Team Report 1.4 – Technical Report

DeepFakeChain

Revision: v.1.0

Due date	14/4/2023
Submission date	14/04/2023
Version	1.0
Authors	Nikolaos Giatsoglou (CERTH) Symeon Papadopoulos (CERTH)

Grant Agreement No.: 957228
Call: H2020-ICT-2018-2020
Topic: ICT-54-2020
Type of action: RIA

COVER LETTER

Dear TruBlo team,

This document contains the paper of the DeepFakeChain (DFC) team, titled “Investigation of ensemble methods for the detection of deepfake face manipulations”, which serves as the 4th deliverable (TR1.4) of the DFC TruBlo project. In particular, our paper compares different designs for ensemble AI models, which have been investigated and developed during the running of the DFC project. In addition, it has been submitted to the arXiv platform to be made publicly available, according to the proof attached in the mail communication. Please note that we plan to submit our work to a peer-reviewed conference or journal, after further refinements in the text and after performing a more comprehensive evaluation of our methods on a larger dataset, which requires more time due to the demanding (in terms of compute time) nature of the experimental work. We estimate that we will proceed with the submission in May. Once this work is completed, we also intend to integrate the new algorithms to the deepfake detection service used in the DFC platform.

We are available for any further communication.

With respect,
The DFC team

Funded by the Horizon 2020
Framework Programme of the European Union

Investigation of ensemble methods for the detection of deepfake face manipulations

Giatsoglou, Nikolaos
CERTH
ngiatsog@iti.gr

Papadopoulos, Symeon
CERTH
papadop@iti.gr

Kompatsiaris, Ioannis
CERTH
ikom@iti.gr

Abstract

The recent wave of AI research has enabled a new brand of synthetic media, called *deepfakes*. Deepfakes have impressive photorealism, which has generated exciting new use cases but also raised serious threats to our increasingly digital world. To mitigate these threats, researchers have tried to come up with new methods for deepfake detection that are more effective than traditional forensics and heavily rely on deep AI technology. In this paper, following up on encouraging prior work for deepfake detection with attribution and ensemble techniques, we explore and compare multiple designs for ensemble detectors. The goal is to achieve robustness and good generalization ability by leveraging ensembles of models that specialize in different manipulation categories. Our results corroborate that ensembles can achieve higher accuracy than individual models when properly tuned, while the generalization ability relies on access to a large number of training data for a diverse set of known manipulations.

1 Introduction

New technologies bring new challenges. This is especially true for *deepfakes*, a type of synthetic media with increased photorealism, which has been recently made possible with advancements in machine learning (ML) and artificial intelligence (AI) [4]. On one side, deepfakes promise to revolutionize the media industry, including films and games, and to offer outlets of creativity to independent creators and Internet users. On the other side, they pose serious threats, for example, by easily generating and spreading political propaganda, revenge porn, and other types of harmful content. Since many deepfake cyberattacks rely on manipulating fake identities, research to date has been primarily focusing on human face forgeries [5], examples of which are shown in Fig. 1.

After acknowledging the serious threat of deepfakes, the research community has shown increased interest on appropriate detection methods. While media

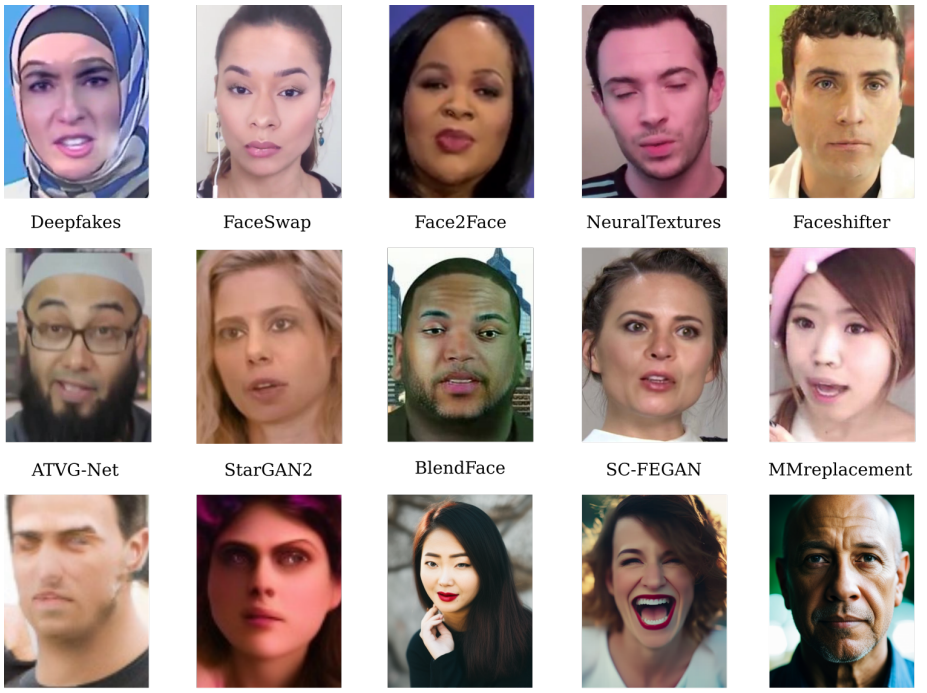

Figure 1: Examples of face forgeries created with different generation and manipulation methods. It is evident that different methods can create deepfakes of varying quality. The examples are from the FaceForensics++ [1], the ForgeryNet [2], and the Generated-Faces-in-the-Wild [3] datasets, except for the stable diffusion images that were generated directly from the website of Stability.AI (`https://dreamstudio.ai`).

forensics based on conventional statistical and ML methods have been proposed, a consensus has emerged that deep neural networks are more effective at detecting the subtle artifacts of deepfakes [1]. In particular, deep learning models can automatically learn highly discriminative features for deepfake detection without relying on laborious handcrafted feature engineering.

Deep learning techniques, however, also have limitations. First, while they achieve high detection accuracy on known manipulations, their accuracy drops considerably on unseen manipulations [6] and is sensitive to processing operations such as compression and resizing [1]. This is the *generalization* problem of deepfake detection. Second, since forgeries are refined and become undetectable to the human eye, detection models need to rely on increasingly subtle artifacts and their decisions are hard to justify and explain. This is the *explainability* problem of deepfake detection. Third, deepfake generation methods continually improve and new methods come up, so that forensics turn into an arms-race kind of problem. Finally, since deepfakes are ultimately assorted pixels like real images and videos, it is conceivable that they will be impossible to detect in the near future. Currently, there are efforts to embed provenance information in media to securely track all possible manipulations but the adoption of these technologies requires compliant capturing devices and is at a preliminary stage[1]. In the meantime, detection algorithms are worthwhile.

From the ML perspective, deepfake detection is typically treated as a classification problem, either at the video- or the image-/frame-level, supplemented by related tasks such as temporal and spatial segmentation. The deepfake class however is not clearly defined. Focusing on facial manipulations, a deepfake can be understood as a face that deviates from a representative collection of real human faces. Due to the photorealism of deepfakes, high level artifacts are becoming difficult if not impossible to detect; hence detection algorithms try to detect the subtle artifacts of the generating algorithms. Since different generation algorithms produce different artifacts, we believe that deepfake detection is best framed as an *attribution* problem.

Attribution techniques have been used in the literature of generative adversarial network (GAN) images [7], where different GAN architectures are identified with specific fingerprints, and are also applicable to face forgeries. An attractive approach to attribution is through an *ensemble* of AI models that specialize in different forgeries, which can readily adapt to new manipulations by incorporating new models to the ensemble. In addition, ensembles are popular in the deepfake literature because they can pool the performance of the base models and potentially generalize to unseen manipulations. Indeed, we hypothesize that a sufficiently large ensemble of manipulation methods can generalize better by detecting similarities of the unseen manipulations with the known manipulations. There are many ways, however, to build an ensemble and it is not clear which design is optimal. Due to this, the aim of our work is to experimentally investigate different designs of ensemble architectures for the detection of deepfakes.

[1] https://c2pa.org/

2 Related Work

This section aims to give a brief overview of the deepfake detection literature in order to better situate our work within it. Section 2.1 describes the basic forensic tasks for deepfakes. Section 2.2 describes the main technical approaches for detection with an emphasis on deep neural networks. Finally, Section 2.3 zooms in on the ensemble techniques and their potential for deepfake detection.

2.1 Overview of deepfake forensic tasks

While traditional media forensics address operations of adding, editing, and removing objects from media, deepfakes cover a variety of manipulations of increased photorealism that are powered by AI technology. These include synthesis of completely artificial images based on a reference collection, a text input, or sketch, as well as style transfers and facial manipulations [4]. For the detection of these forgeries, the literature in divided in two branches: i) detection of GAN-generated images, and ii) detection of face manipulations in videos [4]. Focusing on face manipulations, they can be further distinguished in multiple subtypes such as *face-swaps*, i.e., replacing a face with another one, *face editing*, i.e., changing the characteristics of a face without replacing its identity, *reenactment*, i.e., changing the expression of a face in alignment with another face, while taxonomies are continually refined. Another approach, proposed by [2], is to split the face forgeries into two broad categories, *identity-remained* and *identity-replaced*, depending on if the identity in the tampered video is maintained. This distinction is useful because identity-remained forgeries are more challenging to detect due to the lack of blending artifacts.

The base forensic task for face manipulations is *detection*, i.e., recognizing which images or video contain fake faces, which can be formulated as a binary classification problem. [2] has extended this task to ternary classification which tries to characterize the type of forgery as identity-remained or identity-swapped, in an effort to provide more information for forgeries. Following this line, more fine-grained tasks are possible that distinguish the exact subtype of forgery or the underlying generation algorithm. Identifying the generation algorithm is called *source attribution* and is popular in the literature of GAN-generated images. This task tries to fingerprint the artifacts of GAN models and/or the noise patterns of real cameras left in authentic images [7]. Attribution however is less common in the literature of face manipulation: to the authors' knowledge, [8] is the only work where attribution of face-swap models is the main focus. This is partly explained because existing datasets frequently omit attribution labels to mimic a more realistic setting but the trend may be reversed with newer datasets like ForgeryNet [2]. In addition, [9] has produced preliminary evidence that more fine-grained discrimination of manipulations can lead to better detection and generalization to unseen manipulations.

Beyond classification problems, deepfake forensics also include segmentation tasks either in the temporal domain, i.e., finding which frames of a video are forged, or the spatial domain, i.e., finding which areas of a frame are forged.

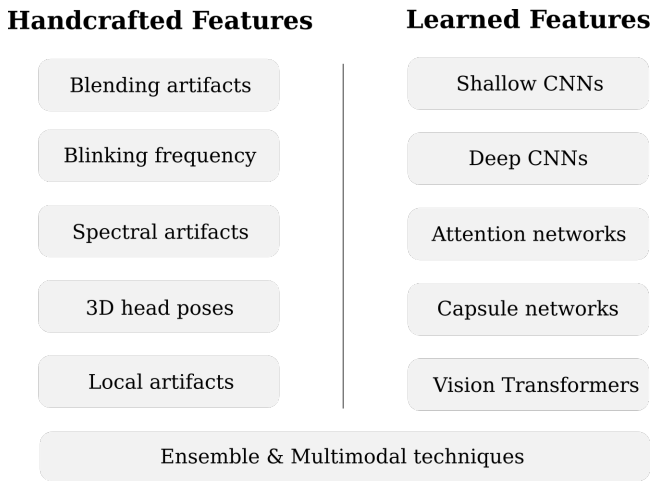

Figure 2: Representative approaches for deepfake detection with handcrafted and learned features. Notice that ensemble and multimodal techniques can be used with either type of features.

Since faces need to be extracted from images, face recognition is also relevant and used in the preprocessing step of the detection pipeline. Although this preprocessing step is frequently taken for granted, false positives can have a negative impact on the pipeline, as explored in [10]. With newer deepfake datasets, algorithms are also expected to discern and keep track of multiple identities. Finally, explainability of detection, i.e., justifying the decision of detection algorithms to humans, is a widely acknowledged problem but hard to evaluate and with no standardized tasks. Currently, the literature uses heatmaps, produced by standard explainable AI (XAI) techniques, that can reveal potentially tampered areas [11]. These heatmaps are evaluated qualitatively [12] and are not straightforward to interpret, considering the photorealism of deepfakes and the subtlety of the detected artifacts.

Our approach: Our work focuses on detection but from an attribution perspective. Inspired by [9], we believe that such as approach will become important for generalization as more data for known manipulations become available with newer datasets, e.g., [2]. We further believe that attribution can be useful for explainabilty, directing human users to suspected types of forgery.

2.2 Overview of deepfake detection methods

Initial approaches to deepfake detection used handcrafted features, such as scale-invariant feature transform (SIFT) vectors, image spectra, media stream descriptions, distorted facial landmarks, and biological signals [13]. While these methods are easier to explain to humans, they have become less popular than neural models that automatically learn features from pixels due to their ability

for optimal feature extraction, e.g., see [14]. In addition, deep neural networks are considered more effective than their shallow counterparts [15], especially after the key experimental work of [1]. Deep learning models are capable of learning both low- and high-level features of media, due to the pooling operations of subsequent layers, but are prone to overfitting: while they display $> 90\%$ accuracy on the test set of the datasets used for training, their accuracy is significantly decreased on unseen datasets. A significant amount of work has thus focused on finding architectures that can generalize better.

From a technical perspective, deepfake detectors operate at the image-/frame- or the video-level. Frame-level detectors detect intra-frame inconsistencies and aggregate the results to the whole video. They are typically based on different flavors of convolutional neural networks (CNNs) [15] and, more recently, on visual transformers (ViTs) that split an image into multiple patches, e.g., [16]. While this approach cannot detect temporal and audiovisual inconsistencies, frame-level detector are lightweight and practical. On the other hand, video-level detectors operate on sequences of frames and, possibly, audio. They typically extract features for consecutive frames which are then passed in a temporal-aware model such as a recurrent neural network (RNN), a long short-term memory (LSTM), or a transformer [15]. Other approaches include 3D CNNs, capsule networks, optical flow models, and fusion techniques to incorporate audio information [15]. Video-level detection has more potential to detect inconsistencies but it is significantly more intensive in terms of computational resources.

Our approach: Our work focuses on frame-level detection due to its usefulness for general image-level detection and its use to incorporate in ensembles. Lightweight methods are also worthwhile for application in resource contrained environments, e.g., edge devices.

2.3 Overview of ensemble techniques for deepfake detection

Ensembles of CNNs have become popular for deepfake detection, after they were found in the top solutions of the DeepFake Detection Challenge (DFDC) competition [2]. Ensemble learning is an ML technique which combines multiple models so that their ensemble becomes more accurate and robust than each individual model [17]. An ensemble architecture comprises the base learners and a combination layer, which can be either another learner or an aggregation function which does not require training. By combining multiple simpler models, ensembles aspire to create more accurate and robust models that generalize better. To achieve this, base models have to be reasonably accurate and diverse.

[18, 19] proposed ensembles of CNNs with different architectures for diversity. In particular, [18] used 7 different architectures [3] and combined their output with another CNN meta learner, while [19] experimented with 26 isolated CCNs and combined the ResNet models, which were found the least reliable, in order

[2] https://www.kaggle.com/competitions/deepfake-detection-challenge/leaderboard

[3] The architectures used were XceptionNet, ResNet101, InceptionResNetV2, MobileNet, InceptionV3, DenseNet121, and DenseNet169.

to boost their performance. In contrast, [20] used a single CNN architecture, Efficient-B4, and diversified it with attention and different training procedures (supervised and unsupervised mode with triple loss). These works achieved high accuracy on the datasets used for training but were not tested on out-of-distribution datasets.

[21, 21] were inspired by the solutions of the DFDC challenge and experimented with the top solutions. For example, [21] reused the architecture of the WM team, which uses an ensemble of 3 models based on XceptionNet and EfficientNet, and incorporated attention maps for explainability and augmentation of the training dataset. [22] mixed the top solutions of DFDC and experimented with different mixing strategies, in order to achieve higher accuracy on the DFDC dataset. Other related works include [23], which combined existing models that detect physiological signals, and [24], which ensembled models that were trained on different facial landmarks. These approaches added inductive bias in learning, which helps with their intepretability but also have the limitation of hand-crafted features.

Finally, [25] comprehensively studied the impact of different fusion techniques on performance, which is frequently unexplored. They used an ensemble of 6 models[4] and tested various parametric and non-parametric fusion methods. They found that parametric techniques work much better than non-parametric technique but their decisions are not easy to interpret and not readily extensible.

Our approach: From the above, we see that ensemble models are popular in the literature as a way to boost the performance of already successful models but they have not been studied consistently due to the large degrees of freedom in the ensemble design. In this work, we want to compare how different configurations of ensembles work and their impact on the generalization of deepfake detection.

3 Methodology

Our deepfake detector operates at the frame level and its architecture is shown in Fig. 3. The preprocessor is responsible for sampling video frames at a constant rate, recognizing and cropping the present human faces, and normalizing them to a specific size. After the faces are extracted, they are clustered to unique identities and outliers are discarded, according to the approach of [10]. This step is taken in order to avoid false positives from the recognition module. Subsequently, the faces pass through the frame detector, which tries to detect the presence of forgeries, and potentially attribute the manipulation method. The resulting scores describe the confidence of the faces being fake and are eventually aggregated to derive a score for the whole video.

In the following, we focus on the design and performance of the frame detector, on which the whole pipeline depends. In particular, we first define the problem of deepfake detection formally from an ML persepctive, and then describe the ensemble design that we investigate.

[4]The architectures used were ResNet50, XceptionNet, EfficientNet-B4 with/out attention and with supervised/unsupervised training.

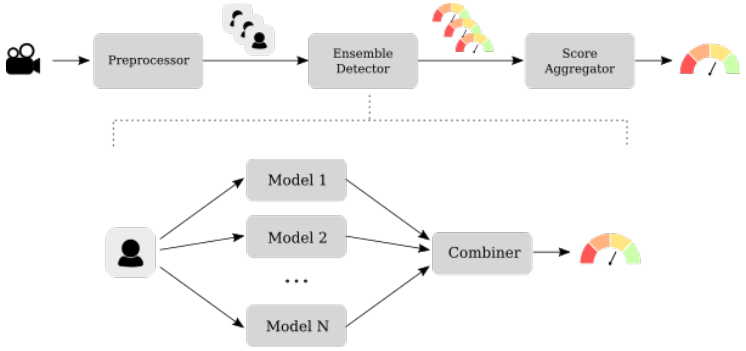

Figure 3: The architecture of our deepfake detector. The preprocessor samples frames from an input video file with a constant rate and extracts faces from them. A clustering approach that identifies faces belonging to the same identity is used in order to avoid false results in the face extraction, like in [10]. The ensemble detector employs a collection of N AI models to analyze the faces and a combiner to reach a robust score. Finally, the score aggregator aggregates the scores of each frame and face to arrive at a video-level score.

3.1 Problem setup

We denote a face by the triplet (x, y, z) where x is the face image, y a multiclass attribution label, and z a binary detection label. In more detail, the image x is an array of dimensions (C, H, W) where C is the number of color channels, and H, W are the image's height and width. Considering a selection of K possible manipulations, the attribution label takes $K + 1$ values where value 0 represents a real face and the remaining values the K manipulations. Finally, the detection label is binary where values 0 and 1 represent a real and a fake face respectively. If we know the attribution label, it is easy to derive the detection label through the following function:

$$z = g(y) = \begin{cases} 0 & \text{if } y = 0 \\ 1 & \text{if } y > 0 \end{cases}. \tag{1}$$

Based on the above, we can define the following tasks:

Detection task: We seek a function $\hat{p}(z|x)$ that approximates the posterior distribution $p(z|x)$, so that:

$$\hat{z} = \arg\max \ \hat{p}(z|x) \tag{2}$$

Attribution task: We seek a function $\hat{p}(y|x)$ that approximates the posterior distribution $p(y|x)$, so that:

$$\hat{y} = \arg\max \ \hat{p}(y|x) \tag{3}$$

With a dataset offering suitable labels, it is straightforward to train single models for the detection or the attribution tasks, e.g., using the binary and the

multiclass cross-entropy loss. Notice that the attribution models can be easily converted to the detection task by employing the function g as $\hat{z} = g(\hat{y})$.

3.2 Binary detection ensemble

For the binary detection ensemble, we group N detection models, denoted by $\hat{p}_i(z|x)$, whose training has been diversified. We then soft-combine the models' outputs so that

$$\hat{p}(z|x) = \frac{\sum_{i=1}^{N} \hat{p}_i(z|x)}{N}. \tag{4}$$

The final result is given by (2). This ensemble cannot be used for attribution because it does not distinguish the different manipulations.

3.3 Multiclass attribution ensemble

For the multiclass attribution ensemble, we group N attribution models, denoted by $\hat{p}_i(y|x)$, whose training again has been diversified. We then soft-combine the models' outputs so that

$$\hat{p}(y|x) = \frac{\sum_{i=1}^{N} \hat{p}_i(y|x)}{N}. \tag{5}$$

The final result is given by (3) and can be used for detection by converting to binary via the function g.

3.4 One-manipulation-vs-real ensemble

For the one-manipulation-vs-real ensemble, each model specializes in one of the K manipulation types and is trained as a binary classifier that discriminate between the i-th manipulation and the real class. Letting s_i be the score of the i-th manipulation and t a threshold, the final decision is given through max pooling as

$$\hat{y} = \begin{cases} \arg\max s_i & \text{if } \max\{s_i\} > t \\ 0 & \text{if } \max\{s_i\} < t \end{cases}, \tag{6}$$

which can be converted to binary via the function g. This ensemble is very convenient because it can be easily extended with new models that specialize in different manipulations.

3.5 One-manipulation-vs-rest ensemble

For the one-manipulation-vs-rest ensemble, each model specializes again in one of the K manipulations but discriminates against all the remaining manipulations. Again, considering the threshold t to discern the real faces, the final decision is taken via (6) The one-vs-rest ensemble is similar to the one-vs-real ensemble but delineates the boundaries of each manipulation class more clearly. The downside is that it requires the knowledge of the other manipulation classes during training, hence it is not as extensible.

4 Results

In this section, we present the results of our experiments. Section 4.1 states our experimental setup, including the used models, hyperparameters, and training procedure. The following sections then describe our results on the intra-dataset attribution task (Section 4.2), the intra-dataset detection task (Section 4.3), and the cross-dataset detection task (Section 4.4).

4.1 Experimental setup

We trained all models with the FaceForensics++ dataset which contains ground truth values about the manipulations. We used the official split of FaceForensics++[5] and preprocessed the videos as described in Section 3 to create a dataset of face images. In particular, we sampled the videos at 1 fps and extracted the faces with the MTCNN face detection module from the facenet-pytorch library[6], which is popular in the literature. The faces are subsequently resized to $(224, 224)$ which is compatible with the model used for detection, and augmented during training for robustness similarly to the winning solution of the DFDC challenge[7].

For the ensemble, we used the EfficientNet-B0 CNN model [26] with pretrained weights on ImageNet for the base classifiers. The EfficientNet-B0 architecture was selected because it is performant and sufficiently lightweight in resources to construct big ensembles. The models were then trained with the PyTorch framework, using binary and multiclass cross-entropy loss, weight decay $5 * 10^{-}4$, and the Adam optimizer with learning rate 0.001. The models were trained on the raw quality videos of the FaceForensics++ dataset for 40 epochs with an early stopping criterion of no improvement in validation accuracy after 5 consecutive epochs. To address the class imbalance, we oversampled the minority classes during training and used the balanced accuracy metric to evaluate the models.

In more details, for evaluation, we tested our models in both the *intra-dataset* scenario, i.e., on the test set of the FaceForensics++ dataset, and the *cross-dataset* scenario, i.e., on the Celeb-DF, DFDC preview, DFDC, and OpenForensics datasets. We evaluated both the attribution and detection task in the intra-dataset scenario, and only the detection task in the *cross-dataset* scenario, since the classes of the other datasets do not match. The balanced accuracy metric is also different in the two tasks: for detection, it is defined as:

$$BA_{det} = 0.5\ p(z = 0|x = 0) + 0.5\ p(z = 1|x = 1) \tag{7}$$

while for attribution, it is defined as:

$$BA_{att} = 0.5\ p(y = 0|x = 0) + 0.5\ \frac{\sum_{i=1}^{K} p(y = i|x = i)}{K}. \tag{8}$$

[5] https://github.com/ondyari/FaceForensics/tree/master/dataset/splits
[6] https://pypi.org/project/facenet-pytorch/
[7] https://github.com/selimsef/dfdc_deepfake_challenge#augmentations

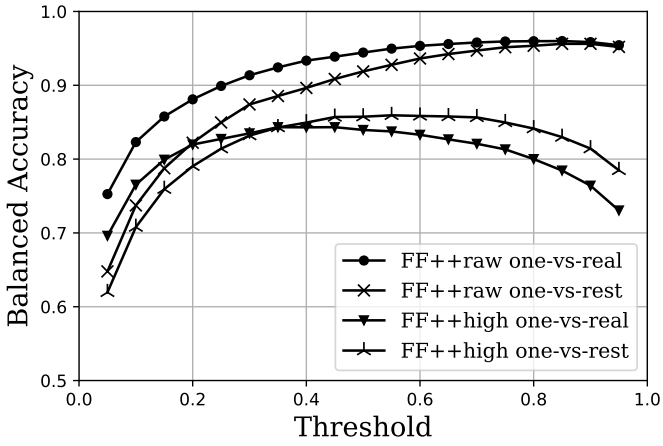

Figure 4: Balanced attribution accuracy for the raw and high quality version of the FaceForensics++ test dataset with different thresholds of the one-vs-real and one-vs-rest ensemble.

The intuition behind (7) and (8) is as follows. Since we do not know the prior distribution of the real and fake faces, we give equal weights to the real and fake class, and we evenly distribute the weight of the fake class to the manipulation categories in the attribution task. In this manner, we avoid bias towards the fake classes and evaluate our models on their performance on all classes.

4.2 Intra-dataset attribution task

We first evaluate our models on the intra-dataset attribution task. Before deriving our results, we investigated appropriate thresholds t for the one-vs-real and one-vs-rest ensembles. In particular, we evaluated the balanced accuracy for the attribution task on the raw and high quality version of FaceForensics++ for different values of t spread evenly between 0.05 and 0.95 with step 0.05. The results are shown in Fig. 4. We see that for the raw version of FaceForensics++, the accuracy increases monotonically with higher thresholds up to the value 0.9. This implies that the ensembles are both confident and accurate in detecting manipulations so that higher thresholds avoid false positive detections for real images. In addition, the one-vs-rest ensemble outperforms the one-vs-real ensemble for all thresholds. The behavior is different however for the high quality version of FaceForensics++ where the accuracy of both ensembles is lower and a better trade-off is achieved in the mid-range thresholds. We also see that the performance of the two ensembles is much closer and the best ensemble depends on the threshold value. For better generalization and simplicity, we selected the natural mid-range threshold 0.5.

With these thresholds, the results of the intra-dataset attribution task are

	Single models	Ensembles		
	Multiclass	Multiclass	One vs real	One vs rest
FF++raw	93.35 - 96.25	96.45	94.43	91.88
FF++high	76.29 - 84.78	83.51	83.95	85.74
FF++low	43.89 - 53.69	52.28	46.00	46.76

Table 1: Balanced accuracy for the attribution task evaluated on the test split of the FaceForensics++ datasets.

shown in Table 1, which compares single attribution models with ensembles of 6 models, as is the number of classes in FaceForensics++. For the raw quality dataset, we see that the accuracy of all models is in the $> 90\%$ range. In addition, the accuracy of the single models has small variation and their ensemble is better that each individual one. Finally, the one-vs-real ensemble is better than the one-vs-rest ensemble, as also seen in Fig. 4.

In the high quality dataset however, the behavior changes appreciably. First, there is a large variation in the accuracies of the single models, which highlights the importance of hyperparameter optimization for generalization. Indeed, considering that the single models differed only on the random parameters of the training procedure, the best single model managed to extract better features for generalization due to pure luck. Second, the accuracy of their ensemble is close to but not better than the accuracy of the best single model. This indicates that ensembles is a practical way to enhance non-optimal individual models but to ensure best ensemble performance, parametric combination is needed. Third, the accuracies of the one-vs-real and one-vs-rest ensembles are higher and that the one-vs-rest ensemble outperforms the one-vs-real ensemble, as also seen in Fig. 4. This implies that the better discriminative ability of the one-vs-rest ensemble is helpful for generalization.

Finally, we see that the results on the low quality dataset are low, in some cases lower than pure chance. This shows that our model cannot distinguish manipulation artefacts when the images are heavily compressed and that a more sophisticated approach is needed for their training. Nevertheless, in the next section we evaluate if the detection models are still able to better detect general manipulation artefacts from authentic faces.

4.3 Intra-dataset detection task

Table 2 shows the intra-dataset detection performance of our models. In this case, we include the multiclass attribution models that are converted to binary labels to check if the increased discrimination helps the detection task. The table includes both single multiclass models and their ensemble. In the raw quality version of the dataset, we see again that all models achieve accuracy $> 90\%$. The single multiclass models can potentially achieve higher accuracy than their binary counterparts although their range of accuracies is wider. The ensembles of both types of models achieve higher performance than individual

Dataset	Single models		Ensembles			
	Binary	Multiclass	Binary	Multiclass	One vs real	One vs rest
FF++raw	94.67 - 95.81	94.32 - 96.61	96.27	96.77	95.26	93.15
FF++high	70.71 - 80.81	77.02 - 86.68	75.50	83.94	85.71	87.41
FF++low	59.60 - 63.16	60.11 - 65.07	62.58	62.68	65.36	63.27

Table 2: Balanced accuracy for the detection task evaluated on the test split of the FaceForensics++ datasets.

model, higher also than the one-vs-real and one-vs-rest ensembles. We note again that the one-vs-real ensemble performs better in the raw quality dataset.

As was the case in Section 4.2, the behavior is markedly changed in the high quality dataset. Again, we note a significant variation in the accuracies of the single models, both for the binary and the multiclass cases, and that their ensembles do not achieve better performance than the individual models. Most importantly, we see that the multiclass models achieve significantly higher accuracy than the binary detectors, confirming the observations of [9]. Even more impressingly, the one-vs-real and one-vs-rest ensembles achieve the highest accuracies, suggesting the ability of specialized models to better detect forgeries than aggregate ones.

Finally, for the low quality dataset, we again note lower accuracies, suggesting the weakness of our models to detect discriminative features under heavy compression, however, the results are improved compared with the attribution task. This implies that despite the poor attribution ability, the models are still able to distinguish some general forgeries from authentic faces. In this case, the one-vs-real achieves the best accuracy along with a single attribution model.

4.4 Cross-dataset detection task

We conclude with our results on the cross-dataset detection task, shown in Table 3. In general, we note that the accuracies are low for all datasets, suggesting the poor generalization ability of our models. Again, we see that the ensemble versions of the single models do not guarantee better performance than the individual models. Indeed, this has been a constant characteristic in all datasets except for the raw version of FaceForensics++ where our models had the best performance. This could be a prerequisite condition for the superiority of the averaging ensemble, otherwise a parametric approach is required. Additionally, the one-vs-real ensemble achieves almost always the best performance among all ensembles but, interestingly, it is comparable or even lower than the best performance of individual models.

Despite the above, we must highlight that the perfomances are too close to random chance to extract safe conclusions. The poor generalizibility of the ensemble can be attributed to the age of the FaceForensics++ dataset. In particular, except for CelebDF, all the unseen datasets were published after FaceForensics++ and contained deepfakes of higher quality and diversity. Additionally, the

	Single models		Ensembles			
	Binary	Multiclass	Binary	Multiclass	One vs real	One vs rest
CelebDF	59.48 - 68.84	62.08 - 66.02	65.97	65.57	65.46	54.13
DFDC preview	55.08 - 63.29	54.17 - 60.39	59.82	57.28	64.49	54.15
DFDC	53.04 - 56.35	51.87 - 56.00	54.61	53.18	55.82	54.36
OpenForensics	45.50 - 57.15	46.22 - 55.65	53.32	51.14	54.89	49.17

Table 3: Balanced accuracy for the detection task on unseen datasets.

manipulations of FaceForensics++ are too few to achieve good generalization. We believe that using a more recent and diverse dataset like ForgeryNet, the generalization ability of the ensembles can be significantly improved.

5 Conclusion

Deepfakes are a product of the recent AI revolution, which we currently do not know how to handle. Fearing the potential damage to society and the individual, the research community has sought to find adequate solutions for the detection of deepfake media but no definitive solution has yet emerged. In this paper, following on encouraging and under-explored leads in the literature, we have investigated the potential of ensemble models to successfully detect face forgeries through attribution, aspiring to generalize on unseen manipulations. Our results have shown that, when properly tuned, ensembles can indeed achieve superior performance than individual models but a small number of manipulations is not sufficient for good generalization. In the future, we plan to enhance our solution with greater diversity of manipulations, specifically, the ForgeryNet dataset which we believe will unlock more opportunities for generalization.

Acknowledgment

This research was supported by the EU H2020 projects TruBlo (Grant Agreement 957228) and AI4Media (Grant Agreement 951911).

References

[1] Andreas Rossler et al. “Faceforensics++: Learning to detect manipulated facial images”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 1–11.

[2] Yinan He et al. “Forgerynet: A versatile benchmark for comprehensive forgery analysis”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 4360–4369.

[3] Ali Borji. "Generated faces in the wild: Quantitative comparison of stable diffusion, midjourney and dall-e 2". In: *arXiv preprint arXiv:2210.00586* (2022).

[4] Luisa Verdoliva. "Media forensics and deepfakes: an overview". In: *IEEE Journal of Selected Topics in Signal Processing* 14.5 (2020), pp. 910–932.

[5] Ruben Tolosana et al. "Deepfakes and beyond: A survey of face manipulation and fake detection". In: *Information Fusion* 64 (2020), pp. 131–148.

[6] Ali Khodabakhsh et al. "Fake face detection methods: Can they be generalized?" In: *2018 international conference of the biometrics special interest group (BIOSIG)*. IEEE. 2018, pp. 1–6.

[7] Brandon Khoo, Raphaël C-W Phan, and Chern-Hong Lim. "Deepfake attribution: On the source identification of artificially generated images". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 12.3 (2022), e1438.

[8] Shan Jia, Xin Li, and Siwei Lyu. "Model attribution of face-swap deepfake videos". In: *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2022, pp. 2356–2360.

[9] Anubhav Jain, Pavel Korshunov, and Sébastien Marcel. "Improving generalization of deepfake detection by training for attribution". In: *2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP)*. IEEE. 2021, pp. 1–6.

[10] Polychronis Charitidis et al. "Investigating the impact of pre-processing and prediction aggregation on the deepfake detection task". In: *arXiv preprint arXiv:2006.07084* (2020).

[11] Alejandro Barredo Arrieta et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI". In: *Information fusion* 58 (2020), pp. 82–115.

[12] Federico Baldassarre et al. "Quantitative Metrics for Evaluating Explanations of Video DeepFake Detectors". In: *arXiv preprint arXiv:2210.03683* (2022).

[13] Momina Masood et al. "Deepfakes Generation and Detection: State-of-the-art, open challenges, countermeasures, and way forward". In: *Applied Intelligence* (2022), pp. 1–53.

[14] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. "Recasting residual-based local descriptors as convolutional neural networks: an application to image forgery detection". In: *Proceedings of the 5th ACM workshop on information hiding and multimedia security*. 2017, pp. 159–164.

[15] Md Shohel Rana et al. "Deepfake detection: A systematic literature review". In: *IEEE Access* (2022).

[16] Davide Alessandro Coccomini et al. "Combining efficientnet and vision transformers for video deepfake detection". In: *Image Analysis and Processing–ICIAP 2022: 21st International Conference, Lecce, Italy, May 23–27, 2022, Proceedings, Part III*. Springer. 2022, pp. 219–229.

[17] Omer Sagi and Lior Rokach. "Ensemble learning: A survey". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8.4 (2018), e1249.

[18] Md Shohel Rana and Andrew H Sung. "Deepfakestack: A deep ensemble-based learning technique for deepfake detection". In: *2020 7th IEEE international conference on cyber security and cloud computing (CSCloud)/2020 6th IEEE international conference on edge computing and scalable cloud (EdgeCom)*. IEEE. 2020, pp. 70–75.

[19] Mrunal Kshirsagar, Shraddha Suratkar, and Faruk Kazi. "Deepfake Video Detection Methods using Deep Neural Networks". In: *2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT)*. IEEE. 2022, pp. 27–34.

[20] Nicolo Bonettini et al. "Video face manipulation detection through ensemble of cnns". In: *2020 25th international conference on pattern recognition (ICPR)*. IEEE. 2021, pp. 5012–5019.

[21] Samuel Henrique Silva et al. "Deepfake forensics analysis: An explainable hierarchical ensemble of weakly supervised models". In: *Forensic Science International: Synergy* 4 (2022), p. 100217.

[22] Anis Trabelsi, Marc Michel Pic, and Jean-Luc Dugelay. "Improving Deepfake Detection by Mixing Top Solutions of the DFDC". In: *2022 30th European Signal Processing Conference (EUSIPCO)*. IEEE. 2022, pp. 643–647.

[23] Sanjeev Rao et al. "Deepfake Creation and Detection using Ensemble Deep Learning Models". In: *Proceedings of the 2022 Fourteenth International Conference on Contemporary Computing*. 2022, pp. 313–319.

[24] Akihisa Kawabe et al. "Fake Image Detection Using An Ensemble of CNN Models Specialized For Individual Face Parts". In: *2022 IEEE 15th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC)*. IEEE. 2022, pp. 72–77.

[25] Sara Concas et al. "Analysis of Score-Level Fusion Rules for Deepfake Detection". In: *Applied Sciences* 12.15 (2022), p. 7365.

[26] Mingxing Tan and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114.