Privacy-Phys: Facial Video-Based Physiological Modification for Privacy Protection

Zhaodong Sun and Xiaobai Li, Member, IEEE

Abstract—The invisible remote photoplethysmography (rPPG) signals in facial videos can reveal the cardiac rhythm and physiological status. Recent studies show that rPPG is a non-contact way for emotion recognition, disease detection, and biometric identification, which means there is a potential privacy problem about physiological information leakage from facial videos. Therefore, it is essential to process facial videos to prevent rPPG extraction in privacy-sensitive situations such as online video meetings. In this letter, we propose Privacy-Phys, a novel method based on a pre-trained 3D convolutional neural network, to modify rPPG in facial videos for privacy protection. Our experimental results show that our approach can modify rPPG signals in facial videos more effectively and efficiently than the previous baseline. Our method can be applied to process facial videos in online video meetings or video-sharing platforms to prevent rPPG from being captured maliciously.

Index Terms—Privacy protection, remote photoplethysmography (rPPG), facial videos.

I. INTRODUCTION

Facial videos are widely captured and shared by ubiquitous cameras in our daily life. Some works [1]–[3] have realized that visible facial appearance in facial videos has privacy issues. However, invisible physiological signals in facial videos also present a potential risk of private information leakage.

Several studies [4]–[13] have shown that it is feasible to extract remote photoplethysmography (rPPG) signals from facial videos to derive physiological parameters such as heart rates. These physiological parameters from rPPG can be used for emotion detection [14]–[16], health monitoring [17], [18], and biometric identification [19]. However, the studies above also raise privacy issues of physiological signals in facial videos. Attackers might measure rPPG from facial videos to obtain emotion, health status, and biometric information without others’ permission. Recently, rPPG was used to estimate heart rates of athletes in Olympic shooting games, which incurred criticism of widespread biometrics capture [20]. Therefore, it is essential to develop a method to conceal rPPG signals in facial videos.

Two papers [21], [22] explored to conceal rPPG in facial videos for privacy protection. Chen et al. [21] used Laplacian pyramid representation to remove the color changes induced by rPPG signals in facial videos. However, their method was designed to eliminate the rPPG, which might produce unrealistic videos as mentioned by [23]. Recently, PulseEdit [22] was proposed to modify rPPG in facial videos. They generated a target rPPG signal and solved an optimization problem with the target and original rPPG signals to get the perturbation. The perturbation was uniformly added to the facial region to embed the target rPPG. However, there are three limitations of PulseEdit.

1) PulseEdit was only validated on simple datasets which are not close to realistic application situations with sudden head motions and facial expressions. 2) The rPPG modification is less effective when deep learning-based rPPG methods try to extract rPPG from the modified videos. 3) The running speed is not high enough to support real-time video processing, especially during online video meetings.

In this letter, we propose Privacy-Phys, an rPPG modification method based on a pre-trained 3D convolutional neural network (3D CNN) [24], and solve the limitations above. The diagram of the rPPG modification is shown in Fig. 1. The original facial video is processed by the rPPG modification algorithm so that the modified facial video will be visibly identical to the original video while the rPPG signal is modified.

Our contributions are listed below. 1) We propose a novel method to modify rPPG in facial videos by leveraging a pre-trained 3D CNN model. 2) We demonstrate that our proposed method can effectively modify rPPG especially against deep learning-based rPPG measurement methods. 3) Our method runs significantly faster than PulseEdit and can be used for real-time video processing. The remainder of this letter is structured as follows. In Section II, we introduce how Privacy-Phys modifies
rPPG in facial videos. In Section III, we describe our experimental setup and results. In Section IV, we draw our conclusions and future work.

II. METHOD

A. Problem Formulation

Given a facial video, two requirements need to be met to modify the rPPG information. One is that the modified video should be visibly identical to the original video, and the other one is that rPPG measurement methods can only capture the target rPPG signal that we embedded but not the original rPPG signal.

Our idea is to use the pre-trained 3DCNN-based rPPG estimator \( G_{\theta}(x) \) in [8] to modify rPPG. The 3DCNN model is an end-to-end rPPG estimator, which can directly accept a face video as the input \( x \in \mathbb{R}^{T \times C \times H \times W} \) where \( T \) is the length of the time dimension, \( C \) is the length of the color channel dimension, \( H \) is the length of the height dimension, and \( W \) is the length of the width dimension. The output \( G_{\theta}(x) \in \mathbb{R}^T \) is a one-dimensional rPPG signal. Since we only need a general rPPG estimator \( G_{\theta}(x) \), other deep learning-based models can also be used as \( G_{\theta}(x) \). We choose 3DCNN for two main reasons. 1) 3DCNN is an end-to-end rPPG estimator which can directly accept a face video as the input, while other deep learning-based rPPG estimators (e.g., [9], [25], [26]) require preprocessing steps to get features that can be fed into the models. 2) The 3DCNN can achieve good performance of rPPG estimation without any complex modules/steps (e.g., [11], [26]) that slow down the speed. Therefore, 3DCNN meets the need of both efficiency and precision. The 3DCNN model \( G_{\theta}(x) \) has the 3D convolution operations and 3D pooling operations to process face videos to extract rPPG induced by facial color change. We can fix the weights \( \theta^* \) in the pre-trained 3DCNN model \( G_{\theta} \) and modify its output rPPG signal so that the rPPG in the input facial video is also consistently modified, which is inspired by the video magnification method of Chen and McDuff [27].

We design our optimization objective according to the two requirements above. The objective should include a video fidelity term \( L_f \) to measure the distance between the modified video \( x \in \mathbb{R}^{T \times C \times H \times W} \) and the original video \( z \in \mathbb{R}^{T \times C \times H \times W} \). It should also include an rPPG term \( L_{rppg} \) to measure the distance between the target rPPG \( y \in \mathbb{R}^T \) and the measured rPPG \( G_{\theta}(x) \in \mathbb{R}^T \) from the modified video \( x \). Our optimization problem is

\[
\min_{x \in \mathbb{R}^{T \times C \times H \times W}} L_{rppg}(G_{\theta}(x), y) + \lambda \cdot L_f(x, z) \tag{1}
\]

where \( \lambda \) is the coefficient to control the effect of fidelity term \( L_f \). The first term \( L_{rppg} \) guarantees that the modified video \( x \) contains the target rPPG \( y \). The second term \( L_f \) makes sure the modified video \( x \) is visibly identical to the original one \( z \).

The next step is to determine the exact form of the two terms \( L_{rppg} \) and \( L_f \) in the objective. Negative Pearson correlation is a good option for \( L_{rppg} \) to measure the similarity between the two one-dimensional signals. It has been validated to be effective in measuring the distance between two rPPG signals in [8]. Since the objective function is minimized, a smaller \( L_{rppg} \) value means a higher Pearson correlation. \( L_{rppg} \) is defined as \( L_{rppg}(\hat{y}, y) = -\text{cov}(\hat{y}, y)/\sigma_{\hat{y}}\sigma_y \) where \( \hat{y} = G_{\theta}(x) \), \( \sigma_y \) is the standard deviation of \( y \) and defined as \( \sigma_y = \sqrt{\frac{1}{T} \sum_{t=1}^T y(t)^2 - (\frac{1}{T} \sum_{t=1}^T y(t))^2} \).

\( \text{cov}(\hat{y}, y) \) is the covariance which is defined as \( \text{cov}(\hat{y}, y) = \frac{1}{T} \sum_{t=1}^T \hat{y}(t)y(t) - \frac{1}{T} \sum_{t=1}^T \hat{y}(t) \frac{1}{T} \sum_{t=1}^T b(t) \).

The fidelity term \( L_f \) can be defined as the mean squared Euclidean distance, which is \( L_f(x, z) = \frac{1}{C \cdot H \cdot W} \| x - z \|^2 \).

B. Background Pixel Constraint

We have the prior knowledge that only skin pixels have rPPG information [28]. When the rPPG is modified, only skin pixels in the facial video should be modified, and the background pixels should be kept intact. Suppose that we have a binary skin mask \( S \in \{0, 1\}^{T \times C \times H \times W} \) for the original video \( z \in \mathbb{R}^{T \times C \times H \times W} \) and 1 in the skin mask \( S \) means skin pixels, the constraint can be represented as \( S \odot x = S \odot z \), where \( S \) is the background mask obtained by swapping 0 and 1 in the skin mask \( S \) and \( \odot \) is the element-wise multiplication. Overall, the optimization problem with the background pixel constraint is shown below.

\[
\min_{x \in \mathbb{R}^{T \times C \times H \times W}} L_{rppg}(G_{\theta}(x), y) + \lambda \cdot L_f(x, z) \quad \text{s.t. } S \odot x = S \odot z \tag{2}
\]

C. Projected Gradient Descent for rPPG Modification

Since we have a background pixel constraint, we can use projected gradient descent to ensure that the background pixel constraint is always satisfied. The iteration for the projected gradient descent is shown below.

\[
x_{t+1} = x_t - \alpha \cdot \text{Opt}(\nabla_x L(x_t), x_t) \\
x_{t+1} = S \odot x_{t+1} + \hat{S} \odot z \tag{3}
\]

The first line is the gradient descent step where \( \text{Opt} \) is the optimizer such as Adam [29], \( \alpha \) is the learning rate, and \( \nabla_x L(x_t) \) is the gradient of the objective function with respect to \( x \). The second line is the projection step for the background pixel constraint, which ensures that only the skin pixels are changed while the background pixels are not changed. The overall diagram of the proposed method is shown in Fig. 2.
III. EXPERIMENTS

A. Experiment Setup

Data: PURE [30] and MMSE-HR [31] datasets are used for experiments. PURE has 60 facial videos recorded from ten subjects with $640 \times 480$ resolution and 30 frames per second (fps) in six different setups, including steady and simple motion tasks. MMSE-HR has 102 facial videos with $1040 \times 1392$ resolution and 25 fps from 40 subjects. The videos contain spontaneous facial expressions and sudden head motions, which are close to realistic situations. MMSE-HR is more challenging than PURE dataset due to sudden facial motions. The original videos are cropped by OpenFace [32] to get full face bounding boxes, and bounding boxes are resized to $128 \times 128$. In order to explore conditions when only partial faces are available (e.g., due to occlusion), we also spatially crop the full face bounding boxes into four equal quarters of upper left, upper right, bottom left, and bottom right.

Experimental Protocol: The pre-trained 3DCNN model $G_{PF}(x)$ was obtained from [8], which is the precondition to perform our rPPG modification method. We then perform rPPG modification on PURE and MMSE-HR datasets, respectively. These videos are divided into 15-sec video clips, and we use the proposed method to embed a 15-sec target PPG signal into these video clips. Similar to PulseEdit [22], we generate a sinusoid of frequency 120 beats per minute (bpm) as the target rPPG signal. For the parameters in the proposed method, we try different values of $\lambda$ and report the results in the following. For each video clip, we perform the projected gradient descent with Adam optimizer [29] for 20 iterations which are enough to converge. The learning rate of the Adam optimizer is 0.1, and other parameters in the optimizer are set to the default values as in [29]. Skin masks $S$ are obtained from the original videos by pixel thresholding used in [33].

rPPG Measurement Methods: The effect of rPPG modification needs to be evaluated against rPPG measurement algorithms. After getting the modified videos, we use several rPPG measurement algorithms to get the heart rates from the modified videos. We choose three traditional rPPG algorithms including POS [7], CHROM [5], ICA [4], and four latest deep learning-based rPPG algorithms including PhysNet [8] trained on OBF dataset [34], TS-CAN [9] trained on AFR dataset [35], Gideon2021 [10] trained on UBFC-rPPG dataset [36], and PhysFormer [11] trained on VPL dataset [26]. Deep learning-based rPPG methods are trained according to the steps in their papers.

Evaluation Metrics: We use two metrics, including peak signal to noise ratio (PSNR) [37] and mean absolute errors (MAE) of heart rates. PSNR is used to evaluate the video similarity between modified videos and original videos. A higher PSNR means better video fidelity. To evaluate if the target PPG signal is successfully embedded into the modified video clips, we calculate the mean absolute error (MAE) between the heart rates in the modified face videos and the heart rates in the target PPG signal. The MAE values should be small if the target PPG is successfully embedded into the new video clips.

B. Experimental Results

The influence of $\lambda$: $\lambda$ is a parameter to balance the fidelity term $L_f$ and rPPG term $L_{rPPG}$. Changing $\lambda$ will cause the change of video quality and rPPG modification success. Fig. 3 shows PSNR/MAE with respect to $\lambda$ on MMSE-HR dataset with full faces. A larger $\lambda$ will result in a higher PSNR and a larger MAE. From the security aspect, a smaller $\lambda$ will have higher rPPG modification success and lower PSNR, which can make stronger privacy protection under the premise of lower video quality. We want to find a $\lambda$ where PSNR is high while MAE is low. The results show that, when $\lambda$ is larger than 0.1, MAE will increase sharply, and PSNR will not increase too much. This means $\lambda = 0.1$ is a suitable level where video quality is good, and most rPPG signals are successfully modified. Therefore, we choose $\lambda = 0.1$ to balance video quality and modification success.

Comparison with PulseEdit: Table I shows the MAE and PSNR for our method with $\lambda = 0.1$ and the baseline method PulseEdit [22] with their default settings on PURE dataset and MMSE-HR dataset with full and partial faces.
Fig. 4. MAE with respect to PSNR. (a) comparison between our method and PulseEdit on traditional rPPG methods. (b) comparison between our method and PulseEdit on deep learning-based rPPG methods.

quarters also demonstrate that our method is still effective when only partial faces are available.

PulseEdit also has a parameter similar to $\lambda$ to trade off video quality and modification success. We can change this parameter to get the curves of MAE with respect to PSNR. Fig. 4 plots the MAE with respect to PSNR to compare our method and PulseEdit on MMSE-HR dataset with full faces. At the same PSNR, our method can achieve a lower MAE than PulseEdit, which means our method can more effectively modify rPPG than PulseEdit. Our method also surpasses PulseEdit much more on deep learning-based rPPG methods than on traditional rPPG methods.

Running Speed: Our method and PulseEdit are tested on a Linux workstation with an Intel Xeon 6230 CPU, 16 GB memory, and a single Nvidia V100 GPU. The speed of PulseEdit on GPU is 13 fps. Our method runs at 42 fps on GPU, which is much faster than PulseEdit since our method uses a deep neural network that GPU can accelerate. However, PulseEdit does not employ parallel computation models, so GPU does not give PulseEdit a significant boost. The results show that our proposed method can be potentially used for real-time video processing during online video meetings.

C. Visual Validation

Fig. 5 shows a case where our method can successfully modify rPPG while PulseEdit fails. This case is challenging since the person has apparent facial expressions. We use PhysNet to measure rPPG signals from the original and modified videos. The rPPG signal from our modified video looks cleaner than that from PulseEdit’s one. Power spectrum densities (PSDs) also indicate our method can successfully suppress the original rPPG with 88 bpm and add a new rPPG with 120 bpm. However, PulseEdit can only add the 120 bpm and fail to suppress the original rPPG signals with 88 bpm in this challenging case. The results indicate that it is still possible for attackers to obtain the original heart rates from the videos processed by PulseEdit when facial expressions occur.

Fig. 6 shows the RGB absolute difference between the original and modified frame for our method and PulseEdit. Our method has large perturbation in the cheek and forehead area, which follows the rPPG spatial distribution [38]. Our method also has the largest perturbation in the green channel, followed by blue and red channels, which follows the rPPG color channel distribution as illustrated in [7]. These observations demonstrate the correctness of our method. However, PulseEdit applies the perturbation uniformly to the face in spatial and color dimensions. Zhan et al. [39] have demonstrated that deep learning-based rPPG methods highly rely on the rPPG spatial and color distribution to extract rPPG signals, which means our perturbation is more compatible with deep learning-based rPPG methods.

IV. CONCLUSION

We propose Privacy-Phys to modify rPPG signals in facial videos by leveraging a pre-trained 3DCNN model. Experimental results demonstrate that the proposed method can modify rPPG more effectively and faster than the baseline method (PulseEdit). Privacy-Phys can be applied to process facial videos in privacy-sensitive situations such as online video meetings to prevent rPPG from being captured maliciously. Future works can focus on concealing other physiological signals such as respiration-induced motions in facial videos for privacy protection.

ACKNOWLEDGMENT

The authors would like to thanks CSC-IT Center for Science, Finland, for providing computational resources.
REFERENCES


