USENIX Security '25 Artifact Appendix: Seeing Through: Analyzing and Attacking Virtual Backgrounds in Video Calls

Felix Weissberg*
BIFOLD & TU Berlin

Jan Malte Hilgefort*
TU Braunschweig

Steve Grogorick
TU Braunschweig

Daniel Arp
TU Wien

Thorsten Eisenhofer
BIFOLD & TU Berlin

Martin Eisemann
TU Braunschweig

Konrad Rieck
BIFOLD & TU Berlin

A Artifact Appendix

A.1 Abstract

Video calls have become an essential part of remote work, but transmitting video from home risks exposing private details. To address this, video conferencing platforms provide *virtual backgrounds* to conceal the real environment. Unfortunately, this protection is not flawless, and occasional pixel leak from the environment. In this paper, we introduce a reconstruction attack that restores the real surrounding of videos protected by virtual backgrounds. To evaluate the effectiveness of this attack, we develop a testing framework that generates a set of videos with different virtual backgrounds and caller environments based on recordings made in front of a green screen. Within this framework, we use two virtual background implementations from common video conferencing services–MediaPipe (Google Meet, Jitsi, BigBlueButton) and Zoom. We further implement two baselines attacks from Sabra et al. and Hilgefort et al. that serve as a reference for the reconstruction quality of our attack and find that our reconstructions reveal at least 53% more leaked pixels from a video.

A.2 Description & Requirements

A.2.1 Security, privacy, and ethical concerns

There are no expected risks or others ethical concerns when executing the artifact.

A.2.2 How to access

We provide our artifact as a stable archive on Zenodo as well as on Github at commit state `0d1238a0c43e4910d62683ff06a2553da35568ab`.

A.2.3 Hardware dependencies

The artifact does not require specialized hardware, but a GPU can significantly reduce the total execution time.

*Authors contributed equally.

A.2.4 Software dependencies

The artifact requires *apptainer* and *curl* to be installed on the system. Detailed instructions on how to install apptainer on common Linux distributions can be found in the apptainer documentation. The curl package can be installed via the package repository on common Linux distributions.

A.2.5 Benchmarks

Models. The artifact requires the U^{2}Net model to execute the baseline attack from Hilgefort et al. as well as the DeepLabV3 model for the approach from Sabra et al. and our attack. Both models are downloaded automatically by the evaluation scripts.

Data. Furthermore, green screen recordings are required from which the evaluation dataset is constructed. We adhered to best practices when collecting the video recordings for our evaluation. As part of the privacy policy agreed upon by the participants, this included limiting the use of the recordings strictly to the minimum required to conduct the evaluation. This policy also ensures that all recordings are deleted at the latest three years after the recording. Consequently, we do not release the participants' videos. Instead, we provide a sample recording that allows to reproduce the attack's effectiveness. This recording is already contained in the archive provided via Zenodo as well as in the Github repository.

Code. We provide the code necessary to re-run the attack as well as the two approaches from Hilgefort et al. and Sabra et al., and the code for the evaluation of their respective reconstruction performances. However, as described in our *Open Science* statement, due to restrictions in Zoom's terms and conditions, we cannot share the tool used to extract portrait masks from the Zoom client. As a remedy, we directly provide the extracted masks to facilitate reproducing the attack without additional reverse-engineering efforts. These masks are also included in the Zenodo archive and the Github repository.

A.3 Set-up

A.3.1 Installation

The artifact can be obtained as a zip compressed archive on Zenodo or cloned from the Github repository. Provided the necessary software dependencies described in Appendix A.2.4 are installed and the artifact was acquired in either of the two ways, the `scripts/build.sh` script can be run from the artifacts' root directory to set up the environment.

A.3.2 Basic Test

To verify that the setup works without running the attack or any baseline approaches, the script `scripts/check.sh` can be executed from the artifacts' root directory. A successful test ends printing: `[*] Success!`.

A.4 Evaluation workflow

A.4.1 Major Claims

In this paper we introduce a novel approach to reconstructing real surroundings in video calls where the environment is concealed with a virtual background. We compare our approach to two previous works from Hilgefort et al. and Sabra et al. and claim that our proposed attack significantly outperforms them with a reconstruction performance that is at least 2.64 and 1.83 times higher, respectively.

We substantiate this claim in Section 5.3 in which we conduct a quantitative assessment of the reconstruction performance of the individual approaches and show our results in Table 3 in the paper.

A.4.2 Experiments

In order to assess the performance of the reconstruction approaches, we generate videos of individuals in video calls featuring diverse surroundings and virtual backgrounds. These videos are created using recordings in front of a green screen, providing ground-truth masks of the caller. These masks enable the assessment of pixel leakage from the real surroundings in each frame. In combination with the masks created by the video conferencing services to insert the virtual background, this allows for a perfect reconstruction of the surroundings, serving as a reference for evaluating the attacks.

We expect the evaluation to require approx. 10 human minutes and 48 hours on consumer hardware (recent CPU with 16 cores and 32GB RAM) and no GPU acceleration.

Preparation. Make sure the necessary dependencies are installed as described in Appendix A.2.4 and the artifact is acquired and setup as described in Appendix A.3.1. That is, it should be downloaded and unpacked from Zenodo or cloned from the Github repository and the script `scripts/build.sh` should have been executed successfully.

Execution. To run the experiment starting from the dataset generation over conducting our attack as well as the two baseline approaches, to finally evaluating the reconstruction performance, the script `scripts/run.sh` has to be executed.

Results. Upon successful termination of the script, the reconstruction performances are calculated as described in paragraph *Measuring reconstructions* in Section 5.3 of the paper and printed on the screen for each individual attack with `vader` referring to the performance of our approach, `hilgefort` to the one from Hilgefort et al. and `sabra` to performance of the attack by Sabra et al. The results are shown per video conferencing system with `zoom` indicating the performance for videos that use the virtual background feature from Zoom and `mp` for MediaPipe. The scores should be close to the following values:

```
Attack: vader
vader-mp-interview: mean 0.1546
vader-zoom-interview: mean 0.1643
Attack: sabra
sabra-mp-interview: mean 0.0595
sabra-zoom-interview: mean 0.0691
Attack: hilgefort
hilgefort-mp-interview: mean 0.0093
hilgefort-zoom-interview: mean 0.0038
```

The reconstruction performance is significantly higher for our attack compared to the two other considered approaches substantiating our claims in the paper (see Appendix A.4.1).

A.5 Version

Based on the LaTeX template for Artifact Evaluation V20231005. Submission, reviewing and badging methodology followed for the evaluation of this artifact can be found at https://secartifacts.github.io/usenixsec2025/.