TAR: Generalized Forensic Framework to Detect Deepfakes Using Weakly Supervised Learning

Sangyup Lee, Shahroz Tariq, Junyaup Kim, Simon S. Woo

TAR: Generalized Forensic Framework to Detect Deepfakes using Weakly Supervised Learning

Sangyup Lee, Shahroz Tariq, Junyaup Kim, and Simon S. Woo

Sungkyunkwan University, Suwon, South Korea
{sangyup.lee,shahroz,yaup21c,swoo}@g.skku.edu

Abstract. Deepfakes have become a critical social problem, and detecting them is of utmost importance. Also, deepfake generation methods are advancing, and it is becoming harder to detect. While many deepfake detection models can detect different types of deepfakes separately, they perform poorly on generalizing the detection performance over multiple types of deepfake. This motivates us to develop a generalized model to detect different types of deepfakes. Therefore, in this work, we introduce a practical digital forensic tool to detect different types of deepfakes simultaneously and propose Transfer learning-based Autoencoder with Residuals (TAR). The ultimate goal of our work is to develop a unified model to detect various types of deepfake videos with high accuracy, with only a small number of training samples that can work well in real-world settings. We develop an autoencoder-based detection model with Residual blocks and sequentially perform transfer learning to detect different types of deepfakes simultaneously. Our approach achieves a much higher generalized detection performance than the state-of-the-art methods on the FaceForensics++ dataset. In addition, we evaluate our model on 200 real-world Deepfake-in-the-Wild (DW) videos of 50 celebrities available on the Internet and achieve 89.49% zero-shot accuracy, which is significantly higher than the best baseline model (gaining 10.77%), demonstrating and validating the practicability of our approach.

Keywords: Digital Forensics · Domain Adaptation · Few-shot Learning · Weakly-supervised Learning · Deepfake Detection

1 Introduction

Deepfakes generated by deep learning approaches such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) have sparked initial amusement and surprise in the field. However, the excitement has soon faded away and worries arose due to potential misuse of these technologies. Such concerns have become real, and deepfakes are now popularly used to generate fake videos, especially pornography, containing celebrities' faces [6,14,25]. In particular, these types of deepfakes are causing severe damage in the US, UK, and South Korea with 41% of the victims being American and British actresses, and 25% being female K-pop stars [7]. These images and videos are starting to become widespread on the Internet and are rapidly propagating through social

medias [6,9]. Therefore, deepfakes-generated videos raise not only significant social issues, but also ethical and privacy concerns. A recent report [18] shows that nearly 96% of deepfake videos are porn with over 134 million views. However, despite the urgency of the problem, concrete media forensic tools for an effective detection of different types of deepfakes are still absent. Recently, several research methods have been proposed to detect GAN-generated images [26,27,16] and deepfake videos [11,3,28,29,25,20]. FaceForensic++ [22] offers fake video datasets. Most of the detection methods generally perform well only in detecting fake images generated in the same domain (i.e., training and testing data are created with the same method), but not in detecting those generated in a different domain (i.e., test deepfakes are generated with different methods). So if new deepfake generation methods are developed, can the existing detection methods effectively detect the new deepfakes? How can we detect new fake videos? Is there a systematic and more generalizable approach to detect new types of deepfakes? The main focus of our work is to develop a generalized detection model, which not only performs well in detecting fake videos generated in one source domain, but also in detecting fake videos created in other domains with high accuracy.

To achieve this, we propose a Transfer learning-based Autoencoder with Residuals (TAR) to improve the detection performance of different types of deepfakes. We also introduce a latent space Facilitator module with shortcut residual connections in the network to learn deep features of deepfakes effectively and apply transfer learning between different domains. Our few-shot sequential transfer learning procedure utilizes only a few data samples (only 50 frames, which is around 2 seconds of a video) to detect deepfake videos in different domains (e.g., transfer learning to Face2Face with FaceSwap trained model). Our contributions are summarized as follows:

- We propose a Transfer learning-based Autoencoder with Residuals (TAR) to improve the detection performance of three different types of deepfakes, achieving over 99% accuracy and outperforming other baseline models.
- We sequentially apply transfer learning from one dataset to another to detect deepfake videos generated from different methods and achieve an average detection accuracy of 98.01% with a single model for all deepfake domains, demonstrating superior generalizability over the state-of-the-art approaches.
- Further, we evaluate and compare our approach using 200 real-world Deepfake-in-the-Wild videos collected from the Internet, and achieve 89.49% detection accuracy, which is significantly higher than the best baseline models.

2 Related Work

Rössler et al. [22] originally developed FaceForensics++, with 5,000 videos (1,000 Real and 4,000 Fake) using four different deepfake generation methods to benchmark deepfakes detection performance. Facebook also released Deep Fake Detection Challenge (DFDC) dataset [8] with 5,250 videos (1,131 Real and 4,119 Fake), while Google released 3,363 videos (363 Real and 3,000 Fake) as part

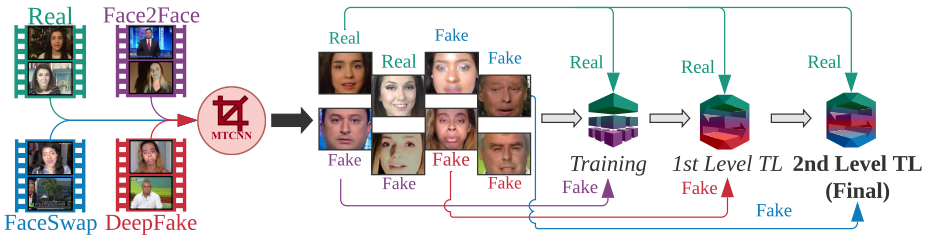

Fig. 1. Overview of our approach. Our proposed method learns features of deepfakes via Autoencoder with Residual blocks. Then, our multi-level transfer learning is sequentially performed with a small amount of training set in new target domains and the final model is created to effectively detect various types of deepfakes, including unseen or new deepfakes.

of the FaceForensics++ dataset [22]. For the detection of such deepfakes, deep learning-based methods in a supervised setting have shown high detection accuracy. Many Convolutional Neural Network (CNN)-based methods rely on content suppression using high-pass filters, either as a first layer [21] or using some handcrafted [4] and learned features [1]. Also, image splice or visual artifacts detection methods [13,23,17] try to exploit the inconsistency arising from splicing near the boundaries of manipulated sections in the image. Zhou *et al.* [31] built upon the idea of a high-pass filter method and attached a CNN-based model to capture both high-level and low-level details of the image. Tariq *et al.* [26,27] proposed ShallowNet, an efficient CNN-based network to detect GAN-generated images with high accuracy even at a resolution as low as 64×64. Rössler *et al.* [22] significantly improved the performance on compressed images, which is essential for detecting deepfakes on social networking sites such as Facebook, Twitter, and WhatsApp. However, in the case of detecting new unseen deepfake generation methods, most approaches have failed to achieve high performance.

To address this challenge, we explore few-shot transfer learning through our TAR model. Since there will be new deepfake generation methods in the future, it would not be always feasible to collect and generate a large amount of new deepfake samples. To cope with this challenge, transfer learning (TL) with few-shot is the key to detect deepfakes generated through different methods (domains), that is, what has been learned in one domain (e.g., FaceSwap (FS)) can be exploited to improve generalization in another domain (e.g., Face2Face (F2F)). In this example sequence, we assume that Face2Face surfaced as a new deepfake method, with only a few videos to train. This approach enables the model to adapt to the new type of deepfake quickly. Nguyen *et al.* [19] implemented fine-tuning with 140 videos to the pretrained model to detect unseen deepfake generation method. However, it is hard to construct a dataset including 140 videos for the newly emerging deepfake generation method in a realistic scenario. Also, Cozzolino *et al.* [5] have explored the possibility of generalizing a single detection method to detect multiple deepfake target domains. In this

work, we compare our approach against ForensicsTransfer [5] to demonstrate an improved generalizability. In particular, we further compare our results with other approaches and test with unseen real-world Deepfake-in-the-Wild videos generated by unknown methods.

3 Our Approach

Our Transfer learning-based Autoencoder with Residuals (TAR) model comprises a residual-based autoencoder with a latent space Facilitator module and transfers knowledge to learn deep features from different deepfake videos. The high-level overview of our approach is presented in Fig. 1. We train our initial model on one source domain dataset, such as DeepFakes [10] and apply transfer learning to the developed model using a small amount (few-shot learning) of data from another domain (e.g., FaceSwap [15]) to effectively learn features from both domains. Similarly, we perform additional transfer learning with a small amount of data in other domains(e.g., Face2Face [30]). After a sequence of transfer learning processes, we obtain our final model, as shown in Fig. 1, which can better detect deepfake videos generated by different methods (domains).

3.1 Base Network Architecture

We design our base network structure, as shown in Fig. 2, and develop an autoencoder architecture consisting of Residual blocks with a Facilitator module to explicitly force and divide the latent space between real and fake embedding. By comparing the activations in real and fake latent spaces, we can differentiate between real and fake images.

In addition, our proposed use of Residual blocks and Leaky ReLU is necessary to overcome the instabilities/weakness of prior approaches [26,27,5], such as "zero activations" (i.e., when both real and fake parts have no activation), while performing transfer learning experiments. We also discuss their limitations which motivate us to develop our TAR model in the Ablation Study section. In particular, the TAR architecture is deeper, more stable, and is readily extendable with more shortcut connections to better capture deep features for different types of deepfake detection, compared to other approaches. In our model, the encoder maps an image $\mathbf{x}$ to the latent space vector $\mathbf{H}$ and the decoder maps $\mathbf{H}$ to the reconstructed image $\mathbf{x}'$ as follows: $\mathbf{H}=\sigma(\mathbf{W}\mathbf{x}+\mathbf{b})$ and $\mathbf{x}'=\sigma'(\mathbf{W}'\mathbf{H}+\mathbf{b}')$, where $\mathbf{W}$ and $\mathbf{W}'$ are the weight matrices, $\mathbf{b}$ and $\mathbf{b}'$ are the bias vectors, and σ and σ' are the element-wise activation functions for the input images and the reconstructed images, respectively. Further, we divide the latent space $\mathbf{H}$ into two parts as follows: $\mathbf{H}=\{\mathbf{H_1},\mathbf{H_2}\}$, where we define the elements of the tensor $\mathbf{H_1}$ for real images and those of $\mathbf{H_2}$ for fake images at coordinates (i,j,k) by $(\mathbf{H_1})_{i,j,k}$ and $(\mathbf{H_2})_{i,j,k'}$. The dimension of each of the tensors $(\mathbf{H_1})_{i,j,k}$ and $(\mathbf{H_2})_{i,j,k'}$ is $M\times M\times N$, and the dimension of $(\mathbf{H})_{i,j,k}$ is $M\times M\times 2N$, where $1\leq i\leq M$, $1\leq j\leq M$, $1\leq k\leq N$, and $N+1\leq k'\leq 2N$ (e.g., $M=15$ and $N=64$ are used in our work).

Facilitator Module. For input image $\mathbf{x}_m$, the encoder generates the latent space output as follows: $Encoder(\mathbf{x}_m)$=$\mathbf{H}_m$=$\{\mathbf{H}_{m,1},\mathbf{H}_{m,2}\}$. In order to separate real and fake latent features, we implemented a Facilitator module shown in Fig. 2, which forces the activations to be zeros for the opposite class and keeps the corresponding class. The activated $(\mathbf{H_m})_{i,j,k}$ by the Facilitator is defined as follows:

$$(H_m)_{i,j,k} = \begin{cases} (H_{m,1})_{i,j,k} & \text{if } k \leq N, c = 1 \ (real) \\ (H_{m,2})_{i,j,k} & \text{if } k > N, c = 2 \ (fake) \\ 0 & \text{otherwise} \end{cases}, \tag{1}$$

where c is the class, $(\mathbf{H_m}) \in \mathbb{R}^{M \times M \times N}$ and $1 \leq i \leq M, 1 \leq j \leq M$, and $1 \leq k \leq 2N$.

Loss Functions. We define two loss functions, L_{Activ} for the real and fake activations, and L_{Recon} for the reconstruction. We combine these two loss functions with Residual blocks with skip connections to improve the performance:

$$L = \lambda L_{Recon} + L_{Activ}. \tag{2}$$

First, we define the per-class activation of the latent space $A_{m,c}$ to be the L^1 norm of $A_{m,c}/K_c$, where K_c is the number of features of $\mathrm{H}_{m,c}$ and $c \in \{1, 2\}$. For the activation of an input $\mathbf{x}_m$ with a given label $l_m \in \{1, 2\}$ and a latent feature tensor $\mathrm{H}_{m,c}$, L_{Activ} and L_{Recon} are defined as follows:

$$L_{Activ} = \sum_m |A_{m,2} - l_m + 1| + \sum_m |A_{m,1} + l_m - 2|, \tag{3}$$

and we use L^1–norm for reconstruction loss (L_{Recon}),

$$L_{Recon} = \frac{\sum_m ||x_m - x_{m'}||_1}{N}. \tag{4}$$

For an efficient embedding of L_{Activ}, we set λ as 0.1 as determined empirically, thereby focusing more on embedding real and fake features through the encoder.

Classification and Leaky ReLU. In testing phase, if $A_{m,2} > A_{m,1}$, we consider the input $\mathbf{x}_m$ as fake. Otherwise, we classify it as real. However, for some small input images, we have empirically observed that $A_{m,1} = A_{m,2} = 0$ (zero activations). In this case, we cannot calculate the loss function or further perform classification. In order to prevent such situation, we use the Leaky ReLU function at the end of the last Convolution layer of the encoder and employ skip connections (see Fig. 2), where the Leaky ReLU function with a small slope allows effectively calculating the loss and classifying very small input images.

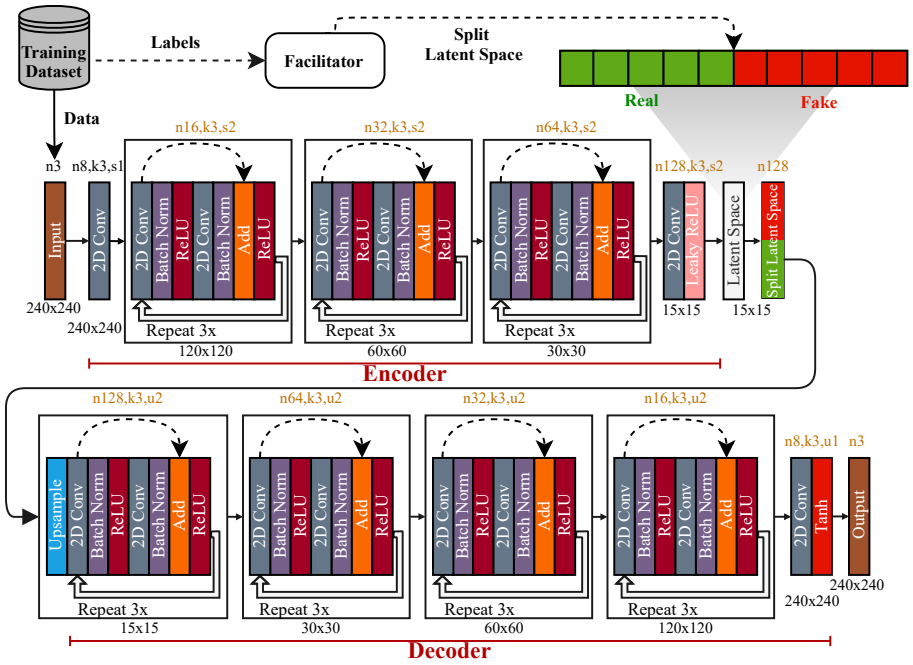

Fig. 2. Our TAR model architecture. Depending on the label of the input, real or fake, different latent space representations are enforced by the Facilitator module.

Residual Block with Shortcut Connections. We implement Residual blocks with shortcut connections inside our autoencoder architecture, which can effectively learn deep features of images and keep the feature information through the network. As shown in Fig. 2, we add three Residual blocks in the encoder and four Residual blocks in the decoder. Each Residual block repeats three times and each block consists of a Convolutional (Conv) layer, a Batch Normalization (BN) layer, and a ReLU, which is followed by another Conv layer and BN layer. Next, we add the input to the Addition layer to build the skip connection and pass it through the final ReLU (see Fig. 2). After applying Residual blocks, the number of Conv layers increases to 45, which can lead to vanishing gradient problems. However, as mentioned by He et al. [12], we can easily optimize a deeper network by adding the residuals or skip connections to the model structure.

Encoder. The input images have a resolution of 240×240× 3 (see Fig. 2). We use three Residual blocks (120×120×16, 60×60×32, and 30×30×64) in the encoder as shown in Fig. 2, where each Residual block repeats three times. The 2D Conv inside each Residual block, downsamples the input with a stride of 2 at the first loop (240×240 to 120×120, 120×120 to 60×60, etc.). After all the Residual blocks, we define one 3×3 Conv layer followed by a Leaky ReLU with a small slope value of 10^{-7} to avoid the zero activation problem described

earlier. At the end of the encoder, we chose $M = 15$ and $2N = 128$, to build a latent feature tensor ($\mathbf{H}$) with a size of 15×15×128. Therefore, the first 64 features ($\mathbf{H_1}$) are used to capture real images, and the other 64 features ($\mathbf{H_2}$), to represent fake images (see Fig. 2). During training, for each input $\mathbf{x}_m$, the selection block is chosen by the Facilitator module according to the class label $l_m \in \{1, 2\}$ to set zero values for non-activated class and activate only for the input class in Eq. 1.

Decoder. We configured the decoder using Residual blocks such as the encoder in reverse order as shown in Fig. 2. The difference from the encoder is that all Conv layers inside the decoder's Residual blocks have a stride of 1 since we are not reducing the size. To recover the original input image size, we scale up the input size by 2, using a 2×2 nearest neighbors upsampling layer before the first Residual block. The kernel size of the Conv layer is set to be the same as that of the encoder network. In the last layer, a single 3×3 Conv layer is used with tanh as the activation function.

3.2 Multi-Level Sequential Transfer Learning

Since our goal is to develop an architecture that can detect deepfakes generated by different methods, we perform multi-level sequential transfer learning, as shown in Fig. 1. First, we train each model with respect to a specific generation method (domain) one at a time. For transfer learning, we performed few-shot learning from source domain s to target domain t. We call it first-level transfer learning, denoted by $s \rightarrow t$, where $s \neq t$, and $s, t \in \{FS, F2F, DF\}$. After this, we also conduct additional transfer learning toward a second target domain u using the previous transfer learned model; we call it second-level transfer learning, which we denote it by $(s \rightarrow t) \rightarrow u$, where $s \neq t \neq u$, and $s, t, u \in \{FS, F2F, DF\}$. The $\rightarrow$ arrow denotes the domain transfer sequence, where we only use 50 fake frames (2 seconds of a video) from the target domain to perform sequential few-shot transfer learning.

The main objective of our work is to develop a deployable, unified deepfake detection framework. If a new deepfake generation method is surfaced, the amount of data for training the model will be limited. We overcome this scenario with our multi-level sequential few-shot transfer learning approach. The first and second-level transfer learning simulates this scenario where each level represents the situation when a new deepfake generation method is present. We only re-train on few frames from a new type of deepfake video and update the model, preserving all the previous detection capacity.

4 Experiments

We evaluated our method on a popular benchmark dataset to measure the performance of a *single model* detecting various types of deepfakes simultaneously.

Table 1. Dataset summary for different experiments.

Category	Total Videos	Extracted Frames	D1: Base Dataset (Real:Fake)	D2: TL Dataset (Real:Fake)	D3: Test Dataset (Real:Fake)
Pristine (Real)	1,000	16,050	-	-	-
DeepFake (DF)	1,000	16,050	15,000:15,000	50:50	500:500
FaceSwap (FS)	1,000	16,050	15,000:15,000	50:50	500:500
Face2Face (F2F)	1,000	16,050	15,000:15,000	50:50	500:500
Deepfake-in-the-Wild (DW)	200	8,164	-	-	8,164:8,164

4.1 Dataset

FaceForensics++ Dataset. We utilized three different types of deepfake videos from Rössler et al. [22]. For each type, we extracted a total number of 16,050 frames from the videos. The details of the dataset are provided in Table 1.

Deepfake-in-the-Wild (DW) Dataset. To validate the effectiveness and practicability of the detectors, we used 200 freely accessible Deepfake-in-the-Wild (DW) videos of 50 celebrities from the Internet. From the 200 DW videos, we extracted a total number of 8,164 frames.

Dataset Settings. For training and testing, we divided the above-mentioned datasets into three categories (**D1**-**D3**) with no overlap to accurately characterize the performance.

- **D1: Base dataset.** The purpose of this dataset is to test the performance of a model for one source domain on which the model is originally trained. We also measure the zero-shot performance for the dataset on which the model is not trained. To evaluate the performance, we use 500 real and 500 fake images from the extracted frames as the test set, hence a total of three pairs of datasets, DF, FW, and F2F, as shown in Table 1. For simplicity, we refer these dataset pair with the fake class dataset name. For example, the Pristine and DeepFake dataset pair is referred to as DeepFake (DF).
- **D2: Transfer (few-shot) learning dataset.** We used the samples of 50 (2 seconds of a video) real and fake frames each. For evaluation, we used the test set from D3 (500 real and 500 fake frames) to check the model's performance.
- **D3: DW dataset.** We used this dataset only for testing the generalizability of our final model, as this dataset contains unseen videos generated with unknown deepfake generation methods. Therefore, this dataset can be served to assess realistic performance of the detection methods.

4.2 Evaluation Settings

Baseline models. We compared our method against ShallowNet [27], Xception [2], and ForensicTransfer [5].

Table 2. Accuracy of base (green) and zero-shot performance accuracy (gray).

Methods	Base Dataset: DF (%)			Base Dataset: FS (%)			Base Dataset: F2F (%)		
	DF	*FS*	*F2F*	*FS*	*DF*	*F2F*	*F2F*	*DF*	*FS*
ShallowNet	63.0	48.0	53.0	58.0	44.0	46.0	57.0	51.0	50.0
Xception	99.6	50.0	50.3	92.0	50.1	51.5	99.3	71.4	50.4
FT	**99.8**	50.0	70.4	94.3	47.1	53.7	99.3	99.0	**73.5**
TAR(*Ours*)	**99.8**	**50.1**	**75.3**	**99.3**	**50.8**	**70.5**	**99.5**	**99.7**	52.2

Machine configurations. We used Intel i9-9900k CPU 3.60GHz with 64.0GB RAM and NVIDIA GeForce Titan RTX. We tried our best to implement the baseline methods to their original description.

Training and Multi-level Sequential Transfer Learning details. We trained each model with respect to one specific generation method at a time. For transfer learning, we performed few-shot learning from source domain (DF or FS or F2F) to target domain (FS or F2F, or DF) as the first-level transfer learning and denote it as $s \rightarrow t$, where $s \neq t$, and $s, t \in \{FS, F2F, DF\}$. We also conducted transfer learning toward a second target domain using the previous transfer learned model as the second-level transfer learning. We denote second-level transfer learning as $(s \rightarrow t) \rightarrow u$, where $s \neq t \neq u$, and $s, t, u \in \{FS, F2F, DF\}$.

5 Results

The detection performances of different methods are presented in Tables 2-4, where the highest accuracy values are in bold face. The last 'Avg. Gain' column in Table 3 and 4 reports the average accuracy gain from the best baseline model of each method in detecting all deepfake domain datasets.

5.1 Base dataset performance (D1)

We compared our approach with three baseline models using D1 as shown in Table 2. First, after training with three source domains, our TAR model achieves higher accuracies of 99.80%, 99.30%, and 99.50% in detecting DF, FS and F2F, respectively, compared to ShallowNet (63.00%, 58.00%, and 57.00%), Xception (99.60%, 92.00%, and 99.30%) and ForensicTransfer (FT) (99.80%, 94.30%, and 99.30%). Generally, Xception and FT also performed well on base datasets, while ShallowNet performed poorly.

5.2 Zero-shot performance (D1)

Next, we also evaluated the base dataset-trained model with other domain datasets (i.e., testing the DF-trained model on F2F) for zero-shot performance comparison. The FT model achieved 73.50% accuracy, which is 21.30% higher

Table 3. Accuracy of first-level TL (blue) from base dataset (green).

Transfer Seq.	Test Data	Models			Avg. Gain
		Xce	FT	*TAR*	
DF → FS	DF	67.2	90.1	**91.6**	-2.0
	FS	54.0	89.9	**93.3**	
	F2F	62.6	**90.9**	80.1	
DF → F2F	DF	95.8	85.0	**98.9**	8.8
	F2F	69.7	81.6	**93.1**	
	FS	53.5	45.3	**53.5**	
FS → DF	FS	**92.0**	48.2	69.0	17.5
	DF	45.8	79.8	**98.5**	
	F2F	68.0	67.4	**90.8**	
FS → F2F	FS	**99.0**	56.6	84.6	16.0
	F2F	57.7	55.1	**84.7**	
	DF	49.8	56.5	**85.3**	
F2F → DF	F2F	95.4	98.7	**99.2**	1.7
	DF	92.6	99.0	**99.5**	
	FS	**68.7**	60.9	64.9	
F2F → FS	F2F	87.6	95.8	**97.7**	3.3
	FS	76.2	92.7	**97.4**	
	DF	87.2	94.4	**97.8**	

Table 4. Accuracy of second-level TL (blue) from first-level (green).

Transfer Seq.	Test Data	Models			Avg. Gain
		Xce	FT	*TAR*	
(DF→FS) →F2F	DF	88.2	64.3	**91.9**	17.1
	FS	55.1	50.0	**85.0**	
	F2F	71.7	62.3	**89.4**	
(DF→F2F) →FS	DF	75.4	90.0	**97.8**	6.5
	F2F	55.0	89.4	**94.9**	
	FS	67.5	89.8	**95.9**	
(FS→DF) →F2F	FS	48.8	66.5	**90.2**	18.1
	DF	**97.0**	47.3	84.9	
	F2F	62.7	57.8	**87.8**	
(FS→F2F) →DF	FS	48.4	77.5	**96.8**	22.7
	F2F	**96.4**	48.6	85.2	
	DF	64.7	59.9	**95.5**	
(F2F→DF) →FS	F2F	76.2	94.0	**98.5**	4.2
	DF	76.2	93.6	**97.4**	
	FS	86.9	93.9	**98.3**	
(F2F→FS) →DF	F2F	92.7	**97.4**	97.3	8.4
	FS	66.6	70.7	**96.7**	
	DF	95.9	**97.8**	97.1	

than that of TAR for F2F-trained model tested on FS. However, in all other zero-shot testing cases, our TAR model outperformed all other approaches for all datasets as shown in Table 2. Overall, our approach performed better than the-state-of-the-art methods in the same source domain, as well as different domains (zero-shot) except for one case.

5.3 Transfer learning performance (D2)

As shown in the last column of Table 3, for the first-level transfer learning, we observed that our TAR model gained average accuracy over the best performing baseline (Xception or FT) on all sequences ($s \rightarrow t$) except for only one case (DF→FS) in which FT achieved 90.30% average performance. By comparing Table 2 and 3, we observed that after transfer learning to a target domain, the accuracy of the target domain increases, but the base class accuracy decreases for all methods. For example, when we trained all three models on the DF base dataset, all methods had a test accuracy of over 99%, and a zero-shot accuracy of around 50% on the FS dataset (see Table 2). Now, when we performed transfer learning from this source domain (DF) to target domain (FS), the accuracy of FT increases from 50.00% to 89.90%, and TAR increases from 50.10% to 93.30%. However, the source domain accuracy, which was 99.80% for both FT and TAR, decreases to 90.10% and 91.60% respectively (see Table 2 and 3). For Xception, the drop was even worse, with a 99.60% to 67.20% drop on the source domain,

Table 5. Zero-shot performance on Deepfake-in-the-Wild (DW) dataset. The → arrow denotes the domain transfer sequence.

Methods	**Domain Transfer Sequence** (Best Performer)	**Original Acc.** (%)	**Contrast (C) Acc.** (%)	**Brightness (B) Acc.** (%)	**C + B Acc.** (%)
Xception	F2F→FS→DF	50.23	58.11	66.92	78.72
FT	F2F→DF→FS	47.05	46.37	47.60	48.04
TAR (*Ours*)	F2F→DF→FS	**67.87**	**69.02**	**77.06**	**89.49**

while gaining only 6% on target domain. Compared to FT and TAR, Xception shows limited learning capacity during few-shot sequential transfer learning.

As shown in Table 4, after the second-level transfer learning, our approach achieves a higher performance over every sequence $((s \rightarrow t) \rightarrow u)$ compared to Xception and FT. The best performing sequence for both TAR and FT has "F2F→DF" as the source domain and FS as target domain, while Xception's best sequence is "F2F→FS" as the source and DF as the target. We can write the resultant domain transfer sequence as "(F2F→DF)→FS" for TAR and FT, and "(F2F→FS)→DF" for Xception. The best performing FT model "(F2F→DF)→FS" achieved an average accuracy of 93.83% on all three datasets, while the best performing TAR model "(F2F→DF)→FS" achieved 98.01% gaining 4.2% (see Table 4). However, Xception was not able to produce a similar generalization performance as FT or TAR, achieving an average accuracy of 85.01%. Although there are 4 cases out of 18 in which the performances of Xception and FT are better than that of TAR, the overall performance of TAR is better. Therefore, our approach generalizes much better and yields above 97% detection accuracy across all three datasets and shows significantly improved performance compared to zero-shot learning, as shown in Table 2. Furthermore, FT's original work [5] conducted experiments on multiple domains only on one case, whereas our research dealt with more complex cases to explore correlations between deepfake domains and to develop more generalized models.

5.4 Unseen DW dataset performance (D3)

It is critical to test the models against real-world deepfake data in addition to benchmark datasets. Therefore, we used 200 Deepfake-in-the-Wild (DW) videos to evaluate the models in real-world scenarios. The model zero-shot detection accuracy on this dataset is shown in Table 5, where we used the best performing multi-level transfer learning sequence, "(F2F→DF)→FS", for both FT and TAR models and "(F2F→FS)→DF" for Xception. As shown in the third column of Table 5, the accuracy of the second best performer from Table 4, FT, was only 47.05%. While TAR achieves an accuracy of 67.87% and 50.23% for Xception, which is higher than that of FT, it is much lower than the results for the previous datasets.

Through our evaluation, this low performance strongly indicated that deepfakes in the wild were generated using more diverse, different, and complex methods than DF, F2F, and FS. Possibly, they could have leveraged additional

post-processing methods to improve the quality of deepfakes. For example, we first noticed that the lighting in the DW dataset tends to be darker than that in the benchmark dataset. Hence, we first tried to balance the brightness and contrast of the DW videos [24] by increasing both the darkness and contrast by 30% to better match the lighting conditions with the benchmark datasets. As shown in Table 5, increasing only the contrast yielded a slight accuracy improvement from 67.87% to 69.02% for the TAR model and 50.23% to 58.11% for Xception, but FT model has a drop in accuracy from 47.05% to 46.37%. After adjusting the brightness of the DW frames, our TAR model achieved a 77.06% accuracy, which is almost 10% higher than in the previous case. After applying both contrast and brightness changes together, Xception yields 78.72% accuracy, while our approach achieves 89.49%, which is much higher. On the other hand, the best performing baseline FT was not significantly improved, only achieving 48.04% accuracy at best. Therefore, our approach adapting to new domains with multi-level transfer learning clearly outperforms other methods on unseen real-world dataset, demonstrating the effectiveness of TAR against real-world deepfakes.

We plan to train our model with both brightness and contrast adjustments using data augmentation in the future. In this way, we can improve the performance of detecting DW or other new types of deepfakes without any inference-stage adjustments.

6 Analysis & Discussion

6.1 Classification Activation Map (CAM)

To observe the fake frame activations of our model, we utilized CAM to generate the implicit attention of CNNs on an image. We used the best performing second-level transfer learned TAR model, "(F2F→DF)→FS", to show the activations from the decoder. In Fig. 3, we present a side-by-side comparison of 1) original real versus fake images, 2) CAM outputs, and 3) overlaid images of original images and CAM, using three different examples from each dataset (DF, F2F, and FS). We also present sample images from the DW dataset in the same order, where we do not have an original real input image. The DW face images are intentionally blurred to hide the identity. Figure. 3 clearly shows that our model produces much intense fake activation around the face than real images; the activated part is focused on the nose, the forehead, the cheek, etc. Since our model is a multi-level transfer-learned across DF, F2F, and FS, we did not observe any particular activation patterns for a specific model, and our approach detects all deepfake types effectively in a similar way.

6.2 Ablation Study

We conducted an ablation study to check the impact of Leaky ReLU and residual connections in our model and summarized our finding. First, we removed the

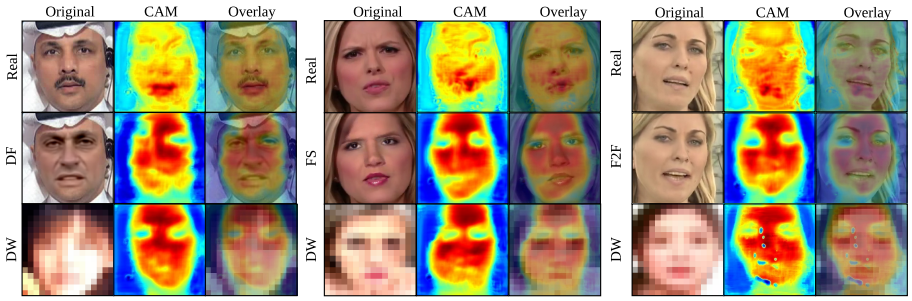

Fig. 3. A side-by-side comparison of 1) real vs. fake frames, 2) Class Activation Map (CAM) outputs, and 3) overlaid images of original input and CAM, using three different examples from each base dataset (DF, F2F, and FS). We also present sample images from the DW dataset in the same order, where we do not have an original real input image. The DW face images are intentionally blurred to hide the identity.

residual addition from our Residual blocks and replaced Leaky ReLU with ReLU. Our model started to show several zero activations in the last layer of the encoder for both real and fake images during training. We observed that this phenomenon occurred more frequently on small images. Further research is needed to find the exact causes of this phenomenon. Next, we replaced the ReLU function with a Leaky ReLU and re-ran the same experiment. This time, there were much less zero activations, because the Leaky ReLU allows small negative values, instead of zeros. To completely eliminate this zero activation issue, we added back the residual addition inside Residual blocks. As a result, we did not observe any zero activation cases during training anymore, even with small, upscaled images. This ablation experiment supports our motivation to use Residual blocks and Leaky ReLU function for our TAR model.

6.3 Discussion

Recently, deepfake porn videos have surged, with the vast majority featuring female celebrities' face photos to develop sexually explicit videos without the celebrities' knowledge or consent. These videos, rapidly spreading throughout the Internet, cause serious issues as they harass innocent people in the cyberspace. Because of the urgency of the problem and the absence of effective tools to detect these videos, we have undertaken this research to investigate real-world deepfake videos, in order to construct the Deepfake-in-the-Wild (DW) video dataset. We obtained freely available videos from the Internet, and all researchers in this study were informed about the detailed research protocol. We also consulted with the Institutional Review Board (IRB) in our institution, and received confirmation saying that approval was not required, since the videos are already available on the Internet. Moreover, we cropped the face regions and deleted the rest of the images, and further discarded the original videos.

To preserve and protect the privacy of the celebrities, we blurred their faces in this work. Also, we do not plan on distributing any explicit content and leak celebrities' private information.

6.4 Deployment

To address the urgency of this matter, we have developed an online testing website since Dec. 2019 [1], where people can upload images to check deepfakes using our model. We developed the interface such that it provides the probability of the image being fake and demonstrated that TAR can be practically deployed and used to circumvent the deployed real-world deepfakes.

7 Conclusion

Malicious applications of deepfakes, such as deepfake porn videos, are becoming increasingly prevalent these days. In this work, we propose TAR to improve the generalized performance of multi-domain deepfake detection and test it on unseen real-world deepfake videos to evaluate its practicability. A facilitator module splits our TAR model's latent space into real and fake latent spaces, enabling the encoder to focus more on learning the latent space representation, which results in a more accurate real and fake classification. Our multi-level sequential transfer learning-based autoencoder with Residual blocks significantly outperforms other state-of-the-art methods in detecting multi-domain deepfakes from the FF++ dataset. Furthermore, TAR achieves 89.49% accuracy in detecting 200 real-world Deepfake-in-the-Wild videos of 50 celebrities, which is significantly higher than the state-of-the-art methods. Our results suggest that sequential learning-based models are an interesting venue to explore for analyzing and detecting fake and manipulated media in real-world settings and other analogous vision tasks that require domain adaptation. The code of our work is available on GitHub[2].

Acknowledgments

This work was partly supported by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.2019-0-00421, AI Graduate School Support Program(Sungkyunkwan University)), (No. 2019-0-01343, Regional strategic industry convergence security core talent training business) and the Basic Science Research Program through National Research Foundation of Korea (NRF) grant funded by Korea government MSIT (No. 2020R1C1C1006004). Additionally, this research was partly supported by IITP grant funded by the Korea government MSIT (No. 2021-0-00017, Original Technology Development of Artificial Intelligence Industry) and was partly supported by the Korea government MSIT, under the High-Potential Individuals Global Training Program (2019-0-01579) supervised by the IITP.

[1] http://deepfakedetect.org
[2] https://github.com/Clench/TAR_resAE

References

1. Bayar, B., Stamm, M.C.: A deep learning approach to universal image manipulation detection using a new convolutional layer. In: Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security. pp. 5–10. ACM (2016)
2. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1251–1258 (2017)
3. Ciftci, U.A., Demir, I., Yin, L.: Fakecatcher: Detection of synthetic portrait videos using biological signals. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)
4. Cozzolino, D., Poggi, G., Verdoliva, L.: Recasting residual-based local descriptors as convolutional neural networks: an application to image forgery detection. In: Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security. pp. 159–164. ACM (2017)
5. Cozzolino, D., Thies, J., Rössler, A., Riess, C., Nießner, M., Verdoliva, L.: Forensictransfer: Weakly-supervised domain adaptation for forgery detection. arXiv preprint arXiv:1812.02510 (2018)
6. Croft, A.: From porn to scams, deepfakes are becoming a big racket—and that's unnerving business leaders and lawmakers (2019), https://fortune.com/2019/10/07/porn-to-scams-deepfakes-big-racket-unnerving-business-leaders-and-lawmakers, [Online; accessed 8-Jan-2021]
7. Dickson, E.: Deepfake Porn Is Still a Threat, Particularly for K-Pop Stars (2019), https://www.rollingstone.com/culture/culture-news/deepfakes-nonconsensual-porn-study-kpop-895605, [Online; accessed 8-Jan-2021]
8. Dolhansky, B., Howes, R., Pflaum, B., Baram, N., Ferrer, C.C.: The deepfake detection challenge (dfdc) preview dataset. arXiv preprint arXiv:1910.08854 (2019)
9. Edwards, C.: Making deepfake porn could soon be as 'easy as using Instagram filters', according to expert (2019), https://www.thesun.co.uk/tech/9800017/deepfake-porn-soon-easy, [Online; accessed 8-Jan-2021]
10. FaceSwap Devs: Deepfakes faceswap - github repository. https://github.com/deepfakes/faceswap (2021), [Online; accessed 8-Jan-2021]
11. Güera, D., Delp, E.J.: Deepfake video detection using recurrent neural networks. In: 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). pp. 1–6. IEEE (2018)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
13. Huh, M., Liu, A., Owens, A., Efros, A.A.: Fighting fake news: Image splice detection via learned self-consistency. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 101–117 (2018)
14. Kan, M.: Most AI-Generated Deepfake Videos Online Are Porn (2019), https://www.pcmag.com/news/most-ai-generated-deepfake-videos-online-are-porn, [Online; accessed 8-Jan-2021]
15. Kowalski, M.: Faceswap - github repository. https://github.com/MarekKowalski/FaceSwap (2021), [Online; accessed 8-Jan-2021]
16. Lee, S., Tariq, S., Shin, Y., Woo, S.S.: Detecting handcrafted facial image manipulations and gan-generated facial images using shallow-fakefacenet. Applied Soft Computing **105**, 107256 (2021). https://doi.org/10.1016/j.asoc.2021.107256

17. Matern, F., Riess, C., Stamminger, M.: Exploiting visual artifacts to expose deepfakes and face manipulations. In: 2019 IEEE Winter Applications of Computer Vision Workshops (WACVW). pp. 83–92. IEEE (2019)
18. Mehta, I.: A new study says nearly 96% of deepfake videos are porn (2019), https://tnw.to/OXuTG, [Online; accessed 8-Jan-2021]
19. Nguyen, H.H., Fang, F., Yamagishi, J., Echizen, I.: Multi-task learning for detecting and segmenting manipulated facial images and videos. In: 2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS). pp. 1–8. IEEE (2019)
20. Nguyen, H.H., Yamagishi, J., Echizen, I.: Capsule-forensics: Using capsule networks to detect forged images and videos. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 2307–2311. IEEE (2019)
21. Rao, Y., Ni, J.: A deep learning approach to detection of splicing and copy-move forgeries in images. In: 2016 IEEE International Workshop on Information Forensics and Security (WIFS). pp. 1–6. IEEE (2016)
22. Rössler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., Nießner, M.: Faceforensics++: Learning to detect manipulated facial images. In: ICCV 2019 (2019)
23. Salloum, R., Ren, Y., Kuo, C.C.J.: Image splicing localization using a multi-task fully convolutional network (mfcn). Journal of Visual Communication and Image Representation **51**, 201–209 (2018)
24. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. Journal of Big Data **6**(1), 60 (2019)
25. Tariq, S., Jeon, S., Woo, S.S.: Am i a real or fake celebrity? measuring commercial face recognition web apis under deepfake impersonation attack (2021)
26. Tariq, S., Lee, S., Kim, H., Shin, Y., Woo, S.S.: Detecting both machine and human created fake face images in the wild. In: Proceedings of the 2nd International Workshop on Multimedia Privacy and Security. pp. 81–87. ACM (2018)
27. Tariq, S., Lee, S., Kim, H., Shin, Y., Woo, S.S.: Gan is a friend or foe?: a framework to detect various fake face images. In: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing. pp. 1296–1303. ACM (2019)
28. Tariq, S., Lee, S., Woo, S.S.: A convolutional lstm based residual network for deepfake video detection (2020)
29. Tariq, S., Lee, S., Woo, S.S.: One detector to rule them all: Towards a general deepfake attack detection framework. In: Proceedings of The Web Conference 2021 (2021). https://doi.org/10.1145/3442381.3449809
30. Thies, J., Zollhofer, M., Stamminger, M., Theobalt, C., Nießner, M.: Face2face: Real-time face capture and reenactment of rgb videos. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2387–2395 (2016)
31. Zhou, P., Han, X., Morariu, V.I., Davis, L.S.: Learning rich features for image manipulation detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1053–1061 (2018)