Generating Realistic Smoke Images With Controllable Smoke Components

CHAO XIE¹ AND HUANJIE TAO²

¹College of Mechanical and Electronic Engineering, Nanjing Forestry University, Nanjing 210037, China
²School of Computer Science, Northwestern Polytechnical University, Xi’an 710072, China

Corresponding author: Huanjie Tao (huanjie_tao@126.com)

This work was supported by the National Natural Science Foundation of China under Grant 61901221 and 52005265.

Abstract
Smoke image generation is an important method to solve sample categories imbalance in smoke detection with applications such as forest smoke detection, smoky vehicles detection, etc. This article presents two Controllable Smoke Image Generation Neural Networks (CSGNet and CSGNet-v2). More specifically, CSGNet is proposed to generate various smoke images by integrating a smoke component control module (SCM) and a smoke image synthesis module (ISM) with a multi-scale strategy. By changing specified smoke components latent codes, we can generate smoke images with specified smoke components. By fine-tuning smoke components latent codes in SCM, we can fine-tune the smoke components in generated smoke images. To further improve CSGNet, CSGNet-v2 is proposed to make generated smoke images more realistic by adding a smoke image fine-tuning module (IFM). The experiment results show that our methods achieve by far the best results in generating smoke images with controllable smoke components.

Index Terms
Controllable smoke components, smoke image generation, smoke image synthesis.

I. INTRODUCTION
Smoke image generation is a very important method to solve sample categories imbalance in smoke detection with applications such as forest smoke detection, smoky vehicle detection [1], [2], etc., in which smoke samples are much lower than non-smoke samples. This article focuses on smoke images generation of the vehicle rear regions, and our methods can also be transplanted to other applications to generate images that contain transparent substances.

Smoke image generation is a subtopic of image generation, which can be divided into two categories: Generative Adversarial Networks (GANs)-based methods [3] and Variational Auto-encoder (VAE)-based methods [4]. GANs-based methods often faced the problem of training stability. VAEs-based methods often faced the problem that the generated images are blurred.

Different from all existing methods, we give a new idea to generate smoke images based on deep observations on smoke characteristics. Smoke is usually composed of many tiny particles, which scatter and absorb lights from light sources or environmental reflections [5]. Lights gradually attenuate in the air, and finally enter into a camera to generate an image. This makes the smoke usually has transparency, and also means that smoke images can be synthetized from non-smoke components and smoke components.

Based on the above considerations, we proposed two Controllable Smoke Image Generation Neural Networks (CSGNet and CSGNet-v2). The main contributions are summarized as follows:

(1) CSGNet is proposed to generate various smoke images by integrating a smoke component control module (SCM) and a smoke image synthesis module (ISM) with a multi-scale strategy. By adjusting smoke components latent codes in SCM, we control the smoke components in generated smoke images.

(2) CSGNet-v2 is proposed to make generated smoke images more realistic by adding a smoke image fine-tuning module (IFM). The smoke in generated smoke images are controlled by adjusting smoke components latent codes in SCM.

(3) Three main advantages: a) Our methods can control the smoke in generated smoke images to generate various smoke images or sequences, such as generate specified light smoke images, but existing methods cannot. b) Our method can generate smoke images as well as their smoke...
components images which are useful in smoke detection, smoke segmentation, smoke pollution levels estimation for smoky vehicle, and smoke motion characteristic estimation for forest smoke etc., but existing methods cannot. c) With the idea of generating smoke images by adding various smoke components to existing non-smoke images, the training step is more stable compared with GANs-based methods, the generated smoke images are less blurry compared with VAEs-based methods, and the non-smoke components in generated smoke image are more realistic since they come from the real world.

This article is organized as follows. Section II introduces related work on image generation. Section III describes the proposed CSGNet to generate smoke image with controllable smoke components. Section IV describes the proposed CSGNet-v2 to generate more realistic smoke images. Extensive experiments are given in Section V. At last, we conclude the paper in Section VI.

II. RELATED WORK
In this section, we introduce related work on image generation, especially these methods that can generate images with some characteristics controllable. We roughly divided existing image generation methods into two categories: GANs-based methods and VAEs-based methods.

A. GANS-BASED METHODS
The original GAN [3] was first proposed by Goodfellow in 2014 and used for image generation. Currently many GANs-based variants are proposed, but we mainly introduce these methods that can generate images with some characteristics controllable.

Chen et al. [6] proposed information GAN (InfoGAN) that is an information-theoretic extension to GAN to learn disentangled representations in a completely unsupervised manner, such as writing styles from digit shapes on the MNIST dataset [7], etc. Mirza and Osindero [8] proposed the conditional version of GAN (CGAN) constructed by simply feeding the data we wish to condition on to both the generator and discriminator, such as generate MNIST digits conditioned on class labels. Li et al. [9] proposed the triple GAN (Triple-GAN) based on a flexible game-theoretical framework for classification and class-conditional generation in semi-supervised learning to disentangle the classes and styles and transfer smoothly on the data level via interpolation on the latent space class-conditionally. Borrowing from style transfer literature, Tero et al. [10] proposed an alternative generator architecture for GAN to achieve an automatically learned, unsupervised separation of high-level attributes (pose and identity on human faces) and stochastic variation in the generated images (freckles, hair). Zhu et al. [11] proposed CycleGAN to learn a mapping to translate an image from a source domain to a target domain in the absence of paired examples using an adversarial loss, such as collection style transfer, object transfiguration, season transfer, photo enhancement, etc. Choi et al. [12] proposed StarGAN to perform image-to-image translations for multiple domains based on simultaneous training of multiple datasets with different domains within a single network. Shaham et al. [13] proposed the single GAN (SinGAN) that can be learned from a single natural image to capture the internal distribution of patches within the image, and generate high quality, diverse samples that carry the same visual content as the image.

B. VAES-BASED METHODS
The original VAE [4] was proposed by Kingma in 2014 and also used for image generation. Currently many VAEs-based variants are proposed.

Sohn et al. [14] proposed a scalable deep conditional generative model (CVAE) for structured output variables using Gaussian latent variables. Bao et al. [15] proposed the variational GAN that combines a VAE with GAN (VAE-GAN) for synthesizing images of fine-grained categories, such as faces of a specific person or objects in a category. Huang et al. [16] proposed the introspective variational autoencoder (IntroVAE) model for synthesizing high-resolution photo-realistic photographic images with stable training and nice latent manifold. Higgins et al. [17] proposed the beta-VAE for automated discovery of interpretable factorized latent representations from raw image data in a completely unsupervised manner by introducing an adjustable hyperparameter beta that balances latent channel capacity and independence constraints with reconstruction accuracy. Xiao et al. [18] proposed a supervised learning model called DNA-GAN to disentangle different factors of images with DNA-like latent representations of images, in which each individual piece (of the encoding) represents an independent factor of the variation. Burda et al. [19] proposed the importance weighted autoencoder (IWAE) with the same architecture as the VAE, but used a strictly tighter log-likelihood lower bound derived from importance weighting to learn richer latent space representations than VAEs. Larsen et al. [20] proposed an autoencoder that leverages learned representations to better measure similarities in data space by combining a VAE with a GAN and use learned feature representations in the GAN discriminator as basis for the VAE reconstruction objective.

Currently various GANs-based methods and VAEs-based methods are proposed and have achieved great success. However, these methods still have some disadvantages in our controllable smoke images generation task: 1) These methods cannot generate smoke images with controllable smoke components. 2) These methods cannot generate smoke images as well as their smoke components images which are useful in smoke detection and smoke segmentations. 3) The smoke images generated by VAEs-based methods are usually blurred.

To solve all the above issues, this article proposes the CSGNet model. To further improve the authenticity of the smoke images generated by CSGNet, this article proposes the CSGNet-v2 model.
III. CONTROLLABLE SMOKE IMAGE GENERATION NEURAL NETWORKS

In this section, we introduce the CSGNet in detail including network structure, smoke components control module (SCM), smoke image synthesis module (ISM) and loss function.

A. NETWORK STRUCTURE

The whole network structure of the CSGNet model can be seen in Fig.1.

FIGURE 1. The network structure of CSGNet.

The proposed CSGNet consists of two main modules including smoke components control module (SCM), and smoke image synthesis module (ISM). SCM is used to learn the potential distribution of smoke components, and ISM is used to synthetize smoke images via non-smoke components images and smoke components images.

In the training, non-smoke components images, smoke components images and their manually synthesized smoke images are given. We first put smoke components images into SCM to obtain decoded smoke components. Then we put non-smoke components images and smoke components images into ISM together to generate smoke images.

In the testing, only non-smoke images are given. We first randomly generate a smoke components latent code vector, and put it into the smoke components decoder to obtain a smoke components image. Then we put the non-smoke components image and smoke components image into ISM together to generate a new smoke image.

B. SMOKE COMPONENT CONTROL MODULE (SCM)

SCM is used to learn potential distribution of smoke components. We first put smoke components images into this module to obtain the mean vector and the logarithm of variance vector, which can form a Gaussian distribution. Then we sampled from this distribution to obtain smoke components latent codes. Finally, we decode the latent codes to obtain a smoke components image.

A detail design of SCM can be seen in Fig.2. By adjusting each dimension of the latent code vector, we can control the reconstructed smoke components.

C. SMOKE IMAGE SYNTHESIS MODULE (ISM)

ISM is used to synthetize smoke images using non-smoke components images and smoke components images.

D. LOSS FUNCTION

The loss function of CSGNet is given by

\[ L_{\text{CSGNet}} = \frac{N}{M} \sum_{i=1}^{M} (\lambda_{\text{SCM}} L_{\text{SCM}} + \lambda_{\text{ISM}} L_{\text{ISM}}) \]  

where \( N \) is the total training samples number, and \( M \) is the training batch size. \( \lambda_{\text{SCM}} \) and \( \lambda_{\text{ISM}} \) are weight coefficients of different terms, respectively. \( L_{\text{SCM}} \) and \( L_{\text{ISM}} \) are the loss terms of SCM and ISM, respectively.

The terms \( L_{\text{SCM}} \) and \( L_{\text{ISM}} \) are given by

\[ L_{\text{SCM}} = -\lambda_{\text{SCM}} \mathbb{E}_{q_{\phi}(z|x)} [\log p_{\theta}(x|z)] + \lambda_{\text{SCM}} D_{KL}(q_{\phi}(z|x)||p_{\theta}(z)) \]

\[ L_{\text{ISM}} = \lambda_{\text{ISM}} \mathbb{E}_{q_{\phi}(z|x)} [\log p_{\theta}(x|z)] + \lambda_{\text{ISM}} D_{KL}(q_{\phi}(z|x)||p_{\theta}(z)) \]
\[ L_{ISM} = \left\| O_{ISM}^{\text{fusion}} - I_{ISM}^{\text{fusion}} \right\| \] (3)

where \( x \) is input data includes non-smoke components image \( I^{\text{non-smoke}} \), smoke components image \( I^{\text{smoke}} \) and synthesized smoke image \( I_{ISM}^{\text{smoke}} \) at the output of \( O_{ISM}^{\text{smoke}} \) are used to learn mapping functions between an \( X \) domain image and \( Y \) domain images and real smoke images, \( G_{1}:X \rightarrow Y \) and \( G_{2}: Y \rightarrow X \) are used to learn mapping functions between two kinds of images of different domains including the synthetic smoke images and real smoke images. \( \lambda_{SCM1} \) and \( \lambda_{SCM2} \) are weight coefficients of smoke components reconstruction error \( \left\| O_{SCM}^{\text{smoke}} - I^{\text{smoke}} \right\| \) and KL divergence error \( D_{KL}(q_{\theta}(z|x)||p_{\theta}(z)) \) that measures the similarity between \( q_{\theta}(z|x) \) and \( p_{\theta}(z) \). The weights of the different loss terms are set to 1.

Given smoke components latent codes \( z \) and observation data \( x \), \( x \rightarrow z \) is described by probability distribution \( q_{\theta}(z|x) \). \( z \rightarrow x \) is described by probability distribution \( p_{\theta}(x|z) \). The parameter \( q_{\theta}(z|x) \) in equation (2) is a probability distribution used to approximate \( p_{\theta}(z|x) \) (calculated by \( p_{\theta}(x|z) \)). \( p_{\theta}(x|z) \) is a distribution that can be set to Bernoulli distribution or Gaussian distribution.

**IV. CONTROLLABLE SMOKE IMAGE GENERATION NEURAL NETWORKS-V2**

In this section, we introduce CSGNet-v2 in detail including network structure, smoke image fine-tuning module (IFM), and loss function.

**A. NETWORK STRUCTURE**

The whole network structure of CSGNet-v2 can be seen in Fig.4.

**FIGURE 4.** The network structure of CSGNet-v2. \( X \) and \( Y \) denote two kinds of images of different domains including the synthetic smoke images and real smoke images, \( G_{1}:X \rightarrow Y \) and \( G_{2}: Y \rightarrow X \) denote two learned mapping functions between an \( X \) domain image and \( Y \) domain images. This part of IFM is inspired from [11], \( D_{1}(Y) \) and \( D_{2}(X) \) are used to judge the input image belongs to \( X \) or \( Y \).

The CSGNet-v2 consists of two main modules including the CSGNet and the IFM. CSGNet is used to generate controllable smoke images, and the IFM is used to make the generated smoke images more realistic.

**B. SMOKE IMAGE FINE-TUNING MODULE (IFM)**

IFM is inspired from [11], and consists of four parts including two learned mapping functions (\( G_{1}:X \rightarrow Y \), \( G_{2}: Y \rightarrow X \)) and two discriminators \( (D_{1}(X), D_{2}(Y)) \).

Given a non-smoke image, we can use CSGNet to synthesize various smoke images. But these smoke images have synthesized feeling and trace. To make it more realistic, the IFM is adopted.

We first put a synthetic smoke image \( I^{\text{synthesis}} \) to \( G_{1}:X \rightarrow Y \) to obtain a translated smoke image \( G_{1}(I^{\text{synthesis}}) \). Then we put \( G_{1}(I^{\text{synthesis}}) \) to \( D_{1}(y) \) to judge it belongs to \( X \) or \( Y \), and also put \( G_{1}(I^{\text{synthesis}}) \) to \( G_{2}: Y \rightarrow X \) to make its output as close as possible to \( I^{\text{synthesis}} \). Simultaneously, we put the real smoke images \( I^{\text{real}} \) to \( G_{2}: Y \rightarrow X \) to obtain the output \( G_{2}(I^{\text{real}}) \). Then we put \( G_{2}(I^{\text{real}}) \) to \( D_{2}(x) \) to judge it belongs to \( X \) or \( Y \), and also put \( G_{2}(I^{\text{real}}) \) to \( G_{1}: X \rightarrow Y \) to make its output as close as possible to \( I^{\text{real}} \), respectively. Training in this way, the \( G_{1}:X \rightarrow Y \) can transform a synthetic smoke image to a realistic smoke image.

\( G_{1} \) is used to generate images that look similar to images from domain \( Y \), while \( D_{1}(Y) \) is used to distinguish between translated samples from \( G_{1}(X) \) and real samples \( y \). \( G_{1} \) aims to minimize this objective against an adversary \( D_{1}(Y) \) that tries to maximize it.

**C. LOSS FUNCTION**

The loss function of CSGNet-v2 is given by

\[ L_{CSGNet-v2} = L_{CSGNet} + \arg \min_{G_{1}, G_{2}, D_{1}, D_{2}} \max L(G_{1}, G_{2}, D_{1}, D_{2}) \] (4)

where \( L(G_{1}, G_{2}, D_{1}, D_{2}) \) contains adversarial losses and cycle consistency losses. The adversarial losses is used to match the distribution of generated images to the data distribution in the target domain; and the cycle consistency losses are used to prevent the learned mappings \( G_{1}(X) \) and \( G_{2}(Y) \) from contradicting each other. The weights of the different loss terms are set to 1.

The term \( L(G_{1}, G_{2}, D_{1}, D_{2}) \) can be expressed by

\[ L(G_{1}, G_{2}, D_{1}, D_{2}) = \lambda_{GAN1}L_{GAN1} + \lambda_{GAN2}L_{GAN2} + \lambda_{cyc}L_{cyc} \] (5)

where \( \lambda_{GAN1}, \lambda_{GAN2} \) and \( \lambda_{cyc} \) are weight coefficients of adversarial loss \( L_{GAN1} \), adversarial loss \( L_{GAN2} \) and cycle consistency loss \( L_{cyc} \).

The terms of \( L_{GAN1}, L_{GAN2} \) and \( L_{cyc} \) can be expressed by

\[ L_{GAN1} = \mathbb{E}_{x \sim p_{data}(x)}[\log D_{1}(y)] + \mathbb{E}_{y \sim p_{data}(y)}[\log(1 - D_{1}(G_{1}(x)))] \] (6)

\[ L_{GAN2} = \mathbb{E}_{x \sim p_{data}(x)}[\log D_{2}(x)] + \mathbb{E}_{y \sim p_{data}(y)}[\log(1 - D_{2}(G_{2}(y)))] \] (7)

\[ L_{cyc} = \mathbb{E}_{x \sim p_{data}(x)}[(G_{2}(G_{1}(x)) - x)_{1}] + \mathbb{E}_{y \sim p_{data}(y)}[(G_{1}(G_{2}(y)) - y)_{1}] \] (8)

where \( G_{1}(X) \) and \( G_{2}(Y) \) are two mapping functions between synthesis smoke image domain \( X \) and real smoke image.
domain $Y$. $x \sim p_{data}(x)$ and $y \sim p_{data}(y)$ denote the synthesis smoke training images $\{x_i\}_{i=1}^{N_1}$, $x_i \in X$ and real smoke training images $\{y_i\}_{i=1}^{N_2}$, $y_i \in Y$. $D_1(Y)$ and $D_2(X)$ are two adversarial discriminators. $D_1(Y)$ is used to distinguish between synthesis smoke images $x$ and translated images $G_1(X)$, and $D_2(X)$ is used to distinguish between real smoke images $y$ and translated images $G_2(X)$.

To obtain a better model, we first train CSGNet model based on loss $L_{CSGNet}$ in (1). Then we train the IFM module in CSGNet-v2 based on loss $L(G_1, G_2, D_1, D_2)$ in (5). Finally, we train CSGNet and the IFM module together based on loss $L_{CSGNet-v2}$ in (4).

V. EXPERIMENTS AND ANALYSIS

In this section, extensive experiments are reported to verify the advantages of our methods. Firstly, we introduce the used datasets in this article. Secondly, experiments on smoke image generation are introduced. Thirdly, experiments on smoke image generation with controllable smoke components are introduced. Fourthly, experiments on smoke components estimation in the generated smoke images are introduced. Fifthly, comparison experiments with image components estimation in the generated smoke images are done. Sixthly, comparison experiments on smoke image generation with controllable smoke components are introduced. Lastly, experiments on smoke image generation with controllable smoke components latent codes in SCM. $L$ denotes the dimension of latent codes, and $a_i(i = 1, 2, \ldots, L)$ denotes the $i$-th dimension. In each dimension, the change range is $[-2, 2]$, and we sample $B$ numerical values linearly. For $M$ non-smoke images, the total number of smoke image kinds is

$$I_{total} = MB^L$$

(9)

If we train $K$ models by setting $L$ to 1, 2, ..., $K$, the total number of smoke image kinds is

$$I_{total2} = M(B^1 + B^2 + \ldots + B^L) = \frac{MB^L - 1}{B - 1}, \quad B \neq 1$$

(10)

We set $L$ to 10. Given non-smoke images and randomly smoke components latent codes, Fig.7 shows some generated smoke images using CSGNet and CSGNet-v2. We can see that our methods can generate various smoke images. The non-smoke components in Fig.7(a) are clear, and the smoke in Fig.7(b) are realistic.

C. EXPERIMENTS ON SMOKE IMAGE GENERATION WITH CONTROLLABLE SMOKE COMPONENTS

Different from existing methods, the smoke components in the generated smoke images using our methods are controllable, i.e., the smoke components can be designed or fine-tuned manually with non-smoke components unchanged.

Given a non-smoke image, our method can generate smoke images with a designed smoke component by using designed latent codes selected from $I_{total}$ kind different smoke components. In addition, we can fine-tune the smoke components by fine-tuning smoke components latent codes.

Fig. 8 shows some generated smoke images by linearly adjusting the last four dimensions of smoke latent

### TABLE 1. Descriptions of the datasets used in this article.

| Datasets       | Descriptions                                           |
|----------------|--------------------------------------------------------|
| Synthetic      | 8400 groups of images. Each group: a simulated smoke  |
| dataset        | components image, a non-smoke components image, and a  |
|                | synthetic smoke image.                                 |
| Real dataset   | 2133 smoke images. These images are cropped from the   |
|                | rear regions of the real smoke vehicles.               |

![Figure 5. Some groups of images from synthetic dataset. (a) Simulated smoke components image. (b) Non-smoke components image. (c) Synthetic smoke image. (d), (e) and (f) are similar with (a), (b) and (c).](image1)

![Figure 6. Some real smoke images from real dataset.](image2)
codes and keeping the other to 0. Our methods can control/fine-tune the smoke components in the generated smoke images while keeping the non-smoke components unchanged. Also, the texture of the road surface can be seen clearly.

Fig. 9 shows some generated smoke images by linearly adjusting the first two dimensions of smoke components latent codes and keeping the other to -1. We can see that our methods are effective to control the added smoke components to the given non-smoke images.

In this selected case, the generated smoke images using CSGNet-v2 are more realistic than CSGNet. But the image clarity of using CSGNet are better than CSGNet-v2.

D. EXPERIMENTS ON SMOKE COMPONENT ESTIMATION IN GENERATED SMOKE IMAGES USING CSGNET

Different from existing methods, CSGNet can generate smoke images as well as their smoke components images which are useful in many applications.

Fig.10 given the smoke components of the generated smoke images in Fig.8 and Fig.9. From this figure, we can clearly see the smoke components characteristics of the generated smoke images.

The smoke in Fig.10(a) is darker than that in Fig.10(b), and the smoke in center of Fig.10(a) is higher than that in edge. These indicate that the bigger the absolute values of latent codes, the higher the smoke components.

E. COMPARISON EXPERIMENTS WITH IMAGE QUALITY OF THE GENERATED SMOKE IMAGES

This section evaluates image quality of generated smoke images.

When smoke images are randomly generated, the phenomenon of generating repeated samples is called mode
C. Xie, H. Tao: Generating Realistic Smoke Images With Controllable Smoke Components

FIGURE 9. Generating smoke images with controllable smoke components by linearly adjusting the first two dimensions of smoke latent codes and keeping the others to -1. The image resolution of all the generated images is $128 \times 128$ pixels. (a) A non-smoke image. (b) Using CSGNet. (c) Using CSGNet-v2.

collapse. The reason for this issue is that there are many peaks in the actual data distribution, but only a few of them are fitted during training. In fact, the proposed models do not have this issue. The diversity of generated smoke image from CSGNet is mainly based on the diversity of background images and smoke component images. There are sufficient and diverse background images, and the smoke component images are also various as described in V(B). Therefore, the diversity of generated smoke images can be guaranteed.

Some quantitative evaluation indicators are used to evaluate the image quality of the proposed models. Currently, IS (Inception Score) [23], FID (Frechet Inception distance) [24], GAN-test [25], GAN-train [25], Kernel MMDs (Maximum Mean Discrepancies) [26] and 1-NN (Nearest Neighor) two-sample tests [27] are important strategies to evaluate image quality. Xu et al. [28] proposed a conclusion.

FIGURE 10. The generated smoke components images of the generated smoke image. (a) Generated smoke components images of the smoke images in Fig.8(a). (b) Generated smoke components images of the smoke images in Fig.9(a).
that Kernel MMD [26] and 1-NN two-sample tests [27] are the best evaluation indicators.

Table 2 shows the image quality evaluation results of the smoke images generated by our methods and the WGAN-GP [29]. Since the mainstream model WGAN-GP is stable in training and can generate realistic samples of considerable quality, we take WGAN-GP as the baseline method. The proposed methods perform better than WGAN-GP in three evaluation indexes, and this shows that the generated smoke image quality by the proposed model is guaranteed, and it is close to the real black smoke image.

TABLE 2. The image quality evaluation results of the smoke images generated by our methods and the WGAN-GP.

| Different image quality evaluation indicators* | FID [24] | Kernel MMD [26] | 1-NN [27] |
|-----------------------------------------------|---------|-----------------|----------|
| Real smoke images                             | 2.9     | ---             | ---      |
| WGAN-GP [29]                                  | 21.0    | 1.96            | 0.41     |
| CSGNet                                        | 18.7    | 1.55            | 0.43     |
| CSGNet-v2                                     | 7.2     | 0.94            | 0.47     |

*For evaluation indicators FID and Kernel MMD, the smaller the better. For the evaluation indicator 1-NN, the closer to 0.5, the better.

F. COMPARISONS EXPERIMENTS WITH EXISTING METHODS

Some existing methods are transplanted to smoke image generation to verify the advantages of our methods.

The main advantages of our methods are 1) can generate smoke images with controllable smoke components and 2) generate smoke images as well as their smoke components images. Since we are the first one to achieve advantage 2), we only select some advanced methods related to feature decoupling to make comparisons with our methods.

Based on the discussion in section II, InfoGAN [6], CycleGAN [11], beta-VAE [17], and IntroVAE [16] are used as baseline methods. It should be noted that some non-core parts of these baseline methods are improved to make them adapted to our datasets and task, and we also tried our best to achieve the best results on our task to achieve fair comparisons. Fig.11 shows the comparison experiments of different methods on controllable smoke images generation.

Fig.11(a) shows the results or by adjusting the latent codes $c$ [6] in InfoGAN [6]. It seems that $c$ controls the smoke components. But the generated smoke images are not realistic. Also, we cannot generate more images since we set class number to 10. Fig.11(b) shows the results using CycleGAN [11]. But we cannot fine-tune the smoke components in the generated smoke images. This may lead to samples imbalance of different categories, since it tends to generate images with heavy smoke. Fig.11(c) shows the results of beta-VAE [17]. None of the ten dimensions controls the smoke components with non-smoke components unchanged. Fig.11(d) shows the results of IntroVAE [16]. We give a smoke image and its non-smoke components, and use IntroVAE to achieve images interpolation. But the images
must in its trained datasets, and the smoke diversity of smoke is limited.

In addition, the training of the first two (GANs-based) methods are instable, and the generated images in the last two (VAEs-based) methods are blurred. Based on the above analysis, we can see that our methods are better than all the baselines methods in controllable smoke images generation.

The advantages of our methods are as follows:

Our methods can generate smoke images as well as their smoke components images. Using the datasets generated by our methods, we can train models to estimate smoke components. The smoke components images are useful in many applications. 1) In smoke detection, if the smoke components in smoke images are higher than a fixed threshold, we can classify this image to smoke. This is a new way to detect smoke. 2) In smoke segmentation, we can segment smoke in different levels by setting different levels of thresholds on the estimated smoke components images using one trained model. But traditional deep learning methods need to label multiple datasets and train multiple models of different levels. 3) In smoke pollution levels estimation, by analyzing the smoke components, we can give a pollution level estimation of the smoke images in the application of smoky vehicle detection. 4) In forest smoke analysis, by analyzing the motion characteristics of smoke components in successive frames, we can forecast the smoke fire burning direction and trend in forest smoke detection. All the above important applications are based on datasets of smoke images as well their corresponding smoke components images. Therefore, generating smoke images and their smoke components images are very useful.

Our methods can control the smoke components with non-smoke components unchanged. This is also useful in many applications. 1) In some special senses with specified smoke characteristics, we should generate some specified kind of smoke images, such as light smoke images, etc. Our method can generate these kinds of smoke images by setting specified smoke components latent codes. 2) Since our methods can generate smoke images with non-smoke components unchanged, we can generate short smoke videos by fine-tuning smoke components latent codes, such the sense of forest smoke which the non-smoke components are almost non-changed. 3) For the same non-smoke components, we can add various smoke components to it through adjusting smoke latent codes, and so increase the diversity of smoke images.

Our methods to generate smoke images by adding various smoke components to existing non-smoke images have some advantages. 1) Compared with GANs-based methods, CSGNet avoids the training instability. 2) Compared with the VAEs-based methods, the generated smoke images are less blurry, and a new non-smoke image can be added to our model without training again. 3) The non-smoke components in the generated smoke image are come from the real world, and so the images are more realistic. The smoke components may be non-realistic, but we can fine-tune it using CSGNet-v2.

G. SMOKE COMPONENTS ESTIMATION

Our methods can generate smoke images as well as their smoke components images. Using the datasets generated by our methods, we can train models to estimate smoke components. The smoke components images are useful in many applications.

Fig.12 shows some experimental results of smoke components estimation of real smoke images using the synthetic smoke images in this article. We can see that the smoke position in the smoke image exactly corresponds to the smoke component position in the smoke component image. The higher smoke concentration in the smoke image, the higher smoke concentration in the smoke component image. This verifies the effectiveness of our generated datasets.

VI. DISCUSSION AND CONCLUSION

Smoke image generation is an important method to solve sample categories imbalance in smoke detection. This article presents two smoke image generation models including CSGNet and CSGNet-v2. CSGNet is proposed to generate various smoke images by integrating the SCM and the ISM with a multi-scale strategy. By adjusting smoke components latent codes in SCM, we can control smoke components in generated smoke images. CSGNet-v2 is proposed to make generated smoke images more realistic by adding the IFM. Compared with existing methods, our methods have three main advantages: 1) can control the smoke components in generated smoke images to generate various smoke images or videos. 2) can generate smoke images as well as their smoke components images which are useful in smoke detection, smoke segmentation, smoke pollution levels estimation for smoky vehicle, and smoke motion characteristic estimation for forest smoke etc. 3) with the idea of generating smoke images by adding various smoke components to existing non-smoke images, the training of our models is stable compared with GANs-based methods, the generated smoke images are...
less blurry compared with the VAEs-based methods, and the non-smoke components in the generated smoke image are more realistic since they come from the real world.

Our methods achieve some success in smoke images generation and can improve the performance in forest smoke detection. But our methods still have some shortcomings. In special cases, the generated smoke images using CSGNet are not realistic.

REFERENCES

[1] P. Pyykonen, P. Peussa, M. Kutila, and K.-W. Fong, “Multi-camera-based smoke detection and traffic pollution analysis system,” in Proc. IEEE 12th Int. Conf. Intell. Comput. Commun. Process. (ICCP), Sep. 2016, pp. 233–238.

[2] H. Tao and X. Lu, “Automatic smoky vehicle detection from traffic surveillance video based on vehicle rear detection and multi-feature fusion,” IET Intell. Transp. Syst., vol. 13, no. 2, pp. 252–259, Feb. 2019.

[3] I. Goodfellow, A. J. Pouget, and M. Mirza, “Generative adversarial nets,” in Proc. Adv. Neural Inf. Process. Syst., 2014, pp. 2672–2680.

[4] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” 2013, arXiv:1312.6114. [Online]. Available: http://arxiv.org/abs/1312.6114

[5] T. James Kajiya and P. Brian von Herzen, “Ray tracing volume densities,” ACM SIGGRAPH Comput. Graph., vol. 18, no. 3, pp. 165–174, 1984.

[6] X. Chen, Y. Duan, and R. Houthooft, “InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets,” in Proc. Adv. Neural Inf. Process. Syst., 2016, pp. 2172–2180.

[7] 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.

[8] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” 2014, arXiv:1411.1784. [Online]. Available: http://arxiv.org/abs/1411.1784

[9] C. Li, K. Xu, and J. Zhu, “Triple generative adversarial nets,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 4088–4098.

[10] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 4404–4410.

[11] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired Image-to-Image translation using cycle-consistent adversarial networks,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2223–2232.

[12] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, “StarGAN: Unified generative adversarial networks for multi-domain Image-to-Image translation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 8789–8797.

[13] T. R. Shaham, T. Dekel, and T. Michaeli, “SinGAN: Learning a generative model from a single natural image,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 4570–4580.

[14] K. Sohn, X. C. Yan, and H. Lee, “Learning structured output representation using deep conditional generative models,” in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 3483–3491.

[15] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua, “CVAE-GAN: Fine-grained image generation through asymmetric training,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2745–2754.

[16] H. Huang, Z. Li, and R. He, “IntroVAE: Introspective variational autoencoders for photographic image synthesis,” in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 52–63.

[17] I. Higgins, L. Matthey, and A. Pal, “Beta-VAE: Learning basic visual concepts with a constrained variational framework,” in Proc. ICLR, 2017, pp. 1–22.

[18] T. Xiao, J. Hong, and J. Ma, “DNA-GAN: Learning disentangled representations from multi-attribute images,” in Proc. ICLR, 2018, pp. 1–17.

[19] Y. Burda, R. Grosse, and R. Salakhutdinov, “Importance weighted autoencoders,” 2015, arXiv:1509.00519. [Online]. Available: http://arxiv.org/abs/1509.00519

[20] A. B. L. Larsen, S. K. Sønderby, and W. Winther, “Autoencoding beyond pixels using a learned similarity metric,” in Proc. 33rd Int. Conf. Mach. Learn. (PMLR), 2016, pp. 1558–1566.

[21] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 2881–2890.

[22] X. Yan, Y. Liu, and M. Jia, “A fault diagnosis approach for rolling bearing integrated SGMID, IMSDE and multiclass relevance vector machine,” Sensors, vol. 20, no. 15, p. 4352, Aug. 2020.

[23] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training GANs,” in Proc. Adv. Neural Inf. Process. Syst., 2016, pp. 2234–2242.

[24] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs trained by a two time-scale update rule converge to a local Nash equilibrium,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 6626–6637.

[25] K. Shmelkov, C. Schmid, and K. Alahari, “How good is my GAN?” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 213–229.

[26] W. Boulniphine, E. Beilisovsky, M. B. Blaschko, I. Antonoglou, and A. Gretton, “A test of relative similarity for model selection in generative models,” 2015, arXiv:1511.04581. [Online]. Available: http://arxiv.org/abs/1511.04581

[27] D. Lopez-Paz and M. Oquab, “Revisiting classifier two-sample tests,” 2016, arXiv:1610.06545. [Online]. Available: http://arxiv.org/abs/1610.06545

[28] Q. Xu, G. Huang, Y. Yuan, C. Guo, Y. Sun, F. Wu, and K. Weinberger, “An empirical study on evaluation metrics of generative adversarial networks,” 2018, arXiv:1806.07755. [Online]. Available: http://arxiv.org/abs/1806.07755

[29] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved training of Wasserstein GANs,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 5767–5777.

C. Xie, H. Tao: Generating Realistic Smoke Images With Controllable Smoke Components

CHAO XIE received the Ph.D. degree from Southeast University, Nanjing, China, in 2018. He is currently a Lecturer with the College of Mechanical and Electronic Engineering, Nanjing Forestry University, Nanjing. His current research interests include image processing and deep learning.

HUANJIE TAO received the M.S. degree in mathematics and information science from Capital Normal University, Beijing, China, in 2016, and the Ph.D. degree in pattern recognition and intelligent system from Southeast University, Nanjing, China, in 2020. He was a Visiting Scholar with the School of Computer Science and Engineering, Nanjing Technological University, from September 2018 to September 2019. He is currently employed as an Associate Professor at the School of Computer Science, Northwestern Polytechnical University (NPU), Xi’an, China. His current research interests include image processing, computer vision, feature extraction, deep learning, and pattern recognition.