27^{th} Computer Vision Winter Workshop
Terme Olimia, Slovenia, February 14–16, 2024

Cross-Dataset Deepfake Detection: Evaluating the Generalization Capabilities of Modern DeepFake Detectors

Marko Brodarič, Vitomir Štruc
University of Ljubljana,
Faculty of Electrical Engineering,
Tržaška cesta 25, 1000 Ljubljana
marko.brodaric@fe.uni-lj.si

Peter Peer
University of Ljubljana,
Faculty of Computer and Information Science,
Večna pot 113, 1000 Ljubljana
peter.peer@fri.uni-lj.si

Abstract. *Due to the recent advances in generative deep learning, numerous techniques have been proposed in the literature that allow for the creation of so-called deepfakes, i.e., forged facial images commonly used for malicious purposes. These developments have triggered a need for effective deepfake detectors, capable of identifying forged and manipulated imagery as robustly as possible. While a considerable number of detection techniques has been proposed over the years, generalization across a wide spectrum of deepfake-generation techniques still remains an open problem. In this paper, we study a representative set of deepfake generation methods and analyze their performance in a cross-dataset setting with the goal of better understanding the reasons behind the observed generalization performance. To this end, we conduct a comprehensive analysis on the FaceForensics++ dataset and adopt Gradient-weighted Class Activation Mappings (Grad-CAM) to provide insights into the behavior of the evaluated detectors. Since a new class of deepfake generation techniques based on diffusion models recently appeared in the literature, we introduce a new subset of the FaceForensics++ dataset with diffusion-based deepfake and include it in our analysis. The results of our experiments show that most detectors overfit to the specific image artifacts induced by a given deepfake-generation model and mostly focus on local image areas where such artifacts can be expected. Conversely, good generalization appears to be correlated with class activations that cover a broad spatial area and hence capture different image artifacts that appear in various part of the facial region.*

1. Introduction

With the advances in generative deep neural networks, there has been a surge in methods capable of synthesizing forged and/or manipulated images and videos. The most widespread synthesis methods are based on Generative Adversarial Networks (GANs) [6, 11, 21], and in recent years, solutions utilizing the concept of denoising diffusion [8, 13, 40]. Human faces have always been one of the most popular targets for such synthesis and manipulation techniques, as this allows for the design of numerous practical applications, ranging from applications in the entertainment industry (e.g., movies and smartphone applications), security systems, privacy-enhancing solutions and many more [22]. However, due to the high level of realism ensured by these methods, they can also be employed for malicious purposes, such as creating fake news or falsifying evidence. All of this has prompted the development of so-called deepfake detectors to alleviate this threat.

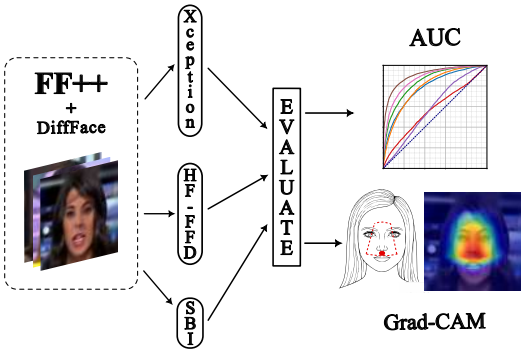


Figure 1: **We evaluate the performance of three conceptually distinct deepfake detection methods in a cross-dataset setup** on the FaceForensics++ database and investigate the reasons for the different generalization capabilities using Gradient-weighted Class Activation Mappings (Grad-CAM). To facilitate the analysis, we also introduce a dataset of deepfakes, generated with a diffusion-based generator.

Among the first detectors developed were techniques that work as binary classifiers. Such discriminative detectors are commonly trained on a dataset to perform classification between images representing original/pristine, unaltered images and images that have been manipulated using one of the existing deepfake generation methods [1, 3]. A limitation of the discriminatively-trained approach is that the errors made by the synthesis method in generating deepfakes are quite specific to that method. This results in poor generalization of the detector, which learns to classify a specific type of deepfake. In real-life deployment scenarios where we lack information about how the forgery was created, it is crucial for the detector to perform well regardless of the type of deepfake encountered. Some solutions have addressed these problems by introducing a specific pipeline before the classifier that extracts additional information from the given image, either by considering multiple modalities [20] or by manipulating the image [27, 39]. The latter proves to be one of the more effective approaches to improving generalization. The idea behind these methods is that they generate so-called pseudo deepfakes and use them as an extension of the training dataset, or they learn exclusively on them. Images can be augmented in various ways, which determines the types of artifacts that are injected into the training set of the detector. However, even these methods can only improve generalization to a certain extent, as they are fundamentally discriminative. In this domain, approaches have also been proposed that use only one class for training [12, 15]. These methods learn only from samples of unaltered images, defining in a way what a normal image is, and anything deviating from it is marked as an anomaly—indicating a potential deepfake. These methods are expected to be robust to different types of deepfakes, as they do not encounter any real deepfake samples during training.

In this paper, we aim *to explore the generalization capabilities of existing deepfake detectors in cross-dataset experiments*, where the term cross-dataset refers to the fact that the detectors are tested on deepfake types that are distinct from those used for training. Additionally, we are *interested in the performance* of existing detectors with the more *recent diffusion-based deepfake generation techniques*, that have not been studied widely yet in the literature. Finally, our goal is also to *understand the causes behind the observed performances*. To this end, we conduct a comprehensive cross-dataset evaluation of various types of detectors on deepfakes from the FaceForensics++ dataset [25] and study the results quantitatively as well as qualitatively through Gradient-weighted Class Activation Mappings (Grad-CAM) [26].

2. Related work

In this section, we present a brief overview of relevant works on deepfake detection. For a more comprehensive review of existing detectors, the reader is referred to some of the excellent surveys on this topic available in the literature [22, 23, 36].

Early Detectors. Early deepfake detectors primarily relied on the identification of known artifacts, introduced into the forged images by the deepfake generation techniques. As a result, this group of detectors used conventional (hand-crafted) descriptors and classifiers to detect blending signs [2, 38], deviations of the face from the surrounding background (e.g., incorrect lighting) [28], identification of face warping artifacts [19], and even methods that observe the broader context of a video, such as detecting unusual eye blinking patterns [18] or observing lip synchronization and corresponding speech [14]. Such detectors provided promising initial results, but were limited in their performance due to their focus on explicit (human-defined) image artifacts, induced by the deepfake-generation models.

Discriminative Detectors. To mitigate the dependence on manual modeling of image artifacts, a more recent group of detectors approached deepfake detection from a machine learning perspective and formulated the problem as a binary classification task. Solutions from this group, commonly learn a discriminative model, e.g., a convolutional neural network (CNN), on a dataset of real and fake images, and during the training process, simultaneously learn relevant features for detection. It turns out that even standard (off-the-shelf) CNN architectures already perform better in addressing deepfake detection than the early hand-crafted techniques discussed above, while more specialized solutions further improve on these results. In [3], for example, the authors introduced Xception, a CNN model that with minor modifications was demonstrated to be highly effective for deepfake detection [24]. Tariq *et al.* [33] showed that vanilla CNN detectors, based on Xception [3] or DenseNet [9] backbones, perform poorly with low-resolution deepfakes. To address this issue, they pro-

posed an ensemble of three Shallow Convolutional Networks with different layer configurations, effectively handling various input image resolutions. Similarly, Afchar *et al.* [1], argued that microscopic image analysis based on image noise is not suitable for compressed images, where the noise induced by the deepfake generation process is strongly degraded, and similarly, that the analysis of high-level semantics is also unsuitable due to the subtle appearance differences between real and fake images. Therefore, an intermediate approach was proposed, where a neural network classifies images based on mesoscopic features, a mid-level image representation.

Although discriminative detectors perform well in detecting forgeries, when they are tested with the same type of deepfakes that was also used for training, their performance tends to deteriorate, when applied to deepfakes created using a previously unseen method. This generalization issue is also generally considered as one of the main problems of modern deepfake detectors, and the causes of the poor generalization are still poorly understood.

Beyond Discriminative Detectors. The problem of generalization was addressed in [20] by introducing a *dedicated feature extractor* that incorporated *specific domain-knowledge* before the classifier. The feature extractor infers task-specific and information-rich features at multiple scales from the input image, combining them into a discriminative representation that is then fed to a classifier. In [4], the authors followed a similar idea and proposed the Hierarchical Memory Network to decide whether an image represents a deepfake or not. The proposed network considers both the current facial content to be classified as well as previously seen faces. Facial features are extracted using a pretrained neural network, consisting of a bidirectional GRU (Gated Recurrent Unit) and an attention mechanism. The resulting output is then compared to previously seen faces to make a decision on whether the input face is a deepfake or not.

One of the more effective methods for improving the generalization of deepfake detectors is the synthesis of forged images, which are then used together with real/pristine face images to train discriminative detection models. These so-called *pseudo-deepfake methods* are in essence learned from real data only and never observe a real deepfake image. Instead, they simulate deepfake artifacts through various augmentation and synthesis strategies, leading to highly effective detection models. Li *et al*. [17], for example, proposed the Face X-ray method, which focuses on identifying image artifacts resulting from the blending process. In the learning stage, real faces are initially blended together to generate blended images, and a detector is then trained on these samples to distinguish between original and blended images. This idea was later extended in [27], where the authors synthesized training samples by blending a face back into its original frame. Because the same face is used as the target as source for swapping, the proposed self-blending process introduces very subtle artifacts from which a deepfake detector is learned, leading to very competitive detection performance.

Since the primary task of deepfake detectors is to distinguish forgeries of any kind from pristine images, solutions have also been proposed that approach the problem within a *one-class anomaly detection setting*. In [12], Khalid *et al.* proposed the OC-FakeDect method that is based on a One-Class Variational Autoencoder. Here, the input images are classified based on the reconstruction score obtained through the encoder-decoder architecture. Similarly, in [15], a one-class method, called SeeABLE, was presented, where the model learns low-dimensional representations of synthetic local image perturbations. To detect forgeries, an anomaly score derived from a prototype matching procedure is used.

Our Contribution. While the evolution of deepfake detectors, discussed above, has led to obvious progress in detection performance and improvements in the generalization capabilities, the characteristics of these models that impact cross-deepfake detection performance are still underexplored. In the experimental section, we therefore study the behavior of a representative set of existing deepfake detectors in cross-dataset detection experiments and analyze class activation mappings to better understand, which image areas contribute to the detection decisions. Additionally, we also explore the performance of the detectors with a new class of deepfakes, generated with modern diffusion-based models. To the best of our knowledge, this issues has not yet been widely explored in the open literature.

3. Methodology

To facilitate the analysis, we select three conceptually distinct deepfake detectors: (i) a **discriminative model** based on the Xception architecture that learns to distinguish between real and forged images through a binary classification problem [24], (ii) the

High-Frequency Face Forgery Detection (HF-FFD) method [20] that aims to improve the generalization capabilities of discriminatively learned deepfake detectors by extracting **informative task-specific features**, and (iii) a **pseudo-deepfake detector** relying on Self-Blended Images (SBI) [27] that learns from pristine images only and simulates deepfake induced artifacts for the training process through a dedicated blending process. Details on the selected deepfake detectors are given in the following sections.

3.1. The Discriminative Xception-Based Detector

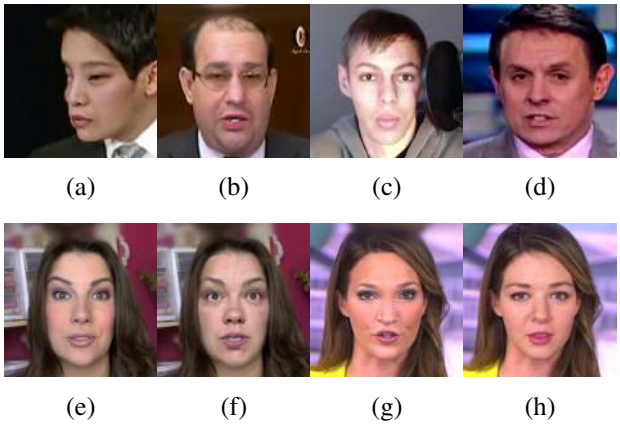
The Xception method conceptually originates from the family of Inception methods [10, 29–31]. Unlike traditional convolutional layers that learn filters in 3D space (two spatial dimensions and one channel dimension), processing both the spatial and cross-channel correlations with each convolutional kernel, the fundamental idea of Inception modules is to divide this process into multiple operations that independently handle the mapping of these correlations. Specifically, in Inception modules, cross-channel correlations are first computed using 1×1 convolutional filters, followed by all other correlations using 3×3 convolutions. If we simplify the module by omitting the average pooling tower and reformulate the architecture as one large 1×1 convolutional layer followed by 3×3 convolutions, we get a streamlined version of the Inception layer. Taking this idea to the extreme by mapping spatial correlations for each output channel, we get a module very similar to depthwise separable convolution. Xception is a convolutional neural network architecture that replaces Inception modules with depthwise separable convolution layers, assuming that mapping cross-channel correlations and spatial correlations in the feature maps of a convolutional neural network are completely decoupled. The proposed architecture consists of 36 convolutional layers structured into 14 modules, each with a linear residual connection (except the first and last). At the end, there is logistic regression and an optional fully-connected layer. The first detector used in this work uses the Xception model to learn a discriminative deepfake detector.

3.2. High-Frequency Face Forgery Detection

Luo *et al.* [20] identified that face manipulation procedures generally consist of two stages: fake face creation and face blending. Since only the facial part is altered in the image while the background remains the same, the blending stage disrupts the original data distribution, and this characteristic discrepancy can be utilized for forgery detection. As a result of this observation, the authors proposed a method that employs both RGB spatial features and high-frequency noises for detecting forgeries. The pipeline comprises three parts: the entry, middle, and exit flows. The input image is first converted into a residual image X_h using SRM filters [5]. The entry flow takes both the RGB image X and the residual image X_h, performing convolution on both to obtain feature maps F^1 and F_h^1. To extract more high-frequency information, an SRM followed by a 1×1 convolution is applied to F_h^1, resulting in $\tilde{F}_h^1$. This result is then added to F_h^1, and the operations are repeated. The output of the entry flow consists of feature maps of two modalities, where the high-frequency F_h carries much more information than the input X_h. The output spatial feature map F is element-wise multiplied with an attention map M obtained from the residual image as: $M = f_{att}(X_h)$, where f_{att} is an attention block, inspired by CBAM [37]. In the middle flow, feature maps of two modalities are fed into a dual cross-modality attention module (DCMA), which captures dependencies between low-frequency textures and high-frequency noises. Each input is divided into two components: a value, representing domain-specific information, and a key, measuring the correlation between these two domains. In the exit flow, high-level features of the two modalities are merged. Classifier training to distinguish between genuine and forged images can then be performed on these obtained features. In this work, we again use the Xception [3] model to learn a deepfake detector over the extracted features.

3.3. Self-Blended Images [27]

The third approach considered for our analysis [27], i.e, Self-Blended Images, falls into the category of detectors that address the generalization issue by generating synthetic forgeries, on which a discriminative detector is learned. Typically, these methods synthesize training samples by blending two distinct faces and generating artifacts based on the gap between source and target images. In contrast, this method performs blending of a slightly altered version of the same face, actively generating artifacts with selected transformations. The socaleld Self-Blended Images (SBIs) are generated in three steps. First, the source-target generator creates pseudo source and target images for blending. The



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 2: **Examples of images generated using DiffFace.** DiffFace produces convincing deepfakes that are almost indistinguishable from real images, e.g., see the pristine images in (e) and (g) and their deepfakes in (f) and (h), but also leads to failure cases in challenging scenarios, e.g., a profile view in (a), facial occlusions, e.g., a visible border around glasses in (b). Sometimes artifacts also remain in the images, e.g., shadows in (c) or hair segments in (d).

input image I is initially duplicated, and both images are augmented to introduce statistical inconsistencies (RGB and HSV color space values are randomly shifted, as well as brightness and contrast; the images are downsampled or upsampled). Blending boundaries in landmark mismatches are reproduced by resizing the source image, zero-padding, or center-cropping, and finally translating it. Pseudo-source and target images end up with the same size as the original image. In the next step, the mask generator creates a grayscale mask used for blending the previously generated images. This is done by having a landmark detector first determine parts of the face based on which a convex hull is calculated. To increase the diversity of the mask, the obtained shape is deformed with elastic deformation and then eroded or dilated. Lastly, the blending ratio of the source image is determined by multiplying the mask by a constant $r \in (0, 1]$. In the final step, the blending of the source image I_s and target image I_t is performed with the blending mask M to generate the self-blended image. With such synthetically generated samples, a binary classifier is then trained to distinguish between genuine images and deepfakes. Following [27], we also use EfficientNet-b4 [32] for this task.

4. Experiments and results

4.1. Datasets

For the experiments, we select the FaceForensics++ dataset [25], which is one of the most popular and challenging datasets publicly available for the development and testing of deepfake detectors in cross-deepfake type experiments. Additionally, to make the analysis more comprehensive, we generate two novel subsets of the FaceForensics++ dataset, one based on a recent GAN-based face swapping procedure, and one based on a diffusion-based model. These two subsets also represents one of the **tangible contributions** of this work. Below, we provide details on the FaceForensics++ dataset and the novel InsightFace and DiffFace subsets.

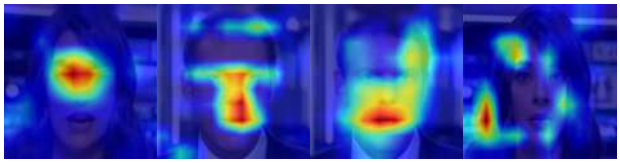
FaceForensics++. For the training and testing of models, we utilize the FaceForensics++ dataset [25], which comprises 1000 videos. These videos are divided into three groups: 720 for training, 140 for validation, and 140 for testing. The dataset is partitioned into several subsets that are generated using 5 distinct deepfake-generating methods: Deepfakes[1], Face2Face [35], FaceShifter [16], FaceSwap[1], and NeuralTextures [34]. These deepfakes are created using predefined target and source face pairs and are mostly based on methods relying on Generative Adversarial Networks (GANs). Additionally, each group includes authentic, unaltered videos. We augment the dataset with two additional subsets. The first uses the InsightFace [7] face swapping procedure, and the second the diffusion-based DiffFace approach from DiffFace [13]. Because deepfakes based on diffusion models have so far not been widely discussed in the literature and no relevant datasets are available in the literature, we discuss the generated DiffFace subset of FaceForensics++ (FF++) in a separate section below.

The DiffFace FF++ Subset. We structure the DiffFace Subset in the same way as all others from the FaceForensics++ collection: it consists of frames from 1000 videos, divided into training, validation, and test sets, with only every tenth frame processed for each recording. Forged images generated using the DiffFace approach are highly convincing and difficult to distinguish from authentic ones at first glance. In Figures 2e to 2h, we see that the generated deepfake can even look more convincing than the original images. However, the method yields poorer

[1] https://github.com/deepfakes/faceswap

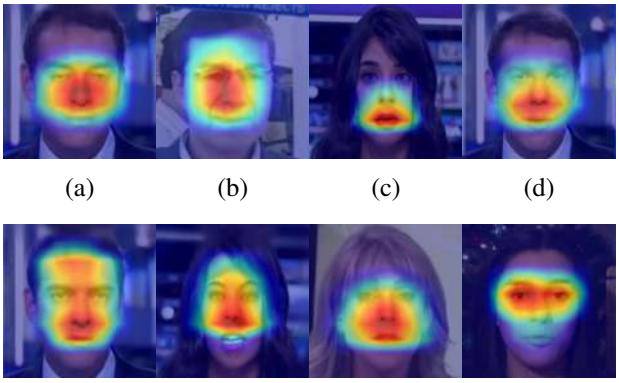
Train set	Test set - AUC						
	Deepfakes	**DiffFace**	**Face2Face**	**FaceShifter**	**FaceSwap**	**InsightFace**	**NeuralTextures**
Deepfakes	**0.9974**	0.7018	0.8844	0.5699	0.6434	0.6130	0.9174
DiffFace	0.6111	**0.9959**	0.5079	0.6128	0.5151	0.6072	0.5199
Face2Face	0.9420	0.6475	**0.9903**	0.6946	0.6562	0.5316	0.8106
FaceShifter	0.6533	0.9368	0.5197	**0.9969**	0.5161	0.6156	0.5696
FaceSwap	0.6647	0.6928	0.8608	0.5050	**0.9955**	0.5361	0.7730
InsightFace	0.6981	0.6473	0.5851	0.8027	0.5473	**0.9298**	0.6535
NeuralTextures	**0.9931**	0.6765	0.9497	0.7302	0.6847	0.5516	0.9862

Table 1: Performance of Xception trained on different databases in cross-dataset scenario.



(a) (b) (c) (d)

Figure 3: **Grad-CAM analysis of the last convolutional layer of the Xception network.** The model trained on Deepfakes (a), Face2Face (b), and NeuralTextures (c) databases typically activates in the regions around the eyes, mouth, and nose. The classifier trained on deepfakes from the DiffFace database (d) typically activates in a circular pattern.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4: **Illustration of Grad-CAM depicting the triggering regions of the last convolutional layer of the Xception network with an added feature-extracting pipeline**: focus on the root of the nose (Deepfakes (a)) and on the edge of the nose (InsightFace (b)), triangular area with the center on the mouth (DiffFace (c)), circular focus on the philtrum (Face2Face (d) and NeuralTextures (g)), hourglass shape (FaceShifter (e)), and truncated triangle (FaceSwaps (f)), focus on the eyes in genuine images (h). Best viewed in color.

results when faced with more challenging scenarios, such as under face orientations that cause the face to be partially visible (e.g., a profile view in Figure 2a) and various occlusions on the face (e.g., glasses in Figure 2b). As the process is of a sequential stochastic nature, artifacts such as shadows (in Figure 2c) or hair segments (in Figure 2d) are sometimes transferred to the output as well.

4.2. Performance metrics

Following standard evaluation methodology [12, 15, 23] we evaluated the performance of the selected methods based on the Area Under the Receiver Operating Characteristic Curve (AUC). We also conduct a qualitative analysis of the results, comparing the characteristics of images and Gradient-weighted Class Activation Mapping (Grad-CAM) heatmaps of samples where the methods are successful and those where they are not [26]. We use Grad-CAM as the primary tool for understanding the generalization capabilities of the tested detectors.

4.3. Results

Xception Results. For the evaluation, we trained the Xception model using deepfakes generated with one of the face forgery methods that constitute the FF++ dataset, and tested the model on the entire testing set to obtain insight about the method's performance detecting various types of deepfakes. The results are compiled in Table 1. It is evident that the method performs best on forgeries generated using the same method as used for the generation of training samples. Clearly, the detector overfits to the textural errors specific to the given deepfake generation method. Consequently, when applied to images manipulated using a different method, the detector's performance significantly decreases.

Additionally, we observe that the model exhibits significantly better generalization across the Deepfakes, Face2Face, and NeuralTextures databases compared to other types of deepfakes. These forgeries contain visually similar artifacts, e.g., blend-

Train set	Test set - AUC						
	Deepfakes	**DiffFace**	**Face2Face**	**FaceShifter**	**FaceSwap**	**InsightFace**	**NeuralTextures**
Deepfakes	**0.9971**	0.7494	0.9403	0.6615	0.5666	0.6353	0.9596
DiffFace	0.5166	**0.9999**	0.5302	0.5076	0.5210	0.5086	0.5294
Face2Face	**0.9965**	0.5277	0.9912	0.7614	0.7343	0.6638	0.9591
FaceShifter	0.7750	0.8228	0.8491	**0.9987**	0.7094	0.6229	0.7910
FaceSwap	0.9407	0.9897	0.9934	0.8823	**0.9969**	0.4995	0.9274
InsightFace	0.6896	0.7830	0.6146	0.5203	0.5447	**0.9725**	0.5843
NeuralTextures	0.9928	0.8561	0.9891	0.9220	0.9302	0.6603	**0.9933**

Table 2: Performance of HF-FFD with an Xception classifier in a cross-dataset scenario.

(a) (b) (c) (d)

Figure 5: **Typical examples of artifacts that the SBI method successfully detects**: obvious blending border (a), color mismatch (b), structural inconsistencies (e.g., partially deleted glasses (c)), poorly generated facial landmarks (e.g., nose (d)).

ing edges, distortions in facial landmarks, and color mismatches. An analysis of the detector using Grad-CAM [26] reveals that the last convolutional layer of the method trained on one of these subsets activates in similar regions during inference, i.e., areas around the eyes, mouth, and nose, as seen in Figure 3a to 3c.

The results also indicate that training the detector on diffusion-based deepfakes leads to poor generalization. Diffusion-based forgeries appear markedly different at first glance and do not exhibit typical artifacts. This suggests that the detector is attentive to entirely different features, as evident in the Grad-CAM analysis shown in Figure 3d, i.e., the triggering area of the last convolutional layer is typically circular, unlike any other training database.

HF-FFD Results. In this case, HF-FFD detector, we are dealing with a discrimantive model that uses the Xception architecture for classification and a specialized pipeline for feature extraction, as described in Section 3.2. We conduct training and testing of this model in the same way as with Xception. The results are shown in Table 2. As can be seen, the introduction of the pipeline significantly improves generalization. However, a more in-depth analysis using Grad-CAM is needed for a better understanding. Based on Grad-CAM analysis, we can roughly categorize the learned bases into three groups based on the focus of the last convolutional layer: nose, mouth, and philtrum (the area between the nose and mouth). The network's focus on the root of the nose and its surroundings occurs when training the network on the Deepfakes dataset. A similar focus is observed when training on the InsightFace dataset, but in this case, the center of focus is not the root of the nose; instead, it is somewhere on the edge (tip, left or right edge, or the top of the nose). In the case of the DiffFace dataset, the network focuses on the mouth, with a triangular area towards the nose. For all other datasets, the network focuses on the philtrum area, but they differ in the shape of the focus area. The Face2Face and NeuralTextures datasets have a circular area similar to Deepfakes, while the FaceShifter and FaceSwaps datasets have areas that stretch upward on the face, with the former having an hourglass shape and the latter a truncated triangle. In the case of genuine images, the model is triggered in the eye area, regardless of the training dataset. These focus areas are illustrated in Figure 4.

From the results in Table 2, it is evident that the method trained on the Deepfakes, Face2Face, and NeuralTextures subsets also generalizes well across those specific deepfake types. Moreover, it is also noticeable that the triggering area of the method on these subsets is very similar, i.e., an approximately circular area around the focus center, with slight variations in the center's position (Figure 4a, 4d, 4g). However, it turns out that the method performs better among datasets where the intersection between the triggering areas of the network is larger. Thus, a model trained on datasets with a larger triggering area (FaceSwap (Figure 4f), FaceShifter (Figure 4e), and NeuralTextures (Figure 4g)) detects deepfakes of almost all types. In contrast, training on datasets with a small triggering area (DiffFace (Figure 4c)) results in very poor generalization. A special case is the InsightFace dataset, where the center and shape of the focus are not constant/consistent. Different spatial/semantic areas in the images seem informa-

Model	Test set - AUC						
	Deepfakes	**DiffFace**	**Face2Face**	**FaceShifter**	**FaceSwap**	**InsightFace**	**NeuralTextures**
SBI	0.9106	0.5708	0.8715	0.7922	0.7851	0.5892	0.8430

Table 3: Performance of EfficientNet-b4 fine-tuned using self-blended images tested on deepfakes created with seven different approaches. Results are shown in terms of AUC.

tive for the method in these types of deepfakes areas in the images seem informative for the method in these types of deepfakes. Consequently, when recognizing forgeries of other types, we correctly detect only those images with a similar informative defect, which is evident in Grad-CAM heatmaps by the center and shape of the focus approximating the typical focusing area of this dataset. However, detection with these subsets also results in many false negatives, as in cases where the network focuses on the top of the nose, it closely resembles the focus of a genuine image (which typically focuses on the eye area). Slightly better performance is achieved only when testing on the DiffFace dataset, as the samples of these two datasets are the most similar, which is why we often obtain a triangular area at the base of the nose that closely resembles the triggering area in the DiffFace dataset.

Self-Blended Images (SBI) Results. This method relies solely on pristine images from the training dataset, eliminating the need for deepfakes in the training dataset. To evaluate its performance, we conduct tests using a pre-trained model that was trained, as described in the paper [27]. The results are summarized in Table 3. This technique utilizes only authentic images to generate pseudo-deepfakes for training the detector. This unique approach enables the direct determination of specific artifacts on which the detector should focus. The authors of this approach categorize these artifacts into four groups: landmark mismatch, blending boundary, color mismatch, and frequency inconsistency. The results indicate that the method performs comparably well in recognizing forgeries of all types where the same artifacts that were synthesized on training images are present. The method achieves its highest success rates on samples from the Deepfakes dataset (Figure 5a) and Face2Face dataset (Figure 5b), where the injected artifacts are most conspicuous. The method is also effective in detecting structural inconsistencies (e.g., partially deleted glasses in Figure 5c) and poorly generated facial landmarks (e.g., nose in Figure 5d). However, the method's performance significantly declines when confronted with forgeries that do not contain the artifacts present in the training set. It notably struggles with forgeries from the DiffFace and InsightFace datasets. In the latter, the method focuses primarily on areas that appear to have been smoothed during the forgery process. However, this is not precise enough, leading to misclassification of many genuine images as deepfakes. Forgeries from the DiffFace dataset present a unique challenge as they do not exhibit typical errors due to a different generation approach. Consequently, a classifier trained on pseudo-deepfakes with typical artifacts faces difficulty distinguishing these forgeries. This approach successfully mitigates the problem of overfitting to a specific deepfake generation method. However, the issue of generalization is then shifted to the level of selecting transformations during the synthesis of training samples. This directly influences what the classifier will decide upon during classification, meaning that in the presence of new types of forgeries expressing different defects, the detector may not successfully identify them.

5. Conclusion

In this paper, we analyzed three face forgery detection methods, evaluating them in a cross-dataset scenario and assessing generalization. Using Grad-CAM, we examined failure cases and observed that discriminative models like Xception generalize primarily among forgeries with similar textural artifacts, while models with a feature-extracting pipeline before the classifier demonstrated improved generalization when trained on datasets that induce larger focus areas in the final convolutional layer. Classifiers trained with pseudo deepfakes proved effective only when artifacts assumed during training sample generation also appeared in the forgeries. Future work will expand the analysis to a broader detector set, explore aspects like the impact of image compression, investigate the characteristics of the detection techniques in the frequency domain, and assess the discriminativness of learned image representations.

References

[1] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen. Mesonet: a compact facial video forgery detection network. In *2018 IEEE international workshop on information forensics and security (WIFS)*, pages 1–7. IEEE, 2018. 2, 3

[2] Z. Akhtar and D. Dasgupta. A comparative evaluation of local feature descriptors for deepfakes detection. In *2019 IEEE International Symposium on Technologies for Homeland Security (HST)*, pages 1–5, 2019. 2

[3] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017. 2, 4

[4] T. Fernando, C. Fookes, S. Denman, and S. Sridharan. Exploiting human social cognition for the detection of fake and fraudulent faces via memory networks, 2019. 3

[5] J. Fridrich and J. Kodovsky. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, 2012. 4

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 1

[7] J. Guo, J. Deng, X. An, and J. Yu. Deepinsight/insightface: State-of-the-art 2d and 3d face analysis project. 5

[8] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1

[9] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 2

[10] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015. 4

[11] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1

[12] H. Khalid and S. S. Woo. Oc-fakedect: Classifying deepfakes using one-class variational autoencoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020. 2, 3, 6

[13] K. Kim, Y. Kim, S. Cho, J. Seo, J. Nam, K. Lee, S. Kim, and K. Lee. Diffface: Diffusion-based face swapping with facial guidance, 2022. 1, 5

[14] P. Korshunov, M. Halstead, D. Castan, M. Graciarena, M. McLaren, B. Burns, A. Lawson, and S. Marcel. Tampered speaker inconsistency detection with phonetically aware audio-visual features. In *International conference on machine learning*, number CONF, 2019. 2

[15] N. Larue, N.-S. Vu, V. Struc, P. Peer, and V. Christophides. Seeable: Soft discrepancies and bounded contrastive learning for exposing deepfakes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21011–21021, 2023. 2, 3, 6

[16] L. Li, J. Bao, H. Yang, D. Chen, and F. Wen. Faceshifter: Towards high fidelity and occlusion aware face swapping. *arXiv preprint arXiv:1912.13457*, 2019. 5

[17] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3

[18] Y. Li, M.-C. Chang, and S. Lyu. In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In *2018 IEEE International workshop on information forensics and security (WIFS)*, pages 1–7. IEEE, 2018. 2

[19] Y. Li and S. Lyu. Exposing deepfake videos by detecting face warping artifacts. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019. 2

[20] Y. Luo, Y. Zhang, J. Yan, and W. Liu. Generalizing face forgery detection with high-frequency features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16317–16326, 2021. 2, 3, 4

[21] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 1

[22] Y. Mirsky and W. Lee. The creation and detection of deepfakes: A survey. *ACM Computing Surveys (CSUR)*, 54(1):1–41, 2021. 1, 2

[23] T. T. Nguyen, Q. V. H. Nguyen, D. T. Nguyen, D. T. Nguyen, T. Huynh-The, S. Nahavandi, T. T. Nguyen, Q.-V. Pham, and C. M. Nguyen. Deep learning for deepfakes creation and detection: A survey. *Computer Vision and Image Understanding*, 223:103525, 2022. 2, 6

[24] S. Pashine, S. Mandiya, P. Gupta, and R. Sheikh. Deep fake detection: Survey of facial manipulation detection solutions. *arXiv preprint arXiv:2106.12605*, 2021. 2, 3

[25] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. Faceforensics++: Learn-

ing to detect manipulated facial images. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1–11, 2019. 2, 5

[26] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 2, 6, 7

[27] K. Shiohara and T. Yamasaki. Detecting deepfakes with self-blended images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18720–18729, 2022. 2, 3, 4, 5, 8

[28] J. Straub. Using subject face brightness assessment to detect 'deep fakes'(conference presentation). In *Real-Time Image Processing and Deep Learning 2019*, volume 10996, page 109960H. SPIE, 2019. 2

[29] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017. 4

[30] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 4

[31] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 4

[32] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 5

[33] 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 *Proceedings of the 2nd international workshop on multimedia privacy and security*, pages 81–87, 2018. 2

[34] J. Thies, M. Zollhöfer, and M. Nießner. Deferred neural rendering: Image synthesis using neural textures. *Acm Transactions on Graphics (TOG)*, 38(4):1–12, 2019. 5

[35] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2387–2395, 2016. 5

[36] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia. Deepfakes and beyond: A survey of face manipulation and fake detection. *Information Fusion*, 64:131–148, 2020. 2

[37] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 4

[38] Y. Zhang, L. Zheng, and V. L. L. Thing. Automated face swapping and its detection. In *2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP)*, pages 15–19, 2017. 2

[39] T. Zhao, X. Xu, M. Xu, H. Ding, Y. Xiong, and W. Xia. Learning self-consistency for deepfake detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15023–15033, 2021. 2

[40] W. Zhao, Y. Rao, W. Shi, Z. Liu, J. Zhou, and J. Lu. Diffswap: High-fidelity and controllable face swapping via 3d-aware masked diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8568–8577, June 2023. 1