FakeBuster: A DeepFakes Detection Tool for Video Conferencing Scenarios

Vineet Mehta  
Indian Institute of Technology Ropar  
2016CSB1063@iitrpr.ac.in

Parul Gupta  
Indian Institute of Technology Ropar  
2016CSB1048@iitrpr.ac.in

Ramanathan Subramanian  
Indian Institute of Technology Ropar  
s.ramanathan@iitrpr.ac.in

Abhinav Dhall  
Monash University  
Indian Institute of Technology Ropar  
abhinav.dhall@monash.edu

ABSTRACT
This paper proposes FakeBuster, a novel DeepFake detector for (a) detecting impostors during video conferencing, and (b) manipulated faces on social media. FakeBuster is a standalone deep learning-based solution, which enables a user to detect if another person's video is manipulated or spoofed during a video conference-based meeting. This tool is independent of video conferencing solutions and has been tested with Zoom and Skype applications. It employs a 3D convolutional neural network for predicting video fakeness. The network is trained on a combination of datasets such as Deepforensics, DFDC, VoxCeleb, and deepfake videos created using locally captured images (specific to video conferencing scenarios). Diversity in the training data makes FakeBuster robust to multiple environments and facial manipulations, thereby making it generalizable and ecologically valid.

CCS CONCEPTS
• Computing methodologies → Computer vision problems;
• Applied computing;

KEYWORDS
Deepfakes detection, spoofing, neural networks

ACM Reference Format:
Vineet Mehta, Parul Gupta, Ramanathan Subramanian, and Abhinav Dhall. 2021. FakeBuster: A DeepFakes Detection Tool for Video Conferencing Scenarios. In 26th International Conference on Intelligent User Interfaces (IUI '21 Companion), April 14–17, 2021, College Station, TX, USA. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3397482.3450726

1 INTRODUCTION AND BACKGROUND
Sophisticated artificial intelligence techniques have spurred a dramatic increase in manipulated media content, which keep evolving and becoming more realistic, making detection increasingly difficult. While their usage in spreading fake news, pornography and other such online content has been widely observed with major repercussions [7], they have recently found their way into video-calling platforms through spoofing tools based on facial performance transfer [17]. Facial transfer enables individuals to mimic others through real-time transfer of facial expressions, and the resulting videos (known as deepfakes) are often convincing to the human eye. This may have serious implications, especially in a pandemic situation, where virtual meetings are primarily employed for personal and professional communication.

There are a few deepfake detection software such as the recently introduced Video Authenticator by Microsoft [4]. However, these can only verify genuineness of pre-recorded videos. Also, Microsoft's tool is not publicly available to prevent its misuse. As an alternative, we present FakeBuster - a deepfake detection tool, which works in both offline (for existing videos) and online (during video conferencing) modes. A snapshot of the tool being used during a Zoom meeting is shown in Figure 1.

Additional Use Cases: Tools for deepfake detection can help in identification of impostors during events such as online examinations, video-based authentication and job interviews. Organisations can use deepfakes detection tools to ensure the legitimacy of a candidate. The same tool can also be used to validate any media content seen online by the user on social media platforms such as YouTube and Twitter.

2 FAKEBUSTER
As seen in Fig. 1, FakeBuster has a compact interface designed for ease of use alongside video-calling applications and internet browsers. It has been developed using PyQt toolkit [2] which offers the following advantages: a) It is a Python binding of QT [1], which is a cross-platform application development framework, and b) use of Python language enables flexible integration of deep learning models. We have used the python MSS [3] library for screen recording, OpenCV [6] for image processing, and Pytorch [16] to train and test the deep learning models. The standalone aspect of the tool enables its usage with different video conferencing tools such as Zoom, Skype, Webex etc.

2.1 Workflow
The tool works in three steps (Figure 2): a) The user clicks on “Detect Faces”, and the tool detects and shows all the faces present on the screen. In case of change in the video conference software’s view
Figure 1: FakeBuster tool enables impostor detection in online meetings. Here, a user’s Zoom video feed is detected as manipulated by FakeBuster. Please see FakeBuster demo video in the supplementary material.

(a) Step 1 - Face Select
(b) Step 2 - Find Imposter
(c) Step 3 - Analysis

Figure 2: Workflow of FakeBuster - a) User selects the faces detected by the tool from videoconferencing tool; b) User clicks on find impostor and the tool does prediction at snippet level; and c) User views the meta-analysis of fakeness of the video feed.

or positioning, the examined faces can be re-initialized by clicking “Detect Faces”. The user then selects a face icon, whose video needs to validated; b) The user next clicks on the “Find Imposter” button to initiate deep inference. The tool runs face capturing, frame segmentation, and deepfake prediction for each segment in the background. The time-series graph on the right side of the tool (see Fig. 2 (b)) shows the prediction score at regular intervals. The prediction score varies between the range 0 to 1, which is color-coded, depicting the extent of face manipulation at a particular instant: green (no manipulation), yellow, orange (some chance of tamper), and red (high probability of manipulation); c) The user can see the overall summary by clicking the “View Summary” button. It opens a new dialog screen (see Fig. 2 (c)), which shows the average imposter score, highest and lowest manipulation intensity, along with the occurrence time stamps. The user can utilize this information to take informed decisions and actions during the virtual meeting.

In the background, FakeBuster initiates impostor detection on the selected face by learning the appearance around the face position. Further, it performs face tracking, face-frame segmentation and deepfake prediction on each video segment in the background. The prediction scores are aggregated, and the time-series graph is updated over time.

2.2 Deepfake Detection

For segment-wise deepfake prediction, we train a 3D ResNet deep learning model [11] based visual stream architecture (Fig. 3), followed by the state-of-the-art deepfake detection proposed by Chugh et al. [8]. The input to the network is a segment of 30 face frames, and the output is the probability of the input chunk being real/fake. A number of large deepfake datasets with accompanying annotations exist such as DFDC [9], Deeperforensics [12] and FakeET [10] consisting of fake videos generated via face-swapping techniques that generate good quality deepfakes. However, one of the successful online deepfake creator tools - Avatarify [5] performs face swapping in video conferencing apps using First Order Motion Model (FOMM) proposed by Siarohin et al.
REFERENCES

[1] 1995. QT: One framework. One codebase. Any platform. https://www.qt.io/

[2] 2016. PyQT. Python bindings for The Qt Company’s Qt application framework. https://riverbankcomputing.com/software/pyqt/intro

[3] 2020. Python MSS- An ultra fast cross-platform multiple screenshots module in pure python using ctypes. https://github.com/BoBoTiG/python-mss

[4] 2020. Reality Defender 2020. A FORCE AGAINST DEEPFAKES. https://rd2020.org/

[5] Ali Aliev. 2019. Avatarify- Photorealistic avatars for video-conferencing apps. https://github.com/aliev/avatarify

[6] G. Bradski. 2000. The OpenCV Library. Dr. Dobb’s Journal of Software Tools (2000).

[7] Tom Bur. 2020. New Steps to Combat Disinformation. https://blogs.microsoft.com/on-the-issues/2020/09/01/disinformation-deepfakes-newsguard-video-authorization/

[8] Komal Chugh, Parul Gupta, Abhinav Dhall, and Ramanathan Subramanian. 2020. Not Made for Each Other- Audio-Visual Dissonance-Based Deepfake Detection and Localization. In Proceedings of the 26th ACM International Conference on Multimedia (Seattle, WA, USA) (MM ’20). Association for Computing Machinery, New York, NY, USA, 439–447. https://doi.org/10.1145/3393191.3413700

[9] Brian Dobanovsky, Russ Howes, Ben Pillaum, Nicole Baram, and Cristian Canton Ferrer. 2019. The Deepfake Detection Challenge (DFDC) Preview Dataset. arXiv:1910.08854 [cs.CV]

[10] Parul Gupta, Komal Chugh, Abhinav Dhall, and Ramanathan Subramanian. 2020. The Eyes Know It: FaceET- An Eye-Tracking Database to Understand Deepfake Perception. In Proceedings of the 2020 International Conference on Multimodal Interaction (Virtual Event, Netherlands) (ICMI ’20). Association for Computing Machinery, New York, NY, USA, 519–527. https://doi.org/10.1145/3382507.3418857

[11] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. 2017. Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet? arXiv:1711.09577 [cs.CV]

[12] Luming Jiang, Ren Li, Wayne Wu, Chen Qian, and Chen Change Loy. 2020. DeepForensics-1.0: A Large-Scale Dataset for Real-World Face Forgery Detection. arXiv:2001.03024 [cs.CV]

[13] Tero Karras, Samuli Laine, and Timo Aila. 2019. A Style-Based Generator Architecture for Generative Adversarial Networks. arXiv:1812.04948 [cs.NE]

[14] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).

[15] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. 2017. VoxCeleb: A Large-Scale Speaker Identification Dataset. InterSpeech 2017 (Aug 2017). https://doi.org/10.21437/interspeech.2017-950

[16] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andres Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Advances in Neural Information Processing Systems 32. Curran Associates, Inc., 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[17] Aliaksandr Siairohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. 2019. First order motion model for image animation. In Advances in Neural Information Processing Systems. 7137–7147.