OC-FakeDect: Classifying Deepfakes Using One-class Variational Autoencoder

Hasam Khalid
Computer Science and Engineering Department
Sungkyunkwan University, South Korea
hasam.khalid@g.skku.edu

Simon S. Woo
Computer Science and Engineering Department
Sungkyunkwan University, South Korea
swoo@g.skku.edu

Abstract

An image forgery method called Deepfakes can cause security and privacy issues by changing the identity of a person in a photo through the replacement of his/her face with a computer-generated image or another person's face. Therefore, a new challenge of detecting Deepfakes arises to protect individuals from potential misuses. Many researchers have proposed various binary-classification based detection approaches to detect deepfakes. However, binary-classification based methods generally require a large amount of both real and fake face images for training, and it is challenging to collect sufficient fake images data in advance. Besides, when new deepfakes generation methods are introduced, little deepfakes data will be available, and the detection performance may be mediocre. To overcome these data scarcity limitations, we formulate deepfakes detection as a one-class anomaly detection problem. We propose OC-FakeDect, which uses a one-class Variational Autoencoder (VAE) to train only on real face images and detects non-real images such as deepfakes by treating them as anomalies. Our preliminary result shows that our one class-based approach can be promising when detecting Deepfakes, achieving a 97.5% accuracy on the NeuralTextures data of the well-known FaceForensics++ benchmark dataset without using any fake images for the training process.

1. Introduction

With significant advancements made in deep-learning technologies, generating highly realistic fake human face images has become much easier than before. Recently, Deepfakes, which are a technique that replaces an individual's picture or video with another person's face using deep learning algorithms, have arisen. Deepfakes usually superimpose or combine an existing source image onto a target image using Autoencoders or Generative Adversarial Networks (GANs)[8] to create a new forged image. Autoencoders and GANs are sometimes exploited together on a single facial image to create fake images or videos. One popular example is the Deepfakes of former U.S. President, Barack Obama, generated as part of a research [29] focusing on the synthesis of a high-quality video, featuring Barack Obama speaking with accurate lip sync, composited into a target video clip. Therefore, the ability to easily forge videos raises serious security and privacy concerns: imagine hackers that can use deepfakes to present a forged video of an eminent person to send out false and potentially dangerous messages to the public. Nowadays, fake news has become an issue as well, due to the spread of misleading information via traditional news media or online social media and Deepfake videos can be combined to create arbitrary fake news to support the agenda of a malicious abuser to fool or mislead the public.

Moreover, according to an article by Dickson [6], about 96% of the Deepfakes on the Internet, such as pornography featuring faces of famous celebrities, are used without consent. For example, 41% of the Deepfakes targeted British or American on-screen characters and almost a quarter of the video involved female South Korean musicians or K-pop artists. Politicians have also been a common target of Deepfakes. Thus, Deepfakes can serve as a platform for a number of malicious use cases.

Methods like Adobe Photoshop, StyleGAN [15], Faceswap, PGGAN [14], and diverse high-fidelity images with VQ-VAE-2 [24] can be used to create fake images. Current facial manipulation methods can be broadly categorized into the following categories: 1) Facial expression manipulation [25], in which one can transfer facial expressions of a person to another using a method such as Face2Face [31], and 2) identity manipulation based on face swapping methods, in which one can replace a person's face with that of another person. Face swapping is usually performed by simple computer graphics techniques combined with deep learning methods, but it requires training with several videos. These deepfake videos are already prevalent on social media platforms, such as Snapchat and Facebook.

Recent deepfakes detection methods involve deep-learning based approaches. Hsu et al. [11] presented a two-phase deep-learning approach for the detection of several GAN-based deepfake images, such as deep convolutional GAN (DCGAN) [23] and Wasserstein GAN (WGAN) [10]. Also, forensic transfer [5] has been proposed to detect Deepfakes with only a small amount of fake images and transfer learning. Most of the proposed detection approaches formulate deep fake detection as a binary classification (real vs. fake) problem. However, detection based on the binary classification requires a large amount of data representing both classes. This poses a challenge when new Deepfakes generation methods are introduced, and only a few amount of deepfake samples are available for training. Therefore, there still exists a fundamental limitation to binary classification-based approaches. Although transfer learning [5] or few-shot-based approaches [12] have been explored, they still require the collection of new fake images for training.

To overcome these limitations regarding extreme data scarcity and dependency, we formulate Deepfakes detection as a one-class classification problem, treating real images as normal and the rest, such as Deepfakes, as anomalies. We propose OC-FakeDect , which is based on a one-class Variational Autoencoder (OC-VAE) with an additional encoder structure [18] for training only on normal data. Our preliminary result shows that our approach achieves an accuracy of 98% on Deepfakes Detection dataset, 86% on FaceSwap, and 97.5% on NeuralTextures (NT) dataset with well-known FaceForensics++ [25] benchmark dataset. Further, for the NT dataset, our model's performance is 18.5% higher than that of the state-of-the-art XceptionNet, which trains on both real and fake images, as opposed to our approach, which only trains on real images. Our method also outperforms two-class MesoNet across all datasets from FaceForensics++. This clearly demonstrates the feasibility and promise of one-class-based approaches for detecting unseen deepfakes using only real images.

The main contributions of our work are summarized as follows:

- **One-class (OC) detection**: We propose a one-class classification model OC-FakeDect, based on Variational Autoencoder (VAE) for detecting Deepfakes as an anomaly detection problem.
- **Generalizability**: Our method can be generalized to detect deepfakes as it only relies on real human face images, compared to binary-classification deepfakes detection models.
- **Detecting Deepfakes from FaceForensics++**: We use fake and real videos from the FaceForensics++ benchmark dataset to evaluate our one-class model, achieving high accuracy trained with only real images.

2. Related Work

In this section, we briefly overview multimedia forensics and research that are directly relevant to our work.

Overview of Multimedia Forensics. Digital media forensics is a field to develop forensic technologies including the validation of authenticity, originality, and provenance of an image or video. There are many approaches to detect image forgeries [7], from analyzing inconsistencies in images captured by a standard camera to extracting specific alterations made to the image. Image noise [13] has been proven to be an excellent method to detect alterations, such as copy-paste from one image to another. Other approaches examine specific artifacts arising from the synthesis process including color or texture, blinking of eyes [19], and shape cues [2]. However, these approaches are generally not suitable for images generated from scratch due to the lack or absence of synthetic artifacts.

Deep-learning based approaches. Neural networks are used for image forensics as well. For instance, CNN-based image classification models can be used to differentiate deepfake images from real ones. For training deepfakes, Faceforensics++ [25], which is comprised of different deepfake videos, serving as an automated benchmark dataset for facial manipulation detection, has been released. It is composed of more than 1.8 million images extracted from 1,000 videos from YouTube. We use this benchmark dataset for our evaluation.

Zhou et al. [38] proposed detection of face swapping manipulations of two types using a two-stream network. Raghavendra et al. [17] proposed a method to detect altered faces using two pre-trained deep Convolutional Neural Networks. They both require real and fake face image data to train their models. By increasing the layer depth of VGG16 [28] and VGG19, these models can also be used for classification, but are very costly in terms of resource consumption, and are more difficult and time-consuming to train. ShallowNet and XceptionNet [3] algorithms, which can also classify real and fake images, showed promising results.

ForensicTransfer [5] addresses this issue using a smaller amount fake images and exploring transfer learning for different domain adaptation. However, despite the transfer learning capability, it still requires both categories of images (real and fake). The common drawback of all these methods is that enough real and fake face image data are required for training. However, since Deepfakes techniques are getting more and more sophisticated and diverse, it is difficult to collect a sufficient amount of data every time a new technique is introduced. Further, we need a generalized approach to detect new Deepfakes, relying mostly on real images. To overcome these practical issues, we propose a one-class-based approach using only real images for the training process to detect non-real images.

One-class detection approaches. One-class classification models are based on the assumption that all the observations only belong to one class, "normal"; the rest of the observations are considered as "anomalies". These types of problems usually belong to the anomaly detection domain. One-class Support Vector Machine (OC-SVM) [26], which is a particular case of support vector machine that separates data points from the origin by learning a hyper-plane in a Reproducing Kernel Hilbert Space (RKHS) [36] and maximizing RKHS distance, is one of the most popular unsupervised learning methods that can detect anomalies. However, the application of non-parametric OC-SVM to the detection of deepfakes can lead to a high error rate with many support vectors.

Another example of a one-class-based approach has been explored by Oza and Patel [22], who proposed One-class Convolutional Neural Network (OC-CNN). The main idea of OC-CNN is to use a zero centered Gaussian noise in the latent space as the negative class and train the network using the cross-entropy loss. Their core objective is to make all negative distributions close to the hyper-plane. However, the objective of this model differs from the main objective of ours, because their model trains on negative or "abnormal" class with standard cross-entropy loss, while our model focuses on the "normal" class only.

Autoencoders (AEs) can also be used for one-class classification [1], or anomaly detection. An AE can be trained to reconstruct the input with a low reconstruction error rate by learning latent features of an input image. Here, "normal" data can be used for training and "abnormal" data can be detected by using reconstruction error as the anomaly score. However, AEs are deterministic and discriminative models without a probabilistic foundation. To further improve AE, Variational Autoencoder (VAE) [18], which is a stochastic generative model that can provide calibrated probabilities, has been proposed and has shown to yield better fake image detection performance. In this work, we apply OC-FakeDect to demonstrate an enhanced detection performance compared to that of AE.

3. Dataset and Pre-processing

In this section, we discuss the dataset we used for our work and describe the pre-processing procedure.

3.1. Dataset description

We used FaceForensics++ [25], which is comprised of 5 types of deepfake data as well as normal data, as our baseline dataset. More than 1,000 YouTube videos have been collected and most videos present frontal faces without occlusions, which enables automated tampering methods for the generation of realistic forgeries. The authors of FaceForensics++ generate 4 different types of deepfake image using these videos, i.e., FaceSwap (FS), Face2Face (F2F), Deepfakes (DF) and NeuralTextures (NT). We used only the real images from this dataset to train our OC-FakeDect and both the real and fake images for testing. A brief description of each dataset type is provided below.

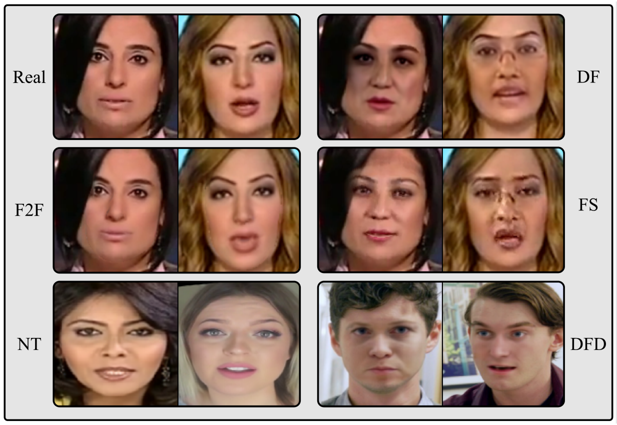


Figure 1: Examples of real and fake human face images extracted from the FaceForensics++ dataset. The first row contains real images, while other the rows below contain fake images of the DF, F2F, FS, NT and DFD datasets.

Real Images **(Real).** FaceForensics++ offers a ground truth video dataset and applies different facial manipulation techniques to generate different deepfakes. We extracted face images from these original videos using Multitask cascaded Convolutional Neural Networks (MTCNN) [37] and obtained 30,000 real human face images. These real images are used to train our OC-VAE.

FaceSwap dataset **(FS).** FaceSwap is a graphics-based approach to transfer a person's face from an image or a video to another. The face region is extracted based on sparse facial landmarks, which are used to fit a 3D template model using blend-shapes. This model is then projected back to the target image, and by using the textures of the input image, the difference between the projected shape and the localized landmarks is minimized.

Face2Face dataset **(F2F).** Face2Face [31] is a real-time facial reenactment of a target video sequence. It animates the facial expressions of a target video from a source individual and re-renders the manipulated output video in a photo-realistic fashion, while maintaining the target person's identity. It is based on two input video streams with manual key frame selection, which are then used for the dense facial reconstruction and the re-synthesis of the face with different manipulations and expressions.

DeepFakes dataset **(DF).** The DeepFakes dataset refers to the replacement of human faces using deep learning techniques. Some of the techniques involve FS [4], FakeApp and Deepfakes [9]. The goal of DF is to replace a person's face in a video with the face of a target individual. For the

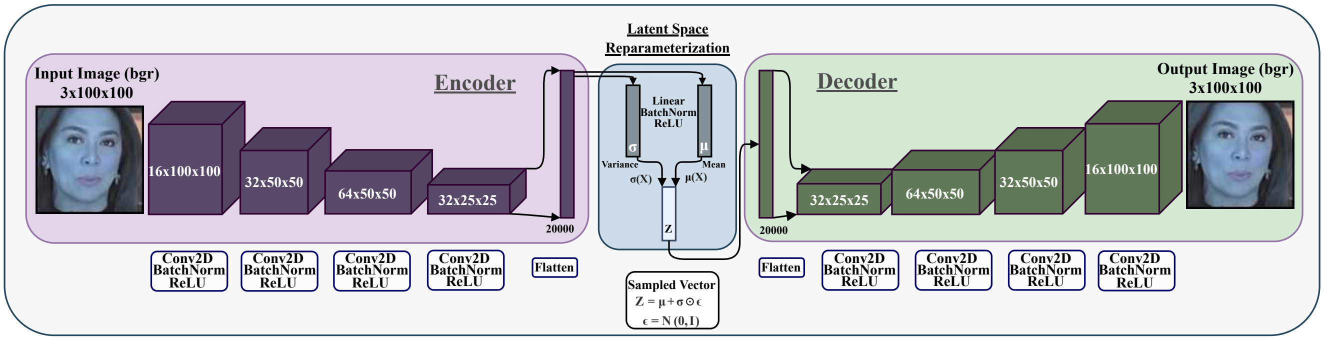


Figure 2: One-class Variational Autoencoder (OC-VAE) Architecture Diagram with latent space reparameterization

generation of forged videos, two Autoencoders [32], sharing a single encoder trained to reconstruct the original and the target person's image, are used.

NeuralTextures dataset **(NT).** NeuralTextures [30] are learned feature maps of a target individual in a video. Originally, NT was trained with a photometric reconstruction loss with an adversarial loss, but as per the implementation by Rossler et al. [25], a patch-based GAN-loss is applied. To generate neural texture information, the tracking module of Face2Face [31] is used to modify the mouth region.

Deepfake-detection dataset **(DFD).** The Deepfake-detection dataset is provided by Google and JigSaw. It contains around 3,000 manipulated videos featuring 28 actors. Some paid actors were hired to record hundreds of videos for the generation of this dataset. From these videos, they created thousands of deepfakes using publicly available deepfakes generation methods. This dataset is now available as part of the FaceForensics++ [25] benchmark dataset.

3.2. Pre-Processing

After collecting these real and fake video datasets, we extracted every frame from each video. Then, we used face detection and alignment using MTCNN [37] to extract human faces (from real and fake videos) and performed vertical alignment. We obtained 30,000 real human face images and 10,000 fake human face images for each dataset. See Fig.1 for example images.

4. Our Approach

In this section, we first introduce the basic structure of VAE, and present our OC-VAE-based approach, OC-FakeDect.

4.1. VAE and loss function

We first discuss the general OC-VAE architecture [18], which is shown in Fig. 2. OC-VAE is a Directed Probabilistic Graphical Model (DPGM) [33] consisting of an encoder and a generator (decoder). A VAE encodes the input as a distribution in the latent space, as opposed to a single point like in the case of Autoencoder (AE). In comparison with AE, the inference changes from learning $f : X \rightarrow Z$ to learning the posterior distribution $q_\phi\left(Z|X\right)$ and from learning $h : Z \rightarrow X'$ to learning the log-likelihood $p_\theta\left(X|Z\right)$, where ϕ and θ denote the parameters of the encoder and decoder, respectively. The objective function of a VAE is defined as follows:

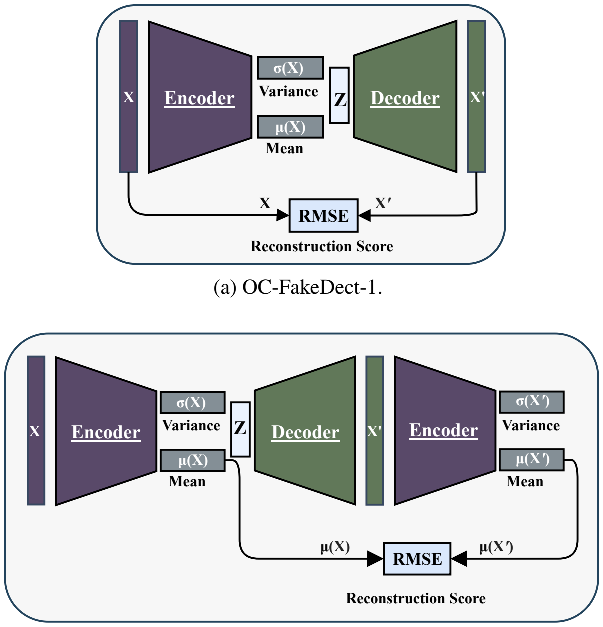
$$\begin{aligned} L(\sigma, \theta, x) =& D_{KL}\left(q_\phi\left(z|x\right) \parallel p_\theta\left(z\right)\right) \\ & - E_{q_\phi(z|x)}\left(p_\theta\left(x|z\right)\right), \end{aligned} \tag{1}$$

where the first term is the KL divergence (KLD)[34] of the approximated posterior and the prior of the latent space and the second term is calculated through the Monte Carlo method [35], which can be understood in terms of the reconstruction of the input from posterior distribution $q_\phi\left(Z|X\right)$ and the likelihood $p_\theta\left(X|Z\right)$. It can be directly expressed in terms of the mean $\mu\left(x\right)$ and covariance $\sigma\left(x\right)$ matrices of the two distributions.

Similarly, our OC-FakeDect's loss function $L_{OC-FakeDect}$ is defined in Eq. 2. We denote X as the input and $p_\theta\left(X|Z\right)$ or $p^*_\theta\left(Z\right)$ as the decoder output, respectively: D_{KL} to force the network to approximate a Gaussian distribution $N\left(\mu\left(x\right), \sigma\left(x\right)\right)$ in latent space, and mean square error to measure the difference between input and output image:

$$\begin{aligned} L_{OC-FakeDect} =& D_{KL}\left[N\left(\mu\left(x\right), \sigma\left(x\right)\right)\ , N\left(0, I\right)\right] \\ & + \parallel X - p^*_\theta\left(Z\right) \parallel^2 \end{aligned} \tag{2}$$

When sampling from the distribution returned by the encoder, the Monte Carlo gradient method used to optimize the variational lower bound suffers from very high variance. To overcome this issue by latent space reparametrization technique, which is used to render the gradient descent possible, as shown in the blue middle block of Fig. 2. Without using this technique, back-propagation is not possible due to the random sampling occurring in the latent space. If we



(a) OC-FakeDect-1.

(b) OC-FakeDect-2.

Figure 3: OC-FakeDect architecture: (a) OC-FakeDect-1 computes the reconstruction score directly from the input and output image, and (b) OC-FakeDect-2 with the additional encoder structure computes the reconstruction score from the input and output latent information.

denote Z as a random variable from a Gaussian distribution with mean $\mu(x)$ and covariance $\sigma(x)$, then Z, involving the reparameterization technique, can be defined as follows:

$$Z = \sigma(x) \times \zeta + \mu(x), \zeta \sim N(0, I). \tag{3}$$

Equation 3 ensures that the latent vector Z follows the posterior distribution, enabling us to train our model similar to training a VAE. The latent space reparameterization is also illustrated in the blue middle block in Fig. 2.

4.2. Anomaly Score

To distinguish real and fake face images, it is important to evaluate them using a same metric. Therefore, we compute the anomaly score, also referred to as the reconstruction loss or reconstruction score, for each image. To calculate the reconstruction score, we compute the Root Mean Squared Error (RMSE) between the input and output images of VAE as follows:

$$rmse = \sqrt{(\frac{1}{n}) \sum_{i=1}^{n} (X_i' - X_i)^2}, \tag{4}$$

where X is the original input and X' is the reconstructed output. By computing the reconstruction score for each image from training set using the Eq. (4), we construct a one-class (OC) (“real”) distribution. Based on this OC distribution, we can determine a static threshold to distinguish non-real images from real images. We use a statistical thresholding by calculating the inter-quartile range (IQR) to mark the 80% quartile (T_{80}) of the distribution.

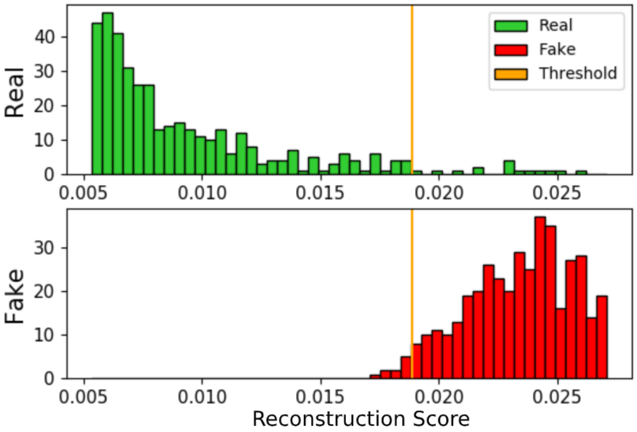


Figure 4: Histogram of reconstruction scores of real (green) and fake (red) images, and the statistical threshold (orange) on the NeuralTextures dataset with 50 real and 50 fake images.

4.3. OC-FakeDect

Based on the OC-VAE architecture as shown in Fig. 2, we propose two different OC-VAE-based approaches, OC-FakeDect-1 and OC-FakeDect-2, to detect real and fake images. Figure 3a represents our first approach, OC-FakeDect-1, and Fig. 3b represents our second approach, OC-FakeDect-2. The first approach uses the same encoder and decoder building blocks, and loss function as shown in Fig. 2. In this first approach, we compute the reconstruction score by computing the RMSE between the original input X and the reconstructed output X' to determine distinguish real and fake images, as shown using Eq. 4. In the second approach, as shown in Fig. 3b, we have an additional encoder block following the decoder, which takes the decoder output X' as an input and produces $\mu(X')$. Then we compute the RMSE between the first encoder output $\mu(X)$ and the second encoder's output $\mu(X')$ of the input image using Eq. 4. We believe that the additional encoder block can effectively extract real image features from the decoder output again, while lacking the ability to extract the features of non-real images such as deepfakes, giving high reconstruction score.

The main difference between our two proposed approaches is the method for the computation of the reconstruction score. More specifically, OC-FakeDect-1 com-

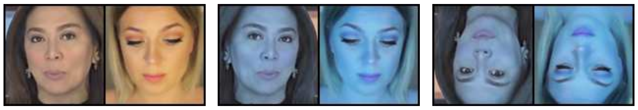

(a) Original (b) Horizontal Flip (c) Vertical Flip

Figure 5: Data augmentation examples using real face images.

putes the reconstruction score between the input and output images directly. On the other hand, OC-FakeDect-2 computes the reconstruction score between the latent representations of the input and output images, such that it better captures their latent characteristics. As an example, we present the histogram of 50 real and 50 fake images to illustrate the differences between two distributions in Fig. 4, where the X-axis is the calculated reconstruction score from OC-FakeDect-1. We can clearly observe the difference in distributions between real and fake images from based on the reconstruction score and the statistical threshold.

Data Augmentation. We applied various data augmentation techniques for the training and validation sets of real images, including horizontal and vertical flipping, and change of brightness, hue, and saturation with a factor of 0.05. Further, we normalized the distribution with a mean and standard deviation of 0.5. Examples of data augmentation techniques on the real image dataset from FaceForensics++ are shown in Figure 5.

5. Experimental Results

In this section, we present our experimental setup, as well as the detailed training procedures. Also, we report our detection performance results and analyze the activation mappings on real and fake images. We also compare the detection performance of a one-class-based approach with that of a two-class-based approach.

5.1. Experiment

We scaled each image to 100 × 100 pixels and obtained 30,000 non-augmented real images, which will serve as the training data. For the training process of our OC-FakeDect, we used the Adam gradient-based optimization method with a learning rate of 0.001 and batch-size of 128. We used convolutional layers and applied batch normalization with ReLU activation for each layer in both the encoder and the decoder. We trained our network for 300 epochs and chose the model yielding the best accuracy score on the validation set. We trained OC-FakeDect only with 30,000 real images, and without the fake images. For the testing phase, we used 500 real images and 500 fake images from the Deepfake, NeuralTexture, FaceSwap, Face2Face, and Deepfake Detection datasets, provided by the FaceForensics++ [25] benchmark dataset.

For comparison, we used the one-class Autoencoder (OC-AE) as the baseline model, since it is widely used for OC classification tasks. We built an Autoencoder with three convolutional layers in the encoder and three convolutional layers in the decoder with batch-normalization and ReLU activation on every layer. Next, our proposed approaches are compared against each other, using the same training and testing procedure for all datasets. We compute the reconstruction score for each image and classify it as fake or real based on the threshold as described earlier.

5.2. Results

We present our models' performances in Table 1 using precision, recall, and F1 score for both the real and fake datesets from 5 different sources in FaceForensics++. We compare our OC-FakeDect with the baseline OC-AE. As shown in Table 2, both OC-FakeDect approaches outperform OC-AE, which performs the worst giving 0.465 to 0.669 F1 score. On the other hand, OC-FakeDect-2 achieves the highest F1 score for all datasets, as marked in bold (from 0.712 to 0.982). In particular, OC-FakeDect-2 achieved highest accuracy on DFD, and the second highest accuracy on NT dataset. Therefore, OC-FakeDect performs significantly better than AE for the OC detection problem, demonstrating that it is a more generalized approach to detect various types of fake images.

We believe that the higher performance of our OC-FakeDect approach compared to that of OC-AE is attributed to the fact that the latter, AE-based method, is trained only to learn how to reconstruct the original input, while the former, VAE-based method, learns the parameters of a probability distribution representing the data. When comparing OC-FakeDect-1 and OC-FakeDect-2, the performance differences are relatively small. Although slight improvements are observed for OC-FakeDect-2 on the DF and NT datasets, both methods achieve above 90% accuracy for NT and DFD. The slightly higher performance of OC-FakeDect-2 compared to that of OC-FakeDect-1 is attributed to the better anomaly scoring method, as well as the additional encoder structure following the decoder of OC-VAE-2 as show in Fig. 3b. The obtained thresholds range from 0.009 to 0.022, slightly varying for the different datasets.

5.3. Analysis

Real vs. Fake in OC-FakeDect. To compare and visualize the learned features between real and fake images by OC-FakeDect, we employed GradCAM [27]. The results are illustrated in Fig. 6. Figure 6a and 6b present the example of an (a) original input image, (b) reconstructed image, (c) class activation map (CAM) of last layer of the decoder, and (d) the overlay of (a) and (c) for real and fake face im-

Table 1: Performance of OC-AE, OC-FakeDect-1, and OC-FakeDect-2 on 5 different types of real and fake benchmark dataset provided by FaceForensics++. The threshold value is obtained based on the reconstruction score for each real and fake image of the testing data. The highest values are marked in bold and thresholds are underlined.

Dataset	**Model**	**OC-AE (Baseline)**			**OC-FakeDect-1 (Ours)**			**OC-FakeDect-2 (Ours)**		
	Type	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Deepfake	**Real**	0.492	0.492	0.492	0.860	0.864	0.862	0.885	0.882	**0.883**
	Fake	0.492	0.492	0.492	0.864	0.860	0.862	0.882	0.886	**0.884**
	Threshold			0.017			0.012			0.014
NeuralTexture	**Real**	0.547	0.548	0.548	0.952	0.954	0.953	0.979	0.970	**0.974**
	Fake	0.547	0.546	0.546	0.954	0.952	0.953	0.970	0.980	**0.975**
	Threshold			0.017			0.026			0.018
FaceSwap	**Real**	0.482	0.482	0.482	0.845	0.852	0.849	0.863	0.858	**0.860**
	Fake	0.482	0.482	0.482	0.851	0.844	0.847	0.858	0.864	**0.861**
	Threshold			0.017			0.010			0.012
Face2Face	**Real**	0.477	0.477	0.465	0.707	0.706	0.707	0.712	0.712	**0.712**
	Fake	0.477	0.477	0.475	0.707	0.708	0.707	0.712	0.712	**0.712**
	Threshold			0.017			0.006			0.009
Deepfake	**Real**	0.669	0.668	0.669	0.978	0.982	0.980	0.989	0.974	**0.981**
Detection	**Fake**	0.669	0.670	0.669	0.982	0.978	0.980	0.974	0.990	**0.982**
	Threshold			0.021			0.038			0.022

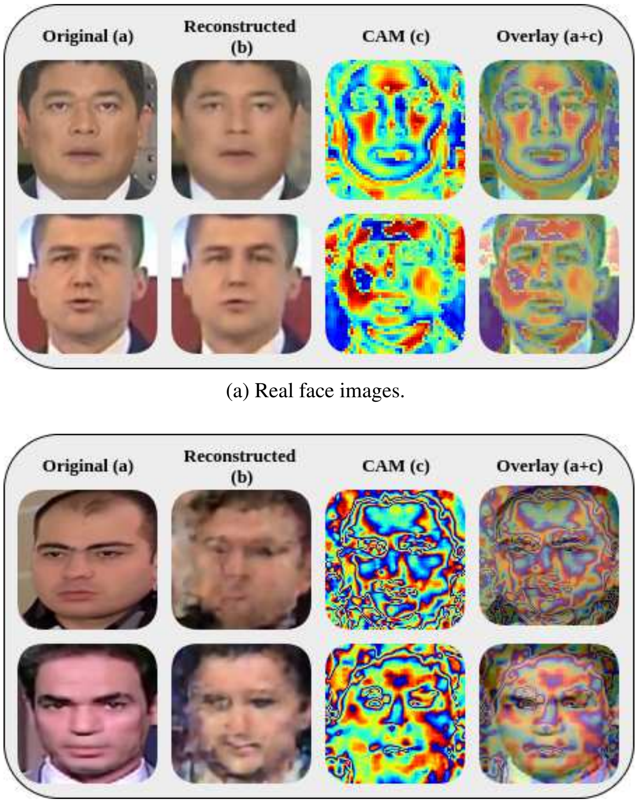
ages, respectively. As shown in Fig 6, we can clearly observe that OC-FakeDect produces intense activation around the face areas of real images. Unlike real images where facial features, such as the nose, the forehead, and the cheeks, are clearly localized via class activation mapping, fake images present rather disordered and dispersed activation patterns, complicating the perception of essential facial regions compared to that in real images. Therefore, our approach better reconstructs the real face images, yielding higher reconstruction scores for fake images. This shows that a one-class-based approach can effectively distinguish real images ("normal") and anomalies ("abnormal"), such as Deepfakes, as shown in Table 1.

In addition, in order to verify that OC-FakeDect is actually learning the difference between real and fake images, we used 9,000 real images and 9,000 fake images from the NeuralTextures dataset to measure the loss from OC-FakeDect. Figure 7 presents the loss plot for the real and fake images, as well as the training loss using only the real images over 100 epochs. The blue curve represents the training loss for real images; the green curve depicts the loss for real images and the red curve describes the loss for fake images. We can observe a clear difference between the red curve (fake) and green/blue curve (real). This means OC-FakeDect is actually learning about the real images and is able to distinguish real and fake images.

One-Class vs. Two-Class performance. The main objective of our approach is to explore the performance of OC detection. However, it would also be interesting to compare our OC performance with state-of-the-art two-class detection methods, such as MesoNet and XceptionNet [25]. Table 2 summarizes the accuracy scores of all three OC models and the two-class low quality trained models such as MesoNet and XceptionNet across the five types of datasets. The results for MesoNet and XceptionNet models are taken from FaceForensics++ [25]. Accuracy scores marked in bold represent the highest accuracy scores and those underlined represent the second highest accuracy scores achieved for each dataset. As shown in Table 2, XceptionNet, which uses both real and fake images for its training process, achieves the highest accuracy scores for DF, FS, and F2F. Surprisingly, OC-FakeDect , which is only trained on real images, outperforms XceptionNet on the NT dataset. In fact, OC-FakeDect achieved the second highest performance across all datasets, outperforming the two-class MesoNet. This demonstrates promising results, that is, OC-based detection is indeed a viable option for the development of a generalized deepfakes detector, without the need for any fake images during the training phase.

6. Limitations and Future Work

Our models outperform OC-AE, as well as MesoNet, which is a binary classification approach. Further, we achieve better performance on the NT dataset using only real images compared to XceptionNet, which is the current state-of-the-art method; however, we still need to improve our performance for other datasets. One of the limitations of our approach is that we rely on the RMSE function to compute the reconstruction score of images: it would be better



(a) Real face images.

(b) Fake face images.

Figure 6: Class Activation Map (CAM) from OC-FakeDect . The (a) original input, (b) the reconstructed output, (c) the CAM outputs, and (d) the overlaid images of the original input and its CAM for real and fake face images from NeuralTextures dataset are shown.

Table 2: Performance summary based on accuracy for all methods on the Deepfakes, NeuralTextures, FaceSwap, Face2Face and Deepfake-detection datasets. The performances of MesoNet and XceptionNet are obtained from [5].

Model	**NT**	**DF**	**FS**	**F2F**	**DFD**
OC-AE	54.60	49.20	48.20	47.80	66.90
OC-FakeDect1	95.30	86.20	84.80	70.70	98.00
OC-FakeDect2	**97.50**	88.40	86.10	71.20	**98.20**
MesoNet [25]	40.67	87.27	61.17	56.20	N/A
Xcep. Net [25]	80.67	**96.36**	**90.29**	**86.86**	N/A

if we can develop a method in which the network itself provides a reconstruction score or develops a better anomaly scoring scheme. Also, we only use the real and fake images dataset from FaceForensics++: it would be interesting to extend our approach and leverage additional real images from CelebA [20], FFHQ [16], and VoxCeleb [21], and further experiment to detect GAN-generated images.

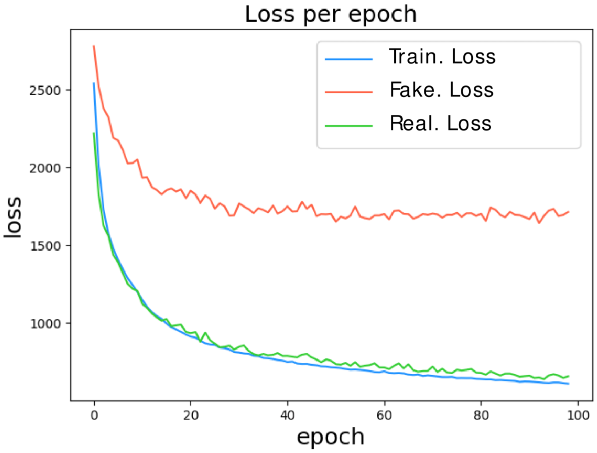


Figure 7: Loss plot from our OC-FakeDect-2 for 100 epochs, where the blue line represents the training loss for real images; the green line represents the loss for real images and red line represents loss for fake images.

7. Conclusion

In this work, we formulate the challenging task of DeepFakes detection as a one-class problem using only real images for training. We propose OC-FakeDect , a model with a novel architecture, consisting of an additional encoder block to effectively learn the features of real images and detect anomalies, such as Deepfakes. Our proposed system outperforms other one-class-based approaches, as well as the two-class MesoNet. We also achieved higher performance on the NT dataset compared to XceptionNet. Using only the real images, our approach demonstrates that one-class-based detection can be a promising option for coping with new or unseen deepfakes generation methods without the need for any of those fake samples. Finally, future work will include the improvement of one-class deepfakes detection methods to develop a more generalized and robust detection model, as well as to better explore the learned features from real images.

Acknowledgements

We thank Siho Han for providing his expertise to greatly improve this work. This research was supported by Energy Cloud R&D Program through the National Research Foundation (NRF) of Korea funded by the Ministry of Science, ICT (No. 2019M3F2A1072217), and was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2017R1C1B5076474, and 2020R1C1C1006004).

References

[1] Andrea Borghesi, Andrea Bartolini, Michele Lombardi, Michela Milano, and Luca Benini. Anomaly detection using autoencoders in high performance computing systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9428–9433, 2019.

[2] Tiago Carvalho, Fabio A Faria, Helio Pedrini, Ricardo da S Torres, and Anderson Rocha. Illuminant-based transformed spaces for image forensics. *IEEE transactions on information forensics and security*, 11(4):720–733, 2015.

[3] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[4] Jon Christian. Experts fear face swapping tech could start an international showdown. *Retrieved July*, 5:2018, 2018.

[5] Davide Cozzolino, Justus Thies, Andreas Rössler, Christian Riess, Matthias Nießner, and Luisa Verdoliva. Forensictransfer: Weakly-supervised domain adaptation for forgery detection. *arXiv preprint arXiv:1812.02510*, 2018.

[6] EJ Dickson. Deepfake porn is still a threat, particularly for k-pop stars - rolling stone, October 2019. (Accessed on 01/05/2020).

[7] Hany Farid. A survey of image forgery detection. *IEEE Signal Processing Magazine*, 26(2):16–25, 2009.

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[9] Douglas Harris. Deepfakes: False pornography is here and the law cannot protect you. *Duke L. & Tech. Rev.*, 17:99, 2018.

[10] Chin-Cheng Hsu, Hsin-Te Hwang, Yi-Chiao Wu, Yu Tsao, and Hsin-Min Wang. Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks. *arXiv preprint arXiv:1704.00849*, 2017.

[11] Chih-Chung Hsu, Yi-Xiu Zhuang, and Chia-Yen Lee. Deep fake image detection based on pairwise learning. *Applied Sciences*, 10(1):370, 2020.

[12] Hyeonseong Jeon, Youngoh Bang, and Simon S Woo. Faketalkerdetect: Effective and practical realistic neural talking head detection with a highly unbalanced dataset. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[13] Thibaut Julliand, Vincent Nozick, and Hugues Talbot. Image noise and digital image forensics. In *International Workshop on Digital Watermarking*, pages 3–17. Springer, 2015.

[14] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[15] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[16] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[17] Ali Khodabakhsh, Raghavendra Ramachandra, Kiran Raja, Pankaj Wasnik, and Christoph Busch. Fake face detection methods: Can they be generalized? In *2018 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–6. IEEE, 2018.

[18] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[19] Yuezun Li, Ming-Ching Chang, and Siwei Lyu. In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–7. IEEE, 2018.

[20] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[21] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: a large-scale speaker identification dataset. *arXiv preprint arXiv:1706.08612*, 2017.

[22] Poojan Oza and Vishal M Patel. One-class convolutional neural network. *IEEE Signal Processing Letters*, 26(2):277–281, 2018.

[23] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[24] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *Advances in Neural Information Processing Systems*, pages 14837–14847, 2019.

[25] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–11, 2019.

[26] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

[27] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[29] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics (TOG)*, 36(4):1–13, 2017.

[30] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural tex-

tures. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019.

[31] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2387–2395, 2016.

[32] Wikipedia contributors. Autoencoder — Wikipedia, the free encyclopedia, 2020. [Online; accessed 28-January-2020].

[33] Wikipedia contributors. Graphical model — Wikipedia, the free encyclopedia, 2020. [Online; accessed 15-February-2020].

[34] Wikipedia contributors. Kullback–leibler divergence — Wikipedia, the free encyclopedia, 2020. [Online; accessed 13-March-2020].

[35] Wikipedia contributors. Monte carlo method — Wikipedia, the free encyclopedia, 2020. [Online; accessed 13-March-2020].

[36] Wikipedia contributors. Reproducing kernel hilbert space — Wikipedia, the free encyclopedia, 2020. [Online; accessed 13-March-2020].

[37] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016.

[38] Peng Zhou, Xintong Han, Vlad I Morariu, and Larry S Davis. Two-stream neural networks for tampered face detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1831–1839. IEEE, 2017.