Neurodata Lab's approach to the Challenge on Computer Vision for Physiological Measurement

Mikhail Artemyev[1], Marina Churikova[1,2], Mikhail Grinenko[1], and Olga Perepelkina[1]

[1]Neurodata Lab LLC, Miami, USA
[2]Lomonosov Moscow State University, Faculty of Biology, Department of Higher Nervous Activity, Moscow, Russia
m.artemyev@neurodatalab.com, m.churikova@neurodatalab.com, m.grinenko@neurodatalab.com, o.perepelkina@neurodatalab.com

Abstract

This paper introduces the Neurodata Lab's approach presented at the 1st Challenge on Remote Physiological Signal Sensing (RePSS) organized within CVPR2020. The RePSS challenge was focused on measuring the average heart rate from color facial videos, which is one of the most fundamental problems in the field of computer vision.

Our deep learning-based approach includes 3D spatio-temporal attention convolutional neural network for photoplethysmogram extraction and 1D convolutional neural network pre-trained on synthetic data for time series analysis. It provides state-of-the-art results outperforming those of other participants on a mixture of VIPL and OBF databases: MAE=6.94 (12.3% improvement compared to the top-2 result), RMSE=10.68 (24.6% improvement), Pearson $R = 0.755$ *(28.2% improvement).*

1. Introduction

The 1st Challenge on Remote Physiological Signal Sensing (RePSS) in CVPR2020 was organized by X. Li et al. [9]. Remote detecting of physiological parameters from videos is a promising and noninvasive method that enables to undertake ubiquitous monitoring of humans in natural living conditions [13].

Heart rate (HR) is one of the most important physiological parameters that let us evaluate an individual's health and affective state [15]. The HR can be measured both with contact and contactless methods. Compared with common electrocardiography (ECG) and photoplethysmography (PPG) measurements which require direct contact of specific sensors with a subject's skin, remote PPG (rPPG) is a contactless technique for HR monitoring that requires only ambient light and a digital camera. Due to this circumstance, this method has many potential applications including those in sports and fitness, individual healthcare, patient/driver status monitoring [4], etc. For this reason, a facial video-based rPPG technique has attracted significant attention in the last few years, and the number of published papers on this subject is growing every year. Yet there is a lack of high-quality publicly available rPPG databases which complicates further development of this research area.

Organizers of the 1st RePSS Challenge have provided a large benchmark dataset that consists of two sets – training and test sets. The training set contains 2500 pieces of 10s videos of 500 persons (i.e. 5 videos for each person) from VIPL-HR V2 database, and the test set contains 1000 pieces of 10s videos of 200 persons (100 from VIPL-HR V2 database, and 100 from OBF database) [9].

2. Related works

In recent years, there has been proposed a number of video-based HR estimation methods that can be divided into three large groups: (a) those based on blind signal separation (BSS), (b) those based on optical model, and (c) data-driven methods [11]. In this section, we review mostly the recent deep learning approaches for remote HR measurement.

The first approach in this group that involves deep convolutional network is DeepPhys – an end-to-end system for video-based HR measurement originally proposed by W. Chen and D. McDuff [3]. R. Spetlik et al. [16] designed the HR-CNN which remotely detects HR using a two-step convolutional neural network (CNN) using aligned face images. Niu et al. designed a method to obtain a large volume of synthetic PPG signals to train a deep heart rate estimator specifically for cases where data is limited [12]. Furthermore, they also proposed a novel effective approach that applies

data augmentation to overcome the limitation of training data [11]. Besides, F. Bousefsaf et al. [2] proposed a 3D CNN trained only on synthetic data. The authors used a public UBFC-RPPG dataset [1] to validate this network and prove that it can accurately measure heart rate from videos.

3. Our method

Our pipeline (see Fig. 1) includes:

1. Video pre-processing and training set augmentation.

2. Deep learning-based heart rate estimation.

3. Post-processing of test set predictions which is specific to the "CVPM2020 challenge: The 1st Challenge on Remote Physiological Signal Sensing" dataset structure.

This section describes these steps in detail.

3.1. Training set augmentation

We change video frame rate, while leaving the original frames sequence unchanged to perform speed-up and slow-down video augmentation. We use it to increase size of both the training dataset and variance of the reference heart rate distribution. The reference heart rate (in beats per minute) has log-normal distribution with $\mu = \ln(85)$ and $\sigma = 0.25$ after the augmentation. The standard deviation of the reference heart rate in the augmented dataset is larger than in the original dataset. This should improve the performance of the algorithm, especially for subjects with very low or very high heart rate.

We use horizontal flip augmentation during training as well.

3.2. Video pre-processing

First, our method detects faces using a RetinaNet network [10] with MobileNet backbone [7] trained with focal loss [10]. In order to simplify the following method description, we assume that there is only one person in the video. Then we perform affine face alignment based on facial landmarks detection [5] for each face. An example of region of interests (ROI) is shown in the Fig. 2.

Figure 2: An example of ROI visualisation.

We use ROI average pooling [14] to resize facial areas to the size of $W \times H$ for the heart rate estimation neural network, where $W = H = 36$. After that, resampling to 25 fps by cubic interpolation is performed. Bandpass finite impulse response (FIR) filter with a length of 2 seconds and (45 beats per minute (bpm), 180 bpm) cutoff frequencies is applied for each (pixel, channel) pair independently in order to filter out signals not related to pulse cycles.

3.3. Heart rate estimation neural network

We train a convolutional neural network (see Fig. 3 (a)) to estimate median heart rate in 10-second video fragments (8 seconds or $T = 200$ frames after bandpass filtering). A common way to obtain a PPG signal using the given ROI is to compose global spatial average pooling with signal source separation methods. While global pooling is an efficient way of getting rid of noise, it can corrupt the signal if a face moves or if ROI is covered by a foreign object (such as hair or hands); such artifacts may be difficult to filter out during the subsequent steps. We use 3D spatio-temporal attention neural network (see Fig. 3 (b)) prior to the global pooling to address this issue. We call this network a 3D CNN (see section 3.4). It enables us to do three things simultaneously: to choose the ROI that best suits the purpose of heart rate detection in each frame, to select the optimal nonlinear function of color channels, and to complete signal filtering using temporal information.

Unlike most computer vision tasks, frequency analysis of temporal signal is critical for the rPPG analysis, while each frame itself does not contain information about the target variable. To address this issue, we include a separate 1-dimensional convolutional part (1D CNN, see section 3.5) in our network architecture, and pre-train it to evaluate heart rate on synthetic PPG-like curves (see section 3.6).

3D CNN outputs 32 time series, one for each channel of the last convolutional layer. Each time series is processed with a pre-trained 1D CNN. Therefore, we get 32 heart rate estimations, which are combined into a single output with a 2-layer perceptron.

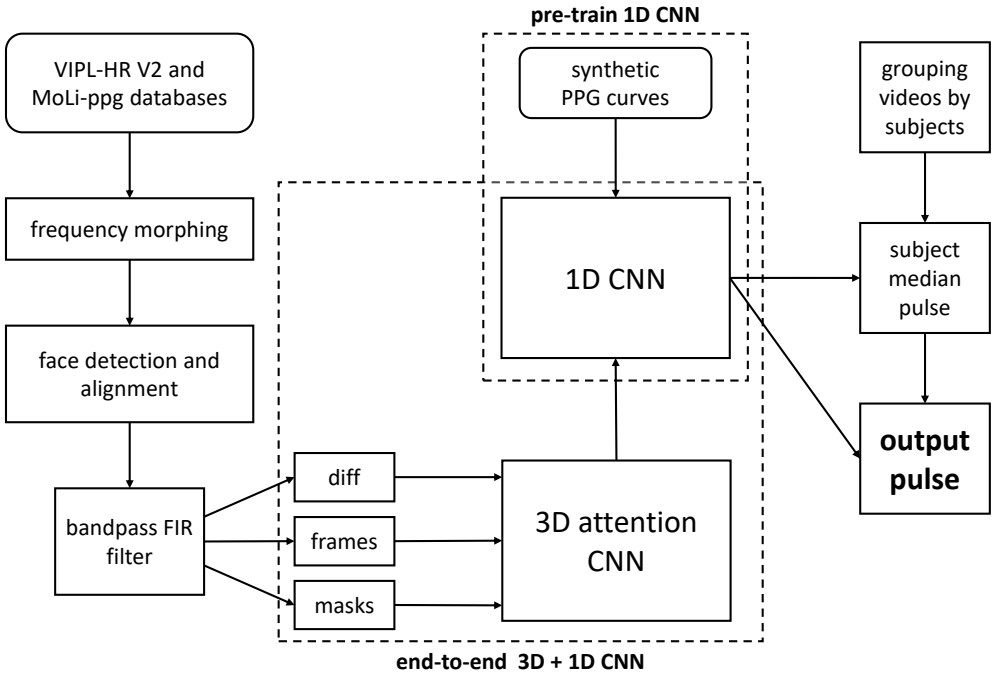

Figure 1: Neurodata Lab's solution pipeline.

3.4. 3D spatio-temporal attention neural network

3D CNN has 3 inputs:

- *diff* input is a discrete time derivative of the pre-processed frame sequence described above. Its size is $batch\_size \times T \times W \times H \times 3$. We use *diff* as the main source of pulse information in our network.

- *frames* input of size $batch\_size \times T \times W \times H \times 3$ consists of the pre-processed frames themselves.

- *masks* input of size $batch\_size \times T \times W \times H \times 1$ consists of frame-wise facial masks. We build a facial mask for each frame first, with the value of each pixel being equal to 0 if the corresponding pixel in the ROI belongs to eyes or mouth or does not belong to the facial area at all; otherwise, its value is considered equal to 1. In order to evaluate *mask* input tensor, we apply ROI mean pooling to the facial mask. Facial landmarks detection [5] is used for face, mouth and eye area localization.

Diff input goes through two 3D convolutional blocks with subsequent average pooling layers. The first block has 16 channels with a kernel size of $3 \times 3 \times 3$, and ends with an average pooling layer with kernel size and stride of $1 \times 2 \times 2$; the second one has 32 channels with a kernel size of $5 \times 3 \times 3$ and ends with a global average pooling layer.

Each convolutional block ("3D Conv Block" at Fig. 3 (b)) with kernel size $t \times w \times h$ and c channels has sequential structure and consists of the following layers:

· 3D Convolution, c channels, $kernel\_size = 1 \times w \times h$
· ReLU activation
· 3D Convolution, c channels, $kernel\_size = t \times 1 \times 1$
· Batch Normalization
· ReLU activation
· 3D Convolution, c channels, $kernel\_size = 1 \times w \times h$
· ReLU activation
· 3D Convolution, c channels, $kernel\_size = t \times 1 \times 1$
· Batch Normalization
· ReLU activation
· Dropout layer $(p = 0.25)$

We also tried to use 3D convolutions with kernel size $t \times w \times h$, but a model with separable convolutions ($1 \times w \times h$ and $t \times 1 \times 1$) performed slightly better in our experiments.

We use *mask* data in two ways. First, it is concatenated to *diff* tensor in channels' axis. Second, after each convolutional block, all elements of the internal representation of *diff* channel are multiplied by zero if any of the corresponding *mask* values is not equal to 1. This way we get rid of possible influence of the background on pulse estimation. In order to choose parts of a face most suitable for pulse tracking at every particular moment, we use attention mechanism (see Fig. 3 (c, d)).

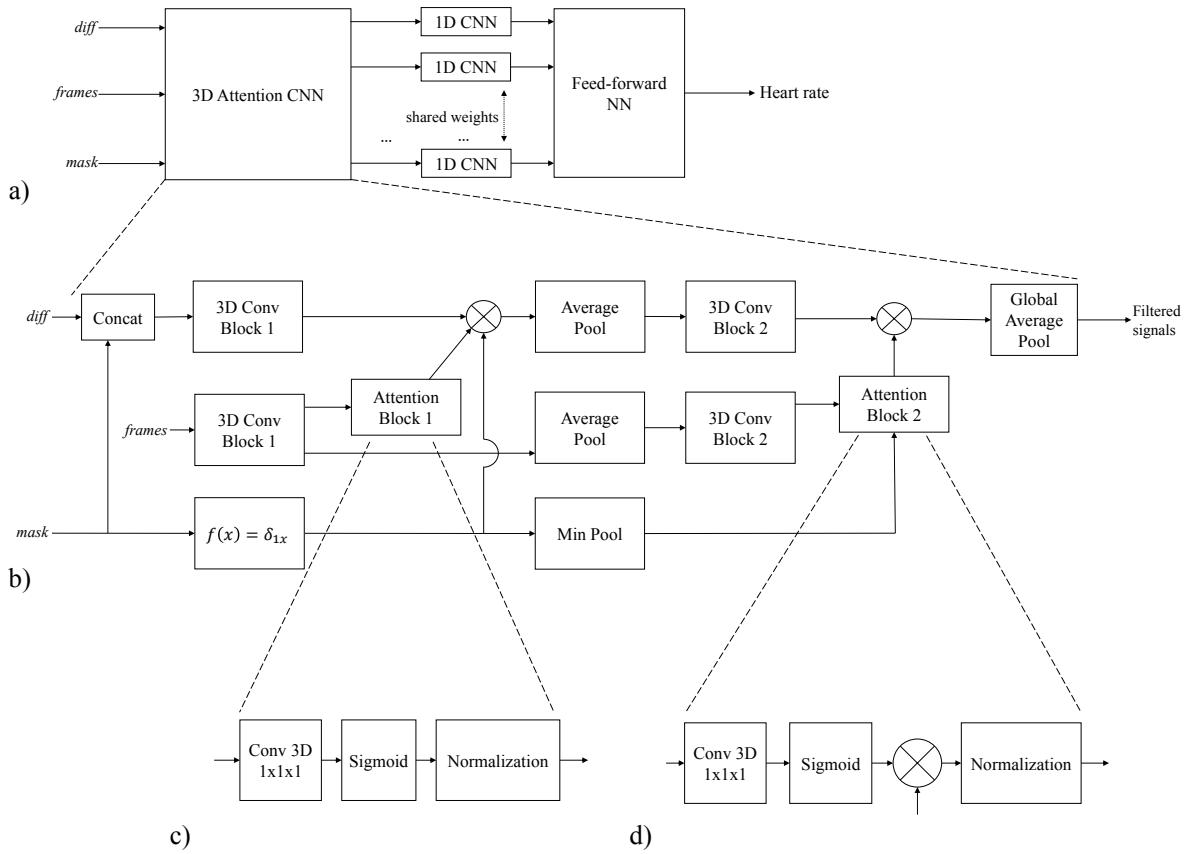

Figure 3: a) Heart rate estimation neural network (see section 3.3) consists of three parts. The first part is a 3D spatio-temporal attention CNN for nonlinear color fusion and weighted average pooling over rPPG-relevant face regions. The second part is a 1D CNN that was pre-trained to evaluate average heart rate using synthetic PPG curves. The said 1D CNN is applied to each channel of the 3D CNN independently. The third part is a 2-layer perceptron which aggregates the heart rate estimations. b) 3D CNN (see 3.4) uses *diff* input for pulse rate evaluation. *Diff* goes through the two spatio-temporal convolution blocks, each of which is followed by attention-based rPPG-relevant region selection and average pooling. *Frames* and *masks* inputs are used for attention weights evaluation. c), d) represent attention blocks. The normalization in these blocks is a division of the value in every pixel by mean value in the corresponding frame

We use (pre-processed) RGB frames in attention blocks, since they are commonly acknowledged to be suitable for detecting face parts and foreign objects. We divide attention weights by their mean value over W, H dimensions in the end of Attention blocks.

3.5. Time Series analysis network

In order to obtain heart rate values from the time series extracted from one of the 3D CNN channels, we use 1D CNN with the following sequential architecture:

- Instance Normalization
- 1D Conv, 16 channels, $kernel\ size = 3$
- ReLU activation
- 1D Conv, 16 channels, $kernel\ size = 3$
- Batch Normalization
- ReLU activation
- Max Pooling, $kernel\ size = 2$, $stride = 2$
- 1D Conv, 32 channels, $kernel\ size = 3$, $dilation = 2$
- Batch Normalization
- ReLU activation
- Max Pooling, $kernel\ size\ = 2$, $stride = 2$
- 1D Conv, 64 channels, $kernel\ size = 3$, $dilation = 2$
- Batch Normalization
- ReLU activation
- Max Pooling, $kernel\ size = 2$, $stride = 2$
- 1D Conv, 128 channels, $kernel\ size = 3$, $dilation = 2$
- Batch Normalization
- ReLU activation
- Global Max Pooling
- Fully Connected Layer, 30 neurons

· tanh activation
· Fully Connected Layer, 30 neurons
· tanh activation
· Fully Connected Layer, 1 neuron

3.6. Synthetic PPG curves

We use synthetic data to pre-train the 1D CNN part of the network. We sample PPG curves with the following formula:

$$s(t) = A\sin\left(2\pi\int_0^t hr(\tau)d\tau + \phi_{hr}\right) +$$

$$A_2\sin\left(4\pi\int_0^t hr(\tau)d\tau + \phi_{hr}\right) +$$

$$B\sin\left(2\pi\int_0^t br(\tau)d\tau + \phi_{br}\right) + Cn(t),$$

where $hr(\tau)$ is an instantaneous heart rate value, $br(\tau)$ is an instantaneous breath rate value, ϕ_{hr} is an initial phase of the heart cycle, ϕ_{br} is an initial phase of the breath cycle, A is a magnitude of the pulse signal, A_2 is a dicrotic pulse magnitude, B is a breath signal magnitude, $n(\tau)$ is a white noise sample, and C is the standard deviation of the noise.

$hr(\tau)$ can be sampled from a uniform distribution $hr_0 \pm \delta_{hr}hr_0$, where hr_0 is a reference heart rate on the segment, and δ_{hr} refers heart rate variability (we use, $\delta_{hr} = 0.05$). In the same way we introduce breath rate variability parameter $\delta_{br} = 0.1$. Amplitudes of the signals are sampled from uniform distributions $A \sim [0.2, 0.7]$, $A_2 \sim [0, 0.3]$, $B \sim [0.3, 2]$. We use $C = 0.05$.

A sampled curve example is shown in Fig. 4.

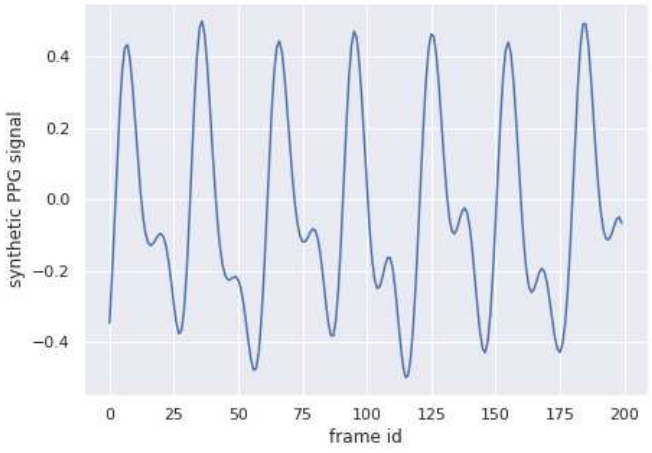

Figure 4: Examples of synthetic PPG signal with heart rate = 51 bpm.

3.7. Training procedure

First, we perform Xavier initialization [6] with a magnitude = 2.34 for all of the model weights.

We pre-train the 1D CNN network for the task of heart rate value estimation by PPG curve. For this purpose, we synthesise 10^6 PPG curves as described in section 3.6 with a reference pulse rate uniformly distributed in [45bpm,180bpm] interval. We use Adam [8] to optimize MSE loss with respect the 1D CNN model weights. We train the model for 100 epochs with $batch\_size = 32, learning\_rate = 3 \times 10^{-5}$.

After that, we train the network end-to-end on MoLi-ppg (see section 4.1) video sequences, with l_2 regularization coefficient equal 10^{-5} using Adam optimizer with the learning rate exponentially decreasing from 10^{-4} to 10^{-5} with batch size= 16 during 200 epochs.

Finally, we train the network end-to-end on VIPL-HR V2 set (see section 4.1) video sequences with the same parameters, except for constant learning rate $= 10^{-5}$ and number of epochs = 900.

We have implemented our heart rate estimation pipeline using MXNet framework (https://mxnet.apache.org). The network was trained on 1 NVIDIA GeForce GTX 1080Ti GPU. Our implementation can be tested via API [1].

3.8. The 1st Challenge on RePSS predictions post-processing

There are 5 video fragments featuring each subject in the RePSS challenge dataset. According to the training set, these 5 fragments have nearly the same reference heart rate for most of the subjects. Let $(p_1, p_2, p_3, p_4, p_5)$ be the neural network outputs on these fragments for some subject. Then our final heart rate estimation of a video fragment is:

$$f_i = 0.01 \times p_i + 0.99 \times \text{median}(p_1, p_2, p_3, p_4, p_5).$$

The fragments were not grouped by subjects in the test set. We match each video with other videos of the same subject to evaluate median value. For this purpose, we first evaluate a simple embedding of the first frame for each video. This embedding for VIPL dataset videos consists of RGB colors of pixels of two 100×150 rectangles (top-left and top-right), each resized to 10×15. So, each of the VIPL videos embeddings consists of $2 \times 10 \times 15 \times 3 = 900$ integer values in the range [0, 255] and represent background color information. All OBF videos have the same background, for this reason we used chest area (bottom 420×1080 pixels rectangle resized to 8×20) as a color embedding for OBF videos. We use $1 - R(a, b)$ as a distance metric on the embeddings described above, where R is a Pearson correlation coefficient. Videos were grouped by subjects using an iterative DBSCAN procedure as follows. First we set $\varepsilon = 0.01$

[1] https://api.neurodatalab.dev

in DBSCAN, and then gradually increase it up to 0.4. If there are any clusters of the size 5 on each iteration, we assume that each of these clusters corresponds to videos of one subject. These videos are not considered in the subsequent clustering iterations.

4. Experiments

4.1. Datasets

We used three datasets for training (**Mo**tion and **Li**ght **p**hoto**p**lethysmo**g**raphy (MoLi-ppg-1) dataset, the MoLi-ppg-2 dataset, and the VIPL-HR V2), and two datasets for testing (VIPL-HR V2 and OBF) [9]. The latter two were provided by the organizers of this challenge.

MoLi-ppg-1 and MoLi-ppg-2 rPPG datasets are new and include videos recorded in complicated and close to natural conditions; in particular, they feature movements, speech, different lighting, various equipment etc.

The first of these three datasets contains 8 hours of video recordings of 25 subjects. The videos were recorded with the following webcams: Logitech C920, Logitech C270, and an HD video camera Canon LEGRIA HFG40. The second dataset was recorded with different cameras and different subjects. It contains 3,5 hours of video recordings of 15 new subjects. The videos were recorded with a webcam Canyon 720p, and an HD video camera Panasonic. The ground truth data collected by contact PPG (cPPG) for both datasets was obtained with an optical pulse sensor Shimmer3 GSR+ (www.shimmersensing.com) attached to subjects' fingers (sampling rate = 256 Hz), and the data was synced with the video recording. The videos from the webcams were in uncompressed bitmap format with either 800x600 or 1280x720 pixel resolution, and 25 fps frame rate. The videos from HD cameras were in uncompressed bitmap format with 1920x1080 pixel resolution and 50 fps. A total of 35 subjects aged 18-35 - both males and females - took part in the experiments. Subjects were lit by fluorescent ceiling lamps and were sitting in front of the cameras at a distance of about 1m from them.

Three first dataset was recorded in three types of conditions (MoLi-ppg-1):

1. **Static.** The subjects were recorded in varying lighting conditions (90-300 lux) while they were sitting naturally in front of the webcam. In particular, they were recorded in the lighting conditions of a) only fluorescent ceiling lamps, b) fluorescent ceiling lamps with an additional spotlight, and c) fluorescent ceiling lamps accompanied by a turned on computer monitor that played videos.

2. **Movements.** Three cases of videos with head motions recorded in standard conditions included large and small movements as well as speech. In the first two cases, the subjects were instructed to perform various types of head movements: left-right, up-down and in circular motion. The amplitude of these movements measured from the position of straight head posture had to be no more than 45 degrees for the task with small head movements, and no more than 80 degrees for the one with large head movements. As for the speech subcategory, the participants were asked to sit facing the cameras and read a text out loud without moving their heads at all.

3. **Recovery after physical stress.** To obtain more broad distribution of heart rate, each subject was asked to do 20-30 squats and was recorded immediately after that.

The second dataset (MoLi-ppg-2) also included three categories of videos:

1. **Static.** The subjects were recorded in varying lighting conditions (20-300 lux): a) daylight without lamps, b) fluorescent ceiling lamps with an additional spotlight, c) fluorescent ceiling lamps accompanied by a turned on computer monitor that played videos.

2. **Speech.** This category includes videos featuring small natural head motions during speech.

3. **Recovery after physical stress.** Each subject was asked to do 20-30 squats and was recorded immediately after that, just like in the first dataset.

VIPL-HR V2 is a large dataset that served as a benchmark for this competition. The train set contains 2500 pieces of 10s videos of 500 persons, and the test data – 1000 pieces of 10s videos of 200 persons (100 from VIPL-HR, 100 from OBF) [9].

4.2. Evaluation Metrics

To evaluate the performance of our approach on the VIPL-HR V2 database, the following metrics were used:

- **Mean Absolute Error (MAE)** in beats per minute (bpm) is calculated as the mean difference between the pulse obtained from rPPG signals and the pulse obtained from cPPG signals with $\frac{\sum_{v \in \text{videos}} \sum_{k=1}^{T_v} |rPPG_{v,k} - cPPG_{v,k}|}{\sum_{v \in \text{videos}} T_v}$, where T_v is the number of frames in the video v.

- **Root mean square error (RMSE)** = $\sqrt{\frac{\sum_{v \in \text{videos}} \sum_{k=1}^{T_v} (rPPG_{v,k} - cPPG_{v,k})^2}{\sum_{v \in \text{videos}} T_v}}$.

- **Pearson correlation coefficient (R)** = $\frac{\sum \left[(rPPG_{v,k} - \frac{\sum rPPG_{v,k}}{T_v})(cPPG_{v,k} - \frac{\sum cPPG_{v,k}}{T_v}) \right]}{\sqrt{\sum \left(rPPG_{v,k} - \frac{\sum rPPG_{v,k}}{T_v} \right)^2 \sum \left(cPPG_{v,k} - \frac{\sum cPPG_{v,k}}{T_v} \right)^2}}$

5. Results

We trained the proposed CNN to perform rPPG monitoring from videos on MoLi-ppg-1, MoLi-ppg-2, and VIPL-HR V2 databases as described above (see section 3.7), and tested (see section 3.8) it on the 1st RePSS Challenge test set. It contains samples from VIPL-HR V2 and OBF databases (see [9]). In this test, our approach has outperformed the methods of other participants as we have achieved MAE = 6.94 (12.3% improvement compared to the challenge top-2 result), RMSE = 10.68 (24.6% improvement), Pearson $R = 0.755$ (28.2% improvement).

6. Conclusion

In this paper, we proposed Neurodata Lab's approach to rPPG monitoring from video. Heart rate data estimated with this method was submitted for the 1st Challenge on RePSS organized within CVPR2020. Challenge organizers provided a large-scale dataset (500 and 200 subjects for training and testing, respectively, with 5 videos for each subject) with different recording scenarios e.g. talking, moving, with different lighting or video frame rate. The amount and diversity of training videos encourage the development of deep learning-based heart rate recognition approaches.

Our data-driven approach includes 3D spatio-temporal attention CNN for PPG extraction, and 1D CNN pre-trained on synthetic data for time series analysis. We believe that our architecture and the way we train the network is specific for the rPPG analysis and heart rate recognition, even though 3D CNN and attention networks are widely used in computer vision. Our method has shown state-of-the-art results in the Challenge. We consider MAE values of 6.94 bpm a good result for "in the wild" videos processed with a mosaic filter.

References

[1] Serge Bobbia, Richard Macwan, Yannick Benezeth, Alamin Mansouri, and Julien Dubois. Unsupervised skin tissue segmentation for remote photoplethysmography. *Pattern Recognition Letters*, 124:82–90, 2017. 2

[2] Frédéric Bousefsaf, Alain Pruski, and Choubeila Maaoui. 3d convolutional neural networks for remote pulse rate measurement and mapping from facial video. *Applied Sciences*, 9(20):4364, 2019. 2

[3] Weixuan Chen and Daniel McDuff. Deepphys: Video-based physiological measurement using convolutional attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 349–365, 2018. 1

[4] Xun Chen, Juan Cheng, Rencheng Song, Yu Liu, Rabab Ward, and Z Jane Wang. Video-based heart rate measurement: Recent advances and future prospects. *IEEE Transactions on Instrumentation and Measurement*, 2018. 1

[5] Xuanyi Dong, Shoou-I Yu, Xinshuo Weng, Shih-En Wei, Yi Yang, and Yaser Sheikh. Supervision-by-Registration: An unsupervised approach to improve the precision of facial landmark detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 360–368, 2018. 2, 3

[6] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010. 5

[7] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 2

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[9] Xiaobai Li, Hu Han, Hao Lu, Xuesong Niu, Zitong Yu, Antitza Dantcheva, Guoying Zhao, and Shiguang Shan. The 1st challenge on remote physiological signal sensing (repss), 2020. 1, 6, 7

[10] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollàr. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017. 2

[11] Xuesong Niu, Hu Han, Abhijit Das, Antitza Dantcheva, Xilin Chen, and Xingyuan Zhao Shiguang Shan. Robust remote heart rate estimation from face utilizing spatial-temporal attention. *IEEE AFGR 2019 Conference paper*, 2019. 1, 2

[12] Xuesong Niu, Hu Han, Shiguang Shan, and Xilin Chen. Synrhythm: Learning a deep heart rate estimator from general to specific. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 3580–3585. IEEE, 2018. 1

[13] Jaromir Przybyło, Eliasz Kańtoch, Mirosław Jabłoński, and Piotr Augustyniak. Distant measurement of plethysmographic signal in various lighting conditions using configurable frame-rate camera. *Metrology and Measurement Systems*, 23(4):579–592, 2016. 1

[14] Ross Girshick Jian Sun Shaoqing Ren, Kaiming He. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015. 2

[15] Rencheng Song, Senle Zhang, Juan Cheng, Chang Li, and Xun Chen. New insights on super-high resolution for video-based heart rate estimation with a semi-blind source separation method. *Computers in Biology and Medicine*, 11 2019. 1

[16] Radim Spetlik, Jan Cech, Vojtěch Franc, and Jiri Matas. Visual heart rate estimation with convolutional neural network. 08 2018. 1