Machine Learning Driven Heart Rate Detection with Camera Photoplethysmography in Time Domain

Viktor Kessler(✉), Markus Kächele, Sascha Meudt, Friedhelm Schwenker, and Günther Palm

Institute of Neural Information Processing, Ulm University, Ulm, Germany
{viktor.kessler,markus.kaechele,sascha.meudt,
friedhelm.schwenker,guenther.palm}@uni-ulm.de

Abstract. Measuring bio signals such as the heart rate in non medical applications is gaining an increasing importance. With camera based photoplethysmography (PPG) it is possible to measure the heart rate remotely with built in webcams of every tablet and laptop. Recent research with machine learning based methods showed great success compared to signal processing based methods. In this paper, we use k-nearest neighbor (kNN) and multilayer perceptron (MLP) with an alternative representation of the input vector. Estimating the quality of peaks with a Gaussian distribution could further improve the detection. Overall we could improve the root mean square error (RMSE) from 23.97 to 8.62.

Keywords: Photoplethysmography (PPG) · remote Photoplethysmography (rPPG) · Camera · Webcam · k-nearest neighbor · Neural network · Gaussian distribution

1 Introduction

Most tablets and laptops are equipped with a front camera and are often used for hours every day. For health care applications this is an interesting means to monitor the health of a person. Several works in the last ten years showed that camera based photoplethysmography (PPG) can be used to remotely measure bio signals such as the heart rate. This allows a long term monitoring system which does not interfere with the user in his/her daily work and at the same time does not need a daily scheduled, explicit measurement. Advances in signal processing based measurement methods for camera based PPG as well as camera technology enables better detection rates but until now not reliable for serious applications.

Machine learning technology methods show great success in replicating systematic occurrences. However, until now only few learning based approaches were presented. One of the early works was presented by Lamonaca et al. [7]. They used a neural network to evaluate the blood pressure from facial videos recorded with a smartphone camera and its flashlight. Hsu et al. [6] used support

F. Schwenker et al. (Eds.): ANNPR 2016, LNAI 9896, pp. 324–334, 2016.
DOI: 10.1007/978-3-319-46182-3_27

vector regression (SVR) in the frequency domain to detect the heart rate. They showed three times better results than a pure signal processing based method. Maaoui et al. [8] used a support vector machine (SVM) and seven features from time and frequency domain with the aim of detecting the stress level.

The remainder of this paper is organized as follows. In Sect. 2 the generation of the signal for the detection is explained. This includes skin extraction, signal filtering and detection of the heart rate. Two machine learning algorithms, k-nearest neighbors (kNN) and multilayer perceptron (MLP), are described in Sect. 3 and analyzed in Sect. 4 on the Open_EmoRec_II dataset [11]. A conclusion follows in Sect. 5.

2 Signal Extraction

2.1 Region of Interest (ROI) Detection

The interesting content of a video for this work is the face of a participant within which we measure the heart rate. We use the Cambridge face tracker [1] implemented by Baltrušaitis et al. [2] to detect the face in each video frame. This detector returns multiple facial landmarks for the eyes, eyebrows, nose, mouth and the lower part of the contour of the face (Fig. 1(a)). Then the area corresponding to the face is estimated by mirroring the contour on an axis of reflection through the eyes. A convex hull over the lower and mirrored contour determines the outermost landmarks (Fig. 1(b)).

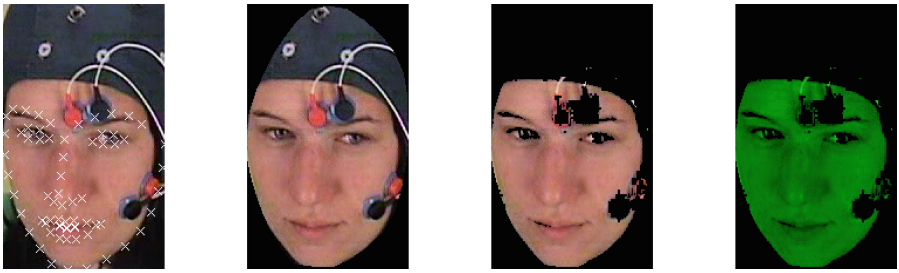
Parts of the face which contains skin (like cheeks and forehead) are more suitable for heart rate estimation in comparison to eyes or mouth. These parts are detected within the convex hull with the skin detection Algorithm of Mahmoud [9]. Therefore the frame is converted from RGB color space into the Y'CbCr color space and all pixels within the value space

$$\begin{aligned} Y' &> 80 \\ 77 \leq C_b &\leq 127 \\ 133 \leq C_r &\leq 173 \end{aligned}$$

will persist in the frame (Fig. 1(c)).

2.2 Signal Preprocessing

Cui et al. [12] showed that the strongest heart rate signal of PPG systems is in the green light wavelength. Therefore we consider only the green color channel in this work (Fig. 1(d)). The raw pulse signal $X^G_{\text{raw}}(t)$ is extracted by computing the average green color value of the skin (the ROI, Fig. 1(d)). The discrete time t is the frame rate of the video. The raw signal $\vec{X}^G_{\text{raw}}$ will be bandpass filtered with a zero-phase digital filter and third order butterworth coefficients [10]. The resulting filtered signal is denoted $\vec{X}^G_{\text{filtered}}$. The desired frequency range is set to $[0.5, 3.0]$ Hz ($[30, 180]$ bpm). The zero-phase filtering removes the bias and the



(a) (b) (c) (d)

Fig. 1. For each frame (a) the face is detected and (b) the ROI is limited to the face. Then (c) the skin is extracted and (d) the mean green color is used for heart rate detection (Sect. 2.2). (Color figure online)

trend from the signal. Complementary information regarding the filtering can also be found in [12] (Sect. 2.3).

The ground truth signal $\vec{X}_{GT}$ is commonly recorded with a electrocardiograph or a blood volume pulse (BVP) sensor. In this work we will use a finger BVP sensor which records the photoplethysmographic signal in the fingertip. The correctness of the ground truth heart rate for our dataset is evaluated.

2.3 Heart Rate Detection

The time-domain based technique detects the peaks (corresponding to the heart beats) in the filtered signal $\vec{X}^{G}_{\text{filtered}}$. This Method will further be denoted as PEAK. For the peak detection we use the algorithm implemented by Carbajal [3]. The detected timestamps of the peaks are defined as $\vec{R}$ and the distances of adjacent timestamps as $\vec{RR}$. The average heart rate f_{HR} is calculated by dividing the frame rate *fps* through the median peak distance $\widetilde{\vec{RR}}$. In case of the average heart rate of a window segment, the peak timestamps $\vec{R}$ are restricted to the window segment:

$$f_{HR} = \frac{fps * 60}{\widetilde{\vec{RR}}_{\text{segment}}}, \quad (1 \text{ bpm} = 1/60 \text{ Hz}) \tag{1}$$

The heart rate f_{HR} is measured in beats per minute (bpm) while the right side of the equation is measured in Herz (Hz). We convert Hz into bpm by multiplying with 60.

Our primary error measurement method is the root mean square error (RMSE). It compares all predicted heart rates $\vec{f}_{HR}$ against the reference heart rates $\vec{f}_{HR,GT}$ from $\vec{X}_{\text{GT}}$:

$$RMSE(\vec{f}_{HR}) = \sqrt{\text{mean}((\vec{f}_{HR} - \vec{f}_{HR,GT})^2)} \tag{2}$$

3 Learning

We evaluate two machine learning algorithms in our work. The first is k-nearest neighbors (kNN) and the second is a multilayer perceptron (MLP). In Sect. 3.1 we discuss the conversion of the data into a form so we can process it.

3.1 Data Preparation

From the filtered signal $\vec{X}^{G}_{\text{filtered}}$, window segments of 3 s length are generated where each segment starts from a detected peak. Each segment relates to a heart rate f_{HR} calculated from the reference signal $\vec{X}_{\text{GT}}$. Therefore, the resulting segments begin at a local maximum. The signal's minimum and maximum should be located at the same positions. Our aim is to learn these similarities.

This segmentation method is extended by placing the peak at the center of the segment and using the segment length of 3 s in both directions. Further extensions consist in placing the peak at the end of the segment (online mode) or in case of the training set, creating the segments from reference BVP signal $\vec{X}^{\text{BVP}}_{\text{GT}}$ (but not from ECG signals).

Each segment is normalized independently for the learning algorithms.

In the following sections we learn the segments with kNN and MLP. As input vector we use all time steps of the 2*3 s of a segment. As target output we use the related heart rate f_{HR}. In Sect. 3.4 we use only the timestamps $\vec{R}$ and $\vec{R}_{\text{GT}}$ of the detected peaks from $\vec{X}^{\text{G}}_{\text{filtered}}$ and $\vec{X}_{\text{GT}}$.

3.2 k-Nearest Neighbor

One of the simplest and successful classifiers for quickly verify a concept is kNN [5]. kNN finds similar segments in the training set for each segment in the test set. The 'k' describes the counts of similar segments (nearest neighbors). For each segment we compute $k = 10$ nearest neighbors with correlation as the distance function. We assume that firstly the first nearest neighbor does not represent the target segment (outliers) so we use 10 neighbors and secondly a lot of the neighbors highly differs from the target segment and thus have a low correlation. To increase the classification quality we remove all neighbors with a correlation below 80 % from each class k. This will remove a huge part of the neighbors. The classification will be repeated for all window segments (in time). Each neighbor (respectively segment) corresponds to a heart rate; the related heart rates of the removed segments are linearly interpolated (independent of other classes). The resulting heart rate of a segment is the mean heart rate of the 10 classes.

3.3 Multilayer Perceptron

Some problems of kNN are that firstly a huge amount of segments is removed and interpolated and secondly we do not have an abstraction of the basic problem. Trials with different machine learning methods showed best results using an

MLP to train from the segments with one hidden layer and 20000 neurons. With increasing count of hidden neurons the performance of the system increases, as well as the execution time. In order to achieve some good balance between the execution time and the performance of the system, the number of neurons in the hidden layer is set to ≈ 20000. The input vector contains all time steps of the 2*3 s. A standard logistic function is used as transfer function. The bias is initialized with '1' and the weights with a Gaussian distribution. In the output layer, a linear neuron targets against the heart rate f_{HR}. The optimization is performed with the Adaptive Subgradient Method (Adagrad) [4]. The learning phase stops after 2000 epochs.

3.4 Gaussian Distribution

The filtered signal $\vec{X}^G_{\text{filtered}}$ has many detected peaks $\vec{R}$ which does not match with the reference peaks $\vec{R}_{\text{GT}}$. Reasons are overshooting of the filter during fast lighting changes (e.g. head motion) or high noise. Therefore we estimate the probability that a detected peak is a real peak. We assume that the heart rate changes slowly and the distance of a peak to its neighboring peaks is nearly constant with $\Delta t_{\text{peak}} = 60/f_{HR}$.

In an ideal case the neighboring peaks have a distance of Δt_{peak}; in case of a misdetection its neighbors should not be multiples of Δt_{peak} (e.g. a peak between two correct peaks like in Fig. 2(a)). For a distance below $0.5\Delta t_{\text{peak}}$ or above $1.5\Delta t_{\text{peak}}$ we assume that this neighbor peak is misdetected (small distance between second and third peak in Fig. 2(a)) or corresponds to another reference peak (big distance between first and fourth peak in Fig. 2(a)). Moreover including the second and third neighbor peaks should improve the estimation (first peak considers second and fourth peak in Fig. 2(a)).

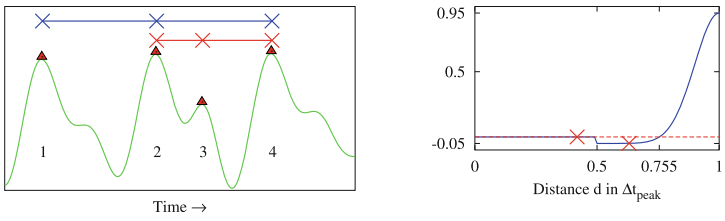


Fig. 2. Left: An example of a misdetected peak. The third peak is wrongly detected, its distances to neighboring peaks (red line) are smaller than Δt_{peak} and this leads to low values in the distribution (-0.05 and -0.049). **Right:** the Gaussian distribution between 0 and 1 with zero for distances below $0.5\Delta t_{\text{peak}}$. The effective values for the third peak are marked with red crosses. (Color figure online)

We define a Gaussian distribution (GD) with the GD center at $d\Delta t_{\text{peak}}$ (Fig. 2(b)):

$$g_d(\vec{r}, \Delta t_{\text{peak}}) = b_d * \exp(-a_d * (\vec{r} - c_d)^2) + m_d \tag{3}$$

$$d\Delta t_{\text{peak}} = \frac{d * 60}{f_{HR}}, \quad d = 1, 2, 3 \tag{4}$$

The probability decreases with the distance of the neighbor peak from the GD center. For every segment we calculate the distances $\vec{r}$ from the center peak to all other peaks $\vec{R}$ and exclude the center peak and all peaks with more than 3 s distance.

We experimentally discovered the following parameters for Eq. 3 (Fig. 3):

$$g_d(\vec{r}, \Delta t_{\text{peak}}) = \frac{1}{d} * \exp(-\frac{50}{\Delta t_{\text{peak}}} * (\vec{r} - d\Delta t_{\text{peak}})^2) - 0.05 \tag{5}$$

The parameter a_d controls the width of the gaussian curve; a consistent value of $50/\Delta t$ for all parameters d showed the best results. The maximum height c_d reduces the influence with increasing d by $1/d$. The center c_d of the gaussian curve is fixed to $d\Delta t_{\text{peak}}$. The parameter m_d controls the generic height and the minimum value; with -0.05 big differences have a negative influence.

In the next step neighbor peaks with a distance of more than $0.5\Delta t_{\text{peak}}$ from the GD center are excluded:

$$(d - 0.5)\Delta t_{\text{peak}} \leq g_d(\vec{r}, \Delta t_{\text{peak}}) < (d + 0.5)\Delta t_{\text{peak}} \tag{6}$$

From this it follows that each neighboring peak corresponds to only one distance d. We normalize each g_d by dividing through the count of corresponding peaks $|\vec{p}_d|$ but minimum 2 because in ideal case each distance d corresponds to two neighboring peaks and having only one is an indication for misdetection. Finally all $g_d(\vec{r}, \Delta t_{\text{peak}})\ \forall\ \vec{r}, d$ are summarized:

$$GD = \sum_{d=1}^{3} \frac{g_d(\vec{r}, \Delta t_{\text{peak}})}{\max(2, |\vec{p}_d|)} \tag{7}$$

The probability is not a percentaged value and has values outside of $[0, 1]$.

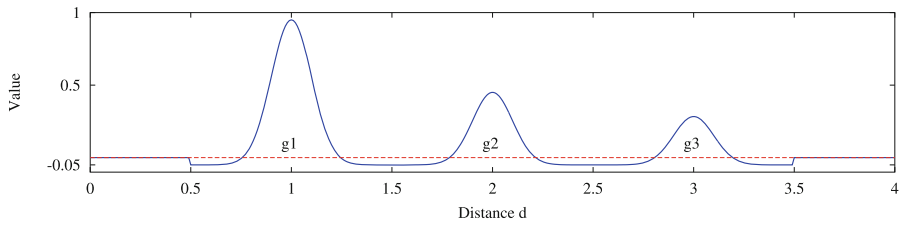


Fig. 3. GD for all distances $d = 1, 2, 3$. Each distribution is truncated to $d \pm 0.5$; every peak corresponds to only one g_d. The minimum value is -0.05 and the maximum height decreases with each d.

4 Experiments

4.1 Dataset

For our evaluation we use the Open_EmoRec_II dataset [11]. It provides 30 participants recorded in a human-computer interaction (HCI) scenario with a session length of 20 min per participant. The participants are sitting still most of the time during the session. The participants were recorded with several Pikes as well as a webcam (720×576@25 resolution, MPEG-4 codec, 1600 kb/s bitrate, manufacturer unknown). The face region has an effective resolution of 240×280 pixels. A finger BVP sensor was recorded for the heart rate signal. The dataset combines two sessions in two different environments. The dataset consist of two parts; the second part (details can be found in [11]) is used for the evaluation. Participants 1, 4, 19 and 28 were excluded because of a bad detection rate of the peaks in the BVP signal.

4.2 Experimental Setting

The reference system for our comparison (denoted as PEAK) extracts the signal $\vec{X}^G_{\text{filtered}}$ from the video and detects the peaks as explained in Sect. 2.3. Then window segments with seven seconds length and a shift of one second are defined from which the heart rate f_{HR} is determined (Eq. 1). With the shift we get a constant response time of one second. For all 26 participants 30655 segments are generated. We get a baseline RMSE value of 23.97.

The evaluation of our methods is performed with leave-one-subject-out (LOSO). The segments are generated as explained in Sect. 3.1. With heart rates higher than 60 bpm the shift of the segments is smaller than one second and a total of 44406 segments for all 26 participants are generated.

The distribution of the heart rates (Fig. 4) is between $[47, 124]$ bpm and shows a high occurrence between $[62, 92]$ bpm. Test samples outside of this area tend to produce higher error rates.

The GD can be applied on the train set as well as on the test set. Applying on the train set removes possibly less qualitative segments from learning phase. Applying on the test set removes possibly misdetected heart rates from the

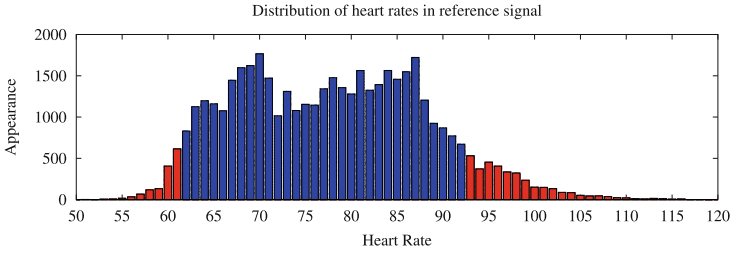


Fig. 4. Histogram of all heart rates $\vec{f}_{HR,GT}$. Heart rates between $[62, 92]$ bpm appear more than half as often as the most often heart rate.

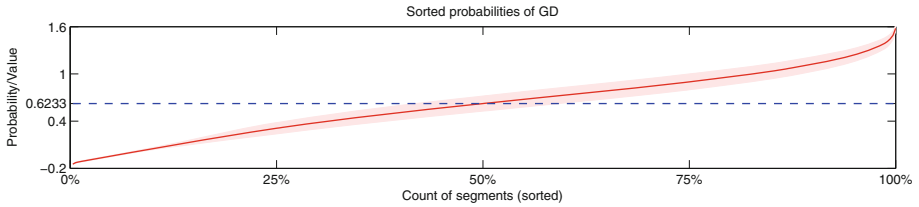


Fig. 5. The GD is applied on the test set. The probabilities are sorted. The red line is the mean of the probabilities computed from all participants and the light red area is the standard deviation. The probability has a value between $[-0.2, 1.6]$. (Color figure online)

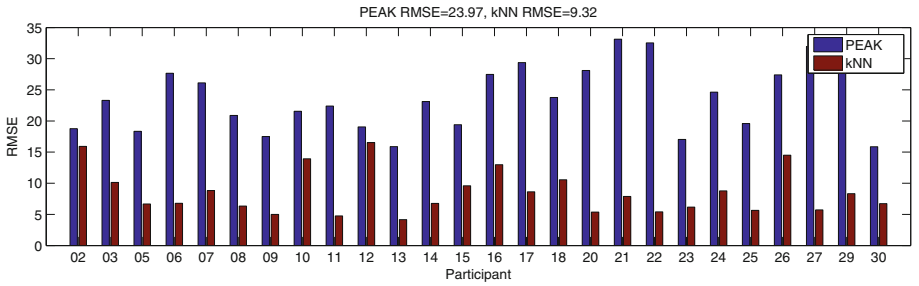


Fig. 6. Comparison between PEAK and kNN shows better results of kNN for all participants.

prediction. This will directly influence the result. For a more accurate comparison the removed heart rates are linearly interpolated. A threshold analysis is shown in Fig. 5. The behavior of the probabilities shows for a threshold of 0.6233 a standard deviation of approximately 20 % (blue line cutting the red area). Based on these findings we decided to use 50 % of the segments instead of a threshold.

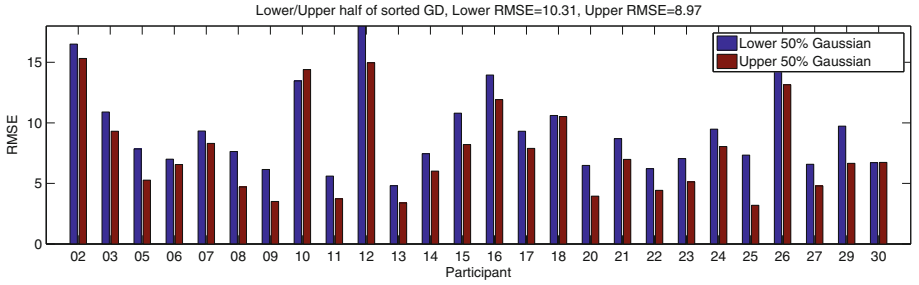


Fig. 7. Evaluation of GD applied on the test set. The heart rates for all participants are detected with kNN and sorted by the probability of GD. The RMSE is computed on lower (inferior) and upper (superior) half of sorted prediction. Restricting to the upper half of GD improves the detection from 10.31 to 8.97.

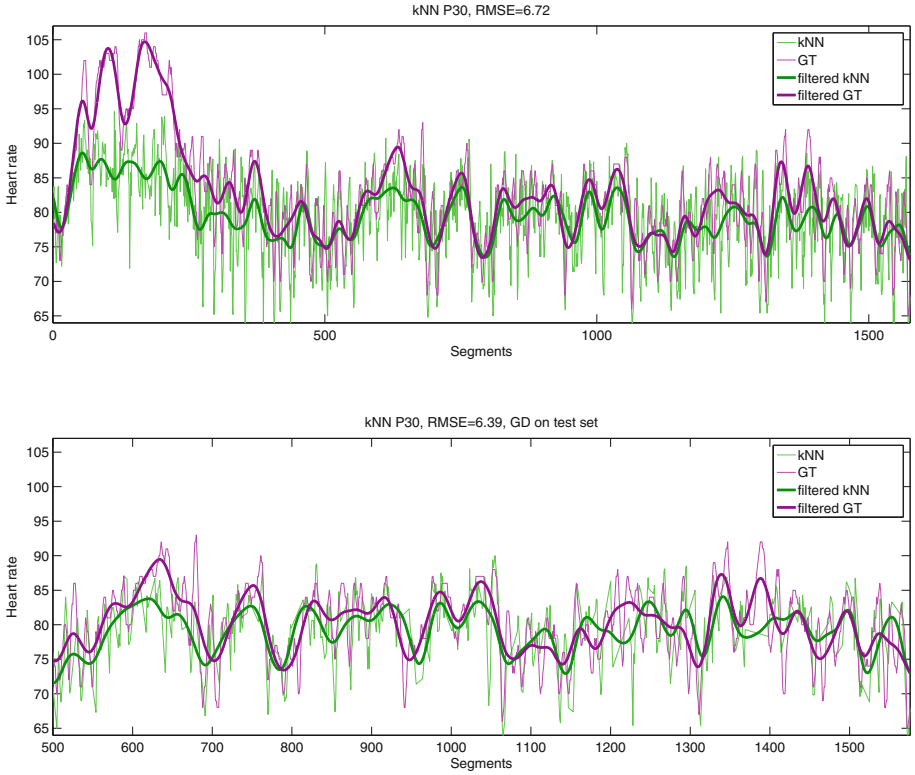


Fig. 8. Prediction (thin green line) with kNN for the participant 30. After removing and interpolating the lower half of GD from the test set (bottom figure, zoomed to $[500, 1580]$), the prediction is less noisy (thin green line). The filtered representations (thick lines) have nearly the same behavior. 30 segments are interpolated around segment 1400. This has a detection delay as consequence. (Color figure online)

4.3 Results

The comparison between the baseline method PEAK and kNN (Fig. 6) shows that kNN outperforms PEAK for all participants. The overall RMSE values are 23.97 for PEAK and 9.32 for kNN.

In Fig. 7 we detected the heart rates with kNN for all participants. On the test set we applied GD (Fig. 5) and sorted the detected heart rates by the probabilities of GD. We split the sorted prediction into a lower (inferior) and upper (superior) half. The two halves have an RMSE of 10.31 (lower) and 8.97 (upper).

Figure 8 compares kNN with and without GD for the participant 30. The signal duration is 18 min. For the bottom figure the heart rates of the lower half are removed from the prediction and interpolated. The error decreases from 6.7 to 6.4. Long detection gaps do not appear but shorter gaps could be problematic for live systems. For segments between $[30, 250]$ kNN has problems following heart rates higher than 90 bpm.

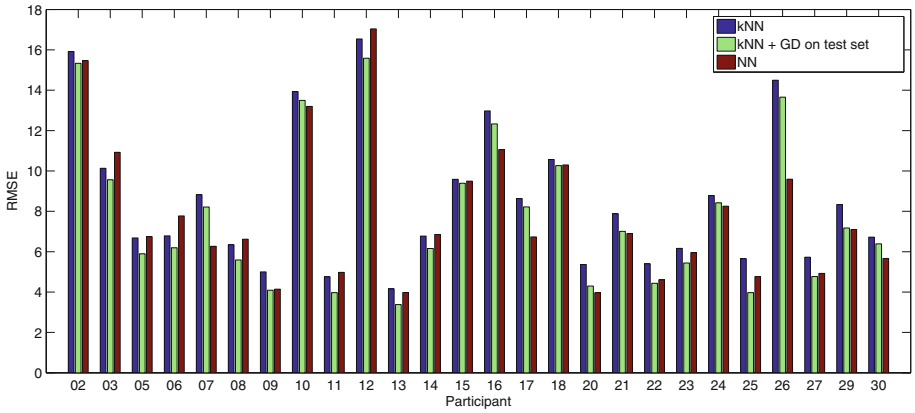


Fig. 9. Per participant comparison shows benefits of GD when applied on the test set. The neural network is better than kNN in most cases

Comparison of kNN, kNN with GD on the test set and MLP (Fig. 9) shows better results for GD over pure kNN for all participants. MLP is better than kNN in most cases.

Finally Table 1 summarizes the results of this work.

Table 1. Results of the methods.

	PEAK	kNN	kNN + GD	MLP
RMSE	23.97	9.32	8.97	8.62

4.4 Discussion

Comparing the PEAK method with kNN in Fig. 6 shows the predominance of machine learning approaches for this task. The RMSE decreases from 23.97 to 9.32. Applying GD on the test set further decreases the RMSE to 8.97. Trials to apply GD on the train set degrades the RMSE both with kNN and MLP.

One big drawback of kNN and especially of GD applied on the test set is that a huge amount of segments is removed. For live systems this leads to detection delays. Therefore post processing methods needs to be designed (e.g. giving a coarse prediction and correcting it to a later time). MLP don't remove segments and with an RMSE of 8.62 it is better than kNN. Applying GD in combination with MLP should improve the detection as well but we aim for a method without interpolating segments.

5 Conclusion

In this work we have shown that learning based methods can extremely improve the heart rate detection with camera based PPG. Hsu et al. [6] came to the same result about the benefit of learning based methods. Further experiments shown an improvement with neural networks and GD. One disadvantage is the need of a well distributed dataset over the whole frequency range of the heart rate.

Acknowledgements. This work is supported by the Transregional Collaborative Research Centre SFB/TRR 62 *Companion-Technology for Cognitive Technical Systems* funded by the German Research Foundation (DFG) and by the BMBF *SenseEmotion*. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research. The authors wish to thank Mohammadreza Amirian for fruitful discussion and the Emotion Lab of Ulm University for providing the Open_EmoRec_II dataset.

References

1. Baltrusaitis, T., Robinson, P., Morency, L.P.: Constrained local neural fields for robust facial landmark detection in the wild. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 354–361 (2013)
2. Baltrušaitis, T.: CLM-Framework. https://github.com/TadasBaltrusaitis/CLM-framework
3. Carbajal, J.P.: Findpeaks. http://sourceforge.net/p/octave/signal/ci/default/tree/inst/findpeaks.m
4. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. **12**, 2121–2159 (2011)
5. Friedman, J.H., Bentley, J.L., Finkel, R.A.: An algorithm for finding best matches in logarithmic expected time. ACM Trans. Math. Softw. (TOMS) **3**(3), 209–226 (1977)
6. Hsu, Y., Lin, Y.L., Hsu, W.: Learning-based heart rate detection from remote photoplethysmography features. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4433–4437, May 2014
7. Lamonaca, F., Barbe, K., Kurylyak, Y., Grimaldi, D., Moer,W.V., Furfaro, A., Spagnuolo, V.:Application of the artificial neural network for blood pressure evaluation with smartphones. In: 2013 IEEE7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), vol. 01, pp. 408–412, September 2013
8. Maaoui, C., Bousefsaf, F.: Automatic human stress detection based on webcam photoplethysmographic signals. J. Mech. Med. Biol. **16**(04), 1650039 (2015)
9. Mahmoud, T.M.: A new fast skin color detection technique. World Acad. Sci. Eng. Technol. **43**, 501–505 (2008)
10. Oppenheim, A.V., Schafer, R.W., Buck, J.R.: Discrete-Time Signal Processing, 2nd edn. Prentice-Hall Inc., Upper Saddle River (1999)
11. Rukavina, S., Gruss, S., Walter, S., Hoffmann, H., Traue, H.C.: OPEN_EmoRec_II-a multimodal corpus of human-computer interaction. World Acad. Sci. Eng. Technol. Int. J. Comput. Electr. Autom. Control Inf. Eng. **9**(5), 1135–1141 (2015)
12. Verkruysse, W., Svaasand, L.O., Nelson, J.S.: Remote plethysmographic imaging using ambient light. Opt. Express **16**(26), 21434–21445 (2008)