Computer Modeling in Engineering & Sciences

Tech Science Press

Doi:10.32604/cmes.2025.064138

ARTICLE

Deepfake Detection Using Adversarial Neural Network

Priyadharsini Selvaraj[1,*], Senthil Kumar Jagatheesaperumal[2], Karthiga Marimuthu[1], Oviya Saravanan[1], Bader Fahad Alkhamees[3] and Mohammad Mehedi Hassan[3,*]

[1]Department of Computer Science and Engineering, Mepco Schlenk Engineering College, Sivakasi, 626005, India
[2]Department of Electronics and Communication Engineering, Mepco Schlenk Engineering College, Sivakasi, 626005, India
[3]Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh, 11451, Saudi Arabia

*Corresponding Authors: Priyadharsini Selvaraj. Email: priyadharsini@mepcoeng.ac.in;
Mohammad Mehedi Hassan. Email: mmhassan@ksu.edu.sa

Received: 06 February 2025; Accepted: 22 April 2025; Published: 30 May 2025

ABSTRACT: With expeditious advancements in AI-driven facial manipulation techniques, particularly deepfake technology, there is growing concern over its potential misuse. Deepfakes pose a significant threat to society, particularly by infringing on individuals' privacy. Amid significant endeavors to fabricate systems for identifying deepfake fabrications, existing methodologies often face hurdles in adjusting to innovative forgery techniques and demonstrate increased vulnerability to image and video clarity variations, thereby hindering their broad applicability to images and videos produced by unfamiliar technologies. In this manuscript, we endorse resilient training tactics to amplify generalization capabilities. In adversarial training, models are trained using deliberately crafted samples to deceive classification systems, thereby significantly enhancing their generalization ability. In response to this challenge, we propose an innovative hybrid adversarial training framework integrating Virtual Adversarial Training (VAT) with Two-Generated Blurred Adversarial Training. This combined framework bolsters the model's resilience in detecting deepfakes made using unfamiliar deep learning technologies. Through such adversarial training, models are prompted to acquire more versatile attributes. Through experimental studies, we demonstrate that our model achieves higher accuracy than existing models.

KEYWORDS: Deepfake; generalization; forgery detection; pixel-wise Gaussian blurring; virtual adversarial training

1 Introduction

The swift advancements in deep learning and synthetic modelling techniques have significantly accelerated the evolution of technologies related to face manipulation, such as deepfake technologies. AI-based face manipulation techniques, like Face2Face [1], Faceswap [2], focus on real-time facial reenactment, which replaces the facial regions with those of another person. These techniques generate images that are remarkably realistic, making it difficult to distinguish them from authentic ones.

The ascendance of new deepfake forgery techniques raises concerns about potential malicious activities. These activities include identity theft and impersonation, cyberbullying, harassment, and privacy violations. The use of deepfakes can pose significant security risks, particularly in sensitive sectors such as national security and law enforcement. As a result, authenticating audio and video may become more difficult, and there is a possibility of spreading misinformation [3]. To safeguard public safety and personal privacy, it is imperative to advance the advancement of methodologies for recognizing and uncovering deepfakes.

Early works focused on identifying anomalies in facial expressions, artifacts, and inconsistencies in multimedia content. The techniques used for detection include analyzing noise variance [4], using watermarks and digital signatures, and digital shadow writing analysis [5]. These methods, though not foolproof, aimed to spot unnatural patterns indicative of deepfake manipulation. Lately, discussions have arisen regarding deep-learning-driven deepfake detection approaches. They employ Convolutional Neural Networks (CNN) and Region Convolutional Neural Networks (RCNN) for manipulation detection. Recent deep learning-based detection models have incorporated adversarial training to enhance robustness. Methods such as FGSM-based training, PGD-based training, and standard Virtual Adversarial Training (VAT) have improved in resisting adversarial attacks. However, these models often struggle with adaptive deepfake generation techniques, where adversarial perturbations become ineffective due to evolving attack strategies. Furthermore, existing approaches fail to capture diverse adversarial examples, limiting their real-world applicability.

The challenge of identifying facial forgeries motivated us to introduce a novel adversarial training approach that involves introducing noise to images and applying blurring techniques. This aims to enhance both discriminative capabilities and generalizability. The synergistic combination significantly improves the model's ability to detect deepfakes created using unfamiliar deep learning technologies. We experimented with many forms of adversarial training, including some additive ones based on blurring. We experimented with the proposed model using adversarial perturbed and blurred instances to improve generalization performance and detect deepfakes. The key contributions of this research are:

(1) We incorporate a novel hybrid adversarial training framework, integrating VAT with Two-Generator Blurred Adversarial Training (Two-Gen-BAT). VAT enhances the model's generalization by introducing perturbations, while Two-Gen-BAT improves robustness by generating diverse adversarial examples through a two-generator adversarial approach.
(2) To enhance generalization to unobserved deepfake technologies, adversarial instances are created based on image blurring. We evaluate our approach on different datasets and show that it outperforms state-of-the-art methods.
(3) We comprehensively analyze our approach and discuss its implications for future research.

2 Related Works

2.1 Deepfake Generation

The exploration of AI-driven techniques for manipulating and forging facial images has a well-established and extensive historical background. Initial detection methods achieved pleasing results only in minimal scenarios. The swift progressions in computer graphics and computer vision have made facial alteration progressively lifelike. As an illustration, Dale et al. [6] applied graphics-based methods to regenerate 3D face models for different individuals and achieved face swaps. Reenactment entails the exchange of facial expressions among individuals. Thies et al. [1] advanced real-time facial expression transfer employing an accessible RGB-D sensor to record facial gestures. The recent upsurge of deep learning has led to the development of many vision-based methods. Generative adversarial networks (GANs) [7] were utilized for the direct generation of entire facial images from random signals. Variational Autoencoders (VAEs) can be employed to create authentic facial expressions and motions. Zao [8], a Chinese face-swapping app, uses deepfake technology to swap faces with those of celebrities, and realistic face animations are created. In March 2020, the mobile application 'Impressions' [9] was introduced as the first platform enabling users to create celebrity deepfake videos directly from their smartphones. Disney has introduced its latest technology—High-Resolution Neural Face Swapping [10], combining deepfakes and facial recognition, to recreate and bring deceased actors back to life, allowing fans to experience their performances once again.

2.2 Deepfake Detection Methods

The fabrication of misleading deepfake images and videos poses a considerable danger to personal privacy and constitutes a grave societal menace. Hence, the advancement of efficient deepfake detection solutions is paramount. Early endeavors [5,11] focused on scrutinizing internal metrics or manually designed attributes of images and videos to differentiate genuine content from falsified ones. However, contemporary approaches predominantly rely on deep learning features [12–14] to distinguish between real and manipulated media.

The existing methods, for instance, Du et al. [15] presented the Locality-Aware AutoEncoder (LAE) for detection, emphasizing enhanced generalization accuracy. However, the method yielded relatively low accuracy rates. Nguyen and Derakhshani [16] proposed exposing deepfake images by focusing on eyebrow matching. Using a cosine distance measure, their model assessed resemblances between the source and target eyebrow. However, this method heavily depends on aligning identities between the source and target, thus necessitating a substantial number of training samples. Akul Mehra et al. [17] unveiled a spatial-temporal model utilizing a CapsuleNet combined with a Long Short-Term Memory (LSTM) network for deepfake identification, and the performance spectrum might deteriorate in slight inconsistencies among frames. Shruti Agarwal et al. [18] proposed a technique for detecting altered videos by exploiting intermittent disparities between the kinetics of mouth configuration, referred to as visemes, and the associated uttered phoneme. Nonetheless, this strategy could prove time-intensive due to the need for numerous manual tasks in aligning phonemes and visemes. Khalid and Woo [19] developed OC (One Class)-FakeDect, a one-class variational autoencoder (VAE), specifically trained for deepfake detection through image reconstruction, yet this approach may not be the best anomaly scoring scheme.

Zhang et al. [20] used a Self-Supervised Decoupling Network (SSDN) to ensure resilient detection of facial forgery amidst diverse compression ratios. This model demonstrated superior performance, particularly in challenging, low-quality scenarios. However, potential challenges may arise with unseen compression rates. Luo et al. [21] developed a technique utilizing an Xception-based detector incorporating Speech-to-Residual features to enhance the universality of face forgery detection. Haliassos et al. [22] prepared grayscale lip-cropped frames as input and trained them with two pre-trained lip-reading networks: a Resnet-18 model and a Multi-Scale Temporal Convolutional Network (MS-TCN) for detecting face forgery. However, this model was unable to identify fake videos with mouth obstructions. Li et al. [23] introduced a Frequency-aware Discriminative Feature Learning (FDFL) framework for detecting forgeries. They addressed the challenge of indistinct feature differentiation in SoftMax loss and the ineffectiveness of manual features by merging SoftMax Contrastive Learning (SCL) with SoftMax loss. Deng et al. [24] suggested a detection technique that entails extracting bands along the edges of faces from video frames. They applied techniques such as convex hull, dilation, and erosion to obtain these face edge bands. In this model, it becomes difficult to predict the output if there is manipulation on regions other than the face edge.

2.3 Adversarial Training

Adversarial training employs adversarial examples to enhance the training set, serving as a primary defense against adversarial attacks [25,26]. Its roots can be traced back to when Fast Gradient Sign Method (FGSM) [27] was unveiled to enhance adversarial resilience. Madry et al. [25] subsequently suggested a more robust multi-step approach named Projected Gradient Descent (PGD), outperforming FGSM and numerous present-day defense techniques [26,28]. Hussain et al. [29] revealed the vulnerability of current deepfake detection models to adversarial instances, whereas Ruiz et al. [30] utilized adversarial instance creation to impede the utilization of photos in producing deepfakes. Wang et al. [31] formulated an adversarial network utilizing image blurring, which can be constructed by introducing two generators to train the

deepfake detection models. Safwat et al. [32] presented a hybrid deep learning model that combines the generative power of Generative Adversarial Networks (GANs) with the discriminative capabilities of Residual Neural Networks (ResNet) for detecting fake faces. The model distinguished real from synthetic faces by leveraging GANs for data augmentation and ResNet for robust feature extraction. Sadhya et al. [33] introduce an attention-based deep learning system for detecting fake faces, integrating Layer-Integrated Channel Attention (LICA) and Scaled Spatial Attention (SSA) mechanisms into the VGG (Visual Geometry Group) network architecture. This model enhances the ability to differentiate between real and manipulated faces by capturing significance across channels and spatial locations. Zhang et al. [34] introduced SRTNet, a two-stream deepfake detection network that integrates information from both the spatial and residual domains. In contrast to these approaches, we focus on incorporating VAT with blurring-based adversarial training to boost the efficiency of classification-oriented deepfake detection models.

3 Proposed Approach

This section unveils our deepfake detection framework, rooted in adversarial training. Firstly, we will discuss the motivation behind advocating adversarial training. Section 3.2 will review the adversarial training methodology used in our proposed work. Then, in Section 3.3, we offer the most specialized method utilizing Gaussian smoothing at the pixel level for executing intrusive manipulations and counter-strategy learning. Section 3.4 briefly outlines the procedure for conducting generator-based adversarial training. Then, in Section 3.5, we discuss combining multiple adversarial techniques to complement each other and provide a more robust training method. Finally, Section 3.6 covers the advantages of VAT over additive adversarial training and proposes the integration of VAT with blurred adversarial training.

3.1 Data Preprocessing

The initial stage in data preprocessing involves extracting images from videos, followed by applying random transformations. These transformations diversify the dataset, aiding models in better generalization by learning invariant features and mitigating overfitting. Subsequently, extracted images undergo a series of randomized preprocessing techniques aimed at enhancement. Image compression reduces file size by eliminating non-essential information while retaining crucial visual details.

In image data preprocessing, several methods are employed: Gaussian noise introduction introduces random pixel variations; horizontal flipping mirrors images along the horizontal axis, Principal Component Analysis (PCA) adjusts color balance, hue saturation modifies color properties, random brightness adds brightness variations, grayscale conversion transforms images to black and white, and geometric transformations such as shift, scale, or rotation are applied. Employing these techniques, as shown in Fig. 1, randomizes and enriches training data diversity, bolstering machine learning models against variations and enhancing their ability to generalize.

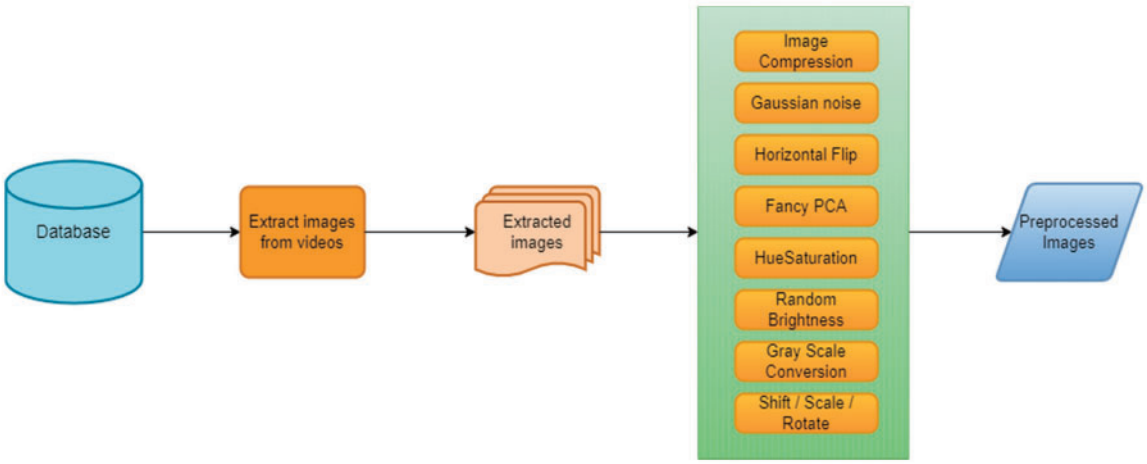

Figure 1: The set of data preprocessing techniques employed to extract images to improve the generalization of the proposed model

3.2 Additive Adversarial Training

In general, deepfake detection is conceptualized as a binary categorization task. A single image (utilized as the model's input) or a sequence of images from a singular video can generate predictions. Here, we focus on a case where the model receives only one image from a series of images in a video. The training set $\mathbb{D}$ consists of a vast collection of images accompanied by corresponding labels. A limitation of numerous existing deepfake detection models is that conventional training on a dataset $\mathbb{D}$ does not guarantee adaptability to counterfeit images generated through unfamiliar technologies or compressed at different quality levels. A potent solution to this challenge would involve integrating an "opponent" that iteratively refines the approach for training counterfeit images and rectifies evident weaknesses that the deepfake detection model could readily detect. This would empower the model to discern more intricate counterfeit images effectively, aligning with the principles of adversarial learning.

Typically, traditional deepfake classification models aim to minimize the prediction loss $L(x, y; \theta)$ for any given input data (x, y). Here, x represents the input image (real or deepfake), y indicates the label (REAL or FAKE), and θ encompasses all the learnable parameters within the classification model. The objective lies in discovering an appropriate parameter set that minimizes the empirical risk $\sum_{(x,y)\in\mathbb{D}}[L(x, y; \theta)]$ across the dataset $\mathbb{D}$, where $\mathbb{D}$ consists of pairs of input data and labels. Adversarial training, however, differs by fortifying models against adversarial vulnerabilities. It accomplishes this by producing adversarial instances and integrating them into the training dataset. This technique aims to tackle deep models' vulnerability to adversarial assaults, bolstering their resilience against such obstacles.

Various adversarial training methods have emerged in recent years, each presenting unique approaches to generating adversarial examples. One such approach, Spatial-transformed Adversarial Training (SAT), which employs adversarial optical flow to transform pixels within natural images spatially. Its goal is to induce significant prediction losses while introducing minimal perturbations. Conversely, Additive Adversarial Training (AAT), akin to FGSM, adds pixel-level perturbations to images. These methods, SAT and AAT, illustrate the diverse spectrum of techniques available for creating adversarial examples in the realm of adversarial training.

The current additive adversarial training technique involves introducing perturbations to images and subsequently training the model using these altered images. The specific method for applying these perturbations can vary between models, with many utilizing the FGSM. This framework generates each adversarial instance x^{adv} by augmenting a scaled input-gradient direction to the original image x within a specified set, as shown in Eq. (1).

$$x^{adv} = x + \epsilon \cdot \text{sign}(\nabla_x L(x, y; \theta)) \tag{1}$$

where ϵ is a constant, ∇ represents the gradient and $L(x, y; \theta)$ is the prediction loss for any given input data (x, y) and learnable parameters θ. Adversarial training operates within a zero-sum game framework, focusing on generating adversarial examples that maximize classification loss. These generated adversarial examples, alongside the original ones, contribute to a more robust training process. The optimization problem in Eq. (2) involves the classifier learning to correctly classify genuine images while encountering perturbed examples crafted by the adversary.

$$\min_{\theta} \sum_{(x,y)\epsilon\mathbb{D}} L(x, y; \theta) + \max_{\delta\epsilon\mathbb{S}} L(x + \delta, y; \theta) \tag{2}$$

the dataset $\mathbb{D}$ consists of pairs of data and labels. We establish a subset $\mathbb{S} \subseteq \mathbb{R}^{h\times w\times c}$ to limit the permissible alteration from every adversarial instance to its corresponding original image. Meanwhile, the adversary's objective is to create perturbations that deceive the detector, leading to misclassification. This methodology is referred to as Additive Adversarial Training (AAT).

3.3 Blurring Adversarial Training

While adversarial training has shown enhanced robustness against adversarial attacks in various examples, its influence on improving the capacity of deepfake detection models to generalize is still undetermined. Notably, in natural image classification tasks such as those on ImageNet [35], adversarial training has been demonstrated to have minimal impact on generalization to standard test data. This is due to the shift in distribution between adversarial instances and standard test samples. An identical obstacle might arise within the realm of identifying deepfakes. To tackle this issue, we present an innovative type of adversarial instance that demonstrates greater efficacy within the adversarial training framework, thereby improving deepfake detection.

Incorporating Gaussian blurring and JPEG compression augmentations will likely enhance deep classification Convolutional Neural Networks (CNNs). Furthermore, introducing a blurring-based adversarial training mechanism could yield even more effective results. To elaborate, consider an input image x with dimensions height (h), width (w), and channels (c). We obtain an adversarial image, x^{adv} by applying pixel-wise Gaussian blurring to x. Denoting the (i, j)th pixel of, x^{adv} as $x^{adv}_{i,j}$, our objective is to acquire a single-channel map σ^{adv} with dimensions $h \times w$. Each element, such as $\sigma^{adv}_{i,j}$, signifies the standard deviation of a gaussian filter intended for use on the region centered at the corresponding pixel in image x, i.e., $x_{i,j}$. Specifically, to determine the value of $x^{adv}_{i,j}$ (Eq.(4)), we initially gather $\sigma^{adv}_{i,j}$ and utilize it to figure the kernel $G_{i,j} \epsilon \mathbb{R}^{k\times k}$ (Eq.(3)) for applying Gaussian blurring around $x_{i,j}$. If the size of the kernel is selected as k, we calculate the dot product between $G_{i,j}$ and $\gamma\left(x_{i,j}, k\right) \epsilon \mathbb{R}^{K\times K}$, which signifies a cluster of pixels centered at the pixel $x_{i,j}$ with a radius of k.

$$G_{i,j}(u, v) = \frac{1}{2\pi\left(\sigma^{adv}_{i,j}\right)^2} exp\left(-\frac{u^2 + v^2}{2\left(\sigma^{adv}_{i,j}\right)^2}\right) \tag{3}$$

$$x_{i,j}^{adv} = \left\langle G_{i,j}, \gamma\left(x_{i,j}, k\right)\right\rangle \tag{4}$$

where the coordinates (u, v) are relative to the center pixel in $\gamma\left(x_{i,j}, k\right)$. We seek to develop a suitable mapping function, denoted as σ^{adv}, for every image used in training.

This Gaussian function plays a vital role in implementing blurring at the pixel level. The implementation of this blurring operation is optimized for efficiency through vectorization. The parameter σ^{adv} serves as a control mechanism for the extent of blurring applied to the original training image. Larger values of σ^{adv} result in more pronounced blurring, producing images with fewer discernible artifacts from the deepfake generator. Conversely, smaller values of σ^{adv} lead to less blurring, making the generated artifacts more conspicuous for the classification model to learn. The objective is to blur image regions in a pixel-wise fashion, allowing for more intense blurring in areas with fewer generalizable features.

This approach ensures that adversarial blurring is strategically applied to improve the generated content's overall quality while minimizing detectable artifacts by the classification model. The process is guided by the gradual reduction of all entries in σ^{adv} towards zero, signifying the convergence of the adversarial blurring towards the original training image. We are motivated to create a reliable mapping, σ^{adv}, for each image in our training set. Our objective with adversarial blurring entails implementing blurring at the level of individual pixels, emphasizing areas with less uniform, transferable characteristics. Similar to other techniques producing adversarial instances, our strategy strives for minimal alteration from the original images. To accomplish this, we utilize a straightforward one-step method, resembling FGSM but lacking the sign function as in Eq. (5):

$$\sigma^{adv} = \sigma + \epsilon \cdot \nabla_{\sigma} L\left(x^{adv}, y; \theta\right) \tag{5}$$

where ∇ represents the gradient, x^{adv} is derived from Eq. (4), σ represents the initialization of σ^{adv}, and ϵ is a constant. In practice, we set σ as a matrix with uniform entries. This method is known as Blurring Adversarial Training (BAT).

3.4 Generator-Based Methods

Traditionally, numerous adversarial training techniques create instances by utilizing the gradient of the loss function L for the input. Usually, resilient adversarial instances are formed via a multi-stage process, resulting in heightened computational overhead as the number of stages increases. In this strategy, an alternate technique for producing adversarial instances is employed, involving the introduction of a CNN-driven generator. Assuming an N-layer Multi-Layer Perceptron (MLP) deepfake detection model with M neurons in each layer, the computational complexity for forward and backward passes, including gradient computation is $O\left((w \times h \times c + 2) \times M + (N - 2) \times M^2\right)$. Incorporating a K-step scheme for adversarial example generation increases the computational complexity to $O\left(2 \times (w \times h \times c + 2) \times K \times M + 2 \times (N - 2) \times K \times M^2\right)$. Significantly, this intricacy is intricately linked to the structure of the deepfake detection model. However, by integrating an adversarial generator, these complexities become independent. They are solely determined by the generator's architecture as shown in Fig. 2. Table 1 shows the input and output size of the layers in the customized generator, featuring the layers such as Reflection padding layer, Convolution layer, InstanceNorm2d layer (normalization), and ReLU layer (activation), Transpose2d layer, RNet layers. Consequently, we can easily control and constrain the computational complexity by managing the size of the generator. This separation of concerns allows for more flexibility and efficiency in managing the overall computational demands of the adversarial example generation process.

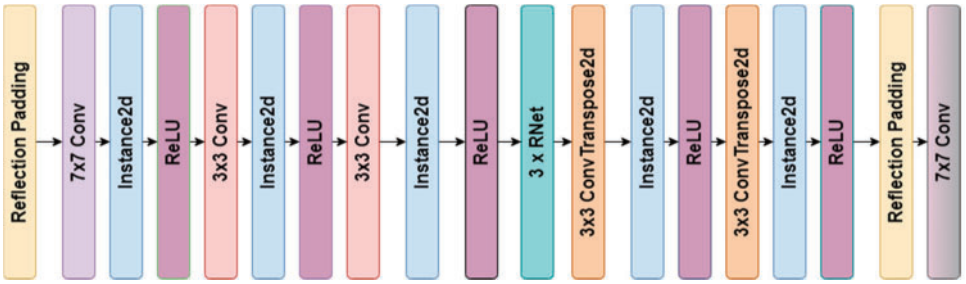

Figure 2: The layers in the architecture of Pixel Kernel Generator

Table 1: Pixel Kernel Generator layer's input and output size

Layer type	Input size	Output size
Padding	256 × 256 × 3	262 × 262 × 3
Convolutional	262 × 262 × 3	256 × 256 × 64
Normalization	256 × 256 × 64	256 × 256 × 64
Activation	256 × 256 ×64	256 × 256 × 64
Convolutional	256 × 256 × 64	128 × 128 × 128
Normalization	128 × 128 × 128	128 × 128 × 128
Activation	128 × 128 × 128	128 × 128 × 128
Convolutional	128 × 128 × 128	64 × 64 × 256
Normalization	64 × 64 × 256	64 × 64 × 256
Activation	64 × 64 × 256	64 × 64 × 256
R-net	64 × 64 × 256	64 × 64 × 256
Transposed convolutional	64 × 64 × 256	128 × 128 × 128
Normalization	128 × 128 × 128	128 × 128 × 128
Activation	128 × 128 × 128	128 × 128 × 128
Transposed convolutional	128 × 128 × 128	256 × 256 × 64
Normalization	256 × 256 × 64	256 × 256 × 64
Activation	256 × 256 × 64	256 × 256 × 64

We employ a CycleGAN [36] generator to produce the adversarial blurring map, σ^{adv}. It is crucial to note that, in contrast to the approach taken by Rusak et al. [37], we generate a specific map, $\sigma^{adv}_{(x,y)}$, for each original training sample (x, y). This individualized mapping accounts for the variation in spatial regions where the most transferable features may be present across different images. Let's say that γ_D and γ_Z stand for the collections of trainable attributes for the generator and discriminator model, respectively. This strategy is devised as a game of minimizing and maximizing, to optimize the subsequent adversarial goal given in Eq. (6).

$$\min_{\gamma_D} \max_{\gamma_Z} \sum_{(x,y)\in\mathbb{D}} L\left(x, y; \gamma_D\right) + \sum_{(x,y)\in\mathbb{D}} L\left(Z\left(x; \gamma_Z\right), y; \gamma_D\right) \tag{6}$$

the presented generator, denoted as Z, can be interpreted as an augmentation model for the initial deepfake generator(s). The optimization problem described in Eq. (6) facilitates the training of Z with the objective of enhancing deepfake images in a manner that contradicts the objectives of the deepfake detection model. In

other words, the aim is to diminish conspicuous artifacts, thereby synthesizing more realistic deepfakes that challenge the capabilities of the deepfake detection model. The successful learning of the generator results in the production of more convincing fake images, consequently enabling the classification model to acquire a deeper understanding of deepfake characteristics and enhance its overall generalizability. As a reciprocal process, the generator refines its ability to generate more realistic content. Crucially, this generator-driven method provides increased adaptability, particularly in mastering the concealment of more universally applicable attributes across various images when integrated with BAT, as depicted in our trials. In practical application, our generator functions to "improve" both counterfeit and authentic training images, facilitating a well-rounded training dataset for both categories. This balanced enhancement further reinforces the effectiveness of the training process.

The generator-based BAT shares similarities with GAN [7], consisting of a generator and a discriminator. However, a crucial distinction lies in the objective: the framework targets the improvement of the deepfake detection model (referred to as our discriminator), whereas GAN concentrates on enhancing its generator. In our method, the generator generates adversarial instances to remove anomalies in deepfake material or introduce comparable anomalies to genuine images to outwit the discriminator. The min-max competition encounters convergence issues, restricting the generator to altering obvious and less transferable anomalies that are readily identified by the deepfake detection model. To tackle the obstacle of differing distributions between authentic and counterfeit images, we propose using two generators, G_r and G_f, each dedicated to processing real and fake images, respectively. This approach allows specialization for each class, mitigating the burden on a single generator to adjust to both categories and demonstrating empirical effectiveness

3.5 Combined Adversarial Training

Integrating AAT with Two-Gen-BAT aimed to bolster the model's resilience against unseen adversarial tactics. While blurring-based adversarial examples strive for minimal deviation from original images, Two-Gen-BAT's diversity goals could inadvertently limit output variety. This constraint might hinder the model's capacity to generate truly unique or varied samples beyond its training scope. Furthermore, this approach might be more sensitive to shifts in data distribution. The model may experience performance degradation if the test data varies significantly from the training set, as it may find it challenging to adjust to these discrepancies. Therefore, integrating AAT with Two-Gen-BAT sought to fortify the model's robustness against unforeseen challenges.

To accomplish the purpose, we present a straightforward one-step technique that is similar to FGSM (apart from the sign function) as in Eq. (7).

$$x^{adv2} = x^{adv} + z \cdot \text{sign}(\nabla_o L\left(x^{adv}, y; \theta\right)) \tag{7}$$

in which x^{adv} is obtained by Eq. (4), $L(x^{adv}, y; \theta)$ denotes the prediction loss for the blurred images, z denotes the random perturbation factor, and x^{adv2} denotes the adversarial examples obtained after adding the perturbation to the blurred examples (x^{adv}).

Grad-AAT's emphasis on gradient manipulation and Two-Gen-BAT's focus on generating diverse adversarial examples could complement each other, potentially strengthening the model's overall robustness.

Integrating these methods can create a more comprehensive defense strategy, making it harder for adversaries to exploit weaknesses in a singular defense mechanism and providing broader defense coverage against various attack strategies. This adversarial training framework is known as Combined Adversarial Training Framework 1 (Combined-AT 1).

3.6 Virtual Adversarial Training

FGSM's simplicity in adding imperceptible perturbations directly to input features might not universally deliver effective results. Additionally, blurring-based adversarial examples aim to retain a closer proximity to original images. However, the amalgamation of Two-Gen-BAT and Grad-AAT could inadvertently lead to overfitting to specific adversarial examples utilized during training. While the objective is to enhance robustness, this approach risks creating a model that becomes excessively specialized, potentially compromising its ability to generalize across unseen adversarial samples or clean data. Therefore, we propose an advanced strategy by integrating Two-Gen-BAT with VAT.

A potent regularization technique for deep neural networks was presented by Miyato et al. [38] to improve generalization performance and strengthen model durability. Virtual adversarial loss, a metric that assesses the conditional label distribution's local smoothness concerning the input, is the foundation of this technique. The durability of the label distribution surrounding individual data points against local perturbations is qualified by virtual adversarial loss. This method, in contrast to adversarial training, establishes antagonistic direction without using label information. This is VAT because these reflect only virtual hostile routes.

The following steps are included in the VAT algorithm:

- Generate a random perturbation r_v with the same shape as the input data.

$$r_v \approx U(-\epsilon, \epsilon) \tag{8}$$

  where $U(-\epsilon, \epsilon)$ represents the uniform distribution between $-\epsilon$ and ϵ, ϵ determine the magnitude of the perturbation.
- Normalize r_v to have a small magnitude. To normalize it here we use tanh function which scales and shifts the values to be within the range [−1, 1]
- Create the adversarial examples x^{adv2} by slightly perturbing the blurred images x^{adv}, which were obtained from Eq. (4). The perturbed examples might be implicit in VAT, whereas it is explicit in methods like FGSM.

$$x^{adv2} = x^{adv} + xi \cdot \tanh r_v \tag{9}$$

  where xi represents a random perturbation.
- Calculate the KL (Kullback-Leibler) divergence between the model's predictions on the perturbated examples (x^{adv}) and the adversarial examples x^{adv2}:

$$KL\left(f_\theta\left(x^{adv2}\right) \parallel f_\theta\left(x^{adv}\right)\right) = \int f_\theta\left(x^{adv2}\right) \times \log\left(\frac{f_\theta\left(x^{adv2}\right)}{f_\theta\left(x^{adv}\right)}\right) dx \tag{10}$$

  where $f_\theta\left(x^{adv}\right)$ represents the Two-Gen-BAT model's output logits or probabilities, $f_\theta\left(x^{adv2}\right)$ represents the output logits for perturbated adversarial examples. This loss is known as virtual adversarial loss when the perturbation is updated iteratively.
- Minimize the KL divergence by adjusting the model parameters θ using backpropagation:

$$\min_{\theta_g} KL\left(f_{\theta_g}\left(x^{adv2}\right) \parallel f_{\theta_g}\left(x^{adv}\right)\right) + \max_{\theta_g} L\left(x^{adv}, y; \theta_g\right) \tag{11}$$

  where θ_g denotes the learnable parameters for generator model.

This process encourages the model to be robust by minimizing the divergence between its predictions on the blurred examples and the adversarial perturbed examples and by maximizing the prediction loss for blurred adversarial examples.

The synergy of VAT and Two-Generated Blurred Adversarial Training (Two-Gen-BAT) enhances the model's resilience against adversarial attacks. VAT excels at fortifying the model against local perturbations by boosting uncertainty. At the same time, blurred adversarial training introduces perturbations through image blurring or noise, fortifying the model against local and global distortions. This combination aims to diminish the model's sensitivity to subtle alterations in input data, bolstering its stability and reliability when faced with adversarial examples. By encompassing VAT and blurred adversarial training, this framework establishes a multi-layered defense mechanism, intensifying the challenge for attackers to devise compelling adversarial examples that deceive the model. This adversarial training framework is known as Combined Adversarial Training Framework 2 (Combined-AT 2), which is shown in Fig. 3. The pseudocode for Algorithm 1 used for VAT is given below:

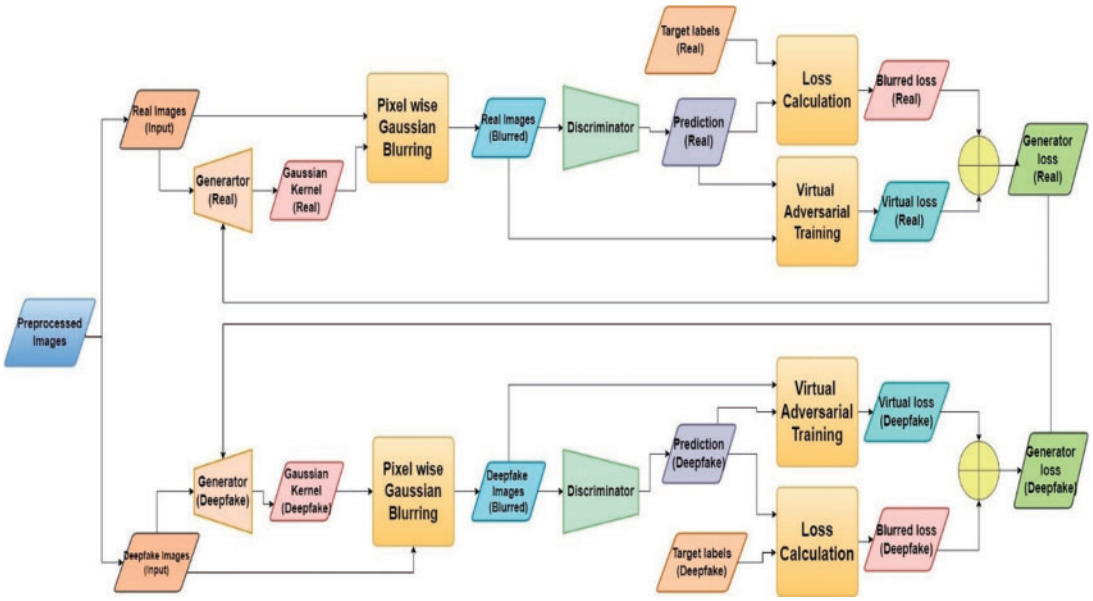

Figure 3: Combined Adversarial Training Framework 2 adopted in the proposed model

Algorithm 1: Virtual adversarial training

Input: f_θ (Discriminator model), $x^{adv1j}(j = 1, \ldots, M)$ Generated images of M samples
Output: Virtual Adversarial Loss

$r_v \approx U(-\epsilon, \epsilon)$ // generate random perturbation by calculating uniform distribution
for each $i \leftarrow 0$ to M do
$x^{adv2} = x^{adv1} + \tau \cdot \tanh r_v$ // addition of perturbation to images
$KL\left(f_\theta\left(x^{adv2}\right) \parallel f_\theta\left(x^{adv1}\right)\right) = \int f_\theta\left(x^{adv2}\right) \times \log\left(\frac{f_\theta\left(x^{adv2}\right)}{f_\theta\left(x^{adv}\right)}\right) \mathrm{d}x$
$r_v^{adv} = r_v + \epsilon \cdot \tanh\left(\nabla\left(KL\left(f_\theta\left(x^{adv2}\right) \parallel f_\theta\left(x^{adv1}\right)\right)\right)\right)$
$loss = KL(f_\theta\left(x^{adv1} + r_v^{adv}\right) \parallel f_\theta\left(x^{adv1}\right))$
return $\nabla_\theta\left(\frac{1}{M}\sum_{k=1}^{M} loss_k\right)$

Virtual adversarial training (VAT) is a machine learning technique aimed at bolstering the resilience and generalization capabilities of neural networks. The algorithm begins by initializing parameters such as perturbation size and scaling coefficient in line 1. Perturbations are iteratively generated in line 3 to amplify the discrepancy between model predictions on original and perturbed inputs, fostering uncertainty. Model predictions for both inputs are computed, and the KL divergence between predicted distributions is calculated in line 4. The perturbation is updated to maximize the KL divergence loss. Adversarial distance between original and perturbed predictions is computed in line 5 and scaled to form virtual adversarial loss in line 6. This loss enhances training objectives, improving model robustness and generalizability against adversarial attacks, thus boosting real-world effectiveness and reliability.

4 Implementation Results

4.1 Dataset

The FaceForensics++ (FF++) [39] dataset, widely utilized in research on detecting deepfakes, serves as a significant resource for this model. FF++ comprises 1000 original short video clips sourced from YouTube. Each original video in the dataset was subjected to manipulation using four sophisticated techniques: DeepFakes (DF) [40], Face2Face (F2F) [1], FaceSwap (FS) [2], and NeuralTextures (NT) [41], producing four fake videos corresponding to each original. Fig. 4 displays sample images extracted from the FF++ dataset.

Figure 4: Sample images extracted from FaceForensics++ dataset. Column 1 represents original images in the dataset. Columns 2 to 5 represent corresponding images in the dataset which have undergone manipulation techniques such as Face2Face, FaceSwap, DeepFakes, and Neural Textures

All videos in the dataset are of C23 quality, a compressed format with relatively high quality. In total, there are 5000 videos (combining real and fake). For training, we extracted 10 frames from each real and

Face2Face manipulated video, totaling 20,000 frames for training and validation. Additionally, 3 frames were extracted from each video of DF, FS, and NT, amounting to 9000 frames for testing purposes. To assess the model's ability to generalize, we trained models on videos from a particular manipulation type and then assessed their performance on videos generated by various manipulation methods.

4.2 Implementation Details

We conducted experiments to assess the effectiveness of adversarial training on established deepfake detection models. Specifically, we applied the EfficientNet [42] architecture, initially crafted for image categorization, to the task of deepfake detection. This adaptation entailed substituting the top fully connected layer with a fresh layer producing two-dimensional logits. While this newly introduced layer was initialized randomly, the remaining layers of the model were pre-trained on ImageNet [35]. For model training, we utilized the RAdam [43] optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, along with a weight decay of 2×10^{-3}. The initial learning rate was set to 5×10^{-4} and decayed by a factor of 0.1 every 5 epochs. In cases where models were trained alongside generators, the learning rate for the generators was initialized to 2×10^{-3}. When employing Gaussian blurring, we defined the blur kernel size k to 9. To maintain numerical stability, we optimized or generated $1/\sigma^{adv}$ rather than σ^{adv} for the Blurred Adversarial Training (BAT) technique in practice.

All experiments were conducted within a PyTorch [44] environment, operating on hardware equipped with an Intel CPU (Santa Clara, CA, USA) and two Nvidia Tesla T4 GPUs (Santa Clara, CA, USA). The assessment criteria included prediction accuracy, recall, precision, and F1-score.

4.3 Experimental Results

Fig. 5 shows the real image alongside the transformed image after applying random transformations and pixel-wise Gaussian blurring. The deepfake images shown in Fig. 5 exhibit lower resolution and lighting, due to the perturbations introduced by deepfake generation methods to generate adversarial images. We compared various methods for generating adversarial examples and conducting adversarial training. These methods include: (i) AAT, employing input gradient-based additive adversarial training, (ii) BAT, utilizing adversarial training based on input gradients for blurring, (iii) Two-Gen-BAT, a variation of BAT with two generators, and (iv) Combined AT1 (CT1), which combines Two-Gen-BAT with AAT. Additionally, we have (v) Combined AT2 (CT2), a combination of Two-Gen-BAT with VAT. All models underwent training exclusively on face2face data and were subsequently tested on other fake data.

Tables 2–5 present an overview of the evaluation results for adversarial training models across diverse datasets, highlighting precision, recall, accuracy, and F1-score metrics. In Table 2, focusing on the face2face dataset (validation), the Combined AT2 (CT2) model demonstrates superior performance with the highest precision (89.12%), recall (94.38%), accuracy (92.83%), and F1-Score (91.67%). BAT exhibits exceptional recall (93.54%), and CT1 shows notable precision (84.46%), while AAT and Two-Gen-BAT display balanced but comparatively lower performance.

Figure 5: Adversarial examples generated by the proposed generator model. Left top and left bottom show original and fake images, respectively, from the dataset; right top shows adversarial examples crafted on the left top image, and the right bottom shows those crafted on the left bottom image

Table 2: Evaluation metrics comparison for different deepfake detection models and our proposed model on F2F dataset

Model	Precision (%)	Recall (%)	Accuracy (%)	F1-Score (%)
PGD [25]	80.13	91.20	90.10	85.31
FGSM [27]	87.13	92.36	91.11	89.67
BAT [31]	87.15	93.54	86.25	89.59
Two-Gen-BAT [31]	87.56	84.99	86.43	86.26
Combined AT-1 [31]	83.87	84.46	75.83	72.89
GAN + ResNet [32]	88.95	93.33	90.43	91.09
Attention + VGG [33]	86.23	90.36	90.13	88.25
Combined AT-2 (ours)	89.12	94.38	92.83	91.67

Table 3: Comparison between different deepfake detection models tested on faceswap dataset

Model	Precision (%)	Recall (%)	Accuracy (%)	F1-Score (%)
PGD [25]	60.12	80.34	62.33	68.77
FGSM [27]	60.59	82.78	64.48	69.97
BAT [31]	65.67	84	70.91	70.64

(Continued)

Table 3 (continued)

Model	Precision (%)	Recall (%)	Accuracy (%)	F1-Score (%)
Two-Gen-BAT [31]	55.28	84.95	58.13	66.98
Combined AT-1 [31]	60	83.86	70.54	70.53
GAN + ResNet [32]	60.34	80.33	65.14	68.91
Attention + VGG [33]	63.24	80.32	69.11	70.76
Combined AT-2(ours)	66.8	86.53	71.8	75.42

Table 4: Comparison between different deepfake detection models tested on neural texture dataset

Model	Precision (%)	Recall (%)	Accuracy (%)	F1-Score (%)
PGD [25]	60	80	63.11	68.57
FGSM [27]	60.25	82	64.08	69.76
BAT [31]	63.53	80	67.7	69.83
Two-Gen-BAT [31]	53.65	84.96	55.76	65.77
Combined AT-1 [31]	66.62	84.17	69.79	69.84
GAN + ResNet [32]	61.22	83.15	68.53	70.52
Attention + VGG [33]	64.33	82.63	67.13	72.34
Combined AT-2(ours)	65.77	86.53	70.75	77.91

Table 5: Comparison between different deepfake detection models tested on deepfake dataset

Model	Precision (%)	Recall (%)	Accuracy (%)	F1-Score (%)
PGD [25]	61.74	82.23	64.56	70.52
FGSM [27]	62.97	82.69	67.02	71.5
BAT [31]	63.19	83	70	70.2
Two-Gen-BAT [31]	57.33	84.97	60.82	68.46
Combined AT-1 [31]	65.41	85.29	73.2	71.59
GAN + ResNet [32]	65.51	83.28	71.16	73.33

(Continued)

Table 5 (continued)

Model	Precision (%)	Recall (%)	Accuracy (%)	F1-Score (%)
Attention + VGG [33]	55.21	81.47	73.33	65.81
Combined AT-2(ours)	69.86	86.53	74.6	77.3

Table 3 illustrates the FS dataset, where CT2 emerges as the top-performing model with precision (66.8%), recall (86.53%), accuracy (71.8%), and an F1-score of 75.42%. CT1 displays competitive accuracy (70.54%). AAT and Two-Gen-BAT exhibit balanced but relatively lower performance.

Moving to Table 4, concerning the neural texture dataset, CT2 stands out as the top performer with 65.77% precision, 86.53% recall, 70.75% accuracy, and an F1-score of 77.91%. CT1 shows a notable recall (84.17%). AAT and Two-Gen-BAT maintain balanced but comparatively lower performance.

In Table 5, examining the deepfake dataset, CT2 again excels with 69.86% precision, 86.53% recall, 74.6% accuracy, and an F1-score of 70.2%. While CT1 displays competitive precision (65.41%) and recall (85.29%).

AAT and Two-Gen-BAT show balanced but comparatively lower performance across metrics. Overall, CT2 consistently performs well across various datasets, balancing precision, recall, accuracy, and F-score. Table 5 shows the consolidated accuracy comparison between different deepfake detection models on data generated using different manipulation technologies (F2F, DF, FS, NT).

The dataset has been compressed and tested using the Combined AT-2 model for the datasets, DF, NT, FS, and F2F. The model shows relatively low accuracy when tested against a compressed dataset, as shown in Fig. 6. F2F shows higher accuracy for compressed as well as uncompressed data. DF and FS show the same level of accuracy, followed by NT.

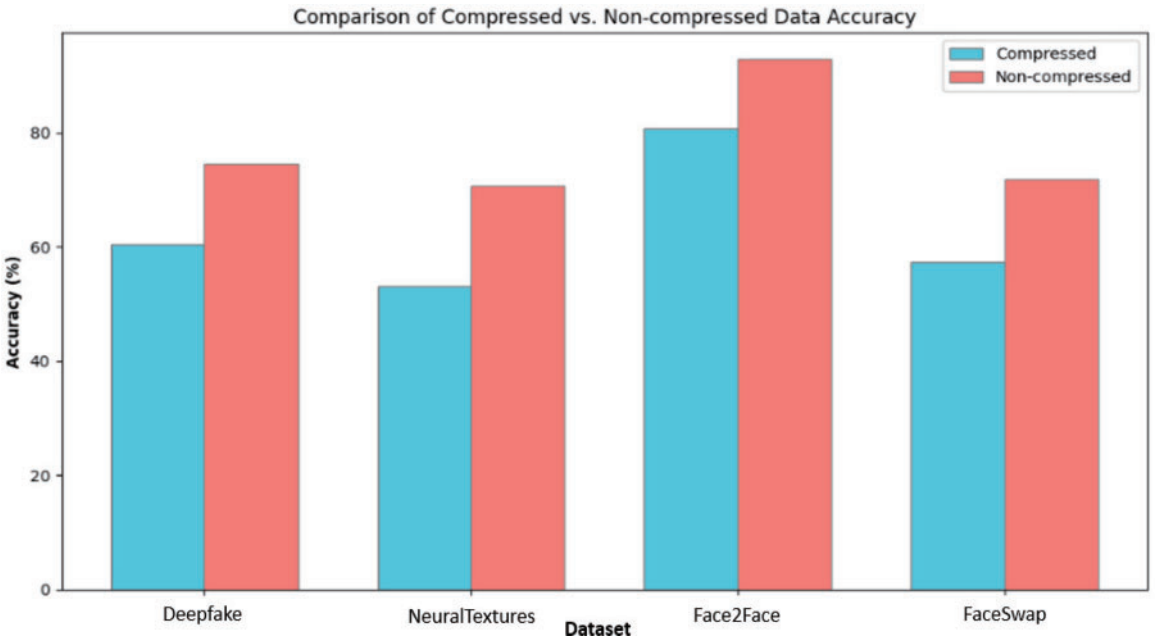

Figure 6: Comparison the performance of the proposed model using compressed and uncompressed data

To evaluate the generalizability of our model, we performed a cross-manipulation experiment, training exclusively on Face2Face (F2F) manipulated videos and testing on other manipulation types in the FaceForensics++ dataset, including DeepFakes (DF), FaceSwap (FS), and NeuralTextures (NT). This setup simulates a realistic deployment scenario where the model encounters unseen manipulation techniques, and the results in Tables 6 and 7 demonstrate that our method generalizes well across manipulation domains.

Table 6: Ablation study for the proposed model

Model	Precision (%)	Recall (%)	Accuracy (%)	F1-Score (%)
Baseline model (No Adversarial Training)	76.9	75.8	78.4	76.3
VAT only	80.5	79.3	82.1	79.9
Two-Gen-BAT only	83.2	82.1	84.7	82.6
VAT + Two-Gen-BAT (Proposed Model)	86.7	85.9	88.3	86.3

Table 7: Performance comparison of FGSM, PGD, and our proposed approach

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
FGSM [25]	80.2	78.5	77.9	78.2
PGD [27]	82.6	81.0	80.4	80.7
VAT + Two-Gen-BAT (Proposed Model)	88.3	86.7	85.9	86.3

To validate the effectiveness of our VAT with the Two-Gen-BAT framework, we conduct an ablation study analyzing how each component contributes to overall deepfake detection performance. We perform experiments on the FaceForensics++ dataset, evaluating different model variations: (1) Baseline Model—Standard CNN-based classifier without adversarial training; (2) Baseline Model + VAT Only—Model trained with VAT but without Two-Gen-BAT; (3) Baseline Model + Two-Gen-BAT Only—Model trained with Two-Gen-BAT but without VAT; (4) Baseline Model + Combined VAT + Two-Gen-BAT—Our proposed approach integrating both methods. Table 6 shows the results of the ablation study.

The baseline model (without adversarial training) struggles with unseen deepfake techniques, achieving only 78.4% accuracy. Incorporating VAT alone improves generalization, increasing accuracy to 82.1%. Two-Gen-BAT alone achieves 84.7% accuracy, proving that blurred adversarial training enhances robustness. The complete model (VAT + Two-Gen-BAT) significantly improves performance, reaching 88.3% accuracy, demonstrating that the proposed approach is the most effective.

In addition to the ablation study in Table 6, we further evaluated the effectiveness of our approach by comparing it with standard adversarial training techniques: Fast Gradient Sign Method (FGSM) [25] and Projected Gradient Descent (PGD) [27]. The results are shown in Table 7. While FGSM and PGD offer modest improvements over the baseline, our VAT + Two-Gen-BAT framework outperforms both, achieving the highest performance across all metrics. This confirms that the proposed dual-stream adversarial approach provides superior robustness against adversarial perturbations.

The model underwent comprehensive training for 100 epochs, with each session lasting approximately 2 h, conducted on an NVIDIA GPU server (Santa Clara, CA, USA), 16 GB RAM, and a 6-core Intel i7 processor (Santa Clara, CA, USA). The total training time of the proposed model is approximately 1 week.

Furthermore, data augmentation techniques, such as rotation, scaling, flipping, noise addition, blurring, and compression, were applied to improve the model's ability to recognize complex deep fakes.

5 Conclusion

A new technique has been developed to improve the versatility of deepfake detection models by incorporating VAT into the current models. Our method introduces a unique form of adversarial attacks based on image blurring, facilitated by two generators during model training and incorporating VAT. Employing VAT encourages the model to learn more transferable and essential features and helps reduce overfitting to specific training data. Unlike conventional techniques that focus on exploiting specific artifacts, our approach encourages the detection models to learn more generalizable features, thereby improving their ability to distinguish between authentic and fake content. Additionally, our adversarial method can be combined with other adversarial techniques to further enhance generalization. Through experimental trials, we showcased that our method notably enhances the generalizability of deepfake detection models across diverse, unseen image/video datasets and deepfake generation techniques. Furthermore, VAT can mitigate the impact of adversarial perturbations, making the model more resilient to various types of attacks. The train time complexity of the proposed model is high. Also, we need GPU support to train the proposed model. Further adversarial blurred examples may be crafted with different kernel sizes to improve the generalization. Refining VAT techniques is also a key area for future research. The model is still susceptible to compression artifacts and unseen deepfake techniques, requiring additional fine-tuning. By optimizing VAT algorithms, researchers can streamline training procedures, leading to the development of highly robust detection models capable of generalizing across diverse datasets and real-world scenarios.

Acknowledgement: None.

Funding Statement: This work was supported by King Saud University, Riyadh, Saudi Arabia, through the Researchers Supporting Project under Grant RSP2025R493.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Priyadharsini Selvaraj, Senthil Kumar Jagatheesaperumal; data collection: Karthiga Marimuthu; analysis and interpretation of results: Oviya Saravanan, Bader Fahad Alkhamees, Mohammad Mehedi Hassan; draft manuscript preparation: Priyadharsini Selvaraj, Senthil Kumar Jagatheesaperumal, Karthiga Marimuthu, Oviya Saravanan, Bader Fahad Alkhamees, Mohammad Mehedi Hassan. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Face Forensics++ dataset is publicly available in Kaggle at https://www.kaggle.com/datasets/xdxd003/ff-c23 (accessed on 5 February 2025). Code will be available on request from the authors.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Thies J, Zollhöfer M, Stamminger M, Theobalt C, Nießner M. Face2Face: real-time face capture and reenactment of RGB videos. In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016 Jun 27–30; Las Vegas, NV, USA. doi:10.1109/CVPR.2016.262.
2. Kowalski M. FaceSwap [Internet]. San Francisco, CA, USA: GitHub; [cited 2019 Sep 30]. Available from: https://www.github.com/MarekKowalski/FaceSwap.
3. Vaccari C, Chadwick A. Deepfakes and disinformation: exploring the impact of synthetic political video on deception, uncertainty, and trust in news. Soc Media Soc. 2020;6(1):2056305120903408. doi:10.1177/2056305120903408.

4. Pan X, Zhang X, Lyu S. Exposing image splicing with inconsistent local noise variances. In: Proceedings of the 2012 IEEE International Conference on Computational Photography (ICCP); 2012 Apr 28–29; Seattle, WA, USA. doi:10.1109/ICCPhot.2012.6215223.
5. Cozzolino D, Gragnaniello D, Verdoliva L. Image forgery localization through the fusion of camera-based, feature-based and pixel-based techniques. In: Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP); 2014 Oct 27–30; Paris, France. doi:10.1109/ICIP.2014.7026073.
6. Dale K, Sunkavalli K, Johnson MK, Vlasic D, Matusik W, Pfister H. Video face replacement. In: Proceedings of the 2011 SIGGRAPH Asia Conference; 2011 Dec 12–15; Hong Kong, China. doi:10.1145/2024156.2024164.
7. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial networks. Commun ACM. 2020;63(11):139–44. doi:10.1145/3422622.
8. ZAO (Google search) [Internet]. [cited 2025 Feb 1]. Available from: https://apps.apple.com/cn/app/zao/id1465199127.
9. Thalen M. You can now deepfake yourself into a celebrity with just a few clicks [Internet]. Austin, TX, USA: The Daily Dot. [cited 2025 Feb 1]. Available from: https://www.dailydot.com/debug/impressions-deepfake-app/.
10. Lindley JA. Disney ventures into bringing back dead actors [Internet]. New York, NY, USA: Tech Times. [cited 2025 Feb 1]. Available from: https://www.techtimes.com/articles/250776/20200702/disney-is-using-deepfakes-and-facial-recognition-to-bring-back-dead-actors.htm.
11. Goljan M, Fridrich J, Cogranne R. Rich model for steganalysis of color images. In: Proceedings of the 2014 IEEE International Workshop on Information Forensics and Security (WIFS); 2014 Dec 3–5; Atlanta, GA, USA. doi:10.1109/WIFS.2014.7084325.
12. Rahmouni N, Nozick V, Yamagishi J, Echizen I. Distinguishing computer graphics from natural images using convolution neural networks. In: Proceedings of the 2017 IEEE Workshop on Information Forensics and Security (WIFS); 2017 Dec 4–7; Rennes, France. doi:10.1109/WIFS.2017.8267647.
13. Zhou P, Han X, Morariu VI, Davis LS. Two-stream neural networks for tampered face detection. In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW); 2017 Jul 21–26; Honolulu, HI, USA. doi:10.1109/CVPRW.2017.229.
14. Li Y, Lyu S. Exposing deepfake videos by detecting face warping artifacts. arXiv:1811.00656. 2019.
15. Du M, Pentyala S, Li Y, Hu X. Towards generalizable deepfake detection with locality-aware autoencoder. arXiv:1909.05999v2. 2020.
16. Nguyen H, Derakhshani R. Eyebrow recognition for identifying deepfake videos. In: Proceedings of the 2020 International Conference of the Biometrics Special Interest Group (BIOSIG); 2020 Sep 16–18; Darmstadt, Germany.
17. Mehra A. Deepfake detection using capsule networks and long short-term memory networks [master's thesis]. Enschede, The Netherlands: University of Twente; 2020. 15 p.
18. Agarwal S, Farid H, Fried O, Agrawala M. Detecting deep-fake videos from phoneme-viseme mismatches. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW); 2020 Jun 14–19; Seattle, WA, USA. doi:10.1109/CVPRW50498.2020.00338.
19. Khalid H, Woo SS. OC-FakeDect: classifying deepfakes using one-class variational autoencoder. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW); 2020 Jun 14–19; Seattle, WA, USA. doi:10.1109/CVPRW50498.2020.00336.
20. Zhang J, Ni J, Xie H. DeepFake videos detection using self-supervised decoupling network. In: Proceedings of the 2021 IEEE International Conference on Multimedia and Expo (ICME); 2021 Jul 5–9; Shenzhen, China. doi:10.1109/ICME51207.2021.9428368.
21. Luo Y, Zhang Y, Yan J, Liu W. Generalizing face forgery detection with high-frequency features. In: Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021 Jun 20–25; Nashville, TN, USA. doi:10.1109/CVPR46437.2021.01605.
22. Haliassos A, Vougioukas K, Petridis S, Pantic M. Lips don't lie: a generalisable and robust approach to face forgery detection. In: Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021 Jun 20–25; Nashville, TN, USA. doi:10.1109/CVPR46437.2021.00500.

23. Li J, Xie H, Li J, Wang Z, Zhang Y. Frequency-aware discriminative feature learning supervised by single-center loss for face forgery detection. In: Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021 Jun 20–25; Nashville, TN, USA. doi:10.1109/CVPR46437.2021.00639.
24. Deng Z, Zhang B, He S, Wang Y. Deepfake detection method based on face edge bands. In: Proceedings of the 2022 9th International Conference on Digital Home (ICDH); 2022 Oct 28–30; Guangzhou, China. doi:10.1109/ICDH57206.2022.00046.
25. Madry A, Makelov A, Schmidt L, Tsipras T, Vladu A. Towards deep learning models resistant to adversarial attacks. arXiv:1706.06083v4. 2019.
26. Uesato J, Alayrac JB, Huang PS, Stanforth R, Fawzi A, Kohli P. Are labels required for improving adversarial robustness? arXiv:1905.13725v4. 2019.
27. Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. arXiv:1412.6572. 2014.
28. Athalye A, Carlini N, Wagner D. Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples. Proc Mach Learn Res. 2018;80:274–83.
29. Hussain S, Neekhara P, Jere M, Koushanfar F, McAuley J. Adversarial deepfakes: evaluating vulnerability of deepfake detectors to adversarial examples. In: Proceedings of the 2021 IEEE Winter Conference on Applications of Computer Vision (WACV); 2021 Jan 3–8; Waikoloa, HI, USA. doi:10.1109/WACV48630.2021.00339.
30. Ruiz N, Bargal S, Sclaroff S. Disrupting deepfakes: adversarial attacks against conditional image translation networks and facial manipulation systems. In: Proceedings of the Computer Vision—ECCV 2020 Workshops; 2020 Aug 23–28; Glasgow, UK. doi:10.1007/978-3-030-66823-5_14.
31. Wang Z, Guo Y, Zuo W. Deepfake forensics via an adversarial game. IEEE Trans Image Process. 2022;31(2):3541–52. doi:10.1109/TIP.2022.3172845.
32. Safwat S, Mahmoud A, Eldesouky Fattoh I, Ali F. Hybrid deep learning model based on GAN and RESNET for detecting fake faces. IEEE Access. 2024;2:86391–402. doi:10.1109/ACCESS.2024.3416910.
33. Sadhya S, Qi X. Complementary attention-based deep learning detection of fake faces. In: Proceedings of the 2023 IEEE International Conference on Big Data (BigData); 2023 Dec 15–18; Sorrento, Italy. doi:10.1109/BigData59044.2023.10386483.
34. Zhang D, Zhu W, Ding X, Yang G, Li F, Deng Z, et al. SRTNet: a spatial and residual based two-stream neural network for deepfakes detection. Multimed Tools Appl. 2023;82(10):14859–177. doi:10.1007/s11042-022-13966-x.
35. Deng J, Dong W, Socher R, Li LJ, Li K, Li FF. ImageNet: a large-scale hierarchical image database. In: Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition; 2009 Jun 20–25; Miami, FL, USA. doi:10.1109/CVPR.2009.5206848.
36. Zhu JY, Park T, Isola P, Efros AA. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV); 2017 Oct 22–29; Venice, Italy.
37. Rusak E, Schott L, Zimmermann RS, Bitterwolf J, Bringmann O, Bethge M, et al. A simple way to make neural networks robust against diverse image corruptions. arXiv:2001.06057. 2020.
38. Miyato T, Maeda SI, Koyama M, Ishii S. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. IEEE Trans Pattern Anal Mach Intell. 2019;41(8):1979–93. doi:10.1109/TPAMI.2018.2858821.
39. Rössler A, Cozzolino D, Verdoliva L, Riess C, Thies J, Nießner M. FaceForensics++: learning to detect manipulated facial images. arXiv:1901.08971. 2019.
40. DeepFakes (Google search) [Internet]. [cited 2025 Feb 1]. Available from: https://www.github.com/deepfakes/faceswap.
41. Thies J, Zollhöfer M, Nießner M. Deferred neural rendering: image synthesis using neural textures. arXiv:1904.12356. 2019.
42. Tan M, Le Q. EfficientNet: rethinking model scaling for convolutional neural networks. arXiv:1905.11946. 2019.
43. Liu L, Jiang H, He P, Chen W, Liu X, Gao J, et al. On the variance of the adaptive learning rate and beyond. arXiv:1908.03265. 2019.
44. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: an imperative style, high-performance deep learning library. Proc Adv Neural Inf Process Syst. 2019;32:8026–37.