FReTAL: Generalizing Deepfake Detection using Knowledge Distillation and Representation Learning

Minha Kim*, Shahroz Tariq*
College of Computing and Informatics
Sungkyunkwan University, South Korea
{kimminha,shahroz}@g.skku.edu

Simon S. Woo†
Department of Applied Data Science
Sungkyunkwan University, South Korea
swoo@g.skku.edu

Abstract

As GAN-based video and image manipulation technologies become more sophisticated and easily accessible, there is an urgent need for effective deepfake detection technologies. Moreover, various deepfake generation techniques have emerged over the past few years. While many deepfake detection methods have been proposed, their performance suffers from new types of deepfake methods on which they are not sufficiently trained. To detect new types of deepfakes, the model should learn from additional data without losing its prior knowledge about deepfakes (catastrophic forgetting), especially when new deepfakes are significantly different. In this work, we employ the Representation Learning (ReL) and Knowledge Distillation (KD) paradigms to introduce a transfer learning-based Feature Representation Transfer Adaptation Learning (FReTAL) method. We use FReTAL to perform domain adaptation tasks on new deepfake datasets, while minimizing the catastrophic forgetting. Our student model can quickly adapt to new types of deepfake by distilling knowledge from a pre-trained teacher model and applying transfer learning without using source domain data during domain adaptation. Through experiments on FaceForensics++ datasets, we demonstrate that FReTAL outperforms all baselines on the domain adaptation task with up to 86.97% accuracy on low-quality deepfakes.

1. Introduction

Synthetic multimedia is becoming increasingly common on the Internet and social media [4, 34]. Its popularity is being driven by the widespread availability of simple tools and techniques for creating realistic fake multimedia information [23, 25, 45]. Recent advances in deep learning have aided in the generation methods for creat-

---
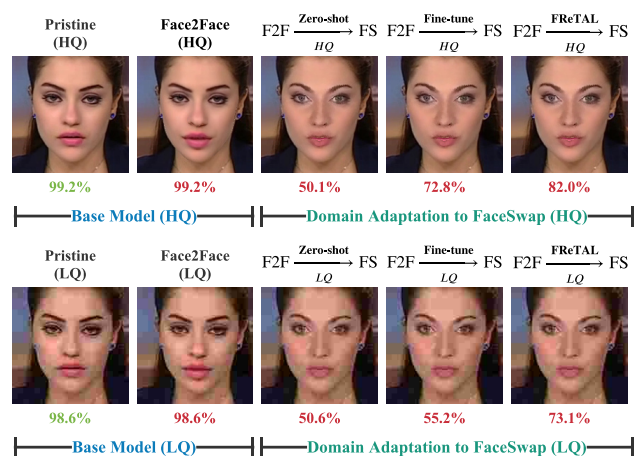
*equal contribution
†corresponding author

Figure 1. **Performance of domain adaptation task.** We use Face2Face (F2F) as the source and FaceSwap (FS) as target dataset. Xception is used as the backbone model. Here, we demonstrate two cases: 1) high-quality deepfakes (*top*) for source and target. 2) low-quality (*bottom*) deepfakes for source and target. Our FReTAL performs significantly better than zero-shot and fine-tuning on low quality deepfakes (more comparison in Section 5).

ing synthetic images and videos that look remarkably close to real-world images [37, 38, 12] and videos [14, 35, 69]. Especially, deepfakes are manipulated multimedia generated using such techniques, which generally involve neural networks such as Autoencoders (AE) [26] and Generative Adversarial Networks (GAN) [27]. Although such tools can help in automating game design [57], photorealistic scenery generation [61], film making [5], human face generation [37] or virtual and augmented reality rendering [62], they can also be very dangerous and misused if utilized for malicious purposes [16, 20, 36, 22]. The line between real and fake media is becoming increasingly blurred as manipulation techniques become more available, practical, and difficult to be detectable.

For example, using face swaps-based deepfakes, an at-

tacker can put the victims in place and setting where they have never been. This type of deepfakes is easily used to generate pornographic videos of celebrities and masses alike [56]. Also, by manipulating the lip movement with facial reenactment techniques and the associated speech signal, deepfake videos of people speaking words that they never said can be produced. For example, many fake propaganda videos are generated by reanimating real videos of political figures [82, 63, 19].

Among the most famous deepfake forgeries are human facial manipulations [67, 80, 45, 23, 79, 37]. However, due to a lack of data, detecting these deepfakes or forged images/videos is challenging in a real-world setting. To increase data availability, the research community has recently released a slew of deepfake datasets to assist other researchers around the world in developing detection mechanisms for such deepfakes. The FaceForensics++ [67] dataset is one of the earliest and most popular benchmark deepfake datasets. Many other deepfake benchmark datasets are recently released, such as Deepfake Detection Challenge [21] from Facebook and CelebDF [52].

As a result, these benchmark datasets facilitate improving performance and diversifying detection approaches. Many deepfake detection methods achieve high test accuracy on a single deepfake dataset [67, 1, 75]. However, they perform poorly on deepfakes created from novel methods that were not introduced during the training process [78]. In other words, they lack generalizability, especially on low-quality (compressed) deepfake videos, which are the focus of our paper as deepfakes shared on social media typically goes through compression. Therefore, such approaches trained on high quality videos generally perform poorly in the real world.

The successful demonstration of the deepfake impersonation attack on commercial and open-source face recognition and authentication APIs by Tariq et al. [74] highlights the importance of developing a generalized classifier that consistently performs well on various types of deepfakes. In particular, it is of paramount importance in the face authentication domain, and not much research has been conducted in that direction. Furthermore, it would also be unrealistic to generate a large dataset tailored toward each deepfake generation method to train a deepfake detector. Therefore, a more widely applicable generalized solution to the problem of deepfake detection is required. In this work, we aim to explore such an approach by keeping data scarcity and domain adaptation into consideration. *Note: it is relatively easy for the model to detect high-quality (uncompressed) deepfakes, as shown by previous research* [78, 67, 1], *thereby we focus more on low-quality deepfakes than high-quality in our work.*

In recent years, several Knowledge Distillation (KD)-based methods are proposed for domain adaptation tasks [6,

48, 58, 59, 60]. However, none of them have studied KD for domain adaptation in the media forensics domain, especially for deepfake detection. To this end, we propose Feature Representation Transfer Adaptation Learning (FReTAL), a Knowledge Distillation-based method for deepfake detection using representation learning (ReL) to transfer representations between the source (teacher) and target (student) domains.

The main contributions of our work are summarized as follows:

1. We propose a novel domain adaption framework, Feature Representation Transfer Adaptation Learning (FReTAL), based on knowledge distillation and representation learning that can prevent catastrophic forgetting without accessing the source domain data.

2. We show that leveraging knowledge distillation and representation learning can enhance adaptability across different deepfake domains.

3. We empirically demonstrate that our method outperforms baseline approaches on deepfake benchmark datasets with up to 86.97% accuracy on low-quality deepfakes.

Our code is available here[1]. The rest of this paper is organized as follows. We discuss related work of deepfake detection, KD, and ReL in Section 2. We explain our FReTAL in Section 3 and describe our experimental settings in Section 4. Section 5 presents the results, and Section 6 provides a discussion and limitations of our work. Finally, we offer our conclusions in Section 7.

2. Background and Related Work

This work spans different fields, such as deepfake detection, domain adaptation, knowledge distillation, and representation learning. In this section, we will briefly cover the background and related research in aforementioned areas.

2.1. Deepfakes

While several sophisticated algorithms for creating realistic synthetic face videos have been developed in the past [9, 17, 72, 80, 44, 73, 37, 42, 10, 39, 79], most of these studies have not been widely available as open-source software applications so that the public can use. On the other hand, a much more straightforward approach focused on neural image style transfer [54, 24] has emerged as the preferred method for creating deepfake videos on a large scale. Now, many open-source implementations are now publicly available in the form of FakeApp [18], DeepFaceLab [64], FaceApp [25], and many others [23, 45]. Even though the

---
[1] https://github.com/alsgkals2/FReTAL

core idea is the same, each method has a slightly different implementation, resulting in different types of deepfakes. And, these methods are continuously improving over the years.

Deepfake Detection. As deepfakes have become a worldwide phenomenon, there has been a surge of interest in deepfake detection methods. The majority of current deepfake detection methods [78, 67, 66, 1, 74, 49] rely on deep neural networks (DNNs). These methods include splice detection [88, 89, 8, 7, 30, 70], abnormal eye blinking [50], signal level artifacts [55, 51], irregular head poses [84], peculiar behavior patterns [3, 2], and many other data-driven methods that do not rely on particular traces or artifacts in the deepfake videos [75, 76, 77, 47, 46, 41, 31, 32, 33]. However, to the best of our knowledge, except for the work of Cozzolino et al. [15] and Tariq et al. [78], not much research is conducted to apply domain adaptation on deepfake detection tasks. Furthermore, Tariq et al. [78] used high-quality deepfakes, and Cozzolino et al. [15] used medium-level compression (c23) on deepfake videos for domain adaptation. Our work is different from these works in that we use high-level compression (low-quality) because such deepfake videos are most common on social media.

2.2. Representation Learning

Representation learning (ReL) is the process of learning representations of input data, usually by transforming or extracting features from it, making a task like classification or prediction easier to perform. For feedforward networks, ReL is simply representing the hidden layers by applying some conditions to the learned intermediate features [26].

Transfer Learning and Domain Adaptation. Transfer learning and domain adaptation apply to the situations in which the information learned in one context (for example, distribution P_1) is used to enhance generalization in another setting (say, distribution P_2). In domain adaptation (DA), a subcategory of transfer learning, we apply an algorithm trained on the source domain to a different but related target domain. The source and target domains have the same feature space but different distributions in DA. In comparison, transfer learning encompasses cases where the target domain's feature space is different from the source feature space [26].

As deepfake video generation techniques are continuously evolving, more types of deepfake videos will emerge in the future. Collecting and producing a large number of new deepfake samples for each dataset would be impractical. In this work, we use feature-based domain adaptation to detect deepfakes generated using various methods. It also reduces the time cost. In order to perform domain adaptation, the model is initialized with the pre-trained weights on the source dataset. That model is then used to learn a new target dataset. Furthermore, if the source and the target

domains are similar, we can improve the performance over existing models. However, if they are not then, it can lead to catastrophic forgetting [40, 43]. Catastrophic forgetting is the tendency of a DNN to entirely and abruptly forget previously learned information upon learning new information. We solve this problem by using knowledge distillation.

Domain Adaptation using Knowledge Distillation. Hinton et al. [29] propose Knowledge distillation (KD). It is a method to compress knowledge of a large model to a small model. The main idea is that the student model can mimic the knowledge of the teacher model. Inspired by mimicking the teacher model, Li et al. [53] propose "Learning without forgetting". It is a method to maintain the source domain's knowledge by applying the knowledge distillation loss while transferring knowledge to the target domain. By adopting the principle of *rehearsal* [11], Rebuffi et al. [65] propose to stores the information of the source domain (i.e., storing exemplars) to overcome catastrophic forgetting in class-incremental learning using KD loss. However, it requires a large amount of memory storage to store the features of the source domain. This may lead to privacy breaches, such as inversion attacks. To prevent this, we propose a Representation Learning-based method that does not need to store or use source data in the model while transfer learning.

3. FReTAL

In this section, we provide details about our Feature Representation Transfer Adaptation Learning (FReTAL) method, including our motivation, and the pre-processing details.

Motivation. Catastrophic forgetting is a big hurdle during domain adaptation tasks [81, 83]. To overcome catastrophic forgetting, Tariq et al. [78] use few data samples from the source domain during transfer learning. However, in practice, for most pre-trained models, either the source domain data is not available or retaining source domain data may raise privacy concerns. Therefore, to encourage maximum applicability in real-world scenarios, we only use the target domain's data and apply knowledge distillation to learn from the pre-trained model (Teacher).

Pre-processing. First of all, we extract the frames (x) from real and deepfake videos using a custom code written on top of the FFmpeg library. Then, we use the MTCNN library [87] for face landmark detection. The faces are cropped and aligned to the center. We set $128 \times 128 \times 3$ to be the resolution of x. *Note: Instead of stretching x to match the square aspect ratio (1:1), we crop a bounding box of 128×128 from x with the face at the center of the frame. This way, we can avoid stretching the face on the horizontal axis and keep the face in more natural ratio (see Fig. 1).*

Teacher. The first step of FReTAL is to train a base model. We will refer to this base model as the pre-trained
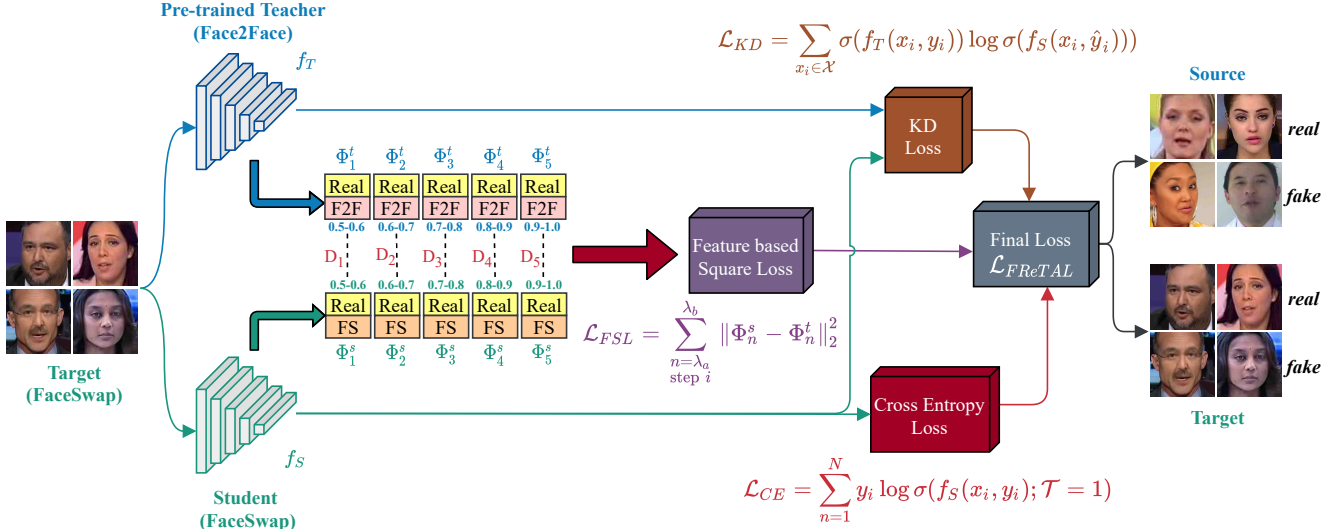
Figure 2. **The architecture of our Feature Representation Transfer Adaptation Learning (FReTAL).** The teacher model is trained with the Xception. Before transfer learning, we set the teacher model as untrainable. Then, we initialize the student model with the weights of the teacher model. Target domain data is provided to both teacher and student models to calculate the features for feature storage. We set the teacher as untrainable so that these features are fixed throughout the whole process. Whereas for the student, they will change in each iteration as training progress. We calculate KD loss between teacher and student models and a separate cross-entropy loss function just for the student model. Here, D1-D5 represents the square distance between feature storage of teacher and student. *Note: for the first iteration of transfer learning, the teacher and student model will be the same.*

or teacher model (f_T). For the deepfake detection task, this teacher model is trained on the source domain as a binary classifier to distinguish between real and deepfake images (e.g., Pristine and Face2Face). We set the teacher as untrainable for the whole domain adaptation process, and only the student model is trained from the step onwards.

Student. Once the teacher model is fully trained on the source domain, we copy the weights of teacher model (f_T) to the student model (f_S). The student model is then trained on the target domain using KD loss and feature-based representation learning. As illustrated in Fig. 2, this process is different from the usual teacher-student model. We provide the details of KD loss and feature-based representation learning in the following sections.

Feature-based Representation Learning. We assume that similar features must exist between different types of deepfakes. Therefore, a model trained on the source domain (Teacher) can help the student learn the target domain with fewer data samples. Before training on the target domain, the student model is just a copy of the teacher model. Then, we provide both the teacher (untrainable) and student model with the target domain's data to obtain its feature representation (Φ_n^t for teacher and Φ_n^s for student), as shown in Figure 2. Instead of storing features of all of the target domain's data, we only store distinguishable features. To do so, we apply softmax to both models' output. The softmax function takes as input a vector v of K real numbers, and normalizes it into a probability distribution consisting of K

probabilities proportional to the exponentials of the input numbers (i.e., between 0 and 1). Using this output, we create a feature storage from λ_a to λ_b in i unit intervals, as shown in Figure 2. It helps to minimize the domain shifting in the learning process by segmenting the features. As the distribution between real and fake data is different, we store the features of real and fake data separately. We calculate the difference between Φ^s and Φ^t using our feature-based square loss $\mathcal{L}_{FSL}$, as follows:

$$\mathcal{L}_{FSL} = \sum_{\substack{n=\lambda_a \\ \text{step } i}}^{\lambda_b} \left\| \Phi_n^s - \Phi_n^t \right\|_2^2 \qquad (1)$$

Domain Adaptation with Knowledge Distillation. To reduce the impact of catastrophic forgetting and domain shift, we apply cross-entropy loss and KD loss proposed by Hinton et al. [29] while training the student model on the target domain. Class probabilities are usually generated by neural networks using a softmax output layer that transforms the logit, x_i, computed for each class into a probability, $\sigma(x)_i$, by comparing x_i with the other logits x_j, as follows:

$$\sigma(x)_i = \frac{exp(\frac{x_i}{\mathcal{T}})}{\sum_{j=1}^{N} exp(\frac{x_j}{\mathcal{T}}))}, \qquad (2)$$

where $\mathcal{T}$ is the temperature that helps the student model mimic the teacher model by softening the probability dis-

4

tribution over the classes. The softmax function's probability distribution becomes softer by increasing $\mathcal{T}$, revealing which classes the teacher considered to be more similar to the predicted class. In general, KD loss is commonly expressed as minimizing the objective function:

$$\sum_{x_i \in \mathcal{X}} \mathcal{L}(f_T(x_i), f_S(x_i)), \qquad (3)$$

where x_i is the input, f_T is the teacher, f_S is the student, and $\mathcal{L}$ is a loss function that penalizes the difference between teacher and the student. In this work, we use cross-entropy for the loss function $\mathcal{L}$. Therefore, from Eq. (2) and (3), we can express our KD loss $\mathcal{L}_{KD}$, as follows:

$$\mathcal{L}_{KD} = \sum_{x_i \in \mathcal{X}} \sigma(f_T(x_i, y_i)) \log \sigma(f_S(x_i, \hat{y}_i))), \quad (4)$$

where σ is the softmax with temperature, y_i is the output label, and $\hat{y}_i$ is the output of the teacher f_T. In addition to KD loss, we also use cross-entropy loss in our student model f_S given as:

$$\mathcal{L}_{CE} = \sum_{n=1}^{N} y_i \log \sigma(f_S(x_i, y_i); \mathcal{T} = 1) \qquad (5)$$

Therefore, the loss function of our Feature Representation Transfer Adaptation Learning method can be written using Eq. (1), (4), and (5), as follows:

$$\mathcal{L}_{FReTAL} = \rho_1 \mathcal{L}_{FSL} + \rho_2 \mathcal{L}_{KD} + \rho_3 \mathcal{L}_{CE}, \qquad (6)$$

where ρ_1, ρ_2, and ρ_3 are scaling factors to control the three loss terms.

4. Experiment

We compared FReTAL with several transfer learning methods. In this section, we will describe the implementation details of FReTAL, as well as training and testing details of all detection models.

4.1. Dataset Description

To compare our method with several baselines, we used DeepFake (DF), Face2Face (FS), FaceSwap (FS), and NeuralTextures(NT) datasets from FaceForensics++ [66]. The pristine videos from [66] are used as real videos. In Table 1, we describe all the datasets used for base training (Teacher) and transfer learning (Student). We used the 750 videos for training the teacher model and only ten videos for training the student model during domain adaptation (or transfer learning for brevity). The remaining 125 videos are used for validation and 125 for testing. In contrast to Tariq et al. [78], we do not use the source domain dataset during transfer learning.

Table 1. The details of datasets used for training and testing.

Datasets	Total Videos	Training Videos	Transfer Learning	Testing Videos
Pristine (Real)	1,000	750	10	250
DeepFake (DF)	1,000	750	10	250
FaceSwap (FS)	1,000	750	10	250
Face2Face (F2F)	1,000	750	10	250
Neural Textures (NT)	1,000	750	10	250

4.2. Baselines

We explored several baselines for comparison. The following is a brief detail on them.

1. **_Güera et al._** [28]: deployed a stack of CNN on top of an LSTM network to detect deepfake. The CNN module outputs the feature vector fed to the LSTM module that generates the sequence descriptors and passes them to a fully connected layer with softmax to generate probabilities.

2. **_Sabir et al._** [68]: used DenseNet with a bidirectional RNN to achieve high accuracy on DeepFake, FaceSwap, and Face2Face datasets.

3. **_ShallowNet_**: Tariq et al. [76] demonstrated that ShallowNet [75] detects GAN-generated images with high accuracy. We developed ShallowNet using Python and TensorFlow.

4. **_Xception_** [13]: is considered as the state-of-the-art deep learning model for image classification task. Also, Rössler et al. [67] demonstrated that Xception achieves the best accuracy on FaceForensics++ dataset. We used the PyTorch implementation of Xception.

The code for CNN+LSTM and DBiRNN are not publicly available; therefore, we implemented them and tried our best to match the original paper's experimental settings.

4.3. Domain Adaptation Methods

In addition to the baselines experiments, we explored several domain adaptation methods as follows:

1. **_FT_**: We apply general transfer learning (fine-tuning) on aforementioned baseline methods without layer freezing.

2. **_T-GD_**: Jeon et al. [31] propose T-GD that can achieve high performance and prevent the catastrophic forgetting by combining with L2-SP and self-training. We use T-GD to perform transfer learning with Xception model.

3. **_KD_**: We only use KD loss $\mathcal{L}_{KD}$ component from our $\mathcal{L}_{FReTAL}$ loss function to perform domain adaptation on Xception.

Table 2. **Teacher model performance on source dataset (HQ).** Xception performs the best among the baselines. All the results are in percentages (%) and best are highlighted in bold.

Method	DF	FS	F2F	NT	Avg.
Güera et al. [28]	78.51	77.75	71.87	90.54	77.80
Sabir et al. [68]	80.54	80.56	73.12	94.38	82.21
ShallowNet	88.97	93.33	75.26	99.45	87.08
Xception	**99.00**	**99.29**	**99.26**	**99.46**	**99.25**

Table 3. **Teacher model performance on source dataset (LQ) and zero-shot performance.** We are only presenting the results of Xception model on low quality as it is the best performer on HQ dataset. The source dataset results (diagonal) are highlighted in bold.

Method	DF (%)	F2F (%)	FS (%)	NT (%)
Xception (DF)	**99.41**	56.05	49.93	66.32
Xception (F2F)	68.55	**98.64**	50.55	54.81
Xception (FS)	49.89	54.15	**98.36**	50.74
Xception (NT)	50.05	57.49	50.01	**99.88**

4.4. Implementation Details of FReTAL

Due to the consistent performance of Xception in many face classification and deepfake detection tasks [67, 76, 75, 47, 31], we select Xception as the backbone model for our FReTAL method. We use the PyTorch implementation of Xception, pre-trained on the ImageNet dataset. We set the value of hyper-parameter values as follows: $\lambda_a = 0.5$, $\lambda_b = 1.0$, $i = 0.1$, $\mathcal{T} = 20$, $\rho_1 = 1.0$, $\rho_2 = 1.0$, and $\rho_3 = 1.0$. Therefore, the range for feature storage is $\{(0.5 - 0.6), (0.6 - 0.7), \ldots, (0.9 - 1.0)\}$. For training, we used the stochastic gradient descent (SGD) with a learning rate of 0.05 with a momentum of 0.1, and the number of iterations is set to 100. We applied early stopping with a patience of 5.

Machine Configuration. We run our experiments using P100 and TITAN RTX GPUs, with 24 GB of dedicated memory. We use Intel Xeon Gold 6230 CPUs with 8 cores each and 256 GB of RAM. The under-lying OS is Ubuntu 18.04.2 LTS 64 bit. We use PyTorch v1.7.0 with CUDA 11.0 and Python 3.8.

Evaluation Metrics. We use F_1-score metric to evaluate the model performance using 125 real and 125 deepfake test videos.

Preprocessing and Data Augmentation. We extract 16 samples such that each sample of origin and manipulated video contain five consecutive frames (16×5 = 80 images per video). To extract the face landmark information from extracted frame, we use multi-task CNN (MTCNN) [87]. We apply the following normalization settings using Py-Torch Transform: [0.5,0.5,0.5]. We use CutMix [85] for data augmentation.

4.5. Configuring Training Models

Teacher Model Training. We use any source domain to train the teacher model f_T using 750 real (pristine) and 750 deepfake (e.g., Face2Face) videos. After this process, we set the f_T as untrainable.

Student Model Training. We initialize the student model f_S by copying the weight from the teacher model. Then, we train f_S on any target domain (e.g., FaceSwap). We do not use any source domain data (e.g., Face2Face) when transfer learning to target domain. To compare general transfer learning, we also train T-GD and KD using the same settings. We perform single source to single target transfer learning using several configuration, as shown in Table 4 and 5.

5. Results

In this section, we present the results for base training (Teacher) and Transfer learning (Student) on both high- and low-quality datasets.

5.1. Performance of Teacher

We evaluate the teacher model f_T using four baseline methods on the high-quality deepfake dataset. As shown in Table 2, we find that Xception is the best performer among all baselines across all datasets. This result is consistent with [77, 67]. Therefore, based on this result, we selected Xception as the best candidate to perform further experiments. This time we train Xception on low-quality deepfake datasets and additionally check zero-shot performance. As shown in Table 3, the model does not perform well except for the source domain. These results are also consistent with the high-quality zero-shot performance result of [78]. This result shows that deepfake detectors such as Xception only perform well against the type of deepfakes on which they are trained. Therefore, there is a need for a domain adaptation-based method that can perform well against all kinds of deepfakes.

5.2. Performance of Student using FReTAL

Following the same settings is the previous experiment. In this experiment, first, the teacher model is trained on the source domain (HQ), and then we fine-tune (FT) the student using transfer learning to learn the target domain (HQ). Furthermore, we apply T-GD, KD, and our FReTAL on Xception. As shown in Table 4, Xception with our FReTAL method performs the best in most scenarios except for F2F→DF and FS→DF, where Xception + *KD* demonstrates better performance. As Xception + (domain adaptation method) provides the best performance on high-quality deepfakes, we use it for further experiments with low-quality deepfakes. Now instead of high quality, we use low-quality images for both teacher and student mod-

Table 4. **Student model performance on target dataset (HQ).** We evaluate all datasets with four baselines using four domain adaptation methods. The top-row indicates the "Source →Target" dataset. Xception + FReTAL demonstrated the best and most consistent performance. The best results are highlighted in bold. Note: Due to space limitation, we show only a selected Source →Target configurations.

Method	Domain	DF→F2F (%)	DF→FS (%)	F2F→DF (%)	F2F→FS (%)	FS→DF (%)	FS→F2F (%)
Güera et al. [28] + *FT*	Source	70.21	72.35	71.87	72.41	70.32	73.15
	Target	50.73	52.75	52.75	63.34	50.73	66.08
	Avg.	60.47	62.55	62.31	67.88	60.53	69.62
Sabir et al. [68] + *FT*	Source	73.56	75.36	76.45	73.83	75.84	76.32
	Target	59.81	55.62	55.25	51.39	50.45	55.17
	Avg.	66.69	65.49	65.85	62.61	63.15	65.75
ShallowNet + *FT*	Source	75.26	75.85	74.84	77.85	73.31	75.19
	Target	55.86	50.92	58.84	42.38	53.83	50.29
	Avg.	65.56	63.39	66.84	60.12	63.57	62.74
Xception + *FT*	Source	93.65	70.00	95.10	90.35	93.77	94.91
	Target	84.59	55.18	91.32	55.26	86.56	83.11
	Avg.	89.12	62.59	93.21	72.81	90.17	89.01
Xception + *T-GD*	Source	92.96	73.92	96.89	90.42	92.55	94.85
	Target	77.89	55.64	84.55	55.60	79.38	78.49
	Avg.	85.43	64.78	90.72	73.01	85.97	86.67
Xception + *KD*	Source	95.58	82.77	96.91	84.57	95.65	96.28
	Target	84.31	59.55	92.51	76.45	87.05	85.12
	Avg.	89.95	71.16	**94.72**	80.51	**91.35**	90.70
Xception + **FReTAL**	Source	95.68	88.60	98.09	93.36	92.57	96.41
	Target	84.54	76.23	89.90	80.63	86.45	88.64
	Avg.	**90.11**	**82.42**	94.00	**82.00**	89.51	**92.53**

Table 5. **Student model performance on target dataset (LQ).** We evaluate all datasets with Xception using four domain adaptation methods. The top-row indicates the "Source →Target" dataset. Xception + FReTAL demonstrated the best performance for all cases. The best results are highlighted in bold. Note: Due to space limitation, we show only a selected Source →Target configurations.

Method	Domain	FS→F2F (%)	F2F→FS (%)	FS→DF (%)	DF→F2F (%)	F2F→NT (%)	DF→NT (%)
Xception + FT	Source	40.93	84.78	80.56	89.84	87.12	88.29
	Target	60.30	52.97	64.61	58.24	76.78	82.40
	Avg.	50.62	75.05	72.59	74.04	81.95	85.35
Xception + T-GD	Source	36.08	84.70	85.98	88.07	83.22	81.23
	Target	56.95	52.95	55.9	49.55	52.69	67.11
	Avg.	46.52	68.83	70.94	68.81	67.96	74.17
Xception + KD	Source	48.07	84.84	80.48	82.59	86.07	89.61
	Target	61.40	65.26	64.63	64.34	74.56	81.03
	Avg.	54.74	75.05	72.56	73.47	80.32	85.32
Xception + **FReTAL**	Source	81.78	82.03	85.93	91.20	82.85	90.56
	Target	64.45	68.79	65.78	62.09	83.87	83.38
	Avg.	**73.12**	**75.41**	**75.86**	**76.65**	**83.36**	**86.97**

els. As shown in Table 5, Xception + FReTAL performs the best on all source to target configurations across all datasets. This result demonstrates that our FReTAL is a better domain adaptation method for deepfake detection than the other baselines. The low performance of fine-tuning in some scenarios, such as FS→F2F in Table 7, is due to catastrophic forgetting. In contrast, our FReTAL method shows robustness against catastrophic forgetting.

5.3. Ablation Study: Feature Representation

We perform an ablation study by removing the feature-based representation learning component $\mathcal{L}_{FSL}$ and the stu-

dent's cross-entropy loss $\mathcal{L}_{CE}$ from our FReTAL method. Without these components, our method becomes similar to the KD. And as shown in Table 4 and 5, Xception + KD performs worse than Xception + FReTAL in most scenarios, which shows that these components are necessary to achieve better performance.

6. Discussion

Evaluation of DFDC and CelebDF. Recently, more sophisticated deepfake datasets such as DFDC [21] and CelebDF [52] have been released. We plan to include these

datasets in our experiment in the future. However, it is important to note that if a deepfake detector fails to perform well on low-quality images of FaceForensics++ [67] dataset, they might also fail on more complex datasets such as DFDC and CelebDF.

Performance on Low-quality Deepfakes. The performance on low-quality deepfakes, especially for the transfer learning task, is relatively lower ($< 90\%$) than high-quality deepfakes. It means that there is still a lot of room for improvement in domain adaptation for low-quality deepfake detection. We believe that applying the super-resolution method as a data augmentation method on low-quality deepfakes may reduce this gap.

Mixed LQ and HQ Deepfake Detection. As we know that, models trained on high-quality deepfakes do not perform well on low-quality deepfakes. However, it is interesting to note that the model trained on low-quality deepfakes does not perform well on high-quality deepfakes as well unless we apply the same compression on the high-quality deepfakes to convert them into low-quality. Therefore, programmatically identifying the quality of deepfake is another venue of research. We also want to focus on detecting mixed quality deepfake datasets like DFDC.

Limitations and Future Work. Detecting talking head types of deepfakes [86] are not explored in this work. Also, recently, full-body gesture-based deepfakes have emerged [71]. It would be interesting to see how FReTAL can be generalized against talking head and full-body deepfakes. For data collection, it is becoming challenging to distinguish deepfake videos visually. Furthermore, it is not easy to use or download these deepfake videos to train a deepfake detector due to privacy and copyright issues. Therefore, using a minimum amount of freely available data to achieve high performance is preferable in such scenarios. To solve these problems, we will explore an augmentation method for few-shot learning to improve practicality and performance with very few videos or images. Furthermore, we will utilize our feature-based representation learning framework to improving the domain adaptability and generalization capabilities of other deepfake detectors. Future work also includes exploring alternative training strategies that can help improve performance and multi-domain adaptation.

7. Conclusion

Performing domain adaptation for detecting deepfakes is becoming more challenging for low-quality images than high-quality ones. We find that similar features exist between the source and target dataset that can help in domain adaptation. Therefore, we propose a domain adaptation method using feature storage and KD loss in a teacher-student network, where the teacher is not trained on the target domain. Moreover, we demonstrate that applying KD loss without even using the source dataset can reduce catastrophic forgetting, i.e., domain shifting in deepfake detection tasks. We show that by using FReTAL, we can quickly adapt to new types of deepfakes with a reasonable performance using as low as ten samples of the target domain. For future work, we plan to explore more augmentation methods on the target domain data to improve practicality and performance. We will also utilize our FReTAL framework to improve domain adaptability and generalization capabilities of other deepfake detection models such as CLRNet and Mesonet.

References

[1] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Mesonet: a compact facial video forgery detection network. In *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–7. IEEE, 2018. 2, 3

[2] Shruti Agarwal, Hany Farid, Tarek El-Gaaly, and Ser-Nam Lim. Detecting deep-fake videos from appearance and behavior. In *2020 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, 2020. 3

[3] Shruti Agarwal, Hany Farid, Yuming Gu, Mingming He, Koki Nagano, and Hao Li. Protecting world leaders against deep fakes. In *CVPR Workshops*, pages 38–45, 2019. 3

[4] Saifuddin Ahmed. Who inadvertently shares deepfakes? analyzing the role of political interest, cognitive ability, and social network size. *Telematics and Informatics*, 57:101508, 2021. 1

[5] Jourdan Aldredge. Is Deepfake Technology the Future of the Film Industry? https://www.premiumbeat.com/blog/deepfake-technology-future-of-film-industry, June 2020. [Online; accessed 25-March-2021]. 1

[6] Taichi Asami, Ryo Masumura, Yoshikazu Yamaguchi, Hirokazu Masataki, and Yushi Aono. Domain adaptation of dnn acoustic models using knowledge distillation. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5185–5189. IEEE, 2017. 2

[7] Jawadul H Bappy, Amit K Roy-Chowdhury, Jason Bunk, Lakshmanan Nataraj, and BS Manjunath. Exploiting spatial structure for localizing manipulated image regions. In *Proceedings of the IEEE international conference on computer vision*, pages 4970–4979, 2017. 3

[8] Jawadul H Bappy, Cody Simons, Lakshmanan Nataraj, BS Manjunath, and Amit K Roy-Chowdhury. Hybrid lstm and encoder–decoder architecture for detection of image forgeries. *IEEE Transactions on Image Processing*, 28(7):3286–3300, 2019. 3

[9] Dmitri Bitouk, Neeraj Kumar, Samreen Dhillon, Peter Belhumeur, and Shree K Nayar. Face swapping: automatically replacing faces in photographs. In *ACM SIGGRAPH 2008 papers*, pages 1–8. 2008. 2

[10] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5933–5942, 2019. 2

[11] Xinyang Chen, Sinan Wang, Bo Fu, Mingsheng Long, and Jianmin Wang. Catastrophic forgetting meets negative transfer: Batch spectral shrinkage for safe transfer learning. 2019. 3

[12] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8789–8797, 2018. 1

[13] 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. 5

[14] Aidan Clark, Jeff Donahue, and Karen Simonyan. Efficient video generation on complex datasets. 2019. 1

[15] 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. 3

[16] Adrian Croft. From porn to scams, deepfakes are becoming a big racket-and that's unnerving business leaders and lawmakers. https://fortune.com/2019/10/07/porn-to-scams-deepfakes-big-racket-unnerving-business-leaders-and-lawmakers, 2019. [Online; accessed 25-March-2021]. 1

[17] Kevin Dale, Kalyan Sunkavalli, Micah K Johnson, Daniel Vlasic, Wojciech Matusik, and Hanspeter Pfister. Video face replacement. In *Proceedings of the 2011 SIGGRAPH Asia conference*, pages 1–10, 2011. 2

[18] Deepfakes Reddit. FakeApp. https://www.malavida.com/en/soft/fakeapp/, January 2018. [Online; accessed 24-March-2021]. 2

[19] Jeffery DelViscio. A Nixon Deepfake, a 'Moon Disaster' Speech and an Information Ecosystem at Risk - Scientific American, 2020. [Online; accessed 25-March-2021]. 2

[20] EJ Dickson. Deepfake porn is still a threat, particularly for k-pop stars. https://www.rollingstone.com/culture/culture-news/deepfakes-nonconsensual-porn-study-kpop-895605, 2019. [Online; accessed 25-March-2021]. 1

[21] Brian Dolhansky, Russ Howes, Ben Pflaum, Nicole Baram, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) preview dataset. *arXiv preprint arXiv:1910.08854*, 2019. 2, 7

[22] Charlotte Edwards. Making deepfake porn could soon be as easy as using instagram filters, according to expert. https://www.thesun.co.uk/tech/9800017/deepfake-porn-soon-easy, 2019. [Online; accessed 25-March-2021]. 1

[23] FaceSwapDevs. Deepfakes_faceswap - github repository. https://github.com/deepfakes/faceswap, 2019. [Online; accessed 24-March-2021]. 1, 2

[24] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 2

[25] Yaroslav Goncharov. FaceApp - Most Popular Selfie Editor. www.faceapp.com, 2019. [Online; accessed 24-March-2021]. 1, 2

[26] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org. 1, 3

[27] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014. 1

[28] David Güera and Edward J Delp. Deepfake video detection using recurrent neural networks. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2018. 5, 6, 7

[29] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3, 4

[30] Minyoung Huh, Andrew Liu, Andrew Owens, and Alexei A Efros. Fighting fake news: Image splice detection via learned self-consistency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 101–117, 2018. 3

[31] Hyeonseong Jeon, Youngoh Bang, Junyaup Kim, and Simon S Woo. T-gd: Transferable gan-generated images detection framework. *arXiv preprint arXiv:2008.04115*, 2020. 3, 5, 6

[32] 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. 3

[33] Hyeonseong Jeon, Youngoh Bang, and Simon S Woo. Fdftnet: Facing off fake images using fake detection fine-tuning network. *arXiv preprint arXiv:2001.01265*, 2020. 3

[34] Craig Jones. 1 in 3 who are aware of deepfakes say they have inadvertently shared them on social media. https://www.newswise.com/articles/1-in-3-who-are-aware-of-deepfakes-say-they-have-inadvertently-shared-them-on-social-media, November 2020. [Online; accessed 24-March-2021]. 1

[35] Emmanuel Kahembwe and Subramanian Ramamoorthy. Lower dimensional kernels for video discriminators. *Neural Networks*, 132:506–520, 2020. 1

[36] Michael Kan. Most ai-generated deepfake videos online are porn. https://www.pcmag.com/news/371193/most-ai-generated-deepfake-videos-online-are-porn, 2019. [Online; accessed 25-March-2021]. 1

[37] 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. 1, 2

[38] 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. 1

[39] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 2

[40] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler L Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *Thirty-second AAAI conference on artificial intelligence*, 2018. 3

[41] Hasam Khalid and Simon S Woo. Oc-fakedect: Classifying deepfakes using one-class variational autoencoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 656–657, 2020. 3

[42] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Niessner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. Deep video portraits. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018. 2

[43] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 3

[44] Iryna Korshunova, Wenzhe Shi, Joni Dambre, and Lucas Theis. Fast face-swap using convolutional neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 3677–3685, 2017. 2

[45] Marek Kowalski. Faceswap - github repository. https://github.com/MarekKowalski/FaceSwap, 2016. [Online; accessed 24-March-2021]. 1, 2

[46] Sangyup Lee, Shahroz Tariq, Junyaup Kim, and Simon S. Woo. Tar: Generalized forensic framework to detectdeepfakes using weakly supervised learning. In *IFIP International Conference on ICT Systems Security and Privacy Protection*. Springer, 2021. 3

[47] Sangyup Lee, Shahroz Tariq, Youjin Shin, and Simon S. Woo. Detecting handcrafted facial image manipulations and gan-generated facial images using shallow-fakefacenet. *Applied Soft Computing*, 105:107256, 2021. 3, 6

[48] Jinyu Li, Michael L Seltzer, Xi Wang, Rui Zhao, and Yifan Gong. Large-scale domain adaptation via teacher-student learning. *arXiv preprint arXiv:1708.05466*, 2017. 2

[49] Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo. Face x-ray for more general face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5001–5010, 2020. 3

[50] 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. 3

[51] Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts. *arXiv preprint arXiv:1811.00656*, 2018. 3

[52] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-df: A large-scale challenging dataset for deepfake forensics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3207–3216, 2020. 2, 7

[53] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 3

[54] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. *arXiv preprint arXiv:1703.00848*, 2017. 2

[55] Falko Matern, Christian Riess, and Marc Stamminger. Exploiting visual artifacts to expose deepfakes and face manipulations. In *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 83–92. IEEE, 2019. 3

[56] Ivan Mehta. A new study says nearly 96 of deepfake videos are porn. https://thenextweb.com/apps/2019/10/07/a-new-study-says-nearly-96-of-deepfake-videos-are-porn, 2019. [Online; accessed 25-March-2021]. 2

[57] Kimberley Mok. Deep Learning AI Generates Realistic Game Graphics by Learning from Videos. https://thenewstack.io/deep-learning-ai-generates-realistic-game-graphics-by-learning-from-videos, January 2019. [Online; accessed 25-March-2021]. 1

[58] Le Thanh Nguyen-Meidine, Atif Belal, Madhu Kiran, Jose Dolz, Louis-Antoine Blais-Morin, and Eric Granger. Knowledge distillation methods for efficient unsupervised adaptation across multiple domains. *arXiv preprint arXiv:2101.07308*, 2021. 2

[59] Le Thanh Nguyen-Meidine, Eric Granger, Madhu Kiran, Jose Dolz, and Louis-Antoine Blais-Morin. Joint progressive knowledge distillation and unsupervised domain adaptation. *arXiv preprint arXiv:2005.07839*, 2020. 2

[60] Mauricio Orbes-Arteainst, Jorge Cardoso, Lauge Sørensen, Christian Igel, Sebastien Ourselin, Marc Modat, Mads Nielsen, and Akshay Pai. Knowledge distillation for semi-supervised domain adaptation. In *OR 2.0 Context-Aware*

Operating Theaters and Machine Learning in Clinical Neuroimaging, pages 68–76. Springer, 2019. 2

[61] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Gaugan: semantic image synthesis with spatially adaptive normalization. In *ACM SIGGRAPH 2019 Real-Time Live!*, pages 1–1. 2019. 1

[62] Walter Pasquarelli. Towards Synthetic Reality: When DeepFakes meet AR/VR. https://www.oxfordinsights.com/insights/2019/8/6/towards-synthetic-reality-when-deepfakes-meet-arvr, August 2019. [Online; accessed 25-March-2021]. 1

[63] Dan Patterson. President's words used to create "deepfakes" at Davos. Video, 2020. [Online; accessed 25-March-2021]. 2

[64] Ivan Petrov, Daiheng Gao, Nikolay Chervoniy, Kunlin Liu, Sugasa Marangonda, Chris Umé, Jian Jiang, Luis RP, Sheng Zhang, Pingyu Wu, et al. Deepfacelab: A simple, flexible and extensible face swapping framework. *arXiv preprint arXiv:2005.05535*, 2020. 2

[65] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. 3

[66] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics: A large-scale video dataset for forgery detection in human faces. *arXiv preprint arXiv:1803.09179*, 2018. 3, 5

[67] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *ICCV 2019*, 2019. 2, 3, 5, 6, 8

[68] Ekraam Sabir, Jiaxin Cheng, Ayush Jaiswal, Wael AbdAlmageed, Iacopo Masi, and Prem Natarajan. Recurrent convolutional strategies for face manipulation detection in videos. *Interfaces (GUI)*, 3:1, 2019. 5, 6, 7

[69] Masaki Saito, Shunta Saito, Masanori Koyama, and Sosuke Kobayashi. Train sparsely, generate densely: Memory-efficient unsupervised training of high-resolution temporal gan. *International Journal of Computer Vision*, 128:2586–2606, 2020. 1

[70] Ronald Salloum, Yuzhuo Ren, and C-C Jay Kuo. Image splicing localization using a multi-task fully convolutional network (mfcn). *Journal of Visual Communication and Image Representation*, 51:201–209, 2018. 3

[71] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *Advances in Neural Information Processing Systems*, pages 7135–7145, 2019. 8

[72] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. What makes tom hanks look like tom hanks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3952–3960, 2015. 2

[73] 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. 2

[74] Shahroz Tariq, Sowon Jeon, and Simon S Woo. Am i a real or fake celebrity? measuring commercial face recognition web apis under deepfake impersonation attack. *arXiv preprint arXiv:2103.00847*, 2021. 2, 3

[75] Shahroz Tariq, Sangyup Lee, Hoyoung Kim, Youjin Shin, and Simon S Woo. Detecting both machine and human created fake face images in the wild. In *Proceedings of the 2nd International Workshop on Multimedia Privacy and Security*, pages 81–87. ACM, 2018. 2, 3, 5, 6

[76] Shahroz Tariq, Sangyup Lee, Hoyoung Kim, Youjin Shin, and Simon S Woo. 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*, pages 1296–1303. ACM, 2019. 3, 5, 6

[77] Shahroz Tariq, Sangyup Lee, and Simon S Woo. A convolutional lstm based residual network for deepfake video detection. *arXiv preprint arXiv:2009.07480*, 2020. 3, 6

[78] Shahroz Tariq, Sangyup Lee, and Simon S Woo. One detector to rule them all: Towards a general deepfake attack detection framework. In *Proceedings of The Web Conference 2021*, 2021. 2, 3, 5, 6

[79] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019. 2

[80] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. *Commun. ACM*, 62(1):96–104, Dec. 2018. 2

[81] Brian Thompson, Jeremy Gwinnup, Huda Khayrallah, Kevin Duh, and Philipp Koehn. Overcoming catastrophic forgetting during domain adaptation of neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2062–2068, 2019. 3

[82] James Vincent. Watch Jordan Peele use AI to make Barack Obama deliver a PSA about fake news, 2018. [Online; accessed 25-March-2021]. 2

[83] Ying Xu, Xu Zhong, Antonio Jose Jimeno Yepes, and Jey Han Lau. Forget me not: Reducing catastrophic forgetting for domain adaptation in reading comprehension. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020. 3

[84] Xin Yang, Yuezun Li, and Siwei Lyu. Exposing deep fakes using inconsistent head poses. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8261–8265. IEEE, 2019. 3

[85] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019. 6

[86] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. Few-shot adversarial learning of realistic neural talking head models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9459–9468, 2019. 8

[87] 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. 3, 6

[88] 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. 3

[89] Peng Zhou, Xintong Han, Vlad I Morariu, and Larry S Davis. Learning rich features for image manipulation detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1053–1061, 2018. 3