathbf{G}_u^{14})}{\sum_{i=1}^{2} \exp(\mathbf{G}_i^{14})}, \quad u = 1, 2. \tag{8}$$

Such a replacement can secure two obvious advantages: one is to largely reduce model parameters and thereby mitigate the over-fitting problem; the other is that the SCNN is available to accommodate various sizes of the input image patch.

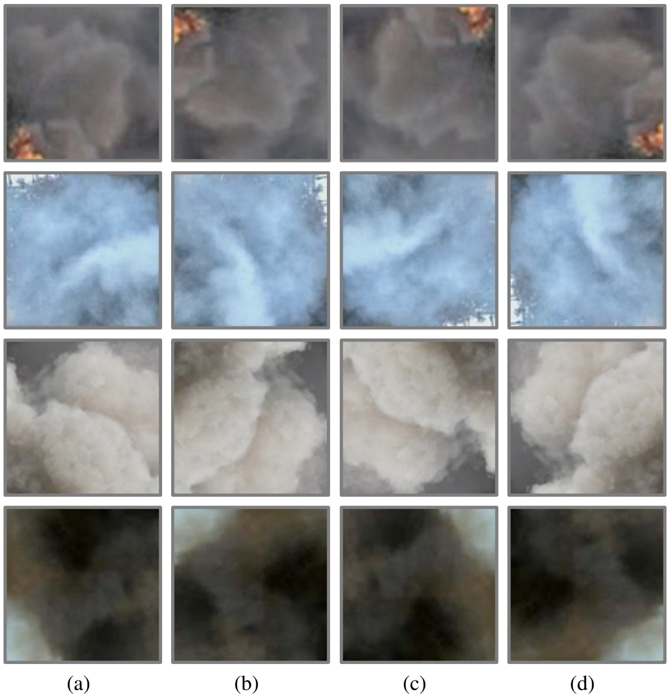


Fig. 4: Examples of data augmentation: (a) Pristine image patches in the dataset for training the network; (b) image patches rotated by 90 degrees; (c) image patches rotated by 180 degrees; and (d) image patches rotated by 270 degrees.

C. DCNN's Architecture

Through substantial experiments, it was found that both the proposed SBNN and SCNN have attained high performance. The SBNN is good at extracting the detail information of smoke, while the SCNN can nicely capture base information of smoke[2]. It is natural to integrate the advantages of SBNN and SCNN to construct the dual-channel DCNN for smoke detection. More concretely, we extract a part of SBNN by eliminating all three fully-connected layers. The extracted part of SBNN, dubbed SBNN$_0$, is leveraged for feature extraction. Similarly, we extract SCNN$_0$ by removing the global average pooling layer from the SCNN. Note that the size of SBNN$_0$'s output is not matched with that of SCNN$_0$'s output. Hence, we further modify SBNN$_0$ by deleting the third max-pooling layer L_9. Then, we incorporate SBNN$_0$ and SCNN$_0$ by means of a concatenation operation followed by a convolutional layer $\hat{M}_{13}$ with its kernel size of one:

$$\hat{\mathbf{G}}^{13} = \max(0, \mathbf{V}^{13} \star [\mathbf{F}^8, \mathbf{G}^{12}] + \hat{\mathbf{b}}^{13}). \tag{9}$$

So far, we have provided the proposed dual-channel network structure for feature extraction and feature fusion, as shown in the left side of Fig. 3.

It still requires some layers for classification in our DCNN. It is apparent that the fully-connected layers include much more learnable parameters than the global average pooling layer. Consequently, we append the global average pooling layer $\hat{M}_{14}$ to the last convolutional layer $\hat{M}_{13}$ to compute two

[2] More discussions about these conclusions will be illustrated in the next section.

TABLE III: Illustration of network parameters of SBNN. In the leftmost column, $L_1 \sim L_{12}$ are highlighted in red ink in Fig. 1.

Layer	Type	Network parameters				
L_1, L_2	Convolution	Filter size: 3×3	Filter number: 32	Stride: 1×1	Padding: Same	Activation function: ReLU
L_3	Pooling	Pooling region size: 3×3		Stride: 2×2	Padding: Same	Pooling method: Max-pooling
L_4, L_5	Normalization and convolution	Filter size: 3×3	Filter number: 64	Stride: 1×1	Padding: Same	Activation function: ReLU
L_6	Pooling	Pooling region size: 2×2		Stride: 2×2	Padding: Valid	Pooling method: Max-pooling
L_7, L_8	Normalization and convolution	Filter size: 3×3	Filter number: 384	Stride: 1×1	Padding: Same	Activation function: ReLU
L_9	Pooling	Pooling region size: 2×2		Stride: 2×2	Padding: Valid	Pooling method: Max-pooling
L_{10}, L_{11}	Full Connection	Neurons number: 2048				Dropout : 0.5
L_{12}	Output	Neurons number: 2				

TABLE IV: Illustration of network parameters of SCNN. In the leftmost column, $M_1 \sim M_{14}$ are highlighted in red ink in Fig. 2.

Layer	Type	Network parameters				
M_1	Convolution	Filter size: 9×9	Filter number: 32	Stride: 1×1	Padding: Same	Activation function: ReLU
M_2	Convolution	Filter size: 1×1	Filter number: 64	Stride: 1×1	Padding: Same	Activation function: ReLU
M_3, M_4, M_5	Normalization and convolution	Filter size: 3×3	Filter number: 64	Stride: 1×1	Padding: Same	Activation function: ReLU
M_6	Concatenation and convolution	Filter size: 1×1	Filter number: 64	Stride: 1×1	Padding: Same	Activation function: ReLU
M_7	Pooling	Pooling region size: 3×3		Stride: 2×2	Padding: Same	Pooling method: Max-pooling
M_8, M_9	Normalization and convolution	Filter size: 3×3	Filter number: 128	Stride: 1×1	Padding: Same	Activation function: ReLU
M_{10}	Pooling	Pooling region size: 3×3		Stride: 2×2	Padding: Same	Pooling method: Max-pooling
M_{11}, M_{12}	Normalization and convolution	Filter size: 3×3	Filter number: 256	Stride: 1×1	Padding: Same	Activation function: ReLU
M_{13}	Convolution	Filter size: 1×1	Filter number: 2	Stride: 1×1	Padding: Same	Activation function: ReLU
M_{14}	Pooling	Pooling method: Global average pooling				Activation function: Softmax

average numbers. Then, we use the softmax function to yield two output probability values. The above processes can be implemented by referring to Eqns. (7)-(8). Fig. 3 presents the whole architecture of DCNN.

D. Network Training

During network training, *first*, we independently train each of the dual channels, namely, SBNN and SCNN. For illustration, consider the training of SBNN. We take advantage of the trial-and-error method to find the optimized network structure, as tabulated in Table III. We then introduce the glorot uniform method to initialize the network weights [35] and apply the momentum and learning rate decay to advance the training effect and prevent it from falling into the local optimum [36]. More specifically, the stochastic gradient descent is exploited to train the SBNN by assigning the momentum coefficient as 0.9, the initial learning rate as 0.01, and the learning rate decay coefficient as 0.0001 [37]. Analogous to the majority of classification tasks, one-hot encoding is deployed during the training of SBNN with the loss function of cross entropy:

$$e(\mathbf{x}, \hat{\mathbf{x}}) = -\sum_{k=1}^{2} x_k \log \hat{x}_k \tag{10}$$

where $\mathbf{x} = [x_1, x_2]^T$ represents the vector of the class label and $\hat{\mathbf{x}} = [\hat{x}_1, \hat{x}_2]^T$ represents the vector of the category probability. The mini-batch size and the trained epoch are set to be 96 and 300, respectively. In the above-mentioned environment, we adjust the SBNN's model parameters based on the training set and determine the optimal model parameters by making the network obtain the best accuracy on the validation set. The same process is carried out to train the SCNN by minimizing the loss function $-\sum_{k=1}^{2} x_k \log \hat{y}_k$. Its optimized network structure is shown in Table IV.

Second, we incorporate the SBNN and SCNN to constitute the whole DCNN, as exhibited in Fig. 3. We train the DCNN by optimizing part of the network parameters (i.e., the convolutional layer $\hat{M}_{13}$) and freezing the others (i.e., SBNN$_0$ and SCNN$_0$)[3]. *Third*, we fine-tune the overall parameters of our proposed DCNN to search for the optimal parameters. During the above-mentioned two steps, we minimize the loss function $-\sum_{k=1}^{2} x_k \log \hat{z}_k$.

To decrease the variance of image patches and improve the network's robustness, we further introduce two image preprocessing methods, namely, patch normalization and data augmentation. Normalization can effectively diminish the influence of brightness changes on smoke detection. This paper uses the pixelwise min-max normalization method for image patch normalization [38], which is calculated by the following:

$$d_n = \frac{d_r - d_{\min}}{d_{\max} - d_{\min}} \tag{11}$$

where d_n is the normalization value of a pixel, d_r means the intensity value of a pixel, and $d_{\min}$ and $d_{\max}$ are the minimum and maximum values of the pixels in the image patch, respectively.

In the classification task, the relative balance of data between the categories has a significant improvement on the performance of the algorithm [22]. For example, in the dataset for training the network, there are approximately 2200 smoke image patches and approximately 8500 smoke-free image patches in total. By 90-degree, 180-degree and 270-degree rotations, the number of smoke image patches is increased to a similar number of the smoke-free image patches. Due to the characteristics of smoke, these image patches can be considered new smoke image patches acquired by rotation

[3] We transfer the well-trained network parameters in SBNN and SCNN to SBNN$_0$ and SCNN$_0$ in the DCNN.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2019.2929009, IEEE Transactions on Multimedia

TABLE V: Comparison with models based on hand-crafted features.

Methods		HLTPMC [6]	MCLBP [41]	DCNN (Prop.)
Set1	AR	96.4%	96.9%	99.7%
	DR	97.7%	97.6%	99.5%
	FAR	4.57%	3.68%	0.12%
Set2	AR	98.4%	97.8%	99.4%
	DR	98.5%	98.4%	99.0%
	FAR	2.44%	2.86%	0.24%

TABLE VI: Comparison of the proposed DCNN with eight mainstream or state-of-the-art deep convolutional neural networks.

Networks		Alex-Net [11]	ZF-Net [12]	VGG-Net [13]	GoogLe-Net [14]	Xception [15]	Res-Net [16]	Dense-Net [27]	DNCNN [22]	DCNN Prop.
Set1	AR	95.6%	96.0%	96.8%	97.0%	97.9%	97.2%	98.6%	97.8%	99.7%
	DR	94.9%	93.6%	95.2%	95.8%	96.7%	95.1%	98.3%	95.2%	99.5%
	FAR	3.85%	2.41%	2.16%	2.17%	0.13%	1.44%	1.08%	0.48%	0.12%
Set2	AR	96.9%	97.6%	97.9%	98.1%	98.4%	98.1%	98.4%	98.0%	99.4%
	DR	96.5%	97.9%	97.9%	97.2%	98.0%	97.4%	98.2%	96.3%	99.0%
	FAR	2.69%	2.57%	2.08%	1.22%	1.10%	1.22%	1.10%	0.48%	0.24%
Number of parameters		60 million	60 million	120 million	7 million	20 million	60 million	7 million	20 million	2.7 million

operations. The smoke image patches generated by the above data augmentation technology are associated with the different flow directions of the smoke. For the convenience of readers, Fig. 4 provides augmentation effects of several representative smoke image patches.

III. Experimental Results

This section will confirm the performance of our proposed DCNN for detecting smoke and demonstrate its superiority compared with state-of-the-art relevant competitors. This section is composed of the experimental protocol, performance comparison, implementation speed, feature map visualization, discussion, and testing of real applications.

A. Experimental Protocol

TensorFlow [39] and Keras [40] are used in our experiment for training the proposed DCNN for smoke detection. The experimental environment is the Windows 10 operation system running on a server with an Intel(R) Core i7-7820X CPU at 3.60 GHz and an NVIDIA GeForce GTX 1080.

In this test, we deploy the publicly available smoke detection database [22], which is composed of four subsets, namely, Set-1, Set-2, Set-3 and Set-4. Specifically, Set-1 (including 831 smoke-free image patches and 552 smoke image patches) and Set-2 (including 817 smoke-free image patches and 688 smoke image patches) are utilized for checking the detection performance of the network. Set-3 consists of 8804 smoke image patches, which is created by exerting the data augmentation on the original 2201 smoke image patches, and 8511 smoke-free image patches for training the network. Set-4 contains 9016 smoke image patches, which were produced by augmenting the original 2254 smoke image patches, and 8363 smoke-free image patches for validating the network. The leftmost column in Fig. 4 presents four typical smoke image patches contained in the smoke detection database.

For quantifying the performance of our proposed network with others, we apply three typical evaluation indicators that include accuracy rate (AR), detection rate (DR) and false alarm rate (FAR), as defined by the following:

$$\mathrm{AR} = \frac{P_1 + N_2}{T_1 + T_2} \times 100\% \tag{12}$$

$$\mathrm{DR} = \frac{P_1}{T_1} \times 100\% \tag{13}$$

$$\mathrm{FAR} = \frac{N_1}{T_2} \times 100\% \tag{14}$$

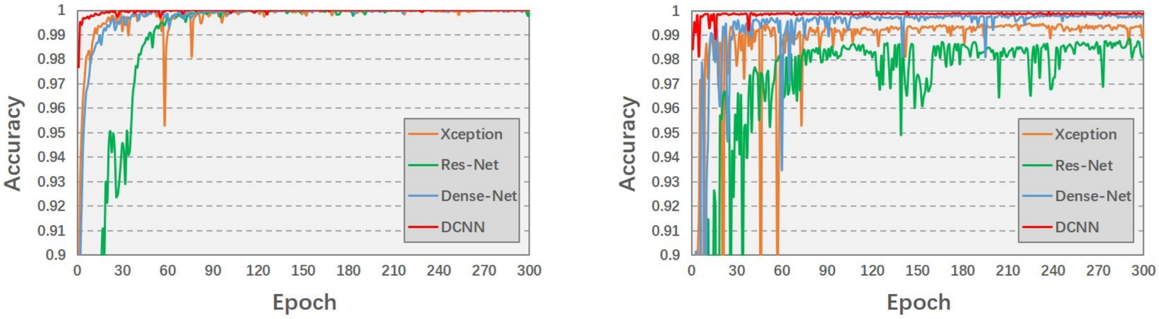
where T_1 and T_2 are the numbers of positive samples and negative samples, respectively; P_1, N_1 and N_2 stand for the number of correctly detected true positive samples, the number of negative samples falsely classified as positive samples, and the number of correctly detected true negative samples. A good model is expected to achieve a high value in AR and DR but a low value in FAR.

B. Performance Comparison

First, we examine the performance of the proposed DCNN and tabulate its results in Table V. As seen, our DCNN has achieved very high performance, even greater than 99.5% on average. To verify the superiority of the DCNN, we compare it with two popular models, HLTPMC [6] and MCLBP [41], which were developed based on hand-crafted features followed by the radial basis function (RBF) kernel-based SVM. Via the grid search, the best SVM parameters can be obtained by training on Set-3 (17315 patches) and validation on Set-4 (17379 patches). Specifically, we set both the penalty coefficient and gamma coefficient in the SVM as 1 for HLTPMC, and set them as 798 and 102 for MCLBP respectively. Their results are tabulated in Table V. It can be observed that our DCNN has given rise to noticeably greater performance than HLTPMC and MCLBP. More concretely, considering the AR

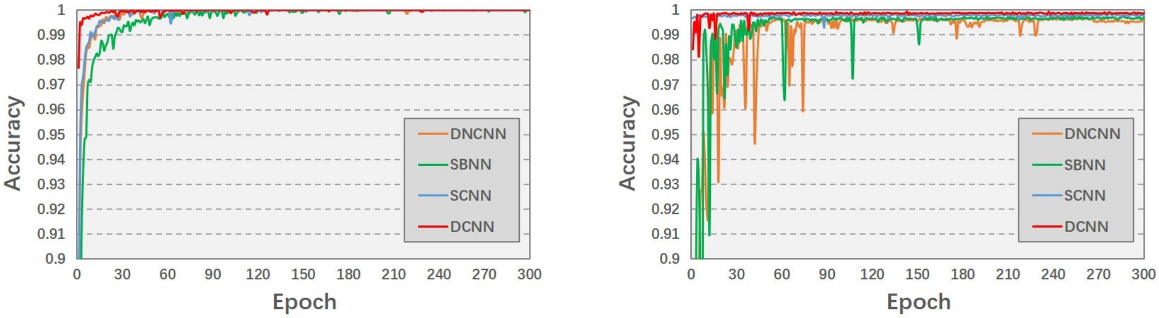
TABLE VII: Comparison of DCNN with its two components and DNCNN.

Methods		DNCNN [22]	SBNN (Prop.)	SCNN (Prop.)	DCNN (Prop.)
Set1	AR	97.8%	98.3%	98.6%	99.7%
	DR	95.2%	97.3%	97.6%	99.5%
	FAR	0.48%	0.96%	0.84%	0.12%
Set2	AR	98.0%	98.7%	98.5%	99.4%
	DR	96.3%	98.4%	97.2%	99.0%
	FAR	0.48%	0.98%	0.48%	0.24%



(a) Training process (b) Validation process

Fig. 5: Plots of accuracy curves of Xception, Res-Net, Dense-Net and DCNN, during the training and validation processes.



(a) Training process (b) Validation process

Fig. 6: Plots of accuracy curves of DNCNN, SBNN, SCNN and DCNN during the training and validation processes.

as the evaluation indicator, the relative performance gains of our proposed DCNN over the HLTPMC and MCLBP models are, respectively, 3.4% and 2.9% on Set-1, as well as 1.0% and 1.6% on Set-2 respectively.

Second, we compare the proposed DCNN with eight popular or state-of-the-art deep neural networks, which include Alex-Net [11], ZF-Net [12], VGG-Net [13], GoogLe-Net [14], Xception [15], Res-Net (152 layers) [16], Dense-Net [27], and DNCNN [22]. The performance indices of these eight networks are illustrated in Table VI. We can easily find that the DCNN has acquired the optimal performance. In view of the AR index, the proposed DCNN has introduced a relative performance gain of 1.9% on Set-1 and 1.4% on Set-2 in comparison with the fourth-performing DNCNN, which is a recently devised deep convolutional network specific to smoke detection. In contrast to the third-place Xception, the relative performance gains achieved by our DCNN are 1.8% on Set-1 and 1.0% on Set-2. The relative performance gains between the DCNN and the second-rank Dense-Net are 1.1% on Set-1 and 1.0% on Set-2. We also compare the number of parameters used in the network because it is also a significant indicator. To achieve an excellent network, it is desirable that the network contains fewer parameters and thus has strong generalization ability. As listed in Table VI, our DCNN just involves 2.7 million parameters, less than one-third of the parameters used in the state-of-the-art deep networks considered in this paper. Furthermore, the standard deviations of the eight deep networks tested and our DCNN across twenty iterations are checked and compared. The standard deviations of Alex-Net, ZF-Net, VGG-Net, GoogLe-Net, Xception, Res-Net, Dense-Net, DNCNN and DCNN are respectively 0.2382, 0.1436, 0.1948, 0.0049, 0.0031, 0.0063, 0.0123, 0.1014 and 0.0020 on Set-1, and 0.2179, 0.1338, 0.1882, 0.0034, 0.0034, 0.0036, 0.0058, 0.0502 and 0.0012 on Set-2. From these results, we can ascertain that the proposed DCNN has a considerably stable performance, superior to the other competitors.

Third, the proposed DCNN is compared with its two components, namely, SBNN and SCNN. We tabulate the results of SBNN, SCNN and DCNN in Table VII. We also include the recently developed DNCNN for comparison, since the SBNN

TABLE VIII: Comparison of ablation analysis on each component of the proposed DCNN.

Networks		DCNN_1	DCNN_2	DCNN_3	DCNN_4	DCNN_5	DCNN_6	DCNN
Set1	AR	97.3%	98.6%	98.4%	98.7%	98.7%	97.6%	99.7%
	DR	96.0%	98.7%	96.9%	98.0%	98.0%	96.4%	99.5%
	FAR	1.07%	1.44%	0.60%	0.84%	0.84%	1.56%	0.12%
Set2	AR	99.1%	99.3%	98.4%	98.7%	98.9%	98.5%	99.4%
	DR	99.0%	99.0%	98.3%	97.5%	98.3%	98.3%	99.0%
	FAR	0.85%	0.49%	1.29%	0.37%	0.49%	1.22%	0.24%

TABLE IX: Implementation speed comparison of our proposed DCNN with eight deep convolutional neural networks.

Networks	Alex-Net [11]	ZF-Net [12]	VGG-Net [13]	GoogLe-Net [14]	Xception [15]	Res-Net [16]	Dense-Net [27]	DNCNN [22]	DCNN Prop.
Speed (millisecond/patch)	0.366	1.027	0.384	0.701	0.610	1.523	1.131	0.369	0.453

is inspired by the DNCNN. It can be readily found that our SBNN is superior to the DNCNN, which might be attributed to the introduction of selectively appending the BN layer after the convolutional layer. In addition, SCNN performs better than SBNN, which is possibly due to the use of the *skip connection* for preventing the vanishing gradient and enhancing the feature propagation as well as the *global average pooling* for decreasing the number of parameters and mitigating the problem of over-fitting. Last, we can find that the proposed DCNN is better than SCNN. This might be due to the appropriate fusion of SBNN and SCNN to complement each other, since they are good at extracting detail information and base information of smoke, respectively. In Section III-D, we will discuss in detail the complementarity of these two networks in capturing the characteristics of smoke.

Fourth, we visualize the curves of training accuracy and validation accuracy to further compare the whole training process of the state-of-the-art Xception, Res-Net, Dense-Net and DNCNN, as well as the proposed SBNN, SCNN and DCNN. For clarity, we divide the above seven networks into two groups. One group is composed of Xception, Res-Net, Dense-Net and DCNN, as shown in Figs. 5(a)-(b), and the other group is composed of DNCNN, SBNN, SCNN and DCNN, as shown in Figs. 6(a)-(b). Let us first consider Fig. 5. From (a), it can be viewed that in comparison with Xception, Res-Net and Dense-Net, the proposed DCNN converges more quickly, and its training accuracy values can approach one as the epoch increases to surpass 30. From (b), we can see that the validation accuracy values are quite different among the four testing networks. According to the accuracy values, we are able to derive the following rank: DCNN $>$ Dense-Net $>$ Xception $>$ Res-Net. Moreover, it can be found that the validation accuracy values of Xception and Res-Net are quite oscillatory, which is possibly because they have a deeper structure and a small change in model parameters may largely affect the validation accuracy. We then observe Fig. 6. Two important observations can be established: 1) the convergence speed of DCNN is remarkably faster than its two components (namely, SBNN and SCNN) and the recently proposed DNCNN during network training; and 2) DCNN has superior and stable accuracy values compared with the other three networks tested during the validation process. In summary, the introduction of fusing SBNN and SCNN for feature extraction can contribute substantially to our DCNN.

Fifth, we conduct the ablation analysis on each component of the proposed DCNN. We remove all the BN layers from DCNN and call such a network DCNN_1. Note that in SBNN_0 and SCNN_0, several convolution layers exist that the BN layers are not appended to. Therefore, we add the BN layers after L_1 and L_2 in SBNN_0 and M_1 and M_2 in SCNN_0. Such a modified network is named DCNN_2. Further, we separately eliminate the skip connection in SCNN_0, the momentum, as well as the learning rate decay, and dub those three networks as DCNN_3, DCNN_4 and DCNN_5. Finally, we replace the global average pooling layer with the typically used fully-connected layers. In particular, L_{10}, L_{11} and L_{12} in SBNN are used to replace $\hat{M}_{14}$ in our proposed DCNN. This modified network is called DCNN_6. We train the DCNN_1, DCNN_2, DCNN_3, DCNN_4, DCNN_5 and DCNN_6 based on the same method applied in DCNN and tabulate their detection performances in Table VIII. According to the results, two dominant conclusions can be drawn. First, by introducing the selective-based BN, skip connection, momentum, learning rate decay, and global average pooling, our proposed DCNN has attained the best classification performance. Second, in contrast to the others, DCNN_1 and DCNN_6 have low testing accuracy values, and this implies that BN and global average pooling have provided greater contributions to the DCNN.

Sixth, we carry out the comparison with other fusion strategies of SBNN and SCNN. Since the task of smoke detection in this work is a binary classification problem, the direct fusions of decisions of SBNN and SCNN are their union and intersection, namely, 'SBNN $\cup$ SCNN' and 'SBNN $\cap$ SCNN'. The AR, DR and FAR results of their union are 98.3%, 96.0% and 0.24% on Set-1, and 98.4%, 97.1% and 0.48% on Set-2 respectively. The AR, DR and FAR results of their intersection are 98.6%, 98.9% and 1.58% on Set-1, and 98.8%, 98.6% and 0.98% on Set-2 respectively. Clearly, the fusion of decisions of SBNN and SCNN is appreciably inferior to the proposed DCNN, which combines SBNN and SCNN in terms of feature fusion.

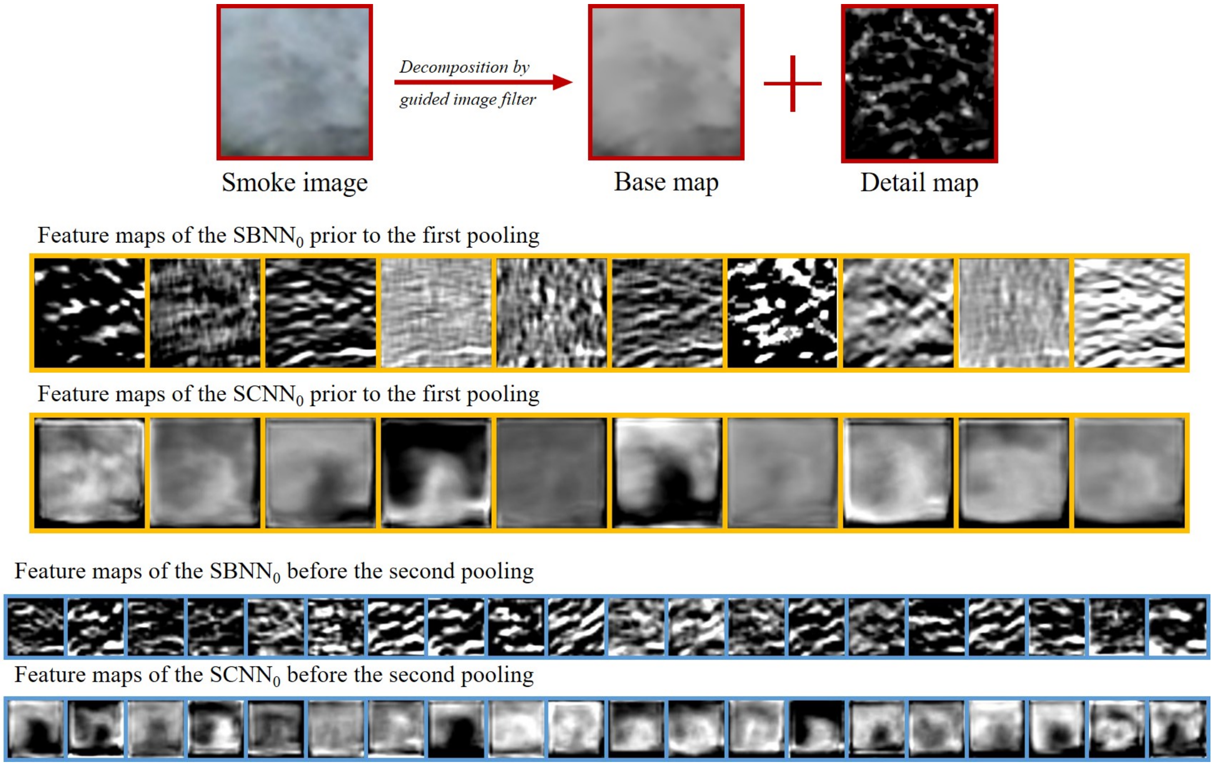

Smoke image
Decomposition by
guided image filter
Base map
Detail map
Feature maps of the SBNN$_0$ prior to the first pooling
Feature maps of the SCNN$_0$ prior to the first pooling
Feature maps of the SBNN$_0$ before the second pooling
Feature maps of the SCNN$_0$ before the second pooling

Fig. 7: Illustration of typical intermediate feature maps extracted using SBNN$_0$ and SCNN$_0$ prior to two pooling layers.

C. Implementation Speed

Implementation efficiency is also a significant indicator. As illustrated in Table IX, we compare the implementation time of our proposed DCNN and the eight deep networks tested. Specifically, we run each network on all 2888 testing RGB image patches of size 48×48×3 (1383 patches from Set-1 and 1505 patches from Set-2) and then compute the average time for each image patch. In this test, a computer was configured with a CPU processor of 2.1 GHz, an NVIDIA TITAN Xp GPU of 43.9 GB, and 64.0 GB of RAM. One can easily find that the proposed DCNN only consumes less than 0.5 millisecond for each patch, obviously faster than the state-of-the-art Res-Net and Dense-Net.

D. Feature Map Visualization

We further discuss the necessity of fusing the feature maps extracted by using SBNN$_0$ and SCNN$_0$ to complement each other. More specifically, we exhibit in Fig. 7 a sample image patch and its associated visualized intermediate feature maps extracted from SBNN$_0$ and SCNN$_0$. The visualized feature maps before each pooling operation are considered as examples. First, let us observe the feature maps prior to the first pooling operation. A large gap of feature maps between SBNN$_0$ and SCNN$_0$ can be found. Then, we compare the feature maps of SBNN$_0$ and SCNN$_0$ before the second pooling operation. We can also find a much larger distance exists between the feature maps of the two subnetworks. Obviously, the SBNN$_0$ and SCNN$_0$ are quite different.

Furthermore, we leverage the popular guided image filter (GIF) [42] to decompose the sample smoke image patch into a base map and a detail map, as shown in Fig. 7. Such an operation has been widely applied in numerous multimedia applications [43]-[45]. As shown, the base map contains the large-scale variations in intensity, whereas the detail map contains the small-scale details. Comparing the base and detail maps with the feature maps mentioned above, we can find that the feature maps of SBNN$_0$ are similar to the detail map, while the feature maps of SCNN$_0$ are similar to the base map. That is, the SBNN$_0$ is mainly devoted to extracting the detailed features from smoke image patches and the SCNN$_0$ mostly focuses on extracting the basic features. The SBNN$_0$ and SCNN$_0$ can nicely complement each other and effectively boost the DCNN's performance.

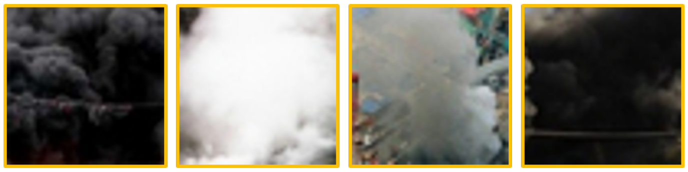


Fig. 8: Falsely detected samples from Set-1 and Set-2.

E. Discussion

Analysing the falsely detected samples is a considerably beneficial insight for improving the performance of our proposed network. Thus, we take into account four typical samples wherein the proposed DCNN fails, as shown in Fig. 8. It is not difficult to observe that our DCNN is not good at detecting smoke that has fewer textures, just as the samples show in Fig.

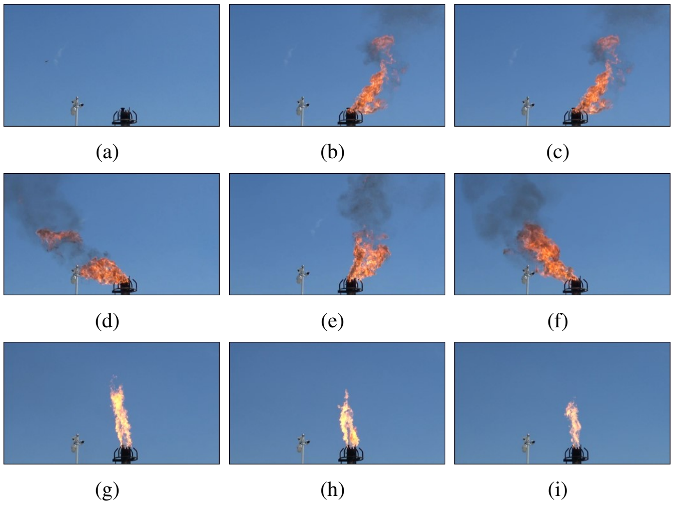

(a) (b) (c)
(d) (e) (f)
(g) (h) (i)

Fig. 9: Typical flame images in petrochemical enterprises.

8. To address such a difficulty, our future work will consider enhancing image textures prior to detection. In addition, it is worthwhile to emphasize that the proposed DCNN systematically integrates contour information and texture information, both of which are very important for smoke detection. In contrast, the well-known deep networks, such as Res-Net and Dense-Net, were developed specifically for image recognition tasks, in which the semantic information (e.g., contour) plays the most crucial function. In summary, our DCNN is proposed particularly for extracting smoke characteristics and detecting smoke and is thereby superior to these famous deep networks that we have tested.

F. Testing of Real Applications

In this section, we will examine the proposed DCNN in two important real applications. The first application is to detect whether there is smoke emitted from a flare that burns waste gas produced by petrochemical enterprises, where such flares are employed to maintain safety and prevent harm to the environment. Importantly, black smoke will be generated from the flare if the exhaust gas is not sufficiently burned. In such a case, some water vapour should be sent to the flame for smoke abatement. However, determining how to automatically adjust the volume of water vapour is a significant problem. The proposed DCNN can be used to solve this problem by detecting black smoke from the camera image, which is then followed by controlling the volume of water vapour. In Fig. 9, we display some typical intermediate flame images. The image number and its associated result of our proposed DCNN are as follows: (a) smoke-free, (b) smoke, (c) smoke, (d) smoke, (e) smoke, (f) smoke, (g) smoke-free, (h) smoke-free, and (i) smoke-free, which are the same as the real results.

The second application is to apply camera images to detect cigarette smoke, as shown in Fig. 10. Ten thousand RGB image patches of size 48×48×3 are randomly selected from these two camera images, and then labelled by five graduate students. According to the label results, we preserved 8919 image patches, each of which has the exact same label results provided by all five graduate students. The detection accuracy values of our DCNN and state-of-the-art Xception, Dense-Net and DNCNN are tabulated in Table X. We are able to derive two crucial conclusions: 1) the proposed DCNN is slightly superior to Dense-Net and obviously better than Xception and DNCNN; and 2) all the deep networks are not greatly adept in detecting cigarette smoke from the camera images.

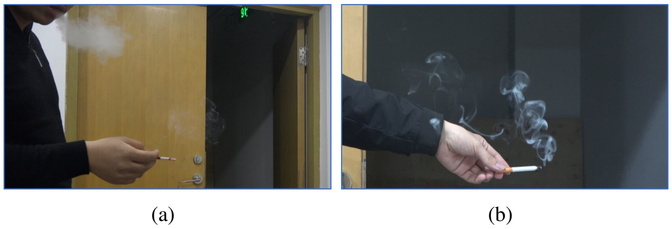

(a) (b)

Fig. 10: Two camera images of cigarette smoke detection.

TABLE X: Comparison on cigarette smoke detection.

AR	Xception	Dense-Net	DNCNN	DCNN
Fig. 10(a)	72.4%	78.5%	57.5%	80.6%
Fig. 10(b)	65.4%	75.8%	59.1%	79.3%
Overall	68.6%	77.1%	58.3%	79.8%

In the future, we plan to focus on detecting black smoke from flame images and light smoke from cigarette images. Specifically, we will first build two large-size image datasets for black smoke detection and cigarette smoke detection. Second, we will design specific deep neural networks that consider the characteristics of black smoke and cigarette smoke for detection.

IV. Conclusions

In this paper, we have investigated the problem of image-based smoke detection by devising a novel deep dual-channel neural network dubbed DCNN. In contrast to the recently proposed deep neural networks, including Alex-Net, ZF-Net, VGG-Net, GoogLe-Net, Xception, Res-Net, Dense-Net, and the smoke-specific DNCNN, the proposed DCNN is established mainly based on the fusion of two channels of deep subnetworks. The first channel's subnetwork is built by first connecting multiple convolutional layers and max-pooling layers sequentially, and then appending the BN layer to part of the last convolutional layers selectively. The second channel's subnetwork is constructed by incorporating the skip connection and global average pooling with the convolutional, BN, and max-pooling layers. The skip connection can help to prevent the vanishing gradient and enhance the feature propagation. The global average pooling is beneficial in decreasing the number of network parameters and mitigating the problem of over-fitting. The proposed DCNN is finally designed to combine the aforementioned two deep subnetworks by a concatenation operation. We implement comparative experiments on the publicly available smoke detection image database to confirm the effectiveness of our deep network.

Compared with the recently developed smoke detection models and state-of-the-art deep neural networks, our DCNN has achieved optimal performance, beyond 99.5% on average,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2019.2929009, IEEE Transactions on Multimedia

with the least network parameters. Furthermore, through a numerical comparison and a visualized comparison, we illustrate that the superiority of the proposed deep network is primarily achievable because the dual deep subnetworks mentioned above can complement each other.

References

[1] B. U. Töreyin, Y. Dedeoğlu, and A. E. Çetin, "Wavelet based real-time smoke detection in video," *in Proc. Eur. Signal Process. Conf.*, pp. 1-4, Sep. 2005.

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

[3] F. Yuan, "A fast accumulative motion orientation model based on integral image for video smoke detection," *Pattern Recognit. Lett.*, vol. 29, no. 7, pp. 925-932, May 2008.

[4] F. Yuan, "Video-based smoke detection with histogram sequence of LBP and LBPV pyramids," *Fire Safety J.*, vol. 46, no. 3, pp. 132-139, Apr. 2011.

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

[6] F. Yuan, J. Shi, X. Xia, Y. Fang, Z. Fang, and T. Mei, "High-order local ternary patterns with locality preserving projection for smoke detection and image classification," *Inf. Sci.*, vol. 372, pp. 225-240, Dec. 2016.

[7] F. Yuan, X. Xia, J. Shi, H. Li, and G. Li, "Non-linear dimensionality reduction and Gaussian process based classification method for smoke detection," *IEEE Access*, vol. 5, pp. 6833-6841, Apr. 2017.

[8] H. Wang and B. Raj, "On the origin of deep learning," *arXiv preprint arXiv:1702.07800*, Mar. 2017.

[9] I. Hadji and R. P. Wildes. "What do we understand about convolutional networks?" *arXiv preprint arXiv:1803.08834*, Mar. 2018.

[10] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F.-F. Li, "Imagenet: A large-scale hierarchical image database," *in Proc. IEEE Conf. Comp. Vis. and Pattern Recognit.*, pp. 248-255, Jun. 2009.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *in Proc. Adv. Neural Inf. Process. Syst.*, vol. 25. pp. 1097-1105, 2012.

[12] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *in Proc. Eur. Conf. Comp. Vis.*, pp. 818-833, Sep. 2014.

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

[14] 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. Comp. Vis. and Pattern Recognit.*, pp. 1-9, Jun. 2015.

[15] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *in Proc. IEEE Conf. Comp. Vis. and Pattern Recognit.*, pp. 1251-1258, Jul. 2017.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *in Proc. IEEE Conf. Comp. Vis. and Pattern Recognit.*, pp. 770-778, Jun. 2016.

[17] J. Guan, S. Yi, X. Zeng, W.-K. Cham, and X. Wang, "Visual importance and distortion guided deep image quality assessment framework," *IEEE Trans. Multimedia*, vol. 19, no. 11, pp. 2505-2520, Nov. 2017.

[18] N. Takahashi, M. Gygli, and L. V. Gool, "Aenet: Learning deep audio features for video analysis," *IEEE Trans. Multimedia*, vol. 20, no. 3, pp. 513-524, Mar. 2018.

[19] Z. Qiu, T. Yao, and T. Mei, "Learning deep spatio-temporal dependence for semantic video segmentation," *IEEE Trans. Multimedia*, vol. 20, no. 4, pp. 939-949, Apr. 2018.

[20] G. Ning, Z. Zhang, and Z. He, "Knowledge-guided deep fractal neural networks for human pose estimation," *IEEE Trans. Multimedia*, vol. 20, no. 5, pp. 1246-1259, May 2018.

[21] S. Zhang, S. Zhang, T. Huang, and W. Gao, "Speech emotion recognition using deep convolutional neural network and discriminant temporal pyramid matching," *IEEE Trans. Multimedia*, vol. 20, no. 6, pp. 1576-1590, Jun. 2018.

[22] Z. Yin, B. Wan, F. Yuan, X. Xia, and J. Shi, "A deep normalization and convolutional neural network for image smoke detection," *IEEE Access*, vol. 5, pp. 18429-18438, Aug. 2017.

[23] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, Feb. 2015.

[24] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: Oportunities, barriers and policy recommendations," *Transportation Research Part A: Policy and Practice*, vol. 77, pp. 167-181, Jul. 2015.

[25] S. Nah, T. H. Kim, and K. M. Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," *in Proc. IEEE Conf. Comp. Vis. Pattern Recognit.*, vol. 1, no. 2, pp. 3883-3891, Jul. 2017.

[26] B. Lim, S. Son, H. Kim, S. Nah, and K. Lee. "Enhanced deep residual networks for single image super-resolution." *in Proc. IEEE Conf. Comp. Vis. Pattern Recognit. Workshops*, vol. 1, no. 2, pp. 136-144, Jul. 2017.

[27] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *in Proc. IEEE Conf. Comp. Vis. Pattern Recognit.*, pp. 4700-4708, Jul. 2017.

[28] C. Liu, F. Tsow, Y. Zou, and N. Tao, "Particle pollution estimation based on image analysis," *PloS one*, vol. 11. no. 2, pp. e0145955, 2016.

[29] K. Gu, J. Qiao, and X. Li, "Highly efficient picture-based prediction of PM2.5 concentration," *IEEE Trans. Ind. Electron.*, vol. 66, no. 4, pp. 3176-3184, Apr. 2019.

[30] P.-W. Soh, J.-W. Chang, and J.-W. Huang, "Adaptive deep learning-based air quality prediction model using the most relevant spatial-temporal relations," *IEEE Access*, vol. 6, pp. 38186-38199, 2018.

[31] K. Gu, J. Qiao, and W. Lin, "Recurrent air quality predictor based on meteorology-and pollution-related factors," *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 3946-3955, Sep. 2018.

[32] K. Gu, Z. Xia, and J. Qiao, "Stacked selective ensemble for PM2.5 forecast," *IEEE Trans. Instrum. Meas.*, 2019.

[33] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," *in Proc. Conf. Artificial Intelligence and Statistics*, pp. 315-323, Jun. 2011.

[34] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929-1958, Jan. 2014.

[35] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *J. Mach. Learn. Res.*, vol. 9, pp. 249-256, Mar. 2010.

[36] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," *Neural networks: Tricks of the trade*, Springer, Berlin, Heidelberg, pp. 437-478, 2012.

[37] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[38] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 3rd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2006.

[39] M. Abadi, A. Agarwal, P. Barham, *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, Mar. 2016.

[40] F. Chollet. Keras, 2015. [Online]. Available: http://keras.io

[41] S. R. Dubey, S. K. Singh, and R. K. Singh, "Multichannel decoded local binary patterns for content-based image retrieval," *IEEE Trans. Image Process.*, vol. 25, no. 9, pp. 4018-4032, Sep. 2016.

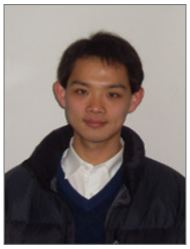
[42] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397-1409, Jun. 2013.

[43] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, "Edge-preserving decompositions for multi-scale tone and detail manipulation," *ACM Trans. Graphics*, vol. 27, no. 3, pp. 249-256, Aug. 2008.

[44] S. Li, X. Kang, and J. Hu, "Image fusion with guided filtering," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2864-2875, Jul. 2013.

[45] G. A. Kordelas, D. S. Alexiadis, P. Daras, and E. Izquierdo, "Content-based guided image filtering, weighted semi-global optimization, and efficient disparity refinement for fast and accurate disparity estimation," *IEEE Trans. Multimedia*, vol. 18, no. 2, pp. 155-170, Feb. 2016.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2019.2929009, IEEE Transactions on Multimedia



Ke Gu (M'19) received the B.S. and Ph.D. degrees in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2009 and 2015, respectively. He is currently a Professor with the Beijing University of Technology, Beijing, China. His research interests include environmental perception, image processing, quality assessment, and machine learning. He received the Best Paper Award from the IEEE Transactions on Multimedia (T-MM), the Best Student Paper Award at the IEEE International Conference on Multimedia and Expo (ICME) in 2016, and the Excellent Ph.D. Thesis Award from the Chinese Institute of Electronics in 2016. He was the Leading Special Session Organizer in the VCIP 2016 and the ICIP 2017, and serves as a Guest Editor for the Digital Signal Processing (DSP). He is currently an Associate Editor for the IEEE ACCESS and IET Image Processing (IET-IPR), and an Area Editor for the Signal Processing Image Communication (SPIC). He is a Reviewer for 20 top SCI journals.

Zhifang Xia received the B.S. degree in measuring and control instrument from Anhui University, Hefei, China in 2008 and received the Master degree in control science and engineering from Tsinghua University, Beijing, China in 2012. She is currently an engineer and a registered consultant (investment) with State Information Center, Beijing, China, and is currently pursuing the Ph.D. degree with Beijing University of Technology, Beijing, China. Her interests include image processing, quality assessment, machine learning and e-government. She won the second prize of National excellent engineering consultation award in 2016.

Junfei Qiao (M'11) received the B.E. and M.E. degrees in control engineer from Liaoning Technical University, Fuxin, China, in 1992 and 1995, respectively, and the Ph.D. degree from Northeast University, Shenyang, China, in 1998. He was a Post-Doctoral Fellow with the School of Automatics, Tianjin University, Tianjin, China, from 1998 to 2000. He joined the Beijing University of Technology, Beijing, China, where he is currently a Professor. He is the Director of the Intelligence Systems Laboratory. His current research interests include neural networks, intelligent systems, self-adaptive/learning systems, and process control systems. Prof. Qiao is a member of the IEEE Computational Intelligence Society. He is a Reviewer for more than 20 international journals, such as the IEEE Transactions on Fuzzy Systems and the IEEE Transactions on Neural Networks and Learning Systems.

Weisi Lin (F'16) received the Ph.D. degree from Kings College London. He is currently a Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include image processing, visual quality evaluation, and perception-inspired signal modeling, with more than 340 refereed papers published in international journals and conferences. He has been on the Editorial Board of the IEEE T-IP, T-CSVT, T-MM, SPL, and JVCI. He has been elected as APSIPA (2012/2013) Distinguished Lecturers. He served as a Technical-Program Chair for Pacific-Rim Conference on Multimedia 2012, the IEEE International Conference on Multimedia and Expo 2013, and the International Workshop on Quality of Multimedia Experience 2014. He is a fellow of Institution of Engineering Technology, an Honorary Fellow of the Singapore Institute of Engineering Technologists, and a Fellow of IEEE.