Real-Time Contactless Heart Rate Estimation from Facial Video

by

Ying Qiu

Thesis submitted to the University of Ottawa
in partial Fulfillment of the requirements for the
M.A.Sc degree in
Electrical and Computer Engineering

uOttawa

School of Electrical Engineering and Computer Science
Faculty of Engineering, University of Ottawa
Ottawa, Canada

Abstract

With the increase in health consciousness, noninvasive body monitoring has aroused interest among researchers. As one of the most important pieces of physiological information, researchers have remotely estimated heart rates from facial videos in recent years. Although progress has been made over the past few years, there are still some limitations, such as the increase in processing time with accuracy and the lack of comprehensive and challenging datasets for use and comparison. Recently, it was shown that heart rate information can be extracted from facial videos by spatial decomposition and temporal filtering. Inspired by this, a new framework is introduced in this thesis for remotely estimating the heart rate under realistic conditions by combining spatial and temporal filtering and a convolutional neural network. Our proposed approach exhibits better performance compared with that of the benchmark on the MMSE-HR dataset in terms of both the average heart rate estimation and short-term heart rate estimation. High consistency in short-term heart rate estimation is observed between our method and the ground truth.

Acknowledgements

First, I would like to express my sincerest gratitude to my supervisor, Prof. Abdulmotaleb El Saddik, for his encouragement, support, precious help and advises for my whole graduate studies. His patience and perseverance on knowledge have given me great strength to finish my master research. I am so honored and proud to be his student and work with every member in MCR Lab. Not only the knowledge, but also his attitude towards life has brought a lot of positive influence on my study and further life. I have got a huge amount of benefits from what he taught me and I believe everyone working with him can always obtain something beyond knowledge.

I would also like to thank my parents who always unconditionally support all my decisions. They give me everything I need to let me study without any annoyance and be the backup of my life forever. I feel so lucky to be their daughter and receive their love.

Furthermore, I appreciate the experience working in MCR Lab and valuable advises from my friends Yang, Yuxiang and Haiwei, who help my with implementation and editing.

Last, but not least, I would like to thank all the members in MCR Lab, for their cooperation, understanding and being great friends.

Glossary of terms

- HR: Heart Rate
- PPG: Photoplethysmography
- ROI: Region of interest
- EVM: Eulerian video magnification
- CNN: Convolutional neural network
- ECG: Electrocardiography
- BVP: Blood volume pulse
- ICA: Independent component analysis
- FFT: Fast Fourier Transform
- PCA: Principal component analysis
- LF: Low frequency
- MF: Medium frequency
- HF: High frequency
- RGB: Red, green, blue
- OpenCV: Open computer vision
- bpm: beat per minute
- fps: frame per second

- DWCNN: Depthwise convolutional neural network
- LBCNN: Linear bottleneck convolutional neural network
- He: Heart rate error
- Me: Mean error
- SD: Standard deviation
- RMSE: Root mean squared error
- M_{eRate}: Mean absolute percentage error
- ρ: Pearson's correlation

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation behind this work

The vision of digital twin as introduced by Prof. El Saddik (IEEE MultiMedia, Volume: 25, Issue: 2, Apr.-Jun. 2018) is a digital replication of a living or non-living physical entity. By bridging the physical and the virtual worlds, data are transmitted seamlessly, allowing the virtual entity to exist simultaneously with the physical entity. A digital twin facilitates the means to monitor, understand, and optimize the functions of the physical entity and provides continuous feedback to improve quality of life and

wellbeing. A digital twin is hence the convergence of several technologies such as AI, AR/VR and Haptics, IoT, Cybersecurity and Communication networks. A component of digital twin is human heart rate (HR), which reflects both physiological and psychological conditions. HR monitoring has benefits for human beings in many aspects, such as health care for the elderly, vital sign monitoring of newborns and lie detection for criminals. Currently, with the growing health consciousness, mobile HR monitoring systems are becoming increasingly popular with users. However, all such devices must be in contact with the skin, which is inconvenient and may cause discomfort. Hence, researchers have been working to create a low-cost and noninvasive way to measure HR in recent years. The main concepts are derived from the principles of photoplethysmography (PPG), which can sense the cardiovascular blood volume pulse via variations in transmitted or reflected light [5]. Furthermore, Verkruysse *et al.* show that a PPG signal can be measured using a standard digital camera, with ambient light as the illumination source [56]. Since then, researchers have made sustainable progress in digital-camera-based remote HR estimation.

However, various challenges remain. Because the face is the least occluded skin region of the human body, extracting a PPG signal from the facial region has become the main approach [33]. The motion artifacts of a subject affect the measurement performance and are divided into two types: One is rigid motion, which includes head tilt and posture changes. The other is nonrigid motion, which includes facial expressions such as eye blinking and smiling. Illumination variations, such as flashes of indoor light, the variation of reflected light from a computer screen and the internal noise of a digital camera, also add noise to the PPG signal [57]. In addition, other challenges affect this technology, such as signal strength and a lack of appropriate datasets.

The approaches proposed by researchers in recent years can be divided into two groups. The methods in the first group select the whole face as the region of interest. Then, most of the methods use a filter to reduce the noise caused by motion artifacts

and illumination variations. The methods in the other group focus on choosing a region of interest (ROI) of the face and estimating the HR based on this reliable region. To some extent, both methods have aspects of superiority, but drawbacks do exist.

Whole-face-region-based approaches usually extract a PPG signal by spatially averaging the face region at the outset and then mainly focusing on reducing the impact of interference caused by noise. These approaches, e.g., [57] and [38], perform better under nonrigid motion conditions than they do under rigid motion conditions because computing the mean of the whole face nullifies the effect of facial muscle movements. However, the whole face region always includes noise caused by multiple environmental factors. Although various filters are applied to eliminate the noise, the processing time increases with the number of filters used. Thus, it is difficult to estimate the HR instantaneously by using this approach.

Partial face region selection can help decrease the time spent filtering a signal. Reliable region selection is a challenge in these approaches since it involves too much indeterminacy, particularly when a subject moves with facial expressions during the experiment. Progress has been made, such as in [54] and [2], with better performance than that of the whole-face-region-based approach being achieved.

The dataset is one of the most important factors affecting the performance of deep learning. In addition, a dataset can be used to make a fair comparison of different approaches. Although the MAHNOB-HCI dataset [36] is widely used in HR estimation research, there are several limitations that must be considered. The experiment conditions are strictly controlled, and the subjects hardly move and do not show notable facial expressions.

Regardless of which of the above methods are considered, there is a common point among them: they all use the power spectral density of the underlying signal to estimate the HR in the last step of the procedure, which requires a clear PPG signal to obtain an accurate result. Many processing steps must be used to obtain this clear signal, which

increases the computing time. To acquire better performance in less time, additional knowledge must be considered. Inspired by deep learning, HR estimation has been one of the hottest topics in recent years and has attracted researchers from a variety of fields, who have used it to carry out various tasks; see [29], [58], and [55]. HR estimation is considered a regression task in this thesis. Because subtle changes in skin color related to the cardiac rhythm can be extracted from a digital camera, the average HR within a short time interval can be defined as a label in the regression task, and the extracted color changes of the corresponding video sequence can be fed into a neural network. In the research of Wu *et al.*, the blood flow through the face due to the cardiac cycle is amplified as the skin color changes, and the vertical scan from the amplified video sequence reveals a plausible human heart rate [23]. Inspired by this, a new paradigm is proposed for estimating the heart rate, namely, using Eulerian video magnification (EVM) to extract face color changes and using deep learning to estimate the heart rate.

1.2 Thesis statement

To estimate the HR instantaneously under realistic conditions, a combination of feature extraction and HR estimation is considered in our work. To automatically detect a subject's face region, a fast and robust face detection is applied. To reduce the effect of rigid motion, tracking is used to achieve stable face region sequences. As introduced before, power spectrum density analysis is a traditional way to estimate the HR from a PPG signal; however, obtaining a good result via this approach requires a clear PPG signal without noise, which requires a more complex procedure and more time to process the signal.

Considering the above, a new paradigm is attempted in this thesis. A convolutional neural network is applied to estimate the HR from feature images. Spatial decomposition and temporal filtering are utilized to extract blood flow due to the cardiac cycle

contained in the feature image. For the purpose of achieving generalizability under realistic conditions, we apply our method to the MMSE-HR dataset [65], which is more challenging, with a larger multimodal spontaneous emotion corpus. Comparison experiments are conducted between our approach and the state-of-the-art methods, with better performance being presented by our approach. Additionally, the proposed approach can estimate the HR instantaneously, which also reflects better performance compared to that of up-to-date methods.

1.3 Practical applications

The proposed method can be applied in many life conditions. For instance, humans' HR can changes abruptly when they tell lies, this phenomenon can be used in custom office at airports to screen the people who are trying to enter the countries. Since the non-contact method can detect the subject without cautiousness, true reactions can be captured and analyzed for security purpose. Such applications can also be used for criminals and psychological analysis. Also for medical purpose, a non-contact approach is much cheaper than contact approaches, since the sensor is expensive and may cause discomfort when it is attached to skin. For general situations, HR estimation can also help analyze humans' emotions, elders' health conditions and newborns' vital signs.

1.4 Objective

In our work, EVM and deep learning are integrated together to achieve instantaneous HR estimation. Specifically, EVM is used to extract the feature image containing the heart rate information within a specified time interval. A convolutional neural network (CNN) is then applied to estimate the HR from the feature image, which is formulated as a regression problem. In summary, the contributions of this thesis are as follows:

1. A new paradigm is proposed for HR estimation using a digital camera under realis-

tic conditions. As a part of EVM, spatial decomposition and temporal filtering are applied to extract face color changes as a feature image. A wider bandpass filter is used here to to extract signals within a typical human HR range.

2. HR estimation is achieved by using a CNN, where the input is the feature image within a certain time interval and the output is the average HR during that time interval. Our approach tackles the problem of computational complexity and high time cost.

3. To demonstrate the performance of our approach, a comparison between the proposed approach and a benchmark [54] on the same dataset for average HR estimation and short-term HR estimation is conducted. Our result indicates the higher accuracy of our approach compared to that of other methods.

1.5 Scholarly Output

As a result of this work, a journal called "EVM-CNN: Real-Time Contactless Heart Rate Estimation from Facial Video" has been accepted by IEEE Transactions on Multimedia.

1.6 Thesis outline

The rest of the thesis is organized as follows.

- In chapter 2, a review of the existing approaches on remote HR estimation is presented.
- In chapter 3, the details of the proposed method are given.
- In chapter 4, the dataset used, evaluation metrics and experimental settings are discussed.

- In chapter 5, our results and analysis are presented.
- In chapter 6, the conclusions of our work and directions for future work are given.

Chapter 2

Related work

Since the main purpose of our work is to estimate the HR from a facial video, the following part focuses on noncontact HR estimation. Although traditional methods are non-real-time, their efforts in terms of HR feature extraction have profound effects on the subsequent progress. Several real-time methods that were proposed in recent years are also introduced. As an important module in our work, the convolutional neural network is described in detail to help readers understand the rest of our work.

2.1 Noncontact HR estimation

Traditional measurement methods for cardiac activity are contact-based methods, including extracting an electrocardiography (ECG) signal by using an ECG sensor and the PPG signal by using a pulse oximeter. Although contact-based methods can be used to measure cardiac activity comprehensively with high accuracy, sensors attached to human bodies are always uncomfortable and inconvenient. For instance, in lie detection, the subject being measured is not expected to know the purpose of that measurement; however, it is difficult to conceal that purpose when the sensor is being attached. Toward creating a noninvasive measurement method, progress has been made by researchers over the past few years.

PPG signal

Noncontact HR estimation is based on extracting the PPG signal via low-cost semiconductor technology with LED and matched photodetector devices working at the red and/or near-infrared wavelengths [5]. This method of measuring the cardiovascular blood volume pulse (BVP) within the body by recording changes in reflected light is inexpensive. During the cardiac cycle, the blood vessels volume changes, which causes variations in the light absorption under the skin. Therefore, a PPG signal is used to analyze physiological information such as the heart rate, respiration, depth of anesthesia, hypervolemia and other circulatory conditions.

In 2009, Verkruysse *et al.* showed that a PPG signal can be measured remotely using ambient light and a simple consumer-level digital camera in movie mode [56]. It was also shown that the green channel features the strongest PPG signal and that the red and blue channels also contain PPG information, as the hemoglobin absorbs green light better than red and penetrates sufficiently deeper into the skin compared to blue

light to probe the vasculature.

2.2 Non-real-time-based HR estimation

Traditional noncontact HR estimation has been achieved by extracting the PPG signal of a whole facial video. The power spectrum density is then applied to find the highest power spectrum within the general human HR frequency band as the average HR of the video. Different approaches to feature extraction involving the PPG signal have been proposed and compared over the past few years, as illustrated in the following part.

ICA-based method

In 2010, Poh *et al.* introduced a noncontact cardiac pulse measurement via color images by using blind source separation [37], which is the first breakthrough in noncontact HR estimation. In their study, a face video was recorded by a webcam, and a mixture of the reflected PPG signal was picked up by the RGB sensors, along with other sources of fluctuations in light due to artifacts such as motion and changes in ambient lighting conditions. They used a boosted cascade classifier to detect the face region, which is based on the work of Viola and Jones [44] as well as that of Lienhart and Maydt [45]. The face detection module was implemented by using OpenCV, which returned the coordinates along with the height and width that define a box around the face. The ROI was defined by selecting the center 60% width and full height of the box.

Poh [37] first computed the spatial average value of a face region for each channel, where three temporal traces were obtained by an RGB video sequence. Independent component analysis (ICA) was applied to separate the observed raw signals and find the underlying PPG signal. Finally, they applied the fast Fourier transform (FFT) to the selected source signal to find the highest power spectrum within the general human HR

frequency band as the average HR frequency of the video recording. However, the signal quality was deteriorated due to motion artifacts, leading to a noisy PPG signal.

In 2011, Poh *et al.* improved their work by adding several temporal filters to clean up the PPG signal [38]. The separated source signal is smoothed by using a five-point moving average filter and bandpass filter. The filtering process reduces the effects of motion artifacts, which leads to higher accuracy.

EVM-based method

In 2012, Wu *et al.* introduced Eulerian video magnification (EVM), which can amplify the blood flow under a human face in a video recording [23]. The magnification reveals the redness variation as blood flows through the face via a cardiac cycle. For this application, temporal filtering needs to be applied to lower spatial frequencies to allow this subtle input signal to rise above the camera sensor and quantization noise.

Each pixel of the face region is processed by spatial decomposition and temporal filtering; then, the resultant frames are multiplied by a magnification factor and upsampled to the original size. Finally, they are added to the original frames. Therefore, the subtle color changes due to the cardiac cycle are visible to the naked eye. Although this method can be used for both color amplification and motion magnification, the result of face color amplification is used to estimate the HR.

PCA-based method

In 2013, Balakrishnan *et al.* showed that the HR can be extracted from videos by measuring subtle head motion caused by Newtonian reactions to the influx of blood at each beat [16]. They first find an ROI containing the head and track feature points within the region. The vertical projection of each point over the whole video is treated as a trajectory. Temporal filtering is then applied to select trajectories whose pulse rates

fall within the human HR frequency band.

They used principal component analysis (PCA) to decompose the selected trajectories and chose the one whose temporal power spectrum best matches the pulse. Then, they calculated the average HR from the one chosen. Apparently, this motion-based approach is better when the whole face is not visible, for example, the subject is sitting with his/her side facing the camera. However, all the subjects measured in their experiments were required to be stationary and sit upright during the video recording.

Illumination rectification and nonrigid motion elimination

After several methods that could successfully estimate the HR remotely were introduced, researchers began to focus on this topic under challenging conditions. In 2014, Li et al. proposed a framework for estimating the HR under realistic conditions [57]. They first used the discriminative response map fitting (DRMF) [1] method to detect facial landmarks and generate a mask of the ROI in the first frame. They then employed the Kanade-Lucas-Tomasi (KLT) [8] algorithm to track the location of the ROI, which can reduce the effect caused by rigid motion. Different from Poh [38], the average value of the green channel was chosen as the raw pulse signal. To reduce the interferences of illumination variations, the average green value was computed as a reference to model the illumination variations in the ROI, which was achieved by segmenting the background region using the distance regularized level set evolution (DRLSE) [6] method. To find the optimized coefficient of the model, a normalized least mean squares (NLMS) [21] filter was applied.

Furthermore, to reduce the interferences caused by sudden nonrigid motions, the pulse signal was divided into segments, and contaminated segments were discarded. Finally, several temporal filters were applied to exclude powers of frequencies outside the

HR range. They also evaluated their performance by comparing with that of previous methods on the MAHNOB-HCI dataset [36], which showed that their method achieved higher accuracy compared to other methods.

Although their work shows better performance than that of previous methods, several drawbacks still exist. In the illumination rectification part, they used the background illumination as a reference to rectify the motion artifacts caused by the illumination variation of the face. This assumption is not always true since the background illumination may not be the same as the illumination on the face. In the nonrigid motion elimination part, they determined the segments with motion artifacts by estimating the highest standard deviation, which could eliminate nonartifact segments. In addition, the overall process takes a long time to obtain the final result since it involves several procedures and each procedure applies different algorithms, which makes it impossible to estimate the HR in real time.

Random patches and good location selection

In 2015, Lam *et al.* improved the blind source separation by randomly choosing a pair of patches from a face to obtain multiple extracted PPG signals; they then use a majority voting scheme to robustly recover the HR [2].

They randomly select many pairs of patches in the first frame; the average green traces are obtained by using the pose-free facial landmark fitting tracker [59]. These traces are then tested on several conditions, and only the correct pairs of points that satisfy all the conditions are put into subsequent procedures, as they can be solved as a linear blind source separation problem. The selected signals are then converted to the frequency domain, and the power spectrum density is obtained. The confidence ratio is computed for each signal by computing the ratio of the amplitude at the highest peak with respect to the amplitude at the second highest peak to find a confident estimation of

the HR. For each extracted PPG signal with a high confidence ratio, the HR is computed and added to a histogram. A majority vote is then performed for the histogram to obtain the final HR estimation.

Compared with [38] and [57], this method yields better results on the same dataset. However, a dominant shortcoming is its speed: the reported MATLAB implementation takes 7 minutes to process 30 seconds of video recording, not including tracking time.

Depth-video-based method

Different from previous methods based on RGB videos, Yang *et al.* proposed a nonintrusive heart rate estimation system via 3D motion tracking in a depth video [60]. They use Kinect 2.0 to estimate a subject's HR. As blood is pumped from the heart to circulate through the head, tiny oscillatory head motions due to Newtonian mechanics can be detected for periodicity analysis, which is similar to Balakrishnan's [16] principle.

Three Kinect 2.0 cameras are used to capture the subject from the front, side and back. They first restore the depth images via a bit-depth enhancement/denoising procedure. Denoising is necessary since it is known that depth sensors are susceptible to acquisition noise [15]. Bit-depth enhancement is necessary because the granularity of a depth pixel captured by the Kinect sensor is not sufficiently fine-grained to capture subtle head motion due to heart beats without processing [12] [16]. Head region tracking is then applied, and the returned vectors are fed back to the last module in a loop to guarantee the consistency between the two modules. The obtained 3D motion vectors are projected along the principal component via principal component analysis. The resultant 1D signal is then processed by nonlinear trend removal to reduce the nonstationary trends of the signal [64] [38]. Similar to previous methods, a bandpass filter is then applied to remove the motion of frequencies from the band of interest. An extra procedure is applied in their work, namely, using wavelet-based motion denoising to reduce noise in

the motion signal via wavelet-domain soft-thresholding [51]. Finally, the HR is estimated by analyzing the power spectrum density.

Their experiment results show accurate HR estimation compared to the values measured by a portable finger pulse oximeter. However, the subjects being measured are required to remain still, and three cameras are used, which indicates the limitations of using their method. In addition, their system does not perform well when the subject is leaning his/her head on a chair or asleep in bed, where the subtle motion is too small to detect.

ICA vs EVM

In 2017, Alghoul *et al.* compared two methods [26], one based on independent component analysis (ICA) and the other based on Eulerian video magnification (EVM). They improved both approaches by modifying the ROI selection and adding filters to clear the PPG signal. They applied these two approaches to estimate not only the HR but also the HRV, which is related to emotion arousal. They finally concluded that the ICA-based method yielded better results in general, while the EVM-based method might be more appropriate when motion is involved.

2.3 Real-time-based HR estimation

Although the accuracy has been improved and the system has become more robust, the processing time is still a drawback. Since researchers add procedures to reduce the effect of motion artifacts and obtain a more reliable PPG signal, the system takes longer to yield the final result. All the methods reviewed above cannot be used to estimate instantaneous HR. However, instantaneous HR is also important for analyzing human emotions as well as other aspects. Therefore, some researchers have begun to find a better way to estimate the HR in real time.

Chrominance-based method

In 2013, De Haan *et al.* presented an analysis of the limitations of blind source separation on motion problems and proposed a chrominance-based approach with better performance [17]. Their method not only improves the estimation accuracy with motion included but also can be used for short video recordings.

They determined that whether a subject performs exercise should be considered when evaluating the work performance, typically a periodic motion inside the pulse rate frequency band. Consequently, the blind source separation method cannot reliably detect which component carries the pulse signal in their experiments, while the chrominance-based method can improve the accuracy under exercise conditions. In addition, the chrominance-based method exhibits a clearly superior performance for a short overlap-add interval, while the blind source separation method performs better on intervals, which is a drawback when the instantaneous HR needs to be estimated.

Self-adaptive matrix-completion-based method

Another breakthrough in terms of instantaneous HR estimation came from Tulyakov *et al.* [54]. They used a self-adaptive matrix completion approach to automatically discard the face region corresponding to noisy features and proposed using only the reliable one to estimate the HR. They also introduced a more challenging dataset [65] with target movements and spontaneous facial expressions.

First, 66 facial landmarks are localized and tracked. The ROI is then defined and warped to a rectangle by using a piecewise linear warping procedure. The rectangle is then divided into a grid containing many regions. For each subregion, the chrominance feature is computed by averaging the values of the chrominance signals over all the pixels. Self-adaptive matrix completion is then applied to estimate the underlying low-

rank feature matrix and the mask to determine a reliable region for further analysis. The final HR frequency is calculated by using the power spectrum density.

They compared their methods with previous approaches on the MAHNOB-HCI dataset [36], and the results show that their method performs better than the others. In addition, they also performed comparison experiments on the new MMSE-HR dataset [65], as the MAHNOB-HCI dataset was collected under limited conditions, namely, the subjects were required to sit still without either large movements or many spontaneous facial expressions and had to wear an invasive EEG measuring device on their head. Experiments on short-term HR estimation with different time intervals were also conducted and compared to De Haan's work [17]. The higher accuracy achieved further confirmed their work for both non-real-time HR estimation and real-time HR estimation.

As introduced above, it is apparent that most researchers focus on solving the problem with movements when estimating the HR, while few researchers focus on short-term HR estimation. Inspired by this, a new framework is introduced here that combines EVM and deep learning to estimate the HR instantaneously. Instead of applying the magnification process of EVM, spatial decomposition and temporal filtering are mainly used to extract the feature image. To achieve generalizability, a wider bandpass filter is used in our approach. As an important part of our work, a convolutional neural network is used to estimate the HR from the feature image. To better understand the convolutional network, some basic knowledge is introduced in the following section.

2.4 Convolutional neural network

An artificial neural network is an information processing paradigm inspired by biological nervous systems. It consists of several highly connected processing neurons working together to solve specific problems. This concept has been mentioned since the middle of the last century, experiencing a surge at the end of the last century, and has

propagated rapidly in recent years.

The basic structure of a feedforward neural network is as shown in Fig. 2.1. It is composed of three basic layers: input layer, hidden layer and output layer. As shown in Fig. 2.1, the input layer is the subject that needs to be processed, which is usually in the form of a multidimensional vector. Input layers are then distributed to hidden layers. Decisions are made by hidden layers, such as how to weigh up a stochastic transform of the previous layers. The final output is then obtained from the hidden layers. Deep learning is derived from this structure, with multiple hidden layers stacked upon each other.

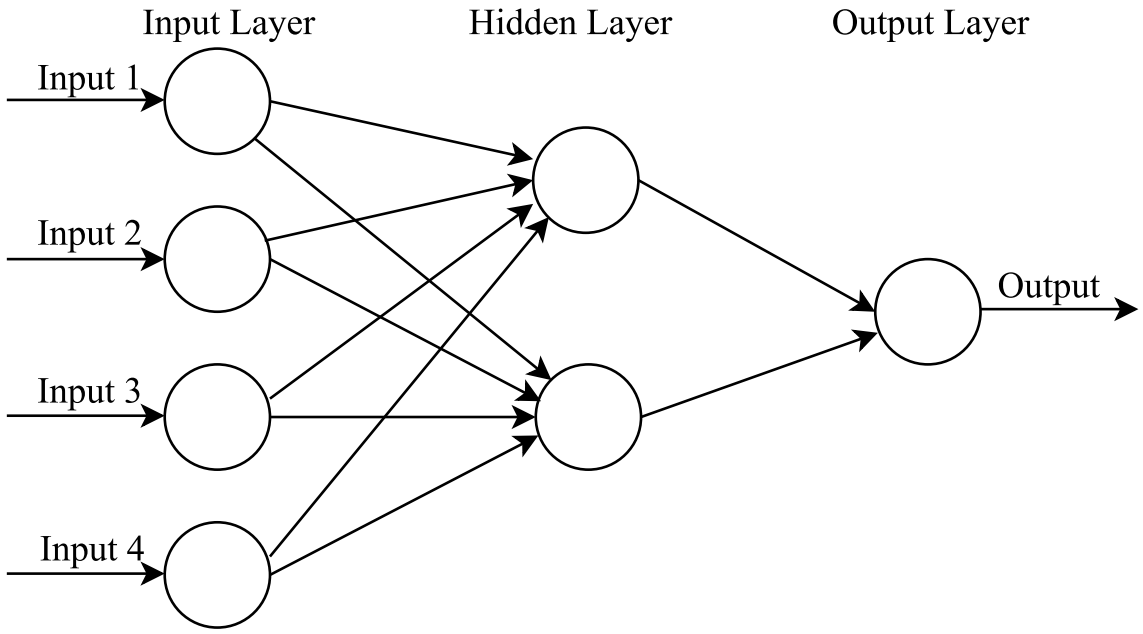


Figure 2.1: The structure of a three-layer feedforward neural network, comprised of an input layer, a hidden layer and an output layer.

In 1989, LeCun's research showed that single-layer networks exhibit poor generalization performance, while multilayer constrained networks perform very well in terms of the handwritten digit recognition problem when organized in a hierarchical structure with shift-invariant feature detectors [62]. In 1990, LeCun updated his research and

stated that a large back propagation network can be applied to real image recognition problems without a large, complex preprocessing stage [61].

Convolutional neural networks are similar to traditional neural networks, which are comprised of neurons that self-optimize via learning. They combine three architectural ideas to ensure some degree of shift and distortion invariance: local receptive fields, shared weights and, sometimes, spatial or temporal subsampling [63]. Different from artificial neural networks, convolutional neural networks are mainly used in the field of pattern recognition within images. A typical convolutional neural network for recognizing characteristics is shown in Fig. 2.2.

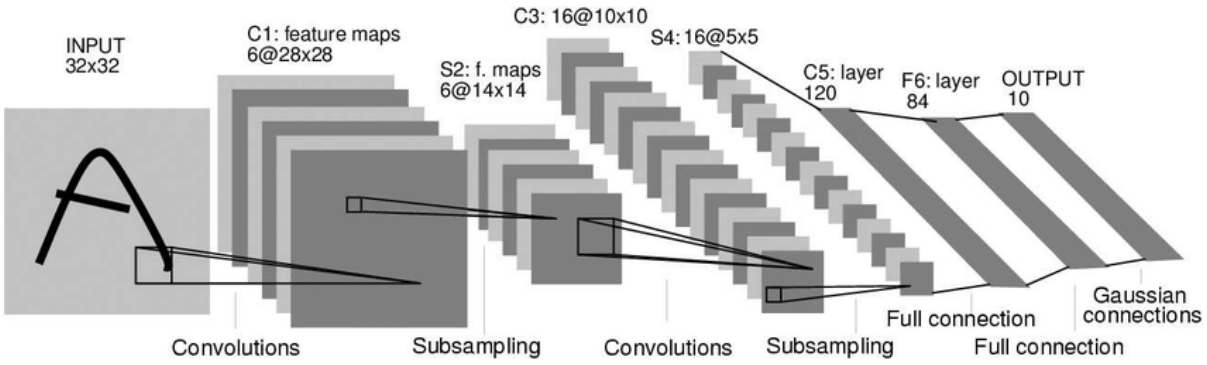


Figure 2.2: Convolutional neural network for image processing, e.g., handwriting recognition.

As shown in the figure, a typical convolutional neural network usually consists of three types of layers: convolutional layer, pooling layer and fully connected layer. Each type of layer will be introduced independently in the following sections, as well as some other types that are not demonstrated in the figure.

Data layer

The data layer is also called the image processing layer. This layer is responsible for optional preprocessing and filtering. Sometimes, the data layer is the raw image, while

other times, it is the image after preprocessing.

Convolutional layer

The convolutional layer plays a significant role, as its operation is related to feature extraction. A kernel is defined as a filter used to extract different characteristics, such as oriented edges, end-points, corners, etc. No matter what the dimension of the input data, each kernel convolves with the input data and produces a 2D feature map. The process is illustrated in Fig. 2.3.

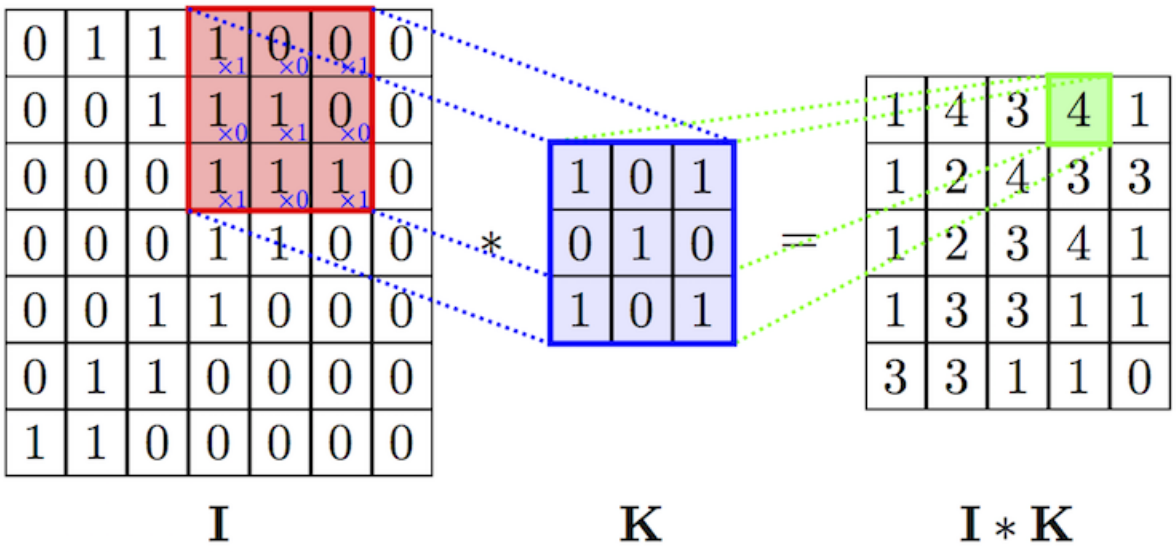


Figure 2.3: Convolutional layer. "I" denotes the input data, "K" denotes the kernel, and " ∗" denotes the convolution operation.

As shown in the figure, the size of the input image is 7×7, while the kernel size is 3×3. The kernel begins filtering the input image from the left-top, multiplication is applied between the corresponding pixels and then addition is applied to calculate the sum as the pixel of the output feature map at the same position. In this example, the kernel moves by one pixel every time; thus, the size of the output feature map is 5×5. However, in practical conditions, the size of the kernel and the stride of the kernel's movement

can be determined by designers. In addition, one kernel can produce one output feature map; to extract adequate features, several kernels can be used with different sizes and pixel values since different kernels are sensitive to different features.

As described before, one kernel can produce one feature map; therefore, the depth of a feature map equals the number of kernels being used. Although the size of the feature map depends on the parameters of the kernel, it can be computed by using a commonly used equation:

$$O = \frac{I - K}{S} + 1 \tag{2.1}$$

where O denotes the output size, I represents the input size, K denotes the kernel size and S denotes the stride. Furthermore, for each layer, the sizes and strides of different kernel are the same; the only difference between kernels is the pixel values.

Apparently, compared with the input image, the width and height of the output feature image are always smaller when using only a kernel to filter the input image. However, the size of the feature map is sometimes not expected to be reduced. To solve this problem, zero padding is applied to expand the border of the input image. The size of the feature map when using zero padding is defined as:

$$O = \frac{I - K + 2P}{S} + 1 \tag{2.2}$$

where P denotes the number of the zero padding, while the others have the same meanings as before. The coefficient 2 preceding P indicates that every padding operation involves adding zeros to both rows and columns.

Pooling layer

The pooling layer is usually used right after the convolutional layer and aims to reduce the dimension of the representation and thus further reduce the number of parameters and the computational complexity of the model [28]. It simply performs down-

sampling along the spatial dimensionality of the given input. By having less spatial information, not only will the computation performance be improved, the probability of overfitting will also be reduced. Generally, two types of pooling are used: max pooling and average pooling. The processes of these two types of pooling are illustrated in Fig. 2.4.

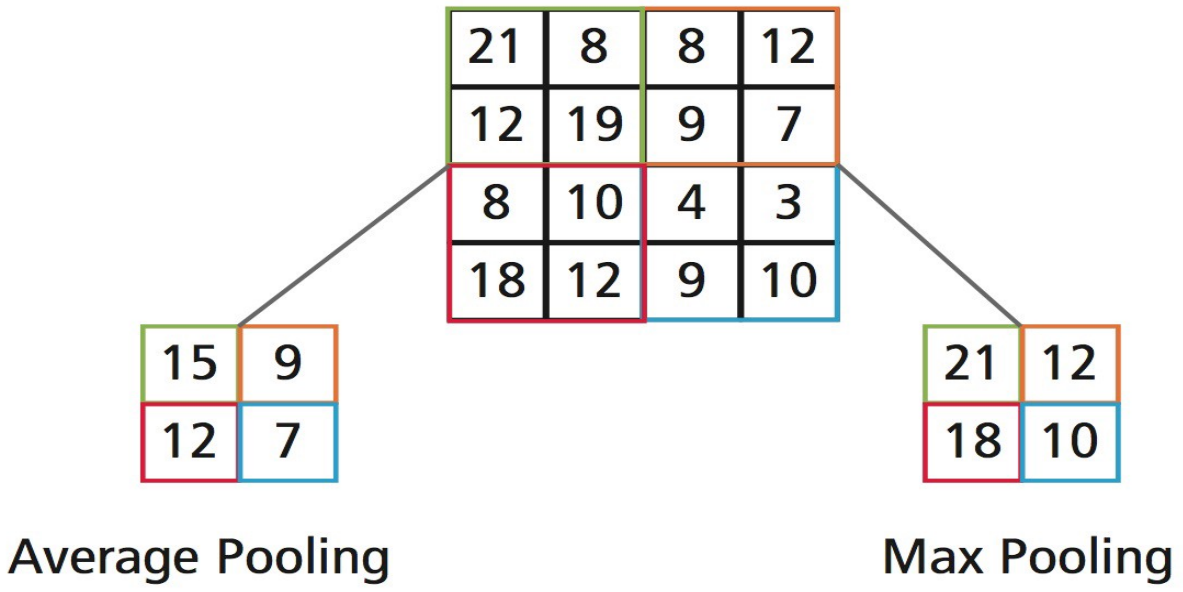


Figure 2.4: Pooling types. The left part represents the process of average pooling with a filter size of 2×2; the right part represents the process of max pooling with a filter size of 2×2.

As shown in the figure, compared with the convolutional layer, the pooling layer has a similar process. A kernel is defined to filter the input, with different results being obtained by using different pooling types. Average pooling is performed to calculate the mean value of the receptive field, while max pooling is simply conducted to select the maximum value of the receptive field. The kernel size and the stride can also be determined by designers. For instance, in the figure, the kernel size is 2×2, and the stride is also 2 for both types (these are commonly used parameters).

Activation layer

The activation layer is actually a node that is added to the end of any neural network to determine the output. It maps the resultant values between 0 and 1, -1 and 1, etc. The activation functions can be divided into two types: linear and nonlinear.

The linear activation functions is hardly used since the output of this function is not confined within any range, which does not help with the parameters of data fed into the neural networks. Nonlinear activation functions are the most used since they help the model to adapt to a variety of data and to distinguish between the outputs. The commonly used nonlinear activation functions are the Sigmoid function, Tanh function and ReLU (rectified linear unit) function.

Dropout layer

The dropout layer is particularly used to avoid the overfitting problem. Deep neural networks are powerful machine learning systems; however, a large number of parameters leads to overfitting, which is a serious problem in networks. The main idea of the dropout layer is to randomly drop units from the neural network during training, which prevents units from co-adapting too much [41]. During the process of training, the dropout layer can stop half of the feature detectors from working, which can improve the generalization of the network capacity. This technique has been popular with deep-learning researchers in recent years.

Fully connected layer

The fully connected layer is also called the inner product layer. It directly connects every neuron in one layer to every neuron in another layer. The number of outputs determined in this layer indicates the number of classifications. If a regression task is conducted, the number of outputs is equal to 1.

2.5 Summary

As introduced before, the convolutional neural network is usually applied to images. Inspired by this, an image containing HR information can be fed into the convolutional neural network to estimate the HR, which is a regression task. Traditional HR estimation involves extracting the PPG signal and using a filter to eliminate the noise caused by the environment or motion artifacts as much as possible. However, these processes may damage some significant information and increase the time consumption.

Eulerian video magnification amplifies the color changes (in the form of redness) on a human face due to cardiac activity, an example of which is shown in Fig. 2.5.

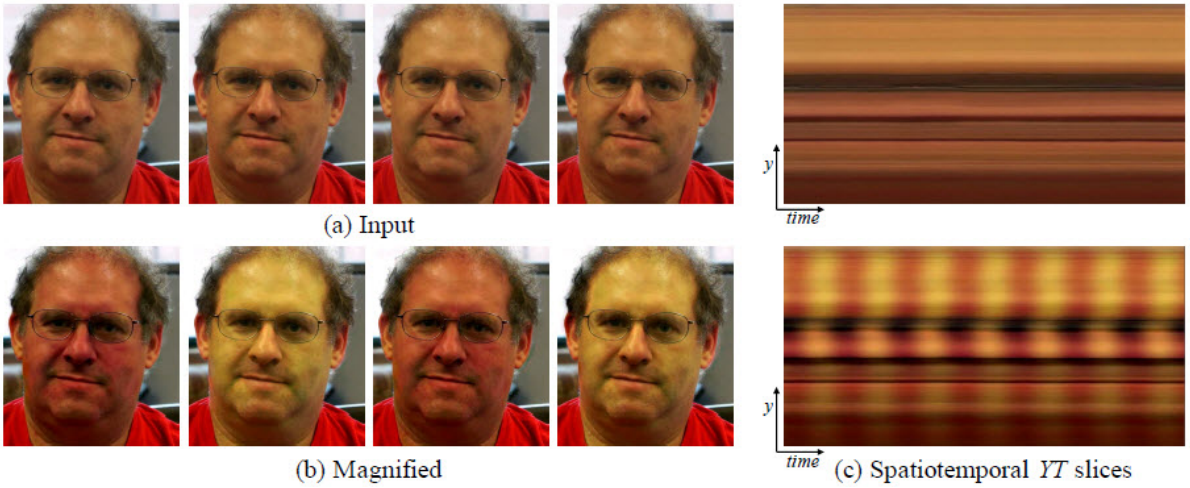


Figure 2.5: An example of using the EVM framework to visualize the human pulse [23]. (a) Four frames from the original video sequence. (b) The same four frames with the subject's pulse signal amplified. (c) A vertical scan line from the input (top) and output (bottom) videos plotted over time shows how our method amplifies the periodic color variation.

Comparing the two results in (c), the vertical scan line of the amplified result shows apparently periodical changes related to the pulse rate. Therefore, the preliminary idea of our work is as follows: an image like this containing HR information can be fed into a

convolutional neural network to estimate the HR. By modifying some details of EVM, a more appropriate module is produced to extract a feature image from a video sequence. The feature image is then fed into a network designed by us. In the next chapter, the algorithm and details of each module will be introduced independently.

Chapter 3

Method:EVM-CNN

In this chapter, a CNN-based contactless real-time HR estimation system is introduced. As mentioned before, noncontact HR estimation is achieved by many researchers via different approaches. In the primary stage, progress was made on non-real-time PPG signal extraction from video recordings. There are two manifolds for acquiring a raw signal that contains the PPG signal: one involves extracting the color variations in the facial region [38] [23] [57] [2], while the other involves extracting the motion variation from subtle head movements [16] [60]. These two types of variations are all due to cardiac activity. Component analysis is then applied to the raw signal to extract the underlying

PPG signal, for which ICA and PCA are commonly used. However, none of these methods can be used to estimate the instantaneous HR. In addition, most of these methods require a well-controlled measurement environment with subjects remaining still. Particularly, the performance of a method based on subtle head motion extraction is poor when movements are included.

Real-time HR estimation has also been achieved in recent years. As introduced before, the chrominance-based method is applied to estimate instantaneous HR. De Haan *et al.* [17] first proposed this approach and showed its advantages in terms of both accuracy under exercise conditions and time consumption. Tulyakov *et al.* [54] proposed a self-adaptive method for automatically selecting a reliable region of a face to estimate the HR and achieved higher accuracy on a challenging dataset.

Inspired by the convolutional neural network, a feature image containing HR information can be fed into a neural network to estimate the HR as a regression task. Spatial decomposition and temporal filtering are applied to extract the feature image. Combined with an ROI extraction module, a new framework for HR estimation is proposed in this chapter. An overview of the proposed approach is given in section 3.1; face detection and tracking, as the first module of our work, is described in section 3.2; feature extraction, as the second module, is described in section 3.4; CNN-based HR estimation, as the last module, is described in section 3.5; and finally, a summary is given in section 3.6.

3.1 Overview

In this section, an overview of the proposed method is presented via three blocks: face detection and tracking, feature extraction, and HR estimation, as illustrated in Fig. 3.1. The face in the input video sequence is first detected and tracked to extract the ROI. Then, the extracted ROI of each frame is processed by using spatial decomposition and temporal filtering to obtain the feature image. Finally, the feature image is input

into a convolutional neural network to estimate the HR. The whole process is a loop, with every iteration (or round) corresponding to every second. In one round, one HR value is obtained. The loop does not stop until reaching the end of the video.

When the RGB video sequence is input into the system, face detection is applied to detect the face and localize the ROI of each frame, which is the cheek region of the face since the cheek region is not involved in the eye blinking and mouth movements. After the bounding box is localized, a tracker is used to track the bounding box to obtain a stable ROI sequence, as shown in the right part of the first block. The tracker is updated every second due to the round duration. This block can automatically extract the ROI sequence and then input it into the next block. The details on how to localize the cheek region and the reason for using the tracker will be given in section 3.2.

The feature extraction block is used to extract the color variation via the ROI sequence and demonstrate it as an image, which is based on EVM. As introduced in section 2.5, EVM can be used to amplify the color changes in a human face in the form of redness due to the pulse rate. A vertical scan of the amplified video sequence shows apparently periodical variation. As a part of EVM, spatial decomposition and temporal filtering are applied to extract the feature image in this block. In section 3.3, spatial decomposition and temporal filtering will be elaborated, as well as the difference between our work and EVM.

In every round, one extracted feature image is fed into the HR estimation block, which is also an RGB image. A convolutional neural network is designed to estimate the corresponding HR from the feature image, which is treated as a regression task. The CNN architecture and parameters will be introduced in section 3.4.

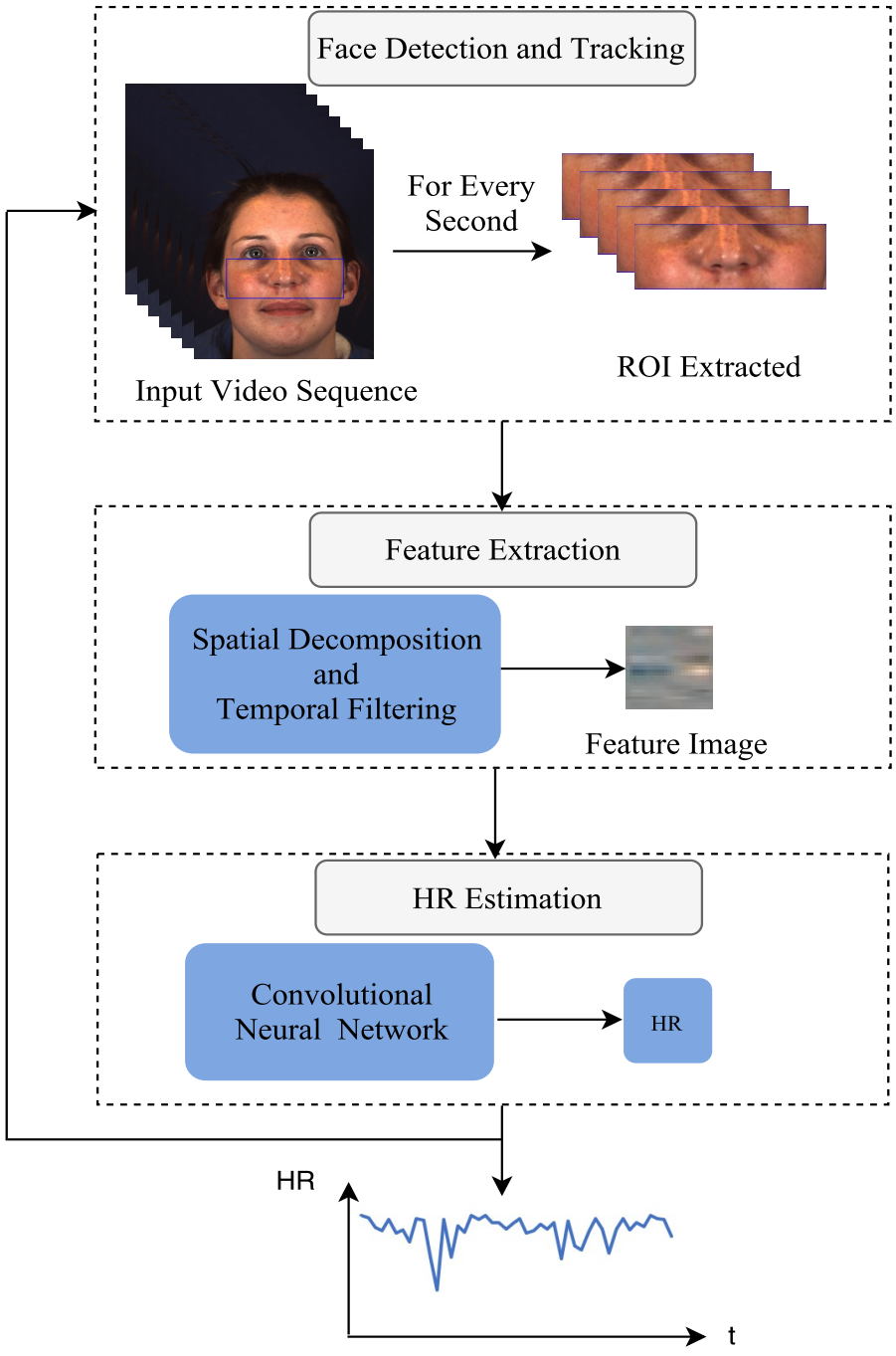


Figure 3.1: Diagram of HR estimation. In the face detection and tracking module, regions of interest are automatically extracted and then input into the next module. Spatial decomposition and temporal filtering are applied to obtain the feature image in the feature extraction module. In the HR estimation module, a CNN is used to estimate the HR from the feature image.

3.2 Face Detection and Tracking

Face detection and tracking are conducted to extract the face regions from the images and maintain the size of the face within a certain time range. The proposed approach is designed for practical conditions, requiring that the ROI be automatically selected and fed into the feature extraction module. Under practical conditions, several factors may impact face region extraction, such as subjects' movements, head rotations and facial expressions. To obtain a steady face region without nonskin pixels, precise facial landmarks are used to define the face region.

A regressing-local-binary-features-based approach is applied to detect the bounding box and 68 facial landmarks inside the bounding box [53]. Considering that eye blinking and mouth movements can cause noise during the measurement, the cheek region is selected as the ROI in our work. To define the ROI, 8 points are used, as shown in Fig. 3.2. The green rectangle is the face detection result, and the 68 facial landmarks are indicated by 68 green points. The blue rectangle is the ROI defined by using the coordinates of the 8 points, which are denoted by numbers in red. The directions of the x- and y-axes are also shown in Fig. 3.2. The left-top vertex coordinate and the size of the blue rectangle are defined in Eq. (3.1) as:

$$\begin{cases} X_{LT} = X_{P_{13}} \\ Y_{LT} = \max(Y_{P_{40}}, Y_{P_{41}}, Y_{P_{46}}, Y_{P_{47}}) \\ W_{rect} = X_{P_{16}} - X_{P_{13}} \\ H_{rect} = \min(Y_{P_{50}}, Y_{P_{52}}) - Y_{LT} \end{cases} \tag{3.1}$$

where X_{LT} and Y_{LT} denote the x and y coordinates of the left-top vertex, respectively, $X_{P_{13}}$ denotes the x coordinate of Point 13, $Y_{P_{40}}$, $Y_{P_{40}}$, $Y_{P_{41}}$, $Y_{P_{46}}$, and $Y_{P_{47}}$ denote the y coordinates of Point 40, Point 41, Point 46, and Point 47, respectively, W_{rect} denotes the

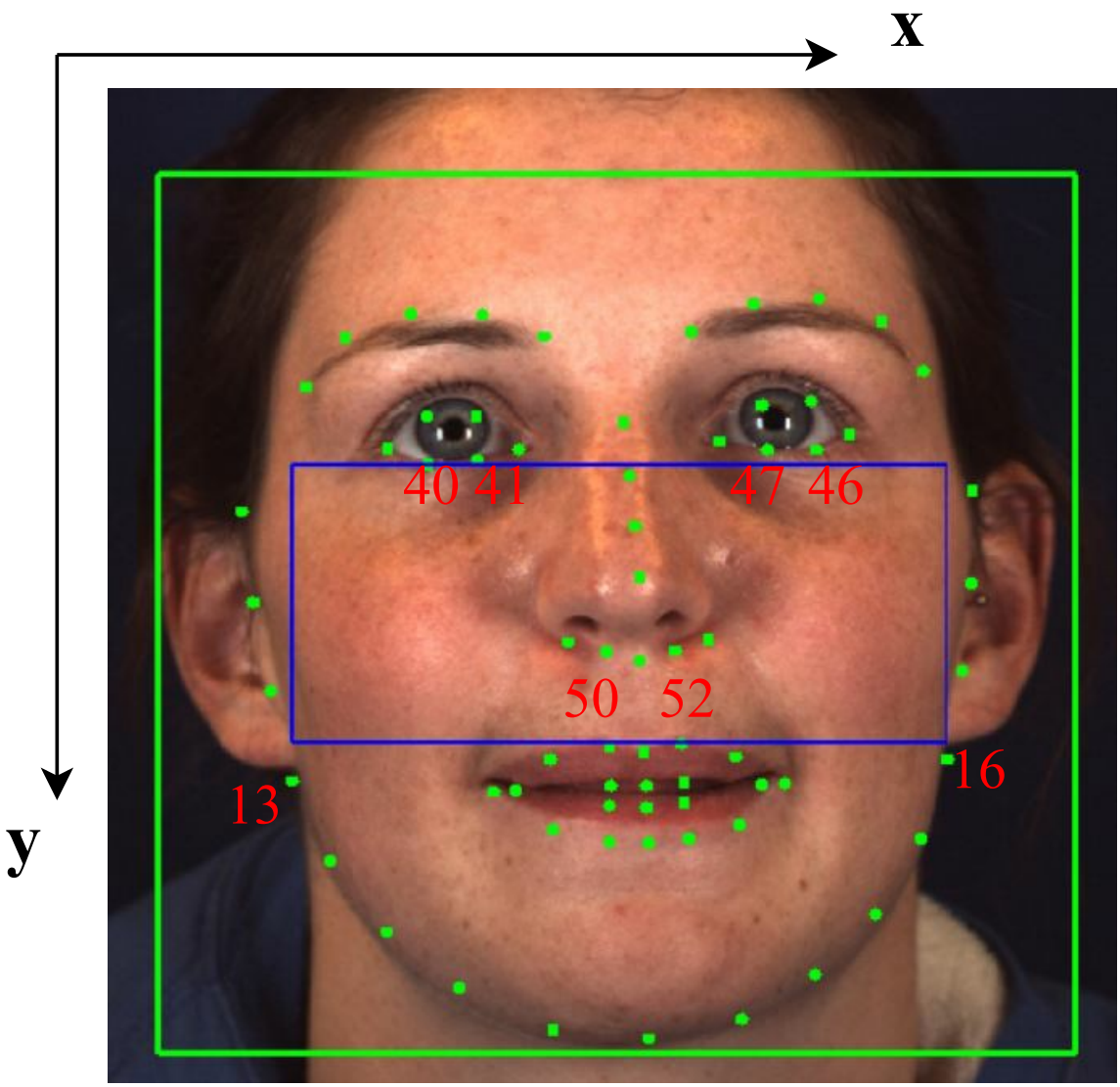


Figure 3.2: Face detection and landmarks. The green rectangle denotes the bounding box of the face detection result. Face landmarks are denoted by the 68 green points. The blue rectangle represents an ROI defined by 8 points, which are denoted by numbers in red.

width of the blue rectangle, $X_{P_{16}}$ and $X_{P_{13}}$ denote the x coordinates of Points 16 and 13, respectively, H_{rect} denotes the height of the blue rectangle, and $Y_{P_{50}}$ and $Y_{P_{52}}$ denote the y coordinates of Points 50 and 52, respectively. By defining the blue rectangle in this way, the ROI always excludes the eye part and the mouth part no matter how the

subjects rotate their heads. Furthermore, the impact caused by the eye blinking and mouth movements is also reduced. Hence, a cheek region without nonfacial pixels is defined.

As the input of the feature extraction system, a fixed-size ROI rectangle is required for a certain time interval since the feature extraction system is used to obtain the color changes of each fixed pixel over time. To achieve this, the scalable kernel correlation filter tracking method [4] is applied to track the ROI rectangle and keep the ROI's size constant within a certain time interval, which can reduce the impact caused by rigid head motion. The tracked cheek region is then input into the feature extraction system to obtain the skin color changes during the cardiac cycle.

3.3 Feature Extraction

In the research of Wu *et al.*, they consider the time series of color values at any spatial pixel and amplify the variation in a given frequency band of interest [23]. Inspired by this, the blood flow information extracted from a video sequence is applied in our approach. Before introducing our work on how to extract the feature image, it is important to understand the main idea of EVM.

The basic approach of EVM is to consider the time series of color values at any pixel and amplify variation in a given temporal frequency band of interest [23]. Temporal filtering needs to be applied to lower spatial frequencies to let the subtle change rise above the camera sensor and quantization noise. Therefore, the input video is first decomposed into different spatial frequency bands. The overview of EVM is shown in Fig. 3.3. Referring to its open source MATLAB code, a Gaussian pyramid is used for color magnification, while a Laplacian pyramid is used for motion magnification. In the MATLAB code, the input RGB video is treated as a 4-dimensional array to be downsampled with 4 levels. The lowest level of the resultant video sequence then

undergoes the temporal filtering procedure.

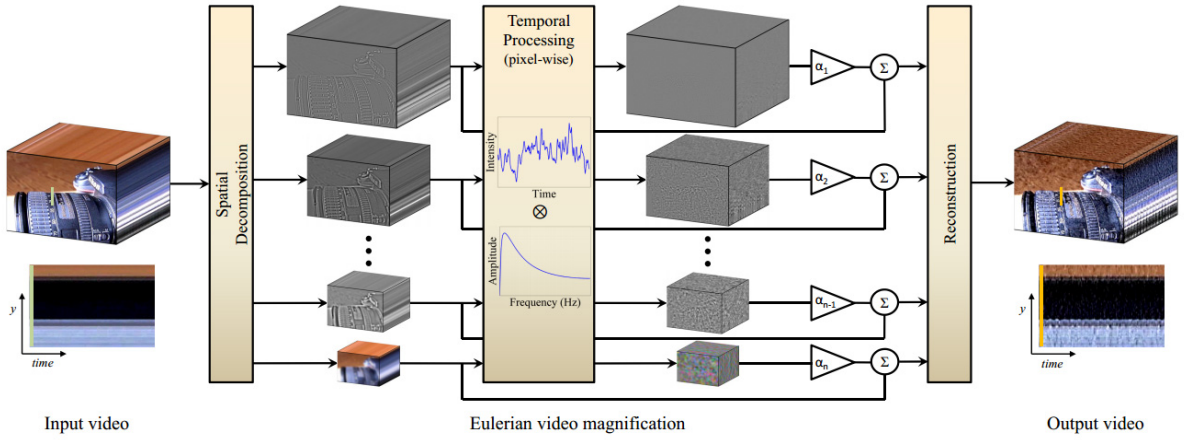


Figure 3.3: Overview of the Eulerian video magnification framework. The system first decomposes the input video sequence into different spatial frequency bands and then applies temporal filtering. The filtered spatial bands are then amplified by a given factor, added back to the original signal, and collapsed to generate the output video.

For temporal filtering, an ideal bandpass filter is designed to eliminate the noise in the band of interest. The given example of color magnification for a face video is filtered using a band of 0.83-1 Hz, which corresponds to 50-60 bpm. This parameter is defined on the basis of the subject being measured using a wearable sensor. The input video is adapted to the frequency domain by applying the fast Fourier transform (FFT) and then multiplied by a 4-dimensional mask designed as the ideal bandpass filter. The resultant 4-dimensional array is adapted back to the time domain to obtain the filtered array. A magnification factor is then determined and multiplied by the filtered array. This amplified array is upsampled and added to the original video to obtain the final result, which is shown in Fig. 2.5.

By using localized spatial pooling and temporal filtering to extract the signal corresponding to the pulse, the extracted signal is multiplied by a magnification factor to amplify the facial color changes over time and make it visible to the naked eye. However,

our purpose is to extract the signal containing the pulse rate. Inspired by this, spatial decomposition and temporal filtering are applied in our approach to extract the feature image that contains the signal related to the HR information. To reduce the processing time, this module is written in C++, and the 4-dimensional array is modified to yield a 2-dimensional image. The overview of spatial decomposition and temporal filtering is presented in Fig. 3.4. As shown in (a), the input sequence is first decomposed into different spatial frequency bands. Then, the lowest-band sequence is reshaped and concatenated to obtain a new image, as shown in (b). Finally, a bandpass filter is used to obtain the feature image containing the signal related to blood flow, as shown in (c).

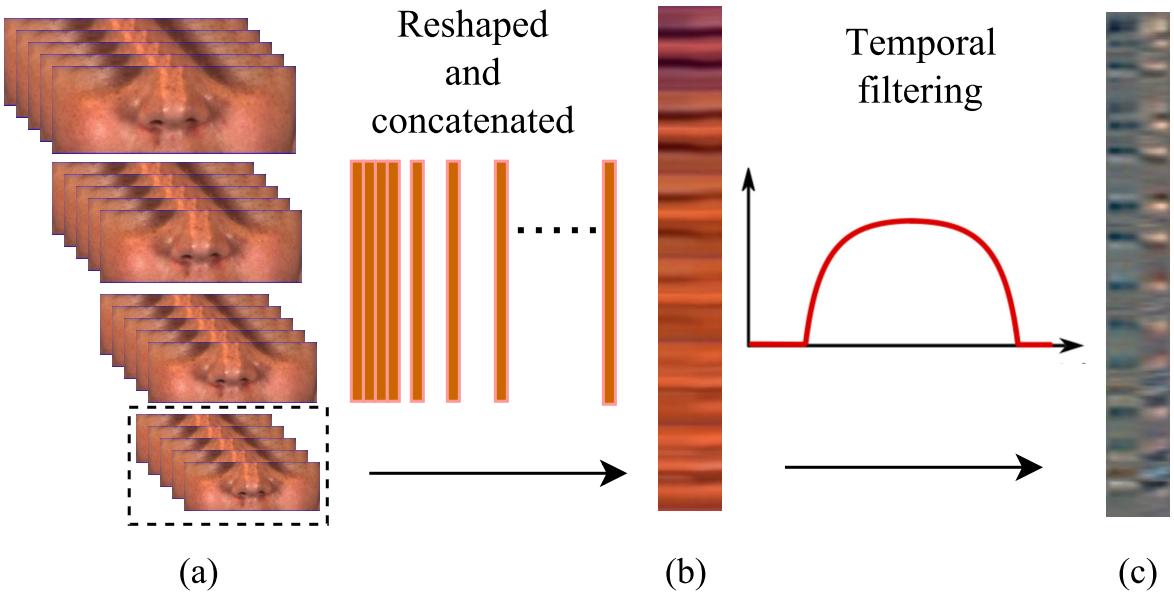


Figure 3.4: Feature extraction. (a) shows that the system decomposes the input sequence of the ROI into different spatial frequency bands. (b) demonstrates the result after reshaping and concatenating the lowest band of (a), which is indicated by a black dashed rectangle. (c) is the feature image obtained by applying temporal filtering to the concatenated image with a frequency band of interest.

Spatial decomposition for color magnification is implemented by using Gaussian pyramid decomposition. Each frame of the input RGB video is downsampled to a given

level. The last layer of each downsampled frame, which is indicated by the black dashed rectangle in Fig. 3.4 (a), is reshaped into one column. All the columns obtained from the last step are then concatenated together to create a new image. To detect the instantaneous HR, a one-second-interval video sequence, rather than the whole video, is input into the feature extraction module every round. Hence, the number of columns of the concatenated image in Fig. 3.4 (b) should be equal to the number of frames per second.

Temporal filtering involves using an ideal bandpass filter to obtain the signal with the frequency of interest. Different from [23], a wider bandpass filter is used here to cope with the usual conditions. A human's HR is between 45-240 bpm; the corresponding frequency band is computed as $f_{HR} = HR/60$, which is 0.75-4.0 Hz. As explained before, each column of the concatenated image corresponds to a downsampled image of the ROI, and each row of the concatenated image corresponds to the variations at a fixed position (pixel) in the ROI over one second. Therefore, each channel of the concatenated image is transferred to the frequency domain by using FFT by rows. Then, a mask with the same size is used to retain the component within the frequency band of interest while making the others equal to zero. Finally, the inverse fast Fourier transform (IFFT) is performed by rows to transfer the signal back to the time domain, and three channel images are merged to obtain the feature image, as shown in Fig. 3.4 (c). The spatial decomposition and temporal filtering algorithm is referred to as feature extraction and summarized in Algorithm 1.

In the given example of the MATLAB source code, the face video used for color magnification shows an exact face region without any movement. However, in our case, natural movement and facial expressions are allowed. Face detection is applied to cope with this issue. By reviewing the whole process of feature extraction, the size of the frame needs to be maintained during a round, which is the reason for using a tracker. If

Algorithm 1 Feature Extraction

Input: The original RGB video frames I, pyramid level Pl, frame rate Fps, set of one-column intermediate images C, low-frequency cut-off Fl and high-frequency cut-off Fh of the ideal bandpass filter

Output: A set of feature images S.

repeat
  for each frame I **do**
    Gaussian pyramid with level Pl
    Reshape the last level to one column C_0
  end for
  $C \leftarrow C_0$
until $size(C) = Fps$
repeat
  for each set C **do**
    Concatenate all the columns to create a new image M
    for each channel of M **do**
      Do FFT to obtain Mf
      Create a mask by using Fl and Fh
      Multiply the mask by Mf to obtain N
      Apply IFFT to N to obtain Ni
    end for
    merge 3 channels to obtain K
  end for
  $S \leftarrow K$
until the whole video ends
return S

not, the size of the returned bounding box of each frame will change due to movements.

3.4 HR Estimation by CNN

The HR is estimated from a temporal image, which is extracted in the previous procedure. This is achieved by using a regression convolutional neural network. A feature image is input into the network, and a corresponding HR is output. The size of the input image is defined as 25×25×3 due to the output of the feature extraction module. As mentioned before, the input RGB video sequence has three channels. In the feature extraction module, for each channel, the extracted ROI is reshaped to one column; then, all the reshaped frames within one second are concatenated together; and finally, three channels are merged to obtain the feature image. Hence, the feature image is also an RGB image, with the number of columns equal to the frame rate. In our case, the frame rate of the input video sequence is 25, which indicates that the column number of the feature image is 25. The horizontal direction of the feature image contains the temporal signal which should not be changed. Therefore, the input image to be fed into the network is defined as having a size of 25×25×3. On this basis, a shallow neural network is designed, as shown in Fig. 3.5.

There are three convolutional layers in the network. Downsampling is achieved by the max pooling layers. Each convolutional layer is followed by a batch normalization and ReLU nonlinearity. A dropout layer is used to avoid overfitting, and the dropout ratio is set to 0.6. Since the HR of each feature image is the only one labeled, the last fully connected layer also has one neuron. All the labels are normalized to the range of 0-1 before training, where 0 corresponds to 45 bpm and 1 corresponds to 240 bpm. The Euclidean distance is chosen as the loss function to measure the difference between the predicted value and the ground-truth value. The equation is defined in Eq. (3.2):

$$L_{Eu} = \frac{1}{2N} \sum_{i=1}^{N} \|x_i^1 - x_i^2\|_2 \tag{3.2}$$

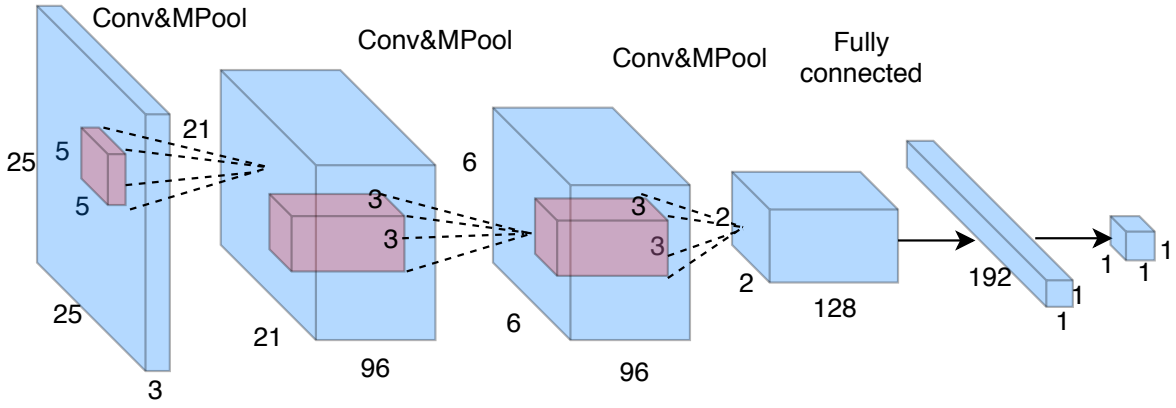


Figure 3.5: The structure of the convolutional neural network, where, for each layer, the light blue cube denotes the input and the light pink cube denotes the reception field. The size of each cube is written next to it, the relative number of channels is represented by the thickness, and the arrow denotes the data flow.

where L_{Eu} denotes the Euclidean distance, which reflects the sum of the squares of differences between the label value, denoted by x_i^1, and the predicted value, denoted by x_i^2, and N indicates the number of samples.

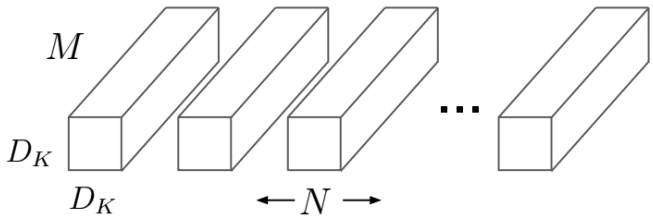
3.5 Network optimization

Option 1

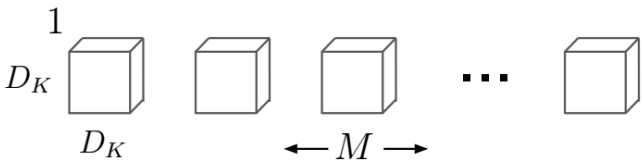
To optimize the network, the model size and time consumption are regarded as the aspects that can be improved. Inspired by MobileNetV1 [22], depthwise separable convolutions are used as the main structure in this part. Depthwise separable convolution is a form of factorized convolution that is used to factorize a standard convolution into a depthwise convolution and a 1×1 pointwise convolution, which can effectively reduce the computational complexity and model size.

A standard convolution filter inputs and combines them into a new set of outputs

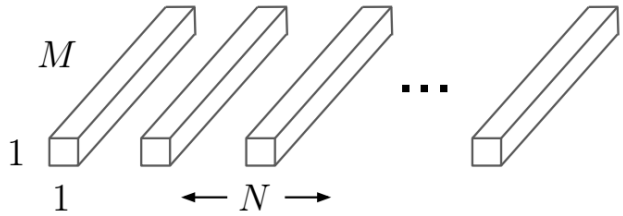
simultaneously. However, this one layer is divided into two layers in the depthwise separable convolution. The depthwise convolution involves applying a single filter to each channel of the input. The pointwise convolution involves applying a 1×1 convolution to linearly combine the outputs of the depthwise convolution. The details of factorization are shown in Fig. 3.6.



(a) Standard Convolution Filters



(b) Depthwise Convolutional Filters



(c) 1×1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

Figure 3.6: Depthwise separable convolution. The standard convolution shown in (a) is divided into two layers: depthwise convolution, shown in (b) and pointwise convolution, shown in (c).

Assume that a $D_F \times D_F \times M$ feature map is input into a standard conlvolution layer and that the output is $D_G \times D_G \times N$, where D_F denotes the width and height of the squared input feature map, M represents the number of input channels, D_G

denotes the width and height of the squared output feature map and N represents the number of output channels. As shown in Fig. 3.6 (a), the standard convolutional layer is parameterized by kernel K of size $D_K \times D_K \times M \times N$, where D_K is the width and height of the squared kernel. The computational cost of the standard convolution is:

$$D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F \tag{3.3}$$

For depthwise separable convolution, depthwise convolution is first applied to filter each input channel; then, pointwise convolution is applied to combine the output of the depthwise convolution. The structure of the depthwise convolutional filters is shown in Fig. 3.6 (b); each filter is $D_K \times D_K \times 1$, and there are M filters in total. The outputs of this layer are then linearly combined by a pointwise convolutional layer, as shown in Fig. 3.6 (c). There are N filters with a size of $1 \times 1 \times M$. The computational cost of the depthwise convolutional layer is:

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F \tag{3.4}$$

The computational cost of the pointwise convolutional layer is:

$$M \cdot N \cdot D_F \cdot D_F \tag{3.5}$$

The reduction in computational cost via depthwise separable convolution is calculated as:

$$\frac{D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F}{D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F} = \frac{1}{N} + \frac{1}{D_K^2} \tag{3.6}$$

The details of the CNN body architecture, optimized by using depthwise separable convolution, are illustrated in Table 3.1. The first layer is a fully convolutional layer, which is denoted by “Conv", while for the next layer, a depthwise convolutional layer

Input Size	Type/Stride	Filter Shape
$25 \times 25 \times 3$	Conv/s1	$5 \times 5 \times 3 \times 96$
$23 \times 23 \times 96$	DwConv/s1	$3 \times 3 \times 96$ dw
$21 \times 21 \times 96$	PwConv/s1	$1 \times 1 \times 96 \times 96$
$21 \times 21 \times 96$	DwConv/s2	$3 \times 3 \times 96$ dw
$11 \times 11 \times 96$	PwConv/s1	$1 \times 1 \times 96 \times 96$
$11 \times 11 \times 96$	DwConv/s2	$3 \times 3 \times 96$ dw
$6 \times 6 \times 96$	PwConv/s1	$1 \times 1 \times 96 \times 128$
$6 \times 6 \times 128$	DwConv/s2	$3 \times 3 \times 128$ dw
$3 \times 3 \times 128$	PwConv/s1	$1 \times 1 \times 128 \times 128$
$3 \times 3 \times 128$	DwConv/s2	$3 \times 3 \times 128$ dw
$2 \times 2 \times 128$	PwConv/s1	$1 \times 1 \times 128 \times 128$
$2 \times 2 \times 128$	AvePool/s1	Pool 2×2
$1 \times 1 \times 192$	FC/s1	128×192
$1 \times 1 \times 192$	Dropout/s1	ratio 0.6
$1 \times 1 \times 192$	FC/s1	192×1
$1 \times 1 \times 1$	Eu/s1	Regression

Table 3.1: CNN body architecture optimized by depthwise separable convolution.

is used to filter each input channel, and then a pointwise layer is used to combine the outputs of the depthwise convolutional layer, which are denoted by “DwConv" and “Pw-Conv", respectively. It can be seen that the first fully convolutional layer is achieved by applying a kernel of size 5×5×3×96, where 5×5 are the height and width of the filter, 3 represents the depth corresponding to the number of input channels (RGB), and 96 is the number of output feature maps. In the following depthwise convolutional layer, a kernel of size 3×3×96 is used to filter the input feature map, where 3×3 denotes the height and width and 96 is the number of input channels. A hidden parameter of this filter

is the depth, which is always equal to 1 due to the property of depthwise convolution. Similarly, the size of the next pointwise filter is $1 \times 1 \times 96 \times 96$, where 1×1 are the height and width, the first 96 is the depth of the filter, and the second 96 is the number of outputs. In the following parts, a similar depthwise separable convolution structure is used. Each convolutional layer is followed by a batch normalization and ReLU nonlinearity. Examples of a fully convolutional layer and a depthwise separable convolutional layer are given in Fig. 3.7. Downsampling is achieved by changing the value of the stride. An average pooling layer downsamples the final feature map to 1 and computes the average value of each channel.

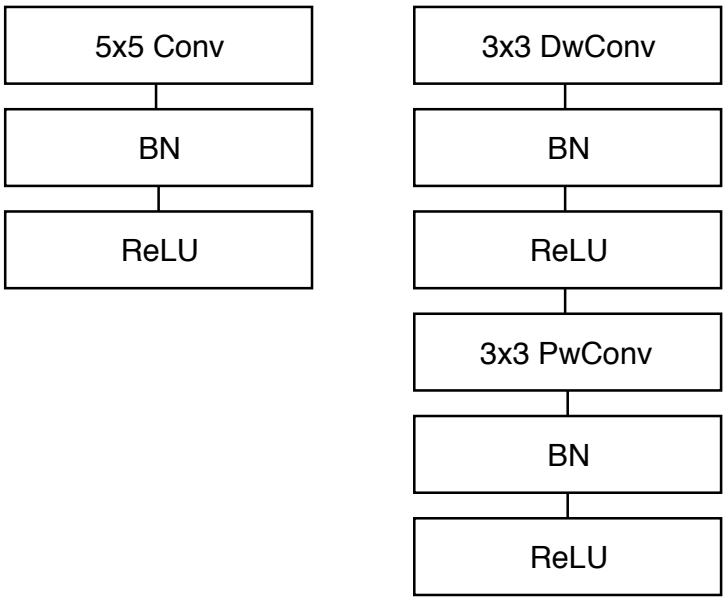


Figure 3.7: Two examples of nonlinearity. A standard convolutional layer followed by batch normalization and ReLU is shown on the left. Depthwise separable convolutions followed by batch normalization and ReLU are shown on the right.

Option 2

In 2018, a new mobile architecture called MobileNetV2 was proposed, which improves upon MobileNetV1 [22] by using an inverted residual with a linear bottleneck

[35]. The input image is treated as a low-dimensional compressed representation by this module. It is first expanded to high dimensions and filtered using a lightweight depthwise convolution. Features are subsequently compressed to low dimensions by using a linear convolution, which is called a linear bottleneck. In their research, the use of nonlinearity after depthwise separable convolution inevitably results in lost information in that channel. Assuming that the manifold of interest is low-dimensional, this can be captured by inserting linear bottleneck layers into the convolutional blocks. The results of their experiments indicate that using linear layers is crucial, as it prevents nonlinearities from destroying too much information. The evolution of the linear bottleneck is shown in Fig. 3.8.

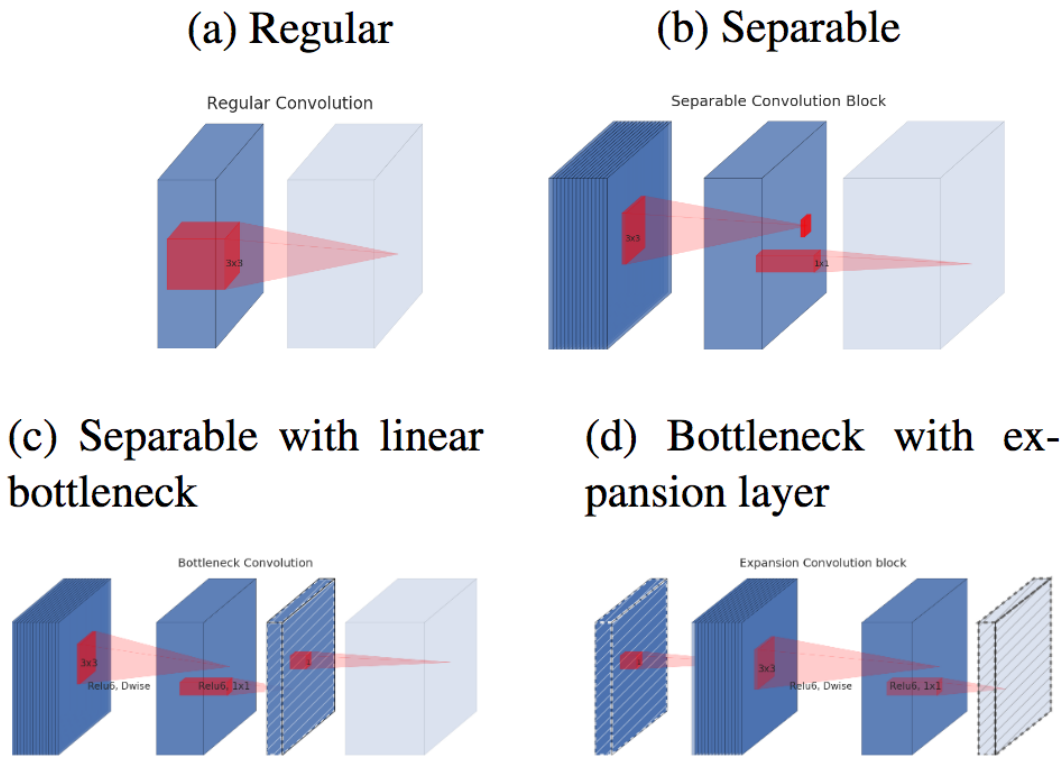


Figure 3.8: Evolution of linear bottleneck blocks. The diagonally hatched texture indicates layers that do not contain nonlinearities. The lightly colored layer indicates the beginning of the next block.

To improve the ability of a gradient to propagate across multiplier layers, shortcuts are inserted in the network. Different from a traditional residual network [27], an inverted residual block is used in MobileNetV2 [35]. This is achieved by using shortcuts directly between the bottlenecks instead of expansion layers. Their experimental results show that the shortcuts connecting bottlenecks perform better than shortcuts connecting the expanded layers. The difference between a residual block and inverted residual block is shown in Fig. 3.9. The classical residuals connect the layers with a large number of channels, as shown in (a), whereas the inverted residuals connect the bottlenecks, as shown in (b).

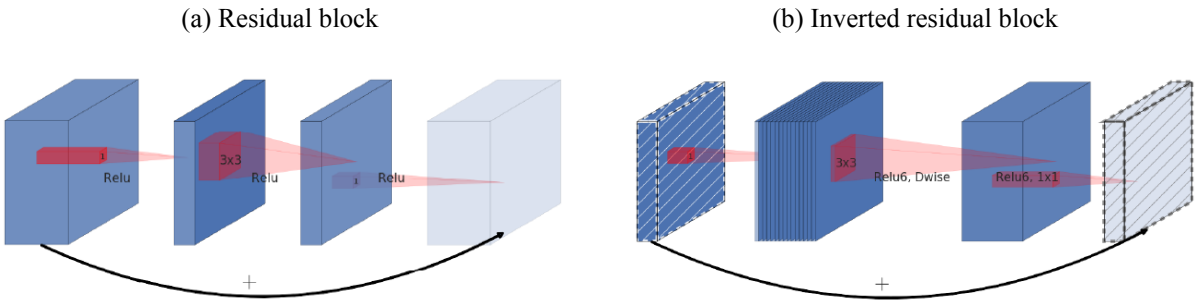


Figure 3.9: The difference between a residual block and an inverted residual. Diagonally hatched layers do not include nonlinearities. The thickness of each block indicates its relative number of channels.

The basic structure of inverted residuals with a linear bottleneck is detailed in Table 3.2. For a feature map of size $h \times w \times d$, the expansion factor t expands the number of

Layer type	Input	Operator	Output
Expansion layer	$h \times w \times d$	1×1 Conv, ReLU	$h \times w \times (td)$
Depthwise convolutional layer	$h \times w \times (td)$	3×3 DwConv s=s, ReLU	$\frac{h}{s} \times \frac{w}{s} \times (td)$
Linear bottleneck layer	$\frac{h}{s} \times \frac{w}{s} \times (td)$	1×1 Conv	$\frac{h}{s} \times \frac{w}{s} \times d'$

Table 3.2: An example structure of a bottleneck residual block.

channels from d to td. Depthwise convolution filters the expanded block with stride s to downsample the pixels of each channel. The linear bottleneck decreases the number of channels to d' using a linear convolution. Assuming that the kernel size is k, the total computational cost is $h \cdot w \cdot d \cdot t(d + k^2 + d')$.

The details of the CNN body architecture optimized by using an inverted residual with a linear bottleneck are given in Table 3.3. Each line of the table represents

Input Size	Operator	e	c	n	s
$25 \times 25 \times 3$	Conv5 $\times$ 5	-	32	1	1
$23 \times 23 \times 32$	bottleneck	1	16	1	2
$12 \times 12 \times 16$	bottleneck	4	32	2	2
$6 \times 6 \times 32$	bottleneck	4	64	3	2
$3 \times 3 \times 64$	bottleneck	4	128	1	1
$3 \times 3 \times 128$	Conv1 $\times$ 1	-	192	1	1
$3 \times 3 \times 192$	AvePool3 $\times$ 3	-	-	1	-
$1 \times 1 \times 192$	Conv1 $\times$ 1	-	1	1	1

Table 3.3: CNN body architecture optimized by an inverted residual with a linear bottleneck.

a sequence of 1 or more identical layers, repeated n times. e denotes the expansion ratio, and c denotes the number of output channels. The first depthwise convolutional layer of each sequence has a stride s; all others have a stride of 1. All the depthwise convolutional layers in the bottleneck sequences use $3 \times 3 \times 1$ kernels. The first layer is a fully convolutional layer followed by four bottleneck blocks. Each bottleneck sequence consists of three convolutional layers, as described in Table 3.2. Particularly, for different

strides, the bottleneck block has different structures to match the dimensions of the shortcut. A structure based on different strides is shown in Fig. 3.10.

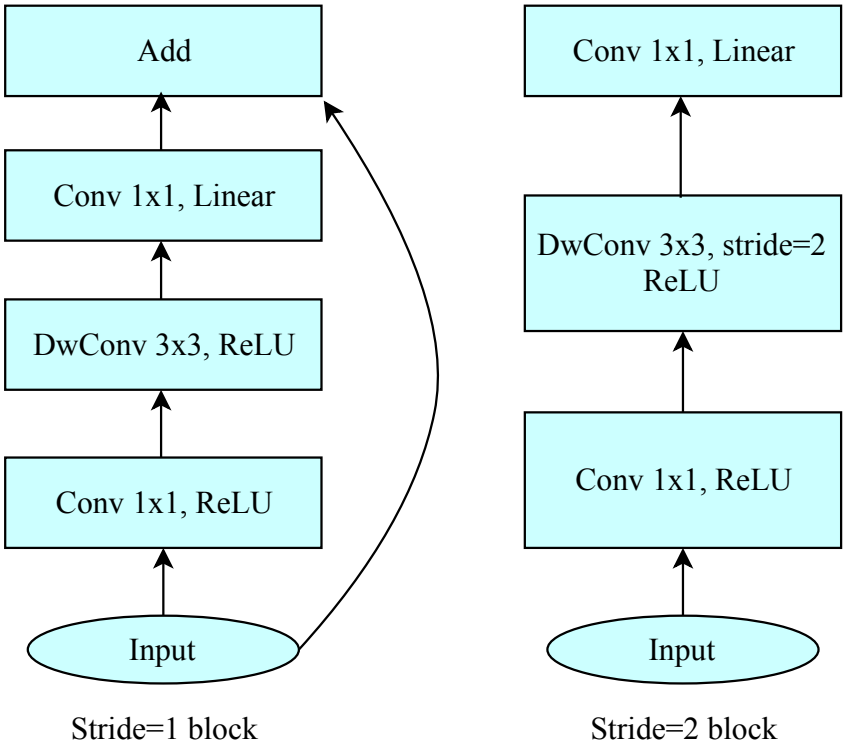


Figure 3.10: Bottleneck blocks with different strides. The left structure with a stride of 1 has a shortcut, while the right structure with a stride of 2 does not have a shortcut.

3.6 Summary

In this chapter, the proposed method is introduced in detail. Three procedures are combined to achieve an end-to-end HR estimation system. Face detection and tracking are applied to define the ROI from the input video sequences. Face landmarks are used to define the bounding box of the cheek region. A tracker is used to maintain the size of the ROI within a given time interval and reduce the effect caused by rigid motion. The sequence of the ROI is then input into the feature extraction module. Spatial decomposition is applied to obtain the lower spatial band of the sequence, which

is then reshaped and concatenated to obtain an intermediate image. Temporal filtering is applied to obtain the frequency band of interest from the intermediate image. The resultant image is the feature image that will be fed into the CNN estimator. A shallow CNN is designed due to the input size of the feature image. Considering the model size and efficiency, two options for optimization are introduced: one involves optimizing the network by using a depthwise separable convolution structure; the other involves using inverted residual blocks with a linear bottleneck. The training results of these three networks will be shown and compared in a subsequent chapter. The best trained model will be used for HR estimation experiments. The model size and run speed of each network will be illustrated in Chapter 5. Overall, our system can automatically detect a human face and estimate the HR from a facial video. To evaluate our system, experiments are designed and introduced in the next chapter.

Chapter 4

Experiments

To test our system comprehensively, a challenging dataset is selected; it is also the benchmark for our comparison experiments. In this chapter, the dataset used in our experiments is first introduced. The evaluation metrics, which are commonly used to evaluate results for this topic, are illustrated and explained in the following. The implementation details are also described. The training results of three networks are demonstrated, and the validation losses are compared. The best result of the trained model with the least validation loss will be used in subsequent experiments. Finally, three experiments, which are used to evaluate the method and compare it with other

state-of-the-art approaches, are described.

4.1 Dataset

For the purpose of estimating the HR instantaneously, our strategy is to extract the feature image at every second of video sequences and then predict the HR from the feature images. In other words, consecutive video frames of one second are input into the feature extraction system, and then one corresponding feature image is output to be used to predict the HR. Therefore, the number of feature images extracted from a whole video is the number of total frames divided by the frame rate. To estimate the HR under practical conditions, a challenging dataset is used in this paper, which is introduced in the following.

The MMSE-HR dataset is a subset of the MMSE database [65], which was generated specifically for challenging HR estimation. There are 40 participants with diverse ethnic ancestries that contributed to the data collection for HR acquisition. The overall structure of the MMSE database is shown in Fig. 4.1. In the HR subset, 102 RGB videos are recorded, and the length of each video is between 30 seconds and 1 minute. The frame rate is 25 fps, and the resolution of each 2D texture RGB image is 1040×1392 pixels. The HR is collected by a contact sensor working with a sample rate of 1 kHz. Since our experiment requires the average HR at every second to be the label of each feature image, the mean of the 1000 HR values in each second is calculated and used in the HR estimation. Each video of the dataset is input into the modules described in Section 3.2 and Section 3.3, ultimately yielding 5839 feature images of the whole dataset.

Considering the importance of data diversity in the training task, HR data play a significant role in our experiments as the labels. To show the HR data diversity of the dataset, the HR distribution is demonstrated by calculating the proportion for each range, which is shown in Fig. 4.2, where the blue bar represents the total HR distribution

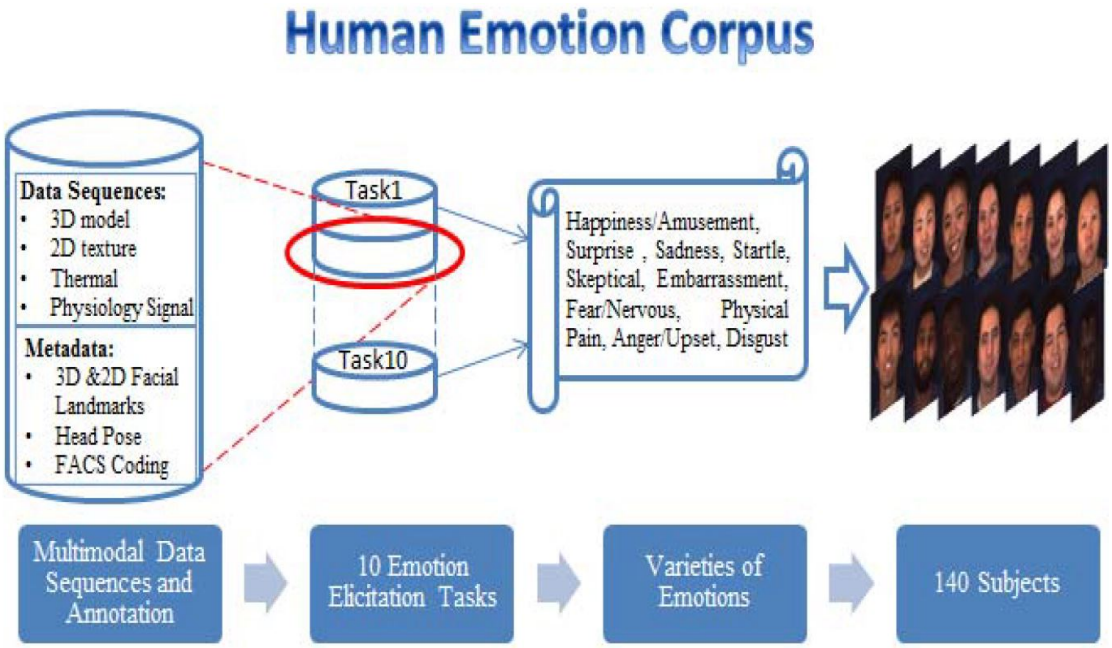


Figure 4.1: Overall structure of the MMSE database.[65]

of the dataset. For the purposes of training and testing for HR estimation, 730 feature images are randomly selected as the test dataset, while the remaining images are used for training and validation. The HR distributions of these two subsets are also illustrated in Fig. 4.2, where the orange bar denotes the training and validation dataset and the gray bar denotes the test dataset. The total HR data has a minimum value of 45 bpm and a maximum value of 185 bpm, the mean value is 84.28, while the variance is 355.9. From Fig. 4.2, it can be seen that the proportion of the total HR data between 65 bpm and 95 bpm is 73% and that the two subsets show similar distributions compared with the total dataset. For the training and validation dataset, the mean of the HR is 84.32, and the variance is 355.6, while for the test dataset, the mean of the HR is 84.0, and the variance is 357.6.

Apparently, the proportion of the HR within a mid-frequency band constitutes most of the dataset. This property can affect the performance of the training result, which will be demonstrated and analyzed in the next chapter. In addition, the two subsets are

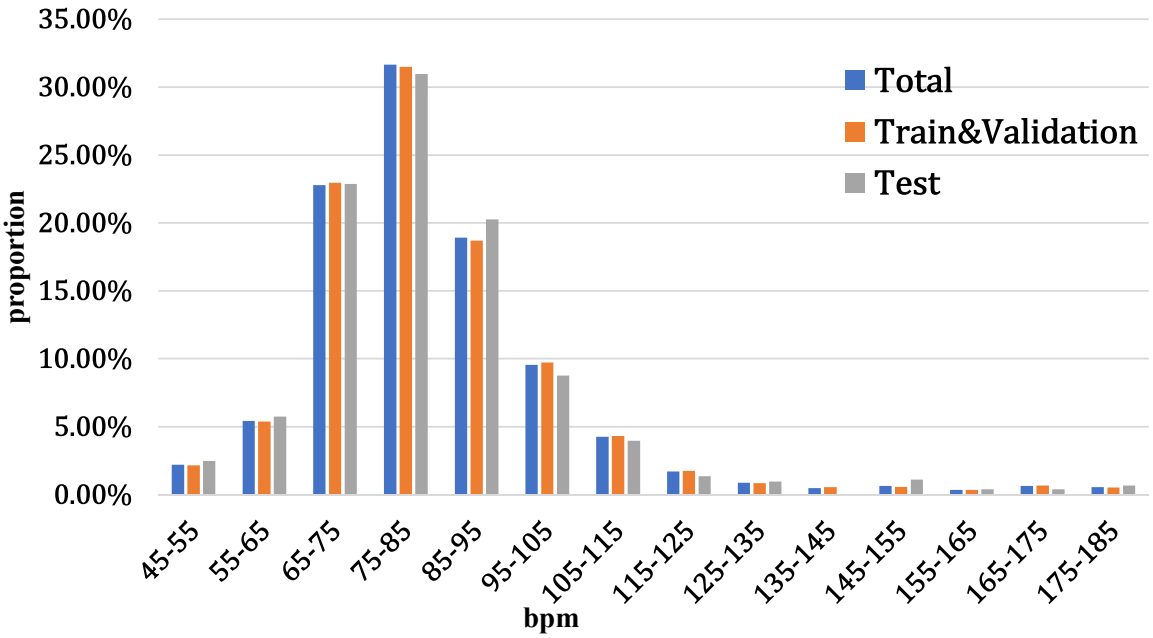


Figure 4.2: Heart rate proportion from different datasets. The blue bar denotes the HR data from the MMSE-HR dataset [65], the orange bar denotes the HR data from the MCR-LAB dataset, and the gray bar denotes the total HR of both datasets.

divided randomly, having a distribution similar to that of the total dataset to guarantee consistency in both training and testing tasks.

4.2 Evaluation Metrics

In our experiments, five evaluation metrics are adopted to evaluate the HR prediction: mean error, standard deviation, root mean squared error, mean absolute percentage error and Pearson's correlation. All these evaluation metrics are commonly used in other methods and are thus used in our evaluation to enable a comparison with these other methods. All the evaluation metrics are based on the HR error, which is the difference between the predicted HR and the ground truth. The HR error is defined in Eq. (4.1):

$$H_e(i) = H_p(i) - H_{gt}(i) \tag{4.1}$$

where H_e denotes the measurement error, H_p denotes the predicted HR and H_{gt} denotes the ground truth of the HR, i indicates the i-th video sequence as well as those in the following equations.

Mean error

The mean of the measurement error is the average value of H_e, which is computed using Eq. (4.2):

$$M_e = \frac{1}{N}\sum_{i=1}^{N} H_e(i) \tag{4.2}$$

where M_e denotes the mean of the measurement error and N denotes the number of measurements.

Standard deviation

This is used to quantify the amount of variation of a set of data values. In our experiments, the standard deviation is used to evaluate the dispersion of H_e, which is defined in Eq. (4.3):

$$SD_e = \sqrt{\frac{\sum_{i=1}^{N}(H_e(i) - M_e)^2}{N}} \tag{4.3}$$

where SD_e denotes the standard deviation. $SD_e \in [0, \infty)$, where a lower value indicates that the data points of H_e tend to be closer to the mean value.

Root mean squared error

This is used to measure the differences between values predicted by an estimator and the values actually observed and is sensitive to outliers. The equation is defined in

Eq. (4.4):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(H_p(i) - H_{gt}(i))^2}{N}} \tag{4.4}$$

where $RMSE$ denotes the root mean squared error. $RMSE \in [0, \infty)$, where a lower value indicates that fewer outliers exist in H_e.

Mean absolute percentage error

This is a measure of the prediction accuracy, which usually expresses the accuracy as a percentage. The formula is defined in Eq. (4.5):

$$M_{eRate} = \frac{1}{N}\sum_{i=1}^{N}\frac{|H_e(i)|}{H_{gt}(i)} \tag{4.5}$$

where M_{eRate} denotes the mean absolute percentage error. $M_{eRate} \in [0, +\infty)$, where a lower value indicates that the predicted value is closer to the ground truth.

Pearson's correlation

This is used to measure the linear correlation between the predicted HR and the ground truth. Its formula is defined in Eq. (4.6):

$$\rho = \frac{\sum_{i=1}^{N}(H_{gt}(i) - \overline{H_{gt}})(H_p(i) - \overline{H_p})}{\sqrt{\sum_{i=1}^{N}(H_{gt}(i) - \overline{H_{gt}})^2}\sqrt{\sum_{i=1}^{N}(H_p(i) - \overline{H_p})^2}} \tag{4.6}$$

where ρ denotes Pearson's correlation, $\overline{H_{gt}}$ denotes the mean value of the ground truth and $\overline{H_p}$ denotes the mean value of the predicted HR. $\rho \in [-1, +1]$, where 1 denotes a total positive linear correlation, while -1 denotes a total negative correlation.

4.3 Implementation details

All the experiments are performed on the Windows 10 platform, and the program is developed by using C++. The convolutional neural network (Section III-C) is utilized on the Caffe deep learning platform [18]. For the face detection and tracking module, the coordinates of 8 facial landmarks are invoked to extract the ROI of each frame. For the

Figure 4.3: An example of failed face detection [65]. Three consecutive frames show that the subject's face is outside the picture due to large movements.

feature extraction module, the level of the Gaussian pyramid in the spatial decomposition is set to 4, while the low cut-off frequency of the ideal temporal bandpass filter (Section III-B) is set to 0.75 Hz, and the high component is 4.0 Hz, which correspond to 45 bpm and 240 bpm respectively.

For the training task, preparation of the training dataset is important. As introduced earlier, one feature image is extracted from a 1 s video frame of the raw dataset. However, not all the frames are qualified to be used. In some frames, the face of the subject is excluded due to large movements; an example is shown in Fig. 4.3. Some of the HR data have values outside the general human HR range due to the sensor contact problem. For our work to yield high-quality results, all these data are removed from the raw data.

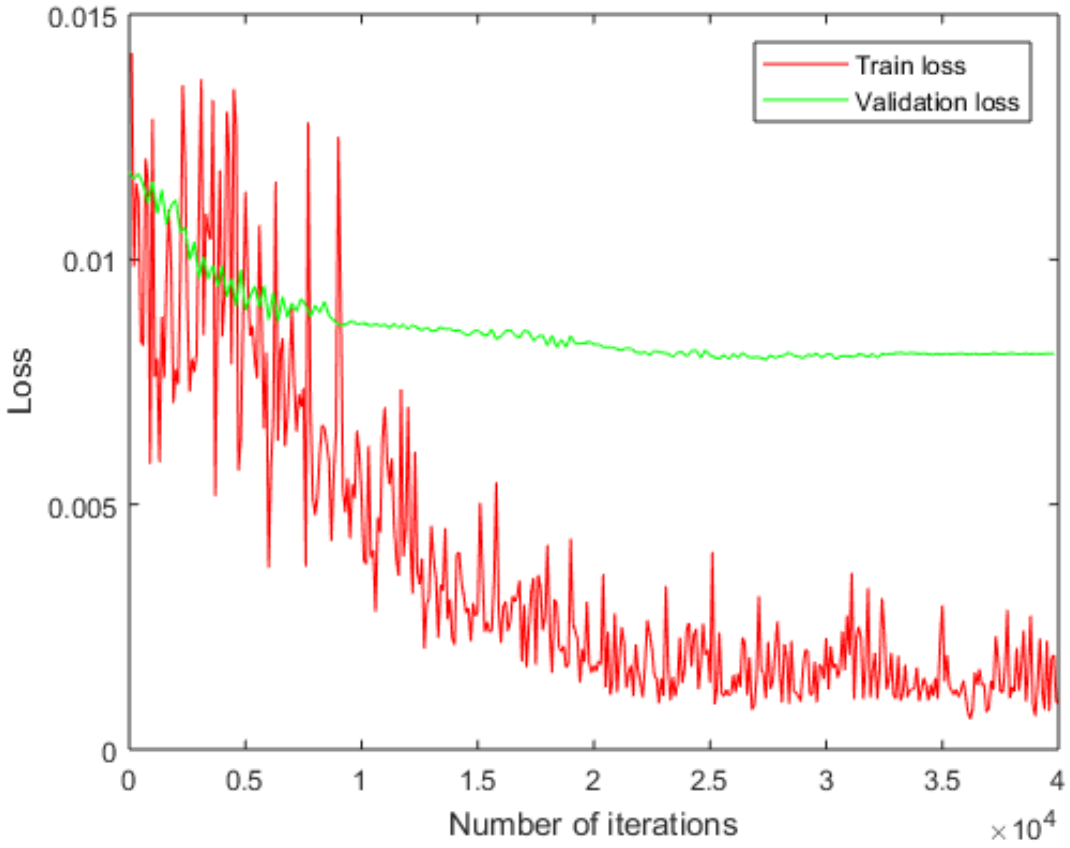


Figure 4.4: Loss curve of the shallow CNN.

Specifically, in a 1 s video frame, if there exists useless frames, all the frames of this second are removed to guarantee synchronization between the feature image and the corresponding HR. Eventually, 5839 feature images are produced after screening out the unqualified data. These images and their corresponding labels are shuffled and converted to the HDF5 format. For the training regression task in Caffe, only an HDF5 file can be used for data transformation.

For the training task, three convolutional neural networks are designed to train the model. As described in Chapter 3, a shallow CNN is designed first. Considering the efficiency, this network is optimized by using a depthwise separable convolution structure and an inverted residual with a linear bottleneck. The model size and number

of parameters of each network will be compared and discussed in the next chapter. For the regression task, Caffe does not have an accuracy layer for evaluating the validation result; therefore, the loss value is chosen as a metric to evaluate the validation result. As described in Eq. (3.2), the loss function is defined by using Euclidean distance which computes the straight line distance between the label value and predicted value in Euclidean space.

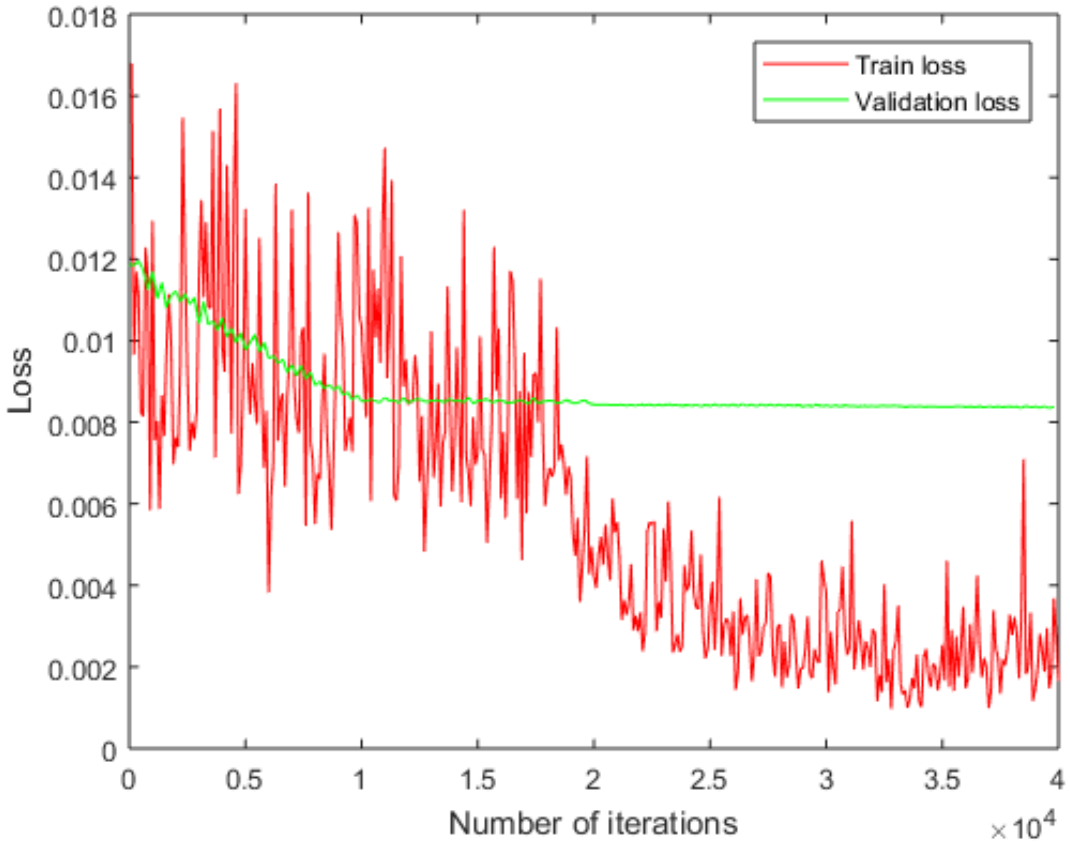


Figure 4.5: Loss curve of the CNN optimized by using the depthwise separable convolution structure.

The loss curve, which consists of every loss value, can demonstrate the convergence property and the training accuracy. The loss curves of each network during the training task are illustrated in Fig. 4.4, 4.5, and 4.6. Since all the label values are normalized to

[0, 1], the predicted values also fall within this range. According to Eq.3.2, the loss value has no unit.

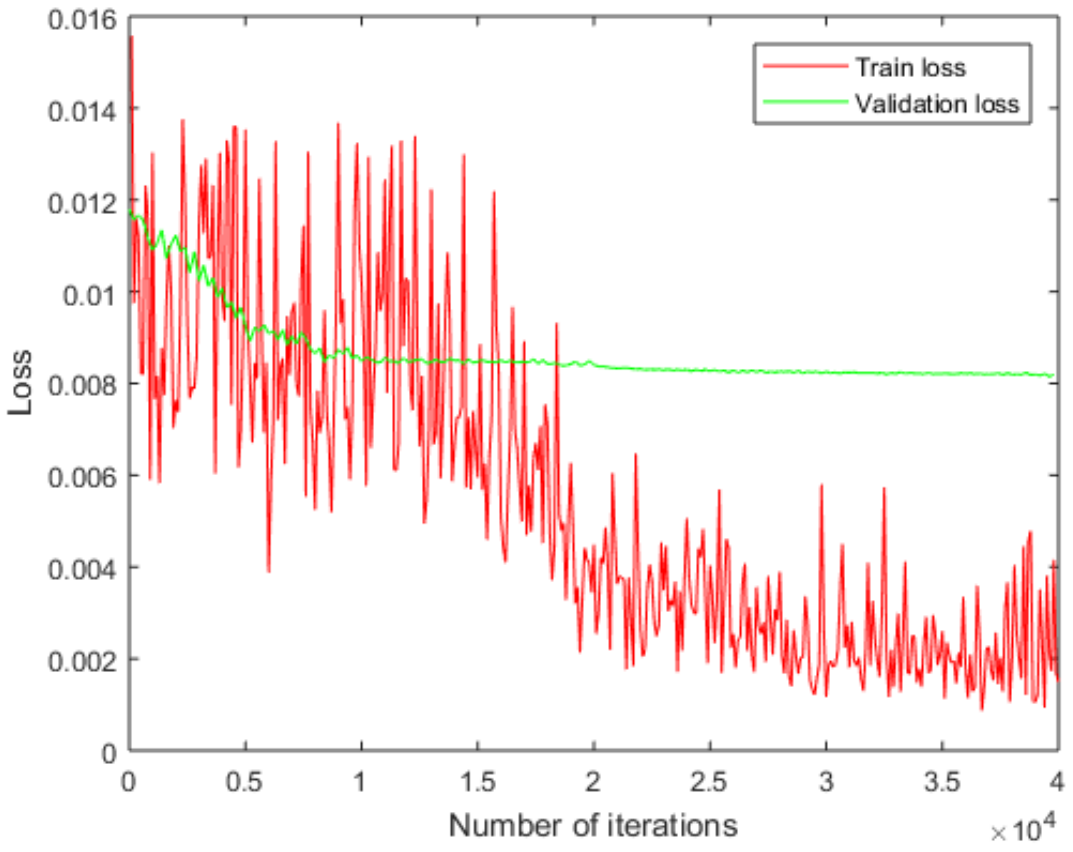


Figure 4.6: Loss curve of the CNN optimized by using an inverted residual with a linear bottleneck.

To obtain the best model with the highest accuracy, the validation loss curves of different networks are combined together to compare the performances, as shown in Fig. 4.7, where the blue curve denoted by CNN is the shallow network described in Fig. 3.5. The orange curve is the network optimized by using the depthwise separable convolution structure. The yellow curve is the network optimized by using inverted residuals with linear bottlenecks. From Fig. 4.7, it can be seen that the shallow network converges to a steady validation loss after 30000 iterations, while the other two networks converge

to a steady value after 20000 iterations, which indicates that the optimized networks converges faster than the shallow network. The results of three networks have almost the same value with the differences being less than 0.0005, and the lowest validation loss is obtained by the shallow network at around 0.0081, while the other two networks have loss values at around 0.0082 and 0.00835 respectively. Although the difference between three loss values is small, to achieve a better performance, the shallow network model with the lowest loss value will be used and analyzed in our experiments. This network will be denoted by CNN in the following chapters.

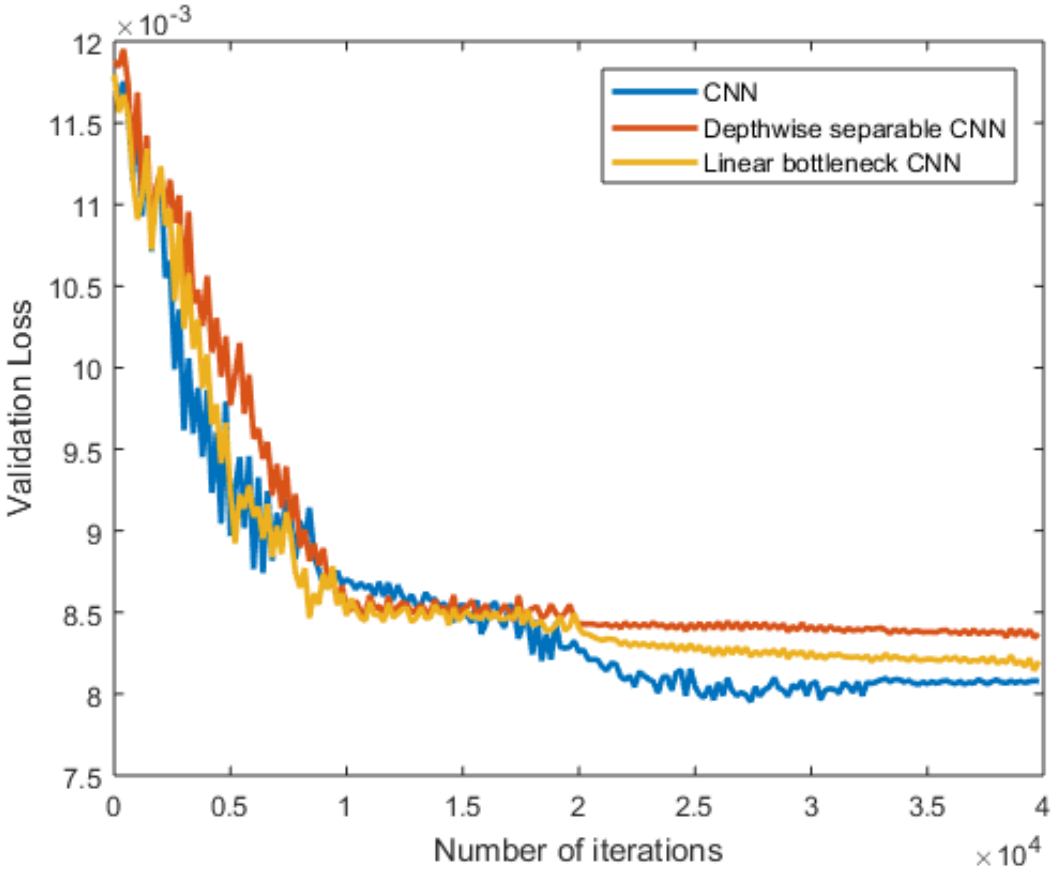


Figure 4.7: Validation loss curve comparison between three networks.

4.4 Experimental Design

In this part, three experiments used to evaluate our approach are introduced. All the experiments are conducted on the Caffe model with the lowest test loss, as introduced earlier. The performance of the CNN heart rate estimator is first evaluated on the test dataset. Then, the average HR estimation and short-term HR estimation are evaluated on the MMSE-HR dataset, and the results are compared with those of other state-of-the-art algorithms.

Experiment 1

The convolutional neural network is used for HR estimation in our method. In this experiment, the CNN is evaluated on the test dataset. As introduced earlier, 730 feature images extracted from the whole dataset are randomly chosen to be the test data, while the rest of the feature images are used in the training (4379 feature images) and validation (730 feature images) steps. The trained model is loaded to estimate the HR of the test dataset. The difference between the predicted result and the ground truth is used to evaluate and analyze the CNN estimator in next chapter.

Experiment 2

In this experiment, the proposed approach is compared with the state-of-the-art methods [57] [54] [17] and evaluated on the challenging MMSE-HR dataset [65]. Following the same process as in previous studies, all the video sequences in this dataset are used for the average HR estimation. For each video sequence, the HR at every second is predicted first by using our approach; then, the mean HR is calculated as the final value of our result. This result is evaluated and compared by using the five metrics introduced earlier.

Experiment 3

To show the ability to estimate the HR instantaneously, comparison experiments on short-term HR prediction are conducted. In these experiments, the proposed approach is compared with the same methods as those in the last experiment except for [57], which cannot estimate the HR instantaneously. Specifically, 20% of the recorded sequences, which have a very strong HR variation, are selected. Each sequence is split into nonoverlapping windows of 4, 6 and 8 seconds; then, the average HR is calculated for each nonoverlapping window. Comparison experiments are conducted independently for each window size.

4.5 Summary

As an important part of our work, the dataset is first introduced in this chapter. A challenging dataset is selected to test our system. The details regarding the data acquisition and participant diversity are given. In addition, the HR distribution is calculated and demonstrated, from which we can see that the mid-frequency band constitutes most of the whole dataset. The effect of this property will be demonstrated and discussed in the next chapter. To evaluate the proposed method comprehensively, five commonly used metrics are illustrated. Three networks, which were introduced in Chapter 3, are trained and demonstrated individually. The validation losses of the three networks are compared, and the model with the least loss is selected to be used in our experiments. Three experiments are designed to test the performance of our approach. The first experiment is designed to test the HR estimator. The other two experiments are designed to compare the performance, in terms of both average HR prediction and short-term HR prediction, with those of the state-of-the-art methods. In addition, the parameters defined in our work and the implementation details are given. The result of the experiments

will be illustrated and analyzed in the next chapter.

Chapter 5

Results and Analysis

In this chapter, to show the performance of the proposed approach, a visualization of short-term HR estimation is first demonstrated. As introduced in Chapter 4, three experiments are designed to evaluate the proposed approach. To evaluate the CNN estimator, the result of experiment 1 will be demonstrated and analyzed from different perspectives. For average HR estimation and short-term HR estimation, the results of the comparison experiments will be illustrated, and the accuracy will be analyzed based on five commonly used metrics. Additionally, the runtime of the proposed method and the efficiency of different networks are also reported.

5.1 Visualization of Short-Term HR Estimation

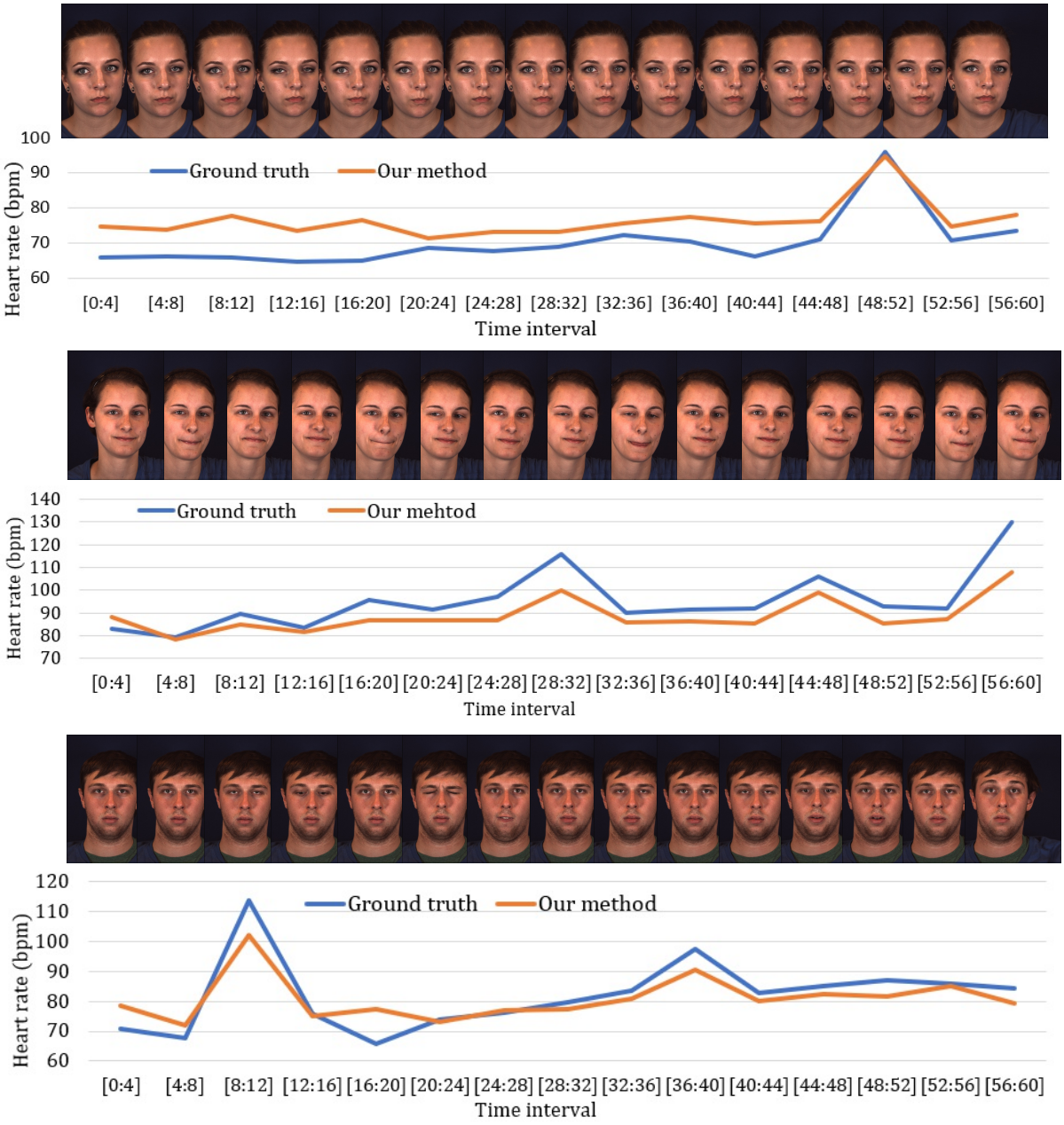


Figure 5.1: HR estimation of three sequences with a window size of 4 s. The blue line denotes the ground truth, while the orange line denotes the result predicted by using our method. The frames represent the subject's facial expression, with the corresponding HR shown below.

To show the performance of the proposed approach for short-term HR estimation,

a visualization of the processing of three challenging sequences with a window size of 4 s is demonstrated. For each video sequence, the mean of the ground truth within 4 s is calculated as the final ground truth, and one frame is selected out of every 4 s video sequence as the presentation of the subject's facial expression during that time interval, as shown in Fig. 5.1, where the horizontal axis represents the time interval, while the vertical axis represents the HR value.

Video number	Failure prediction	$\|H_e\| > 5$ *bpm*	$\|H_e\| < 5$ *bpm*
1	5	0	5
2	4	2	2
3	4	2	2

Table 5.1: Failure prediction.

From the first result in Fig. 5.1, the ground truth shows a low-frequency HR at the beginning and a sudden increase and then decrease in the HR frequency at the end. For the monotonicity of the ground truth, there are 8 increase stages and 6 decrease stages, 5 of which are falsely predicted, with all 5 changes being small, with a difference less than 5 bpm. In the second result, the ground truth shows large waves during the whole time interval; 4 changes are falsely predicted, with one of them being larger than 5 bpm. In the third result, 4 changes are again falsely predicted, with two of them being larger than 5 bpm. The number of failed predictions is presented in Table 5.1. Overall, as shown in the figure, the predicted results always have a similar trend to that of the ground truth: 69% of the HR changes are correctly predicted, while in the falsely predicted samples, 76.9% of them have differences less than 5 bpm. For these small changes, the predicted results may be influenced by the cheek muscle movements, leading to a result with large changes.

5.2 Evaluation of the CNN HR Estimator

These results are used to evaluate the CNN heart rate estimator and were obtained from experiment 1. To demonstrate the differences between the labels and the predicted values, which are defined in Eq. (3.2), the error is calculated for each sample. Additionally, to evaluate the accuracy of the estimator for different frequency bands, the test data are divided into three bands according to the HR distribution in the total dataset (Fig. 4.2). The range of $[45, 65)$ is defined as the low-frequency (LF) band; the range of $[65, 95)$ is defined as the mid-frequency (MF) band and the range of $[95, 185]$ is defined as the high-frequency (HF) band. The number of samples of each part is 60, 541 and 129, respectively. The result of this experiment is shown in Fig. 5.2, where the LF, MF, and HF parts are represented by an orange, a gray and a yellow bar, respectively, while H_e is denoted by a blue bar.

Frequency part	$\lvert H_e \rvert < 15$ *bpm*	$H_e < -15$ *bpm*	$H_e \geq 15$ *bpm*
LF	2.4%	0	58.8%
MF	91.5%	12%	41.2%
HF	6.1%	88%	0

Table 5.2: HR error proportion for each frequency part.

As shown in Fig. 5.2, the errors between the labels and predicted values are distributed in 9 sections, which is demonstrated along the horizontal axis, while the vertical axis represents the proportion. The mean of H_e is -1.25, while the variance is 311.25. There are 541 samples having an absolute H_e within 15 bpm, which indicates that 74.13% of the test data are well estimated. In this range, as shown in Fig. 5.2, most of the samples are from the MF band, which constitutes 91.5%, while the LF and HF bands constitute 2.4% and 6.1%, respectively. In contrast, regarding an absolute H_e greater than 15, most

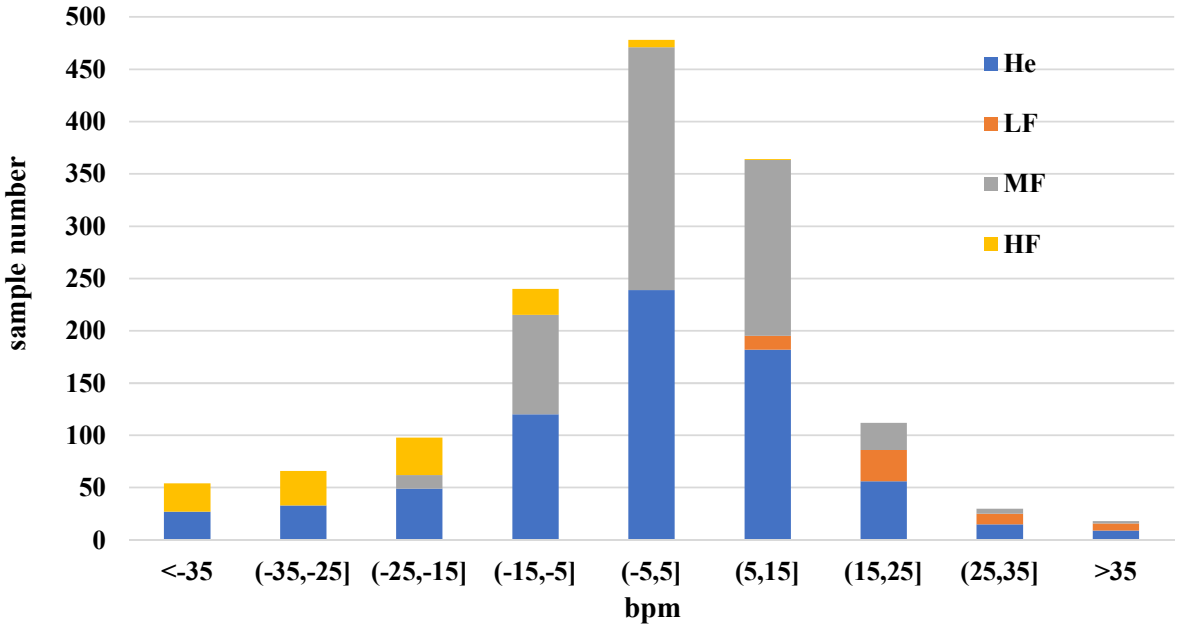


Figure 5.2: HR error distribution on the test dataset. The horizontal axis denotes the range of H_e, while the vertical axis denotes the number of samples. The blue bar represents H_e, the orange bar represents the LF band (45-65 bpm), the gray bar represents the MF band (65-95 bpm) and the yellow bar represents the HF band (95-185 bpm).

of the samples are from the either HF or LF band, which means that for $H_e < -15$, the HF band constitutes 88%, while the MF band constitutes 12%. For $H_e \geq 15$, the LF band constitutes 58.8%, while the MF band constitutes 41.2%. For each frequency band, the proportion of different H_e ranges is presented in Table 5.2. This phenomenon is due to the scarcity of the LF and HF samples in the training data, which leads to lower accuracy in these two bands. In terms of the LF band, 78.3% of the predictions are much higher than the ground truth when $He > 15$. Regarding the HF band, 74.4% of the predictions are much lower than the ground truth when $H_e < -15$. However, only 8.5% of the MF band is predicted with lower accuracy when $|H_e| > 15$. Comparing the percentages of each band predicted with higher HR error, it is shown that if a certain frequency band constitutes a larger proportion of the dataset, then the predicted result of this band has higher accuracy than that of others, which also indicates that the diversity

of the HR dataset is important.

5.3 Average HR Prediction on MMSE-HR Dataset

These results are obtained from experiment 2 and are used to compare the performance of our approach in terms of average HR prediction with that of other methods on the same dataset. The results are evaluated by using the five metrics discussed earlier, which were commonly used in previous studies. The results are shown in Table 5.3.

Method	$M_e(SD_e)$	$RMSE$	M_{eRate}	ρ
Li et al.[57]	11.56 (20.02)	19.95	14.64%	0.38
Haan et al.[17]	9.41 (14.08)	13.97	12.22%	0.55
Tulyakov et al.[54]	7.61 (12.24)	11.37	10.84%	0.71
Ours (CNN)	-1.17 (**6.85**)	**6.95**	**6.55**%	**0.98**

Table 5.3: Average HR prediction.

As shown in Table 5.3, compared with that of the previous methods, the SD_e of our result is reduced to 6.85, which shows that the HR error tends to be closer to the mean value. The $RMSE$ is reduced to 6.95, indicating that the magnitudes of the errors in the predictions are decreased. M_{eRate}, demonstrating the prediction accuracy, is improved to 6.55% by using our approach. In addition, ρ is improved to 0.98, which is closer to 1, meaning that the linear correlation between the ground truth and predicted HR is stronger. Overall, on all aspects, our approach performs better than the previous methods in terms of average HR prediction. Comparing each metric value among all the methods, it is shown that as ρ increases, all the other values decrease.

5.4 Short-Term HR Estimation

These results are taken from experiment 3 and are used to evaluate the performance in terms of short-term HR estimation and to compare our approach with other methods. The five metrics are also used here to evaluate the results, as shown in Table 5.4, 5.5, and 5.6.

Method	$M_e(SD_e)$	$RMSE$	M_{eRate}	ρ
Haan et al.[17]	-1.85 (15.77)	15.83	9.92%	0.67
Tulyakov et al.[54]	2.12 (11.51)	11.66	9.15%	0.78
Ours (CNN)	-1.31 (**8.19**)	**8.30**	**6.93**%	**0.93**

Table 5.4: Short-term HR prediction with window size of 4 s.

Method	$M_e(SD_e)$	$RMSE$	M_{eRate}	ρ
Haan et al.[17]	-2.21 (19.21)	19.27	11.81%	0.33
Tulyakov et al.[54]	0.32 (8.29)	8.27	7.30%	0.80
Ours (CNN)	-1.29 (**7.53**)	**7.64**	**6.74**%	**0.95**

Table 5.5: Short-term HR prediction with window size of 6 s.

As shown in the table, compared with the other methods, for each different window size, SD_e, $RMSE$ and M_{eRate} of our approach are all less than those of the previous methods, which indicates that the variations in predicted HR are distributed in a narrower range with a higher accuracy. ρ is also improved by our approach for each window size, which shows a stronger linear correlation between the predicted values and the ground truth. Regarding Haan's [17] method, it has the worst performance under a window size

Method	$M_e(SD_e)$	$RMSE$	M_{eRate}	ρ
Haan et al.[17]	0.81 (11.49)	11.46	8.60%	0.63
Tulyakov et al.[54]	1.62 (9.67)	9.76	7.52%	0.71
Ours (CNN)	-1.24 (**7.24**)	**7.34**	**6.58%**	**0.96**

Table 5.6: Short-term HR prediction with window size of 8 s.

of 6 s, with ρ equal to 0.33. Conversely, Tulyakov's [54] method performs best under a window size of 6 s compared to the other results. However, from the results of our method for different window sizes, it can be seen that as the window size increases, SD_e, $RMSE$ and M_{eRate} all decrease, while ρ increases, which shows that the performance our method steadily improves in all aspects. Since our approach is to estimate the HR every second and then calculate the mean value for a predefined time interval, as the time interval increases, the accuracy increases.

5.5 Runtime

To calculate the runtime, the entire framework is divided into two parts. The first part includes face detection, tracking, ROI cropping and feature image extraction. The second part is the HR estimator. Part 1 runs at 110 fps, while part 2 has different speeds for different networks, as shown in Table 5.7. The runtimes are measured using a conventional laptop with an Intel Core i5-7300HQ CPU and 8.0 GB of RAM. Thus, no matter which network is used, the proposed approach runs fast enough to be used for real-time HR estimation.

In the table, "Params" denotes the number of parameters in the convolutional neural network, and "MAdds" denotes the multiply-adds of the network. "CNN" represents the shallow convolutional network, "DWCNN" represents the CNN optimized by using the

Network	Params	MAdds	Speed
CNN	595.3K	24.41M	290 fps
DWCNN	104.19K	11.51M	197 fps
LBCNN	75.87K	2.21M	448 fps

Table 5.7: Comparison of the performances of different networks on the test dataset.

depthwise separable convolution structure and "LBCNN" represents the CNN optimized by using an inverted residual with a linear bottleneck. From the table, it can be seen that the CNN has the largest number of parameters and multiply-adds, while the LBCNN has the smallest numbers of these two factors. The DWCNN has medium values for both parameters. These results indicate that the depthwise separable convolution structure helps reduce the model size and computational complexity; however, the number of frames that can be processed within one second decreases. The inverted residual with a linear bottleneck achieves the most efficient model and highest speed. Therefore, if the model size and efficiency are regarded as important factors in an HR estimation program, the LBCNN should be considered. For higher accuracy, the CNN is the best choice.

5.6 Further Discussion

As introduced before, the proposed method is based on the ROI extraction, however, the face detection has limitations in some situations. In realistic conditions, subject's head can rotate randomly in front of the camera, while in the proposed approach, only faces that rotate within a certain angle range can be detected. Specifically, the subject's head can rotate around yaw from -90 degree to 90 degree, around roll from -20 degree to 20 degree and around pitch from -20 degree to 20 degree. For the positions that 68 landmarks are not identified, there is no ROI extracted and no corresponding HR

estimated. The whole process only works when consecutive faces are detected within 1 s.

We also evaluate the proposed method on the MAHNOB-HCI dataset [36]. There are 27 participants (12 males and 15 females) involved, 527 videos are recorded and can be used totally. In our experiments, same as [54], 30 seconds interval (frame 306 to 2135) is extracted from each video sequence. The second channel (EXG2) of ECG signal is used to obtain the ground truth. Half of the video sequences are randomly chosen to extract the feature images and trained. A comparison result is shown in Table. 5.8, where all the video sequences are tested and the comparison results for [37, 38, 16, 57, 17, 54] are taken from [54]. Apparently, the proposed approach also shows better performance on the MAHNOB-HCI dataset.

Method	$M_e(SD_e)$	$RMSE$	M_{eRate}	ρ
Poh *et al.* [37]	-8.95(24.3)	25.9	25.0%	0.08
Poh *et al.* [38]	2.04(13.5)	13.6	13.2%	0.36
Balakrishnan *et al.* [16]	-14.4(15.2)	21.0	20.7%	0.11
Li *et al.* [57]	-3.30(6.88)	7.62	6.87%	0.81
Haan *et al.* [17]	4.62(6.50)	6.52	6.39%	0.82
Tulyakov *et al.* [54]	3.19(5.81)	6.23	5.93%	0.83
Ours	-1.68(**2.79**)	**3.26**	**3.67**%	**0.95**

Table 5.8: Average HR Prediction: comparison results on MAHNOB-HCI dataset

5.7 Summary

The results of three experiments are demonstrated and analyzed in this chapter. Regarding the result of experiment 1, the CNN estimator is capable of estimating most of the test data within a reasonable error range. However, the accuracy is related to the HR proportion of different frequency ranges in the training dataset. For experiments 2 and 3, the proposed approach performs better than the other methods in terms of average HR estimation and short-term HR estimation. Particularly, the accuracy of our approach increases steadily as the window size increases for short-term HR estimation, while the other approaches exhibit unstable performances as the window size increases. The runtime of the proposed approach is reported via two parts. For the network part, the runtime is demonstrated for different network structures.

Chapter 6

Conclusions

6.1 Conclusions

In this thesis, a new framework is introduced for contactless HR estimation from facial videos under realistic conditions. Different from the traditional HR estimation approach, which usually extracts a signal related to an HR and uses power spectrum density analysis to estimate the average HR, a convolutional neural network is used to estimate the HR in the proposed approach. Instead of applying a series of filters to clean the underlying signal, the HR is directly estimated from a feature image that is

obtained by using spatial decomposition and temporal filtering, which decreases both the computational complexity and the processing time.

The results of testing the trained model on the testing dataset show that 74.13% of the data in the testing dataset are well estimated, while the remaining data have large errors, which is due to the lack of high- and low-frequency components in the dataset. In addition, comparison experiments are conducted on the MMSE-HR dataset for both average HR estimation and short-term HR estimation. The results show that the proposed approach achieves higher accuracy than that of the other methods.

6.2 Future work

Our approach can be improved in the future. For instance, the HR diversity of the dataset can be improved. As shown in Fig. 4.2, the low-frequency and high-frequency parts constitute no more than 30% of the data, which affects the estimation performance on these two parts. Therefore, a larger dataset with well-proportioned HR data should be collected in future work.

Second, some additional physiological information can be estimated via a similar procedure, and a combination analysis can be added to facilitate emotion prediction. For emotion analysis, HRV(Heart Rate Variability) is also needed to help recognize emotion changes and which kind of emotion the subject might be aroused.

Another part that can be improved is the neural network. A convolutional neural network is applied in this thesis; however, other types of neural networks may also be applicable to this problem. Since the proposed approach can estimate instantaneous HR, a robust system can be developed for real-life HR estimation. However, because of the complexity of realistic conditions, such as the camera property, the lights of the environments, multi-subjects detection and so on, the system can be improved to tackle these problems. For portable purpose, an embedded system and mobile application can

be developed in the future.

References

[1] A. Asthana and S. Zafeiriou and S. Cheng and M. Pantic. "Robust discriminative response map fitting with constrained local models". In: *the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2013, pp. 3444–3451.

[2] A. Lam and Y. Kuno. "Robust heart rate measurement from video using select random patches". In: *the Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 3640–3648.

[3] A. Malliani and F. Lombardi and M. Pagani. "Power spectrum analysis of heart rate variability: a tool to explore neural regulatory mechanisms." In: *British heart journal* 71.1 (1994), p. 1.

[4] A. S. Montero and J. Lang and R. Laganiere. "Scalable Kernel Correlation Filter with Sparse Feature Integration." In: *the Proceesings of the IEEE International Conference on Computer Vision Workshop*. 2015, pp. 587–594.

[5] J. Allen. "Photoplethysmography and its application in clinical Physiological measurement". In: *Physiological Measurement* 28.3 (2007), R1–R39.

[6] C. Li and C. Xu and C. Gui and M. Fox. "Distance regularized level set evolution and its application to image segmentation". In: *IEEE Transactions on Image Processing* 19.12 (2010), pp. 3243–3254.

[7] C. Liu and A. Torralba and W. T. Freeman and F. Durand and E. H. Adelson. "Motion magnification". In: *ACM transactions on graphics (TOG)* 24.3 (2005), pp. 519–526.

[8] C. Tomasi and T. Kanade. *Detection and Tracking of Point Features*. Tech. rep. International Journal of Computer Vision, 1991.

[9] D. Datcu and M. Cidota and S. Lukosch and L. Rothkrantz. "Noncontact automatic heart rate analysis in visible spectrum by specific face regions". In: *Proceedings of the 14th International Conference on Computer Systems and Technologies*. ACM. 2013, pp. 120–127.

[10] D. McDuff and S. Gontarek and R. Picard. "Remote measurement of cognitive stress via heart rate variability". In: *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*. IEEE. 2014, pp. 2957–2960.

[11] D. McDuff and S. Gontarek and R. W. Picard. "Improvements in remote cardiopulmonary measurement using a five band digital camera". In: *IEEE Transactions on Biomedical Engineering* 61.10 (2014), pp. 2593–2601.

[12] D. Pagliari and L. Pinto. "Calibration of kinect for xbox one and comparison between the two generations of microsoft sensors". In: *Sensors* 15.11 (2015), pp. 27569–27589.

[13] D. Trichopoulos and X. Zavitsanos and K. Katsouyanni and A. Tzonou and P. Dalla-Vorgia. "Psychological stress and fatal heart attack: the Athens (1981) earthquake natural experiment". In: *The Lancet* 321.8322 (1983), pp. 441–444.

[14] E. H. Hon, and ST. Lee. "Electronic evaluation of the fetal heart rate. VIII. Patterns preceding fetal death, further observations." In: *American journal of obstetrics and gynecology* 87 (1963), pp. 814–826.

[15] E. Lachat and H. Macher and MA. Mittet and T. Landes and P. Grussenmeyer. "First experiences with Kinect v2 sensor for close range 3D modelling". In: *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 40.5 (2015), p. 93.

[16] G. Balakrishnan and F. Durand and J. Guttag. "Detecting pulse from head motions in video". In: *the Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2013, pp. 3430–3437.

[17] G. De Haan and V. Jeanne. "Robust pulse rate from chrominance-based rPPG". In: *IEEE Transactions on Biomedical Engineering* 60.10 (2013), pp. 2878–2886.

[18] G. R. Elliott, "Stress and human health. Analysis and implications of research". In: *A study by the Institute of Medicine, National Academy of Sciences* (1982).

[19] Greneker, EF. "Radar sensing of heartbeat and respiration at a distance with applications of the technology". In: (1997).

[20] H. Osman and M. Eid and A. Saddik. "U-biofeedback: a multimedia-based reference model for ubiquitous biofeedback systems". In: *Multimedia tools and applications* 72.3 (2014), pp. 3143–3168.

[21] H. Simon. "Adaptive filter theory". In: *Prentice Hall* 2 (2002), pp. 478–481.

[22] Howard. A. G. and Zhu. M. and Chen. B. and Kalenichenko. D. and Wang. W. and Weyand. T. and Andreetto. M. and Adam. H. "Mobilenets: Efficient convolutional neural networks for mobile vision applications". In: *ArXiv Preprint ArXiv:1704.04861* (2017).

[23] H.-Y. Wu and M. Rubinstein and E. Shih and J. Guttag, F. Durand and W. Freeman. “Eulerian video magnification for revealing subtle changes in the world”. In: *ACM Transactions on Graphics*. 2012, 65:1–65:8.

[24] J. Cardoso. “High-order contrasts for independent component analysis”. In: *Neural computation* 11.1 (1999), pp. 157–192.

[25] JPA. Delaney and DA. Brodie. “Effects of short-term psychological stress on the time and frequency domains of heart-rate variability”. In: *Perceptual and motor skills* 91.2 (2000), pp. 515–524.

[26] K. Alghoul and S. Alharthi and H. Osman and A. Saddik. “Heart Rate Variability extraction from videos signals: ICA vs. EVM comparison”. In: *IEEE Access* 5 (2017), pp. 4711–4719.

[27] K. He and X. Zhang and S. Ren and J. Sun. “Deep residual learning for image recognition”. In: *the Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[28] K. O’Shea and R. Nash. “An introduction to convolutional neural networks”. In: *arXiv preprint arXiv:1511.08458* (2015).

[29] K. Zhang and Z. Zhang and Z. Li and Y. Qiao. “Joint face detection and alignment using multitask cascaded convolutional networks”. In: *IEEE Signal Processing Letters* 23.10 (2016), pp. 1499–1503.

[30] L. A. Aarts and V. Jeanne and J. P. Cleary and C. Lieber and J. S. Nelson and S. B. Oetomo and W. Verkruysse. “Non-contact heart rate monitoring utilizing camera photoplethysmography in the neonatal intensive care unit—a pilot study”. In: *Early human development* 89.12 (2013), pp. 943–948.

[31] L. Bernardi and J. Wdowczyk-Szulc and C. Valenti and S. Castoldi, and C. Passino and G. Spadacini and P. Sleight. “Effects of controlled breathing, mental activity

and mental stress with or without verbalization on heart rate variability". In: *Journal of the American College of Cardiology* 35.6 (2000), pp. 1462–1469.

[32] L. Salahuddin and D. Kim. "Detection of acute stress by heart rate variability (HRV) using a prototype mobile ECG sensor". In: *Proceedings of the International Conference on Hybrid Information Technology, Cheju Island, Korea.* 2006, pp. 9–11.

[33] M. A. Hassan and G. S. Malik and N. Saad and B. Karasfi and Y. S. Ali and D. Fofi. "Optimal source selection for image photoplethysmography". In: *the Proceedings of IEEE International Conference on Instrumentation and Measurement Technology.* 2016, pp. 1–5.

[34] M. Garbey and N. Sun and A. Merla and I. Pavlidis. "Contact-free measurement of cardiac pulse based on the analysis of thermal imagery". In: *IEEE transactions on Biomedical Engineering* 54.8 (2007), pp. 1418–1426.

[35] M. Sandler and A. Howard and M. Zhu and A. Zhmoginov and L. Chen. "Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation". In: *arXiv preprint arXiv:1801.04381* (2018).

[36] M. Soleymani and J. Lichtenauer and T. Pun and M. Pantic. "A multimodal database for affect recognition and implicit tagging". In: *IEEE Transactions on Affective Computing* 3.1 (2012), pp. 42–55.

[37] M.-Z. Poh and D. J. McDuff and R. W. Picard. "Non-contact, automated cardiac pulse measurements using video imaging and blind source separation." In: *Optics Express* 18.10 (2010), pp. 10762–10774.

[38] M.-Z. Poh and D. J. McDuff and R.W. Picard. "Advancements in noncontact, multiparameter physiological measurements using a webcam". In: *IEEE Transactions on Biomedical Engineering* 58.1 (2011), pp. 7–11.

[39] N. Hjortskov and D. Rissén and A. K. Blangsted and N. Fallentin and U. Lundberg and K. Søgaard. "The effect of mental stress on heart rate variability and blood pressure during computer work". In: *European journal of applied physiology* 92.1-2 (2004), pp. 84–89.

[40] N. Miljković and D. Trifunović. "Pulse rate assessment: Eulerian video magnification vs. electrocardiography recordings". In: *Neural Network Applications in Electrical Engineering (NEUREL), 2014 12th Symposium on*. IEEE. 2014, pp. 17–20.

[41] N. Srivastava and G. Hinton and A. Krizhevsky and I. Sutskever and R. Salakhutdinov. "Dropout: A simple way to prevent neural networks from overfitting". In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958.

[42] P. B. Chambino. "Android-based implementation of Eulerian Video Magnification for vital signs monitoring". In: (2013).

[43] P. Comon. "Independent component analysis, a new concept?" In: *Signal processing* 36.3 (1994), pp. 287–314.

[44] P. Viola and M. Jones. "Rapid object detection using a boosted cascade of simple features". In: *the Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 1. IEEE. 2001, pp. I–I.

[45] R. Lienhart and J. Maydt. "An extended set of haar-like features for rapid object detection". In: *the Proceedings of International Conference on Image Processing*. Vol. 1. IEEE. 2002, pp. I–I.

[46] S. Akselrod and D. Gordon and F. A. Ubel and D. C. Shannon and AC. Berger and R. J. Cohen. "Power spectrum analysis of heart rate fluctuation: a quantitative probe of beat-to-beat cardiovascular control". In: *science* 213.4504 (1981), pp. 220–222.

[47] S. Arora and J. Bhattacharjee, "Modulation of immune responses in stress by Yoga". In: *International journal of yoga* 1.2 (2008), p. 45.

[48] S. Bakhtiari and T. W. Elmer and N. M. Cox and N. Gopalsami and A. C. Raptis and S. Liao and I. Mikhelson and A. V. Sahakian. "Compact millimeter-wave sensor for remote monitoring of vital signs". In: *IEEE Transactions on Instrumentation and Measurement* 61.3 (2012), pp. 830–841.

[49] S. C. Segerstrom and G. E. Miller. "Psychological stress and the human immune system: a meta-analytic study of 30 years of inquiry." In: *Psychological bulletin* 130.4 (2004), p. 601.

[50] S. Cerutti and A. L. Goldberger and Y. Yamamoto. "Recent advances in heart rate variability signal processing and interpretation". In: *IEEE Transactions on Biomedical Engineering* 53.1 (2006), pp. 1–3.

[51] S. Mallat. *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. 3rd. Academic Press, 2008.

[52] S. Prakash and C. S. Tucker. "Bounded Kalman filter method for motion-robust, non-contact heart rate estimation". In: *Biomedical Optics Express* 9.2 (2018), pp. 873–897.

[53] S. Ren and X. Cao and Y. Wei and J. Sun. "Face alignment at 3000 fps via regressing local binary features". In: *the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1685–1692.

[54] S. Tulyakov and X. Alameda-Pineda and E. Ricci and L. Yin and J. Cohn and N. Sebe. "Self-adaptive matrix completion for heart rate estimation from face videos under realistic conditions". In: *the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 2396–2404.

[55] T. Zhang and W. Zheng and Z. Cui and Y. Zong and J. Yan and K. Yan. "A deep neural network-driven feature learning method for multi-view facial expression recognition". In: *IEEE Transactions on Multimedia* 18.12 (2016), pp. 2528–2536.

[56] W. Verkruysse and L. O. Svaasand and J. S. Nelson. "Remote plethysmographic imaging using ambient light." In: *Optics Express* 16.26 (2008), pp. 21434–21445.

[57] X. Li and J. Chen and G. Zhao and M. Pietikainen. "Remote heart rate measurement from face videos under realistic situations". In: *the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 4264–4271.

[58] X. Lu and Z. Lin and H. Jin and J. Yang and J.Z. Wang. "Rating image aesthetics using deep learning". In: *IEEE Transactions on Multimedia* 17.11 (2015), pp. 2021–2034.

[59] X. Yu and J. Huang and S. Zhang and W. Yan and D. Metaxas. "Pose-free facial landmark fitting via optimized part mixtures and cascaded deformable shape model". In: *the Proceedings of the IEEE International Conference on Computer Vision*. IEEE. 2013, pp. 1944–1951.

[60] Y. Chen and C. Gene and S. Vladimir. "Estimating heart rate and rhythm via 3D motion tracking in depth video". In: *IEEE Transactions on Multimedia* 19.7 (2017), pp. 1625–1636.

[61] Y. LeCun and B. Boser and J. Denker and D. Henderson and R. Howard and W. Hubbard and L. Jackel. "Handwritten digit recognition with a back-propagation network". In: *the Proceedings of Advances in neural information processing systems*. 1990, pp. 396–404.

[62] Y. LeCun and others. "Generalization and network design strategies". In: *Connectionism in perspective* (1989), pp. 143–155.

[63] Y. LeCun and Y. Bengio and others. “Convolutional networks for images, speech, and time series”. In: *The handbook of brain theory and neural networks* 3361.10 (1995), p. 1995.

[64] Z. Wu and N. Huang and S. Long and C. Peng. “On the trend, detrending, and variability of nonlinear and nonstationary time series”. In: *Proceedings of the National Academy of Sciences* 104.38 (2007), pp. 14889–14894.

[65] Z. Zhang and J. Girard and Y. Wu and X. Zhang and P. Liu and U. Ciftci and S. Canavan and M. Reale and A. Horowitz and H. Yang and J. F. Cohn and Q. Ji and L. Yin. “Multimodal spontaneous emotion corpus for human behavior analysis”. In: *the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 3438–3446.