IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 30, 2019, accepted January 8, 2020, date of publication March 4, 2020, date of current version March 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2968612

Generative Adversarial Ensemble Learning for Face Forensics

JAE-YONG BAEK[1], YONG-SANG YOO[2], AND SEUNG-HWAN BAE[3], (Member, IEEE)

[1]R&D Center, Autonomous A2Z, Gyeongsan 14057, South Korea
[2]Department of Computer Science and Engineering, Incheon National University, Incheon 22012, South Korea
[3]Department of Computer Engineering, Inha University, Incheon 22212, South Korea

Corresponding author: Seung-Hwan Bae (shbae@inha.ac.kr)

This work was supported in part by the INHA UNIVERSITY Research Grant and in part by the National Research Foundation of Korea (NRF) Grant funded by the Korea Government (MSIT) under Grant NRF-2018R1C1B6003785.

ABSTRACT The recent advance of synthetic image generation and manipulation methods allows us to generate synthetic face images close to real images. On the other hand, the importance of identifying the synthetic face images increases more and more to protect personal privacy from those. Although some deep learning-based image forensic methods have been developed recently, it is still challenging to distinguish synthetic images generated by recent image generation and manipulation methods such as the deep fake, face2face, and face swap. To resolve this challenge, we propose a novel generative adversarial ensemble learning method. We train multiple discriminative and generative networks based on the adversarial learning. Compared to the conventional adversarial learning, our method is however more focused on improving the discrimination ability rather than image generation one. To this end, we improve the discriminabilty by ensembling outputs from different two discriminators. In addition, we train two generators in order to generate general and hard synthetic images. By ensemble learning of all the generators and discriminators, we improve the discriminators by using the generated synthetic face images, and improve the generators by passing the combined feedback of the discriminators. On the FaceForensics benchmark challenge, we thoroughly evaluate our methods by comparing the recent methods. We also provide the ablation study to prove the effectiveness and usefulness of our method.

INDEX TERMS Digital image forensics, generative adversarial ensemble learning, deep learning, synthetic image detection, face image.

I. INTRODUCTION

The face plays an important role in confirming a person's identity and understanding between human interactions [1]. Therefore, face images are considered as important cues in computer vision and machine learning areas, and many progress has been made in face detection, face recognition, and facial emotion recognition over the last decades. In addition, due to the advances of deep learning and adversarial learning [2], the recent face image generation methods can generate synthetic face images close to real ones without using image manipulation and editing by users. However, the progress of the face image generation and manipulation methods also incurs many social issues. Therefore, the image forensic problem to detect fake images has been more highlighted, and many fake image detection methods have been developed. Since some traces remain after manipulation or editing, some detection methods [3]–[6] identify fake images by detecting the artifacts. Reference [3] presents a manipulated image localization method using a probability distribution for DCT coefficients of single and double JPEG compressing regions. Reference [5] exploits inconsistencies of illumination in a manipulated photo. However, these methods are not suitable for various problems because the many prior knowledges of the image manipulation methods are needed in advance.

Recently, fake image detection methods [7]–[9] based on deep learning have been flourished. By using the learned features with convolutional neural networks (CNNs), [7] detects manipulate images. In [8], two light CNNs are used to detect image manipulation with low computational cost. Reference [9] detects manipulated faces by evaluating inconsistent 3D orientations of synthesized faces.

However, it is still challenging to identify fake images generated from the elaborate fake image generation and manipulation methods (*e.g.* Deepfake [10], Face2Face [11],

The associate editor coordinating the review of this manuscript and approving it for publication was Inês Domingues.

Faceswap [10]). To resolve this challenge, we propose a novel generative adversarial ensemble learning method. Based on adversarial learning, we train multiple discriminators and generators together. However, a goal of conventional adversarial learning methods is to improve the generation ability. On the other hand, our method is more focused on improving the discrimination ability of discriminators. For achieving this, we ensemble outputs of two discriminators by stacking fully connected layers on the discriminators. By combining different discriminators (*i.e.* ResNet and DenseNet), we also increase the diversity of discriminators and the generalization performance.

In addition, we use two generators with the same architecture. However, we train the generators in different manners to improve the diversity of fake images. The first one is pre-trained on the CelebA-HQ dataset, but the pre-trained parameters of the network are fixed during advesarial ensemble learning. The main reason is that we make this generator provide easy but various fake images to the discriminators. However, other one is trained in an adversarial manner with ensembled discriminators. As a result, this one can be trained to produce hard fake samples with the combined feedbacks of the discriminators. By feeding the generated synthetic samples to the discriminators, we further improve discrimination ability of the ensembled models.

On the FaceForensics challenge benchmark dataset [10], we train and evaluate our generative adversarial ensemble method. By comparing other fake image detection methods [12], we throughly evaluate our method. In addition, we provide some ablation study to prove benefits of our method.

To sum up, the main contribution of this paper can be summarized as follows:

- Proposition of the adversarial ensemble method for improving discrimination ability;
- Proposition of designing ensemble framework and loss function for adversarial ensemble learning;
- Proposition of training strategies to make generators generate various and hard negative fake images.

II. RELATED WORKS

Generating synthetic or manipulated images has been studied for decades ago. To achieve that, texture synthesis [13], image inpainting [14], and image stylization [15] methods have been developed, but the generated images by them with low level or hand-crafted features can be detected easily because of the low quality.

Recently, a remarkable progress of synthetic image generation has been achieved by generative adversarial networks (GANs) [2]. In GANs, a generator learns a distribution of train samples from the feedback of a discriminator over generated images. Some improved versions of GANs have been also presented for stabilizing GAN training and preventing mode collapses. For instance, WGAN [16] and SNGAN [17] redefine a GAN loss function by using the Wasserstein distance and the spectral normalization, respectively.

For generating more realistic fake images for human faces, Deepfake [10], Face2Face [11] and Faceswap [10] have been developed. In Deepfake, two Autoencoders are trained independently on source and target face datasets to reconstruct each face set. However, an encoder is shared between those Autoencoders, and all latent face vectors are produced by the same one to learn general facial representations. By passing a latent vector for a source image to a decoder for a target image, the facial expressions or orientations of the target image can be matched with the those of the source image while maintaining its face structure.

Face2Face [11] can also transfer facial image expression of a source image to a target image using the facial reenactment algorithm. However, a dense reconstruction and facial expression tracking are performed to a stream of source and target videos for generating more realistic images. In addition, Faceswap [7] fits some face region of a source image to a face of a target image. Therefore, it first detects facial landmarks from a source face, and generate a 3D model to those landmarks. Then, the 3D model is fitted to the located landmarks of a target face, and textures and colors of the 3D model are rendered with initial textures and color correction.

To identify fake images generated from these methods, fake image detection methods have been also developed. The most of traditional methods such as color filter array (CFA) analysis [18], double JPEG localization [3], [4], illumination model [5] and steganalysis feature classification [6] detect fake images by finding some traces or artifacts of those.

In CFA analysis, the difference between CFA patterns of real and fake images is used for detection. Therefore, it can be easily failed to classify fake images with similar CFA patterns in the real images or ones with distorted CFA patterns by noises and resizing. Double JPEG localization methods detect manipulated images using aligned double JPEG compression and non-double JPEG compression. Therefore, these methods investigate how well the quantization factors are aligned with the original image after JPEG compressions. In [3], a probability model is designed for computing DCT coefficients of single and double JPEG compression regions. For extruding artifacts of JPEG compression, 2-D array features [4] are extracted by computing differences between the magnitude of JPEG coefficient 2-D array of a JPEG image and its shifted version along various directions. In addition, illumination inconsistency [5] between authentic and forged regions is investigated to detect manipulated regions. The steganalysis feature [6] is also exploited to detect fake images. However, these methods using low level features can be easily failed to detect fake images when fake images have similar features with real ones, and extracted features are distorted by a noise.

Due to the recent advance of deep learning, several methods using deep learning have been proposed for solving the image forensic problem. Compared to the traditional methods mentioned above, they have shown the more better accuracy in many applications. In [19], steganalysis features are extracted by the learned convolution filters. To detect

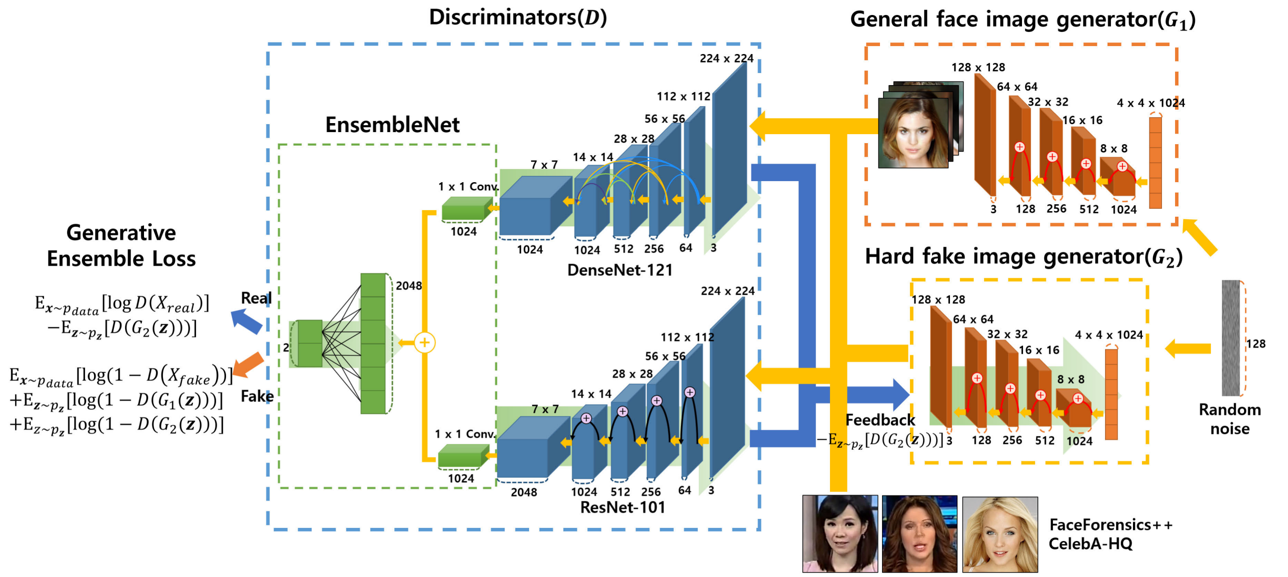
Discriminators(D)
General face image generator(G_1)
EnsembleNet
Generative Ensemble Loss
$\mathrm{E}_{x\sim p_{data}}[\log D(X_{real})]$
$-\mathrm{E}_{z\sim p_z}[D(G_2(z)))]$
Real
$\mathrm{E}_{x\sim p_{data}}[\log(1-D(X_{fake}))]$
$+\mathrm{E}_{z\sim p_z}[\log(1-D(G_1(z)))]$
$+\mathrm{E}_{z\sim p_z}[\log(1-D(G_2(z)))]$
Fake
2
2048
1 x 1 Conv.
1024
7 x 7
14 x 14
28 x 28
56 x 56
112 x 112
224 x 224
1024
1024
512
256
64
3
DenseNet-121
1 x 1 Conv.
1024
2048
1024
512
256
64
3
ResNet-101
128 x 128
64 x 64
32 x 32
16 x 16
8 x 8
4 x 4 x 1024
3
128
256
512
1024
Hard fake image generator(G_2)
Feedback
$-\mathrm{E}_{z\sim p_z}[D(G_2(z)))]$
128
Random noise
FaceForensics++
CelebA-HQ

FIGURE 1. Proposed generative adversarial ensemble learning framework consisting of 2 discriminators (ResNet and DenseNet) and 2 generators (ResNets). To combine high-level features of both discriminators and output confidences for real and fake images, we add fully connected layers on the top of both discriminators. In order to learn the ensembled discriminator, we use the FaceForensics++ and CelebA-HQ datasets as real samples, and face images generated by two generators as fake samples. Here, for providing general and diverse fake images to discriminators, the one of the generators uses the pre-trained network parameters from CelebA-HQ [24], and the parameters are fixed during the adversarial learning. On the other hand, the other one is trained from scratch with the feedback of the ensembled discriminators during the learning. Therefore, it can generate more realistic fake images to fool the discriminators during the learning.

splicing images [20], a multi-task fully connected network is learned for the surface label and edges or boundaries of splicing regions. In [21], a CNN-LSTM model is used to learn discriminative features between authentic and manipulated regions. A two-stream network [22] consisting of face classification and patch triplet streams is learned with a triplet loss for fake image detection. For identifying manipulated face images, a network [23] is learned using AdaBoost and XGBoost to handle the imbalanced dataset. In [9], inconsistency between warped face region and its nearby regions is leveraged to detect manipulated faces by the Deepfake. MesoNet [8], which combines variants of the CNN and the InceptionNet, is developed to detect manipulated faces in a video.

In this work, we also present a deep learning-based network for resolving the fake image detection problem. However, we leverage ensemble learning and generative adversarial learning. As a result, we can improve the detection ability of our network using ensemble learning.

III. GENERATIVE ADVERSARIAL ENSEMBLE LEARNING

Figure 1 shows the proposed ensemble learning framework. This framework includes two discriminators and two generators. We use the DenseNet [25] and ResNet [26] as discriminators because they show promising results for many applications. The output feature maps of these discriminators are combined by the added fully connected layers. More details of the discriminators are discussed in Sec.III-A. On the other hand, we use SNGAN [17] as generators to generate fake face images. The one of the generators uses the pre-trained network, and its parameters are not updated. However, the other generator is trained from scratch with feedbacks of an ensembled discriminator. The more details of the generators is given in Sec.III-B. In Sec. III-C, we provide our learning method to train all the discriminators and generators in the adversarial manner.

A. DISCRIMINATORS

In ensemble learning, improving diversity between networks is crucial in order to improve accuracy and generality of an ensemble model. To achieve this, we use the DenseNet [25] and ResNet [26] which have different network architectures. We then connect concatenated output feature maps with both networks to a fully connected layer. In the next section, we briefly discuss the ResNet and DenseNet to be ensembled.

1) RESNET

This network [26] is based on deep residual learning, and applied for various applications due to its high accuracy. The residual function of the ResNet can be represented as $f(\mathbf{x}) + \mathbf{x}$ with a shortcut connection which bypasses more than one layers, where $f(\mathbf{x})$ and $\mathbf{x}$ mean residual and identity mapping, respectively. The shortcut connections can mitigate the vanishing gradients when training a very deep network.

2) DENSENET

In this network, a dense block is developed to connect feature maps of all other preceding layers to all subsequent layers. More specifically, feature maps extracted from the last layers of preceding blocks are aggregated via a channel-wise concatenation, and the concatenated maps are used for an input of a subsequent layer. The benefits of DenseNet are the alleviation of the vanishing gradients, strengthened feature propagation, feature reuse, and reductions of network parameters.

3) NETWORK ENSEMBLE

For improving discriminability of real and fake images, we combine the feature maps of a DenseNet and a ResNet. To this end, we feed a same image to both networks, and extract 1024-dimensional output features by using a global average pooling and a convolution layer with 1 × 1 × 1024 filters. Then, a 2048-dimensional feature vector is generated by concatenating the output features of both networks. This concatenated feature is connected with two fully connected layers of the size 2048 × 2. We use the softmax function to normalize the 2-dimensional output scores.

B. GENERATORS

1) GENERATOR SCHEME

For providing various fake images to discriminators, we use two generators with a same architecture, but train them in a different manner. We use a variant [17], [27] of the ResNet, and a hinge loss as in [17]. Figure 2 shows the network scheme of each generator. By feeding a latent vector of dimension 128 to a FC layer, we generate a feature map of size 4 × 4 × 1024. Repeatedly, by each ResBlock the resolution and the number of channels of the feature map are increased and deceased by 2 times, respectively. From the last convolutional layer with a kernel of size 3 × 3 × 64 × 3 and hyperbolic tangent function (Tanh), we generate an image with 3 channels. In addition, we pre-train the one of generators on the CelebA-HQ dataset by using spectral normalization [17], whereas we train the other generator from scratch with the discriminators based on adversarial learning.

2) SPECTRAL NORMALIZATION FOR GENERATIVE ADVERSARIAL NETWORKS (SNGANS)

For stabilizing a discriminator during GAN training, a spectral normalization method [17] is presented. Different from other regularization methods [27], [28] which tune the Lipschitz constant by adding an input based regularization term, the spectral normalization can control Lipschitz constant without tuning. To this end, this method normalizes the spectral norm of the a weight matrix W so that $\sigma(W) = 1$ as $\overline{W}_{SN}(W) := W/\sigma(W)$. Here, the spectral norm $\sigma(W)$ is equivalent to the largest singular value of W. To compute $\sigma(W)$, the power iteration method is used. As a result, they can reduce the computational cost of estimating $\sigma(W)$.

C. GENERATIVE ADVERSARIAL ENSEMBLE LEARNING

Even though our generative adversarial ensemble learning is based on adversarial learning, the main goal of our learning is to increase the discriminability. This is different from it of the conventional GAN methods which is to improve the generation ability. For achieving our goal, we train two generators in different manners. The first one that is pre-trained on the CelebA-HQ dataset can generate diverse face images as done in the conventional GANs. However, the other generator that is adversarially trained with discriminators learns features to remove artifacts rather than generating face features.

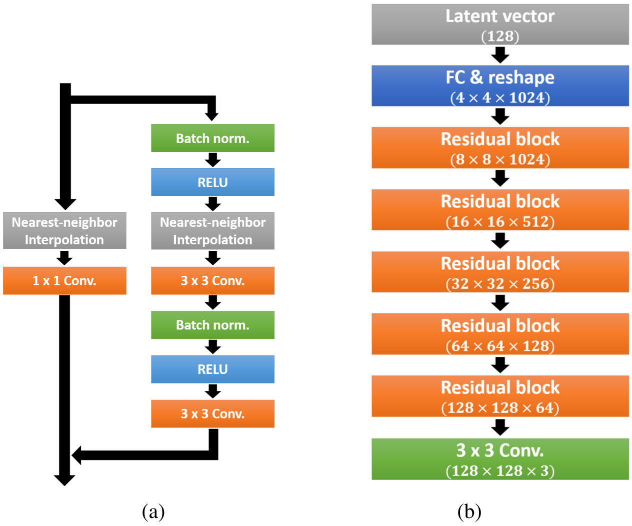
Batch norm.
RELU
Nearest-neighbor Interpolation
1 x 1 Conv.
Nearest-neighbor Interpolation
3 x 3 Conv.
Batch norm.
RELU
3 x 3 Conv.
Latent vector (128)
FC & reshape (4 × 4 × 1024)
Residual block (8 × 8 × 1024)
Residual block (16 × 16 × 512)
Residual block (32 × 32 × 256)
Residual block (64 × 64 × 128)
Residual block (128 × 128 × 64)
3 x 3 Conv. (128 × 128 × 3)
(a)
(b)

FIGURE 2. (a) The scheme of a residual block with up-sampling and (b) the overall structure of our generators G_1 and G_2.

Due to the advances of synthetic image generation methods, the difference of real and synthetic face images is subtle for a shape, color, texture of a whole face and its components (*e.g.* eyes, noise, etc.). In fact, in a real and fake face image identification, a main cue used for discriminating them is artificial traces around synthetic regions as discussed in Sec. I. Figure 3 also demonstrates the artificial traces when using the recent face image generation methods. Therefore, during the adversarial learning, it is essential that a discriminator should be trained to detect those traces, whereas a generator should be trained to generate images excluding artifacts. In this sense, we train both generators in the different manners, and they can improve discriminators further by improving the diversity of fake images.

For improving the discriminability, we combine the output feature maps of different discriminators, D_1 and D_2, and yield a confidence score of predicting real and fake image classes by using the aggregated feature. We denote the combined discriminators as D. In addition, we provide a variety of synthetic face images to D by using both generators, G_1 and G_2. A generator G_1 is pre-trained on the CelebA-HQ dataset [24] which contains different face images. Therefore, G_1 is trained for producing more general synthetic images similar to real face images. Note that G_1 is not fine-tuned further during the adversarial learning step because the generality of G_1 is reduced and the detection accuracy is degraded as shown in Table 1. On the other hand, G_2 is trained adversarially from scratch. The image quality of G_1 and G_2 is evaluated by D. In Eq.(1), we propose our adversarial ensemble loss in order

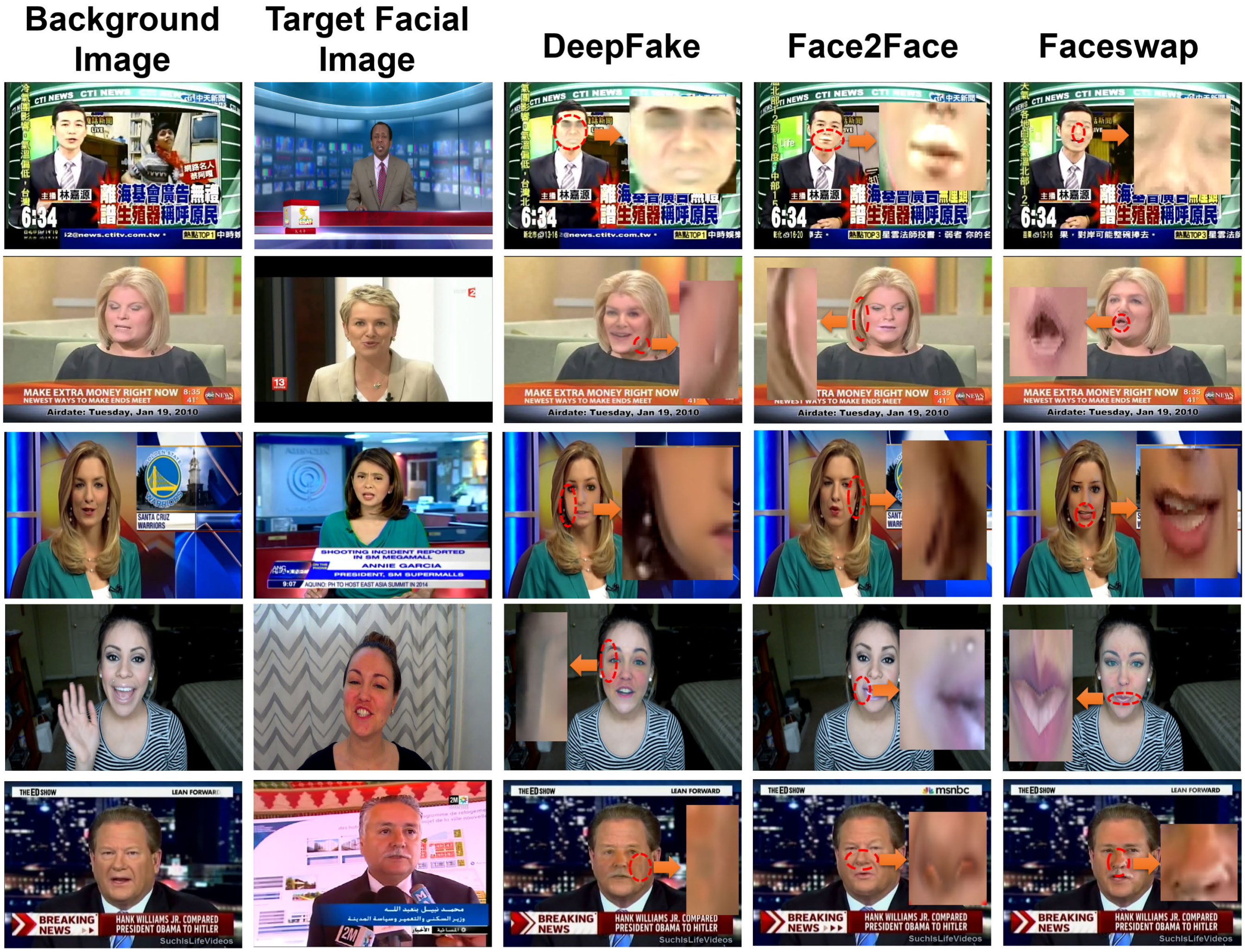

FIGURE 3. For background and target facial images, generated face images by using the deepfake, face2face, and faceswap methods. We mark some artificial traces with red circles, and highlight them by scaling up the corresponding regions. Because the global face and face components' appearances of real and fake images are similar, these artifacts should be detected for discriminating them.

to train G_2 and D

$$\begin{aligned}\min_{G_2}\max_{D} V(D, G_1, G_2) = {} & \mathbb{E}_{\mathbf{x}\sim p_{data}}[\log D(\mathbf{x}_{real}))] \\ & + \mathbb{E}_{\mathbf{x}\sim p_{data}}[\log (1 - D(\mathbf{x}_{fake}))] \\ & + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}}[\log (1 - D(G_1(\mathbf{z})))] \\ & + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}}[\log (1 - D(G_2(\mathbf{z})))] \\ & - \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}}[D(G_2(\mathbf{z}))] \qquad (1)\end{aligned}$$

To maximize V, the ensemble discriminator D should produce a higher score for a real image, but a lower score for fake image. In addition, G_2 should make D produce a higher score for fake images to minimize V.

IV. EXPERIMENTS

In this section, we prove the effects and benefits of our method via ablation study and comparisons with other face image detection methods.

A. DATASET

To show the discrimination ability of our method between real and fake images, we use FaceForensics++ [10] and CelebA-HQ [24] for training and evaluating our detector shown in Fig. 1.

The FaceForensics++ dataset includes manipulated and authentic face images. For manipulating the authentic images, Deepfake, Face2Face and Faceswap methods are exploited. For 1k pristine videos, 509,914 pristine images are captured. Then, these images are manipulated by applying those methods, and in total 1.5 million images are contained in the dataset. Each image is labeled with DeepFake, Face2Face, Faceswap, Pristine (*i.e.* real image) according to their origin and the applied manipulation method.

In addition, the image quality in the dataset can be categorized into raw, HQ (high quality) and LQ (low quality). In the raw images any compression is not applied, whereas H.264 coding with 23 and 40 constant quantization rates is used for generating HQ and LQ compression images, respectively. For the ablation study discussed in Sec. IV-C, we only use the raw images as a train set. However, we use all the raw, HQ, and LQ images for training when comparing our

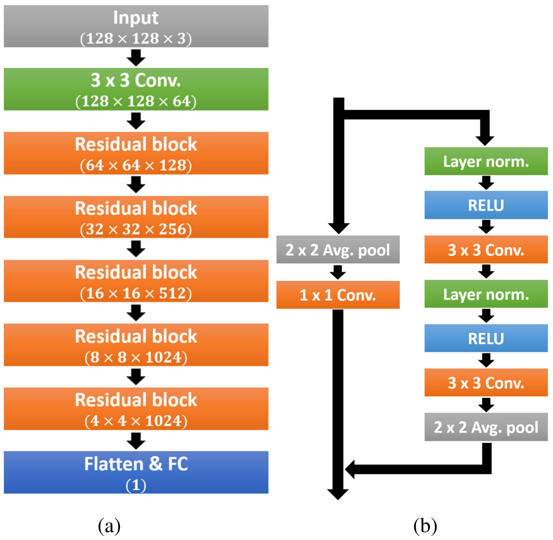

FIGURE 4. (a) The overall structure of a discriminator D_{sn} for pre-training our G_1, and (b) A residual block used for our discriminator.

method with other methods on the benchmark challenge as in Sec. IV-D.

We also use a CelebA-HQ [24] dataset for generating high quality synthesis face image as done in PGGAN [24]. We use the 30k images of 1024 × 1024 resolutions for pre-training.

B. IMPLEMENTATION DETAILS

In our experiments, we use three discriminators and two generators. ResNet-101 [26][1], DenseNet-121 [25] [2] and VGG-19 [34] [3] are used as our discriminators in our detector.

The network structure of two generators is based on ResNet of SNGAN [17] for generating 128 × 128 resolution images as shown in Fig. 2. The training procedures of both generators are described in Algorithm 1.

In the step 1, G_1 pre-trained on CelebA-HQ dataset can generate diverse synthetic face images similar to the conventional GAN. Therefore, we first train G_1 during T_{step1}[4] on the CelebA-HQ dataset. For this training, as a discriminator (D_{sn}), we use the same network presented by SNGAN [17] using the spectral normalization as described in Sec. III-B.2. D_{sn} consists of a 3 × 3 convolutional layer, 5 residual blocks and a FC layer. Figure 4 shows the overall architecture of D_{sn}. Then, D_{sn} is adversrially learned to distinguish between target dataset and generated samples from G_1 by maximizing the hinge loss in Eq.(2), and G_1 is trained over this discriminator

Algorithm 1 The Proposed Generative Adversarial Ensemble Learning Algorithm

```
Input : Mini batch of noise samples $\{\mathbf{z}^i\}_{i=1}^m$ and
        data samples
        $\{\mathbf{X}_F, \{\mathbf{X}_C, \mathbf{X}_{G_1}\}, \{\mathbf{X}_C, \mathbf{X}_{G_2}\}\} = \{\mathbf{X}_j\}_{j=1}^{T_D}$,
        $\mathbf{X}_j = \{\mathbf{x}^i\}_{i=1}^m$
//Step1: pre-training $G_1$ on CelebA-HQ
for $k_{step1} = 1,..., T_{step1}$ iterations do
    //update $D_{sn}$
    for $k_{D_{sn}} = 1,..., T_{D_{sn}}$ iterations do
        $g_{D_{sn}} \leftarrow \nabla\theta_{D_{sn}} \frac{1}{m} \sum_{i=1}^m H_{D_{sn}}(D_{sn}, G_1)$ using Eq. (2)
        $\theta_{D_{sn}} \leftarrow Adam(g_{D_{sn}}, \theta_{D_{sn}}, \eta_1, \beta_{step1}, \beta_2)$
    end
    //update $G_1$
    for $k_{G_1} = 1,..., T_{G_1}$ iterations do
        $g_{G_1} \leftarrow \nabla\theta_{G_1} \frac{1}{m} \sum_{i=1}^m H_{G_1}(D_{sn}, G_1)$ using Eq. (3)
        $\theta_{G_1} \leftarrow Adam(g_{G_1}, \theta_{G_1}, \eta_1, \beta_{step1}, \beta_2)$
    end
end
//Step2: generative adversarial ensemble learning on
FaceForensics++ and CelebA-HQ
for $k_{step2} = 1,..., T_{step2}$ iterations do
    //Update $D$ on FaceForensics++ only:
    for $k_F = 1, ..., T_F$ iterations do
        $\mathbf{X} \leftarrow \mathbf{X}_F$
        $g_D \leftarrow \nabla\theta_D \frac{1}{m} \sum_{i=1}^m V(D, G_1, G_2)$ using Eq. (1)
        $\theta_D \leftarrow Adam(g_D, \theta_D, \eta_2, \beta_{step2}, \beta_2)$
    end
    //Update $D$ on FaceForesics++, CelebA-HQ, $G_1$ and $G_2$:
    for $k_D = 1, ..., T_D$ iterations do
        $\mathbf{X} \leftarrow \mathbf{X}_{k_D}$
        $g_D \leftarrow \nabla\theta_D \frac{1}{m} \sum_{i=1}^m V(D, G_1, G_2)$ using Eq. (1)
        $\theta_D \leftarrow Adam(g_D, \theta_D, \eta_2, \beta_{step2}, \beta_2)$
    end
    //Update $G_2$
    $g_{G_2} \leftarrow \nabla\theta_{G_2} \frac{1}{m} \sum_{i=1}^m V(D, G_1, G_2)$ using Eq. (1)
    $\theta_{G_2} \leftarrow Adam(g_{G_2}, \theta_{G_2}, \eta_1, \beta_{step2}, \beta_2)$
    if $k_{step2} == \frac{1}{2} T_{step2}$ then
        $\eta_2 \leftarrow 0.94 \times \eta_2$
    end
end
```

by minimizing the hinge loss in Eq. (3).

$$\max_{D_{sn}} H_{D_{sn}}(D_{sn}, G_1) = \mathbb{E}_{\mathbf{x}\sim p_{data}}[\min(0, -1 + D_{sn}((\mathbf{x}))] + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}}[\min(0, -1 - D_{sn}(G_1(\mathbf{z})))] \tag{2}$$

$$\min_{G_1} H_{G_1}(D_{sn}, G_1) = -\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}}[D_{sn}(G_1(\mathbf{z}))] \tag{3}$$

[1] ResNet-101 is available at https://github.com/tensorflow/models/tree/master/research/slim

[2] DenseNet-121 is available at https://github.com/pudae/tensorflow-densenet

[3] VGG-19 is available at https://github.com/tensorflow/models/tree/master/research/slim

[4] T_{step1} and T_{step2} are $\frac{\text{epoch number}\times\text{total image number}}{\text{batch size}}$. The details are given in Sec. IV-C and IV-D.

For training D_{sn} and G_1, we set $T_{D_{sn}} = 5$ and $T_{G_1} = 2$ as done in [17] within step 1 of Algorithm 1. Therefore, D_{sn} and G_1 are updated five and two times per each training iteration (k_{step1}), respectively. We use the Adam optimizer [35] with $\beta_{step1} = 0.5$ and $\beta_2 = 0.999$, and fix a learning rate to $\eta_1 = 10^{-4}$.

In the step 2, we adversarially train the ensemble discriminator D and G_2 from scratch during T_{step2} while freezing the learned parameters of G_1. D is trained by maximizing the loss Eq. (1) over the real and fake images from FaceForensics++ (*i.e.* real and fake), CelebA-HQ (*i.e.* real), and generated images (*i.e.* fake) by G_1 and G_2. Subsequently, we train G_2 by minimizing Eq. (1) with the feedbacks of D.

For adversarial learning, it is known that more training of a discriminator than a generator [2] gives the better results. In addition, the parameters of our ensemble discriminator D are much more than those of G_2. For this reason, we set $T_D = 3$ within step 2 of Algorithm 1.

More specifically, a mini-batch set $\mathbf{X}$ for updating D comes from three types of real and synthetic image subsets, $\mathbf{X}_F$, $\{\mathbf{X}_C, \mathbf{X}_{G_1}\}$, and $\{\mathbf{X}_C, \mathbf{X}_{G_2}\}$. Here, $\mathbf{X}_F$ and $\mathbf{X}_C$ are real images of FaceForensics++ and CelebA-HQ, but $\mathbf{X}_{G_1}$ and $\mathbf{X}_{G_2}$ are generated images by G_1 and G_2, respectively. In our experiments, the size (m) of each subset is fixed by 16. Therefore, $\mathbf{X}_F$, $\{\mathbf{X}_C, \mathbf{X}_{G_1}\}$ and $\{\mathbf{X}_C, \mathbf{X}_{G_2}\}$ contain 16 FaceForensics++ images, 8 CelebA-HQ images and 8 generated images by G_1, and 8 CelebA-HQ images and 8 generated images by G_2, respectively. Then, D can be updated three times iteratively by changing training sets.

The Adam optimizer with $\beta_{step2} = 0.9$ and $\beta_2 = 0.999$ is also used. For training D, we set the initial learning rate to $\eta_2 = 10^{-4}$, and decay the rate by a factor of 0.94 after 5 epoch. When training G_2, we fix the learning rate to 10^{-4}. We summarize this generative ensemble adversarial learning algorithm in Algorithm 1.

We use the Tensorflow [36]. All our experiments are conducted on a NVIDIA TITAN Xp GPU and an Intel Xeon E5-2640-v4 CPU.

C. ABLATION STUDY

In this evaluation, we use 398, 908 images captured only from the raw videos of the FaceForensics++ dataset for training our detector. In this case, there exist 134, 197 Pristine, 57, 063 Deepfake, 116, 132 Face2Face and 91, 516 Faceswap images. Since the number of fake images is two times more than that of pristine images, we handle this data imbalance by replicating each pristine image. For extracting a face region within an image, we use the face recognition library[5] which is based on the face landmark estimation. Because the detected face region is tight somewhat, we enlarge the face crop area by 1.4 times centered on the extracted region.

We also use the 30k images of the CelebA-HQ dataset for training G_1, D and D_{sn}. Since the center-aligned images are provided, we only resize them to 128 × 128. In addition, the whole input images used for training all the networks are normalized to become zero mean and unit variance. We also use zero-padding to fit each image size to the input size of D ($224 \times 224 \times 3$).

We set T_{step1} and T_{step2} to 18,750 and 333,191. In addition, we do not pre-train D on FaceForensics++ in step 2 of Algorithm 1 (*i.e.* $T_F = 0$). Once a detector is trained, we evaluate it on a test set. This set contains 700 images encoded by raw, HQ, and LQ. As an evaluation metric, we also use the classification accuracy as also used in other works [10]. However, we emphasize that we evaluate the classification rates of the test images via the benchmark server [6] since their GT is unavailable.

1) GENERATIVE ENSEMBLE LEARNING

To verify the effects of the generators G_1 and G_2, we implement different versions of ensemble detectors D. We use ResNet-101, DenseNet-121 and VGG-19 as our discriminators as described in Sec. IV-C. As shown in Algorithm 1, we can define the training phases as step 1 and step 2, and the description of the trained D is given as:

- (M1) D without any generators;
- (M2) D with G_1, where G_1 is trained in step 1, but G_1 is not fine-tuned further in step 2;
- (M3) D with G_1, where G_1 is trained in both steps;
- (M4) D with G_1, where G_1 is trained from scratch in step 2;
- (M5) D with G_1 and G_2, where G_1 and G_2 are trained in step 1 and step 2, respectively;
- (M6) D with G_1 and G_2, where G_1 is trained in step 1, but G_2 is trained in step 1 and step 2.

In Table 1, we compare these detectors (M1-M6) with different generative ensemble learning methods. We confirm that (M5) using our proposed learning method shows the best rate. This also proves that our method is effective to discriminate between authentic and manipulated images.

In addition, (M2) and (M6) also show the high performance for the classification of manipulated images by Deepfake and Face2Face. This indicates that exploiting G_1 for generic fake image generation is an better way to improve our D more. In addition, we find that (M3) and (M4) show the high accuracy on Pristine images. In return, they show the low accuracy on the fake image classification. Therefore, G_2 trained in step 2 contributes to improve the manipulated image detection more. It supports our argument in Sec III-C that G_1 is trained to generate a generic fake image, but G_2 is trained to generate a fake image excluding synthetic artifacts.

From these comparisons, we verify that our generative ensemble learning is indeed beneficial for the discrimination between pristine and manipulated images. In addition, G_1 and G_2 can contribute differently to fake image detection by the proposed learning method.

[5] Code is available at https://github.com/ageitgey/face_recognition

[6] FaceForensics Benchmark is available at http://kaldir.vc.in.tum.de/faceforensics_benchmark/

TABLE 1. Comparison with training manners of generators. In the model of G_1 and G_2, Pre-trained and Scratch mean that using pre-trained generator on CelebA-HQ and training the generator from scratch, respectively. In the training of G_1 and G_2, Training and Fixed represent that updating and freezing parameters of the generator during training, respectively.

Name	G_1		G_2		DeepFake	Face2Face	Faceswap	Pristine	Total
	Model	Training	Model	Training					
M1	-	-	-	-	0.555	0.431	0.485	0.877	0.681
M2	Pre-trained	Fixed	-	-	0.691	0.460	0.456	0.829	0.680
M3	Pre-trained	Training	-	-	0.618	0.307	0.311	0.903	0.654
M4	Scratch	Training	-	-	0.555	0.380	0.320	0.897	0.657
M5	Pre-trained	Fixed	Scratch	Training	0.700	0.482	0.537	0.803	**0.684**
M6	Pre-trained	Fixed	Pre-trained	Training	0.618	0.467	0.447	0.837	0.673

2) DISCRIMINATOR ENSEMBLE

To find out the best ensemble combination among DenseNet-121, ResNet-101, and VGG-19 discriminators, we train detectors (S1)-(S5) with different *D*, and evaluate them on the FaceForensics++ test images. Here, we fix generators as (M5) which achieves the best score as in Table 1. The details of the implemented (S1)-(S5) are given in Table 2.

The results are also evaluated on FaceForensics++ test images in terms of the classification accuracy. As shown, the (S4) and (S5) using multiple discriminators also achieve the better rate than (S1)-(S3) with a single discriminator. However, it turns out that (S3) with VGG-19 is too biased to the fake image classifications. This implies that a discriminator with few layers and parameters can be easily fooled by generators. Therefore, it is very crucial to balance between a discriminator and a generator to prevent this problem. Due to this reason, (S4) without VGG-19 also is superior to (S5).

D. COMPARISON ON FACEFORENSICS BENCHMARK CHALLENGE

To prove the benefit of our method, we have participated in the FaceForensics Benchmark challenge, and compared our method with other state-of-the-art fake image detectors.

For this challenge, we use the recently updated FaceForensics++ dataset [12]. This updated dataset contains new videos manipulated by the NeuralTextures [37] method, and the videos also consist of raw, HQ and LQ compressed videos as mentioned in Sec. IV-A. The NeuralTextures can change a facial expression of a source image with an expressions of a target image while keeping its identity. To this end, it first generates a rendering map (*i.e.* UV map) and a Neural Texture map with the identity of a target image and expression of a source image. Then, it feeds a target image background, the UV map, and the Neural Texture map to a rendering network for generating a photo-realistic image. However, in NeuralTexutres videos on the FaceForensics++ [12] dataset, the facial expressions around mouth regions are manipulated only although this method can change the expression of eyes. In addition, [12] detects faces of source and target images using the Face2Face face track model, and use the PatchGAN [38] as a discriminator of the rendering network of the NeuralTextures. Figure 5 shows some NeuralTextures images.

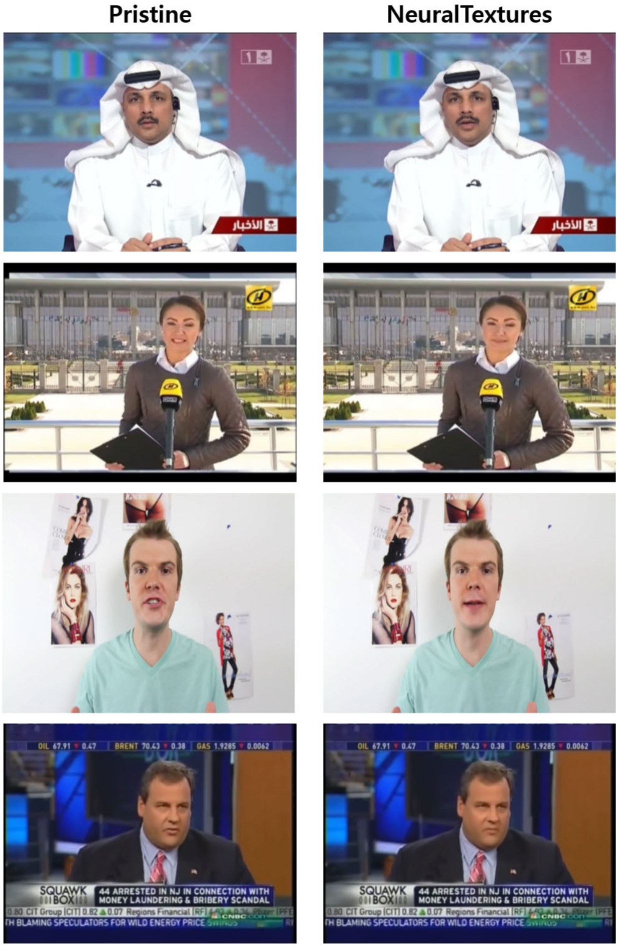

FIGURE 5. The first frame images captured from several Pristine and NeuralTextures videos are shown. Because the NeuralTextures images are generated by modifying the facial expression of the corresponding Pristine images only, it is very challenging to discriminate both images.

As NeuralTextures videos are included, an additional change of this challenge is that only videos are provided instead of images and videos. For 1k pristine videos, manipulated and HQ/LQ compressed videos are generated by Deepfake, Face2Face, Faceswap, and NeuralTextures methods.

TABLE 2. Comparison between different discriminator combinations. Here, the *G*1 and *G*2 using (M5) are used as generators.

Name	Networks			DeepFake	Face2Face	Faceswap	Prinstine	Total
	DenseNet-121	ResNet-101	VGG-19					
S1	O	-	-	0.645	0.431	0.417	0.817	0.656
S2	-	O	-	0.664	0.518	0.485	0.914	0.684
S3	-	-	O	1.000	1.000	1.000	0.000	0.500
S4	O	O	-	0.573	0.401	0.592	0.874	**0.693**
S5	O	O	O	0.700	0.482	0.537	0.803	0.684

TABLE 3. Comparison with state-of-the-art forensic detectors on the FaceForensics benchmark challenge. More details can be found in the FaceForensics benchmark website. The standard deviation (Std. Dev.) scores are also calculated with the accuracies of all the 5 sets.

Name	DeepFake	Face2Face	Faceswap	NeuralTextures	Pristine	Total	Std. Dev.
Steganalysis Features [29]	0.736	0.737	0.689	0.633	0.340	0.518	0.166
Recasting [30]	0.855	0.679	0.738	0.780	0.344	0.522	0.198
Rahmouni [31]	0.855	0.642	0.563	0.607	0.500	0.581	0.135
Bayar and Stamm [32]	0.845	0.737	0.825	0.707	0.462	0.616	0.153
XceptionNet Full Image [33]	0.746	0.759	0.709	0.733	0.510	0.624	0.103
MesoNet [8]	0.873	0.562	0.612	0.407	0.726	0.660	0.175
Xception [33]	0.964	0.869	0.903	0.807	0.524	0.710	0.171
Ours	**0.718**	**0.686**	**0.631**	**0.707**	**0.562**	**0.625**	**0.065**

As a result, a total of 15k videos are contained in the dataset. We have then captured 20 images per video, and extracted face landmarks using the same face recognition library as discussed in Sec. IV-C. We enlarge a face crop region by 1.3 times centered on the extracted region as done in [12]. To handle the imbalance between pristine and fake images, we reuse each pristine image 4 times. As a result, we use 240k Pristine, 60k Deepfake, 60k Face2Face, 60k Faceswap and 60k NeuralTextures images for training our detector.

As discriminators, we use pre-trained DenseNet-169 and ResNet-152 on ImageNet [39]. We set T_{step1} and T_{step2} to 18,750 and 300,000. For training our discriminators, we set $T_F = 8$ within step 2 of Algorithm 1. Therefore, we first train them only with the images of FaceForensics++ for 9 iterations (*i.e.* when $k_F = 1, \ldots, 8$ and $k_D = 1$) Subsequently, they are trained with generated images by G_1 and G_2. The main reason of this training strategy is to improve the discriminability more on the FaceForensics++ dataset. When training D, we set the initial learning rate to $\eta_2 = 10^{-5}$, and decay the rate by a factor 0.94 after 5 epoch. Other hyper parameters and training manners for learning D and G are same as mentioned in Sec. IV-C.

We evaluate our trained detector on the benchmark test set. This set contains 1k pristine and manipulated images encoded by raw, HQ, and LQ. As described in Sec. IV-C, we evaluate detection accuracy from the benchmark server. We determine whether each image is real or fake by comparing a confidence (*or* output) score of our detector and a threshold 0.5. Once uploading the predicted classes for all the test images to the server, it provides detection accuracy.

In Table 3, we compare our ensemble detector with other recent detectors. Even though two detectors [8], [33] show the better performance, our detector is superior to most detectors [29]–[33]. Especially, the accuracy of our method shows almost similar to it of MesoNet [8]. However, [8] achieves the performance by designing the specialized network for fake image detection, but we achieve it without modifying the base networks (*i.e.* ResNet and DenseNet). In addition, for the NeuralTextures and Face2Face sets our detector is better than [8]. As mentioned, it is more difficult to detect fake images within both sets because the facial expression is modified only.

In addition, for Pristine our method has the better accuracy than other methods [29]–[33]. Remarkably, the most of them in Table 3 show biased accuracies toward real or fake sets. To show this problem clearly, we calculate the standard deviation (Std. Dev.) of each method with the accuracy scores of all the image sets. Here, a lower variance indicates less biased to a specific set. As shown, our method achieves the lowest score. It means that our detector has high generality.

V. CONCLUSION

In this paper, we have proposed the generative adversarial ensemble learning for improving discriminability of manipulated face images. We present a novel ensemble forensic detector that consists of two different discriminators and two same generators. Based on the prediction of the combined discriminators, we can enhance forensic detection results. In addition, we improve its discriminability using the synthetic face images generated by the generators that can produce different types of images. In order to train our detector in the adversarial manner, we also present our generative adversarial ensemble learning algorithm and the adversarial ensemble loss function.

In general, the purpose of conventional adversarial learning is to improve a generator. In this learning, the discriminator cannot discriminate both distributions between real and fake images further if the generator captures the source data distribution completely.

On the other hand, our generative ensemble learning can improve the discrimination ability. To this end, we make the network structure of two generators and two discriminators asymmetry. In specific, we use the ensembled model of ResNet and DenseNet as a discriminator. Therefore, the discriminator has a deeper structure than it of the generator, and this sustains the discrimination ability of the ensemble one during adversarial learning.

We have verified the effectiveness of the proposed methods throughout extensive ablation study. We show that our proposed generator learning method can improve the detection rate by implementing and comparing different generator learning methods. Then, we also prove that our ensemble discriminator with DenseNet-121 and ResNet-101 is superior to other single discriminators and other ensemble discriminators.

We further compare our detector with state-of-the-art detectors on the FaceForensics benchmark challenge. Our detector has showed the comparable performance with the recent detectors. In particular, our detector shows the lowest bias on several types of real and fake image sets. From these results, we confirm that our generative adversarial ensemble learning is indeed beneficial to improve the accuracy of a face forensic detector.

To sum up, we have confirmed that our method shows two main advantages. The first one is that our generative adversarial ensemble learning can improve the discrimination ability indeed. The other one is that our method is to alleviate the problem, which recent works tend to be biased toward real or fake classes. Moreover, our learning method does not depend on a network architecture. Then, it can be easily applicable for the existing detectors.

In addition, we expect that our method can be used for other domain image forensic (*e.g.* passport, driver licence, ID card and SNS) that can cause social issues.

ACKNOWLEDGMENT

(Jae-Yong Baek and Yong-Sang Yoo contributed equally to this work.)

REFERENCES

[1] C. Frith, "Role of facial expressions in social interactions," *Philos. Trans. Roy. Soc. B*, vol. 364, no. 1535, pp. 3453–3458, Dec. 2009.

[2] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst. 27*, Montreal, QC, Canada, Dec. 2014, pp. 2672–2680.

[3] T. Bianchi, A. De Rosa, and A. Piva, "Improved DCT coefficient analysis for forgery localization in JPEG images," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2011, pp. 2444–2447.

[4] C. Chen, Y. Q. Shi, and W. Su, "A machine learning based scheme for double JPEG compression detection," in *Proc. 19th Int. Conf. Pattern Recognit.*, Dec. 2008, pp. 1–4.

[5] T. Carvalho, H. Farid, and E. Kee, "Exposing photo manipulation from user-guided 3D lighting analysis," in *Proc. Media Watermarking, Secur., Forensics*, San Francisco, CA, USA, Mar. 2015, p. 940902.

[6] D. Cozzolino, D. Gragnaniello, and L. Verdoliva, "Image forgery localization through the fusion of camera-based, feature-based and pixel-based techniques," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 5302–5306.

[7] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics: A large-scale video dataset for forgery detection in human faces," *CoRR*, vol. abs/1803.09179, pp. 1–21, 2018.

[8] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "MesoNet: A compact facial video forgery detection network," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Hong Kong, Dec. 2018, pp. 1–7.

[9] Y. Li and S. Lyu, "Exposing deepfake videos by detecting face warping artifacts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 46–52.

[10] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics++: Learning to detect manipulated facial images," *CoRR*, vol. abs/1901.08971v1, pp. 1–14, Aug. 2019.

[11] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Niebner, "Face2Face: Real-time face capture and reenactment of RGB videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 2387–2395.

[12] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++: Learning to detect manipulated facial images," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 1–14.

[13] A. Efros and T. Leung, "Texture synthesis by non-parametric sampling," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, vol. 2, Sep. 1999, pp. 1033–1038.

[14] N. Komodakis and G. Tziritas, "Image completion using efficient belief propagation via priority scheduling and dynamic pruning," *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2649–2661, Nov. 2007.

[15] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, "Image analogies," in *Proc. 28th Annu. Conf. Comput. Graph. Interact. Techn.*, 2001, pp. 327–340.

[16] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," *CoRR*, vol. abs/1701.07875, pp. 1–32, 2017.

[17] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, Vancouver, BC, Canada, Apr./May 2018, pp. 1–26.

[18] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva, "Image forgery localization via fine-grained analysis of CFA artifacts," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 5, pp. 1566–1577, Oct. 2012.

[19] Y. Rao and J. Ni, "A deep learning approach to detection of splicing and copy-move forgeries in images," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Abu Dhabi, United Arab Emirates, Dec. 2016, pp. 1–6.

[20] R. Salloum, Y. Ren, and C.-C. Jay Kuo, "Image splicing localization using a multi-task fully convolutional network (MFCN)," *J. Vis. Commun. Image Represent.*, vol. 51, pp. 201–209, Feb. 2018.

[21] J. H. Bappy, A. K. Roy-Chowdhury, J. Bunk, L. Nataraj, and B. Manjunath, "Exploiting spatial structure for localizing manipulated image regions," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 4980–4989.

[22] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Two-stream neural networks for tampered face detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Honolulu, HI, USA, Jul. 2017, pp. 1831–1839.

[23] L. M. Dang, S. I. Hassan, S. Im, and H. Moon, "Face image manipulation detection based on a convolutional neural network," *Expert Syst. Appl.*, vol. 129, pp. 156–168, Sep. 2019.

[24] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, Vancouver, BC, Canada, Apr./May 2018, pp. 1–26.

[25] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 2261–2269.

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

[27] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, pp. 5769–5779. [Online]. Available: http://papers.nips.cc/paper/7159-improved-training-of-wasserstein-gans

[28] G. Qi, "Loss-sensitive generative adversarial networks on lipschitz densities," *Int. J. Comput. Vis.*, pp. 1–23, Nov. 2019.

[29] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 868–882, Jun. 2012.

[30] D. Cozzolino, G. Poggi, and L. Verdoliva, "Recasting residual-based local descriptors as convolutional neural networks: An application to image forgery detection," in *Proc. 5th ACM Workshop Inf. Hiding Multimedia Secur. (IH&MMSec)*, Philadelphia, PA, USA, Jun. 2017, pp. 159–164.

[31] N. Rahmouni, V. Nozick, J. Yamagishi, and I. Echizen, "Distinguishing computer graphics from natural images using convolution neural networks," in *Proc. IEEE Workshop Inf. Forensics Secur. (WIFS)*, Rennes, France, Dec. 2017, pp. 1–6.

[32] B. Bayar and M. C. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," in *Proc. 4th ACM Workshop Inf. Hiding Multimedia Secur. (IH&MMSec)*, Vigo, Galicia, Spain, Jun. 2016, pp. 5–10.

[33] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 1800–1807.

[34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–14.

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, May 2015.

[36] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, and M. Kudlur, "Tensorflow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Oper. Syst. Design Implement. (OSDI)*, Savannah, GA, USA, Nov. 2016, pp. 265–283.

[37] J. Thies, M. Zollhöfer, and M. Nießner, "Deferred neural rendering: Image synthesis using neural textures," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–12, Jul. 2019.

[38] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 5967–5976.

[39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 248–255.

JAE-YONG BAEK received the M.S. degree in computer science and engineering from Incheon National University, South Korea, in 2019. He is currently a Research Manager with the R&D Center, Autonomous A2Z, South Korea. His research interests include generative adversarial networks, image segmentation, object detection, image classification, and deep learning.

YONG-SANG YOO is currently pursuing the B.S. degree with the Department of Computer Science and Engineering, Incheon National University, South Korea. His current research interests include generative adversarial networks, objection detection, image segmentation, machine learning, multiobject tracking, and deep learning.

SEUNG-HWAN BAE (Member, IEEE) received the B.S. degree in information and communication engineering from Chungbuk National University, in 2009, and the M.S. and Ph.D. degrees in information and communications from the Gwangju Institute of Science and Technology (GIST), in 2010 and 2015, respectively. He was a Senior Researcher with the Electronics and Telecommunications Research Institute (ETRI), South Korea, from 2015 to 2017. He was an Assistant Professor with the Department of Computer Science and Engineering, Incheon National University, Korea, from 2017 to 2020. He is currently an Assistant Professor with the Department of Computer Engineering, Inha University, South Korea. His research interests include multiobject tracking, object detection, deep learning, dimensionality reduction, medical image analysis, and generative adversarial networks.