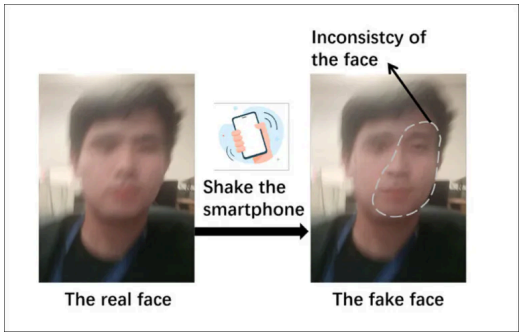
Detecting Video-conference Deepfakes With a Smartphone's 'Vibrate' Function

Originally published at https://www.unite.ai/detecting-video-conference-deepfakes-with-a-smartphones-vibrate-function/

New research from Singapore has proposed a novel method of detecting whether someone on the other end of a smartphone videoconferencing tool is using methods such as DeepFaceLive to impersonate someone else.

Titled *SFake*, the new approach abandons the passive methods employed by most systems, and causes the user's phone *to vibrate* (using the same 'vibrate' mechanisms common across smartphones), and subtly blur their face.

Though live deepfaking systems are variously capable of replicating motion blur, so long as blurred footage was included in the training data, or at least in the pre-training data, they cannot respond quickly enough to unexpected blur of this kind, and continue to output non-blurred sections of faces, revealing the existence of a deepfake conference call.



DeepFaceLive cannot respond quickly enough to simulate the blur caused by the camera vibrations. Source: https://arxiv.org/pdf/2409.10889v1

Test results on the researchers' self-curated dataset (since no datasets featuring active camera shake exist) found that SFake outperformed competing video-based deepfake detection methods, even when faced with challenging circumstances, such as the natural hand movement the occurs when the other person in a videoconference is holding the camera with their hand, instead of using a static phone mount.

The Growing Need for Video-Based Deepfake Detection

Research into video-based deepfake detection has increased recently. In the wake of several years' worth of successful voice-based deepfake heists, earlier this year a finance worker was tricked into transferring $25 million dollars to a fraudster who was impersonating a CFO in a deepfaked video conference call.

Though a system of this nature requires a high level of hardware access, many smartphone users are already accustomed to financial and other types of verification services asking us to record our facial characteristics for face-based authentication (indeed, this is even part of LinkedIn's verification process).

It therefore seems likely that such methods will increasingly become enforced for videoconferencing systems, as this type of crime continues to make headlines.

Most solutions that address real-time videoconference deepfaking assume a very static scenario, where the communicant is using a stationary webcam, and no movement or excessive environmental or lighting changes are expected. A smartphone call offers no such 'fixed' situation.

Instead, SFake uses a number of detection methods to compensate for the high number of visual variants in a hand-held smartphone-based videoconference, and appears to be the first research project to address the issue by use of standard vibration equipment built into smartphones.

The paper is titled *Shaking the Fake: Detecting Deepfake Videos in Real Time via Active Probes*, and comes from two researchers from the Nanyang Technological University at Singapore.

Method

SFake is designed as a cloud-based service, where a local app would send data to a remote API service to be processed, and the results sent back.

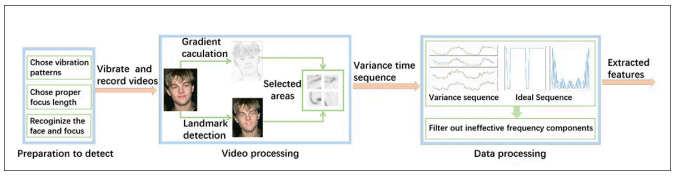
However, its mere 450mb footprint and optimized methodology allows that it could process deepfake detection entirely on the device itself, in cases where network connection could cause sent images to become excessively compressed, affecting the diagnostic process.

Running 'all local' in this manner means that the system would have direct access to the user's camera feed, without the codec interference often associated with videoconferencing.

Average analysis time requires a four-seconds video sample, during which the user is asked to remain still, and during which SFake sends 'probes' to cause camera vibrations to occur, at selectively random intervals that systems such as DeepFaceLive cannot respond to in time.

(It should be re-emphasized that any attacker that has not included blurred content in the training dataset is unlikely to be able to produce a model that can generate blur even under much more favorable circumstances, and that DeepFaceLive cannot just 'add' this functionality to a model trained on an under-curated dataset)

The system chooses select areas of the face as areas of potential deepfake content, excluding the eyes and eyebrows (since blinking and other facial motility in that area is outside of the scope of blur detection, and not an ideal indicator).



Conceptual schema for SFake.

As we can see in the conceptual schema above, after choosing apposite and non-predictable vibration patterns, settling on the best focal length, and performing facial recognition (including landmark detection via a Dlib component which estimates a standard 68 facial landmarks), SFake derives gradients from the input face and concentrates on selected areas of these gradients.

The variance sequence is obtained by sequentially analyzing each frame in the short clip under study, until the average or 'ideal' sequence is arrived at, and the rest disregarded.

This provides extracted features that can be used as a quantifier for the probability of deepfaked content, based on the trained database (of which, more momentarily).

The system requires an image resolution of 1920×1080 pixels, as well as at least a 2x zoom requirement for the lens. The paper notes that such resolutions (and even higher resolutions) are supported in Microsoft Teams, Skype, Zoom, and Tencent Meeting.

Most smartphones have a front-facing and self-facing camera, and often only one of these has the zoom capabilities required by SFake; the app would therefore require the communicant to use whichever of the two cameras meets these requirements.

The objective here is to get a *correct proportion* of the user's face into the video stream that the system will analyze. The paper observes that the average distance that women use mobile devices is 34.7cm, and for men, 38.2cm (as reported in *Journal of Optometry*), and that SFake operates very well at these distances.

Since stabilization is an issue with hand-held video, and since the blur that occurs from hand movement is an impediment to the functioning of SFake, the researchers tried several methods to compensate. The most successful of these was calculating the central point of the estimated landmarks and using this as an 'anchor' – effectively an algorithmic stabilization technique. By this method, an accuracy of 92% was obtained.

Data and Tests

As no apposite datasets existed for the purpose, the researchers developed their own:

> *'[We] use 8 different brands of smartphones to record 15 participants of varying genders and ages to build our own dataset. We place the smartphone on the phone holder 20 cm away from the participant and zoom in twice, aiming at the participant's face to encompass all his facial features while vibrating the smartphone in different patterns.*
>
> *'For phones whose front cameras cannot zoom, we use the rear cameras as a substitute. We record 150 long videos, each 20 seconds in duration. By default, we assume the detection period lasts 4 seconds. We trim 10 clips of 4 seconds long from one long video by randomizing the start time. Therefore, we get a total of 1500 real clips, each 4 seconds long.'*

Though DeepFaceLive (GitHub link) was the central target of the study, since it is currently the most widely-used open source live deepfaking system, the researchers included four other methods to train their base detection model: Hififace; FS-GANV2; RemakerAI; and MobileFaceSwap – the last of these a particularly appropriate choice, given the target environment.

1500 faked videos were used for training, along with the equivalent number of real and unaltered videos.

SFake was tested against several different classifiers, including SBI; FaceAF; CnnDetect; LRNet; DefakeHop variants; and the free online deepfake detection service Deepaware. For each of these deepfake methods, 1500 fake and 1500 real videos were trained.

For the base test classifier, a simple two-layer neural network with a ReLU activation function was used. 1000 real and 1000 fake videos were randomly chosen (though the fake videos were exclusively DeepFaceLive examples).

Area Under Receiver Operating Characteristic Curve (AUC/AUROC) and Accuracy (ACC) were used as metrics.

For training and inference, a NVIDIA RTX 3060 was used, and the tests run under Ubuntu. The test videos were recorded with a Xiaomi Redmi 10x, a Xiaomi Redmi K50, an OPPO Find x6, a Huawei Nova9, a Xiaomi 14 Ultra, an Honor 20, a Google Pixel 6a, and a Huawei P60.

To accord with existing detection methods, the tests were implemented in PyTorch. Primary test results are illustrated in the table below:

Methods	SBI [59]		FaceAF [37]		CnnDetect [65]		LRNet [62]		DFHob [11]		Dware [16]		Ours	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
MFS	0.862	0.885	0.625	0.655	0.784	0.843	0.848	0.891	0.766	0.792	0.832	0.873	**0.956**	**0.962**
HFace	**0.964**	**0.971**	0.802	0.860	0.909	0.933	0.892	0.946	0.822	0.869	0.928	0.942	0.954	0.972
Fsgan	0.815	0.867	0.574	0.627	0.736	0.791	0.757	0.802	0.697	0.775	0.832	0.864	**0.952**	**0.968**
DFL	0.842	0.926	0.677	0.713	0.763	0.785	0.826	0.853	0.822	0.878	0.742	0.796	**0.988**	**0.999**
RAI	0.820	0.841	0.652	0.689	0.745	0.792	0.681	0.748	0.773	0.815	0.765	0.813	**0.968**	**0.976**

Results for SFake against competing methods.

Here the authors comment:

> *'In all cases, the detection accuracy of SFake exceeded 95%. Among the five deepfake algorithms, except for Hififace, SFake performs better against other deepfake algorithms than the other six detection methods. As our classifier is trained using fake images generated by DeepFaceLive, it reaches the highest accuracy rate of 98.8% when detecting DeepFaceLive.*
>
> *'When facing fake faces generated by RemakerAI, other detection methods perform poorly. We speculate this may be because of the automatic compression of videos when downloading from the internet, resulting in the loss of image details and thereby reducing the detection accuracy. However, this does not affect the detection by SFake which achieves an accuracy of 96.8% in detection against RemakerAI.'*

The authors further note that SFake is the most performant system in the scenario of a 2x zoom applied to the capture lens, since this exaggerates movement, and is an incredibly challenging prospect. Even in this situation, SFake was able to achieve recognition accuracy of 84% and 83%, respectively for 2.5 and 3 magnification factors.

Conclusion

A project that uses the weaknesses of a live deepfake system against itself is a refreshing offering in a year where deepfake detection has been dominated by papers that have merely stirred up venerable approaches around frequency analysis (which is far from immune to innovations in the deepfake space).

At the end of 2022, another system used monitor brightness variance as a detector hook; and in the same year, my own demonstration of DeepFaceLive's inability to handle hard 90-degree profile views gained some community interest.

DeepFaceLive is the correct target for such a project, as it is almost certainly the focus of criminal interest in regard to videoconferencing fraud.

However, I have lately seen some anecdotal evidence that the LivePortrait system, currently very popular in the VFX community, handles profile views much better than DeepFaceLive; it would have been interesting if it could have been included in this study.

First published Tuesday, September 24, 2024

Writer on machine learning, domain specialist in human image synthesis. Former head of research content at Metaphysic.ai, until its dissolution into DNEG's Brahma.ai.
Portfolio site: martinanderson.ai
Contact: martin@martinanderson.ai

Discover More