Generalized Zero and Few-Shot Transfer for Facial Forgery Detection

Shivangi Aneja
Visual Computing Lab
Technical University of Munich
shivangi.aneja@tum.de

Matthias Nießner
Visual Computing Lab
Technical University of Munich
niessner@tum.de

Abstract

We propose Deep Distribution Transfer (DDT), a new transfer learning approach to address the problem of zero and few-shot transfer in the context of facial forgery detection. We examine how well a model (pre-)trained with one forgery creation method generalizes towards a previously unseen manipulation technique or different dataset. To facilitate this transfer, we introduce a new mixture model-based loss formulation that learns a multi-modal distribution, with modes corresponding to class categories of the underlying data of the source forgery method. Our core idea is to first pre-train an encoder neural network, which maps each mode of this distribution to the respective class labels, i.e., real or fake images in the source domain by minimizing wasserstein distance between them. In order to transfer this model to a new domain, we associate a few target samples with one of the previously trained modes. In addition, we propose a spatial mixup augmentation strategy that further helps generalization across domains. We find this learning strategy to be surprisingly effective at domain transfer compared to a traditional classification or even state-of-the-art domain adaptation/few-shot learning methods. For instance, compared to the best baseline, our method improves the classification accuracy by 4.88% for zero-shot and by 8.38% for the few-shot case transferred from the FaceForensics++ to Dessa dataset.

1 Introduction

The rapid progress of image and video generation techniques has sparked a heated discussion regarding the authenticity and handling of visual content, for instance, on social media or online video platforms. In particular, so-called *DeepFakes* videos have become a central topic, as they have shown the potential to create manipulated fake videos of human faces convincingly. These methods include face-swapping techniques that artificially insert a specific person into a given target video [9; 34; 36], as well as facial reenactment methods where the goal is to modify the expression of a face such that the person appears to be saying something different [46; 22; 47]. As a result, these methods could be misused, for instance, the speech of a politician or news commentator could be deliberately altered to communicate malicious propaganda backed with forged visual content to create a compelling forgery.

This increasing availability of manipulation methods calls for the need to reliably detect such forgeries in an automated fashion. In particular, learning-based techniques have shown promising results on both images and videos proposed [40; 33; 28; 53; 55; 1; 3]. While training supervised classifiers with one particular manipulation technique performs quite well on the same method, the main challenge lies in keeping the underlying training up to date with respect to the most recent forgery approaches. For instance, the popular FaceForensics [40] effort first provided a dataset of Face2Face [46] videos; a year later, Face Swap [25] and DeepFakes [9] were added, and finally they authors released videos that were edited using NeuralTextures [47]. Unfortunately, providing constant updates for this supervised training is highly impractical since new manipulation techniques could appear from one day to another without any prior information. This motivates us to re-think facial forgery detection from a zero and few-shot learning perspective. Here, our aim is to generalize features between each method and reliably detect manipulations, despite having seen no or only very few training samples of a particular forgery technique.

Domain adaptation methods [48; 44; 32; 52] have the potential to be applicable in this scenario. For instance, they aim to align classes from source and target domain irrespective of their original domain by learning a common subspace. Other few-shot learning methods [45; 43; 49; 13; 27] successively train a model on a large variety of classes with very few samples per class across different episodes, ultimately learning how to generalize for unseen classes. Both of these directions have been well-studied on the standard benchmarks with a larger number of classes and where samples from different classes are structurally very different from each other; e.g., Omniglot [26], CUB [50], miniImageNet [49], Office-31 [41], VisDA [35] etc.. However, we found that traditional domain adaption methods struggle in our forgery detection scenario, which maps to a binary classification problem with a high visual similarity between each class.

To address these challenges, we introduce Deep Distribution Transfer (DDT), a novel transfer learning method tailored for forgery detection that encompasses both zero and few-shot learning to construct a domain-agnostic embedding space. Specifically, we propose a new mixture model-based loss formulation that learns a multi-modal distribution, with modes corresponding to classes from the source domain. First, we pre-train an encoder network, such that the latent codes of each mode of the learned distribution are mapped to the respective class labels (Real and Fake) for the source dataset. To transfer this model to a new domain, we associate each sample from the target domain with one of the previously trained modes. Classification is then performed by assigning the class label of the closest mode.

To summarize, the key contributions in this paper are:

- We propose DDT, a novel zero and few-shot transfer learning method for facial forgery detection method that explicitly models the underlying data with a multimodal distribution.
- We introduce a spatial mixup augmentation strategy that improves the model's generalization capability in unseen scenarios (w.r.t. manipulation methods and datasets).
- We provide an extensive analysis of transfer among different facial forgery methods within and across different datasets, showing that our approach outperforms state-of-the art on the Dessa dataset by 4.88% in the zero-shot case and by 8.38% for the few shot case.

2 Related Work

Facial Manipulation Methods: Facial video manipulation methods have a long history in computer graphics, coupled with the 3D reconstruction of the underlying 3D face geometry [5; 15; 46]. Recently, we have also seen GAN-based synthesis methods generate high-quality facial imagery at remarkable detail [21; 20; 19; 7; 42; 51; 51]; however, they struggle with temporal coherency. Hybrid methods, such as Deep Video Portraits [22] or Deferred Neural Rendering [47], combine the advantages of both directions by using rendered facial reconstructions as conditioning for generative neural networks, hence providing stable anchors for the temporal domain and enabling high-quality, photo-realistic video editing. From an application standpoint, we can categorize these techniques into (a) facial re-enactment, which changes the expressions of the person while keeping the same identity, and (b) identity swapping, which replaces the facial region with another person's face.

Facial Manipulation Detection: Traditional facial manipulation leverages handcrafted features such as gradients or compression patterns, in order to find inconsistencies within an image [2; 12; 30]. While such self-consistency can produce good results, these methods are less accurate than more recent learning-based techniques based on convolutional neural networks [53]. Hybrid methods can leverage both traditional and learned features in a two-stream fashion [55], or focus on face-specific expressions and motion to identify an individual person [40; 3]. These detection methods are, however, supervised in nature, and struggle to generalize between domains and datasets.

Zero and Few-Shot Learning: Zero and few-shot learning methods [6; 13; 49; 43; 45; 14; 39; 37; 38; 27; 23; 29] have the potential to facilitate transfer across manipulation techniques or datasets; for instance, meta-learning based methods successively train a model on a large variety of tasks across different episodes. An alternative direction is ProtoNets [43], which learns the prototype embedding for every class and uses the similarity between each prototype and query embedding for classification; or Relation Nets [45] which also computes the prototype for every class while proposing a learned similarity metric. Very recently, Cozzolino et al. [8] propose an autoencoder-based approach targeting forensics applications, including facial manipulations, which we consider an important baseline for our work. Their method proposes to use a pre-trained embedding on the source domain, which is then fine-tuned on the target data. While this leads to better results than a traditional binary cross-entropy loss, generalization nonetheless remains limited. In contrast, we learn to model each class as a distribution, leading to significantly better generalization to new forgery methods as well as unseen datasets.

3 Proposed Approach

The key idea of our method is to model classes as distributions, irrespective of their domain, where data points are projected into a learned embedding space where activated components reflect each activated class. This multi-modal distribution can then be used to efficiently facilitate zero and few-shot learning in the respective target domain; see Fig. 1 for an overview.

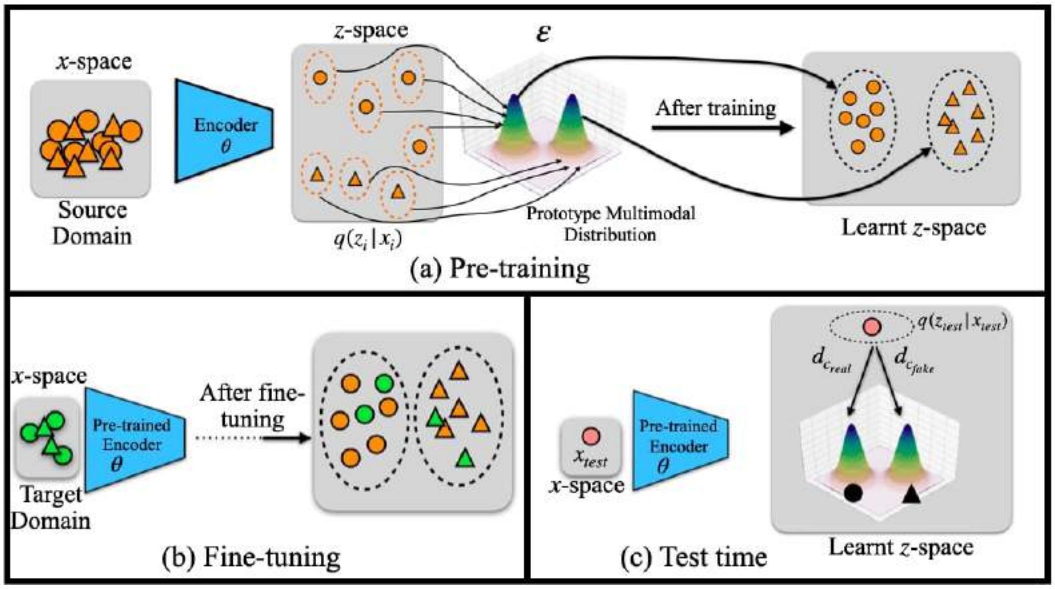


Figure 1: Method overview. (a) Pre-training: $\boldsymbol{\theta}$ encodes samples from the source domain into a latent distribution $q(\boldsymbol{z}_i|\boldsymbol{x}_i)$. $\boldsymbol{\epsilon}_c$ then maps class labels c to the encoded distributions of the prototype multi-modal distribution $\boldsymbol{\varepsilon}$. (b) Fine-tuning: the pre-trained encoder $\boldsymbol{\theta}$ is used to map the few-shot samples from the target dataset with the same prototype multi-modal distribution $\boldsymbol{\varepsilon}$, which learns a common subspace between samples across domains. (c) Test-time: a test sample $\boldsymbol{x}_{test}$ is encoded into the latent distribution $q(\boldsymbol{z}_{test}|\boldsymbol{x}_{test})$ by using the pre-trained encoder $\boldsymbol{\theta}$. We then compute the distance of the latent code with respect to all components of the distribution, and assign a class label based on the component it is closest to.

3.1 Modeling of Class Distributions

We are given a large source dataset $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ of N labeled samples and a small target dataset $\mathcal{T} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{M}$ of M labeled samples such that $M \ll N$, where each $\boldsymbol{x}_i \in \mathbb{R}^D$ is a D-dimensional input image and $y_i \in \{0, 1\}$ is the corresponding label (0 for Real, 1 for Fake). $\mathcal{S}_c$ and $\mathcal{T}_c$ denotes the set of data points labeled with class c in source and target dataset, respectively.

To model each class as a distribution, we learn a multi-modal Gaussian distribution $\boldsymbol{\varepsilon}$ with non-overlapping means, which enforces that every component $\boldsymbol{\epsilon}_c$ acts as a unimodal distribution in its own space. By learning a single multi-modal distribution $\boldsymbol{\varepsilon}$ with non-overlapping means, we obtain many unimodal distributions $\boldsymbol{\epsilon}_c$, each representing a particular class c, which we refer to as a prototype multi-modal distribution. This distribution is a Gaussian mixture model consisting of several Gaussian distributions in the latent space, each identified by $c \in \{0, ..., C-1\}$, where C is the number of classes in our dataset. Each Gaussian $\boldsymbol{\epsilon}_c$ in the mixture represents a class distribution and consists of following parameters:

- Mean $\boldsymbol{m}_c$, defining its center.
- Covariance Σ_c, defining its width. For brevity, we assume $\Sigma_c = I$.

$$\boldsymbol{\varepsilon} = \begin{cases} \boldsymbol{\epsilon_0} = \mathcal{N}(\boldsymbol{z}; \boldsymbol{m}_0, I) \\ \boldsymbol{\epsilon_1} = \mathcal{N}(\boldsymbol{z}; \boldsymbol{m}_1, I) \\ \vdots \\ \boldsymbol{\epsilon_{C-1}} = \mathcal{N}(\boldsymbol{z}; \boldsymbol{m}_{C-1}, I) \end{cases} \tag{1}$$

Naively learning the class means $\boldsymbol{m}_c$ using the source training data $\mathcal{S}_c$ would severely overfit to specific forgery method or dataset; i.e., it would learn a specific feature representation for this dataset but without the capability to transfer to other manipulation methods or datasets. To this end, we regularize the model to ensure that the class means $\boldsymbol{m}_c$ do not overlap with each other. Specifically, we constrain these means to be vectors consisting of 1s and 0s, with the equal number of neurons activating for each class. From the number of classes C and size of the embedding space K, the class mean $\boldsymbol{m}_c$ with class label c is then:

$$\boldsymbol{m}_c = \left[\{1\}^{p}_{i=p*(c...(c+1))}, \{0\}^{p*(C-1)}_{i \neq p*(c...(c+1))} \right]^{K}, \tag{2}$$

where p is number of activated neurons per class and is given by $p = \frac{K}{C}$, with i denoting the index location and c the class label.

3.2 Learning Class Distributions

Using an embedding function $\boldsymbol{\theta}$ with learnable parameters, we first project the data points from the D-dimensional input space ($\boldsymbol{x}$) to a smaller K-dimensional embedding ($\boldsymbol{z}$) space. Every data point $\boldsymbol{x}_i$ is modeled as a distribution $q(\boldsymbol{z}_i|\boldsymbol{x}_i) \sim \mathcal{N}(\boldsymbol{z}_i; \boldsymbol{\mu}_i, \Sigma_i)$ in this embedding space. To model each class as a single distribution component, we map all data point distributions $q(\boldsymbol{z}_i|\boldsymbol{x}_i)$ belonging to a particular class c with a fixed component $\boldsymbol{\epsilon}_c$ of the prototype multi-modal distribution $\boldsymbol{\varepsilon}$ described above, given by Equation 1. For our binary classification case, this model learns to project each data point to a latent space $\boldsymbol{z}$, where each data point is mapped to only one component of a bimodal distribution ($\mathcal{N}(\boldsymbol{m}_0, I)$ for *Real* and $\mathcal{N}(\boldsymbol{m}_1, I)$ for *Fake*); each component will then represent one of the classes.

(1) Pre-training: Every sample x_i in the source domain $\mathcal{S}$ is encoded into a latent distribution $q(\boldsymbol{z}_i|\boldsymbol{x}_i)$, where we minimize the distribution divergence metric d between this encoded latent distribution $q(\boldsymbol{z}_i|\boldsymbol{x}_i)$ with the corresponding component of our fixed prototype multi-modal distribution $\boldsymbol{\epsilon}_c$ (based on its class label c):

$$\mathcal{L}_{pretrain} = \frac{1}{N} \left[\sum_{i=1}^{N} d \Bigg(q(\boldsymbol{z}_i|\boldsymbol{x}_i, y_i), \boldsymbol{\epsilon}_c \Bigg) \right] \tag{3}$$

such that $y_i = c$, where c is the ground truth label for sample $\boldsymbol{x}_i$ and $\boldsymbol{\epsilon}_c$ is the corresponding prototype distribution component. This minimization learns an embedding subspace such that all samples belonging to a class c are aligned close to each other with the component $\boldsymbol{\epsilon}_c$ of the prototype distribution; samples belonging to different classes are mapped with the separate components (see Fig. 1(a)).

(2) Fine-tuning (few-shot only): For the target domain $\mathcal{T}$, we associate the few available training samples with the same prototype distribution $\boldsymbol{\varepsilon}$ used before for learning the source domain $\mathcal{S}$, according to its class label. We then fine-tune the above pre-trained model $\boldsymbol{\theta}$ with the target dataset. In order to further mitigate overfitting during fine-tuning, we propose a spatial mixup augmentation strategy. Specifically, we spatially mix two images in a vertical fashion, forming a new image by taking half of the face from one image and another half from the other images.

(3) Testing: We encode each test sample $\boldsymbol{x}_{test}$ into the latent distribution $q(\boldsymbol{z}_{test}|\boldsymbol{x}_{test})$ by using the trained encoder $\boldsymbol{\theta}$. We then compute the distance of this encoded latent distribution with respect to all the components of the prototype multi-modal distribution, and assign the label to the component it is closest to:

$$\hat{y}_{test} = argmin\Big\{d_0, d_1, ...d_{C-1}\Big\} \tag{4}$$

where d_c is the distribution divergence distance between $q(\boldsymbol{z}_{test}|\boldsymbol{x}_{test})$ and $\boldsymbol{\epsilon}_c$.

Distribution Alignment Distance: Every sample $\boldsymbol{x}_i$ that is fed to encoder model $\boldsymbol{\theta}$ outputs a distribution $q(\boldsymbol{z}_i|\boldsymbol{x}_i) \sim \mathcal{N}(\boldsymbol{z}_i; \boldsymbol{\mu}_i, \boldsymbol{S}_i)$. To align this distribution with the class component $\boldsymbol{\epsilon}_c$ of the prototype distribution, we minimize the Wasserstein distance between them.

In case of multivariate Gaussians, a closed-form solution of the 2-Wasserstein distance [16] between two distributions $\mathcal{P}$ and $\mathcal{Q}$ is given by:

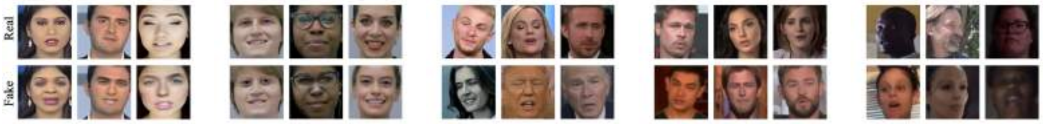
$$W_{\mathcal{P}\mathcal{Q}} = \left[\|\mu_{\mathcal{P}} - \mu_{\mathcal{Q}}\|_2^2 + Tr\big[\Sigma_{\mathcal{P}} + \Sigma_{\mathcal{Q}} - 2\big(\Sigma_{\mathcal{P}}^{\frac{1}{2}}\Sigma_{\mathcal{Q}}\Sigma_{\mathcal{P}}^{\frac{1}{2}}\big)^{\frac{1}{2}}\big]\right]^{\frac{1}{2}} \tag{5}$$

In our case, we assume that encoder $\boldsymbol{\theta}$ predicts diagonal covariance matrix, which allows us to simplify the distance:

$$d(q_i, \boldsymbol{\epsilon}_c) = W_{q_i\boldsymbol{\epsilon}_c} = \left[\|\boldsymbol{\mu}_i - \boldsymbol{m}_c\|_2^2 + \|\boldsymbol{S}_i^{\frac{1}{2}} - \boldsymbol{I}^{\frac{1}{2}}\|_{\text{Frob}}^2\right]^{\frac{1}{2}} \tag{6}$$

4 Datasets

With the rapid progress in synthetic media generation, a number of datasets focusing on facial forgery detection have been developed recently [40; 18; 54; 11; 24; 10]. Since our goal is to generalize across datasets as well as methods, we focus on those with sufficient diversity; see Fig. 2.



(a) FF++ [40] (b) Google DFD [11] (c) Dessa [31] (d) Celeb DF [54] (e) AIF [4]

Figure 2: Sample frames from the datasets used for evaluation of our experiments. The top row shows the frames from the real videos and bottom row shows the frames from the corresponding fake videos for paired datasets (FF++ and Google DFD) and randomly selected videos for unpaired datasets (Dessa, AIF, and Celeb DF). For FF++, the frames from DF manipulation method are shown.

FaceForensics++ (FF++) [40]: This is the one of the largest datasets in terms of variety and manipulations. The dataset contains over 1000 different videos (different identities), with each video manipulated by four different forgery methods, including DeepFakes (DF) [9], FaceSwap (FS) [25], Face2Face (F2F) [46], NeuralTextures (NT) [47]. DF are the most circulated videos on the internet while NT produce high-quality re-enactment videos; hence, we decided to focus on these two manipulation methods.

Google DeepFake Detection (Google DFD) [11]: This dataset consists of 3000 deepfake videos with 28 unique actors in different locations. To avoid data redundancy, we select all videos from a particular location (i.e., 228 fake and 28 real videos). In order to obtain an equal number of videos per class, we randomly pick one deepfake video per identity.

Celeb DF (v2) [54]: Celeb-DF (v2) is a very challenging dataset of real and synthesized deepfake videos of celebrities with similar visual quality with the online circulated deepfakes. It includes 590 original videos and 5639 corresponding DeepFake videos. Similar to Google DFD, we randomly pick 590 deepfake videos and use all 590 real videos for a fair comparison.

Dessa In-the-Wild [31]: The dataset consists of real and fake videos (84 each) of celebrities and politicians created with very high-quality deepfake generation methods.

AIF In-the-Wild [4]: The AI Foundation (AIF) dataset is a new video dataset which we introduce[1]. It consists of ∼110 unpaired real and deepfake videos. This is a relatively difficult dataset, with videos captured in low-quality lighting conditions and extreme blurring in some cases; however, this makes it particular interesting as a real-world test case.

5 Results

5.1 Experimental Settings

All of our experiments are run on an Nvidia GeForce GTX 1080 TI and all models use an ILSVRC 2012-pretrained ResNet-18 [17] backbone. Our models are implemented in Pytorch trained with an Adam optimizer (default parameters) using a batch-size of 128. Given that we want to study the transfer results of different loss formulations, we did not focus on hyper-parameter optimization of individual models.

All videos are converted to individual frames as described in Figure 3, and we report accuracies on a per frame basis. Unless otherwise stated, we apply our proposed spatial mixup augmentation followed by standard flipping to all baselines as well as our method; for fine-tuning it is applied with source images corresponding to the same class.

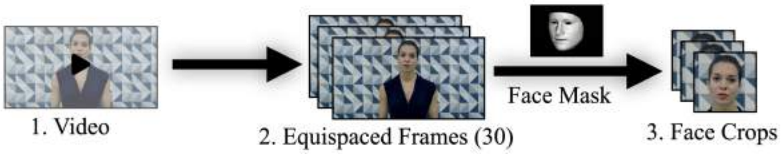


Figure 3: Video pre-processing: We extract 30 frames from every video spaced at equal intervals and automatically crop each frame (256 × 256 pixels). We ensure that at least 90% of the crop is covered by a face region. FF++ and Google DFD, already provide face masks; for the others, we use OpenCV DLib.

5.2 Transfer between Manipulation Methods

We investigate how well our method transfers between different manipulation methods, and compare against seven state-of-the-art domain-adaption/few-shot learning methods on the FF++ dataset [40]: Deep Domain Confusion (DDC) [48], Deep Correlation Alignment (CORAL) [44], Classification and Contrastive Semantic Alignment (CCSA) [32], d-SNE [52], ForensicTransfer (FT) [8], ProtoNets [43], and RelationNets [45]. The domain adaptation methods [48; 44; 32; 52] learn a common latent space for source and target domain and train a classifier over this joint embedding. Thus, the zero-shot scenario for these methods boils down to training a classifier on the source domain. Results are shown in Tab. 1 and Fig. 4.

Table 1: Classification accuracy for zero-shot transfer: the models are trained on one forgery method and then tested on data from a another (unseen) manipulation technique. We achieve comparable results within the same domain; however, our method generalizes significantly better.

	Trained with DF		**Trained with NT**		
Model	DF	NT	DF	NT	**Mean**
Classifier	95.92	57.80	68.75	91.30	78.44
Prototypical Nets [43]	**98.15**	60.58	69.57	**95.27**	80.89
Relation Nets [45]	98.11	57.15	68.70	91.13	78.77
FT [8]	91.29	62.86	75.50	83.50	78.28
DDT (ours)	98.01	**64.10**	**78.82**	92.05	**83.25**

[1] The AIF dataset is donated by the AI Foundation to the authors

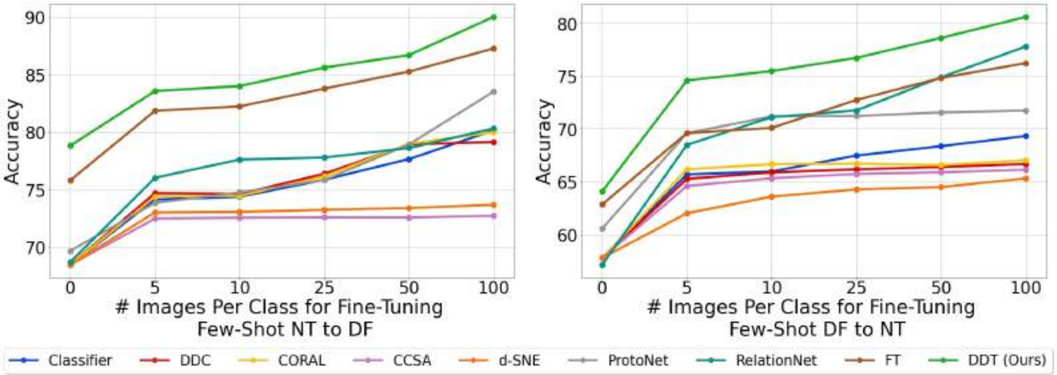


Figure 4: Few-shot manipulation transfer[2]: we pre-train a model with one manipulation method and fine-tune with a varying number of images of another manipulation approach. We outperform all the other methods, and achieve 90.01% accuracy for DF and 80.61% for NT when only using 100 images.

5.3 Transfer between Forgery Datasets

Following the previous section, we observe that NT and DF manipulations exhibit transfer. We further validate the claim by exploring the transfer on different datasets. Since there are no benchmark datasets to evaluate NT manipulation, we evaluate on a variety of DF manipulation datasets: Google DFD [11], Celeb DF [54], Dessa [31], and AIF [4]. Results for zero-shot transfer from FF++ to other datasets are shown in Tab. 2; few-shot results are visualized in Fig. 5.

Table 2: Zero-shot classification accuracy from FF++ to four other datasets: although there is a significant shift in domains (e.g., Google DFDC and Dessa contain mostly frontal faces in contrast to AIF which varies much more in pose and lightning), our method achieves transfer performance significantly higher than all baselines (last column).

Method	FF++	Google DFD	AIF	Dessa	Celeb DF	Mean
Classifier	91.02	79.87	54.26	63.45	65.32	70.78
Prototypical Nets [43]	**97.16**	71.34	61.73	63.57	58.03	70.36
Relation Nets [45]	96.46	72.75	58.08	56.19	63.19	69.84
FT [8]	84.71	67.82	**62.94**	69.40	47.83	66.54
DDT (ours)	92.23	**81.21**	60.79	**74.28**	**68.83**	**75.47**

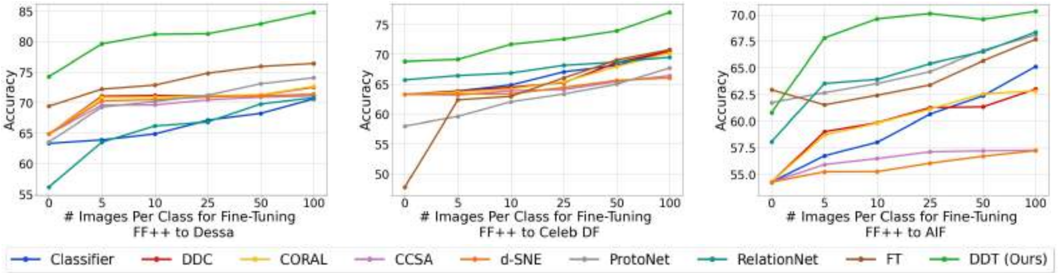


Figure 5: Few-shot transfer from FF++ [40] to three other dataset. Our method outperforms all the other approaches giving 84.80%, 77.07%, 70.32% accuracy for Dessa, Celeb DF, and AIF dataset, respectively after fine-tuning with 100 images.

[2] All the few-shot experiments are averaged over 10 runs.

5.4 Effect of Spatial Augmentation

All previous results, including baselines, use our proposed spatial augmentation. We now evaluate the effectiveness of the data augmentation strategy itself for the binary classifier and our method. Fig. 6 shows that our augmentation yields similar performance when tested on the same manipulation method; however, it shows significantly better generalization results on unseen forgery methods. Similar to zero-shot experiments, we evaluate the effect of spatial augmentation for few-shot learning; see Fig. 7. To this end, we mix images from the target domain with the images from the source domain. Every time we randomly pick a different image of the same class from the source domain to mix with the images of the target domain. In contrast to zero-shot self-manipulation results, where we did not see a significant effect on the same manipulation method, we now notice a small improvement using spatial augmentation. For instance, for the NT to DF manipulation transfer task, after fine-tuning with 100 images accuracy improves by 3.81% (76.33% to 80.14%) for the binary classifier and 2.3% (87.71% to 90.01%) for our approach.

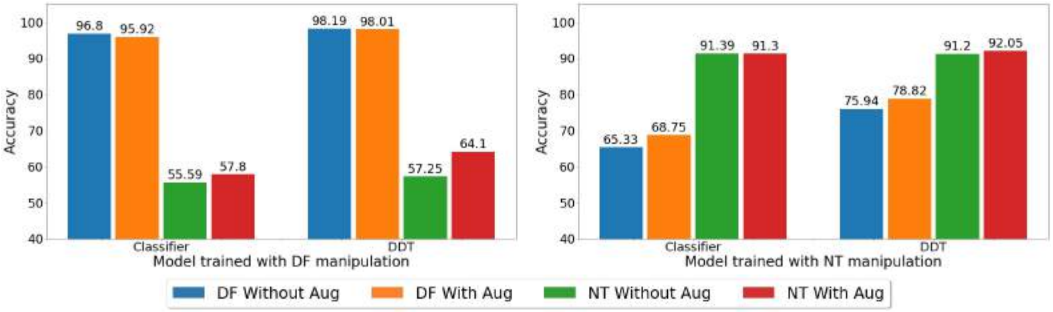


Figure 6: Effect of our proposed spatial augmentation for manipulation transfer: applying augmentation during training increases the zero-shot accuracy for both the methods and both experiments: (1) NT to DF (75.94% to 78.82% for DDT) (2) DF to NT (57.25% to 64.10% for DDT)).

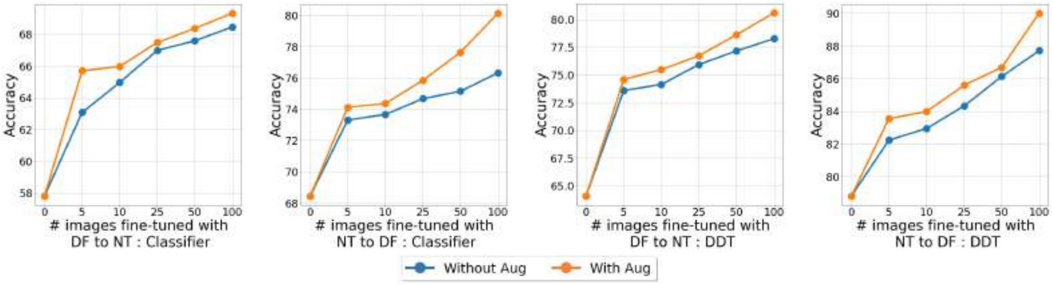


Figure 7: Effect of applying spatial augmentation during few-shot manipulation transfer with a binary classifier and our method. Blue and orange colors indicate fine-tuning w/o and w/ spatial augmentation, respectively. We observe a small improvement during fine-tuning across all experiments.

6 Conclusion

We have presented Deep Distribution Transfer, a new method for generalized zero and few-shot transfer learning for facial forgery detection. The core idea of our method is a new distribution-based loss formulation that can be efficiently trained to bridge the gap between domains of different facial forgery methods or unseen datasets. In a series of experiments, we show that the proposed method transfers well between widely-used face forgery data datasets while outperforming state-of-the-art baselines by a significant margin in both zero-shot and few-shot settings. For instance, we achieve a 4.88% higher detection accuracy for zero-shot and 8.38% for the few-shot case transferred from the FaceForensics++ to Dessa dataset. Overall, we believe that our generalized forensics transfer method is an important stepping stone towards making automated media forensics viable in practical settings, given that we ultimately need to have a reliable detector which is capable of handling previously unseen fakes and new data sources.

Broader Impact

The rapid development of facial manipulation methods such as deepfakes has become a severe issue in the context of social media and online video platforms, hence making the research on media forensics a critical research area within the machine learning community. Although there have been works focusing on automated detection methods with neural networks, such methods train and test within the same domain of data and methods. In this work, we directly address this limitation, and aim to generalize across forgery techniques and datasets with a new zero-shot and few-shot formulation. We believe that this transfer is critical towards making learned forgery detection methods practical, given that new fake methods appear at a rapid rate and it is unrealistic to expect to have access to a large training corpus for each new method. Ultimately, we hope that our method is a first stepping stone towards automated forgery detection on in-the-wild videos, for instance on Youtube, Facebook, or Twitter, rather than focusing on individual (potentially biased) academic datasets. In addition, we hope that the release of our code and trained models can be already of practical use to fact checkers as well as other researchers building upon on our work.

Acknowledgements

We would like to thank Andreas Rössler for helpful discussions and Davide Cozzolino for providing the Forensic Transfer [8] source code. We gratefully acknowledge the support of this research by the AI Foundation, a TUM-IAS Rudolf Mößbauer Fellowship, the ERC Starting Grant Scan2CAD (804724), and Google Faculty Award.

References

[1] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Mesonet: a compact facial video forgery detection network, Dec 2018. URL http://dx.doi.org/10.1109/WIFS.2018.8630761.

[2] S. Agarwal and H. Farid. Photo forensics from jpeg dimples. In *2017 IEEE Workshop on Information Forensics and Security (WIFS)*, pages 1–6, 2017.

[3] Shruti Agarwal, Hany Farid, Yuming Gu, Mingming He, Koki Nagano, and Hao Li. Protecting World Leaders Against Deep Fakes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, page 8, Long Beach, CA, June 2019. IEEE.

[4] AI Foundation. Ai foundation, 2017. URL https://aifoundation.com/.

[5] Christoph Bregler, Michele Covell, and Malcolm Slaney. Video rewrite: Driving visual speech with audio. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, page 353–360, USA, 1997. ACM Press/Addison-Wesley Publishing Co. ISBN 0897918967. doi: 10.1145/258734.258880. URL https://doi.org/10.1145/258734.258880.

[6] Da Chen, Yuefeng Chen, Yuhong Li, Feng Mao, Yuan He, and Hui Xue. Self-supervised learning for few-shot image classification, 2019.

[7] 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, 2017.

[8] Davide Cozzolino, Justus Thies, Andreas Rössler, Christian Riess, Matthias Niessner, and Luisa Verdoliva. Forensictransfer: Weakly-supervised domain adaptation for forgery detection, 2018.

[9] DeepFakes GitHub. Deepfakes, 2017. URL https://github.com/deepfakes/faceswap.

[10] Brian Dolhansky, Russ Howes, Ben Pflaum, Nicole Baram, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) preview dataset, 2019.

[11] Nicholas Dufour, Andrew Gully, Per Karlsson, Alexey Victor Vorbyov, Thomas Leung, Jeremiah Childs, and Christoph Bregler. Deepfakes detection dataset by google & jigsaw, 2019.

[12] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva. Image forgery localization via fine-grained analysis of cfa artifacts. *IEEE Transactions on Information Forensics and Security*, 7(5): 1566–1577, 2012.

[13] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks, 2017.

[14] Victor Garcia and Joan Bruna. Few-shot learning with graph neural networks, 2017.

[15] Pablo Garrido, Levi Valgaerts, Ole Rehmsen, Thorsten Thormaehlen, Patrick Perez, and Christian Theobalt. Automatic face reenactment. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Jun 2014. doi: 10.1109/cvpr.2014.537. URL `http://dx.doi.org/10.1109/CVPR.2014.537`.

[16] Clark R. Givens and Rae Michael Shortt. A class of wasserstein metrics for probability distributions. *Michigan Math. J.*, 31(2):231–240, 1984. doi: 10.1307/mmj/1029003026. URL `https://doi.org/10.1307/mmj/1029003026`.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016. doi: 10.1109/cvpr.2016.90. URL `http://dx.doi.org/10.1109/CVPR.2016.90`.

[18] Liming Jiang, Ren Li, Wayne Wu, Chen Qian, and Chen Change Loy. Deeperforensics-1.0: A large-scale dataset for real-world face forgery detection. In *CVPR*, 2020.

[19] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation, 2017.

[20] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2018.

[21] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan, 2019.

[22] Hyeongwoo Kim, Christian Theobalt, Pablo Carrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Niessner, Patrick Pérez, Christian Richardt, and Michael Zollhöfer. Deep video portraits. *ACM Transactions on Graphics*, 37(4):1–14, Jul 2018. ISSN 0730-0301. doi: 10.1145/3197517.3201283. URL `http://dx.doi.org/10.1145/3197517.3201283`.

[23] Junsik Kim, Tae-Hyun Oh, Seokju Lee, Fei Pan, and In So Kweon. Variational prototyping-encoder: One-shot learning with prototypical images, 2019.

[24] Pavel Korshunov and Sebastien Marcel. Deepfakes: a new threat to face recognition? assessment and detection, 2018.

[25] Marek Kowalski. Faceswap github, 2016. URL `https://github.com/MarekKowalski/FaceSwap/`.

[26] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[27] Wenbin Li, Lei Wang, Jinglin Xu, Jing Huo, Yang Gao, and Jiebo Luo. Revisiting local descriptor based image-to-class measure for few-shot learning, 2019.

[28] Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts, 2018.

[29] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. Learning to propagate labels: Transductive propagation network for few-shot learning, 2018.

[30] Siwei Lyu, Xunyu Pan, and Xing Zhang. Exposing region splicing forgeries with blind local noise estimation. *International Journal of Computer Vision*, 110:202–221, 11 2013. doi: 10.1007/s11263-013-0688-y.

[31] Rayhane Mama and Sam Shi. Dessa deepfake detection, 2019. URL `https://github.com/dessa-research/DeepFake-Detection`.

[32] Saeid Motiian, Marco Piccirilli, Donald A. Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. doi: 10.1109/iccv.2017.609. URL `http://dx.doi.org/10.1109/ICCV.2017.609`.

[33] Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen. Capsule-forensics: Using capsule networks to detect forged images and videos. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019. doi: 10.1109/icassp.2019.8682602. URL `http://dx.doi.org/10.1109/ICASSP.2019.8682602`.

[34] Yuval Nirkin, Yosi Keller, and Tal Hassner. Fsgan: Subject agnostic face swapping and reenactment, 2019.

[35] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge, 2017.

[36] Ivan Petrov, Daiheng Gao, Nikolay Chervoniy, Kunlin Liu, Sugasa Marangonda, Chris Umé, Mr. Dpfks, Carl Shift Facenheim, Luis RP, Jian Jiang, Sheng Zhang, Pingyu Wu, Bo Zhou, and Weiming Zhang. Deepfacelab: A simple, flexible and extensible face swapping framework, 2020.

[37] Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan Yuille. Few-shot image recognition by predicting parameters from activations, 2017.

[38] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017.

[39] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization, 2018.

[40] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images, 2019.

[41] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains, 09 2010.

[42] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing, 2019.

[43] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning, 2017.

[44] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. *Computer Vision – ECCV 2016 Workshops*, page 443–450, 2016. ISSN 1611-3349. doi: 10.1007/978-3-319-49409-8_35. URL `http://dx.doi.org/10.1007/978-3-319-49409-8_35`.

[45] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning, 2017.

[46] 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, December 2018. ISSN 0001-0782. doi: 10.1145/3292039. URL `http://doi.acm.org/10.1145/3292039`.

[47] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics 2019 (TOG)*, 2019.

[48] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance, 2014.

[49] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning, 2016.

[50] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[51] Yandong Wen, Rita Singh, and Bhiksha Raj. Reconstructing faces from voices, 2019.

[52] Xiang Xu, Xiong Zhou, Ragav Venkatesan, Gurumurthy Swaminathan, and Orchid Majumder. d-sne: Domain adaptation using stochastic neighborhood embedding. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019. doi: 10.1109/cvpr.2019.00260. URL `http://dx.doi.org/10.1109/CVPR.2019.00260`.

[53] Xin Yang, Yuezun Li, and Siwei Lyu. Exposing deep fakes using inconsistent head poses. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019. doi: 10.1109/icassp.2019.8683164. URL `http://dx.doi.org/10.1109/ICASSP.2019.8683164`.

[54] Pu Sun Honggang Qi Yuezun Li, Xin Yang and Siwei Lyu. Celeb-df: A large-scale challenging dataset for deepfake forensics. In *IEEE Conference on Computer Vision and Patten Recognition (CVPR)*, 2020.

[55] Peng Zhou, Xintong Han, Vlad I. Morariu, and Larry S. Davis. Two-stream neural networks for tampered face detection, Jul 2017. URL `http://dx.doi.org/10.1109/CVPRW.2017.229`.

A Datasets

A.1 Dataset Statistics

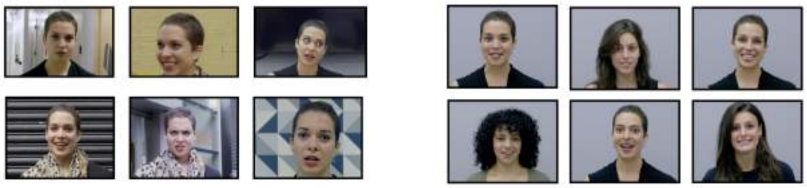
We evaluate our zero/few-shot transfer approach on five different forgery detection benchmark and in-the-wild video datasets: FaceForensics++ [40], Google DFD [11], Celeb DF [54], Dessa [31], and AIF [4] dataset. The exact details of train-test split used for our experiments are listed in Tab. 3.

Table 3: Dataset statistics showing videos per class. For FF++ and Dessa, we use the already provided split; for the other datasets, we provide our own train-test split. Note that for Google DFD, we have very few real videos; hence, we only evaluate zero-shot performance for this dataset. For all other datasets, we explore both zero-shot and few-shot transfer. All results in the main paper are reported on these train-test splits.

Mode	**FF++** [40]	**Google DFD** [11]	**Celeb DF** [54]	**Dessa** [31]	**AIF** [4]
Train	720	-	500	70	12
Val	140	-	-	-	-
Test	140	28	90	14	99

A.2 Detailed of Google DFD Selection Strategy

In order to avoid redundant videos in Google DFD, we select 228 fake and 28 real videos following the strategy described in Fig. 8.



(a) Same actor but different location

(b) Different actors but same location.

Figure 8: Google DFD dataset selection strategy: (a) we have multiple videos of the same actor (recorded in different locations) with deepfakes are created for all the real videos; i.e., we have multiple videos with the same facial identity in different locations. (b) We pick one location (podium in our case) and use the videos (real and fake) for this particular location.

B Details on Hyperparameters

We use the ILSVRC 2012-pretrained ResNet-18 [17] network as a backbone for our experiments to obtain a 256-dimensional embedding vector. We then apply one or more dense layers on top of this embedding based on what transfer learning approach we are training with. For ForensicTransfer (FT) [8], we used the original architecture and setup as proposed in the original paper.

For pre-training (zero-shot) experiments, we use a learning rate of 1e-3 with a decay when validation loss plateaus for five consecutive epochs, early stopping with the patience of 10 successive epochs on the validation loss. For fine-tuning (few-shot) and domain transfer experiments, we use a learning rate of 1e-5 without any decay, early stopping with the patience of 30 consecutive epochs on the training loss since no validation set is available. During fine-tuning, we use the same seed across runs and for each method to ensure consistency.

C Additional Experiments

C.1 Effect of Adding More Manipulation Methods

In this section, we evaluate how well a model trained with DF or NT or both manipulations from FF++ dataset [40] is able to detect fake videos from a different and unseen dataset.

Table 4: Zero-shot transfer comparison from different manipulation methods. A model pre-trained with different manipulation methods from FF++ [40] dataset is evaluated on four different datasets: Google DFD, AIF, Dessa, and Celeb DF. We experiment with three combination of manipulations, models trained with DF only, NT only and both manipulations.

Method	Manipulation	Google DFD	AIF	Dessa	Celeb DF	Mean
Classifier	DF	79.10	53.78	55.35	63.37	62.90
	NT	68.01	53.01	58.80	52.90	58.18
	DF+NT	**79.87**	**54.26**	**63.45**	**65.32**	**65.72**
DDT (ours)	DF	**81.94**	53.06	59.28	67.98	65.56
	NT	75.38	55.47	59.52	66.99	64.34
	DF+NT	81.22	**60.79**	**74.28**	**68.83**	**71.28**

Tab. 4 shows that a model pre-trained with both DF and NT manipulations boosts accuracy by 0.85%(from 67.98% to 68.83%) for Celeb DF, 5.32% (from 55.47% to 60.79%) for AIF and a large 14.76% (from 59.52% to 74.28%) for Dessa dataset, for our proposed method. Similarly, for the naive classifier, the accuracy is boosted by 1.95%(from 63.37% to 65.32%) for Celeb DF, 0.48% (from 53.78% to 54.26%) for AIF and 4.65% (from 58.80% to 63.45%) for Dessa dataset. An important observation is that adding more variety in manipulations methods for pre-training yields to better generalizability; i.e., we obtain better detection results on different, previously unseen, datasets. This is very promising and shows a path forward to the future of forgerey detection when more datasets will become publicily available, which would allow us to obtain even stronger models with our method. At the same time, there is no performance degradation on the Google DFD dataset when we move from DF pre-trained model to DF+NT pretrained model. Hence, adding more variety in terms of manipulation methods for pre-training does not come at the cost of already included target forgery techniques.

C.2 Effect of Applying Spatial Augmentation on Other Datasets

In this section, we study the effect of applying our proposed augmentation on other datasets. We evaluate the genralization results on two challenging AIF and Dessa datasets.

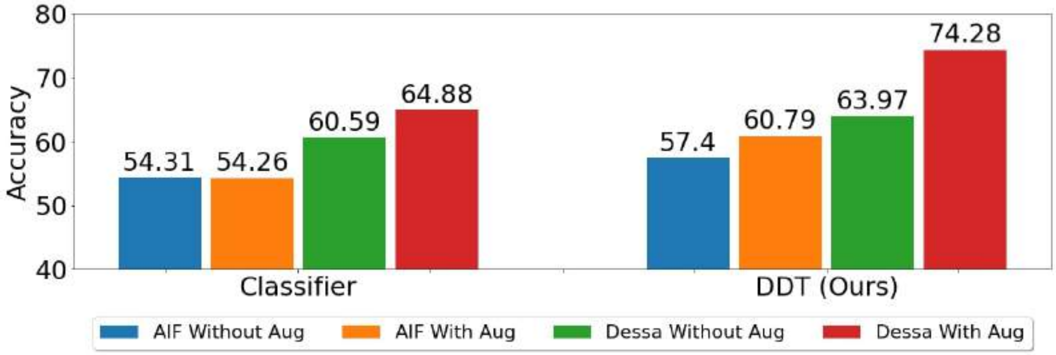


Figure 9: Zero-shot transfer comparison from FF++ to other datasets for a naive classifier and our method. All the models are pre-trained on (NT + DF) manipulation both, with and without spatial mixup augmentation.

Fig. 9 shows that applying spatial mixup augmentation during pre-training generalizes better to other datasets as well. For the Dessa dataset, we observe 4.29% (from 60.59% to 64.88%) improvement for Classifier and a massive 10.31% improvement (from 63.97% to 74.28%) for DDT.

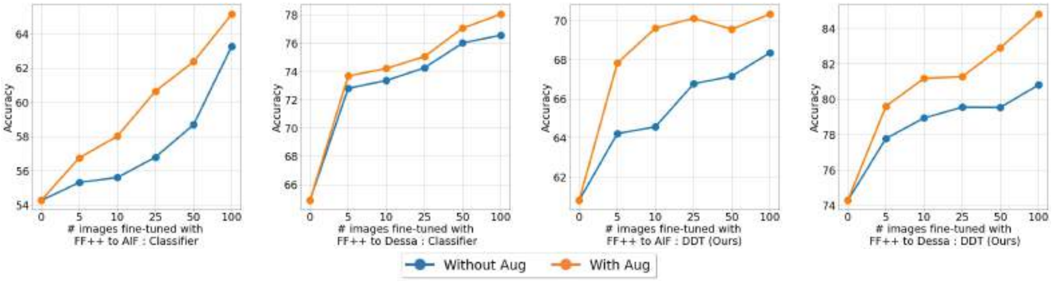


Figure 10: Effect of applying our spatial augmentation during few-shot transfer to other datasets for a binary classifier and our method. During fine-tuning, we again observe consistent improvements across all experiments.

D Result Tables

In this section, we document the line graphs of all the few shot experiments.

Table 5: Few-shot manipulation transfer from NT to DF.

Few-Shot Images	Classifier	DDC	Deep CORAL	CCSA	d-SNE	Prototypical Nets	Relation Nets	Forensic Transfer	**DDT (Ours)**
0 image	68.46	68.46	68.46	68.46	68.46	69.67	68.70	75.80	**78.82**
5 images	74.13	74.69	74.40	72.47	73.0	73.84	76.02	81.85	**83.56**
10 images	74.37	74.62	74.44	72.55	73.05	74.75	77.60	82.23	**83.99**
25 images	75.85	76.37	76.15	72.58	73.23	75.83	77.78	83.77	**85.60**
50 images	77.64	78.92	78.97	72.57	73.39	78.92	78.60	85.25	**86.69**
100 images	80.14	79.13	79.98	72.71	73.67	83.52	80.31	87.26	**90.01**

Table 6: Few-shot manipulation transfer from DF to NT.

Few-Shot Images	Classifier	DDC	Deep CORAL	CCSA	d-SNE	Prototypical Nets	Relation Nets	Forensic Transfer	**DDT (Ours)**
0 image	57.80	57.80	57.80	57.80	57.80	60.58	57.15	62.86	**64.10**
5 images	65.70	65.28	66.18	64.60	62.02	69.61	68.49	69.61	**74.59**
10 images	65.97	65.89	66.66	65.34	63.60	71.19	71.12	70.10	**75.48**
25 images	67.48	66.17	66.71	65.74	64.28	71.23	71.76	72.75	**76.73**
50 images	68.37	66.40	66.57	65.90	64.50	71.57	74.89	74.83	**78.62**
100 images	69.33	66.68	67.04	66.13	65.32	71.74	77.81	76.23	**80.61**

Table 7: Few-shot transfer from FF++ to Dessa

Few-Shot **Images**	Classifier	DDC	Deep CORAL	CCSA	d-SNE	Prototypical Nets	Relation Nets	Forensic Transfer	**DDT** **(Ours)**
0 image	64.88	64.88	64.88	64.88	64.88	63.57	56.19	69.40	**74.28**
5 images	73.67	71.04	70.85	69.57	70.27	69.25	63.52	72.20	**79.60**
10 images	74.20	71.15	70.91	69.60	70.48	70.16	66.19	72.86	**81.18**
25 images	75.04	71.08	71.08	70.43	70.92	71.21	66.82	74.83	**81.27**
50 images	77.05	71.24	71.28	70.88	71.09	73.09	69.76	75.94	**82.91**
100 images	78.03	72.54	72.65	71.08	71.37	74.07	70.78	76.42	**84.80**

Table 8: Few-shot transfer from FF++ to Celeb DF

Few-Shot **Images**	Classifier	DDC	Deep CORAL	CCSA	d-SNE	Prototypical Nets	Relation Nets	Forensic Transfer	**DDT** **(Ours)**
0 image	63.32	63.32	63.32	63.32	63.32	58.03	65.75	47.83	**68.83**
5 images	63.89	63.82	63.70	63.20	63.38	59.61	66.47	62.39	**69.14**
10 images	64.88	64.49	64.23	63.92	63.40	62.07	66.90	62.98	**71.68**
25 images	67.10	65.30	65.43	64.12	64.47	63.37	68.20	66.09	**72.59**
50 images	68.21	68.64	68.01	65.45	65.66	65.03	68.72	69.07	**73.94**
100 images	70.60	70.54	70.25	66.47	66.09	67.77	69.51	70.78	**77.07**

Table 9: Few-shot transfer from FF++ to AIF

Few-Shot **Images**	Classifier	DDC	Deep CORAL	CCSA	d-SNE	Prototypical Nets	Relation Nets	Forensic Transfer	**DDT** **(Ours)**
0 image	54.26	54.26	54.26	54.26	54.26	61.73	58.08	**62.94**	60.79
5 images	56.75	59.02	58.70	55.92	55.24	62.67	63.54	61.52	**67.82**
10 images	58.02	59.85	59.82	56.48	55.25	63.53	63.92	62.43	**69.61**
25 images	60.65	61.26	61.12	57.12	56.04	64.65	65.40	63.40	**70.11**
50 images	62.37	61.33	62.56	57.21	56.70	66.64	66.51	65.66	**69.56**
100 images	65.14	63.011	62.84	57.26	57.23	68.08	68.35	67.68	**70.32**

Table 10: Few Shot Transfer with and without spatial augmentation for DF to NT within the FF++ dataset.

Few-Shot	**Classifier**		**DDT (Ours)**	
Images	Without Aug	With Aug	Without Aug	With Aug
0 image	57.80	57.80	64.10	64.10
5 images	63.07	65.70	73.60	74.59
10 images	64.96	65.97	74.14	75.48
25 image	66.98	67.48	75.94	76.73
50 image	67.59	68.37	77.17	78.62
100 image	68.46	69.33	78.27	80.61

Table 11: Few-shot transfer with and without spatial augmentation for NT to DF within the FF++ dataset.

Few-Shot	**Classifier**		**DDT (Ours)**	
Images	Without Aug	With Aug	Without Aug	With Aug
0 image	68.46	68.46	78.82	78.82
5 images	73.32	74.13	82.24	83.56
10 images	73.67	74.37	82.95	83.99
25 image	74.68	75.85	84.33	85.60
50 image	75.15	77.64	86.14	86.69
100 image	76.33	80.14	87.71	90.01

Table 12: Few-shot transfer with and without spatial augmentation for FF++ to AIF.

Few-Shot	**Classifier**		**DDT (Ours)**	
Images	Without Aug	With Aug	Without Aug	With Aug
0 image	54.26	54.26	60.79	60.79
5 images	55.31	56.75	64.21	67.82
10 images	55.60	58.02	64.56	69.61
25 image	56.79	60.65	66.76	70.11
50 image	58.69	62.37	67.15	69.56
100 image	63.26	65.14	68.35	70.32

Table 13: Few-shot transfer with and without spatial augmentation for FF++ to Dessa.

Few-Shot	**Classifier**		**DDT (Ours)**	
Images	Without Aug	With Aug	Without Aug	With Aug
0 image	64.88	64.88	74.28	74.28
5 images	72.80	73.67	77.78	79.60
10 images	73.35	74.20	78.94	81.18
25 image	74.25	75.04	79.55	81.27
50 image	76.00	77.05	79.54	82.91
100 image	76.55	78.03	80.81	84.80