The MeVer DeepFake Detection Service: Lessons Learnt from Developing and Deploying in the Wild

Spyridon Baxevanakis
ITI-CERTH
Thessaloniki, Greece
spirosbax@iti.gr

Giorgos Kordopatis-Zilos
ITI-CERTH
Thessaloniki, Greece
georgekordopatis@iti.gr

Panagiotis Galopoulos
ITI-CERTH
Thessaloniki, Greece
gpan@iti.gr

Lazaros Apostolidis
ITI-CERTH
Thessaloniki, Greece
laaposto@iti.gr

Killian Levacher
IBM Research
Dublin, Ireland
killian.levacher@ibm.com

Ipek B. Schlicht
Deutsche Welle
Bonn/Berlin, Germany
ipek.baris-schlicht@dw.com

Denis Teyssou
Agence France-Presse
Paris, France
denis.teyssou@afp.com

Ioannis Kompatsiaris
ITI-CERTH
Thessaloniki, Greece
ikom@iti.gr

Symeon Papadopoulos
ITI-CERTH
Thessaloniki, Greece
papadop@iti.gr

ABSTRACT

Enabled by recent improvements in generation methodologies, DeepFakes have become mainstream due to their increasingly better visual quality, the increase in easy-to-use generation tools and the rapid dissemination through social media. This fact poses a severe threat to our societies with the potential to erode social cohesion and influence our democracies. To mitigate the threat, numerous DeepFake detection schemes have been introduced in the literature but very few provide a web service that can be used in the wild. In this paper, we introduce the MeVer DeepFake detection service, a web service detecting deep learning manipulations in images and video. We present the design and implementation of the proposed processing pipeline that involves a model ensemble scheme, and we endow the service with a model card for transparency. Experimental results show that our service performs robustly on the three benchmark datasets while being vulnerable to Adversarial Attacks. Finally, we outline our experience and lessons learned when deploying a research system into production in the hopes that it will be useful to other academic and industry teams.

CCS CONCEPTS

• **Information systems** → **Multimedia information systems**; **Web services**; **Data analytics**; • **Security and privacy** → **Human and societal aspects of security and privacy**.

KEYWORDS

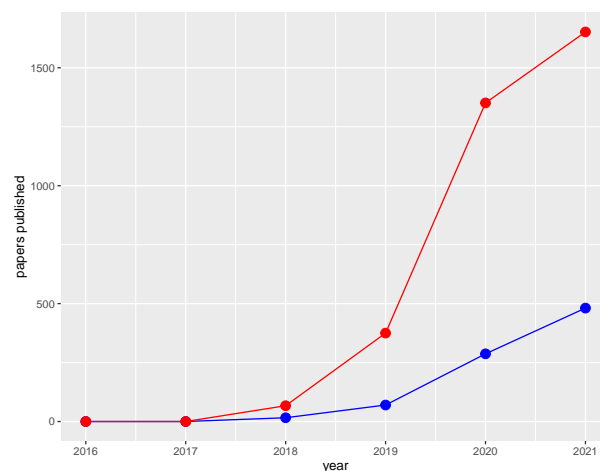DeepFake detection, Web service, Trustworthy AI

Figure 1: **The red line illustrates the number of papers where the term "DeepFake" appears at least once in the text, while the blue line illustrates the term has to be in the title and the abstract. Data obtained from https://app.dimensions.ai.**

1 INTRODUCTION

In the fight against disinformation, facial manipulation technologies are one of the most formidable weapons that malicious actors have in their arsenal in order to deceive the public's opinion. DeepFakes stand out as perhaps the most prominent of these technologies due to the photo-realistic results and the effectiveness in social media dissemination. A DeepFake refers to any fake image or video, typically containing facial manipulations to the displayed person(s), created using Deep Learning methods. Furthermore, non-face scenes/imagery can be the subject of DeepFakes such as satellite images [79].

Nowadays, DeepFakes have gained popularity owing to various free and easy-to-use tools available[1] to anyone who wishes to create fake images and videos. In combination with the drastic increase in quality fueled by the research in the area of image/video generation [25, 63, 83], DeepFakes pose a serious threat to society with far-reaching impacts. Some notable DeepFake examples include: a DeepFake of the US president Donald Trump in which he urges Belgian politicians to pull out of the Paris climate agreement[2], a DeepFake of Meta CEO Mark Zuckerberg in which he gives a sinister speech about the influence of Facebook on its users[3], and a fake video of US president Barack Obama during which he insults Donald Trump[4].

This has attracted the interest of the multimedia community for the development of methods to tackle this threat, and as a result, the generated research in the field has skyrocketed in the last recent years. Figure 1 shows the number of papers published that mention the term *DeepFake* since 2016. Furthermore, data availability has also seen such an increase in activity that in 2021 only, eight new DeepFake datasets have been released [16, 22, 24, 29, 37, 39, 56, 82].

Despite these facts, the DeepFake problem remains challenging, especially in the case of novel manipulations that have not been included in the training set of DeepFake detection systems. We argue that it is in part due to two reasons. The first relates to the challenge of training Neural Networks that are robust to out-of-distribution samples. In this context, by out-of-distribution samples we refer to DeepFakes generated with different manipulation methods than those used for training. The second reason relates to the misalignment between the synthetic datasets, developed by researchers that exhibit a strong bias towards selecting trimmed videos containing only a single face, and DeepFakes on the Internet, where videos are longer and contain many shots with multiple faces of which one or more may have been manipulated. Thus, it is evident that there is a growing need for systems that can effectively tackle these issues and mitigate the threat of DeepFakes. Such systems have to also be transparent for identifying and addressing potential issues and evaluated based on their robustness to standard adversarial attacks.

To contribute to the discussion around the problem, in this paper, we present our DeepFake detection service, its design, implementation details, and our experience deploying a multi-model system for image and video DeepFake detection in the wild. Our system receives the URL address of an image or video as input, and generates a single DeepFake probability score as output. A new input to the service triggers a multi-stage processing pipeline, including dedicated functions for the downloading and pre-processing of the input for the extraction of the contained faces. The detected faces are submitted to an ensemble scheme of five DeepFake detection models. The outputs are aggregated to derive a single probability score indicating whether the input medium contains DeepFake faces. To provide a transparent documentation for our service, we have compiled a model card. We evaluate our service on three well-known datasets and also assess the service robustness to adversarial

attacks in the spirit of trustworthy AI. Finally, we document the practical challenges we faced when pivoting to a robust service API from the point of view of research code, hoping that our experience will be helpful to other academic or industry teams in the field.

2 RELATED WORK

Numerous surveys and literature reviews have been published following the recent explosion in DeepFake research [38, 45, 49, 55]. After reviewing the creation tools and detection approaches of DeepFakes, the authors of [45] focus on the challenges for robust DeepFake detection, such as the handling of adversarial attacks. Also, [49] reviews extensively the technical background of Deep-Fakes in terms of Generative Adversarial Networks (GANs), Neural Networks and Loss functions with a particular focus on Facial Reenactment techniques, such as [70].

2.1 DeepFake Generation

DeepFakes can be classified in five major categories based on the type of applied manipulation [46]: (i) **FaceSwap**: This is a manipulation method where the face region of a target image is replaced with that of a source image. Most publicly available tools apply this kind of manipulation to generate DeepFakes. (ii) **Face Reenactment (Puppet Mastery)**: In these methods, only the facial movements and expressions are transferred from a source to a target video. A seminal such method is Face2Face [70]. (iii) **Face Attribute Editing**: This manipulation modifies a selected facial attribute (e.g. eyes, skin tone, hair) while leaving the remaining face unaltered. The evolution of Generative Adversarial Networks (GANs) in works such as [21] has significantly improved the realism of this kind of manipulations. (iv) **Face Synthesis** is concerned with synthesizing entirely new images of faces and also belongs to the GAN-related family of manipulations. Notable works include StyleGAN2 [26], used for the generation of synthetic faces in popular websites[5]. (v) **Lip-syncing**: In this manipulation, the mouth portion of an input video is altered to match an unrelated audio clip. Among the most influential lip-syncing works was one targeting President Barack Obama [65].

For the development of our DeepFake detection service, we focus on the detection of the generated media from the first category, i.e., FaceSwap, which is the most common.

2.2 DeepFake Detection Approaches

Given the growing threat of tampered media to society, a lot of methods have been proposed for DeepFake detection. One of the earliest works in the field is MesoNet [1], where a relatively shallow Convolutional Neural Network (CNN) with five layers was proposed. In their landmark work [61], the researchers benchmarked the performance of several state-of-the-art CNNs on their proposed novel FaceForensics++ dataset, showing that an XceptionNet network [7] outperformed the competition.

Research since then has evolved by combining CNNs with other architectures such as Recurrent Neural Networks (RNNs) [19], Long Short-Term Memories (LSTMs) [36, 47] or Attention heads [14, 30, 73, 80, 81, 84]. In [3] the authors propose an ensemble of numerous CNN classifiers, based on the popular EfficientNet network

---

[66] in tandem with attention mechanisms and Siamese training with the goal of accurately detecting DeepFakes. In contrast, [68] takes a different approach by using a video-level Convolutional LSTM-based Residual Network combined with a transfer learning training strategy to perform detection. Furthermore, the authors experimented with *Merge Learning*, i.e., directly train the model with all manipulations, and *Transfer Learning* or in other words using a pre-trained model from a source manipulation domain to train on a few videos from the target domain. Following the advent of Transformer-based architectures [74] to the Deep Learning scene, many authors have incorporated attention mechanisms to solve the DeepFake Detection problem [28, 64]. Besides these works, Capsule Networks [62] have also been applied to the DeepFake problem [27, 47, 51, 52]. Additionally, some methods incorporate domain specific physiological signals such as head poses [76] or eye blinking [41] in order to exploit the inconsistencies resulting from video modification. Furthermore, since new DeepFake manipulation methods are introduced at a very rapid pace, a robust DeepFake detector should be able to generalize to examples from novel manipulation/generation models. Works that attempt to tackle the generalization problem are [31, 32, 40].

Following the trend in the state of the art, we build five models using the EfficientNet [66, 67] as a backbone, combined with a Transformer-based architecture, i.e., the DETR network [5].

2.3 DeepFake Detection Services

Seeing the threat that DeepFakes pose on society, several companies and academics have developed DeepFake detection web services.

DeepWare[6] developed an online DeepFake scanner as well as an Android application for the identification of DeepFake videos. Their approach uses an EfficientNet-B7 [66] pre-trained on ImageNet [11] and fine-tuned on the DFDC dataset [12] that operates at frame level. Since the dataset is imbalanced, containing approximately 20K real and 100K fake videos, they balanced it at training time by randomly selecting equal number of real or fake videos.

DuckDuckGoose[7] has created the DeepDetector, which is a DeepFake detection system, as well as a browser detector plugin named DeepfakeProof. Additionally, they have created the so-called Replicant DeepFake creation system that can be used to test the reliability of biometric authentication systems. Unfortunately, they do not offer more information with regards to their model architecture, training strategy, or training data.

DeepFake-o-meter[8] is an academic non-profit work created by the University of Buffalo's Media Forensics Lab. Introduced in [43], it is a web service where a user can upload a video link or file and have the DeepFake detection results be sent to the user's email. It consists of 12 DeepFake detection algorithms from the literature.

2.4 Trustworthy AI

Autonomous AI systems are embedded into every aspect of daily life and deployed in high-impact tasks such as driving vehicles [58] and most currently, controlling a nuclear fusion reactor [10]. Thus it is evident that AI systems need to be reliable, explainable,

and transparent for building trust and preventing harmful decisions. In this paper, we are mostly concerned with the aspects of transparency and robustness.

Initially proposed by [50], *model cards* are a form of documentation meant to accompany trained AI models. The main scope is to inform and guide end users for the proper use of the underlying tool, as well as help them interpret the output results. Among others, a model card includes details regarding the deployed model, i.e., the model's architecture or the processing pipeline that is applied given an input. It also comprises details about the data used and the process followed for the training and evaluation of the models. Additionally, model cards usually follow a versioning scheme similar to the accompanied models, where the changes from prior tool versions are described. Also, the model card facilitates the developers of such AI models so as to describe the caveats and relevant factors that may affect model performance and make recommendations for the intended use of the tool.

Adversarial attacks are a common practice that malicious actors can use to affect the performance of similar systems. These attacks come in various shapes and forms but can be categorised as: *Evasion attacks* intentionally perform targeted alterations to an image or video so as to confuse a machine learning system [75] in making a wrong prediction. *Poisoning attacks* [2] attempt to alter the dataset used to train an AI model. This type of attack occurs prior to the deployment of the AI system. *Extraction attacks* [23] operate on a different dimension than previous attacks. These aim at stealing the underlying parameters of AI models and thus reproducing the same model at very little cost compared to the one invested for development. *Inference attacks* [8] finally consist in identifying the characteristics of specific samples that were used to train an AI model. This can be particularly problematic when personal information was used to train a system, which could be breached and damage individuals' privacy. A noteworthy publication is [17] where the authors evaluate the robustness of DeepFake detectors against multiple DeepFake attacks and subsequently experiment with defense methodologies against them.

To this end, in the spirit of robust and trustworthy AI, we accompany our DeepFake detection service with proper documentation, i.e., a model card, as well as evaluate it based on its robustness against adversarial attackers using evasion attacks.

2.5 Content Authenticity Initiative

Another interesting approach to countering the challenge of digital media manipulation is the Content Authenticity Initiative (CAI) [18], which proposes a toolset to track the origin and manipulation history of media via an embedded Content Record. It tracks, among other things when a specific media file was produced and by whom, what editing was performed and with what tools as well as the original file before any manipulations occurred. CAI's members include companies such as Adobe, Twitter and the New York Times.

3 SERVICE DESIGN AND IMPLEMENTATION

In this section, we describe the processing pipeline and implementation of our DeepFake detection service (Section 3.1). Also, we elaborate on the deployed networks for DeepFake detection and their training process (Section 3.2). We go into detail regarding our

---

micro-service architecture for the service implementation (Section 3.3). Finally, we present the compilation of a Model Card for the service (Section 3.4).

3.1 Processing Pipeline

Once the service receives the link of an image or video as input by the user, the following processing pipeline takes place.

Download Media: The image/video at the URL is identified and downloaded by our custom download module that supports popular file sharing services such as Dropbox[9] and Google Drive[10], as well as social media platforms like YouTube[11] and Twitter[12].

Media Type: If the downloaded resource is an image, then only the *Face Detection* and *Inference* steps that are described below are applied to get the final results. Therefore, the following steps are described below as if the resource is a video.

Video Segmentation: During this step, a video similarity network is used in order to segment the video in multiple shots. We follow the feature extraction and similarity calculation process described in [35]. For an input video, we extract one frame per second and derive their region-level features from a ResNet50 [20] using R-MAC pooling [72]. Then, we calculate the distance between consecutive frames by applying Chamfer Similarity [34] on their region descriptors. Finally, we extract the peaks in the distance plot in order to determine the shot transitions. The detected shots have to be at least 1.5 seconds long. Per shot DeepFake probability scores are also displayed on the front-end, providing the user with useful information about the final video-level prediction.

Face Detection: We apply a pre-trained MTCNN face detection network from [15] to selected sample frames of the video (in the case of images, the face detector is applied once). We sample at most 64 unique frames per shot in order to detect and extract faces. The face detector provides squared bounding boxes that indicate the locations of the faces detected in the input image. To ensure that possible artifacts between the face and background are included, we use a margin value of 1.3, which practically means that we enlarge the detected bounding boxes by 30% per dimension.

Face Clustering: At this stage, the Face Clustering methodology described in [6] is applied to all detected faces of a video shot in order to reduce the noise that is introduced by the falsely detected faces. In more detail, facial embeddings and their similarities are computed per detected face. In that way, we generate a face graph by connecting the faces with similarity greater than 0.8. We then form face clusters by extracting the graph's connected components. We filter out face clusters with only few faces, i.e., less than 20% of the video shot's frames. The remaining faces are further processed.

Inference: Each detected face is resized to 300×300, normalized by the ImageNet [11] mean and standard deviation, and fed to an ensemble scheme that contains five models operating in parallel. See Section 3.2 for details regarding the ensemble model. Subsequently, all five model predictions are averaged to get a DeepFake probability score per input face that ranges in $(0, 1)$.

Video-level Aggregation: The predictions resulting from the above processing steps are at a frame level. In order to derive an aggregated video-level DeepFake probability score, we use the following aggregation strategy:

(1) The face predictions of each face cluster are averaged to generate a cluster prediction.
(2) Shot predictions are derived based on the maximum prediction of their clusters.
(3) The final video-level prediction is the maximum of the shot predictions.

3.2 DeepFake Detection Model

3.2.1 Architecture. The service consists of an ensemble of the following five models with the final DeepFake probability being the ensemble's average probability.

As a backbone network for feature extraction, we used one of the EfficientNet [66, 67] networks. These are CNN models that have been automatically assembled through neural architecture search, based on a compound scaling method that uniformly scales the depth, width, and resolution of the network layers/components. We employ the EfficientNet-b4 [66] and the EfficientNet-V2-m [67]. Additionally, we use the DETR [5] head on top of a backbone for some of our models. This is a Transformer Encoder-Decoder [74] network applied on the region-level activations generated by the backbone to aggregate them with trainable queries equal to the number of the detection classes. Since our problem is binary classification, we use only a single trainable query to derive the final prediction. Also, transformers are usually combined with positional embeddings, which can be fixed or learned. We use both for our models. Overall, we have developed the following models:

(1) **Model 1**: a vanilla EfficientNet-b4 [66],
(2) **Model 2**: a Transformer head based on DETR [5] with *fixed* positional embeddings on top of an EfficientNet-b4 [66],
(3) **Model 3**: a Transformer head based on DETR [5] with *learned* positional embeddings on top of an EfficientNet-b4 [66],
(4) **Model 4**: a multi-head Transformer based on DETR [5] on top of an EfficientNet-b4 [66],
(5) **Model 5**: a vanilla EfficientNet-V2-m [67].

3.2.2 Training process. Models 1-4 were trained on the Facebook DeepFake Detection Challenge (DFDC) dataset [12] while Model 5 was trained on the WildDeepFake (WDF) dataset [84]. For the former models, we used the Adam optimizer [33] with a learning rate of 10^{-4} and 32 batch size for 25 epochs respectively. The networks are initialized with pre-trained weights on ImageNet-1k [11]. For the latter model, we trained an EfficientV2-M-in21k pre-trained on ImageNet-21k [60] and fine-tuned using the Adam optimizer with 10^{-4} learning rate and 32 batch size for 2 epochs. Furthermore, during training, we employ the following augmentations using the Albumentations library [4]: *Geometric augmentations* (Rotate, HorizontalFlip), *Color augmentations* (ColorJitter, ToGray), *Blurring* (MotionBlur, GaussianBlur), *Image Corruption* (ISONoise, CoarseDropout), *External Effects* (RandomSunFlare, RandomRain). Also, for Models 1-4, we used dynamic face augmentations [9].

---

[9]https://dropbox.com
[10]https://drive.google.com
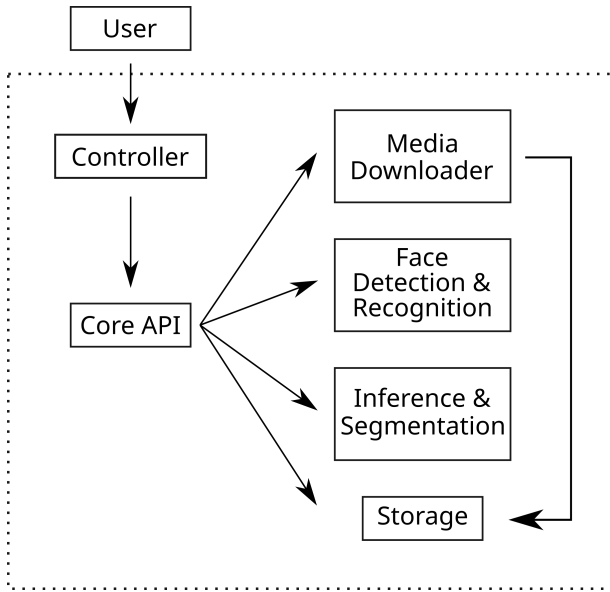[11]https://youtube.com
[12]https://twitter.com

Figure 2: Service Architecture

3.3 Implementation

We implemented our pipeline as a set of micro-services for better modularity and scalability. Each block in Figure 2 corresponds to an independently deployed micro-service.

Our users send requests to the controller, which implements an asynchronous job API with an additional caching layer. Because the DeepFake processing can take a long time to complete, which is especially true for long videos with many segments, each request is assigned a unique job ID which is returned to the client immediately, while in the background, our system starts processing the request. The client uses the returned job ID to monitor the status of their request, and when the processing is complete, they can fetch the results. To provide low latency for repeated queries and to also reduce the strain on our system, results are stored in a Redis [59] cache. We used FastAPI [71] to implement the controller's REST API, and Python-RQ [13] to dispatch and monitor jobs asynchronously.

Each job is implemented as a blocking HTTP call to the core service, which provides a synchronous REST API and orchestrates the necessary computations. First, it dispatches a request to the media download service that is implemented on top of Youtube-dlp [78]. Once the video is successfully downloaded, we use OpenCV [54] to load it and extract frames. The extracted frames are then fed to a segmentation model running on a Triton inference server [53].

Triton is an open-source optimized inference server for executing deep learning models on CPU or GPU. GPU memory management, batching, and model versioning are seamlessly handled. To load our models on Triton, we use the torchscript serialization of PyTorch [57].

After splitting the video into shots, we use the facenet-pytorch library [15] to run face detection and recognition on the frames of each segment. The extracted faces are then clustered based on the calculated face embeddings, and for each cluster component, we

execute the DeepFake ensemble model on Triton. As each shot is independent, their processing is executed in parallel.

Finally, we aggregate the per shot and per cluster DeepFake predictions to calculate the final video-level score. We also generate "gallery plots", i.e. plots that present all keyframes per shot, with each keyframe drawn using a border colored based on its DeepFake score. These plots are fetched from the MinIO object store [48], an open-source S3 compatible storage framework.

3.4 Model Card

We have documented our DeepFake detection service using less formal language in a Model Card format[13]. The model card includes a description of our service's intended use, an account of caveats and recommendations that potential novice users should take into consideration when interpreting the service results, as well as a performance evaluation over three datasets accompanied with a clear explanation of the reported metrics. The compiled model card has been reviewed by AI experts from different disciplines. Based on these reviews, the current version is intended for experts having technical experience, i.e., other researchers working on the problem of DeepFake detection or media verification companies/organizations/groups. Yet, there is room for improvement for other non-technical audiences. The compiled model card is provided in the supplementary materials.

4 EVALUATION

We have evaluated the performance of the presented service across three well-known DeepFake detection datasets as well as using adversarial attacks.

4.1 Evaluation settings

4.1.1 Datasets. We employ three evaluation datasets to assess the performance of our DeepFake Detection service:

- **FaceForensics++ (FF++)** [61] This is organized in two manipulation categories, *Identity Swap*, implemented based on *FaceSwap* and *DeepFakes*, and *Expression Swap*, implemented using *NeuralTextures* and *Face2Face*. FF++ contains 1000 real videos and 4000 fake videos derived by applying the four models on each real video. Evaluation on FF++ provides a performance indicator on different manipulation categories and methods. Compared to more recent datasets (e.g. CelebDF, DFDC) the DeepFake quality in FF++ is visibly worse.
- **CelebDF-V2 (CelebDF)** [42] This comprises videos from celebrity interviews that have been manipulated using improved versions of the DeepFake manipulation methods used in FF++. It consists of 590 real and 5639 fake videos.
- **WildDeepFake (WDF)** [84] In contrast to the above datasets where manipulations were generated by the dataset creators, this contains real-world DeepFakes sourced from various video-sharing websites and their corresponding real versions. It consists of 3800 real and 3500 fake videos. Due to its real-world nature, it is considered a challenging dataset.

4.1.2 Evaluation metrics. Given that the evaluation datasets are imbalanced, we want to avoid skewed metrics that might favor

---

[13]https://mever.iti.gr/deepfake/model_card.pdf

Dataset	MeVer		DeepWare	
	BA	AUC	BA	AUC
FaceForensics++	70.31%	0.7705	68.77%	0.7681
CelebDF	82.75%	0.9259	77.54%	0.9493
WildDeepFake	84.94%	0.9373	66.96%	0.8646

Table 1: BA and AUC for the MeVer service (ours) and Deep-Ware on three datasets.

Manipulation	BA	AUC
FaceSwap	78.40%	0.8674
DeepFakes	86.20%	0.9468
NeuralTextures	57.65%	0.6276
Face2Face	59.02%	0.6402

Table 2: BA and AUC for each manipulation in FF++.

Dataset	norm-1	norm-2	norm-inf
FaceForensics++	70.31%	64.04%	50.53%
CelebDF	82.75%	76.01%	50.00%
WildDeepFake	84.94%	63.04%	50.00%

Table 3: BA on three datasets attacked with the PGD adversarial attack with three output normalization setting.

one class or alter the datasets via sampling. Hence, we choose to report the Balanced Accuracy (BA) rather than raw Accuracy. BA is defined as the mean of the recall computed on each class. Its possible values are in the range 0%-100% (higher is better). Moreover, we report the Area Under the Curve (AUC) as it is the most often used metric in the literature. It is defined as the area under the Receiver Operating Characteristic (ROC) curve with possible values ranging from 0 to 1 (higher is better).

4.1.3 Adversarial robustness. To set up the adversarial robustness evaluation, we used IBM's Adversarial Robustness Toolbox[14] (ART). More specifically, we used ART's *PyTorchClassifier* class to wrap our model ensemble and subsequently used the Projected Gradient Descent adversarial attack [44] attack class. Regarding its hyperparameters, we use $\epsilon = 0.2$ and *max_iterations* = 5, due to computational constraints. To benchmark our model, for each video in each dataset, (i) we feed it to the *PyTorchClassifier* unaltered, (ii) we generate an adversarial example on a frame-by-frame basis, and (iii) feed it through our classifier again.

4.2 Experimental results

4.2.1 Evaluation on different datasets. Table 1 presents our evaluation results on the three aforementioned datasets. Our system performs much better on the CelebDF and WDF datasets rather than the FF++, in which we observe an average 13% performance drop in terms of BA. Specifically, on the WDF dataset, we achieve an 84.94% BA which is close to other state-of-the-art methods [84] and is expected since our ensemble's fifth model was trained on this dataset (see Section 3.2). In the case of the CelebDF dataset, even though none of our models have been trained with this dataset, we achieve an 82.75% BA. Additionally, we compare our system with the publicly available model by DeepWare[15]. We follow the same settings as used for our models for pre-processing, which are slightly different from those used by the original authors. The two systems perform comparably in FF++ and CelebDF, with our service having a small but clear edge. Our system significantly outperforms its competitor in WDF, which is expected since it has been used for training.

4.2.2 Evaluation on different manipulations. To delve into the performance discrepancy between FF++ and the other two datasets, we performed a more extensive evaluation on the FF++ dataset in terms of manipulation type; this is presented in Table 2. Our models are considerably better at detecting *FaceSwap* and *DeepFake* manipulations than *NeuralTextures* and *Face2Face* manipulations.

The former two belong to the *Identity Swap* manipulation category while the latter two are examples of *Expression Swapping* [46]. It can be argued that this is due to our training data lacking *Expression Swapping* examples; therefore, we expect our service to perform better on *Identity Swap* manipulations.

4.2.3 Adversarial robustness. Table 3 illustrates the service performance in terms of its robustness to adversarial attacks with the PGD attack with three different output normalization settings. Even though PGD is a white-box attack, meaning that the attacker would need access to the weights of all the ensemble models, we maintain that in the spirit of Reliable AI, it is preferable to consider such a worst-case scenario. All attacks try to fool the detector into assessing that the input media are real. The norm-1 attack does not have any noticeable effect on the performance in comparison to the original performance from Table 1. However, the norm-2 attack considerably affects the detection accuracy, even though the models still retain decent performance. The strongest norm-inf attack highlights the susceptibility of our model to such attacks as it can no longer distinguish between real and DeepFake videos. Yet, traces of an adversarial attack are visible with the naked eye in images attacked with the norm-inf.
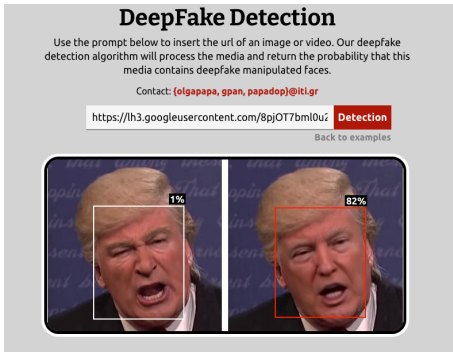
5 DEPLOYING IN THE WILD

Transferring our code from research to production proved to be challenging for multiple reasons. We go through the challenges we faced hoping that our experience will be of use to other academics deploying their research to more real-world settings (Section 5.1). Also, we have built a User Interface for demonstration purposes (Section 5.2). We finally discuss our versioning process (Section 5.3) and some considerations regarding access and availability of the tool (Section 5.4)

5.1 Practical Challenges

During our research, we paid little attention to error handling, and as expected, that was not enough to implement a robust API with helpful explanations when things go wrong. Our micro-service design, however, helped enforce modularity with clear error boundaries. For example, Triton was responsible for GPU-related issues,

---

[14]https://adversarial-robustness-toolbox.readthedocs.io/en/latest/
[15]https://github.com/deepware/deepfake-scanner

Figure 3: Image Analysis User Interface for the DeepFake detection service

separate micro-services were dedicated to downloading, face detection and recognition pipelines, while MinIO [48] offered a storage abstraction. Internally in our APIs, we consistently checked for expected errors and made use of the standard application/problem + JSON content type[16] to propagate them externally when necessary.
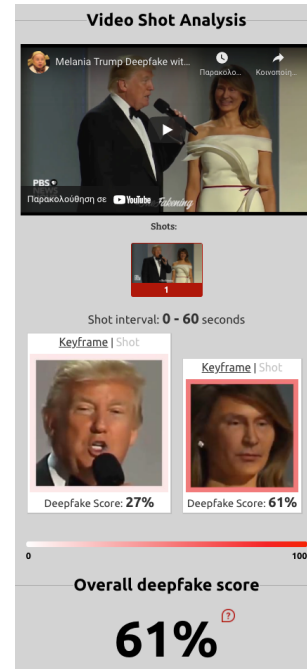
We also faced a number of difficulties with the video downloading process. We tried to support as many video sources as we could, the most popular being YouTube, Twitter, and Facebook. Initially, our download back-end depended on Youtube-dl [77]; however, we experienced very slow download speeds. We averaged around 50KB/s, which for a 10MB video would translate to about three and a half minutes of download time, adding significantly to the total latency. Favoring high-definition video versions - since video quality is one of the most important factors for getting higher accuracy results - would further exacerbate the issue. Fortunately, Youtube-dlp [78], a fork of Youtube-dl, allowed us to consistently achieve much higher download speeds, averaging around 700KB/s. Another complication was that downloading using our public IP could result in additional throttling or even denial of service. For this reason, we chose to run our downloader behind a TOR[17] proxy to provide us with anonymity. Finally, an issue that we faced and still have not resolved is that Facebook downloads are rarely successful without Facebook user authentication.

5.2 User Interface

In terms of User Interface (UI), our service provides two modes: Image and Video analysis. In Image analysis, a DeepFake probability score is presented for each detected face and displayed on top of the face's bounding box as shown in Figure 3. An example of video analysis can be seen in Figure 4. First, an embedded player allows users to playback the video, and second, the UI displays a shot selector as well as a shot interval that the user may choose to inspect the DeepFake probabilities for a specific shot. By default, the initially selected shot is the one with the highest DeepFake probability. Once a shot is selected, the UI displays a window for each detected face accompanied with its corresponding DeepFake probability. The window provides the "Keyframe" (Figure 5) and "Shot" (default) views which show a selected frame and a collage

---

[16]https://datatracker.ietf.org/doc/rfc7807
[17]https://torproject.org

Figure 4: Video Analysis User Interface for the DeepFake detection service

of frames from the shot, respectively. Furthermore, the window allows the user to use the "Hover-to-Zoom" functionality for closer inspection of each view, as seen in Figure 6. Last, the "Overall" DeepFake score is displayed at the bottom of the page, which results from applying the aggregation strategy described in Section 3.1.

5.3 Versioning

As with every software project, it is important to have a clear versioning scheme. For the presented service, we have decided to use an adaptation of the Semantic Versioning 2 scheme[18]. In particular, we follow the $x.y.z$ scheme where:
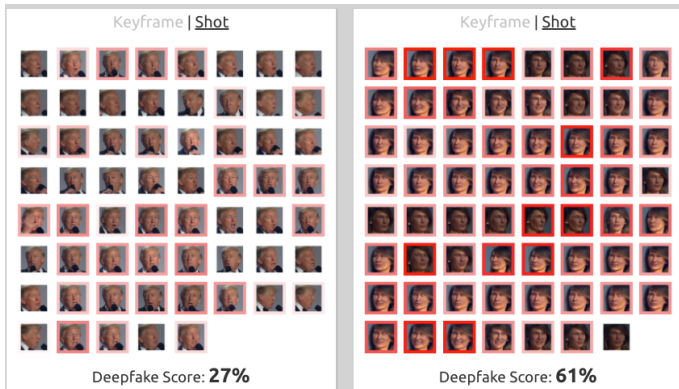
- x: is used for backward-incompatible changes, i.e., change of the output of the service.
- y: refers to changes in the processing pipeline such as the video segmentation methodology, the deployed models used, and others as described in Section 3.1,
- z: is reserved for minor changes such as the aggregation strategy or changes in the model's input dimensions and minor bug fixes.

5.4 Availability

We have chosen to keep the code and the model's weights private and to require user credentials for granting access to our UI. This is due to two main reasons. First, we argue that since our models are vulnerable to adversarial attacks - as shown in Section 4.2.3 - it is essential to protect the service from white-box attacks that would be very easy if the internals were public. We believe that this action

---

[18]https://semver.org/

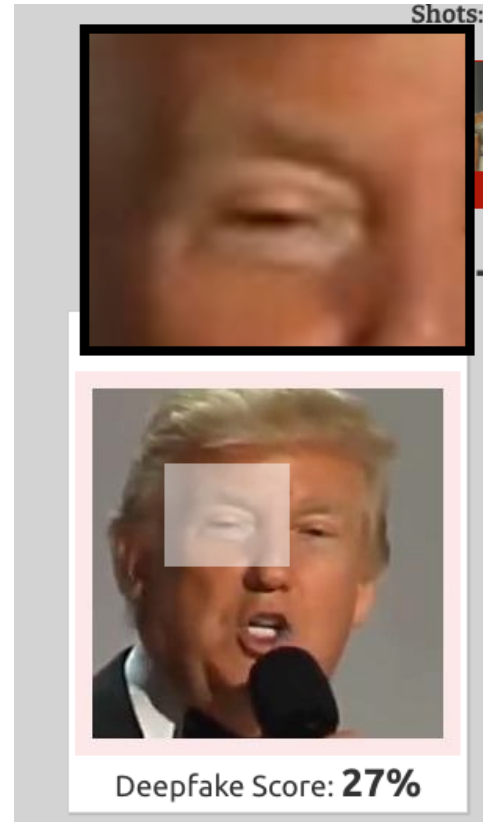**Figure 5: Frame collage view from the selected shot during Video analysis**

will deter the large majority of malicious actors from performing white-box adversarial attacks. Second, given the scarcity of publicly accessible DeepFake detection services, we lack the computational resources to handle the potentially high traffic from end users, and other complications including Denial of Service (DoS) attacks. For the above reasons, we grant access to the service and UI only to trusted partners upon request. Finally, our service has been integrated and is accessible through the InViD-WeVerify Verification plugin [69] (for approved users) and the Truly Media[19] application.

6 CONCLUSION

In this work, we introduced the MeVer DeepFake detection service, a complex multi-model system that detects DeepFake videos and images. We discussed the overall processing pipeline, including a number of pre-processing steps. Also, we presented the model architectures and training processes for the deployed models as well as implementation details for the service. The service has been evaluated on three well-known datasets: FaceForensics++, Celeb-DF, and WildDeepFake. For Celeb-DF and WildDeepFake, our service performed robustly and better compared to the publicly available DeepWare model. For FF++, evaluation by manipulation type revealed that our service performed robustly only in Identity Swap manipulations. In the spirit of *Trustworthy AI*, we also performed an Adversarial Robustness evaluation, and we provided a model card for the service. From the results of the adversarial evaluation, we observe a vulnerability to the *Projected Gradient Descent* attack, which opens new directions for future research. Last but not least, we discuss at length the practical challenges we faced moving from a research codebase to a real-world system that would need to be robust to arbitrary media content from the Internet.

In the future, we plan to improve the detection accuracy by continuously employing the most recent advancements in the field, i.e., by using better datasets for training and evaluation and using state-of-the-art model architectures. In addition, we plan to experiment with various promising Deep Learning architectures and training techniques in order to keep up with the ever-increasing visual quality of DeepFakes. Also, we plan to enhance our service

---
[19] https://www.truly.media/

Figure 6: Hover-to-Zoom functionality in Image and Video analysis

with methods for the detection of fully synthetically generated faces (e.g. based on StyleGAN3 [25]), which is not supported in the current version. Furthermore, we plan to compile more versions of model cards targeted at wider non-technical audiences, e.g., journalists or business managers. Last but not least, we are committed to maintaining our service as well as improving on the underlying API and User Interface.

ACKNOWLEDGEMENTS

REFERENCES

[1] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. 2019. MesoNet: A compact facial video forgery detection network. *10th IEEE International Workshop on Information Forensics and Security, WIFS 2018* (2019). https://doi.org/10.1109/WIFS.2018.8630761 ISBN: 9781538665367.

[2] Hojjat Aghakhani, Dongyu Meng, Yu-Xiang Wang, Christopher Kruegel, and Giovanni Vigna. 2021. Bullseye polytope: A scalable clean-label poisoning attack with improved transferability. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 159–178.

[3] Nicolò Bonettini, Luca Bondi, Edoardo Daniele Cannas, Paolo Bestagini, Sara Mandelli, and Stefano Tubaro. 2020. Video face manipulation detection through ensemble of CNNs. *Proceedings - International Conference on Pattern Recognition* (2020), 5012–5019. https://doi.org/10.1109/ICPR48806.2021.9412711 ISBN: 9781728188089.

[4] Alexander Buslaev, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin. 2020. Albumentations: Fast and Flexible Image Augmentations. *Information* 11, 2 (Feb. 2020), 125. https://doi.org/10.3390/info11020125 Number: 2 Publisher: Multidisciplinary Digital Publishing Institute.

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *European conference on computer vision.* Springer, 213–229.

[6] Polychronis Charitidis, Giorgos Kordopatis-Zilos, Symeon Papadopoulos, and Ioannis Kompatsiaris. 2020. Investigating the impact of pre-processing and prediction aggregation on the deepfake detection task. *Proceedings of the 2020 Truth and Trust Online* (2020), 44–54.

[7] François Chollet. 2017. Xception: Deep learning with depthwise separable convolutions. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* 2017-Janua (2017), 1800–1807. https://doi.org/10.1109/CVPR.2017.195 ISBN: 9781538604571.

[8] Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. 2021. Label-only membership inference attacks. In *International Conference on Machine Learning.* PMLR, 1964–1974.

[9] Sowmen Das, Selim Seferbekov, Arup Datta, Md Islam, Md Amin, et al. 2021. Towards solving the deepfake problem: An analysis on improving deepfake detection using dynamic face augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 3776–3785.

[10] Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de las Casas, Craig Donner, Leslie Fritz, Cristian Galperti, Andrea Huber, James Keeling, Maria Tsimpoukelli, Jackie Kay, Antoine Merle, Jean-Marc Moret, Seb Noury, Federico Pesamosca, David Pfau, Olivier Sauter, Cristian Sommariva, Stefano Coda, Basil Duval, Ambrogio Fasoli, Pushmeet Kohli, Koray Kavukcuoglu, Demis Hassabis, and Martin Riedmiller. 2022. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature* 602, 7897 (Feb. 2022), 414–419. https://doi.org/10.1038/s41586-021-04301-9 Number: 7897 Publisher: Nature Publishing Group.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition.* 248–255. https://doi.org/10.1109/CVPR.2009.5206848 ISSN: 1063-6919.

[12] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton Ferrer. 2020. The DeepFake Detection Challenge (DFDC) Dataset. (2020). arXiv: 2006.07397.

[13] Vincent Driessen. 2021. Python-RQ. https://python-rq.org/docs/.

[14] Mengnan Du, Shiva Pentyala, Yuening Li, and Xia Hu. 2020. Towards generalizable deepfake detection with locality-aware autoencoder. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management.* 325–334.

[15] Tim Esler. 2022. Face Recognition Using Pytorch. https://github.com/timesler/facenet-pytorch original-date: 2019-05-25T01:29:24Z.

[16] Joel Frank and Lea Schönherr. 2021. WaveFake: A data set to facilitate audio DeepFake detection. https://doi.org/10.5281/zenodo.5642694 Type: dataset.

[17] Apurva Gandhi and Shomik Jain. 2020. Adversarial perturbations fool deepfake detectors. In *2020 international joint conference on neural networks (IJCNN).* IEEE, 1–8.

[18] Sam Gregory. 2021. Deepfakes, misinformation and disinformation and authenticity infrastructure responses: Impacts on frontline witnessing, distant witnessing, and civic journalism. *Journalism* (2021), 14648849211060644.

[19] David Güera and Edward J. Delp. 2018. Deepfake Video Detection Using Recurrent Neural Networks. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS).* 1–6. https://doi.org/10.1109/AVSS.2018.8639163

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 770–778.

[21] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. 2019. Attgan: Facial attribute editing by only changing what you want. *IEEE transactions on image processing* 28, 11 (2019), 5464–5478.

[22] Jiajun Huang, Xueyu Wang, Bo Du, Pei Du, and Chang Xu. 2021. DeepFake MNIST+: A DeepFake Facial Animation Dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 1973–1982.

[23] Matthew Jagielski, Nicholas Carlini, David Berthelot, Alex Kurakin, and Nicolas Papernot. 2020. High accuracy and high fidelity extraction of neural networks. In *29th USENIX Security Symposium (USENIX Security 20).* 1345–1362.

[24] Anubhav Jain and Pavel Korshunov. 2021. Improving Generalization of Deepfake Detection by Training for Attribution. (2021). ISBN: 9781665432887.

[25] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2021. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems* 34 (2021).

[26] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020. Analyzing and Improving the Image Quality of StyleGAN. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* 8107–8116. https://doi.org/10.1109/CVPR42600.2020.00813 ISSN: 2575-7075.

[27] Samar Samir Khalil, Sherin M. Youssef, and Sherine Nagy Saleh. 2021. A Multi-Layer Capsule-based Forensics Model for Fake Detection of Digital Visual Media. In *2020 International Conference on Communications, Signal Processing, and their Applications (ICCSPA).* 1–6. https://doi.org/10.1109/ICCSPA49915.2021.9385719

[28] Sohail Ahmed Khan and Hang Dai. 2021. Video Transformer for Deepfake Detection with Incremental Learning. In *Proceedings of the 29th ACM International Conference on Multimedia.* 1821–1828.

[29] Ali Khodabakhsh, Raghavendra Ramachandra, Kiran Raja, Pankaj Wasnik, and Christoph Busch. 2018. Fake Face Detection Methods: Can They Be Generalized?. In *2018 International Conference of the Biometrics Special Interest Group, BIOSIG 2018.* https://doi.org/10.23919/BIOSIG.2018.8553251

[30] Aminollah Khormali and Jiann Shiun Yuan. 2021. Add: Attention-based deepfake detection approach. *Big Data and Cognitive Computing* 5, 4 (2021). https://doi.org/10.3390/bdcc5040049

[31] Minha Kim, Shahroz Tariq, and Simon S. Woo. 2021. *CoReD: Generalizing Fake Media Detection with Continual Representation using Distillation.* Vol. 1. Association for Computing Machinery. https://doi.org/10.1145/3474085.3475535

[32] Minha Kim, Shahroz Tariq, and Simon S Woo. 2021. Fretal: Generalizing deepfake detection using knowledge distillation and representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 1001–1012.

[33] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.).

[34] Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Ioannis Kompatsiaris. 2019. Visil: Fine-grained spatio-temporal video similarity learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 6351–6360.

[35] Giorgos Kordopatis-Zilos, Christos Tzelepis, Symeon Papadopoulos, Ioannis Kompatsiaris, and Ioannis Patras. 2021. DnS: Distill-and-Select for Efficient and Accurate Video Indexing and Retrieval. *CoRR* abs/2106.13266 (2021). arXiv:2106.13266

[36] Pavel Korshunov and Sebastien Marcel. 2018. DeepFakes a New Threat to Face Recognition? Assessment and Detection. (2018), 1–5. arXiv: 1812.08685.

[37] Patrick Kwon, Jaeseong You, Gyuhyeon Nam, Sungwoo Park, and Gyeongsu Chae. 2021. Kodf: A large-scale korean deepfake detection dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 10744–10753.

[38] Trung-Nghia Le, Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen. 2022. Robust Deepfake On Unrestricted Media: Generation And Detection. *CoRR* abs/2202.06228 (2022). arXiv:2202.06228

[39] Trung-Nghia Le, Huy H Nguyen, Junichi Yamagishi, and Isao Echizen. 2021. Openforensics: Large-scale challenging dataset for multi-face forgery detection and segmentation in-the-wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 10117–10127.

[40] Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo. 2020. Face X-ray for more general face forgery detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2020), 5000–5009. https://doi.org/10.1109/CVPR42600.2020.00505

[41] Yuezun Li and Siwei Lyu. 2019. Exposing DeepFake Videos By Detecting Face Warping Artifacts. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).*

[42] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. 2020. Celeb-DF: A Large-Scale Challenging Dataset for DeepFake Forensics. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2020), 3204–3213. https://doi.org/10.1109/CVPR42600.2020.00327

[43] Yuezun Li, Cong Zhang, Pu Sun, Lipeng Ke, Yan Ju, Honggang Qi, and Siwei Lyu. 2021. DeepFake-o-meter: An Open Platform for DeepFake Detection. In *2021 IEEE Security and Privacy Workshops (SPW).* IEEE, 277–281.

[44] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations.* https://openreview.net/forum?id=rJzIBfZAb

[45] Asad Malik, Minoru Kuribayashi, Sani M. Abdullahi, and Ahmad Neyaz Khan. 2022. DeepFake Detection for Human Face Images and Videos: A Survey. *IEEE Access* (2022), 1–1. https://doi.org/10.1109/ACCESS.2022.3151186 Conference Name: IEEE Access.

[46] Momina Masood, Marriam Nawaz, Khalid Mahmood Malik, Ali Javed, and Aun Irtaza. 2021. Deepfakes Generation and Detection: State-of-the-art, open challenges, countermeasures, and way forward. (2021). arXiv: 2103.00484.

[47] Akul Mehra, Luuk Spreeuwers, and Nicola Strisciuglio. 2021. Deepfake Detection using Capsule Networks and Long Short-Term Memory Networks:. In

Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications. SCITEPRESS - Science and Technology Publications, Online Streaming, — Select a Country —, 407–414. https://doi.org/10.5220/0010289004070414

[48] MinIO. 2022. MinIO. https://min.io.

[49] Yisroel Mirsky and Wenke Lee. 2021. The Creation and Detection of Deepfakes. *Comput. Surveys* 54, 1 (2021), 1–41. https://doi.org/10.1145/3425780

[50] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model Cards for Model Reporting. *Proceedings of the Conference on Fairness, Accountability, and Transparency* (Jan. 2019), 220–229. https://doi.org/10.1145/3287560.3287596

[51] Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen. 2019. Capsule-forensics: Using Capsule Networks to Detect Forged Images and Videos. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* 2019-May (2019), 2307–2311. https://doi.org/10.1109/ICASSP.2019.8682602 ISBN: 9781479981311.

[52] Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen. 2022. Capsule-Forensics Networks for Deepfake Detection. In *Handbook of Digital Face Manipulation and Detection: From DeepFakes to Morphing Attacks*, Christian Rathgeb, Ruben Tolosana, Ruben Vera-Rodriguez, and Christoph Busch (Eds.). Springer International Publishing, Cham, 275–301. https://doi.org/10.1007/978-3-030-87664-7_13

[53] NVIDIA. 2022. NVIDIA TRITON INFERENCE SERVER. https://github.com/triton-inference-server/server.

[54] OpenCV-team. 2022. OpenCV. https://opencv.org.

[55] Leandro A. Passos, Danilo Jodas, Kelton A. P. da Costa, Luis A. Souza Júnior, Danilo Colombo, and João Paulo Papa. 2022. A Review of Deep Learning-based Approaches for Deepfake Content Detection. *arXiv:2202.06095 [cs]* (Feb. 2022).

[56] Jiameng Pu, Neal Mangaokar, Lauren Kelly, Parantapa Bhattacharya, Kavya Sundaram, Mobin Javed, Bolun Wang, and Bimal Viswanath. 2021. *Deepfake videos in the wild: Analysis and detection*. Vol. 1. Association for Computing Machinery. https://doi.org/10.1145/3442381.3449978 Publication Title: The Web Conference 2021 - Proceedings of the World Wide Web Conference, WWW 2021 Issue: 1.

[57] PyTorch. 2022. TorchScript. https://pytorch.org/docs/stable/jit.html.

[58] Qing Rao and Jelena Frtunikj. 2018. Deep learning for self-driving cars: chances and challenges. In *Proceedings of the 1st International Workshop on Software Engineering for AI in Autonomous Systems*. ACM, Gothenburg Sweden, 35–38. https://doi.org/10.1145/3194085.3194087

[59] Redis. 2021. Redis. https://redis.io/.

[60] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. 2021. ImageNet-21K Pretraining for the Masses. *arXiv:2104.10972 [cs]* (Aug. 2021).

[61] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Niessner. 2019. FaceForensics++: Learning to detect manipulated facial images. *Proceedings of the IEEE International Conference on Computer Vision* 2019-Octob (2019), 1–11. https://doi.org/10.1109/ICCV.2019.00009

[62] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. *Advances in neural information processing systems* 30 (2017).

[63] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. 2020. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9243–9252.

[64] Yuyang Sun, Zhiyong Zhang, Changzhen Qiu, Liang Wang, and Zekai Wang. 2021. FakeTransformer: Exposing Face Forgery From Spatial-Temporal Representation Modeled By Facial Pixel Variations. (2021). arXiv: 2111.07601.

[65] Supasorn Suwajanakorn, Steven M. Seitz, and Ira Kemelmacher-Shlizerman. 2017. Synthesizing Obama: learning lip sync from audio. *ACM Trans. Graph.* 36, 4 (July 2017), 1–13. https://doi.org/10.1145/3072959.3073640

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

[67] Mingxing Tan and Quoc Le. 2021. Efficientnetv2: Smaller models and faster training. In *International Conference on Machine Learning*. PMLR, 10096–10106.

[68] Shahroz Tariq, Sangyup Lee, and Simon Woo. 2021. One detector to rule them all: Towards a general deepfake attack detection framework. *The Web Conference 2021 - Proceedings of the World Wide Web Conference, WWW 2021* (2021), 3625–3637. https://doi.org/10.1145/3442381.3449809 ISBN: 9781450383127.

[69] Denis Teyssou, Jean-Michel Leung, Evlampios Apostolidis, Konstantinos Apostolidis, Symeon Papadopoulos, Markos Zampoglou, Olga Papadopoulou, and Vasileios Mezaris. 2017. The InVID plug-in: web video verification on the browser. In *Proceedings of the first international workshop on multimedia verification*. 23–30.

[70] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. 2016. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2387–2395.

[71] tiangolo. 2022. FastAPI. https://fastapi.tiangolo.com/.

[72] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. 2016. Particular object retrieval with integral max-pooling of CNN activations. In *Proceedings of the International Conference on Learning Representations*.

[73] Van Nhan Tran, Suk Hwan Lee, Hoanh Su Le, and Ki Ryong Kwon. 2021. High performance deepfake video detection on cnn-based with attention target-specific regions and manual distillation extraction. *Applied Sciences (Switzerland)* 11, 16 (2021). https://doi.org/10.3390/app11167678

[74] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[75] Ying Xu, Kiran Raja, Raghavendra Ramachandra, and Christoph Busch. 2022. Adversarial Attacks on Face Recognition Systems. In *Handbook of Digital Face Manipulation and Detection: From DeepFakes to Morphing Attacks*, Christian Rathgeb, Ruben Tolosana, Ruben Vera-Rodriguez, and Christoph Busch (Eds.). Springer International Publishing, Cham, 139–161. https://doi.org/10.1007/978-3-030-87664-7_7

[76] Xin Yang, Yuezun Li, and Siwei Lyu. 2019. Exposing Deep Fakes Using Inconsistent Head Poses. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* 2019-May (2019), 8261–8265. https://doi.org/10.1109/ICASSP.2019.8683164 ISBN: 9781479981311.

[77] yt dl. 2022. Youtube-DL. https://youtube-dl.org.

[78] yt dlp. 2022. Youtube-DLP. https://github.com/yt-dlp/yt-dlp.

[79] Bo Zhao, Shaozeng Zhang, Chunxue Xu, Yifan Sun, and Chengbin Deng. 2021. Deep fake geography? When geospatial data encounter Artificial Intelligence. *Cartography and Geographic Information Science* 48, 4 (2021), 338–352.

[80] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. 2021. Multi-attentional deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2185–2194.

[81] Tianchen Zhao, Xiang Xu, Mingze Xu, Hui Ding, Yuanjun Xiong, and Wei Xia. 2021. Learning self-consistency for deepfake detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 15023–15033.

[82] Tianfei Zhou, Wenguan Wang, Zhiyuan Liang, and Jianbing Shen. 2021. Face forensics in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5778–5788.

[83] Yuhao Zhu, Qi Li, Jian Wang, Cheng-Zhong Xu, and Zhenan Sun. 2021. One shot face swapping on megapixels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4834–4844.

[84] Bojia Zi, Minghao Chang, Jingjing Chen, Xingjun Ma, and Yu Gang Jiang. 2020. WildDeepfake: A Challenging Real-World Dataset for Deepfake Detection. *MM 2020 - Proceedings of the 28th ACM International Conference on Multimedia* (2020), 2382–2390. https://doi.org/10.1145/3394171.3413769 ISBN: 9781450379885.