Semi-supervised Deep Domain Adaptation for Deepfake Detection

Md Shamim Seraj Ankita Singh Shayok Chakraborty
Department of Computer Science, Florida State University

Abstract

With the advent and popularity of generative models such as GANs, synthetic image generation and manipulation has become commonplace. This has promoted active research in the development of effective deepfake detection technology. While existing detection techniques have demonstrated promise, their performance suffers when tested on data generated using a different faking technology, on which the model has not been sufficiently trained. This challenge of detecting new types of deepfakes, without losing its prior knowledge about deepfakes (catastrophic forgetting), is of utmost importance in today's world. In this paper, we propose a novel deep domain adaptation framework to address this important problem in deepfake detection research. Our framework can leverage a large amount of labeled data (fake / genuine) generated using a particular faking technique (source domain) and a small amount of labeled data generated using a different faking technique (target domain) to induce a deep neural network with good generalization capability on both the source and the target domains. Further, deep neural networks are data-hungry and require a large amount of labeled training data, which may not always be available in the context of deepfake detection; our framework can also efficiently utilize unlabeled data in the target domain, which is more readily available than labeled data. We design a novel loss function and use the stochastic gradient descent (SGD) method to optimize the loss and train the deep network. Our extensive empirical studies on the benchmark FaceForensics++ dataset, using three types of deepfakes, corroborate the promise and potential of our framework against competing baselines.

1. Introduction

Due to the unprecedented progress of generative AI techniques, synthetic multimedia has become extremely common in social media and the Internet [10,25]. Their popularity is mainly driven by easily accessible, sophisticated tools for artificially generating realistic multimedia data [3, 19]. Such a technology can be judiciously used in a variety of applications, such as photorealistic scenery generation [52], film making [11] and human face generation [26]. However, the strong capability of generative AI to produce realistic multimedia has also threatened the authenticity and integrity of digital images, and has allowed people to misuse it for malicious purposes [1, 4–6]. In particular, *deepfake*, which is arbitrarily defined as fake multimedia created by training generative neural network architectures such as autoencoders or generative adversarial networks (GANs), has emerged as one of the most popular multimedia tampering techniques [47] [1]. Among the most popular deepfake forgeries are human facial manipulations. For instance, *FaceSwap (FS)* is a type of deepfake that replaces a face in a target video sequence with a face from a different video or image collection; *Face2Face (F2F)* is another type of deepfake that transfers the expressions of a source video to a target video while preserving the target person's identity [55, 63]. These techniques can be easily used for creating child sexual abuse materials, celebrity pornographic videos and fake propaganda videos for gaining unlawful political influence [17, 53, 67].

With the line between real and fake media becoming increasingly blurred, deep fake detection has gained increasing popularity in the computer vision research community [46]. This is usually framed as a binary classification task of predicting whether a given image / video is genuine or fake. Existing detection techniques primarily use traditional hand-crafted features [29,34], biological features that exploit unique biometric information of the human face [38,69] and most popularly, hierarchical feature representations learned automatically using deep Convolutional Neural Networks (CNNs) [7, 9, 12, 13, 37, 56, 69, 74, 75].

While these methods have depicted promising performance, they suffer from poor generalization; that is, their performance is adversely affected when new types of manipulations are presented, even though they are semantically similar [62, 73]. The deep neural networks (DNNs) tend to overfit to the manipulation-specific artifacts and learn features that are informative for the given task, but cannot be transferred to detect forgeries generated using a different technology [16, 31]. To overcome this challenge, a large

[1] we use the terms *manipulation technique*, *faking technique*, *forgery* and *deepfake* interchangeably in this paper

amount of labeled training data from the new domain is necessary to fine-tune the DNN (as deep neural networks are data-hungry). Due to the rapid progress in the field of digital content creation, obtaining abundant labeled data for every single manipulation technique is not feasible. Ideally, we would like to detect a forgery even if only a few (or none) labeled samples, and some unlabeled samples are available from the new manipulation technique (since unlabeled data is more readily available than labeled data; for instance, we may have access to several images which may or may not be forged using the new faking technique, and that information is not available to us, that is, the labels of these images are unknown). Further, once the DNN is trained to detect the new types of deepfakes, it should still be able to furnish high accuracy in the original detection task, so as to mitigate the catastrophic forgetting (knowledge forgetting) problem [27, 33, 64, 68].

We thus pose the research task as follows: *we are given abundant data generated using a particular type of deepfake (source domain data). The data in the source domain are all labeled (genuine / fake). We are also given a small amount (or none) of labeled data and a moderate amount of unlabeled data from a different faking technology (target domain data). Our objective is to train a deep CNN to effectively identify fake and genuine images in both the source and target domains.*

In this paper, we propose a novel semi-supervised domain adaptation technique to address this challenging and practical problem. *Domain Adaptation (DA)* or *Transfer Learning (TL)* algorithms are instrumental in utilizing abundant labeled data in one domain to develop a model for a related domain of interest, where there is a paucity of labeled data [50]. The domain of interest is referred to as the *target* domain and the other domain is called the *source* domain. The probability distributions generating the data in the two domains are different, which implies that a deep model trained on the source domain data may not directly generalize to the target domain. We propose a novel loss function to train the deep CNN, and leverage adversarial DA techniques to address the disparity between the source and target domains. We validate our framework on challenging low resolution data and with varied number of labeled images from the target domain. Our framework depicts impressive performance even when the target domain contains only unlabeled samples, and no labeled data is available in the target domain.

The rest of the paper is organized as follows: we present a survey of related techniques on deepfake detection in Section 2; our proposed framework is detailed in Section 3; we present the results of our empirical studies in Section 4 and conclude with discussions in Section 5.

2. Related Work

Deepfake Detection: With the advent of several open-source implementations of deepfakes, such as FakeApp [2], DeepFaceLab [41], FaceApp [3] etc., deepfake detection has garnered sufficient research attention in the vision community. Most of the current detection techniques rely on deep neural networks (DNNs) [7, 37, 54, 55, 58, 62]. These methods include splice detection [12, 13, 21, 56, 74, 75], abnormal eye blinking [38], signal level artifacts [39, 45], irregular head poses [69], peculiar behavior patterns [8, 9], and many other data-driven methods that do not rely on particular artifacts in the deepfake videos [22–24, 28, 35, 36, 59–61]. As mentioned before, these methods suffer from poor generalization, when tested on deepfakes of a different type than those in the training data.

Domain Adaptation: Domain Adaptation (DA) or Transfer Learning is a well-researched problem in machine vision. Please refer to [50] for a comprehensive survey. DA techniques based on deep learning have outperformed their non-deep counterparts, which used hand-crafted features [49, 51]. The Maximum Mean Discrepancy (MMD) has been extensively used as a metric to quantify the disparity between the source and target domains and learn domain invariant features using a DNN [43, 44, 65, 66]. Techniques based on Generative Adversarial Networks (GANs) have depicted particularly commendable performance for DA. Algorithms in this category include the Domain Adversarial Neural Network (DANN) which incorporates a domain classifier, whose gradient is reversed when learning the feature extractor weights [20], the Coupled Generative Adversarial Network (CoGAN) model, which shares weights at different layers of the GAN to train a coupled network, and the combination of CoGAN with Variational Autoencoder (VAE) [32] to develop an image translation network [42] among others. Concepts from Wasserstein GAN have also been used for domain adaptation [57]. Recent research efforts in this area include Universal Domain Adaptation, which addresses the practical problem where the label set between the source and target domains may not be exactly identical [70], Source-free Domain Adaptation, where only a trained model (and no data) is available from the source domain, due to privacy concerns [71], and Active Domain Adaptation, which attempts to address the disparity between the source and target domains, and simultaneously identifies the exemplar unlabeled samples in the target domain for manual annotation [48].

Domain Adaptation for Deepfake Detection: Even though both DA and deepfake detection have been extensively studied, DA for deepfake detection is much less explored. Kim *et al.* [31] employed representation learning and knowledge distillation to perform domain adaptation on new types of deepfakes, while minimizing catastrophic forgetting. The same authors also combined the paradigms of continual learning, representation learning and knowledge distillation to perform DA on new deepfake datasets [30]. Tariq *et al*. [62] first trained a DNN on the source data; the first half of the model layers were then frozen, while the deeper layers were fine-tuned with the target domain data. Along similar lines, Cozzolino *et al*. [16] introduced the *ForensicTransfer* framework, which learned a forensic embedding on the sourse domain videos using an autoencoder, which was fine-tuned using a handful of training videos from the target domain. However, all these methods require all the data in the target domain to be labeled. In the context of deepfake detection, we may encounter a situation where we have access to a large number of images, but we are unable to verify whether they are forged using the new faking mechanism or not, that is, the labels of these images are not available to us. To address this, very recently, researchers have begun to explore unsupervised / semi-supervised DA techniques for deepfake detection, which can also utilize unlabeled data in the target domain. Chen and Tan [14] used the domain adversarial neural network (DANN) architecture to train a deep CNN with labeled source domain data and unlabeled target domain data. Zhang *et al*. [72, 73] used the maximum mean discrepancy (MMD) to quantify the disparity between the source and target domain videos and proposed to train a deep CNN to minimize the MMD. Both MMD and DANN require access to the domain labels (whether a sample is derived from the source or target domain), rather than the task labels (fake / genuine) and can thus leverage unlabeled videos in the target domain.

In our method, we use adversarial training to address the disparity between the source and target domains; we further formulate an unsupervised entropy loss term which operates on the unlabeled target data, and imposes each target sample to align closely with exactly one of the source categories and be distinct from the other category. We conduct extensive experiments to study the performance of our framework under challenging conditions, such as low-resolution images and very few (including none) labeled samples from the target domain of interest.

3. Proposed Framework

3.1. Problem Setup

In our problem setup, we are given data from two domains: source and target, where each domain represents data generated using a particular faking technology (FaceSwap, Face2Face etc.). The data in the source domain are all labeled: $D_S = \{x_i, y_i\}_{i=1}^{N_S}$. In the target domain, we are given labeled samples: $D_T^L = \{x_j, y_j\}_{j=1}^{N_T^L}$, as well as unlabeled samples: $D_T^U = \{x_j\}_{j=1}^{N_T^U}$. As explained in Section 1, the amount of labeled samples in the target domain is scarce, that is, $|D_T^L| \ll |D_T^U|$. Here $\{x\}$ denotes the deep feature representation of a particular image and $\{y\}$ denotes the binary label (fake / genuine). Our objective is to train a deep convolutional neural network (CNN) which will furnish good generalization performance on both the source and target domains; that is, we would like our trained CNN to reliably detect deepfakes generated using both the faking techiques. We propose to formulate a novel loss function and train the network to optimize that loss. Our loss function consists of three components: (i) supervised loss on the labeled data, which encourages the network to be consistent with the labeled data, that is, incur minimal prediction error on the labeled source and labeled target samples; (ii) a strategy to address the disparity between the source and target domains (since the data in the two domains are derived from different faking techniques) and learn feature representations accordingly; and (iii) unsupervised loss on unlabeled target data, which encourages the network to deliver high confidence predictions on the unlabeled target samples. These are detailed in the following sections.

3.2. Supervised Loss on the Labeled Source and Labeled Target Data

The goal of this term is to ensure that the network furnishes accurate predictions on the labeled source and target data. Let $D^L = D_S \cup D_T^L = \{x_1, x_2, \ldots, x_{n_L}\}$ be the labeled source and target data with corresponding labels $\{y_1, y_2, \ldots, y_{n_L}\}$. Since the labels are binary in our problem (fake / genuine), we use the binary cross entropy (BCE) loss to train the deep CNN:

$$\mathcal{L}_{BCE} = -\frac{1}{n_L}\sum_{i=1}^{n_L} y_i.\log(p(y_i)) + (1-y_i).\log(1-p(y_i)) \tag{1}$$

where $p(y_i)$ denotes the probability obtained from the softmax activation layer of the CNN.

3.3. Adversarial Domain Alignment Loss on Source and Target Data

Our domain alignment strategy is inspired by Domain Adversarial Neural Network (DANN) [20]. For the sake of completeness, we review the main idea here. Given an input sample x, our deep CNN will predict its task label y (fake / genuine) and also its domain label, where the target domain is labeled as y_T and the source domain as y_S. Our deep network consists of three components: a feature extractor $F(x;\theta_f)$ which maps an input x into a feature vector f; a task classifier $C(x;\theta_c)$ which maps the feature vector f to a task label y; and a domain classifier $D(x;\theta_d)$ which maps the same feature vector f to a domain label y_T or y_S. The feature extractor will be updated in an adversarial manner with two competing objectives. The feature extractor and the task classifier will be updated such that the task classifier correctly classifies the labeled source and target data. Similarly, the domain classifier will be updated such that it correctly classifies which domain a given sample comes from. At the same time, the feature extractor will also be updated such that the domain classifier cannot correctly classify the domain of a given sample (adversarial component). This will ensure that our model learns domain invariant features, so that a classifier trained on the abundant source domain data can generalize well on the target domain, due to the domain aligned feature distributions.

Let $\mathcal{L}_y$ and $\mathcal{L}_d$ denote the cross entropy loss functions for task classification and domain classification respectively. The loss functions to train the deep neural network are depicted below.

$$\begin{aligned}\arg\min_{\theta_f,\theta_c}&\Big[\mathcal{L}_y(C(F(x)),y) - \mathcal{L}_d(D(F(x)),y_S)\Big]_{(x,y)\in S}\\ &+\Big[\mathcal{L}_y(C(F(x)),y) - \mathcal{L}_d(D(F(x)),y_T)\Big]_{(x,y)\in T}\end{aligned} \tag{2}$$

$$\begin{aligned}\arg\min_{\theta_d}&\Big[\mathcal{L}_d(D(F(x)),y_S)\Big]_{(x,y)\in S}\\ &+\Big[\mathcal{L}_d(D(F(x)),y_T)\Big]_{(x,y)\in T}\end{aligned} \tag{3}$$

Equation (2) updates the parameters of F and C such that C correctly predicts the source and the target task labels and D incorrectly predicts the domain labels (the $\mathcal{L}_d$ terms are negated). Equation (3) updates the parameters of D to correctly predict the domain labels. Using the gradient reversal layer (GRL), Equations (2) and (3) can be combined together into a single equation. The GRL is placed in the deep network between the feature extractor and domain classifier; it flips the gradient during backpropagation while updating the weights. The GRL can be represented as $\mathcal{R}(x)$ with different forward and backward propagation behavior:

$$\mathcal{R}(x) = x; \qquad \frac{\partial \mathcal{R}}{\partial x} = -k\mathcal{I} \tag{4}$$

where $\mathcal{I}$ denotes the identity matrix and k is a constant. The optimization problem for domain alignment thus reduces to:

$$\begin{aligned}\arg\min_{\theta_f,\theta_c,\theta_d}&\Big[\mathcal{L}_y(C(F(x)),y) + \mathcal{L}_d(D(\mathcal{R}(F(x))),y_S)\Big]_{(x,y)\in S}\\ &+\Big[\mathcal{L}_y(C(F(x)),y) + \mathcal{L}_d(D(\mathcal{R}(F(x))),y_T)\Big]_{(x,y)\in T}\end{aligned} \tag{5}$$

We refer to this term as the adversarial loss $\mathcal{L}_{adv}$.

3.4. Unsupervised Loss on Unlabeled Target Data

One of the attractive features of our framework is that it can leverage unlabeled data in the target domain (which is more readily available than labeled data) to further improve the generalization capability of the deep CNN. Inspired by [66], we propose a class alignment loss term which enforces the CNN to predict each unlabeled target sample confidently, and learn feature representations accordingly. Since this is a binary classification problem, each unlabeled target sample can belong to exactly one of the 2 classes (fake / genuine). We assume the presence of M samples from each class j in the labeled source data, where $j \in \{1, 2\}$, and let w_S^{jm} be the m^{th} source output from class j. The fundamental idea is to ensure that the output w_T^i of an unlabeled target sample x_i is similar to all the M source outputs from one of the classes j and dissimilar to the other class (we used the dot product to compute similarity). Enforcing similarity with all the M data points (instead of a single data point) results in a more robust target data class assignment. We define a measure to capture this idea, which quantifies the probability that the target sample x_i is assigned to class j:

$$p_{ij} = \frac{\sum_{m=1}^{M} exp\langle w_T^i, w_S^{jm}\rangle}{\sum_{c=1}^{2}\sum_{m=1}^{M} exp\langle w_T^i, w_S^{cm}\rangle} \tag{6}$$

Here, $\langle\cdot,\cdot\rangle$ denotes the dot product between two vectors, the exponential function $exp(.)$ has been used for ease of differentiability and the denominator ensures that the meaure is normalized, that is, $\sum_j p_{ij} = 1$. When the output of the target sample is similar to exactly one class and dissimilar to the other class, the probability vector p_i tends to be a one-hot vector, with one entry high and the other entry low. This implies that the unlabeled target sample aligns well with exactly one class (fake / genuine), and can thus be interpreted as having low prediction uncertainty (entropy). The class alignment loss is therefore defined to capture the entropy of the target probability vectors:

$$\mathcal{L}_{CA} = -\frac{1}{N_T^U}\sum_{i=1}^{N_T^U}\sum_{j=1}^{2} p_{ij}\log p_{ij} \tag{7}$$

where N_T^U denotes the number of unlabeled target samples. Minimizing this loss produces probability vectors p_i that tend to be one-hot vectors, that is, the unlabeled target data sample outputs are similar to source data outputs from exactly one class. This ensures that the deep network furnishes confident predictions on the unlabeled target data. Computing the similarity with M source samples ensures that the feature representations are learned based on a common similarity between multiple source category data points and the target data point. Note that the probability values in Equation (7) are derived using the class alignment score in Equation (6) and not using class prediction probabilities, as done conventionally. The overall loss function to train the deep CNN can thus be expressed as:

$$\mathcal{L} = \mathcal{L}_{BCE} + \lambda_1\mathcal{L}_{adv} + \lambda_2\mathcal{L}_{CA} \tag{8}$$

where λ_1 and λ_2 are weights governing the relative importance of the terms.

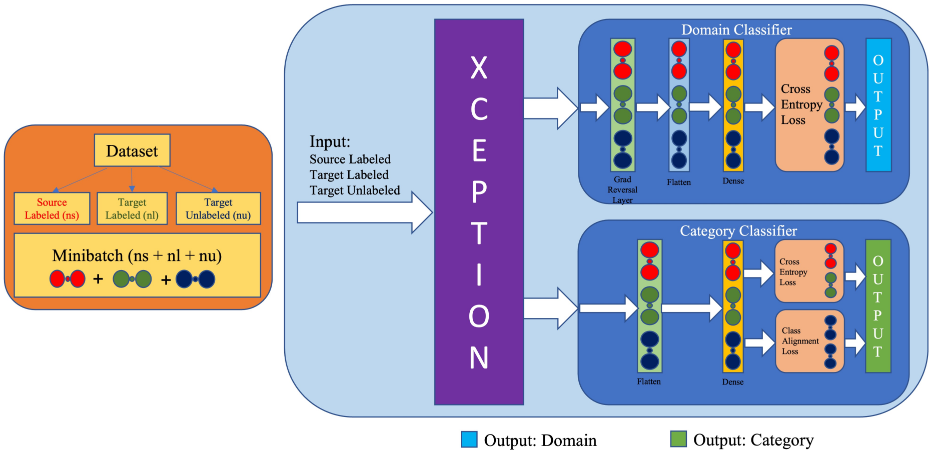


Figure 1. Schematic diagram of the deep neural network architecture used in our study. The network is trained using mini-batches consisting of labeled source samples (red), labeled target samples (green) and unlabeled target samples (blue). The cross-entropy loss for the domain classifier operates on all three types of samples; the cross-entropy loss for the category / label classifier operates only on the labeled source and target samples; the class alignment loss operates only on the unlabeled target samples. Best viewed in color.

4. Experiments and Results

Dataset: The *FaceForensics++ (FF++)* dataset [54] is a benchmark dataset for research in deepfake detection. It includes **Face2Face (F2F)**, **FaceSwap (FS)**, **DeepFakes (DF)** and **NeuralTextures (NT)**. We used the first three faking mechanisms to study the performance of our framework in this paper. The dataset contains 1000 videos for each of these categories. Apart from this, it also contains 1000 Pristine videos. We generated 50 images (128×128) per video, which produced a total of $50,000$ images per faking technique. We used face recognition [2] to detect and crop facial areas in these images.

Comparison Baselines: Since our objective was to address the specific problem of deepfake detection, we selected our comparison baselines from the DA algorithms that have been studied explicitly for this problem. Existing DA techniques for this problem are mostly supervised, that is, they require labeled data in the target domain. We used the following algorithms as comparison baselines in our work: (i) *FT* [62], which freezes some of the layers of the network (trained on the source data) and fine-tunes the deeper layers using the target domain data; (ii) *TGD* [22], a Transferable GAN-images Detection (TGD) framework, which is composed of a teacher and a student model that iteratively teach and evaluate each other to improve the detection performance; (iii) *FReTAL* [31], which employs representation learning and knowledge distillation to perform domain adaptation on new deepfakes in the target domain; (iv) *KD* [31], which uses only the knowledge distillation component of *FReTAL* to perform domain adaptation; and (v) *UDA* [14], an unsupervised DA framework that uses a domain adversarial network to address the disparity between the source and target domains and learn feature representations accordingly. Except *UDA*, all the baselines are supervised. The other unsupervised DA techniques for deepfake detection use MMD for domain alignment [72, 73]; however, considering the popularity and remarkable success of adversarial learning for DA, we only include *UDA* (that uses adversarial domain alignment) as a comparison baseline in this research.

Experimental Setup: In each experiment, we were given images from a source domain and a target domain (each domain represents a particular faking technique). The data in the source set were all labeled. The target set was divided into two parts: a labeled set and an unlabeled set. The number of labeled target samples was much less than the number of unlabeled target samples, to appropriately mimic a real-world application. The test contained an equal number of samples from both the source and target domains to assess the performance of our algorithm on both types of faking technologies, and prevent catastrophic forgetting. We used $42,000$ images as labeled source domain data, $2,000$ images as labeled target domain data, $6,000$ images as the unlabeled target domain data and $20,000$ images ($10,000$ from each of the source and target domains) as the test set. Each experiment was conducted 3 times and the results were averaged to rule out the effects of randomness. The parameters λ_1 and λ_2 were taken as 5 and 25 respectively.

Following [31], we used the *Xception* [15] as the backbone deep neural network architecture in our experiments. Further, Rossler *et al.* [55] demonstrated that Xception achieves the best accuracy on the FaceForensics++ dataset. A schematic diagram of the network architecture used in

[2] https://pypi.org/project/face-recognition/

DA Task	Domain	Proposed	UDA	FReTAL	KD	FT	TGD
DF → F2F	Source	**99.36** ± 0.04	97.29 ± 0.08	92.27 ± 0.93	93.16 ± 5.51	89.73 ± 1.66	91.85 ± 4.65
	Target	**91.35** ± 0.25	86.86 ± 0.35	86.02 ± 0.32	64.23 ± 6.98	82.53 ± 0.53	73.88 ± 4.99
F2F → DF	Source	**99.09** ± 0.24	97.01 ± 0.16	91.00 ± 0.27	90.16 ± 6.14	86.30 ± 0.26	87.87 ± 2.81
	Target	**94.34** ± 0.29	89.17 ± 0.46	91.54 ± 0.13	82.29 ± 2.31	91.05 ± 0.33	85.72 ± 0.47
DF → FS	Source	**99.40** ± 0.13	97.19 ± 0.25	83.10 ± 3.94	95.75 ± 1.25	83.82 ± 0.74	87.34 ± 1.72
	Target	**89.02** ± 1.56	86.80 ± 0.52	85.46 ± 1.83	70.78 ± 4.75	86.68 ± 0.40	82.36 ± 1.09
FS → DF	Source	**98.98** ± 0.18	96.98 ± 0.12	86.49 ± 2.15	92.99 ± 0.56	83.75 ± 1.22	86.71 ± 3.39
	Target	**93.38** ± 0.89	88.86 ± 1.03	89.51 ± 0.38	74.40 ± 1.05	90.92 ± 0.31	83.79 ± 4.54
FS → F2F	Source	**99.16** ± 0.10	95.95 ± 0.04	89.84 ± 0.88	94.56 ± 1.33	87.66 ± 0.62	90.69 ± 1.01
	Target	**89.36** ± 0.30	84.27 ± 1.24	84.91 ± 0.55	67.97 ± 1.56	81.77 ± 0.16	75.48 ± 0.98
F2F → FS	Source	**99.17** ± 0.14	96.17 ± 0.10	90.42 ± 0.88	82.12 ± 3.01	86.96 ± 0.45	81.59 ± 3.84
	Target	**90.86** ± 0.35	85.43 ± 0.61	86.70 ± 0.65	73.46 ± 0.73	85.63 ± 0.16	77.81 ± 2.73

Table 1. Mean (± std) F1 scores (in percentage) of all the methods for 6 out-of-domain deepfake detection tasks. Best F1 values are marked in **bold**. The notation $x \rightarrow y$ implies that x is the source domain and y is the target domain. Results are averaged over 3 runs.

our study is depicted in Figure 1. We used the F1 score on the test set to evaluate the performance of the algorithms, similar to [31].

Implementation Details: We employed the *Stochastic Gradient Descent (SGD)* optimization algorithm to train our Xception model. The learning rate was set to 0.01. We used a batch size of 200 and executed a total of 75 epochs to train the deep model. To enhance the efficiency of training, we incorporated early stopping with a patience value of 10. This mechanism allowed us to halt training if the validation performance did not show any improvement for consecutive epochs, thus preventing overfitting. The experiments were conducted on a DELL ALIENWARE AURORA R15 machine. This machine is equipped with the NVIDIA GeForce RTX 3090 GPU with 24GB of memory. The underlying OS was Ubuntu 22.04.2 LTS. We used TensorFlow 2.12.0 with CUDA 12.1 and Python 3.10.9 to build, train and evaluate our model.

4.1. Main Results

Table 1 reports the performance of all the methods on 6 different out-of-domain deepfake detection tasks (the notation $x \rightarrow y$ implies that x is the source domain and y is the target domain). We note that our framework comprehensively outperforms all the baselines, both in terms of source and target domain F1 scores, across all the 6 tasks. The performance improvement achieved by our method is quite substantial; for instance, in the DF → F2F task, the performance improvement achieved by our method is more than 4% on the target domain, compared to the closest competitor (*UDA*). The supervised detection methods (*FReTAL, KD, FT* and *TGD*) cannot leverage the unlabeled samples in the target domain, and hence depict much lower accuracy values. Even though *UDA* utilizes unlabeled target domain data, it is only used to align the source and target domains; it does not involve any strategy to further leverage the information contained in the unlabeled target samples. In contrast, our framework efficiently utilizes the information in the unlabeled target data through the class alignment loss term. The results show the efficacy of our framework to address the disparity between the source and target domains and also utilize the unlabeled data in the target domain to train a robust detection network. Our framework is not only able to accurately identify deepfakes in the new domain (target), but also retains the knowledge acquired in the original domain (source). The results unanimously corroborate the promise and potential of our method for out-of-domain deepfake detection in real-world applications.

4.2. Performance on Low Resolution Deepfakes

The goal of this experiment was to study the performance of our framework in the challenging setup of low resolution deepfake images. We used images of resolution 64×64 for this experiment (original resolution was 128×128). The results are presented in Table 2. Our framework once again depicts impressive performance and surpasses all the baselines consistently both in the source and target domains. Thus, similar to the previous experiment, our method efficiently retains the knowledge to detect the deepfakes in the source domain, while also learning to detect new types of deepfakes accurately in the target domain. This shows the robustness of our framework to operate in the presence of low quality data, and identify deepfakes even in low resolution data. These results are particularly important from a practical standpoint, since high quality images are not always available in real-world applications.

4.3. Study of the Effect of Labeled Target Samples

In a real-world setup, obtaining a large number of labeled samples in the target domain (new type of deepfake) may not always be feasible. Ideally, we would like to detect deepfakes reliably, even when the target domain contains only unlabeled samples, and no labeled samples are available in the target domain. The goal of this experiment was to study the performace with varying number of labeled images in the target domain. We considered **DF** as the source domain and **F2F** as the target domain for this experiment. The results are reported in Table 3. We studied the per-

DA Task	Domain	Proposed	UDA	FReTAL	KD	FT	TGD
DF → F2F	Source	**99.25** ± 0.01	96.03 ± 0.59	91.60 ± 0.68	89.19 ± 6.49	90.43 ± 1.44	90.37 ± 5.55
	Target	**88.00** ± 0.54	81.93 ± 2.73	83.88 ± 0.35	58.80 ± 1.50	80.04 ± 0.40	68.90 ± 2.25
F2F → FS	Source	**98.94** ± 0.05	93.84 ± 0.04	89.65 ± 0.50	89.80 ± 3.76	85.63 ± 0.51	86.40 ± 4.39
	Target	**87.31** ± 0.45	81.11 ± 0.17	84.44 ± 0.49	66.64 ± 1.66	83.91 ± 0.77	76.34 ± 1.32
FS → DF	Source	**99.03** ± 0.25	95.56 ± 0.11	90.97 ± 1.76	97.36 ± 0.62	89.43 ± 0.89	90.81 ± 0.72
	Target	**92.97** ± 0.39	87.51 ± 0.56	90.89 ± 0.05	80.85 ± 2.75	90.81 ± 0.20	79.87 ± 5.38

Table 2. Mean (± std) F1 scores (in percentage) of all the methods for detecting low resolution deepfakes. Best F1 values are marked in **bold**. The notation $x \rightarrow y$ implies that x is the source domain and y is the target domain. Results are averaged over 3 runs.

LT	Domain	Proposed	UDA	FReTAL	KD	FT	TGD
0	Source	**97.74** ± 0.91	97.55 ± 0.41	N/A	N/A	N/A	N/A
	Target	73.36 ± 2.27	**82.17** ± 1.67	N/A	N/A	N/A	N/A
800	Source	99.25 ± 0.31	97.44 ± 0.15	**99.44** ± 0.07	94.36 ± 4.48	90.30 ± 0.55	85.49 ± 2.23
	Target	81.34 ± 3.06	**83.49** ± 1.26	51.32 ± 0.23	65.65 ± 4.79	76.07 ± 0.72	75.51 ± 0.87
1600	Source	**99.33** ± 0.07	97.24 ± 0.46	91.29 ± 0.91	82.68 ± 4.18	70.49 ± 10.96	72.65 ± 9.64
	Target	**89.54** ± 0.29	86.04 ± 0.85	85.88 ± 0.10	60.79 ± 0.88	69.00 ± 9.65	64.91 ± 4.40
2400	Source	**99.31** ± 0.03	97.24 ± 0.34	92.26 ± 0.14	96.38 ± 1.03	91.04 ± 0.65	82.61 ± 5.99
	Target	**92.33** ± 0.59	87.09 ± 0.71	89.28 ± 0.31	64.08 ± 6.97	84.85 ± 0.66	82.61 ± 5.99

Table 3. Mean (± std) F1 scores (in percentage) of all the methods with varying number of labeled samples in the target domain for the $DF \rightarrow F2F$ task. Best F1 values are marked in **bold**. Results are averaged over 3 runs. **LT** denotes the number of labeled samples in the target domain.

formance with $0, 800, 1600$ and 2400 labeled samples in the target domain; all the other parameters were kept constant. The supervised detection methods (*FReTAL, KD, FT* and *TGD*) require labeled samples in the target domain and are hence not applicable when the target domain contains only unlabeled data (first row of the table). Our method and *UDA* can operate even in the absence of labeled target domain data, which corroborates their practical usefulness. Our framework once again depicts impressive performance with varying number of labeled samples in the target domain. This shows that it can be deployed in applications where very little supervision information is available about a new type of deepfake, that we are interested to detect. This further reinforces the usefulness of our framework for real-world applications. We also note that with an increase in the number of labeled samples in the target domain, the F1 score on the target domain improves, which is intuitive.

4.4. Study of the Effect of Unlabeled Target Samples

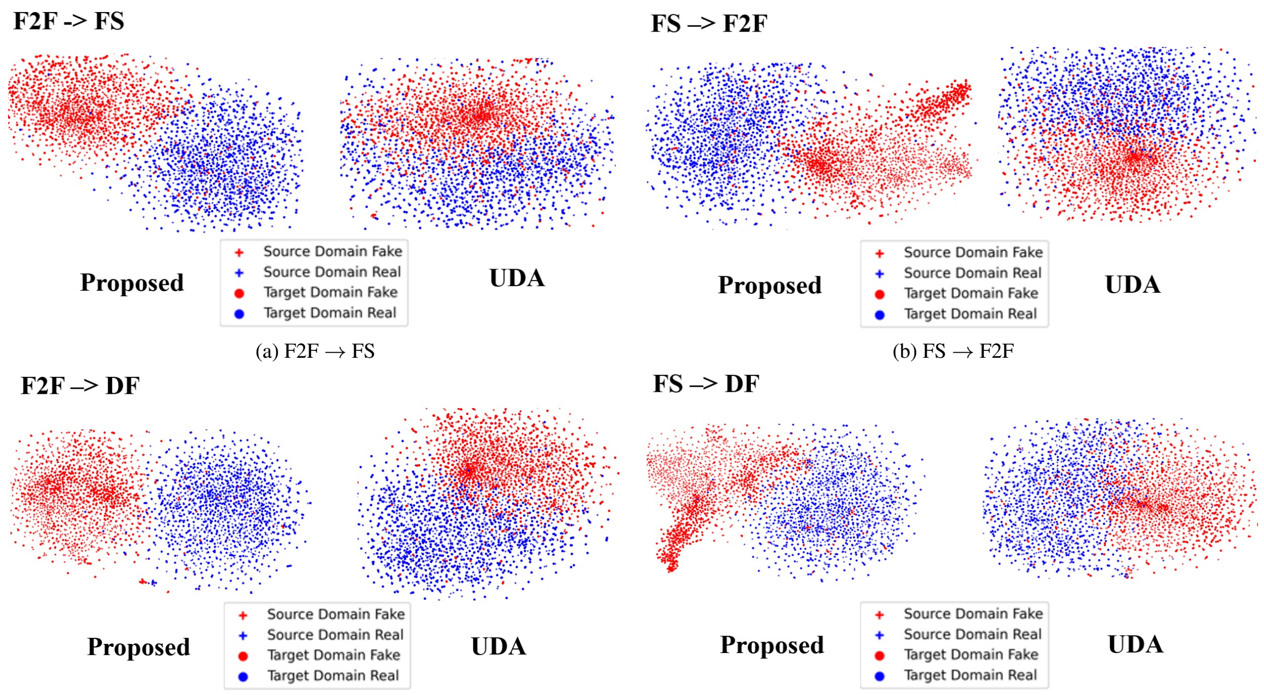
In this experiment, we studied the effect of the number of unlabeled samples in the target domain. The performance of the supervised detection methods (*FReTAL, KD, FT* and *TGD*) will not be affected by a change in the number of unlabeled samples in the target domain; they were hence excluded from this study. We conducted experiments with $3600, 7200, 9000$ and 10800 unlabeled samples in the target domain. The results are presented in Table 4. The proposed framework consistently outperforms *UDA* in both the source and target domains, across all the different number of unlabeled samples in the target domain, which corroborates its efficacy. With 3600 unlabeled samples in the target domain, the improvement in F1 score in the target domain is almost 5%. The performance in the target domain of both methods increases slightly with an increase in the number of unlabeled target samples.

UT	Domain	Proposed	UDA
3600	Source	**99.32** ± 0.03	97.19 ± 0.25
	Target	**89.30** ± 0.54	84.68 ± 0.18
7200	Source	**99.29** ± 0.06	96.87 ± 0.03
	Target	**90.41** ± 0.40	86.55 ± 0.74
9000	Source	**99.27** ± 0.11	96.68 ± 0.04
	Target	**90.77** ± 0.07	86.69 ± 0.18
10800	Source	**99.15** ± 0.07	96.79 ± 0.03
	Target	**90.79** ± 0.31	87.52 ± 0.74

Table 4. Mean (± std) F1 scores (in percentage) of all the methods with varying number of unlabeled samples in the target domain for the $DF \rightarrow F2F$ task. Best F1 values are marked in **bold**. Results are averaged over 3 runs. **UT** denotes the number of unlabeled samples in the target domain.

4.5. Feature Visualizations

In this experiment, we studied the t-SNE embeddings of the features learned by the proposed framework. We compared our results with the *UDA* method, as it is also an unsupervised method that can utilize unlabeled target domain data. The results are depicted in Figure 2 for 4 different out-of-domain deepfake detection tasks. Here, each color denotes a category (*blue* denotes genuine, *red* denotes fake) and each symbol denotes a domain (*plus* denotes source, *circle* denotes target). As evident visually, the proposed method shows a better clustering of the two categories (blue and red clusters) and a better overlap between the source and target domains, compared to *UDA*. Thus, using our domain alignment and class alignment loss functions, the



(a) F2F → FS

(b) FS → F2F

(c) F2F → DF

(d) FS → DF

Figure 2. t-SNE visualization results. Best viewed in color.

deep model is able to learn discriminating feature representations, which minimize the disparity between the source and target deepfakes and also separate the real and fake images from the two domains. This accounts for its superior performance, as evidenced in Tables 1, 2, 3 and 4.

4.6. Ablation Study

We conducted an experiment to study the performance of our framework without the unsupervised class alignment loss term $\mathcal{L}_{CA}$ in Equation (7). The F1 score on the target test set for three different detection tasks are reported in Table 5. We note that the performance of our framework is affected in the absence of the unsupervised class alignment loss term on the target domain data. This shows the usefulness of $\mathcal{L}_{CA}$ to leverage the information in the unlabeled target domain data, learn discriminating feature representations and boost the performance of our method.

DA Task	**Proposed**	**Proposed w/o $\mathcal{L}_{CA}$**
DF → F2F	91.35	87.27
F2F → DF	94.34	89.81
FS → DF	93.38	89.01

Table 5. Ablation study results.

5. Conclusion and Future Work

Thanks to the tremendous progress of generative AI, detecting deepfakes reliably has become a problem of immense practical importance in today's world. While CNNs have demonstrated promise in detecting deepfakes, their performance is affected drastically when validated on deepfakes generated using a different faking technique (out-of-domain). We proposed a novel semi-supervised deep domain adaptation algorithm to address this challenging problem. Contrary to most methods that have studied this problem, our framework can leverage unlabeled data in the target domain (which is more readily available than labeled data) through a class alignment loss term. Our extensive experimental studies on the benchmark FaceForensics++ dataset demonstrated the efficacy of our method against competing baselines. We hope this research will motivate the development of other unsupervised / semi-supervised DA algorithms for the challenging problem of out-of-domain deepfake detection.

As part of future work, we plan to incorporate data augmentation in our DA pipeline, which has shown remarkable success in deepfake detection [18]. We also plan to validate the performance of our algorithm on other challenging deepfake datasets, such as DFDC [18] and CelebDF [40].

6. Acknowledgment

This research was supported in part by the National Science Foundation under Grant Number: IIS-2143424 (NSF CAREER Award).

References

[1] Deepfake porn is still a threat, particularly for k-pop stars. https://www.rollingstone.com/culture/culture-news/deepfakes-nonconsensual-porn-study-kpop-895605/. [Online; Accessed: 2023-03-25]. 1

[2] Deepfakes Reddit. Fakeapp. https://www.malavida.com/en/soft/fakeapp/. [Online; Accessed: 2023-03-25]. 2

[3] Faceapp-most popular selfie editor. https://www.faceapp.com/. [Online; Accessed: 2023-03-25]. 1, 2

[4] 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/. [Online; Accessed: 2023-03-25]. 1

[5] 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/. [Online; Accessed: 2023-03-25]. 1

[6] Most AI-generated deepfake videos online are porn. https://www.pcmag.com/news/most-ai-generated-deepfake-videos-online-are-porn. [Online; Accessed: 2023-03-25]. 1

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

[8] S. Agarwal, H. Farid, T. El-Gaaly, and S. Lim. Detecting deep-fake videos from appearance and behavior. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, 2020. 2

[9] S. Agarwal, H. Farid, Y. Gu, M. He, K. Nagano, and H. Li. Protecting world leaders against deep fakes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019. 1, 2

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

[11] J. Aldredge. Is deepfake technology the future of the film industry? *Prem. Beat by shutterstock*, 2020. 1

[12] J. H. Bappy, A. K. Roy-Chowdhury, J. Bunk, L. Nataraj, and B. S. Manjunath. Exploiting spatial structure for localizing manipulated image regions. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 1, 2

[13] J. H. Bappy, C. Simons, L. Nataraj, B. S. Manjunath, and A. K. Roy-Chowdhury. Hybrid LSTM and encoder–decoder architecture for detection of image forgeries. *IEEE Transactions on Image Processing (TIP)*, 28(7):3286–3300, 2019. 1, 2

[14] B. Chen and S. Tan. Featuretransfer: Unsupervised domain adaptation for cross-domain deepfake detection. *Security and Communication Networks*, 2021, 2021. 3, 5

[15] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5

[16] D. Cozzolino, J. Thies, A. Rossler, C. Riess, M. Niebner, and L. Verdoliva. Forensictransfer: Weakly-supervised domain adaptation for forgery detection. In *arXiv:1812.02510v2*, 2019. 1, 3

[17] J. DelViscio. A nixon deepfake, a ”moon disaster” speech and an information ecosystem at risk. *Scientific American*, 20, 2020. 1

[18] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, and C. Ferrer. The deepfake detection challenge (DFDC) preview dataset. In *arXiv preprint arXiv:1910.08854*, 2019. 8

[19] FaceSwapDevs. Deepfakes faceswap - github repository. [Online; accessed 23-March-2021], 2019. 1

[20] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research (JMLR)*, 17, 2016. 2, 3

[21] M. Huh, A. Liu, A. Owens, and A. A. Efros. Fighting fake news: Image splice detection via learned self-consistency. In *European Conference on Computer Vision (ECCV)*, 2018. 2

[22] H. Jeon, Y. Bang, J. Kim, and S. S. Woo. TGD: Transferable GAN-generated images detection framework. In *arXiv:2008.04115*, 2020. 2, 5

[23] H. Jeon, Y. Bang, and S. S. Woo. Faketalkerdetect: Effective and practical realistic neural talking head detection with a highly unbalanced dataset. In *IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 1285–1287. IEEE, 2019. 2

[24] H. Jeon, Y. Bang, and S. S. Woo. Fdftnet: Facing off fake images using fake detection fine-tuning network. In *IFIP international conference on ICT systems security and privacy protection*, pages 416–430. Springer, 2020. 2

[25] C. Jones. 1 in 3 who are aware of deepfakes say they have inadvertently shared them on social media. [Online; accessed 24-March-2021], November 2020. 1

[26] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *arXiv:1710.10196*, 2017. 1

[27] R. Kemker, M. McClure, A. Abitino, T. Hayes, and C. Kanan. Measuring catastrophic forgetting in neural networks. In *AAAI Conference on Artificial Intelligence*, 2018. 2

[28] H. Khalid and S. S. Woo. Oc-fakedect: Classifying deepfakes using one-class variational autoencoder. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 656–657, 2020. 2

[29] A. Khodabakhsh, R. Ramachandra, K. Raja, P. Wasnik, and C. Busch. Fake face detection methods: Can they be generalized? In *International Conference of the Biometrics Special Interest Group (BIOSIG)*, 2018. 1

[30] M. Kim, S. Tariq, and S. Woo. Cored: Generalizing fake media detection with continual representation using distillation. In *ACM International Conference on Multimedia*, pages 337–346, 2021. 3

[31] M. Kim, S. Tariq, and S. S. Woo. Fretal: Generalizing deepfake detection using knowledge distillation and representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1001–1012, 2021. 1, 2, 5, 6

[32] D. Kingma and M. Welling. Auto-encoding variational bayes. In *arXiv preprint arXiv:1312.6114*, 2013. 2

[33] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. Rusu, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. 2

[34] P. Korshunov and S. Marcel. Deepfakes: a new threat to face recognition? assessment and detection. In *arXiv:1812.08685v1*, 2018. 1

[35] S. Lee, S. Tariq, J. Kim, and S. S. Woo. Tar: Generalized forensic framework to detect deepfakes using weakly supervised learning. In *IFIP International Conference on ICT Systems Security and Privacy Protection*, pages 351–366. Springer, 2021. 2

[36] S. Lee, S. Tariq, Y. Shin, and S. S. Woo. Detecting handcrafted facial image manipulations and gan-generated facial images using shallow-fakefacenet. *Applied Soft Computing*, 105, 2021. 2

[37] L. Li, J. Bao, T. Zhang, H. Yang, D. Chen, F. Wen, and B. Guo. Face X-Ray for more general face forgery detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[38] Y. Li, M. Chang, and S. Lyu. In Ictu Oculi: Exposing AI created fake videos by detecting eye blinking. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–7. IEEE, 2018. 1, 2

[39] Y. Li and S. Lyu. Exposing deepfake videos by detecting face warping artifacts. In *arXiv:1811.00656*, 2018. 2

[40] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu. A large-scale challenging dataset for deepfake forensics. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 8

[41] K. Liu, I. Perov, D. Gao, N. Chervoniy, W. Zhou, and W. Zhang. Deepfacelab: Integrated, flexible and extensible face-swapping framework. *Pattern Recognition*, 141, 2023. 2

[42] M. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *Advances of Neural Information Processing Systems (NIPS)*, 2017. 2

[43] M. Long, Y. Cao, J. Wang, and M. Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning (ICML)*, 2015. 2

[44] M. Long, H. Zhu, J. Wang, and M. Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances of Neural Information Processing Systems (NIPS)*, 2016. 2

[45] F. Matern, C. Riess, and M. Stamminger. Exploiting visual artifacts to expose deepfakes and face manipulations. In *IEEE Winter Conference on Applications of Computer Vision Workshops (WACVW)*, pages 83–92. IEEE, 2019. 2

[46] Y. Mirsky and W. Lee. The creation and detection of deepfakes: A survey. *ACM Computing Surveys (CSUR)*, 54(1):1–41, 2021. 1

[47] T. T. Nguyen, Q. V. H. Nguyen, D. T. Nguyen, D. T. Nguyen, T. Huynh-The, S. Nahavandi, T. Nguyen, Q. Pham, and C. Nguyen. Deep learning for deepfakes creation and detection: A survey. *Computer Vision and Image Understanding (CVIU)*, 223, 2022. 1

[48] M. Ning, D. Lu, D. Wei, C. Bian, C. Yuan, S. Yu, K. Ma, and Y. Zheng. Multi-anchor active domain adaptation for semantic segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[49] S. Pan, I. Tsang, J. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2009. 2

[50] S. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 22(10), 2010. 2

[51] D. Pardoe and P. Stone. Boosting for regression transfer. In *International Conference on Machine Learning (ICML)*, 2010. 2

[52] T. Park, M. Liu, T. Wang, and J. Zhu. Gaugan: semantic image synthesis with spatially adaptive normalization. In *ACM SIGGRAPH 2019 Real-Time Live!*, pages 1–1. 2019. 1

[53] D. Patterson. President's words used to create "deepfakes" at davos. *Video*, 2020. 1

[54] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner. Faceforensics: A large-scale video dataset for forgery detection in human faces. In *arXiv:1803.09179*, 2018. 2, 5

[55] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner. Faceforensics++: Learning to detect manipulated facial images. In *IEEE/CVF International conference on computer vision (ICCV)*, pages 1–11, 2019. 1, 2, 5

[56] R. Salloum, Y. Ren, and C. J. Kuo. Image splicing localization using a multi-task fully convolutional network (MFCN). *Journal of Visual Communication and Image Representation*, 51:201–209, 2018. 1, 2

[57] J. Shen, Y. Qu, W. Zhang, and Y. Yu. Wasserstein distance guided representation learning for domain adaptation. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2018. 2

[58] S. Tariq, S. Jeon, and S. S. Woo. Am I a real or fake celebrity? Measuring commercial face recognition web apis under deepfake impersonation attack. In *arXiv:2103.00847*, 2021. 2

[59] S. Tariq, S. Lee, H. Kim, Y. Shin, and S. S. Woo. Detecting both machine and human created fake face images in the wild. In *International workshop on multimedia privacy and security*, pages 81–87, 2018. 2

[60] S. Tariq, S. Lee, H. Kim, Y. Shin, and S. S. Woo. GAN is a friend or foe? a framework to detect various fake face images. In *ACM/SIGAPP Symposium on Applied Computing*, pages 1296–1303, 2019. 2

[61] S. Tariq, S. Lee, and S. S. Woo. A convolutional LSTM based residual network for deepfake video detection. In *arXiv:2009.07480*, 2020. 2

[62] S. Tariq, S. Lee, and S. S. Woo. One detector to rule them all: Towards a general deepfake attack detection framework. In *Proceedings of the web conference (WWW)*, pages 3625–3637, 2021. 1, 2, 3, 5

[63] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Niessner. Face2face: Real-time face capture and reenactment of rgb videos. In *IEEE conference on computer vision and pattern recognition (CVPR)*, pages 2387–2395, 2016. 1

[64] B. Thompson, J. Gwinnup, H. Khayrallah, K. Duh, and P. Koehn. Overcoming catastrophic forgetting during domain adaptation of neural machine translation. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2062–2068, 2019. 2

[65] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[66] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 4

[67] J. Vincent. Watch jordan peele use ai to make barack obama deliver a psa about fake news. *The Verge*, 17, 2018. 1

[68] Y. Xu, X. Zhong, A. Yepes, and J. Lau. Forget me not: Reducing catastrophic forgetting for domain adaptation in reading comprehension. In *IEEE International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020. 2

[69] X. Yang, Y. Li, and S. Lyu. Exposing deep fakes using inconsistent head poses. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8261–8265, 2019. 1, 2

[70] K. You, M. Long, Z. Cao, J. Wang, and M. Jordan. Universal domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[71] Z. Yu, J. Li, Z. Du, L. Zhu, and H. Shen. A comprehensive survey on source-free domain adaptation. In *arXiv:2302.11803v1*, 2023. 2

[72] M. Zhang, H. Wang, P. He, A. Malik, and H. Liu. Exposing unseen gan-generated image using unsupervised domain adaptation. *Knowledge-Based Systems*, 257, 2022. 3, 5

[73] M. Zhang, H. Wang, P. He, A. Malik, and H. Liu. Improving GAN-generated image detection generalization using unsupervised domain adaptation. In *IEEE International Conference on Multimedia and Expo (ICME)*, 2022. 1, 3, 5

[74] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis. Two-stream neural networks for tampered face detection. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1831–1839. IEEE, 2017. 1, 2

[75] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis. Learning rich features for image manipulation detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1053–1061, 2018. 1, 2