Estimating Heart Rate from Unlabelled Video

John Gideon Simon Stent
Toyota Research Institute
Cambridge, MA, USA
{john.gideon, simon.stent}@tri.global

Abstract

We describe our entry for the ICCV 2021 Vision4Vitals Workshop [6] heart rate challenge, in which the goal is to estimate the heart rate of human subjects from facial video. While the challenge dataset contains extensive training data with ground truth blood pressure and heart rate signals, and therefore affords supervised learning, we pursue a different approach. We disregard the available ground truth blood pressure data entirely and instead seek to learn the photoplethysomgraphy (PPG) signal visible in subjects' faces via a self-supervised contrastive learning technique. Since this approach does not require ground truth data, and since the challenge competition rules allow it, we therefore can train directly on test set videos. To boost performance further, we learn a supervised heart rate estimator on top of our "discovered" PPG signal, which more explicitly tries to match the ground truth heart rate. Our final approach ranked first on the competition test set, achieving a mean absolute error of 9.22 beats per minute.

1. Introduction

The success of supervised machine learning is highly dependent on the quality and diversity of the ground truth dataset. In the field of heart rate estimation from facial videos, ground truth is a tricky business.

Firstly, there is no perfect method to capture ground truth. There are a variety of mechanisms to measure cardiac activity through contact with the human body, including photoplethysmography (PPG), blood volume pulse (BVP) or electrocardiography (ECG). Each measure has a slightly different characteristic waveform, and is often out of phase with remotely observable signs of cardiac activity in the face from skin flushing (as measured by remote PPG) and head motion (as measured by ballistocardiography). The phase variability is often due to synchronization errors between video camera and ground truth sensor, but may also change from subject to subject when sensors are placed on different parts of the body, far from the face. Training a model to mimic such signals from the visible evidence of cardiac activity in the face requires learning to overcome these differences.

Secondly, there is no universally used method to compute heart rate – which is the main measurement (and performance metric) of interest. Heart rate can be computed in many different ways from many physiological signals. It can be computed from instantaneous predictions based on spectral analysis of a moving window, or smoothed predictions using acausal filtering, or low-frequency updates via peak-to-peak estimates. Since the key metric in the task of video-based heart rate estimation is to minimize some distance measure between estimated and ground truth heart rate, it is important to understand *how* the ground truth heart rate is computed from the actual observable signal, in order to best match it. Unfortunately, as this computation is often carried out on hardware, it is not always clear.

For this challenge, we decided to apply a new, self-supervised approach to estimate a remote PPG signal from facial video without annotations [1]. A significant advantage of this approach is that it removes the need for ground truth PPG training data, and therefore avoids any known or unknown synchronization issues. While the approach of [1] used a deterministic heart rate estimator (converting the estimated PPG signal into a heart rate), we instead attempted to match the dataset ground truth more explicitly via supervised learning on our predicted PPG signal, as described in Sec. 3. Due to the existence of domain shift between the training and test datasets (Sec. 4), we found that several tricks were required to achieve a strong result on the challenge, as described in Sec. 5. We believe that our final approach, although arguably somewhat tailored to the challenge, is attractive in its simplicity, forgoing some of the complicated pre-processing or feature extraction steps that are popular in remote PPG methods, as well as the need for ground truth data. We discuss our findings and highlight directions for future exploration in Sec. 6.

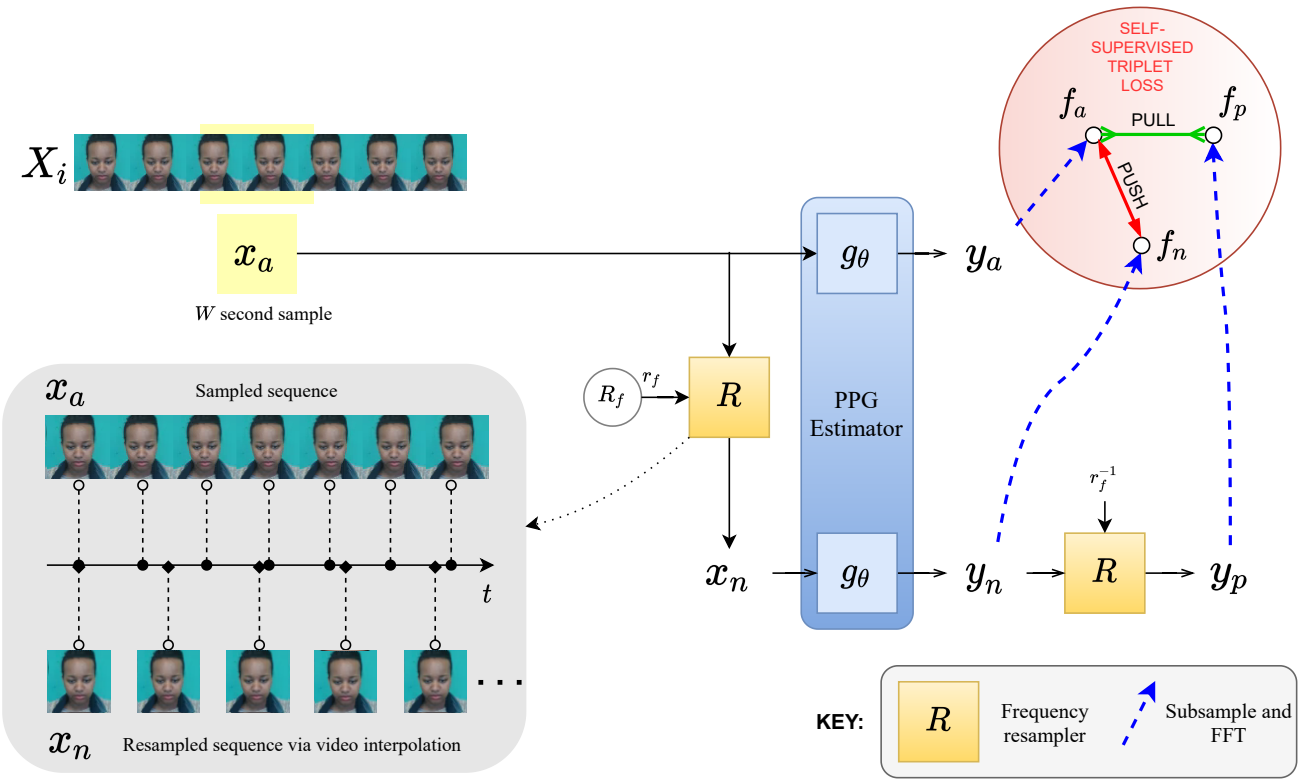

Figure 1. **Overview of our contrastive approach to learn a PPG signal [1].** We first sample an anchor video clip, x_a, of length W from the source video. This anchor is passed through the PPG Estimator g_θ to get y_a. A random frequency ratio r_f is sampled from a prior distribution. The warped clip x_a^s is then passed through the frequency resampler R to produce the negative sample x_n^s. This sample is passed through g_θ to produce the negative example PPG y_a. The negative sample is again resampled with the inverse of r_f to produce a positive example PPG y_p. Finally, the contrastive loss is applied to the PPG samples, using a PSD MSE distance metric, as described in Sec. 2.3.

2. Base Model

An overview of our PPG estimation approach is shown in Fig. 1 and is closely based on concurrent work described in [1]. We now describe each stage of the pipeline in sequence.

2.1. Preprocessing

We preprocess the videos by first estimating a bounding box around the face for each frame using a single-shot scale-invariant face detector [8]. We add a buffer of 25% to the box width and height, to make sure the head and neck are fully captured. For each video, we then smooth the bounding box locations in time using some simple logic: if the non-buffered face box for the current frame is outside the buffered bounding box for the previous frame, then we update. If not, then we keep the bounding box position constant. Updates are carried out smoothly by interpolation over a number of frames (in practice, 0.25 seconds). This ensures relative stability of the video, while still allowing for occasional smooth movement and re-alignment if the face moves significantly. From the smoothed buffered bounding boxes, we extract a $192 \times 128 \times N$ pixel volume, where N is the number of frames in the sequence. In the initial experiment and following [1], we extract these in RGB colorspace, but convert to YUV colorspace in later experiments.

2.2. PPG Estimator

We use a modified version of the 3DCNN-based PhysNet architecture [7] as our PPG estimator, as described in Table 1. The core of PhysNet is a series of eight 3D convolutions with a kernel of (3, 3, 3), 64 channels, and ELU activation functions. This allows for the network to learn spatio-temporal features over the input video. Average pooling and batch normalization are also employed between the layers. In the PhysNet paper, two transposed convolutions are used to return the encoded representation to the original video length. However, we found that these introduced aliasing in the output PPG signal. We modify this part of the network to instead use upsampling interpolation ($\times 4$) and a 3D convolution with a (3, 1, 1) kernel. This upsampling step is repeated twice and removes the aliasing in the output. Next, we perform adaptive average pooling to collapse the spatial

ENCODER	in	out	kernel	stride	pad
Conv3d + BN3d + ELU	3	32	(1,5,5)	1	(0,2,2)
AvgPool3d			(1,2,2)	(1,2,2)	0
Conv3d + BN3d + ELU	32	64	(3,3,3)	1	(1,1,1)
Conv3d + BN3d + ELU	64	64	(3,3,3)	1	(1,1,1)
AvgPool3d			(2,2,2)	(2,2,2)	0
Conv3d + BN3d + ELU	64	64	(3,3,3)	1	(1,1,1)
Conv3d + BN3d + ELU	64	64	(3,3,3)	1	(1,1,1)
AvgPool3d			(2,2,2)	(2,2,2)	0
Conv3d + BN3d + ELU	64	64	(3,3,3)	1	(1,1,1)
Conv3d + BN3d + ELU	64	64	(3,3,3)	1	(1,1,1)
AvgPool3d			(1,2,2)	(1,2,2)	0
Conv3d + BN3d + ELU	64	64	(3,3,3)	1	(1,1,1)
Conv3d + BN3d + ELU	64	64	(3,3,3)	1	(1,1,1)
DECODER					
Interpolate			(2,1,1)		
Conv3d + BN3d + ELU	64	64	(3,1,1)	1	(1,0,0)
Interpolate			(2,1,1)		
Conv3d + BN3d + ELU	64	64	(3,1,1)	1	(1,0,0)
AdaptiveAvgPool3d			(-,1,1)		
Conv3d	64	1	(1,1,1)	1	(0,0,0)

Table 1. **Modified PhysNet-3DCNN architecture**. The architecture follows an encoder-decoder structure with 3D convolutions to represent patterns through time; "s" corresponds to stride, "p" to padding.

dimension and produce a 1D signal. A final 1D convolution is applied to convert the 64 channels to the output single channel PPG.

2.3. Training

Sampling. When training, we randomly sample W seconds from a video X_i. For our experiments, we set W to ten seconds. We denote these subset clips as x_a. We randomly augment our training data by artificially stretching shorter video clips to W seconds using trilinear interpolation.

Contrastive Training. When performing contrastive training, we randomly choose a resampling factor R_f between 66% and 80%. We then pass the anchor video clip x_a through the trilinear resampler R to produce the negative sample x_n. This effectively increases the frequency of the heart rate by a factor of 1.25 to 1.5. Both x_a and x_n are passed through the PPG Estimator g_θ, producing y_a and y_n, respectively. We then resample y_n using the inverse of R_f to output the positive signal y_p, whose frequency should match y_a. Finally, we apply a multi-view triplet loss (MVTL): from the three output branches – anchor (y_a), positive (y_p), and negative (y_n) – we take V_N subset views of length V_L, calculate the distance between all combinations of anchor and positive views (P_{tot}) and anchor and negative views (N_{tot}), then compute $P_{tot} - N_{tot}$ and scale by the total number of views, V_N^2. As the distance metric, we use the power spectral density mean squared error (PSD MSE). We first calculate the PSD for each signal and zero out all frequencies outside the relevant heart rate range of 40 to 180 bpm. We then normalize each to have a sum of one and compute the MSE between them.

Training Parameters. In all experiments we use the AdamW optimizer with a learning rate of 10^{-5}. We set the number of views (V_N) to four and the length (V_L) to five seconds, and we used a batch size of 4. Out models were implemented using PyTorch 1.7.0 [4] and trained on a single NVIDIA Tesla V100 GPU.

2.4. Heart Rate Calculation

Given an estimated PPG signal, we calculate heart rate by (1) zero-padding the signal for higher frequency-resolution, (2) calculating the PSD, and (3) locating the frequency with the maximum magnitude within the relevant heart rate range. We use a simple PSD-based method instead of a learned one to maintain determinism. When calculating instantaneous heart rate, we apply this method over a sliding window of 10 seconds, with a step size of one second. The instantaneous heart rate for each frame within the window is then smoothed using a Hamming window.

3. Confidence Model

In our experiments we noticed that the PPG often becomes noisy when sudden movement is present. In these moments, it is unlikely that an instantaneous heart rate could be learned from the noisy signal and any estimated value is likely to have high error. Because of this, we trained a confidence model to detect these faulty PPG signals and replace the previously estimated instantaneous heart rate with the estimated per-sample median heart rate. This allows for the system to revert to a baseline value when severe noise is present.

The confidence model takes the estimated PPG signal as input and converts it to the frequency domain using a short-time Fourier transform with a window size of 64 and a step size of one. We then take the absolute value of the frequency domain and drop the last value, returning a tensor with 33 channels and matching the original input length. We pass this representation through an encoder consisting of eight alternating 1D convolutions and ELU activations. We employ a kernel size of five, a stride of one, and padding of two for each convolution and use a hidden channel size of 64. The final convolution returns only one channel and is followed by a Sigmoid activation. This produces the final confidence estimate c for each frame of the input PPG and instantaneous heart rate (h_{instant}). We then calculate the median instantaneous heart rate over the entire sequence (h_{median}). The final heart rate prediction, $h_{\text{estimated}}$, for each image frame of the video input, is calculated as follows:

$$h_{\text{estimated}} = c \times h_{\text{instant}} + (1 - c) \times h_{\text{median}} \quad (1)$$

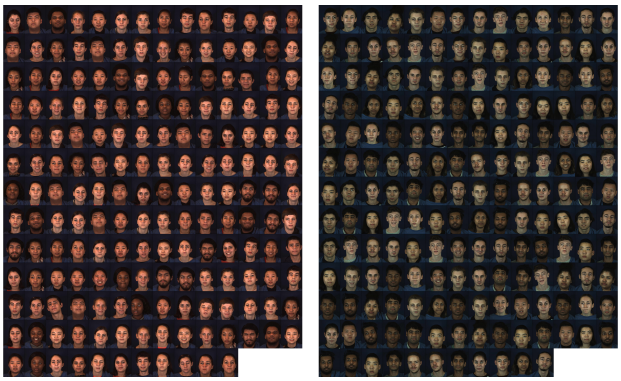

Figure 2. **Out-of-domain lighting variation in the test set.** Clustering the test set by background color shows two distinct lighting modes: bright (left) and dark (right). Since the training dataset only contains bright samples and not dark, methods which only train on the training set must generalize across this lighting domain gap to perform well at test time.

We train this model for 20 epochs using the challenge dataset training/validation split, using an L_1 loss between $h_{\text{estimated}}$ and the ground truth heart rate.

4. Dataset

The challenge data, which was derived from the "BP4D+" dataset of the BP4D Multi-Modal Spontaneous Emotion (MMSE) Corpus [9], contains a total of 1,358 videos with a mean sample duration of 44 seconds and a standard deviation of 32 seconds. As described in [9], there are 140 subjects in the BP4D+ dataset, including 58 males and 82 females, with ages ranging from 18 to 66 years old. The videos are recorded at 25 Hz and 1392×1040 pixel resolution. The dataset is subdivided into train, validation, and test sets with 724, 276, and 358 samples, respectively. Both the train and validation sets are annotated with instantaneous heart and respiratory rates, recorded at 25 Hz. We did not attempt to model respiratory rate in this work. The train set is further annotated with blood pressure estimates recorded at 1 kHz, which we resample to 25 Hz to match the video frame rate. While the train and validation sets only contain one lighting condition, we observe an additional dim lighting condition in the test set, as seen in Fig. 2.

5. Experiments

Here we describe the progression of experiments leading to our final submission. The test set performance of these experiments from the Codalab competition server[1] are shown in Table 2.

[1] https://competitions.codalab.org/competitions/31978

5.1. Baseline

Hypothesis: Our model from [1] should be able to learn to predict heart rate using this large dataset.

Experiment: Train with training set for 100 epochs, using 200 random samples per epoch. Test using the model returned by the final epoch and predict the per-video median heart rate for all frames.

Conclusion: Our initial experiment resulted in an MAE performance of 10.70 bpm, significantly higher than the training and validation MAE. After inspecting the test set, we discovered a domain shift in lighting affecting around half of the test set as illustrated by Fig. 2.

5.2. Train on test set

Hypothesis: Because our model does not use supervision, we can train it directly on the test set, which removes any domain mismatch between training and testing. The domain gap can be further addressed by normalizing the input videos to have zero mean and unit standard deviation across time and transforming them to a YUV color space, as described in [3].

Experiment: Train with the test set for 20 epochs, using all test samples per epoch. Transform all input videos to the YUV color space and Z-normalize them.

Conclusion: By directly trying to address the domain shift and training on the test set, we get an improved MAE performance of 10.32. However, as we still only predict a per-sequence median heart rate at test time, the system is unable to adapt its predictions to changing heart rates in longer videos.

5.3. Instantaneous heart rate

Hypothesis: While estimating the median heart rate likely works well for short sequences, estimating instantaneous heart rate will provide better accuracy for longer videos.

Experiment: We use the PPG estimates from our prior experiment to compute heart rate over a sliding window of 10 seconds, with a step size of one second. The instantaneous heart rate for each frame within the window is smoothed using a Hamming window.

Conclusion: The estimation of instantaneous heart rate was sufficient to improve MAE to 9.96.

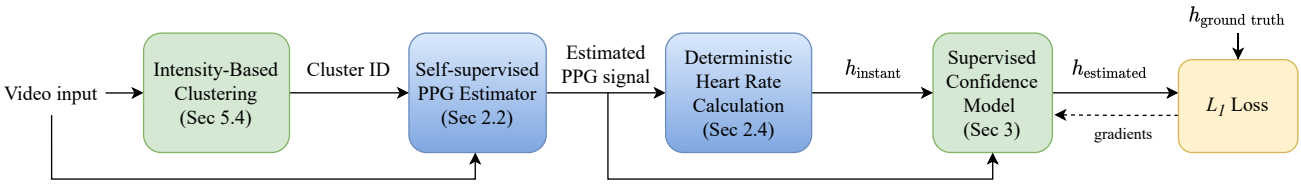

Figure 3. **System diagram for our best-performing test result.** The base model is shown in blue and is fully self-supervised. Refinements to the model are shown in green and their impact on test performance is described in Sec. 5.

Approach	MAE ↓	RMSE ↓	R ↑
Sec. 5.1: Baseline	10.70	15.17	0.36
Sec. 5.2: Train on test set	10.32	15.10	0.40
Sec. 5.3: Instantaneous heart rate	9.96	14.98	0.43
Sec. 5.4: Specialist Models	9.44	14.44	0.45
Sec. 5.5: Confidence detection	**9.22**	**14.18**	**0.47**

Table 2. **Selected Heart Rate Challenge results.** For detailed descriptions please see Sec. 5

5.4. Specialist models

Hypothesis: We still believe that the presence of multiple lighting modes in the test set is creating issues for our model, which might not have the capacity to learn both modes well. We hypothesize that models which specialize in single lighting modes may perform better than a model which tries to learn both. We also believe that we can further stabilize the training process by monitoring the negative maximum cross-correlation (MCC) [1] on the held-out train set and selecting the epoch with the minimum value.

Experiment: We cluster the training set based on average background intensity, which yields two clusters of equal size, and we train a specialist model for each cluster for 20 epochs each. We calculate the MCC using the training set at each epoch, and select as our test model the snapshot at the epoch with minimum MCC. We follow the instantaneous heart rate method as before.

Conclusion: This specialization in models results in the largest performance improvement yet – to an MAE of 9.44.

5.5. Confidence detection

Hypothesis: When observing the estimated PPG signals, there are clear instances when the signal is lost – usually due to sudden facial movement. While it is common to spend more effect to try to undo facial movement through more precise facial landmark tracking, due to time restrictions we think a coarse but simpler approach may be still be valuable to try to detect and smooth over these moments of uncertainty.

Experiment: We use the estimated PPG and instantaneous heart rates from the prior experiment. We then train a confidence model, described in Section 3, to determine a confidence weight for each frame of the PPG. This weight is then used to determine the contribution of either the instantaneous heart rate or the per-sequence estimated median heart rate per frame. An illustration of the complete system at this point is shown in Fig. 3.

Conclusion: Our confidence method allows the heart rate estimation to recover from some bad failures in PPG estimation, improving the MAE performance of our system to 9.22.

6. Discussion

We have shown how a relatively simple self-supervised approach can perform well at the task of heart rate estimation, potentially even against more complex approaches which rely on precise facial landmark tracking or supervision from expensive ground truth data. While our final challenge entry relied on training on the test set – a luxury afforded to us by the challenge rules and by virtue of our method – we note that our learned model also permits online estimation on truly held-out test data.

This challenge has brought into focus two important issues. Firstly, we believe it has helped to highlight the potential limitations of heart rate metrics when evaluating remote PPG estimation methods. On the challenge dataset, we believe that significant further performance gain could be found by improving our PPG to heart rate estimator. To avoid artificial performance gains or losses due to this, a fairer comparison between video-based PPG estimation methods might be achieved by (i) the inclusion of metrics reliant on only the underlying physiological signal of interest (i.e. PPG), or (ii) adopting a common heart rate estimation approach for all methods, including the ground truth.

Secondly, it is clear that dealing with domain shift remains a problem in remote PPG, as shown by the relative success of our approach of learning specialized rather than general models. While we did not have time to try a meta-learning approach such as [2], we are curious as to whether such techniques might help to address this problem when moving from training to test.

Task ID	Task target emotion	Head pitch variation	Train set MAE ↓
3	sadness	4.6	4.1
9	angry	5.9	4.5
2	surprise	4.5	5.0
5	skeptical	4.6	5.1
7	fear/nervous	6.1	5.4
1	happiness/amusement	4.9	5.7
8	physical pain	5.7	6.0
6	embarrassment	6.1	6.1
10	disgust	7.3	6.4
4	startle/surprise	5.6	6.9

Table 3. **Breakdown of estimated heart rate estimation error per video task ID.** Since the training data includes both ground truth heart rate plus the task ID for each video – corresponding to the task that the subject was doing while being recorded – we can compute the training set MAE of our model on a per-task basis, and compare this against the task target emotions and average head pitch variation [9]. Here we show head pitch variation in green for "lower" ($< 5^\circ$) and red for "higher" ($\geq 5^\circ$). The five task IDs with the lowest training set MAE have an average head pitch variation of 5.1 degrees and average MAE of 4.8 bpm. In contrast, the five *highest* have average pitch variation 5.9 (+0.8) and average MAE 6.2 (+1.4). This suggests that head pose variation has a significant negative influence on heart rate estimation. Note that while these figures are computed on the workshop challenge "train" set (in order to know the ground truth), the model used here did not have access to the train set during training.

6.1. Things we thought about but did not try

Finer-grained specialist networks: We speculate that clustering in subject identity (rather than image intensity) in order to train a larger number of per-subject PPG models might yield further performance improvements. However, our initial efforts suggest that the per-subject data volume was not quite sufficient to support this level of specialization when training from scratch. However, it may be possible to use fine-tuning approaches or to create an improved system that determines the confidence of specialist models and falls back to more general models when necessary.

Improving input video stability: One significant source of error in PPG estimation is caused by subject motion, as shown in Table 3. It is likely that improved facial tracking could help to alleviate this by making the input features more spatially stable. Alternatively, it could be beneficial to recognize *when* subject motion is present and fall back to a known baseline heart rate estimate. Although we achieved this to some extent through a supervised confidence mechanism, we believe other, potentially unsupervised approaches may yield better results.

Test adaptation: While it was feasible to train on the test data in this competition, this would not be possible for on-line use cases. Besides trying to expand the domain over which video heart rate estimation methods work well, it is also important to explore domain adaptation or meta-learning methods which can adapt quickly to unseen domains of data at test-time.

Acknowledgements. We wish to thank the Vision4Vitals Workshop organizers for arranging the challenge and sharing access to their interesting dataset, and the workshop reviewers for their thoughtful feedback. We have added further discussion in response to reviewer comments to the Appendix.

References

[1] John Gideon and Simon Stent. The way to my heart is through contrastive learning: Remote photoplethysmography from unlabelled video. In *ICCV*, 2021. 1, 2, 4, 5, 7

[2] Eugene Lee, Evan Chen, and Chen-Yi Lee. Meta-rPPG: Remote Heart Rate Estimation Using a Transductive Meta-Learner. In *ECCV*, 2020. 5

[3] Xuesong Niu, Shiguang Shan, Hu Han, and Xilin Chen. Rhythmnet: End-to-end heart rate estimation from face via spatial-temporal representation. *IEEE Transactions on Image Processing*, 29:2409–2423, 2019. 4

[4] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *NeurIPS*, pages 8024–8035. 2019. 3

[5] Adria Recasens, Petr Kellnhofer, Simon Stent, Wojciech Matusik, and Antonio Torralba. Learning to zoom: a saliency-based sampling layer for neural networks. In *ECCV*, pages 51–66, 2018. 7

[6] Ambareesh Revanur, Zhihua Li, Umur A. Cifti, Lijun Yin, and László A. Jeni. The First Vision For Vitals (V4V) Challenge for Non-Contact Video-Based Physiological Estimation. In *ICCV Workshops*, 2021. 1

[7] Zitong Yu, Xiaobai Li, and Guoying Zhao. Remote Photoplethysmograph Signal Measurement from Facial Videos Using Spatio-Temporal Networks. In *BMVC*, 2019. 2

[8] Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and Stan Z Li. S3FD: Single shot scale-invariant face detector. In *ICCV*, pages 192–201, 2017. 2

[9] Zheng Zhang, Jeffrey M. Girard, Yue Wu, Xing Zhang, Peng Liu, Umur Ciftci, Shaun Canavan, Michael Reale, Andrew Horowitz, Huiyuan Yang, Jeffrey F. Cohn, Qiang Ji, and Lijun Yin. Multimodal spontaneous emotion corpus for human behavior analysis. In *CVPR*, pages 3438–3446, 2016. 4, 6

A. Appendix

A.1. Reviewer-Prompted Discussion

How robust is a contrastive approach vs. a supervised approach? We did not have time during the test phase to measure the relative performance of a contrastive vs. supervised approach on this dataset. However, in our concurrent work we have found that on numerous other remote PPG datasets, our contrastive method can perform comparably to and sometimes better than supervised learning [1].

We agree that it remains unclear how well the contrastive method may perform compared to a supervised method if the training data is highly varied. If there is input noise within the valid heart range (here we set this to be 40-180 Hz), this could certainly lead to our model learning the wrong signal. This is particularly troubling if the dataset has no ground truth, since there is no way to diagnose the error through a quantitative metric. However, in our concurrent work we introduce the use of a saliency sampler [5] to help with qualitative diagnosis: by highlighting which parts of the input image are used by the model to help determine the estimated heart rate, a practitioner can get visual confirmation that the model is behaving as expected. We did not use a saliency sampler here for expediency, as ground truth metrics were available and qualitative analysis takes time.

We think it is true that a supervised method is less likely to suffer from input noise, since the noise would have to closely match the ground truth target signal in order to cause the model to "learn a wrong shortcut". However, we are yet to see strong evidence that this is a problem in practice, as measured across five independent datasets (four in [1] and a further one here, assuming competing methods in the challenge used supervised approaches). That said, making the contrastive approach more robust during training is certainly an area for future work.

Is the approach applicable to respiration rate (RR) detection? It is possible that the same method could be used to try to detect respiration rate, with a different prior over "valid" frequencies (e.g. 5-25 breaths per minute rather than 40-180 beats per minute). However, we note that there are several key differences in the application: firstly, respiration rate is largely visible through the motion cue of a person's chest rising and falling, mouth opening and closing, or nostrils flaring. In contrast, the primary cue used by our network for rPPG is the appearance change in skin tone. For RR detection, the model might therefore benefit from using optical flow or frame-to-frame differencing to help accentuate the motion cue. Secondly, while the PPG signal tends to fluctuate in a reasonably smooth manner, since it is tied to the physics of blood pumping through body tissue, respiration can be far more disjointed and inconsistent in time, since breathing can be disrupted by many factors such as talking, motion and conscious control. Therefore the frequency spectrum of a respiration signal is likely to be more spread out than that of a PPG signal, which we think would make the contrastive approach more challenging. We agree, though, that it may be another interesting avenue to try to extend the method.