Augmentation of rPPG Benchmark Datasets: Learning to Remove and Embed rPPG Signals via Double Cycle Consistent Learning from Unpaired Facial Videos

Cheng-Ju Hsieh, Wei-Hao Chung, and Chiou-Ting Hsu

National Tsing Hua University, Hsinchu, Taiwan
peter55180831@gmail.com, godofyax@gmail.com, and cthsu@cs.nthu.edu.tw

Abstract. Remote estimation of human physiological condition has attracted urgent attention during the pandemic of COVID-19. In this paper, we focus on the estimation of remote photoplethysmography (rPPG) from facial videos and address the deficiency issues of large-scale benchmarking datasets. We propose an end-to-end RErPPG-Net, including a Removal-Net and an Embedding-Net, to augment existing rPPG benchmark datasets. In the proposed augmentation scenario, the Removal-Net will first erase any inherent rPPG signals in the input video and then the Embedding-Net will embed another PPG signal into the video to generate an augmented video carrying the specified PPG signal. To train the model from unpaired videos, we propose a novel double-cycle consistent constraint to enforce the RErPPG-Net to learn to robustly and accurately remove and embed the delicate rPPG signals. The new benchmark "Aug-rPPG dataset" is augmented from UBFC-rPPG and PURE datasets and includes 5776 videos from 42 subjects with 76 different rPPG signals. Our experimental results show that existing rPPG estimators indeed benefit from the augmented dataset and achieve significant improvement when fine-tuned on the new benchmark. The code and dataset are available at https://github.com/nthumplab/RErPPGNet.

Keywords: Remote Photoplethysmography, Data Augmentation, Double-Cycle Consistency, Remote Heart Rate Measurement

1 Introduction

Contactless and video-based methods for heart rate (HR) estimation have attracted enormous research interests. Especially, remote photoplethysmography (rPPG), which analyzes the subtle chrominance changes reflected on skin, captures the heart rate related information [3, 4]. Recent learning-based methods for rPPG estimation fall into two categories. The first category [8, 10, 17, 22] involved a number of preprocessing steps, such as facial landmark detection or regions-of-interest detection, to obtain a spatial-temporal representation as the

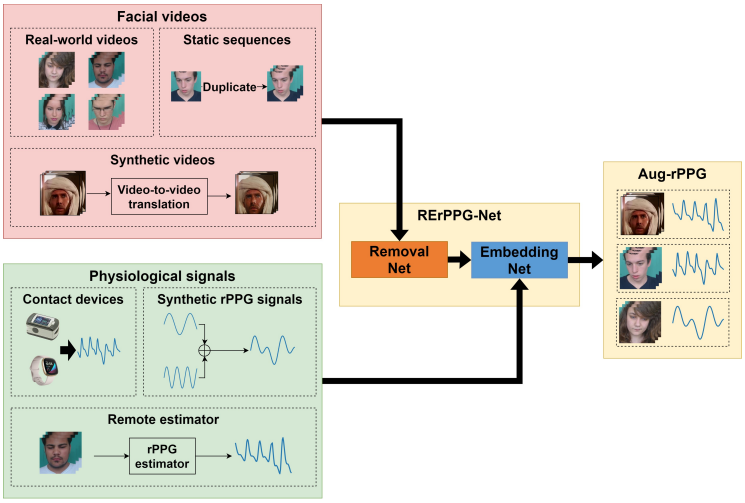

Fig. 1. The proposed scenario of rPPG data augmentation.

input to the CNN-based model. The second category [2, 3, 5, 13, 14, 16, 23] focused on training an end-to-end architecture to directly estimate either rPPG signals or HR from an input facial video.

There remain a number of challenges in developing robust rPPG or HR estimation. First, since the success of deep learning-based methods heavily relies on large-scale supervised datasets, there are unfortunately only few datasets publicly available for rPPG or HR estimation. In addition, the ground truth labels of these datasets are not always accurate and thus usually lead to unstable estimation. Finally, because of the lack of large-scale dataset, previous methods tend to overfit to a certain dataset but poorly generalize to others.

In this paper, to tackle the aforementioned issues, we propose the RErPPG-Net to augment existing rPPG datasets. As shown in Fig. 1, the RErPPG-Net consists of a Removal-Net and an Embedding-Net. We first use the Removal-Net to erase any possible rPPG-relevant signals in the input video and then use the Embedding-Net to embed a PPG signal into the resultant video.

However, training the Removal-Net and Embedding-Net is highly challenging, because no paired videos (i.e., facial videos from the same subject with and without PPG signals) are available for model training. Inspired by the success of cycle consistency learning [25] from unpaired data, we propose a novel double-cycle consistent constraint into our model training. We use Fig. 2 to illustrate the idea of single cycle consistent and the proposed double-cycle consistent learning. In Fig. 2, when given an input X and two translators T_1 and T_2, the original single cycle consistency [25] enforces X' to be consistent with X. In our case, because the rPPG signals are extremely delicate in comparison with the facial content, we found this single cycle consistency between X' and X tends to focus

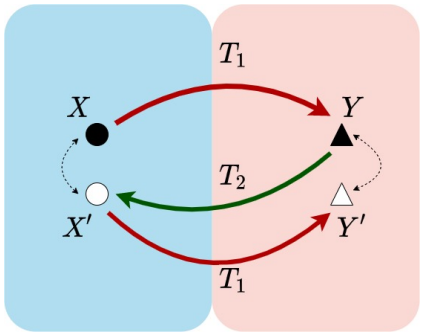

Fig. 2. Illustration of the double-cycle consistency.

on facial contents instead of the rPPG signals. Therefore, we adopt an additional cycle consistent constraint on Y and Y' to ensure that rPPG-related information in X' and X are well preserved. In addition, we also refer to the background signals of the input videos to guide the model training.

Our contributions are summarized below:

1) We propose the RErPPG-Net to generate an augmented rPPG estimation dataset: Aug-rPPG dataset, for public use on research of rPPG estimation.
2) We devise a novel double-cycle consistent constraint into the learning of RErPPG-Net and successfully generate high-quality videos with specified PPG signals.
3) Experimental results on UBFC-rPPG and PURE datasets show that existing rPPG estimators substantially improve the estimation accuracy and achieve state-of-the-art performance when fine-tuned on Aug-rPPG dataset.

2 Related Work

2.1 Remote Photoplethysmography Estimation

The goal of rPPG estimation is to remotely measure the heart rate from facial videos. Traditional approaches [4, 7, 12, 18, 20, 21] focused on separate physiological signals from facial videos under different prior assumptions. These methods generally perform well on videos recorded under controlled environment but may not generalize well to other scenarios. Many learning-based methods [2, 3, 5, 8, 10, 13, 14, 16, 17, 23] have also been developed for rPPG estimation. In [8], the authors proposed a Dual-GAN framework to learn a noise-resistant mapping from input spatial-temporal maps to ground truth blood volume pulse signals. In [3], the DeepPhys framework was proposed to simultaneously generate an attention mask for RoI detection and to recover rPPG signals using the convolutional attention network. In [23], the authors proposed a STVEN network to enhance hidden rPPG information from highly compressed videos and an rPPGNet to predict rPPG signals.

2.2 Data Augmentation

Data augmentation has been widely adopted to alleviate the shortage of well-labeled training data. Traditional augmentation methods include image flipping, rotating, cropping, scaling, shifting, and so on. With the success of Generative Adversarial Networks (GANs) [6, 19, 25] and autoencoder [16] in generating high fidelity data, many methods are proposed to use generators to automate the data augmentation. In [6], the authors used conditional GANs to achieve both age progression and regression. In [9], the authors utilized 3D avatars to synthesize facial videos with blood flow and breathing patterns. In [25], the authors proposed an unsupervised method with cycle-consistency to solve image-to-image translation from unpaired data. In [16], the authors proposed a multi-task framework to predict rPPG signals and to augment data simultaneously.

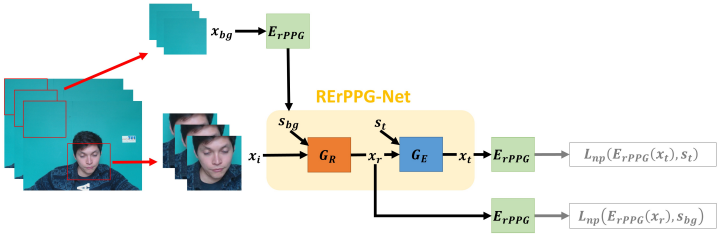

Fig. 3. The proposed RErPPG-Net, which consists of a Removal-Net G_R and an Embedding-Net G_E. The rPPG-removed video x_r is expected to carry no rPPG signal; whereas the rPPG-embedded video x_t is expected to carry the specified signal s_t.

3 Proposed Method

3.1 Overview

In this paper, we propose a RErPPG-Net to augment existing rPPG datasets by embedding ground-truth PPG signals into any existing facial videos. As shown in Fig. 3, the proposed RErPPG-Net consists of a Removal-Net G_R and an Embedding-Net G_E and aims to remove any inherent rPPG signals existing in the input videos and then to embed the specified PPG signals into the rPPG-removed videos. To train the model from unpaired videos, we propose a novel double-cycle consistent learning to enforce the Embedding-Net G_E and the Removal-Net G_R to learn to robustly and accurately embed and remove the delicate rPPG signals.

3.2 RErPPG-Net

Fig. 3 illustrates the proposed RErPPG-Net. Let $x_i \in \mathbb{R}^{H\times W\times C\times T}$ be an input facial video; $s_t \in \mathbb{R}^T$ denote the specified PPG signal, and $x_t \in \mathbb{R}^{H\times W\times C\times T}$

denote the generated facial video, where H, W, C, and T denote the height, width, the number of channels, and the length of the video, respectively. E_{rPPG} denotes an off-the-shelf rPPG estimator.

Note that, the input facial video may come from different sources, e.g., broadcast videos, video-to-video translated results, spoof videos, or temporally duplication of static images. Therefore, we first need to ensure that its inherent rPPG signals (if any) are completely erased before we embed another PPG signal. However, because we do not know whether the input video x_i carries rPPG signals or not, training the Removal-Net G_R is not a trivial task. Hence, we assume that the background region of any input video should carry no rPPG information and propose to use the signal estimated from the background region as the pseudo ground truth of "no rPPG signal". As shown in Fig. 3, we crop the upper left corner of each video frame as the reference background and let $s_{bg} \in \mathbb{R}^T$ denote the signal predicted by the rPPG estimator E_{rPPG} from the background region. Given the input x_i, we refer to s_{bg} to remove the rPPG signals by the Removal-Net G_R:

$$x_r = G_R(x_i, s_{bg}), \tag{1}$$

where $x_r \in \mathbb{R}^{H \times W \times C \times T}$.

Next, we embed the specified PPG signal s_t into the rPPG-removed video x_r by the Embedding-Net G_E:

$$x_t = G_E(x_r, s_t). \tag{2}$$

To ensure that the rPPG-removed video x_r carries no rPPG signal and that the rPPG-embedded video x_t carries the signal s_t, we formulate the rPPG loss L^{re}_{rPPG} as:

$$L^{re}_{rPPG} = L_{np}(s_{bg}, E_{rPPG}(x_r)) + L_{np}(s_t, E_{rPPG}(x_t)), \tag{3}$$

in terms of the negative Pearson correlation loss L_{np} defined by

$$L_{np}(s, s') = 1 - \frac{(s - \bar{s})^t (s' - \bar{s'})}{\sqrt{(s - \bar{s})^t (s - \bar{s})}\sqrt{(s' - \bar{s'})^t (s' - \bar{s'})}}, \tag{4}$$

where $s \in \mathbb{R}^T$ and $s' \in \mathbb{R}^T$.

3.3 Double-Cycle Consistent Learning for Embedding-Net

Nevertheless, training RErPPG-Net in terms of only the rPPG loss L^{re}_{rPPG} is far from enough. Specifically, there is no guarantee that the output video x_t is perceptually satisfactory and that x_t carries only the specified signal s_t. Therefore, we devise a double-cycle consistent learning to constrain the Embedding-Net to learn to generate perceptually plausible results embedded with only the specified PPG signals.

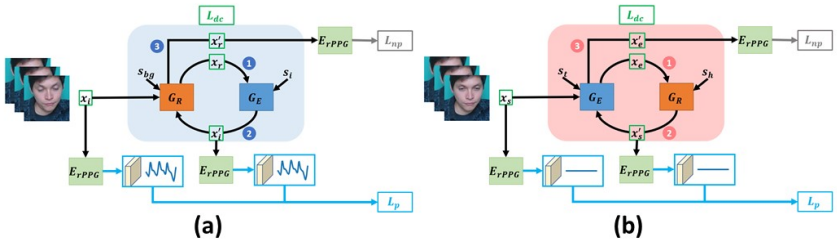

Fig. 4. (a) Double-cycle consistent learning for Embedding-Net G_E; and (b) double-cycle consistent learning for Removal-Net G_R.

As shown in Fig. 4 (a), we illustrate the double-cycle consistent learning with three stages. Here we assume the input video x_i and its ground truth PPG signal s_i are available during this training stage. First, we obtain the rPPG-removed video x_r by Equ. (1). Second, we embed the original PPG signal s_i of x_i back into x_r and obtain x'_i by

$$x'_i = G_E(x_r, s_i), \tag{5}$$

where $x'_i \in \mathbb{R}^{H\times W\times C\times T}$.

Third, we input x'_i into the Removal-Net G_R to obtain its rPPG-removed video x'_r by

$$x'_r = G_R(x'_i, s_{bg}), \tag{6}$$

where $x'_r \in \mathbb{R}^{H\times W\times C\times T}$.

From Equs. (5) and (6), we expect that the rPPG-carrying videos x'_i should be consistent with x_i and that the rPPG-removed videos x'_r should be consistent with x_r. Thus, we formulate the double-cycle consistent loss L_{dc}^{embed} by

$$L_{dc}^{embed} = ||x_i - x'_i||_1 + ||x_r - x'_r||_1. \tag{7}$$

In addition, to ensure the predicted signals from x'_i and x'_r are highly correlated with s_i and s_{bg}, respectively, we define an rPPG-embedding loss term by:

$$\begin{aligned} L_{rPPG}^{embed} &= L_{np}(s_i, E_{rPPG}(x'_i)) \\ &\quad + L_{np}(s_{bg}, E_{rPPG}(x'_r)). \end{aligned} \tag{8}$$

To constrain the perceptual consistency between x'_i and x_i in the feature space, we further include a multi-layer perception loss [24] by

$$L_p^{embed} = \sum_k ||E_{rPPG}^k(x_i) - E_{rPPG}^k(x'_i)||_1, \tag{9}$$

where $E_{rPPG}^k(\cdot)$ is the feature of the k-th layer of the rPPG estimator E_{rPPG}.

3.4 Double-Cycle Consistent Learning for Removal-Net

To train the Removal-Net G_R, the major difficulty lies in the lack of a paired video without rPPG signals. Therefore, we randomly select one frame from the input video x_i and temporally duplicate this frame to create a static video $x_s \in \mathbb{R}^{H\times W\times C\times T}$ as the reference ground-truth of rPPG-removed video. In addition, because there exists no chrominance change on the facial skin of the static video x_s, we assume x_s carries only a flat DC signal s_h.

By referring to the static video x_s and its ground truth signal s_h, we then devise a double-cycle consistent learning to train the Removal-Net G_R. As shown in Fig. 4 (b), we illustrate the training with three stages. First, we embed the PPG signal s_t into the static video x_s by

$$x_e = G_E(x_s, s_t), \tag{10}$$

where $x_e \in \mathbb{R}^{H\times W\times C\times T}$.

In the second and third stages, similarly to the case in Section 3.3, we remove the rPPG signal from the rPPG-embedded video x_e to obtain x'_s and then again embed s_t back to x'_s to obtain an embedded video x'_e by:

$$x'_s = G_R(x_e, s_h), \tag{11}$$

and

$$x'_e = G_E(x'_s, s_t), \tag{12}$$

respectively.

Similar to Equ. (7), we again impose the double-cycle consistent constraints between x'_s and x_s and between x'_e and x_e and define the loss by

$$L_{dc}^{remove} = ||x_s - x'_s||_1 + ||x_e - x'_e||_1. \tag{13}$$

To ensure that x_e and x'_e indeed carry the signal s_t and that x'_s carries a flat signal, we define the rPPG-removal loss by

$$\begin{aligned} L_{rPPG}^{remove} &= L_{np}(s_t, E_{rPPG}(x_e)) \\ &+ L_{np}(s_t, E_{rPPG}(x'_e)) \\ &+ L_{var}(E_{rPPG}(x'_s)), \end{aligned} \tag{14}$$

where we use the signal variance L_{var} to measure whether x'_s carries a DC signal or not, because the negative Pearson correlation coefficient is inapplicable in this case. L_{var} is defined by

$$L_{var}(s) = (s - \bar{s})^t (s - \bar{s}). \tag{15}$$

Here, we again adopt the multi-layer perception loss to ensure the consistency between x'_s and x_s in the feature space by

$$L_p^{remove} = \sum_k ||E_{rPPG}^k(x_s) - E_{rPPG}^k(x'_s)||_1. \tag{16}$$

3.5 Loss function

Finally, we include the rPPG loss, the double-cycle consistent loss, and the perceptual loss to define the total loss for training G_E and G_R by:

$$L_{total} = \lambda_1 L_{rPPG} + L_{dc} + L_p, \tag{17}$$

where

$$L_{rPPG} = L_{rPPG}^{re} + L_{rPPG}^{embed} + L_{rPPG}^{remove}, \tag{18}$$

$$L_{dc} = L_{dc}^{embed} + L_{dc}^{remove}, \text{and} \tag{19}$$

$$L_p = L_p^{embed} + L_p^{remove}, \tag{20}$$

and λ_1 is a hyper-parameter and is empirically set as 0.01 in all our experiments.

4 Experiments

4.1 Datasets

The UBFC-rPPG dataset [1] contains 42 RGB videos, each is recorded from a single individual. All the videos are recorded by Logitech C920 HD Pro with resolution of 640 × 480 pixels in uncompressed 8-bit format and 30 fps. CMS50E transmissive pulse oximeter is used to monitor the PPG signals and corresponding heart rates.

Because there is no pre-defined data split for training and testing on UBFC-rPPG dataset, previous methods did not all follow the same setting for evaluation. In [8, 10, 13], the training and the testing sets contain the first 30 subjects and the rest 12 subjects, respectively. In [16], the training and testing sets contain 28 and 14 subjects, respectively. In [2, 5], no description about the data split is given. In our experiment, to have a balanced rPPG distribution within the training and testing sets, we include 35 subjects and the rest 7 subjects in the training and testing sets. In addition, to have a fair comparison with [8, 10, 13], we also conduct experiments using their setting with 30 and 12 subjects in training/testing sets. More detailed description and results are given in Sec. 4.5.

The PURE dataset [15] contains 10 subjects performing six different and controlled head motions in front of the camera. The six setups include: (1) sitting still, (2) talking, (3) slowly moving the head, (4) quickly moving the head, (5) rotating the head with 20° angles, and (6) rotating the head with 35° angles. All the videos are recorded by evo274CVGE camera with resolution of 640 × 480 pixels and 30 fps. Pulox CMS50E finger clip pulse oximeter is adopted to capture PPG signals with sampling rate of 60 Hz. The PPG signals are reduced to 30 fps with linear interpolation to align with the videos. We follow [16] to split the dataset into the training and testing sets with videos from 7 and 3 subjects, respectively.

The VIPL-HR dataset [11] contains 2378 RGB videos of 107 subjects captured with 9 scenarios and recorded by 3 different devices. Because the sampling

rates between videos and PPG signals are different, similar to [8], we additionally resample the PPG signals to the corresponding video frame rates by cubic spline interpolation.

4.2 Implementation Details

We develop the proposed Removal-Net G_R and Embedding-Net G_E using the generator proposed in [6]. As to the rPPG estimator E_{rPPG}, we adopt the rPPG model in [16] and then follow [23] to aggregate the features in the middle layer to predict the rPPG signals. We train the RErPPG-Net (i.e., G_R and G_E) and the rPPG estimator E_{rPPG} with Nvidia RTX 2080 and RTX 3080 for 900 and 500 epochs, respectively, and use Adam optimizer with the learning rate of 0.001. The RErPPG-Net is trained with batch size 1 and E_{rPPG} is trained with batch size 3. In each epoch, we randomly sample 60 consecutive frames from each training video to train RErPPG-Net and E_{rPPG}.

4.3 Evaluation Metrics

To assess how the proposed data augmentation improves the rPPG estimation, we follow [13] to derive heart rate (HR) from the predicted rPPG signals and then evaluate the results in terms of the following metrics: (1) Mean absolute error (MAE), (2) Root mean square error (RMSE), (3) Pearson correlation coefficient (R), (4) Peak signal-to-noise ratio (PSNR), and (5) Structural similarity (SSIM).

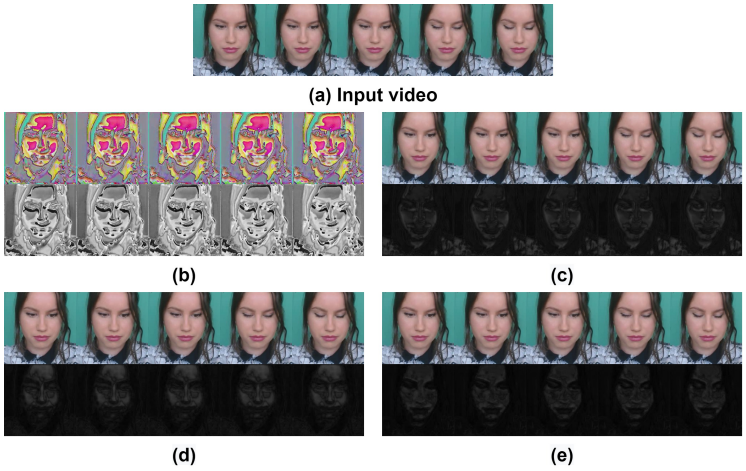

Fig. 5. Visualized examples of ablation study on UBFC-rPPG dataset, when training G_R and G_E using: (b) L^{re}_{rPPG}; (c) L^{re}_{rPPG} with double-cycle consistent loss and rPPG loss on Embedding-Net; (d) L^{re}_{rPPG} with double-cycle consistent loss and rPPG loss on Embedding-Net and Removal-Net; and (e) the proposed total loss.

Table 1. Ablation study on UBFC-rPPG dataset.

L_{rPPG}^{re}	L_{dc}^{embed}	L_{dc}^{remove}	L_{rPPG}^{embed}	L_{rPPG}^{remove}	L_p	MAE↓	RMSE↓	PSNR↑	SSIM↑
✓						44.79	48.53	5.11	0.1642
✓	✓		✓			4.14	9.51	49.72	0.9995
✓	✓	✓	✓	✓		2.30	5.63	51.08	0.9997
✓	✓	✓	✓	✓	✓	**0.71**	**1.48**	**52.71**	**0.9998**

4.4 Ablation Study

We conduct several ablation studies on UBFC-rPPG dataset and show the results in Table 1 and Fig. 5. First, to evaluate how the rPPG estimator E_{rPPG} may benefit from the proposed RErPPG-Net, we train E_{rPPG} using the augmented data from UBFC-rPPG training set and then test on the UBFC-rPPG testing set. Next, to evaluate the perceptual quality of the augmented videos, we embed the original PPG signals into the rPPG-removed videos and then measure the PSNR and SSIM between the augmented videos and the input videos.

In Table 1, the first column L_{rPPG}^{re} indicates that the RErPPG-Net is trained using only the rPPG loss L_{rPPG}^{re} but without the double-cycle consistent learning or the perceptual loss. We first evaluate the effectiveness of the double-cycle consistent loss. When including the loss terms L_{dc}^{embed} and L_{rPPG}^{embed}, we significantly reduce MAE by about 91% and RMSE by about 80%. As shown in Fig. 5 (b) and (c), the visual quality of Fig. 5 (c) is greatly improved with PSNR increased from 5.11 to 49.72 and SSIM from 0.1642 to 0.9995. This improvement comes from that the double-cycle consistent loss on the Embedding-Net effectively constrains the generated video to be consistent with the input on the pixel level. Next, when including the loss terms L_{dc}^{remove} and L_{rPPG}^{remove}, we further reduce MAE by about 44% and RMSE by about 41% and also increase PSNR and SSIM with 1.36 and 0.0002, respectively. These results again verify that double-cycle consistent learning effectively constrains the RErPPG-Net to learn to remove and embed rPPG signals as well as to generate photo-realistic facial videos. Finally, when including the perception loss L_p in the training stage, we enforce the augmented videos to be consistent with the input videos in the feature space and achieve the best performance among all settings.

In Fig. 5, we show the visual comparisons of videos generated by RErPPG-Net trained using different losses. In each setting, the first row shows the augmented video and the second row shows the residual between the augmented video and the input video. The residual results are intensity enhanced by gamma transformed with $\gamma = 3$ to highlight the differential areas. As shown in Fig. 5 (c), (d), and (e), the major differences in the residual videos locate around the face area; these results show that RErPPG-Net indeed learns to erase and embed the rPPG information on the facial regions.

In Table 2, we compare the performance of using single-cycle and double-cycle consistent learning. To train the single-cycle framework, we remove all the loss terms related to x_r' and x_e' in Equs. (7) (8) (13) and (14). The results show

Table 2. Comparison of single-cycle and double-cycle framework on UBFC-rPPG dataset.

Method	MAE↓	RMSE↓	R↑
Ours (single-cycle)	2.38	4.73	0.84
Ours (double-cycle)	0.71	1.48	0.96

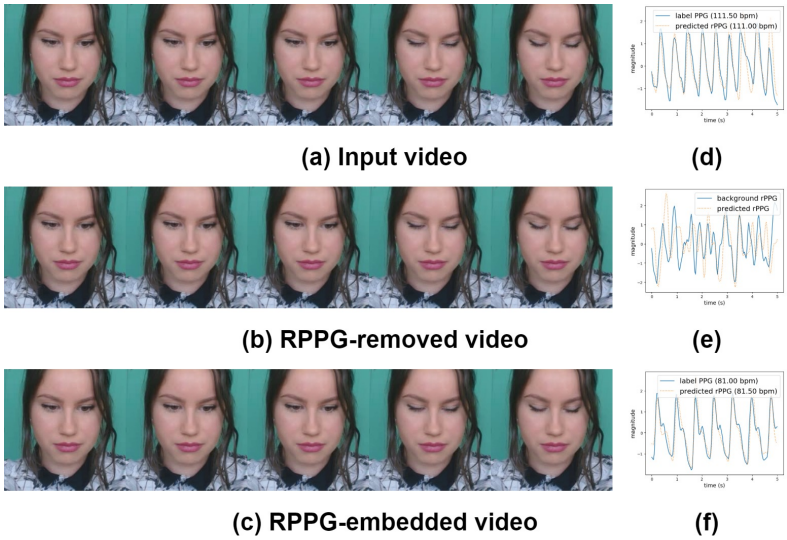

Fig. 6. (a) An input video x_i from UBFC-rPPG dataset; (b) The rPPG-removed video x_r; (c) The rPPG-embedded video x_t; (d) The ground truth PPG signal s_i (blue) and the predicted rPPG signal of x_i (orange); (e) The background signal s_{bg} (blue) and the predicted signal of x_r (orange); and (f) The specified PPG signal s_t (blue) and the predicted rPPG signal of x_t (orange).

that double-cycle consistent learning significantly reduces MAE by about 70% and RMSE by about 69% over the single-cycle consistent constraint and verify its effectiveness for generating photo-realistic augmented videos.

Finally, in Fig. 6, we give an example to visualize the rPPG-removed and rPPG-embedded videos. As shown in Fig. 6 (b) and (c), both videos are visually indistinguishable from the input one. Moreover, in Fig. 6 (e) and (f), the estimated rPPG signals from the rPPG-removed and rPPG-embedded videos are highly correlated with the background signal and the ground truth signal, respectively. These results verify that the proposed RErPPG-Net successfully erases the rPPG signal from the input video and embeds the specified PPG signal into the rPPG-removed video.

Table 3. Comparison on UBFC-rPPG dataset.

Method	MAE↓	RMSE↓	R↑
Meta-rPPG [5]	5.97	7.42	0.53
SynRhythm [10]	5.59	6.82	0.72
3D CNN [2]	5.45	8.64	-
PulseGAN [13]	1.19	2.10	0.98
Multi-task [16]	0.47	2.09	-
Dual-GAN [8]	0.44	0.67	0.99
rPPGNet [23]	0.72	1.47	0.96
rPPGNet (All) [23]	0.56	0.73	0.991
Ours* (Original)	0.64	0.95	0.94
Ours* (Aug)	1.84	3.81	0.85
Ours* (All)	0.75	1.05	0.94
Ours (Original)	0.66	1.40	0.96
Ours (Aug)	0.71	1.48	0.96
Ours (All)	**0.41**	**0.56**	**0.994**

4.5 Results and Comparison

In Table 3, we compare the HR predictions using our rPPG estimator E_{rPPG} with other methods [2, 5, 8, 10, 13, 16, 23] on the UBFC-rPPG dataset. The settings "Original", "Aug", and "All" indicate that we train E_{rPPG} using (1) only the original training videos in UBFC-rPPG dataset, (2) only the augmented videos, and (3) both the original training data and the augmented videos. The methods: 3D CNN [2], Meta-rPPG [5], Multi-task [16], and rPPGNet [23] are developed with the end-to-end architecture; whereas the other three methods [10, 13, 8] need to compute the spatial-temporal maps before using convolutional neural networks. The result of "Ours (Aug)" shows that E_{rPPG}, even only trained with augmented data, performs pretty well. The setting "Ours (All)" achieves the best performance with MAE 0.41, RMSE 0.56, and Pearson correlation coefficient (R) 0.994.

To have a fair comparison with [8, 10, 13], we follow the setting in [13] to train our RErPPG-Net and rPPG estimator E_{rPPG} and mark the results with "*". Although we observe degraded performance in "Ours* (Aug)" and "Ours* (All)", we believe the reason comes from the imbalanced HR distribution between the training and testing sets adopted in [13]. As shown in Fig. 7 (a), the HR distributions between the training and testing sets are very different; therefore, this imbalanced issue may grow even worse in the augmented data. On the other hand, in Fig. 7 (b), the data distribution in our setting is more balanced and thus the augmented data serve a better training set.

Furthermore, to show that our augmented dataset can also boost the performance of other rPPG estimators, we re-implement the rPPGNet [23] and show the results in Table 3. In comparison with "rPPGNet", the setting "rPPGNet (All)" has MAE reduced by about 22% and RMSE reduced by about 50% when training with both the original training data and the augmented videos. These

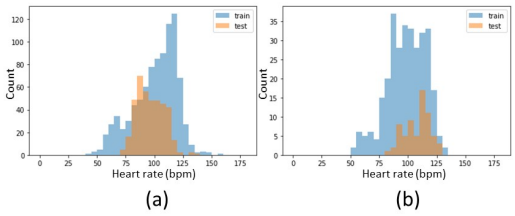

Fig. 7. Heart rate distribution of the training/testing data split in UBFC-rPPG dataset using (a) the setting in [13] and (b) our setting.

results verify that existing rPPG estimators can indeed benefit from the proposed augmented dataset.

Comparisons with the methods [4, 7, 8, 14, 16, 21] on PURE dataset are shown in Table 4, where 2SR [21], CHROME [4], and LiCVPR [7] are not learning-based methods. The setting "Ours (All)" again outperforms the other rPPG estimation methods with MAE 0.38 and RMSE 0.54.

Table 4. Comparison on PURE dataset.

Method	MAE↓	RMSE↓	R↑
LiCVPR [7]	28.22	30.96	-0.38
2SR [21]	2.44	3.06	0.98
CHROME [4]	2.07	2.50	**0.99**
HR-CNN [14]	1.84	2.37	0.98
Dual-GAN [8]	0.82	1.31	**0.99**
Multi-task [16]	0.40	1.07	0.92
Ours (Original)	0.69	1.24	0.96
Ours (All)	**0.38**	**0.54**	0.96

Finally, we conduct a cross-dataset validation to evaluate the generalization capability of the rPPG estimator when fine-tuning E_{rPPG} on our augmented videos. In Table 5, the settings "UBFC", "UBFC + Aug-U", "PURE", "PURE + Aug-P", "UBFC + PURE", and "UBFC + PURE + Aug-rPPG" indicate that the rPPG estimator E_{rPPG} is obtained by (1) training on UBFC-rPPG dataset, (2) training on the UBFC-rPPG dataset and then fine-tuning with the augmented videos of UBFC-rPPG dataset, (3) training on PURE dataset, (4) training on PURE dataset and then fine-tuning with the augmented videos of PURE dataset, (5) training on UBFC-rPPG and PURE datasets, and (6) training on both UBFC-rPPG and PURE dataset and then fine-tuning with our Aug-rPPG dataset. When involving the augmented data in the training stage, the results show that we reduce the MAE by about 69%, 41%, and 12% and RMSE

Table 5. Comparison of cross-dataset testing.

Training	Testing	MAE↓	RMSE↓	R↑
UBFC	PURE	14.18	20.20	0.34
UBFC+Aug-U	PURE	4.36	6.69	0.60
PURE	UBFC	3.78	6.69	0.71
PURE+Aug-P	UBFC	2.23	4.66	0.78
UBFC+PURE	VIPL-HR	28.94	33.73	0.18
UBFC+PURE+Aug-rPPG	VIPL-HR	25.40	31.14	0.15

by about 67%, 30%, and 8% in cross-dataset testing on PURE, UBFC-rPPG, and VIPL-HR datasets, respectively, Note that, because VIPL-HR is a much larger dataset and includes various head movements and illumination conditions, the cross-dataset testing on VIPL-HR usually results in poorer performance when training on small-scale datasets (e.g. UBFC-rPPG and PURE). Nevertheless, when fine-tuning E_{rPPG} on the proposed Aug-rPPG, we show that the cross-dataset testing on VIPL-HR indeed benefits from the proposed augmented data and reaches a better generalization capability.

4.6 Aug-rPPG Dataset

To generate the Aug-rPPG dataset, we use all the 76 training videos and the corresponding PPG signals from UBFC-rPPG and PURE datasets as the inputs to the proposed RErPPG-Net. The 76 input videos are from 42 subjects, where 35 subjects are from UBFC-rPPG training set and 7 subjects are from PURE training set. By running every possible combination of the videos and PPG signals, we generate $76^2 = 5776$ videos of resolution 200×200 pixels. Note that, because we only include the facial region of 200×200 pixels in the data augmentation, our generated videos are of the same quality as the two benchmark datasets.

5 Conclusion

In this paper, we propose the RErPPG-Net to augment existing rPPG benchmark datasets. The proposed RErPPG-Net includes (1) a Removal-Net to erase any inherent rPPG signals in facial videos and (2) an Embedding-Net to embed the specified PPG signals into the videos. To train the model from unpaired videos, we propose a novel double-cycle consistent constraint to enforce the Embedding-Net and the Removal-Net to learn to robustly and accurately embed and remove the delicate rPPG signals. The Aug-rPPG dataset is augmented from UBFC-rPPG and PURE datasets and includes 5776 videos with the same resolution as the original datasets. Experimental results on UBFC-rPPG, PURE, and VIPL-HR datasets verify the effectiveness of the proposed RErPPG-Net and also show that the augmented data indeed improve the estimation accuracy and the generalization capability of existing rPPG estimators.

References

1. Bobbia, S., Macwan, R., Benezeth, Y., Mansouri, A., Dubois, J.: Unsupervised skin tissue segmentation for remote photoplethysmography. Pattern Recognition Letters **124**, 82–90 (2019)
2. Bousefsaf, F., Pruski, A., Maaoui, C.: 3d convolutional neural networks for remote pulse rate measurement and mapping from facial video. Applied Sciences **9**(20), 4364 (2019)
3. Chen, W., McDuff, D.: Deepphys: Video-based physiological measurement using convolutional attention networks. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 349–365 (2018)
4. De Haan, G., Jeanne, V.: Robust pulse rate from chrominance-based rppg. IEEE Transactions on Biomedical Engineering **60**(10), 2878–2886 (2013)
5. Lee, E., Chen, E., Lee, C.Y.: Meta-rppg: Remote heart rate estimation using a transductive meta-learner. In: European Conference on Computer Vision. pp. 392–409. Springer (2020)
6. Li, Q., Liu, Y., Sun, Z.: Age progression and regression with spatial attention modules. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 11378–11385 (2020)
7. Li, X., Chen, J., Zhao, G., Pietikainen, M.: Remote heart rate measurement from face videos under realistic situations. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4264–4271 (2014)
8. Lu, H., Han, H., Zhou, S.K.: Dual-gan: Joint bvp and noise modeling for remote physiological measurement. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12404–12413 (2021)
9. McDuff, D., Liu, X., Hernandez, J., Wood, E., Baltrusaitis, T.: Synthetic data for multi-parameter camera-based physiological sensing. In: 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). pp. 3742–3748. IEEE (2021)
10. Niu, X., Han, H., Shan, S., Chen, X.: Synrhythm: Learning a deep heart rate estimator from general to specific. In: 2018 24th International Conference on Pattern Recognition (ICPR). pp. 3580–3585. IEEE (2018)
11. Niu, X., Han, H., Shan, S., Chen, X.: Vipl-hr: A multi-modal database for pulse estimation from less-constrained face video. In: Asian Conference on Computer Vision. pp. 562–576. Springer (2018)
12. Poh, M.Z., McDuff, D.J., Picard, R.W.: Advancements in noncontact, multiparameter physiological measurements using a webcam. IEEE transactions on biomedical engineering **58**(1), 7–11 (2010)
13. Song, R., Chen, H., Cheng, J., Li, C., Liu, Y., Chen, X.: Pulsegan: Learning to generate realistic pulse waveforms in remote photoplethysmography. IEEE Journal of Biomedical and Health Informatics **25**(5), 1373–1384 (2021)
14. Špetlík, R., Franc, V., Matas, J.: Visual heart rate estimation with convolutional neural network. In: Proceedings of the british machine vision conference, Newcastle, UK. pp. 3–6 (2018)
15. Stricker, R., Müller, S., Gross, H.M.: Non-contact video-based pulse rate measurement on a mobile service robot. In: The 23rd IEEE International Symposium on Robot and Human Interactive Communication. pp. 1056–1062. IEEE (2014)
16. Tsou, Y.Y., Lee, Y.A., Hsu, C.T.: Multi-task learning for simultaneous video generation and remote photoplethysmography estimation. In: Proceedings of the Asian Conference on Computer Vision (2020)

17. Tsou, Y.Y., Lee, Y.A., Hsu, C.T., Chang, S.H.: Siamese-rppg network: Remote photoplethysmography signal estimation from face videos. In: Proceedings of the 35th annual ACM symposium on applied computing. pp. 2066–2073 (2020)
18. Verkruysse, W., Svaasand, L.O., Nelson, J.S.: Remote plethysmographic imaging using ambient light. Optics express **16**(26), 21434–21445 (2008)
19. Wang, T.C., Liu, M.Y., Zhu, J.Y., Liu, G., Tao, A., Kautz, J., Catanzaro, B.: Video-to-video synthesis. arXiv preprint arXiv:1808.06601 (2018)
20. Wang, W., den Brinker, A.C., Stuijk, S., De Haan, G.: Algorithmic principles of remote ppg. IEEE Transactions on Biomedical Engineering **64**(7), 1479–1491 (2016)
21. Wang, W., Stuijk, S., De Haan, G.: A novel algorithm for remote photoplethysmography: Spatial subspace rotation. IEEE transactions on biomedical engineering **63**(9), 1974–1984 (2015)
22. Wang, Z.K., Kao, Y., Hsu, C.T.: Vision-based heart rate estimation via a two-stream cnn. In: 2019 IEEE International Conference on Image Processing (ICIP). pp. 3327–3331. IEEE (2019)
23. Yu, Z., Peng, W., Li, X., Hong, X., Zhao, G.: Remote heart rate measurement from highly compressed facial videos: an end-to-end deep learning solution with video enhancement. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 151–160 (2019)
24. Zhang, X., Ng, R., Chen, Q.: Single image reflection separation with perceptual losses. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4786–4794 (2018)
25. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2223–2232 (2017)