



## BE (IoT) Degree Program

### Stage 4 Project Group Thesis Receipt

Project Title:


Supervisor:

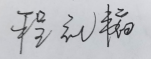
Student Name(s)	UCD Student Number	BJUT Student Number
LiXiang(李响)	17206022	17371214
ChengLitao(程礼韬)	17206018	17371230
YuanXiaoran(原潇然)	17205911	17371221

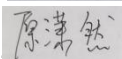
**Plagiarism:** the unacknowledged inclusion of another person's writings or ideas or formally presented work (including essays, examinations, projects, laboratory presentations). The penalties associated with plagiarism designed to impose sanctions seriousness of University's commitment to academic integrity. Ensure that you have read the University's **Briefing for Students on Academic Integrity and Plagiarism** and the UCD **Statement, Plagiarism Policy and Procedures**, (<http://www.ucd.ie/registrar/>)

#### Declaration of Authorship

- 1) I/we declare that all of the following are true:
  - 2) I/we fully understand the definition of plagiarism.
  - 3) I/we have not plagiarized any part of this project and it is my original work.
- All material in this report is my/our own work except where there is clear acknowledgement and appropriate reference to the work of others.

Signed.......... Date .....2021.5.27.....

Signed .......... Date .....2021.5.27.....

Signed.......... Date .....2021.5.27.....

Office Use Only

#### Report Tracking

Date and Time Received:

N/A


---

## Contents

Abstract.....	3
1.Introduction.....	4
1.1 Research background.....	4
1.2 Research purposes.....	4
1.3.Technical routes & contributions.....	5
2Literature review.....	6
2.1 Region of interest (ROI) Tracking.....	6
2.2 Colour space selection.....	6
2.3 Heart rate detection methodology.....	7
3. Lixiang.....	10
3.1 Data set description.....	10
3.2 face interest area tracking.....	11
3.2.1 Related toolbox.....	11
3.2.2 ROI detected algorithm.....	12
3.2.3 Compare different ROI.....	12
3.3 feature extraction.....	14
3.3.1 Different color space comparison.....	14
3.3.2 calculate 1D HR signal from 2D image.....	16
3.3.3 conclusion.....	17
3.4 GUI prototype.....	17
4. Chenglitao.....	19
4.1 Traditional method(signal process).....	19
4.1.1 Power spectral density (PSD) with filter.....	20
4.1.2 Singular Spectrum Analysis.....	22
4.1.3 Wavelet filter.....	26
4.2 1D-machine learning network.....	29
4.2.1 CorNet (LSTM/CNN).....	29
5.Yuan xiaoran.....	34
5.1 Modified Cornet.....	34
5.2 2D-machine learning network.....	39
5.2.1 ResNet.....	41
5.2.2 Inception v3.....	41
5.2.3 Inception-ResNet v2 .....	43
6 Conclusion and future work.....	44
6.1Conclusion.....	44
6.2Further work.....	45
Acknowledgment.....	46
References.....	47

---

# Abstract

In a video signal, the subtle change in human skin can be detected by the computer, then physiological signals can be determined. This technology is known as rppg (remote Photoplethysmography). With the help of Professor John and Dr. Zhao's team in Wuhan, we used the database named MANHOB to verify different detect different methods.

In this paper, we discussed three topics around this technology, including the extraction of physiological signals from the face, traditional signal processing methods to detect heart rate, and machine learning methods to detect heart rate. By signal processing approach, we find that Luv is the best color space among all. For ROI selection, we find that it is not that important as we thought. In general, a smaller ROI contains a strong signal with more noise and a larger ROI contains less signal with less noise, but all ROI on the face includes a heart rate signal. In the traditional processing method part, we found that wavelet transform performs best among three methods, which reach MAE of 7.3 bpm in the whole data set.

Except for the traditional method, we also employ advanced deep learning methods to improved performance. Taking advantage of our preprocessed 1D IPPG signal using the traditional method, we intend to build a suitable Neural Network to generate heart rate from this 1D signal. Two approaches are illustrated in our report. One is built according to literature, using 1D-CNN and LSTM, the other one is proposed by Dr. Liang and modified and tuned by us to further improve the performance. We also put effort to explore the frequency feature of IPPG signal. In the final time, we use STFT(Short Time Fourier Transform ) to generate spectrogram for 1D IPPG signal and then experiment with a few classical 2D CNN networks like ResNet, Inception network.

Key words: rppg, heartrate detection, wavelet transform, Cornet, Inception

---

# 1. Introduction

## 1.1 Research background

Heart rate is one of the most important physiological signals used by the medical profession to determine the physiological state of the human body. Nowadays, one of the gold standard techniques of measuring the cardiac pulse is electrocardiogram (ECG) which requires patients to wear adhesive gel patches or chest straps. However, this traditional approach may cause skin irritation and slight pain. That is the motivation of discovering a remote sensing technology which requires less equipment, doesn't risk any cross-contamination and can be used for mass screening etc.

The major remote sensing technology now is Photoplethysmography (PPG). Subtle changes in human skin can reveal crucial hidden signal indicating the physical signal which cannot be detected by eyes. However, computer analysis of video and images can reveal that subtle signal. PPG is a non-contact, straightforward, inexpensive optical technology for measuring clinical parameters such as blood oxygen saturation, heart rate and cardiac output. The principle of PPG is that blood absorbs more light than ambient tissue, so changes in blood volume will affect transmittance or reflectance accordingly. PPG as a remote detection method has some obvious advantages that contact measurement do not have. On one hand, with non-contact methods, HR monitoring of the patients without causing an infection will be possible. On the other hand, non-contact methods can monitor cardiac signal in a more comfort way, therefore, make long-term monitoring a more feasible way. In addition, PPG can also be applied in contact HR measurement. The advantage of using PPG in a short distance can amplify the PPG signal reflected from skin and therefore result in a better prediction.

## 1.2 Research purposes

The whole project can be described as a biosignal processing problem, and we try to detect heartbeats from videos using signal processing and machine learning methods. Heart rate detection can be divided into two stages. In the first stage, the video is processed frame by frame using signal processing methods to obtain a one-dimension heart rate signal with less interference; in the second stage, the one-dimension signal is processed using machine learning methods to obtain the heart rate value of the experimental subject. It is also necessary to develop a real-time heart rate detection software demo to help us identify problems in real situations.

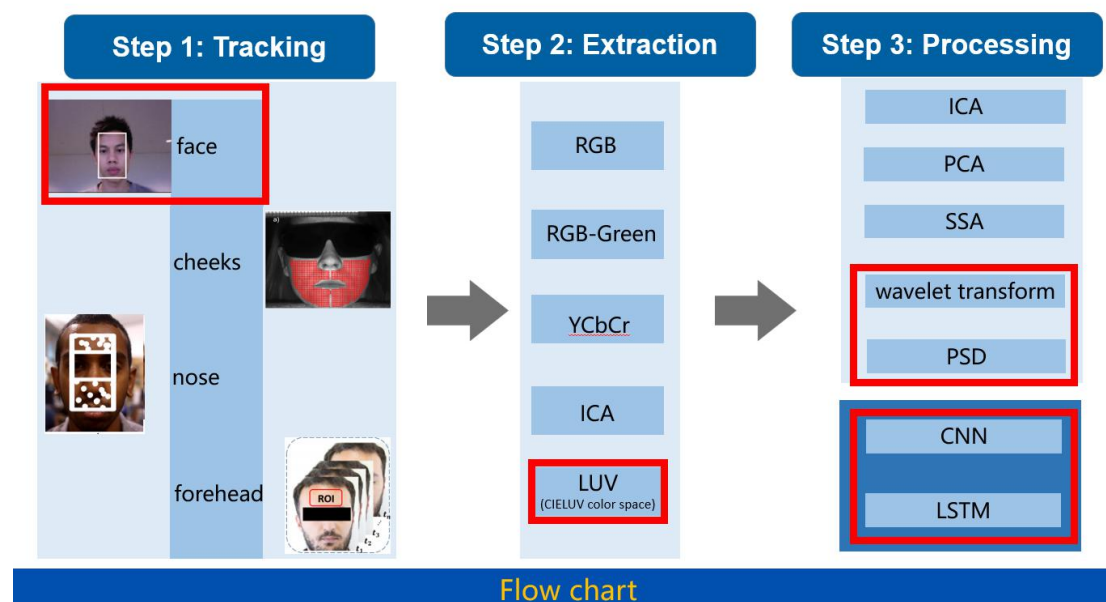
In summary, the purposes of this project can be shown below:

1. implement and compare different image processing methods
2. implement and compare different machine learning methods
3. Develop software for real-time heartbeat detection

### 1.3. Technical routes & contributions

To achieve the purposes demonstrated above, after reading and analyzing corresponding literature, we found that there are two main stages in image processing methods, including face tracking and color space selection (Step1 and Step2 in fig1). Face interest areas contain face, cheek, nose, forehead, and color spaces are RGB, GREEN, and LUV. After gaining the one-dimension signal, we should implement either traditional signal processing methods or machine methods to get the estimated heart rate values, such as SSA, wavelet transform, and other machine learning networks (Step3 in fig1).

We mainly simulated our algorithms on the data set named MAHNOB-HCI, which contains approximately 500 videos and corresponding ECG signals. In this project, the heart rate calculated from ECG signal can be regarded as the actual value of heart rates, which means that ECG value can be a criterion in signal processing method or label value in machine learning. To obtain comparable results from different processing methods, we use the PSD method frequently, which is used to related one-dimensional signal with the actual value of heart rates under the different image processing methods.



**FIGURE 1.** Technical routes. This flow chart was provided by the Wuhan team

---

## Literature review

### 2.1 Region of interest (ROI) Tracking

In Poh, *et al.* [1] in 2010, automatic face tracker for detecting faces in video frames and locating the region of interest (ROI) for each video frame. The automatic face tracker they use is MATLAB-compatible version OpenCV which uses full face as ROI. The OpenCV face detection algorithm is based on work by Viola and Jones [2], as well as Lienhart and Maydt [3]. As shown by Alexel [4] in 2016, the cheek is also a proper area for recognized as ROI. He selects both cheeks as ROI with no overlap. In Balakrishnan, Durand, and Guttag, the nose area is used as the ROI [5]. However, in Guha's experiment, subtle color changes of skin reflected by ambient light is just complementary to the extraction of pulse rate from video. The main objective of this experiment is detecting pulse from head motion rather than using PPG, so this is for reference only. In 2020, Reza specifies the forehead area of each video frame as ROI [6] because he indicates that the human face has more blood vessels in the head than any other part of the body and therefore, more blood flows into this region, so color reflection is more obvious.

As conclusion, almost every part of face has been used in previous research which means that there is still not an obvious tendency in selecting ROI. However, as Wim concluded [7], using a very small area in the forehead as ROI has the same power as using the whole forehead as ROI. The main difference is that a small ROI contains more noise. What's more, it also shows that using the whole face as ROI reduces the amplitude of the HR but also reduces noise. Therefore, in Section 3.2, we use a small area in forehead, forehead, a larger area than forehead and whole face as ROI and try to find the best ROI for our project.

### 2.2 Colour space selection

After detection of ROI from the whole image, the next step is to extract a useful color signal that is related to the HR signal. In that case, color space selection is concerned, because it's important to find a way that extracts more information needed. For color space selection, several major color spaces are used widely Color space is how the color signal represented in computer and by using different color space, different specific color signal in image is enhanced. For example, RGB is the color space that be used widely which considers color as the combination of red, green and blue and Luv takes lightness into account, etc. What's more, independent component analysis (ICA) is also a method that at the same procedure as color space selection however, it's not a kind of color space. In the field of PPG research, color space is crucial because it needs a method that can show color differences caused by Vasodilation.

In Wim, *et al.* [7], RGB-Green color space is used. Before his research, the light sources

---

used in PPG are typically red or infra-red (IR) wavelengths. Due to the historical focus of PPG on pulse oximetry and the associated need to take samples of relatively deep veins and arteries, the visible spectrum is usually ignored as a reference source for PPG [8]. In Poh, *et al.* [1], RGB color space is used and by comparing signal trace that extracts from each color channel, he discovers that the green channel contains more useful information related to cardiac pulse. However, the red and blue channels also contain relative information that can help to reveal real HR signals. He suggests that it may be helpful to analyze the whole RGB channel and using the red and blue channels as complementary data. YCbCr is a new mathematical model that first is proposed by Wang [9] in 2017. This model is based on optical and physical considerations and assuming a constant spectrum for a single light source. In Bousefsaf, *et al.* [10], he has chosen to segment the frames using a set of criteria based on the image luminance distribution, which is formed by calculating a histogram using the  $L^*$  luminance component of the CIE  $L^*u^*v^*$  color space. This particular color space is calculated indirectly from CIE XYZ, a derived version of the red, green, and blue (RGB) color space.

In conclusion, we find that color space plays a critical role in the PPG field which influences the data quality extracted from the image. Therefore, we are going to implement and compare some of the color spaces listed above where we find that the comparison among color spaces has not been contained in previous research. However, some clues hint at the rank among color spaces. RGB-Green is just one channel of RGB and in Poh, *et al.* [1], the red and blue channels can also provide complementary information, so we think that result using RGB will be better than using RBG-Green. What's more, LUV is calculated indirectly from CIE XYZ, a derived version of the red, green, and blue (RGB) color space. This means LUV may have some improvement in results. However, ICA is one of BSS method which will Discover independent source signals from a set of observations that consist of a linear mixture of the underlying sources and now is typically based on RGB color space. So theoretically, ICA is also better than RGB. The problem is that the comparison of LUV and ICA has not been done before and there is no result of which is better. In our work, we will compare among color spaces and focus more on the comparison of LUV and ICA.

## 2.3 Heart rate detection methodology

After extracting color signals using specific color space, we get a series of signal data on heartbeat signals under time series. Although a proper color space will include more useful information, this data series still need to be filtered and processed.

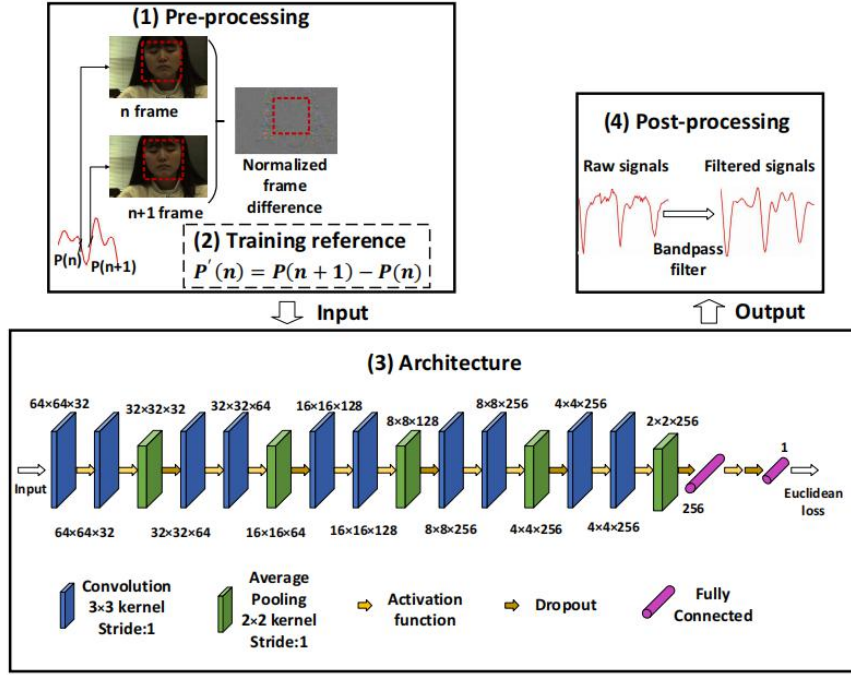
Yu, *et al.* [11] state that Principal component analysis (PCA) is proper for recover the blood volume pulses (BVP) which is a critical quota in heart rate estimation. PCA is a less computationally intensive method than ICA to extract instantaneous heart rate from a short video period that varies dynamically which means PCA is more reliable in real-time estimation. Yu, *et al.* [12] proposed a method combining ICA and mutual information to calculate the dynamic heart rate variation of short video sequences. For short video sequences, the challenge with the ICA method is that there may not be sufficient

---

independence between ICA sources. Therefore, mutual information is used to create independent sources to obtain accurate readings. However, this method is computationally intensive. Wang, *et al.* [13] use a new method called self-adaptive singular spectrum analysis (SSA). This method can remove irregular noise and extract clean pulse waves by combining singular spectrum analysis with an adaptive function. In Bal [14], wavelet transform (WT) is used which can construct a time-frequency representation of a signal. The advantage of WT is that due to its variable window width, the wavelet transform can detect rapid changes in frequency in time. What's more, WT is appropriate for non-stationary signal analysis. These advantages have led to the increasing use of wavelet transforms for bio-signal analysis although its shortage is shift variance and aliasing.

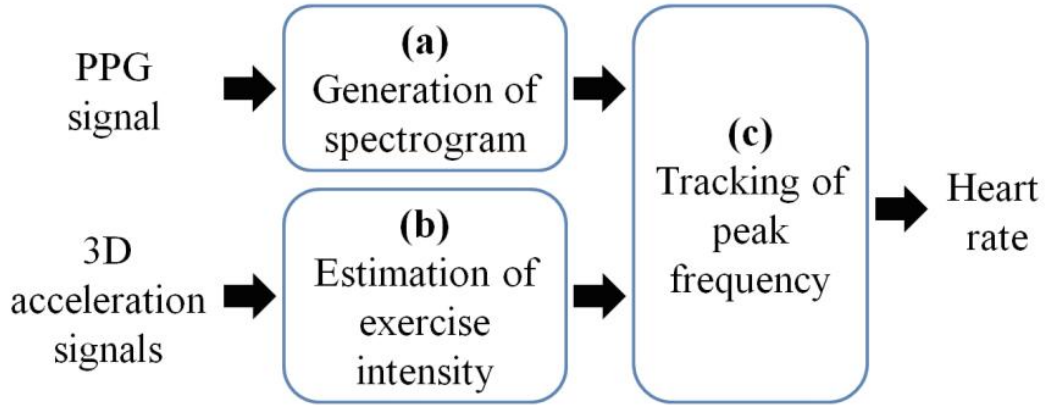
Although traditional methods have met with considerable success, there are still some parameters expertly tuned in these methods, and this could prevent the generalization of the developed methodologies. In addition, there are also some researches in this field using machine learning methods like CNN and LSTM. In 2019, Biswas, *et al.* [19] supported a novel four-layer deep learning framework named Cornet, which contains two CNN layers and two LSTM layers, to handle the PPG signal from the wrist-worn equipment. Their network can detect the heart rate and breath rate at the same time, and reach the MAE of  $1.47 \pm 3.37$  BPM for HR estimation. They mention that to remove motion artifacts(MA) from the wrist-worn device, adaptive filtering, Kalman filtering, Wiener filtering, independent component analysis and empirical mode decomposition (EMD) can be used. In 2018, Chen, *et al.* [20] designed an end-to-end network architecture(DeepPhys) based on the convolution attention principle for RGB video-based heartbeat and breath rate measurement, which reached an MAE of 4.57 bpm in the MANHOB-HCI data set. A 6th-order Butterworth filter(cut-off frequencies of 0.7 and 2.5 Hz) was applied to heart rate detection preprocessing. They also test six different subjects with various head-motion, confirm that the network performs well in the presence of motion. And Qi Zhan, *et al.* [21] gives a further explanation of the DeepPhys method, they believe that the physical signal contributes more than motion





**FIGURE 2.** DeepPhys method from [21]

Converting one-dimensional signals to spectrogram is another method to detect the physiological signals. In 2016, Hayashi, *et al.* [27] proposed a method to predict heart rate from PPG signals. They convert PPG signal into a spectrum using STFT (Short-Time Fourier Transform) and by tracking the peak of frequency to find the heart rate of people. And In 2017, Fukunishi, *et al.* [28] proposes a similar approach to extract hemoglobin components from video frames and finally convert them into spectrograms to find frequency trajectories over time and then the heart rate variability.



**FIGURE 3.** Hayashi's method from [27]

In conclusion, there are series of methods implemented in PPG field. Currently, our work are focus on wavelet transform, PSD, and some machine learning methods trying to improve the accuracy of HR prediction.

---

## 3. Lixiang

As a beginning, our project cooperates with a research group in Central China Normal University. The coding scheme of using signal processing way in iPPG prediction is given by them and what I do is add some modification and try several different methods based on their code.

First, I choose the data source we will use throughout the whole project. Secondly, I use a face detection algorithm to detect and extract ROI. Then I compare the performance of different ROI. Thirdly, the color signal is extracted using different color spaces, then the result is compared to select the best color space to use. Then, the 2D color signal is used to calculate the 1D heart rate signal. Finally, a software GUI prototype is used to give a more visual expression of our work.

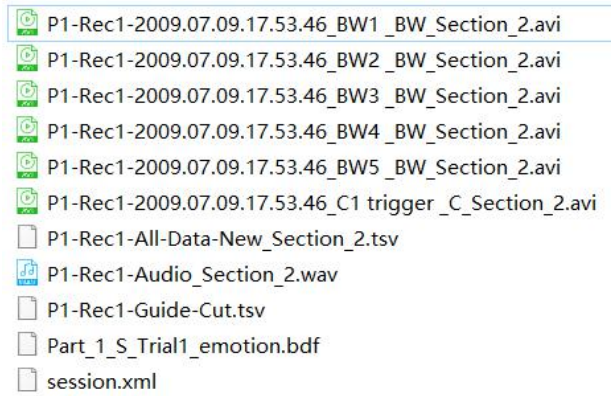
### 3.1 Data set description

The first thing we need to do in this project is figuring out the data source because the relevant data is hard to be extracted by ourselves which means an external source is needed. By reading relevant literature in iPPG fields, we notice that a database is used several times in recent years which is the HCI-tagging database from MANHOB Databases that is contributed by Imperial College London.

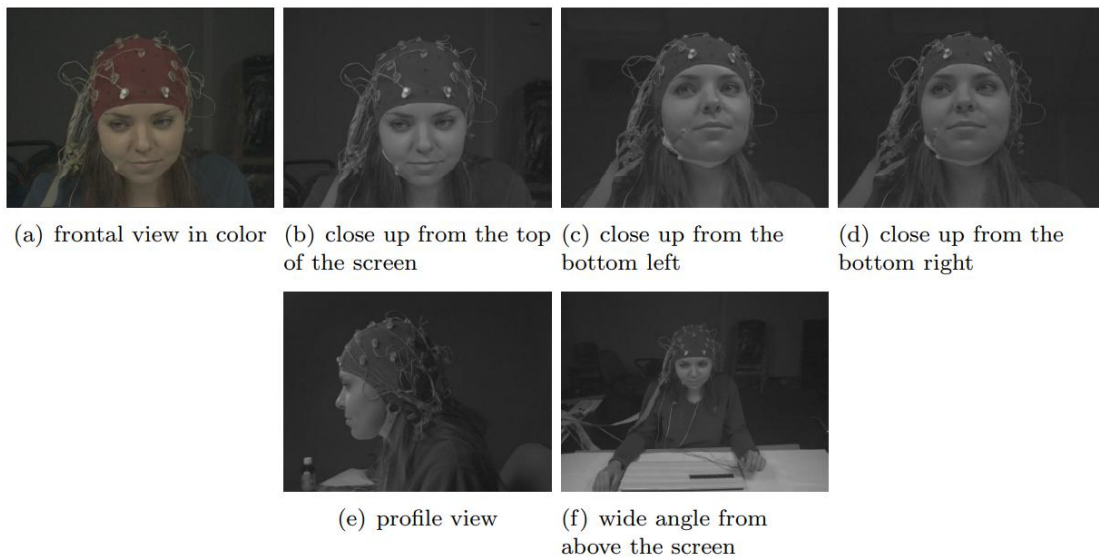
In that case, we decide to use this database as our data source for several reasons. Firstly, this database is referred to by literature which is a side-by-side demonstration of the authority of the data set. Secondly, after we get our result, we can compare our result with the output they get to validate the efficiency and feasibility of our algorithm. Thirdly, this is an easy way to get high-quality data without spending too much time.

In this project, most of our codes are running on this data set named HCI-tagging database which is 184 GB in total. This database contains 562 sets of video samples (Fig 4.) collected from 27 persons (12 males and 15 females) that each last approximately 2 minutes. All videos are at 61 fps with 780\*580 resolution. Each set contains 6 videos that are filmed at different angles and only one angle taken from the front is in color (image a in Fig.5). In addition, this data set also contains eye gaze data, electrocardiogram (ECG) data, electroencephalogram (EEG) data, and galvanic skin resistance (GSR) data. Among all data, we use the color video filmed from the first 5s to 35s of video named trigger as input to our algorithm and ECG signal as true heart rate value.

ECG signal is calculated from bdf files by using Peak searching algorithm which is already be done by the team in Wu Han. They first change the bdf files into txt files which are readable for python, then they use PSD algorithm to calculate the ECG signal.



**FIGURE 4.** the content of one video set from overall 562 sets. avi files are video files. Wav file is an audio file. Tsv files are eye gaze data files. Bdf file contains ECG and EEG signals.



**FIGURE 5.** Videos filmed from different angles in initial data set.

As the result, the first process is to extract the full-color videos. So, I write a batch file operation code to grab the video from the initial data set which name including 'trigger', which are shown in Fig 4. The 'trigger' videos are the input to either our traditional signal processing method or machine learning method.

## 3.2 face interest area tracking

### 3.2.1 Related toolbox

MATLAB-compatible version of the Open Computer Vision. OpenCV is a cross-platform open-source computer vision and machine learning software library distributed under the BSD license. Because all post-processing and video signal extraction is done on MATLAB, so a

---

MATLAB-compatible version is needed. In our project, we use the default algorithm provided by OpenCV to obtain the coordinate of face location.

### 3.2.2 ROI detected algorithm

An automatic face tracker is used to examine the faces in the video frames and to locate the region of interest (ROI) for each video frame. After that, we used the library provided by OpenCV to obtain the coordinate of face location. The algorithm in OpenCV is supported by Viola and Jones [1] as well as Lienhart and Maydt [2] which detect the whole face in default but can also be shifted to detect cheek, forehead, etc. supported by Viola and Jones work, a pre-trained frontal face classifier available with OpenCV 2.0 is used. This is a cascade classifier using 14 Haar-like digital image features trained by both positive and negative samples.

For each face detected, the algorithm returns the x and y coordinates as well as the height and width of the box defining the face's surroundings. From this output, we select the center 60% width and full height of the box as the ROI for subsequent calculations. To prevent face segmentation errors from affecting the performance of the algorithm, the face coordinates of the previous frame are used if no face is detected. In next chapter, I will illustrate the selection of ROI.

### 3.2.3 Compare different ROI

We measure the difference using different ROI. The blue area in Fig 6 is used and analyzed by Goudarzi, *et al.* [6] which is the reference of our project. I select an area smaller than the blue area and an area bigger than the blue area trying to illustrate the Reasonableness of using the blue area. What's more, I also select the whole face as a comparing subject.

To compare the results, I use the power spectra of signals extracted from the video as the standard. From Fig 7. We find that the blue area contains the HR signal which frequency is around 1 and that verifies Goudarzi, *et al.* [6] However, a smaller area than the blue area which is green area, also contains a similar or even stronger heart rate signal comparing with the blue area. This fact may show that some areas in ROI are not contributing to heart rate signals. But a small area will also be more vulnerable to movement artifact illustrated by Kumar, *et al.* [18]. We think that is because there is an averaging process when calculating the color data in one frame and this step can reduce noise and increase SNR when the area is proper bigger. Movement artifact is the phenomenon that the reflection of skin is influence by body movement which is unavoidable for human and those movements are noises that reduce the accuracy to some extends and is hard to be removed. The red ROI area seems to contain a stronger signal than others. However, the signal contains more noise what means it contains more noise associated with movement at random frequency. Finally, the black area

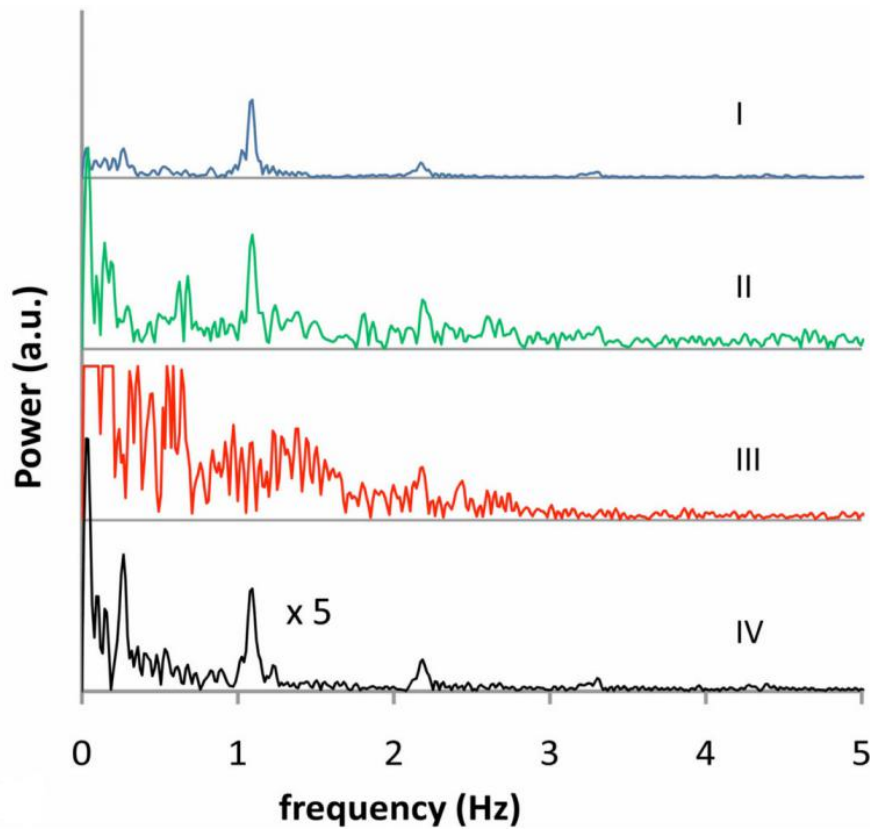
---

includes the entire face which illustrates the selection of ROI is not that important in measuring heart rate signal. The reason is, although including the whole face takes more background pixels into account which causes the reduction of heart rate signal amplitude, the noise is also reduced due to averaging with background pixels. This makes the SNR unchanged, and even the 3rd harmonic is above the noise level.

In a conclusion, using an area of face may introduce some uncertainty in the experiment. If ROI is set to a small region, the movement artifact will be a big problem. If ROI is set to a large region, the signal will also contain lots of noise introduced by movement. In that case, the ROI extracted from the face should be a proper size which needs several modifications. Therefore, ROI we use is the whole face which is more stable against movement artifact although the signal power is lower than using other ROI.



**FIGURE 6.** This is the ROI we test. Green (1) area is a little area on forehead. Blue area is the ROI that has already used by previous research. Red area is a relevant bigger area than blue area. Black area is the whole face with surrounding.



**FIGURE 7.** the corresponding power spectra of using different region of interest. The black line is corresponding to ROI using whole face. The power spectra amplitude of black line are multiplied by 5.

### 3.3 feature extraction

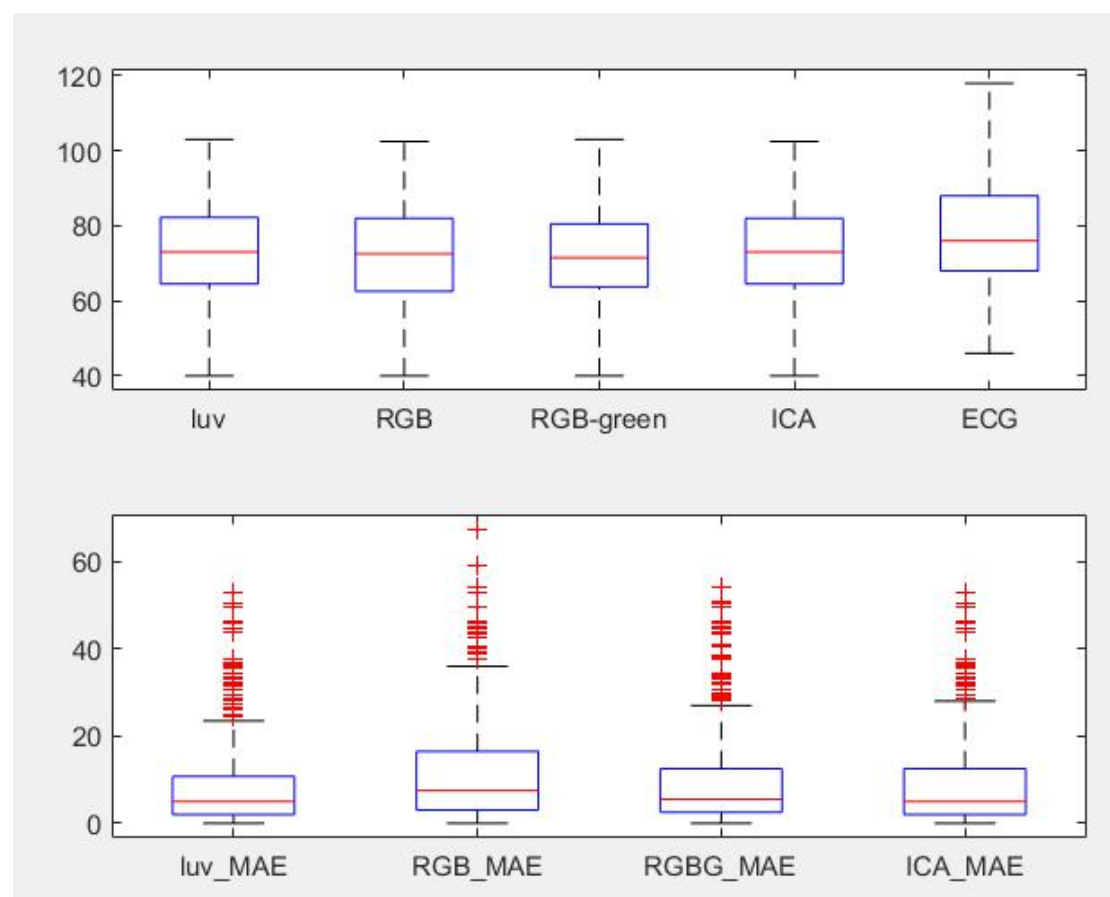
#### 3.3.1 Different color space comparison

As shown in Fig 4. above, the file name pattern of ECG files and color trigger videos are different. What's more, the reference ECG signal sheet we extract is related to a file name that looks like 'Part\_1\_S\_Trial3\_emotion.bdf' and iPPG signal sheet calculated by our algorithm is related to a file name pattern like 'P1-Rec1-2009.07.09.17.53.46\_C1 trigger\_C\_Section\_6.avi'. Now, we need to match those two sheets. Two filename patterns are quite different but fortunately, they are under the same folder. So, I search all bdf files and then find files that have 'trigger' in their filename using regular expression and make a matching list. Using that list, I import POI to a java maven project by adding it to dependency and can finally output the result to an excel file. Now, we match the result we get and the reference ECG result (Fig 8.) and can make a comparison between each method. In our subject, we compare 4 color spaces which are RGB, RGB-Green, Luv, and ICA. Those color spaces are all mentioned and have been used by previous works of literature. However, the comparison of all 4 of them is seldom made especially between ICA and Luv.

Firstly, the initial extracted ROI uses RGB color space. And RGB-Green just needs to use the green channel only which is easy for MATLAB to realize. Luv is calculated indirectly from XYZ space which is a derived version of RGB. So, to use Luv, we need to transfer RGB to XYZ first and this is easy to manipulate in Matlab and then transform XYZ to Luv using makecform function. ICA is a method treat each channel of RGB as an independent channel want to find a linear combination of 3 channel to optimize the result of RGB. ICA intends to find the component that truly influences the result of forecasting heart rate and reduce the influence of Gaussian noise.

Theoretically, we forecast the result is that using only RGB is the worst and RGB-Green is better than RGB, ICA is better than RGB-Green, and Luv is also better than RGB-Green. The reason is that the green channel can best influence the heart rate signal and RGB color space includes more noise. However, by using the ICA method which can reduce Gaussian noise by using the red and blue channels as complementary data, the result may be better than just using RGB-Green. What's more, the comparison between Luv and ICA is equivocal because they are all based on RGB color space.

The statistical result is shown in Fig 9, from which we can see the distribution of MAE using each color space. And Fig 10 shows the average heart rate MAE using whole data set, from which we can directly notice that Luv is better than others.



**FIGURE 8.** This box diagram shows the comparison of result using different color space in our project. In the graph, red line is the median value, the upper and lower boundaries are the 1<sup>st</sup> qrt. and 3<sup>rd</sup> qrt. What's more, it also shows the max and min value within 2 standard variation and red crossbars are the extreme values. Beneath graph is the MAE between calculated result and ECG. From the graph, we can find that luv is better than ICA is better than RGB-Green is better than RGB.

file_name	luv	RGB	RGB-g	ICA	ECG	RGB_ERROR	RGB-g_ERROR	luv_ERROR	ICA_ERROR	
Part_1_S_Trial10_emotion_data-PPG	69.5	81.5	69.5	68.5	68	13.5	1.5	1.5	0.5	13.5
Part_1_S_Trial11_emotion_data-PPG	88.5	72.5	71.5	70.5	72	0.5	0.5	16.5	1.5	0.5
Part_1_S_Trial12_emotion_data-PPG	73	68	68	60	68	0	0	5	8	0
Part_1_S_Trial13_emotion_data-PPG	92.5	69	68.5	68.5	70	1	1.5	22.5	1.5	-1
Part_1_S_Trial14_emotion_data-PPG	68	69	68	63	74	5	6	6	11	-5
Part_1_S_Trial15_emotion_data-PPG	71.5	77	71.5	53	72	5	0.5	0.5	19	5
Part_1_S_Trial16_emotion_data-PPG	69.5	65	70	53	76	11	6	6.5	23	-11
Part_1_S_Trial17_emotion_data-PPG	94.5	81	81	53.5	74	7	7	20.5	20.5	7
Part_1_S_Trial18_emotion_data-PPG	71	71.5	71.5	52	72	0.5	0.5	1	20	-0.5
Part_1_S_Trial19_emotion_data-PPG	81	59.5	67.5	51	68	8.5	0.5	13	17	-8.5
Part_1_S_Trial20_emotion_data-PPG	71.5	71.5	71.5	53.5	72	0.5	0.5	0.5	18.5	-0.5
Part_1_S_Trial8_emotion_data-PPG	69.5	50.5	69.5	50.5	70	19.5	0.5	0.5	19.5	-19.5

**FIGURE 9.** The processed result using different color space of traditional signal processing method.

	MAE	RMSE
Luv	8.6250	11.0511
RGB	11.4606	13.6467
RGB-Green	9.3799	12.0452
ICA	9.0246	11.6975

**FIGURE 10.** the average MAE value using each color space, from which we can find that luv is a little bit better than others. And ICA is the second best while RGB has the worst performance which is like our prediction.

### 3.3.2 calculate 1D HR signal from 2D image

After extract the 2D color information using Luv color space, we fusion the data to get a final 1D heart rate signal. In our method, all pixels in ROI are used to calculate the average PPG value by a spatial averaging method. What's more, pixels are transformed into scalar to improve signal to noise ratio. The method we use is as followed.

$$signal(t) = \frac{1}{N_x \cdot N_y} \sum_{x=1}^{N_x} \sum_{y=1}^{N_y} I(x, y, t)$$

In this method,  $I(x, y)$  represent the pixel intensity at a particular  $(x, y)$  position, and  $N_x$ ,  $N_y$  represents the width and height of target region separately. By using this function, we can



get a 1D PPG sequence (Fig 11.).

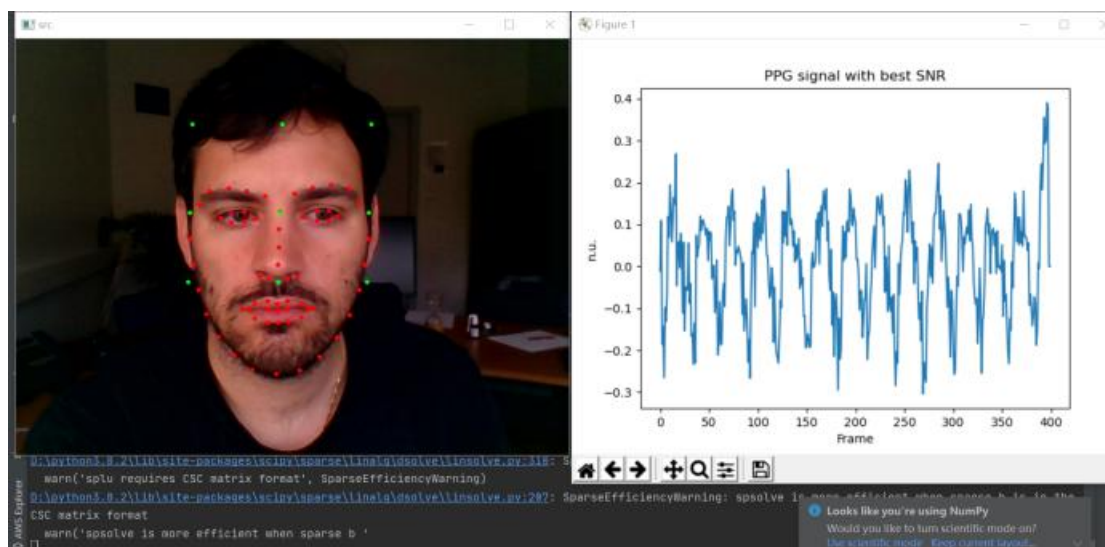
After averaging, the result 1D signal still need to be filtered properly to get the final HR value and PSD algorithm is applied here.

### 3.3.3 conclusion

For face extraction algorithm, functions provided by cv2 in MATLAB is used. For ROI, whole face is used. For color space, Luv is used. For filtering process, PSD is used. For 1D signal calculation, an spatial averaging algorithm is used. And up to now, the best result we get on whole data set is as followed:

```
luv_MAE_avg =  
  
8.6250
```

However, there is also something need to be improved. For example, the filter we use need to be considered and compared among several filters. All other optimization work are finished by Cheng who is my teammate.



**FIGURE 11.** The spatial averaging result of one video

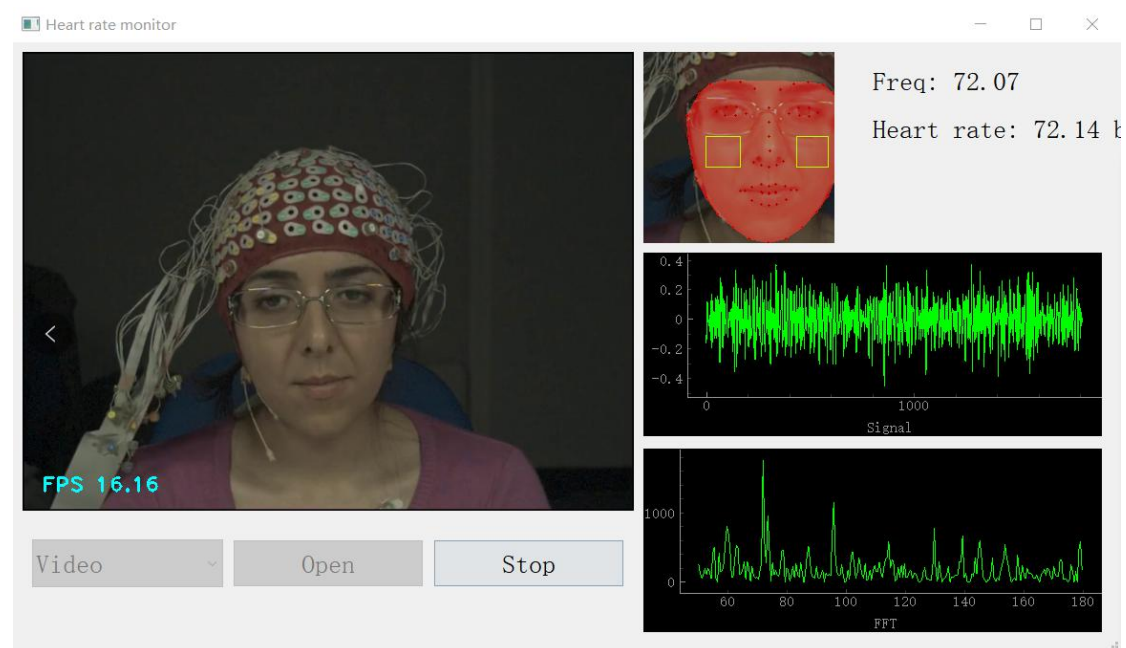
## 3.4 GUI prototype

This GUI prototype can give a detected HR of a video or by using the default computer webcam in real-time. It can also show the real-time HR waveform and its FFT result. The architecture of the software is downloaded from Github[29] which is open-sourced, and we

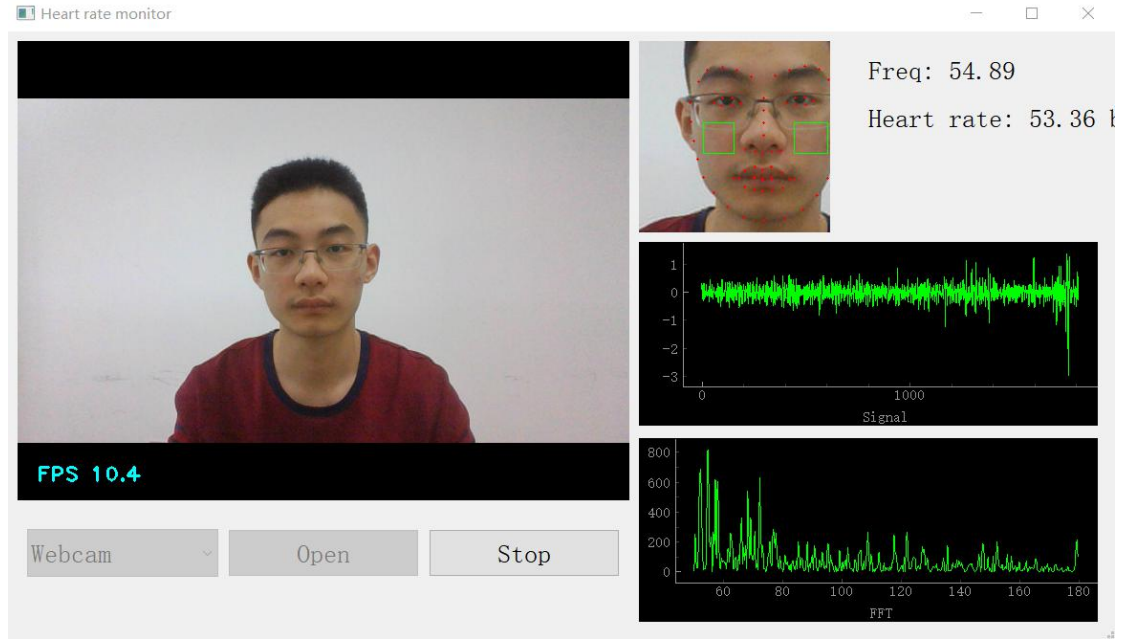
substitute its core detection and prediction algorithm to our own work. The video or webcam detection is realized by the cv2 package and if the video file path is incorrect or no webcam detected, an error is raised.

The only difference of realize real-time HR detection is that we need to use a short period of the video stream as input rather than use a 30s period like what we do in previous work. In that case, we set a buffer to store frames read from Video streaming, and only if the buffer is full, the prediction algorithm begins to process. Once the buffer is full, the frames in the buffer are processed and the buffer is flashed which means the result is the prediction of a period and periods do not have overlaps. What's more, the fps at the left bottom corner are the times that the computer process frames in a second, namely times that buffer filled within a second. The output result of using video as input is shown in Fig 12. and the result of using a webcam is shown in Fig 13.

The result is that the initialization of a webcam or loading a video needs time, so there is a short delay for the detection since the program has begun. What's more, at the very beginning, the performance is not very good and then the result becomes better and better gradually. The reason maybe is the lack of data which will directly influence the resulting quality. However, the overall performance is acceptable which proves the feasibility to apply this technic in the production environment of real life.



**FIGURE 12.** Video mode test (ECG heart rate =68.0 bpm)



**FIGURE 13.** Webcam mode test(sitting state, Cheng)

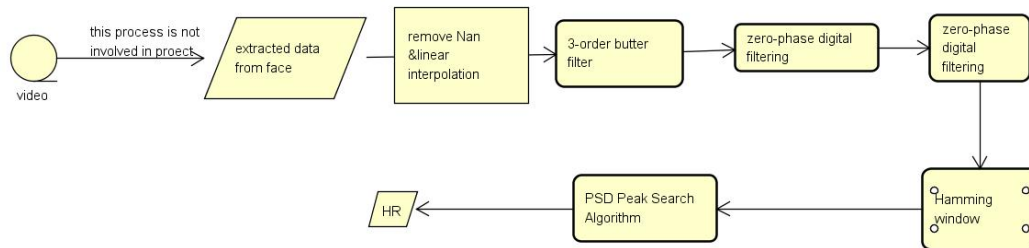
## 4. Chenglitao

### 4.1 Traditional method(signal process)

My work is based on Li's part. After getting the one-dimension heart rate signal, we try to get a signal with less noise. From the literature, we know that the error of heart rate detection rate is caused by several reasons, such as movement of the test object and flicker of the light source. And as far as many scholars mention, the heartbeat signal has a certain analyzable pattern, and the value of the signal at a given moment is strongly dependent on the overall characteristics of the time serial signal. So it is really necessary and possible to deal with heartbeat signals with the traditional method in the frequency domain to get better performance.

The first part of my individual work is about the traditional signal processing method. I implement and compare three possible solutions of signal processing methods, which contain PSD, SSA, and wavelet transform. Each method has a different complexity level and generates different accuracy on the entire data set. After comparing different methods, I decided to use the comprehensive method to deal with a one-dimension heart rate signal.

#### 4.1.1 Power spectral density (PSD) with filter

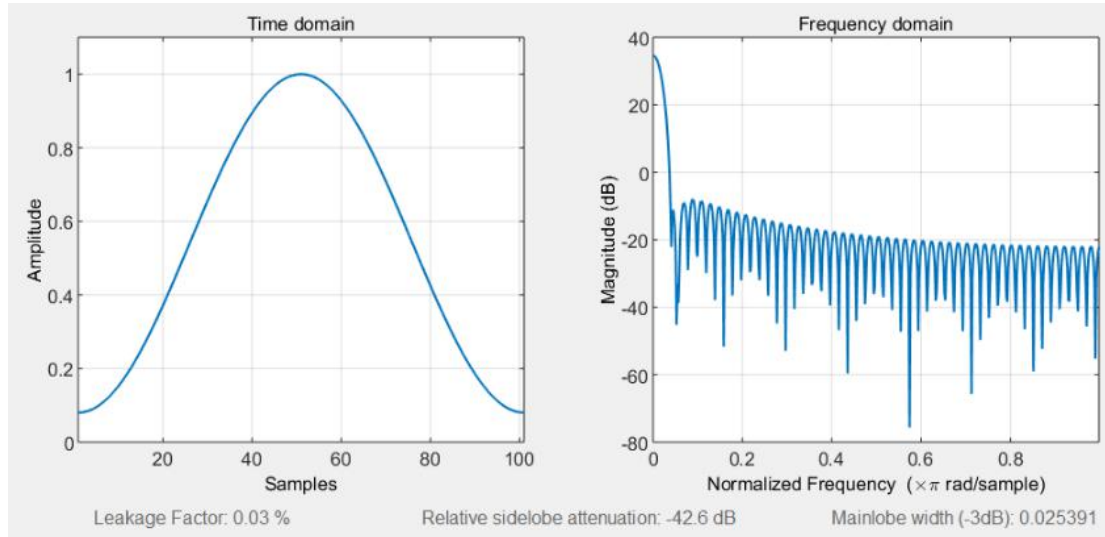


**FIGURE 14.** Flow chart of PSD peak search algorithm

The most common way to detect the heart rate in the signal processing area is to find the peak point of power spectral density(PSD). This method of heart detection can be described as a peak search algorithm. The principle of this algorithm is to extract the main frequency from the ippg signal. However, to eliminate the effects of secondary frequency, we should try some filters to rebuild the initial one-dimension signal in the frequency domain. So this period of work is testing the performance of different filter combinations, and this test work was carried out by Dr. Zhao's team in Wuhan and us. This algorithm is important in the whole project because its Principle is relatively simple when it works as the benchmark method when choosing different parameters. The process with the best performance is illustrated below.

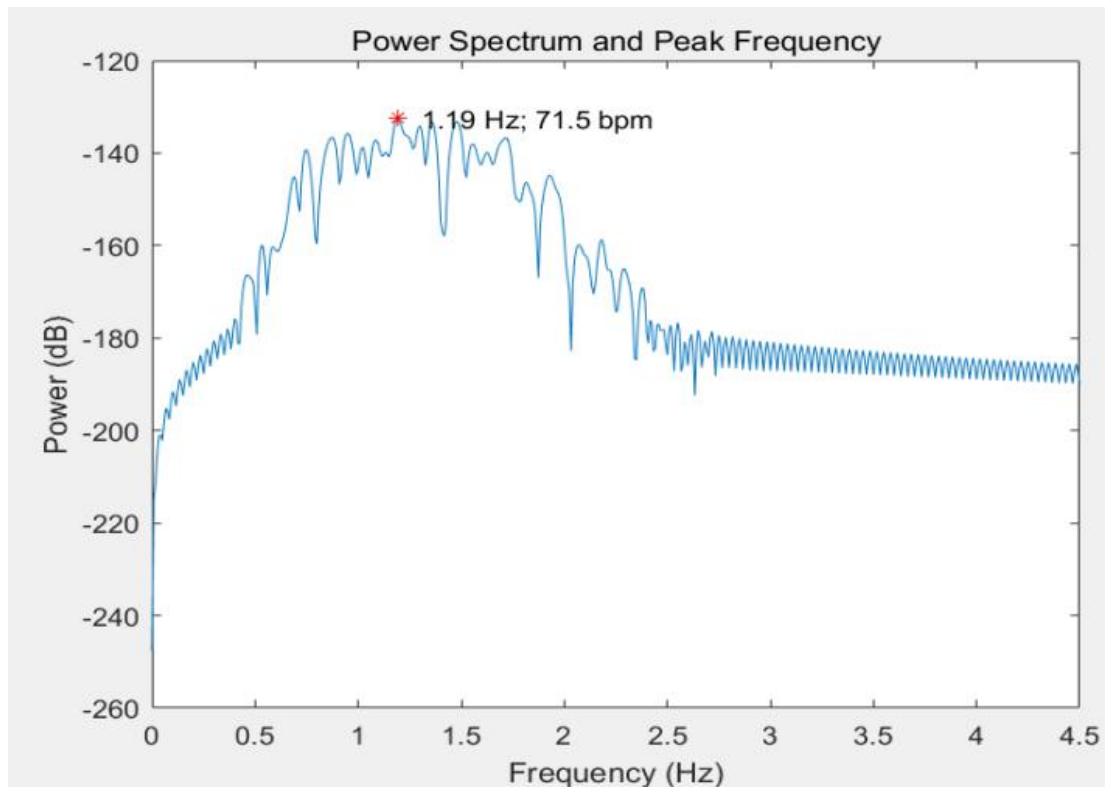
Firstly, we note that some values are missing, or displayed as Nan in time serial, so I carry out linear interpolation for missing values and replace Nan with the average value of whole time serial. In this way, the complete serials are got for further processing. Secondly, the zero-phase digital filtering method is used with a 3-order bandpass Butter Worth filter. The low frequency and high frequency of the band are set to 0.7Hz and 2.5Hz, which correspond to 42 and 150 per minute respectively.

Then, power spectral density(PSD) is generated by the discrete Fourier transform. PSD is the distribution of spectral energy per unit of time. According to Fourier analysis, any physical signal can be decomposed into several discrete frequencies or a spectrum of frequencies in a special range. Meanwhile, a Hamming window with the same length of time serial is applied for PSD generation, because direct truncation of the signal with a rectangular window will result in frequency leakage.



**FIGURE 15.** Hamming window.

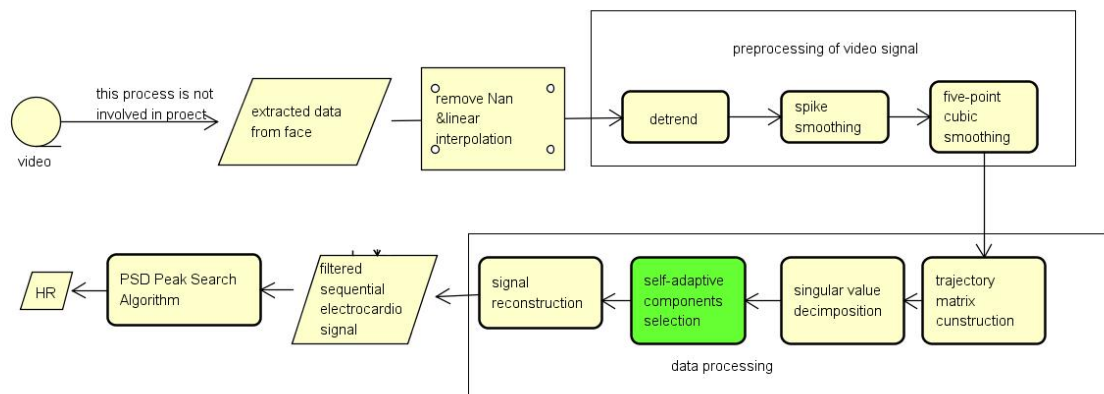
Finally, the peak point on the power spectral density(PSD) in a special frequency range is found, and this frequency is regarded as the estimation of heart rate. To get the heart rate per minute, this frequency will be multiplied by 60.



**FIGURE 16.** sample result of PSD peak search algorithm

In conclusion, peak search algorithm in PSD is the relatively simple to implement, so we use this methods as the baseline method in parameters selection. In whole data set, we get the performances: MAE=8.62 bpm, and RMSE=11.05bpm

### 4.1.2 Singular Spectrum Analysis



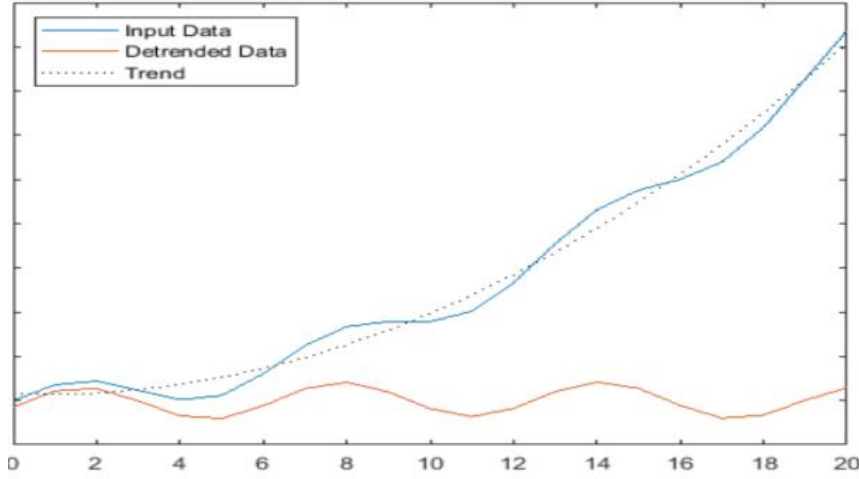
**FIGURE 17.** SSA flow chart

Singular Spectrum Analysis(SSA) is based on the principle of singular value decomposition. SSA usually works together with some other smoothing algorithms on the time domain. They work together as a filter for the time series. The complete process can be expressed as: firstly, linear interpolation is carried out for missing values and Nan values are replaced with the average value of whole time serial. This operation is inevitable in different methods. Then, some three smoothing algorithms are applied in the time domain to eliminate burrs and unsteadiness, and they can be described as preprocessing part of the flow chart. Finally, the singular value decomposition method will rebuild a trajectory matrix and select the primary singular value, and it can be used for filtering the unnecessary signal frequency. This step is named the data processing part in the flow chart. There are 3 sub steps in preprocessing step:

#### (1) Detrend

Detrend operation is subtracting its best linear fit line from the time series. Specifically, the original pulse is cut into segments that do not overlap at several M points, and the signal is interpreted by making the average of each segment equal to the average of the entire original pulse. This algorithm eliminates trends and sharp drops or rises in the original pulse. According to the literature, it is recommended to use the signal sampling rate as the value of M, so here we determine M as 61.

Removing a trend from your data allows the model to focus on analyzing the fluctuations in the data concerning the trend. A linear trend usually indicates a systematic increase or decrease in the data. In heartbeat detection, systematic trends include the light source at the scene of the shot or the movement of the tester. Removing trends, therefore, helps the model to produce better insight. Fig 18. shows how baseline canceling works.



**FIGURE 18.** How detrend works

### (2) Spike smoothing

Spike smoothing algorithm focus on some edge with dramatic changing. These edges usually originate from the high-frequency part of the signal. Similar to the previous step, the time series is cut into several segments with non-overlapping  $M$  points. If the standard deviation of a segment exceeds two times the overall standard deviation, the corresponding segment is considered to be highly noisy and the segment region needs to be scaled. Equation 1 shows exactly how scaling is achieved:

$$CR = 0.6 + 0.4 * \frac{sd_i - 2 * sd}{sd_i}$$

Where  $CR$  is the compression rate used to smooth the noise segment,  $sdi$  is the standard deviation of the noise segment and  $sd$  is the overall standard deviation of the time series. If a segment is judged to be a noisy segment, all wave values in it will be multiplied by  $(1-CR)$  to achieve spike smoothing. The degree of scaling increases when the noise increases.

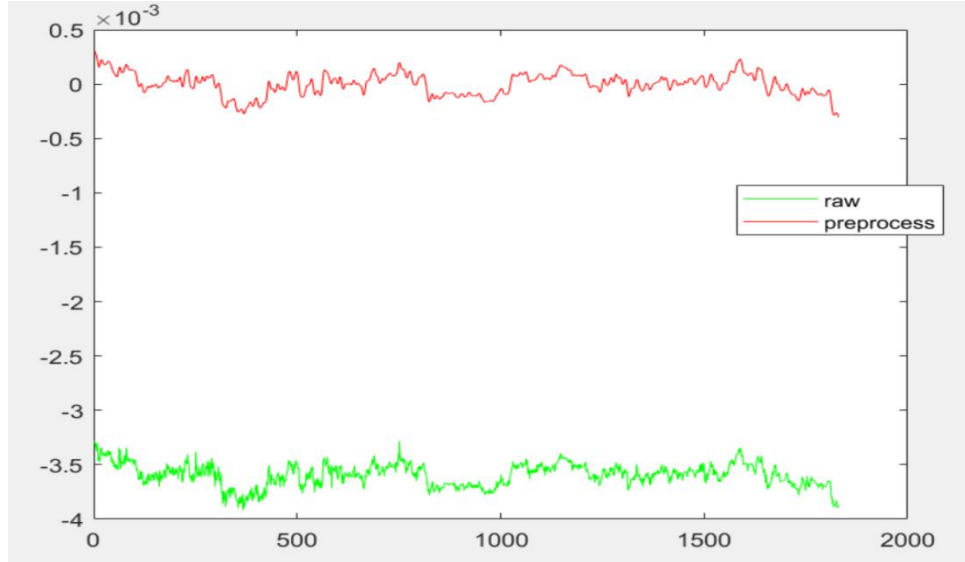
### (3) Five-point cubic smoothing

The five-point cubic smoothing algorithm can remove high-frequency random noise effectively from the signal and is widely used in digital signal processing. This algorithm uses polynomial least squares to approximate sampling points, which produces better results than traditional moving average filters in terms of detail preservation. This algorithm can produce better results with multiple iterations. The five-point cubic smoothing algorithm is expressed in the following mathematical form:



$$\left. \begin{aligned}
y_1 &= \frac{1}{70} [69x_1 + 4(x_2 + x_4) - 6x_3 - x_5] \\
y_2 &= \frac{1}{35} [2(x_1 + x_5) + 27x_2 + 12x_3 - 8x_4] \\
&\vdots \\
y_i &= \frac{1}{35} [-3(x_{i-2} + x_{i+2}) + 12(x_{i-1} + x_{i+1}) + 17x_i] \\
&\vdots \\
y_{m-1} &= \frac{1}{35} [2(x_{m-4} + x_m) - 8x_{m-3} + 12x_{m-2} + 27x_{m-1}] \\
y_m &= \frac{1}{70} [-x_{m-4} + 4(x_{m-3} + x_{m-1}) - 6x_{m-2} + 69x_m]
\end{aligned} \right\} (i = 3, 4, \dots, m-2,)$$

After the smoothing process in the above three steps, the high-frequency part of the time series and the noise are removed. Fig 19 shows the result of the processing of a complete sequence of length 1830, which shows that the degree of smoothing of the sequence has been optimized.



**FIGURE 19.** Comparison of original and pre-processed signals

SSA is a method for the time series data analysis, which got its name from the singular value decomposition of matrices. This method allows the signal to be reconstructed and decomposed to extract the different components of the time series, such as long-term trend signals, periodic signals, and noisy signals, and thus optimize the prediction results. There are four sub-steps:

#### **(1) Trajectory matrix construction and singular value decomposition**

For time series with length  $N$ , the length of the window used to construct the trajectory matrix is  $L$ , and defining  $K = N - L + 1$ , the trajectory matrix can be expressed below.



---


$$\mathbf{X} = \begin{bmatrix} x_1 & x_2 & \cdots & x_K \\ x_2 & x_3 & \cdots & x_{K+1} \\ \vdots & \vdots & \vdots & \vdots \\ x_L & x_{L+1} & \cdots & x_N \end{bmatrix}$$

The singular value decomposition of trajectory matrix  $\mathbf{X}$  is given by:

$$\mathbf{X} = \sum_{i=1}^d \lambda_i \mathbf{U}_i \mathbf{V}_i^T$$

Where  $d$  is the number of non-zero singular values of  $\mathbf{X}$ . Clearly  $d = \text{rank}(\mathbf{X}) \leq \min(L, K)$ , and  $\mathbf{U}_i$  and  $\mathbf{V}_i$  are the left and right singular vectors of  $\mathbf{X}$  respectively. Here, the singular values are arranged decreasingly, and the value of  $\lambda_i$  decreases as the index  $i$  increases, indicating that components with low index numbers contribute more to the original signal.

## (2) Self-adaptive components selection

The main energy of the signal is concentrated on the first  $r$  ( $r < d$ ) larger singular values, while the smaller singular values are considered as noise components. The aim of component selection is to determine an appropriate value of  $r$  so that a noise-free pulse wave can be reconstructed from the previous  $r$  singular values. If  $r$  is too small, useful detail is lost. Therefore the inexact Augmented Lagrange Multiplier algorithm (IALM) is used to implement adaptive component selection. The core of this algorithm is to test different numbers of singular values from large to small and determine whether the sum of these components is within the acceptable range. This part of the code comes from the proposer of the algorithm. After the  $r$  most important components have been selected, the approximation of the Trajectory matrix can be expressed as:

$$\mathbf{RCA} = \mathbf{U}(1, 2, \cdots, r) * \mathbf{V}^T \begin{pmatrix} 1 \\ 2 \\ \vdots \\ r \end{pmatrix}$$

The dimension of  $\mathbf{RCA}$  is  $L \times K$ , and  $\mathbf{U}$  and  $\mathbf{V}$  are the left and right singular vectors of  $\mathbf{X}$  respectively.

## (3) Signal reconstruction

Considering the case where the original sequence is odd or even, the reconstructed sequence can be expressed as:

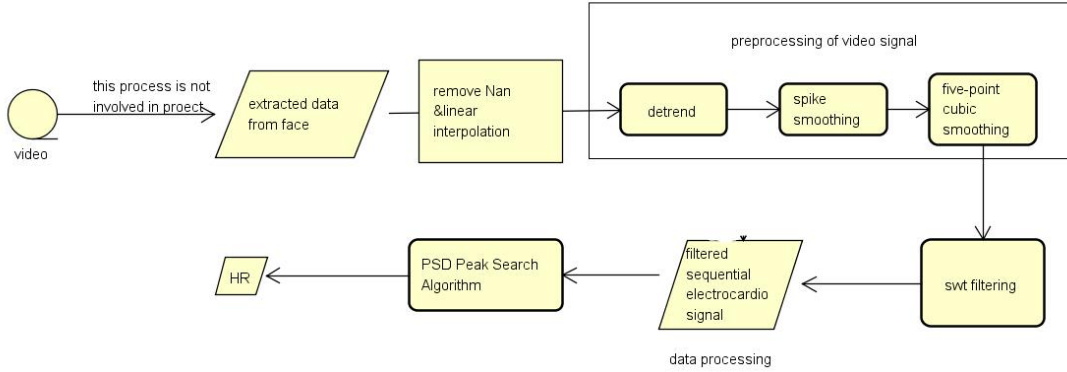
$$y_{rc} = \begin{cases} \frac{1}{k} \sum_{m=1}^k y_{m,k-m+1} & 1 \leq k < L^* \\ \frac{1}{L^*} \sum_{m=1}^{L^*} y_{m,k-m+1} & L^* \leq k \leq K^* \\ \frac{1}{N-k+1} \sum_{m=k-K^*+1}^{N-K^*+1} y_{m,k-m+1} & K^* < k \leq N \end{cases}$$

Where  $L^* = \min(L, K)$ ,  $K^* = \max(L, K)$ .

After the above steps, the denoised time serials could finally be obtained. The estimation of heart rates of the whole data set is got from PSD peak search algorithm, which mentioned in 4.1.1.

In conclusion, this algorithm is relatively complex. In whole data set, we get the performances: MAE=8.61 bpm, and RMSE=13.37bpm .The performance of SSA is not stable enough, but the smooth algorithm is able to improve the performance.

#### 4.1.3 Wavelet filter



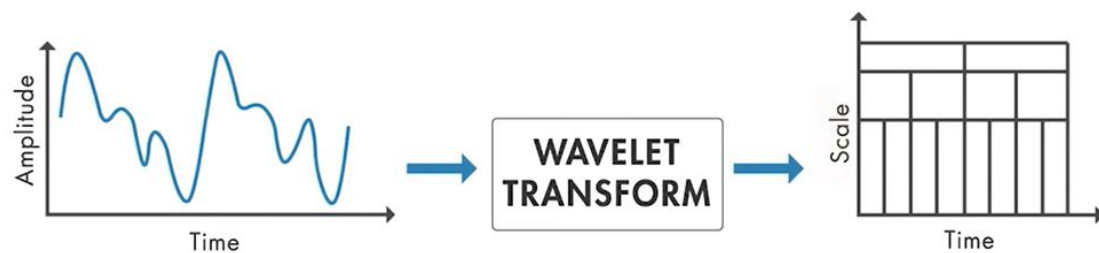
**FIGURE 20.** Flow chart of algorithm

Firstly, linear interpolation is carried out for missing values and Nan values are replaced with the average value of whole time serial.

In the previous section we found that smoothing algorithms on the time domain can achieve good results. Therefore, we decided to use a pre-processing smoothing algorithm to remove the trend.

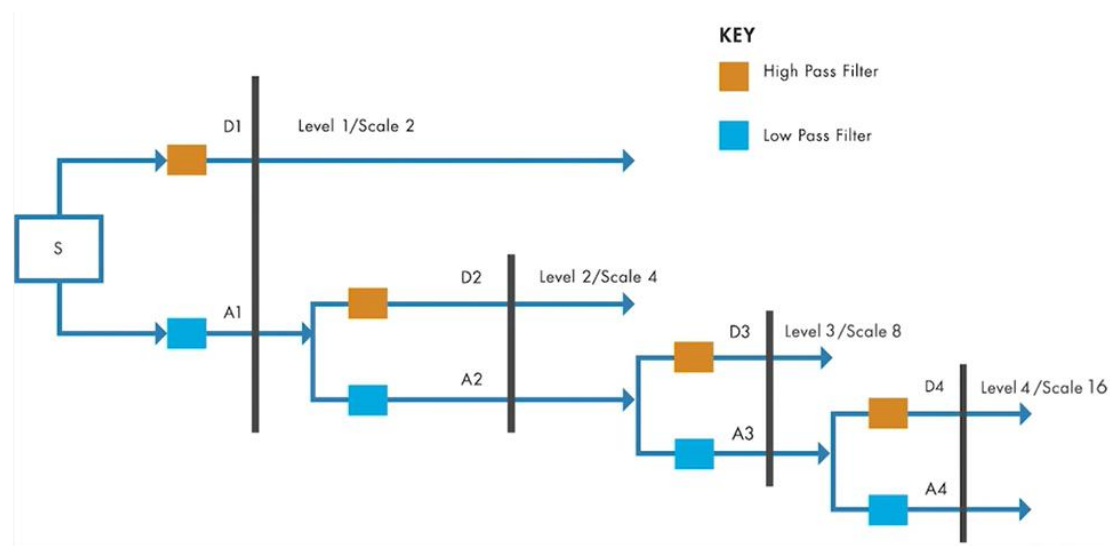
Then discrete wavelet transform is applied in this time serial. Wavelet is various types of

rapidly decaying oscillation wave with zero mean. Wavelet transform can be classified in both wavelet and continuous wavelet transform. Continuous wavelet transform(CWT) aims to obtain a simultaneous time-frequency analysis of the signal. The output of CWT is two-dimension coefficients from scale(or frequency) and time. The basic principle is to slide wavelet signals with different scaling over the time series and decompose the different frequencies in the time window. This part of the work is used in image generation in machine learning part.



**FIGURE 21.** Continuous wavelet transform(CWT)

Another type of wavelet transform is named discrete stationary wavelet transform. The discrete wavelet transform implies a power-level discretization of the scale, which means a lower frequency recognition rate. The SWT algorithm splits the signal into low pass part and high pass part, which are named approximation and detail level respectively. The decomposing process can continue for a finer scale. Then a threshold will be determined to find suitable parts, the new signal will be rebuilt for these parts left. Since discrete wavelet transform can be used in signal filtering by removing the unnecessary frequency components.



**FIGURE 22.** Discrete stationary wavelet transform(SWT) process

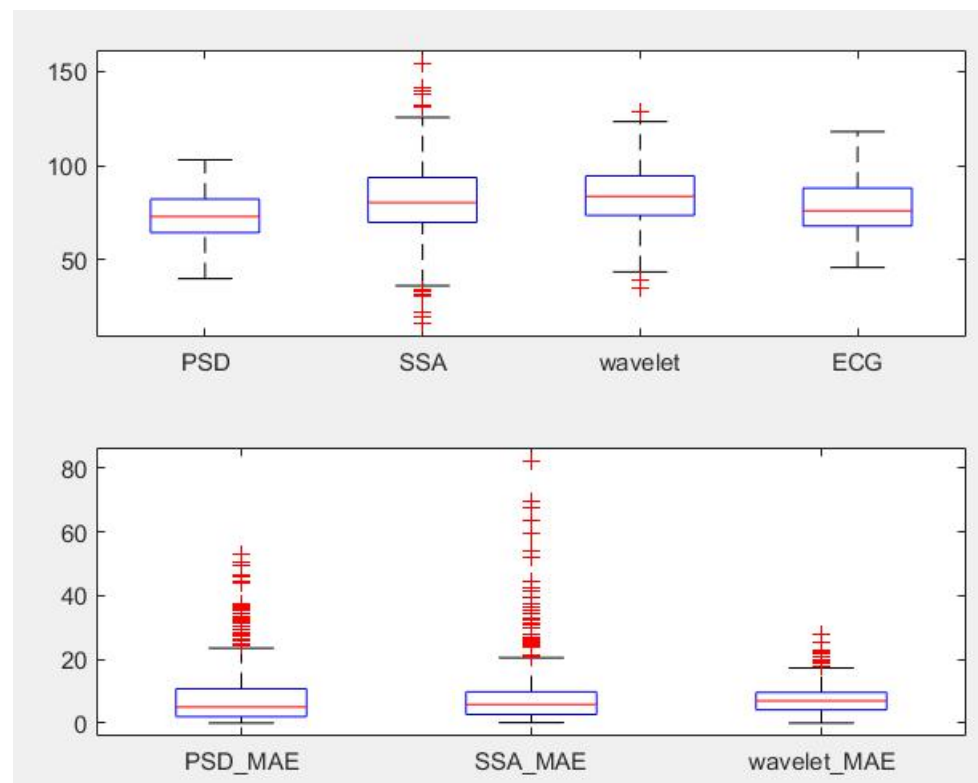
After the above steps, the denoised time series could finally be obtained. The estimation of heart rates of the whole data set is got from PSD peak search algorithm, which mentioned in 4.1.1.

In conclusion, this algorithm is relatively simple to simulate in both Matlab and python. In whole data set, we get the performances: MAE=7.34 bpm, and RMSE=8.532 .

#### 4.1.4 Conclusion

This section discusses three different signal processing methods, and we implement and compared them in Matlab. The combined method has the lowest error and is also the most stable. Therefore, this algorithm is our final choice and the input to the machine learning is obtained by this method.

	MAE	RMSE
PSD	8.6250	11.0511
SSA	8.6196	13.3714
Wavelet	7.3452	8.5320



**FIGURE 23.** This box diagram shows the comparison of result using different signal processing in our project. In the graph, red line is the median value, the upper and lower boundaries are the 1<sup>st</sup> qrt. and 3<sup>rd</sup> qrt. What's more, it also shows the max and min value within 2 standard variation and red crossbars are the extreme values. Beneath graph is the MAE between calculated result and ECG. From the graph, we can find that wavelet method is better than PSD and SSA is not stable enough for PSD baseline method.

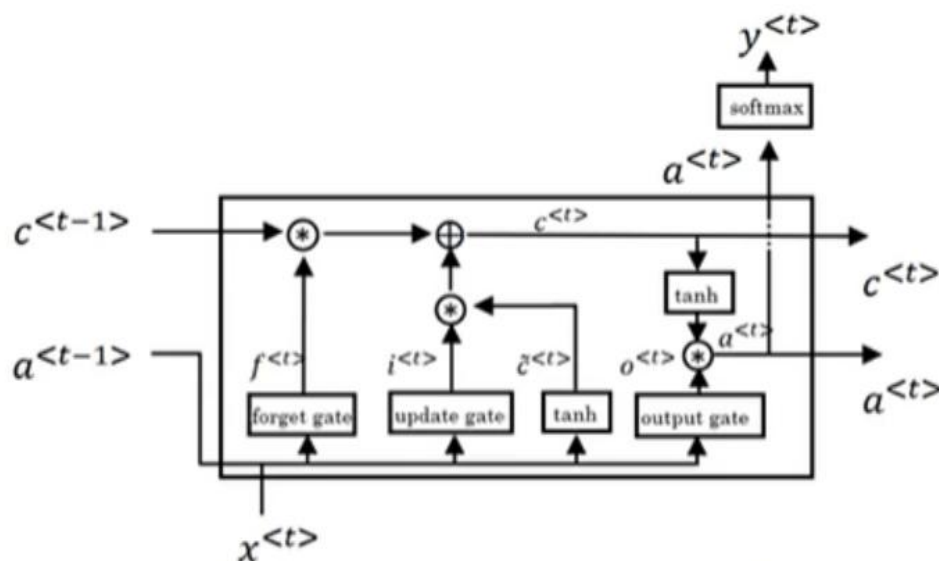
## 4.2 1D-machine learning network

In addition to traditional methods, deep learning methods have been introduced to address the heart rate detection problem due to the boom of neural networks. Now there are two general approaches been used by researchers. One is to put the facial video directly into the network, while the other is to feed pre-processed 1D rPPG signal into the neural network. As we have explored some of the traditional methods, we decide to use them as a pre-processing tool to generate a 1D rPPG signal and then feed it into the whole network.

### 4.2.1 CorNet (LSTM/CNN)

Convolution neural network is a classical and widely used structure in deep neural networks. The local connectivity, weight sharing and pooling features of convolution neural network allow them to effectively reduce the complexity of the network, reduce the number of training parameters, make the model invariant to translation, distortion and scaling to a certain extent, with strong robustness and fault tolerance, and is easy to train and optimize. Based on these superior properties, it outperforms standard fully connected neural networks in a variety of signal and information processing tasks.

The basic structure of CNN consists of an input layer, a convolution layer, a pooling layer, a fully connected layer and an output layer. The convolution and pooling layers are generally taken as a number of layers, and are alternated, i.e. a convolution layer is connected to a pooling layer, a pooling layer is connected to another convolution layer, and so on. Each neuron in the output of the convolution layer is locally connected to its input, and the corresponding connection weights are weighted and summed with the local input plus a bias value to obtain the input value of the neuron.



---

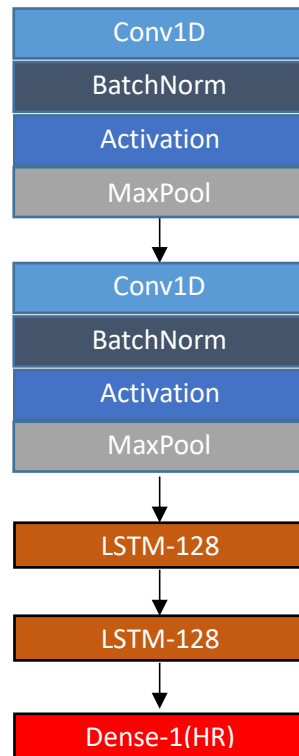
**FIGURE 24.** LSTM structure

LSTM neural networks include not only the external circulation between cells in the hidden layer involved in RNNs, but also the self-cycling within cells, allowing for more complete consideration of historical information and sequence dependencies in financial time series data. A cell contains a memory store (Cell) and three gates (Gates), the Cell records the neuronal state, the Input Gate and Output Gate are used to receive and output parameters and correct parameters, and the Forget Gate is used to control the extent to which the previous cell state is forgotten. Backpropagation takes the direction of all information transfer in the opposite direction, also known as time-dependent backpropagation in recurrent neural networks, and uses gradient descent to update parameters in backpropagation.

$$\begin{aligned}\tilde{c}^{<t>} &= \tanh(W_c[a^{<t-1>}, x^{<t>}] + b_c) \\ \Gamma_u &= \sigma(W_u[a^{<t-1>}, x^{<t>}] + b_u) \\ \Gamma_f &= \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f) \\ \Gamma_o &= \sigma(W_o[a^{<t-1>}, x^{<t>}] + b_o) \\ c^{<t>} &= \Gamma_u * \tilde{c}^{<t>} + \Gamma_f * c^{<t-1>} \\ a^{<t>} &= \Gamma_o * \tanh c^{<t>}\end{aligned}$$

**FIGURE 25.** Lstm mathematical expressions.  $x$ : input sequence;  $x^{<t>}$ :  $t$ th element;  $x(i)$ :  $i$ th sample;  $T_x(i)$ : the length of the sequence of the  $i$ th sample;  $y$ : output sequence;  $W$ : parameter matrix

After searching and reading for relevant materials, one paper grabbed our attention for its similarity in methodology and excellence at the result. In this paper, A dedicated network architecture composed of 1D CNN and LSTM has been proposed and tested for detecting heart rate [25]. They achieve an MAE of around 2~3 on their data set.



**FIGURE 26.** Architecture of corNet

According to this architecture, We have successfully reproduced corNet. But when we test the result on the good-30s data set( provided by Dr. Liang team (where ‘good ’means  $MAE < 5$  using PSD peak search algorithm. There are only 251 1D signals in this data set) the result seems not as good as shown in the paper, it seems the model does not learn anything and just predicts the mean of samples’ heart rate.

According to the rigorous test, the reason for this result is probably the selection of activation function, because we chose ReLU by default, which is likely to cause the neuron output value to be 0, this is what researchers called dying ReLU problem where some ReLU Neurons essentially die for all inputs and remain inactive no matter what input is supplied. So we change the network activation function to Leaky ReLU, a variant of ReLU that when we input less than 0, it would output some negative value instead of 0.

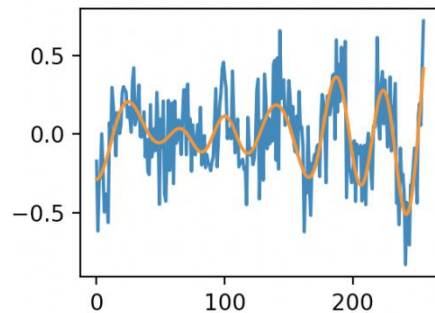
We’ve also changed some of the detail of our model to make it perform better like adding some batch normalization layer and change the filter size and filter number and so on.

The result after change the activation function can be found in Table 1(we sort the data set from the lowest heart rate to the highest to better observe the trends), the MSE( mean square error) of the test set is about 85.60 and MAE( mean absolute error) is about 6.93. But from the Heart Rate Prediction graph on the test set, though the result MAE is better than we have than the traditional method, the trend of prediction does not follow up with the real one so well.

According to the supervisor's suggestion, the problem we are facing is obviously that the generalization ability of the model is weak and our model is over-fitting. Then we tried many ways to solve this problem like adding L1 and L2 regularization to our model, but the result didn't get better after these methods. We guess the most critical reason is that the data is insufficient. The supervisor suggests that we do some data augmentation work.

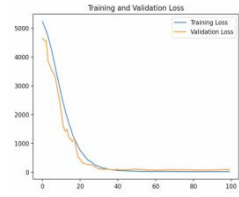
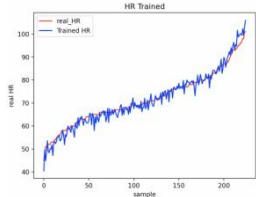
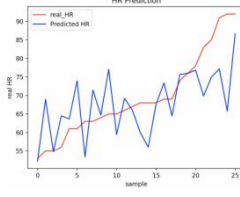
After looking up some of the material, there are many different methods to augment data like flipping, rotation, translation, cropping. All these methods are dedicated to image augmentation. For 1D time series, a possible solution is adding Gaussian and the other is cutting out the time series using a sliding window.

We first tried Gaussian noise. By this method, We can remove some invisible features in the data which are unimportant and give some randomness to the input, which will reduce over-fitting to some extent. An example of original data and data with Gaussian noise added is shown in Fig 27.

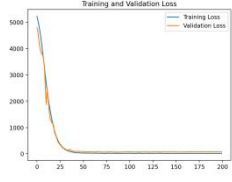
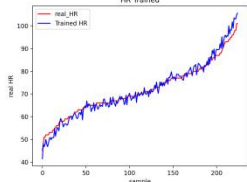
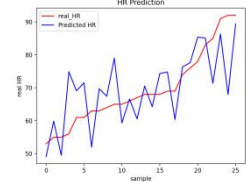
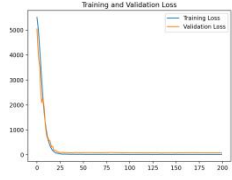
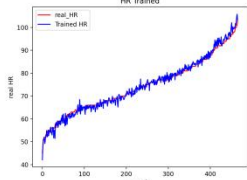
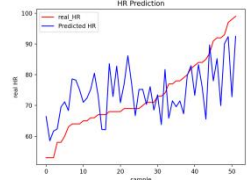
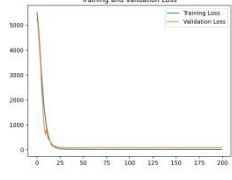
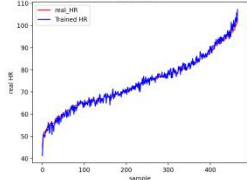
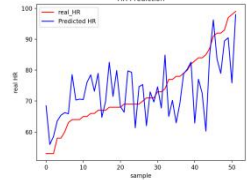


**FIGURE 27.** example of data with and without Gaussian noise. The orange line is the original data and the blue line is the data with Gaussian noise.

We've tried many different SNRs and finally find when  $SNR=5$ , the result better than others. After adding Gaussian noise, the MSE of the test set is 79.91 and the MAE is 7.12. we can observe that the MSE decreased while the MAE increase compared to without Gaussian noise. This result told us though our prediction's mean absolute error increased, the deviation of predicted HR from the real HR decreased, further proving that the prediction follows the trends better.

Modification Of model	Loss during training	Performance on train set	Performance on test set	Data set
Using Leaky ReLU				Good 30s



Adding Gaussian noise				Good 30s
Using Leaky ReLU				All 30s
Adding Gaussian noise				All 30s

**Table 1** Result of cornet

So after these two comparisons, we test our model on all 518 data sets, try to see the performance on it. The result of our model can be found in Table 1. The performance on all data is not so good and the trend on the test set is worse than just on good data. The result of MSE and MAE can be found in Table 2.

Modification Of model	Mean Squared Error on test set	Mean Absolute Error on test set	Data set
Using Leaky ReLU	85.6	6.93	Good 30s
Adding Gaussian noise	79.91	7.12	Good 30s
Using Leaky ReLU	98.65	8.25	All 30s
Adding Gaussian noise	90.03	7.66	All 30s

**Table 2** Statistic results

After all the final trials and modifications, we found that there was a strong correlation between the performance of the model and how the quality of the data set. Although more data were used at the end, the results were worse because the data used were not particularly good themselves. And we can conclude that adding Gaussian noise would do

---

better to our model for both good data set and all data set.

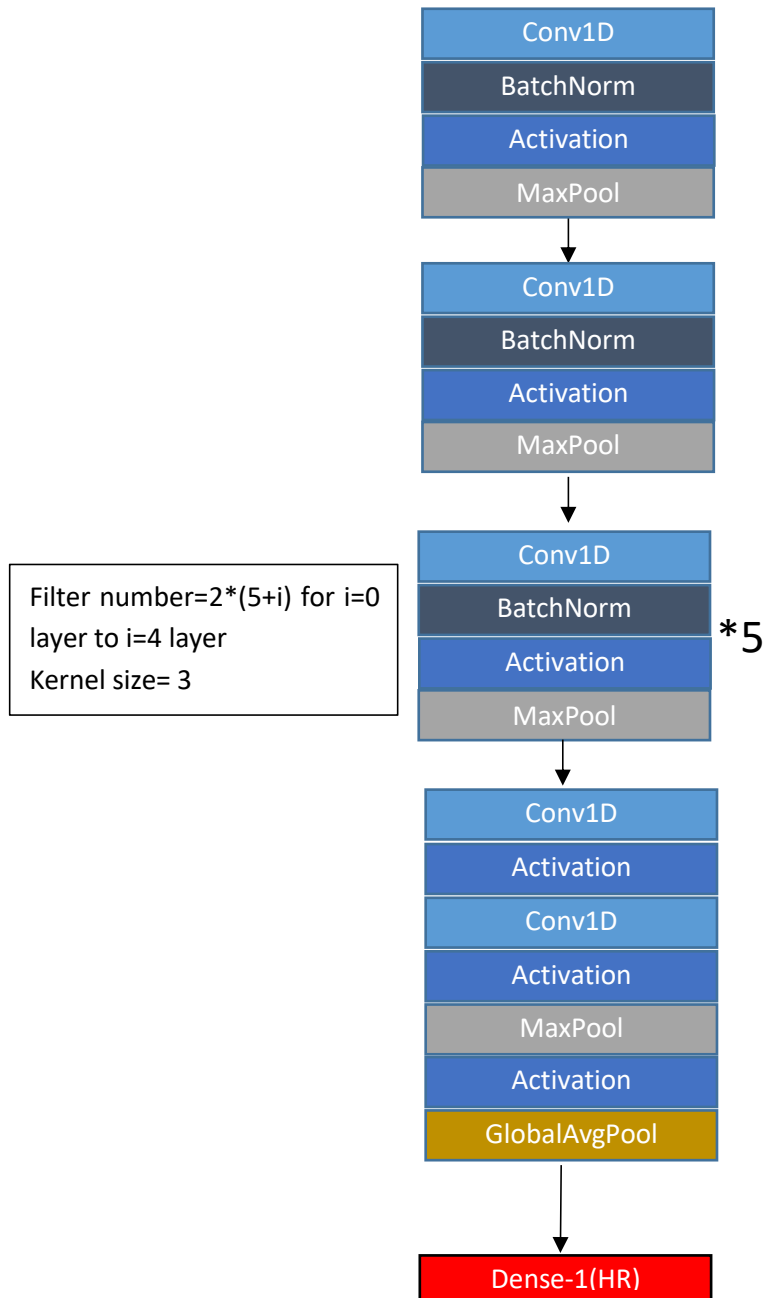
There is still a lot of work to be done on this CorNet, such as experimenting more extensively with the effect of adding Gaussian noise on the data and tweaking the size and number of convolution kernels. Whether more convolutional layers could be added, etc. The results we have obtained so far are only a small fraction of them, and our work could be more detailed and accurate.

## 5. Yuan xiaoran

### 5.1 Modified Cornet

Except for corNet, Dr.Liang has searched for many other possible network architecture that might suit our project. They provided us a new model modified from another network architecture for detecting heart rate from ECG signal, we would call it Modified Cornet.

This architecture only uses 1D CNN and removed LSTM part. Dr. Liang suggests there's many things we can try to tune the model. One is the loss function. The benchmarks of literature typically use MAE and MSE to evaluate the result. Previously, we used MSE as loss function and we can explore more about this. Another part we can do better is the optimizer. We used Adam before and actually some other powerful optimizer can be used to train our model more quickly and easily.



**FIGURE 28.** The architecture of Modified Cornet

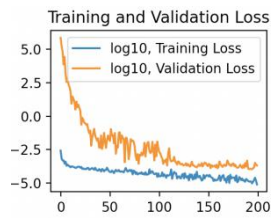
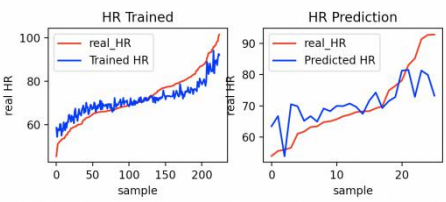
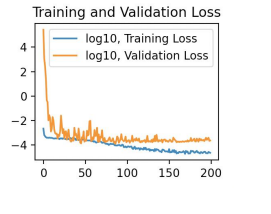
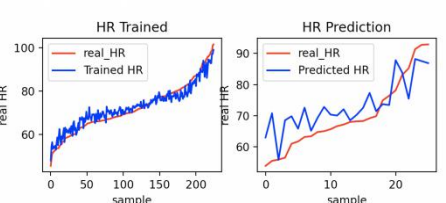
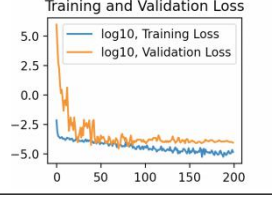
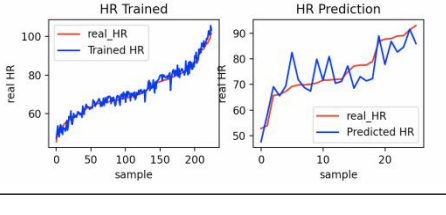
For a problem that we want MAE and MSE of the result to be both as small as possible, Huber loss might be an excellent tool for it. Huber loss is more like a combination of MSE and MAE.

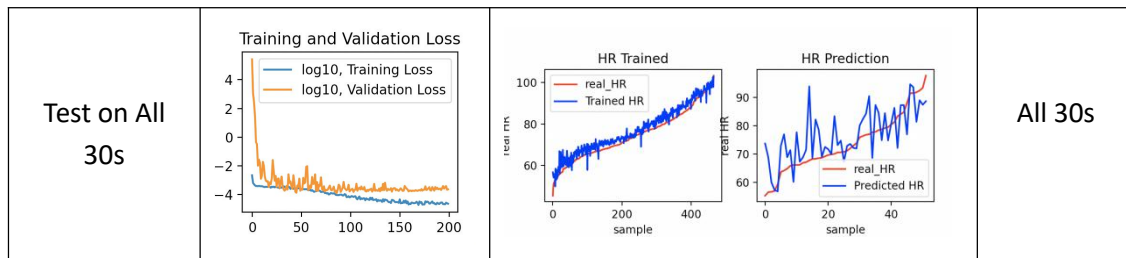
$$L_{\delta}(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{for } |y - f(x)| \leq \delta, \\ \delta |y - f(x)| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$

The Huber loss is less sensitive to outliers in data than the Mean Squared Error. It's basically an absolute error, which becomes quadratic when the error is small. The hyper-parameter, delta can be tuned to decide how small that error has to be to make it quadratic. Huber loss approaches MSE when  $\delta \sim 0$  and MAE when  $\delta \sim \text{infinite}$ . So we changed MSE with Huber loss as our loss function.

As for optimizer, at the beginning we used RMSprop for Cornet because the paper[23] suggests this has been tested good enough for the Cornet model and the experience from material says it indeed performs well for RNN( we uses LSTM in Cornet). Since we have moved from the Cornet model to only the 1D CNN model. I tried Adam optimizer which is the combination of RMSprop and momentum. The result shows that when using the RMSprop optimizer, training loss decreases more steadily and but slowly than Adam optimizer, the final converged points are almost the same. When we compare Nadam and Adam optimizer, the converge time is almost the same but Nadam can achieve more smooth optimization than Adam. We choose Nadam at last.

And this time, to better observe the changing process of training loss and validation loss, we decide to plot it on log scale so fluctuations can be observed more clearly.

Modification Of model	Loss during training	Performance on train set and test set	Data set
-			Good 30s
Adding Gaussian noise			Good 30s
Reduce Learning rate during training			Good 30s



**Table 3. the result before and after modification**

When I use the model on a good 30s data set, the result is relatively good than Cornet(The result after change the activation function can be found in Table 3), the MSE is about 56.48 and the MAE is about 5.79. Then by intuition from Cornet, I try to generate a signal with noise SNR=5. The result is slightly better with MSE equals 44.11 and MAE is about 5.43

However, as we looked into the training loss and validation loss, I found in both of them, there are some fluctuations during training. This could be a problem generated by the wrong choice of the learning rate, so I implement a callback function to reduce the learning rate during training every time by a factor of 0.9 when validation loss does not decrease for 4 epochs. The result after using it would be better, The MSE is reduced to 31.52 and MAE is 4.42 in this case. It's obvious the trend follows better with the real ones and fluctuations while training is reduced.

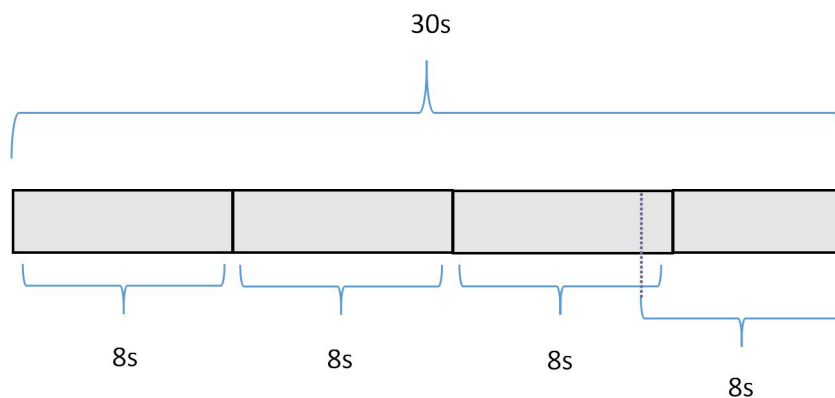
We also test this model for All 518 data sets. The result goes worse a little bit compared to using good data set. The MSE of it is 64.39 and MAE is 6.11. The performance on the training set becomes worse and while the performance on the test set could still follow the trend.

So, the overall performance is better than Cornet architecture.

Modification Of model	Mean Squared Error on test set	Mean Absolute Error on test set	Data set
-	56.48	5.79	Good 30s
Adding Gaussian noise	44.11	5.43	Good 30s
Reduce Learning rate during training	31.52	4.42	Good 30s
Test on All 30s	64.39	6.11	All 30s

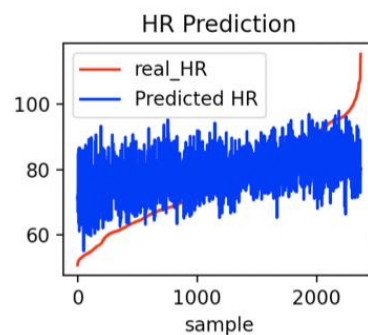
**Table 4.** Statistics results

So, after modification, this model is relatively good enough for further experiment. We decide to try to transfer this model to 8s data set. Firstly, we directly cut the 30s into four 8s-pieces with 2s overlap in final two pieces and use same ECG label for four segments:



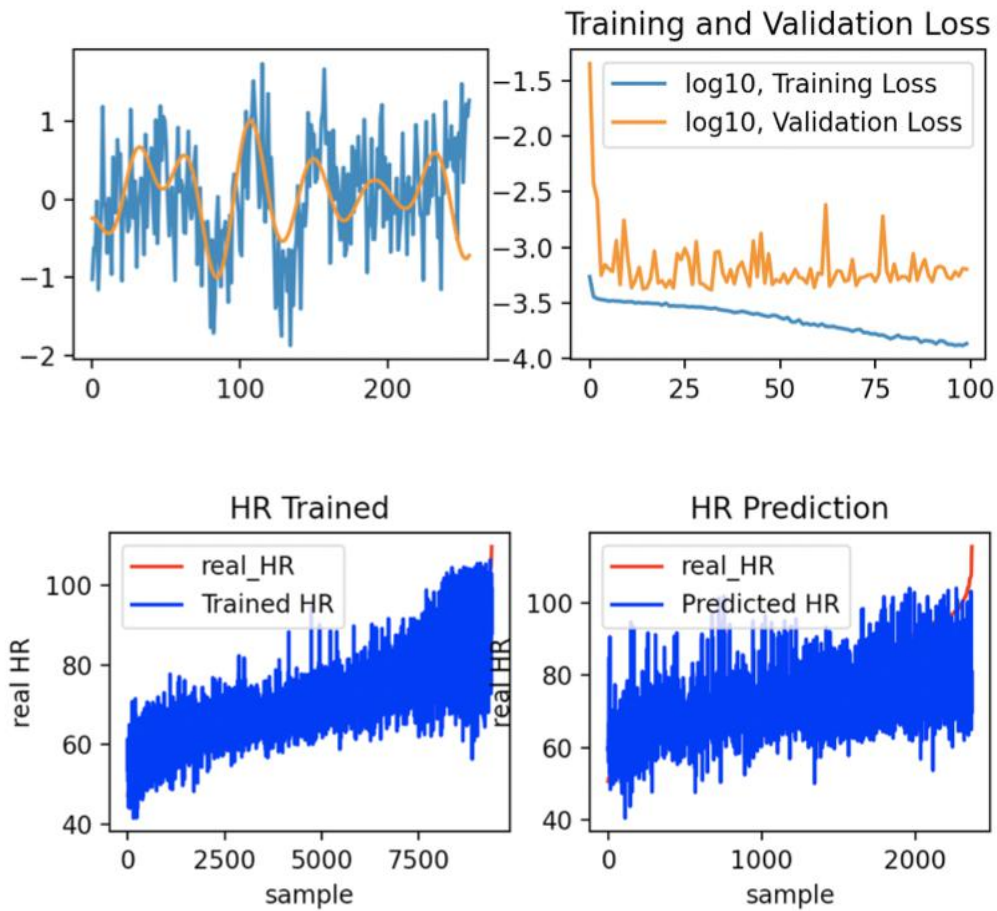
**FIGURE 29.** Cut 30s data into four 8s-pieces.

When we feed this 8s data set into our model, the result is not good and the prediction of heart rate is more like a Gaussian noise.



**FIGURE 30.** The heart rate prediction using 8s-pieces

After this, the Dr. Liang provides us with 8s data created by using sliding window with 1s step on All-30s data set( about 11753 data in total). We change our data set to this and because there's many overlaps in data set, we need to forbid overlapping in test set and training set. So we use video names to split our training set and test set. Then we feed our data into the model.



**FIGURE 31.** The result of modified cornet for All 8s data

From the result, we can see the training loss is steadily decreasing, but for the validation loss, the result converges so fast and it vibrates a lot. This is quite strange because as we using the 8s sliding window, we got more data, so we would expect better performance for our model. After discussion with our supervisor and searching for materials, we think this is might because our thoughts to use the same model for 30s to predict 8s time series is wrong. A good model for 8s data might inherently perform badly on the model for 30s.

For this part, many unfinished work is reserved for us to further improve our model, for model fed with 30s data, although we have done many modification and experiments to figure out best hyper-parameters, we didn't do it in a comprehensive way, may be after some trying, we can use grid search to find best parameters for our model. As for model fed with 8s data, we might need to change the model structure to some extent or do transfer learning from 30s-8s to make our model more suitable for 8s data.

## 5.2 2D-machine learning network

Finding heart rate is the same as identifying the heart rate frequency in nature. So after trying to generate heart rate from the original 1D signal, we'd like to think that whether

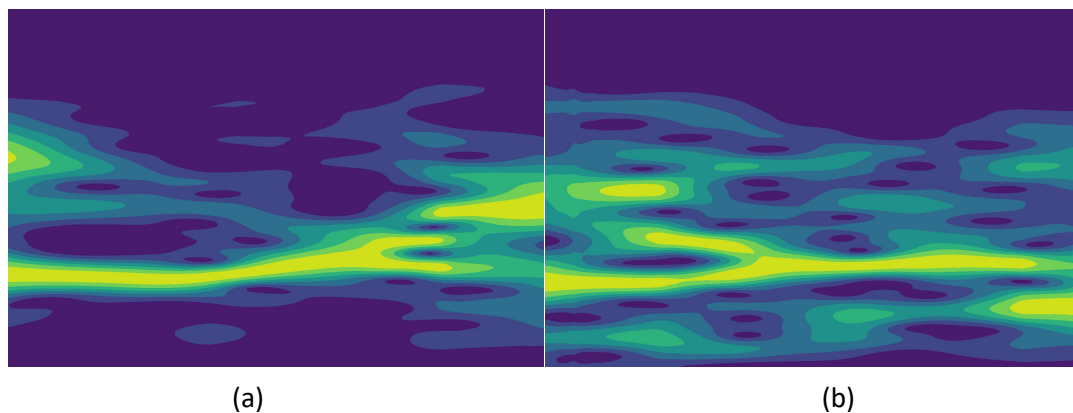
---

extracting the frequency feature of the IPPG signal and then feed it into the neural network may outperform the 1D CNN model. That's quite a big possible solution for heart rate detection.

Once we decide to use this method, we need to determine how to represent the frequency feature of the 1D IPPG signal. From the suggestion of Dr. Liang, we think the spectrogram might be a great tool to use. A spectrogram is a visual representation of the spectrum of frequency of a signal, it's more like a combination of frequency domain and time domain to reveal the frequency feature and at the same time its variation during the time. Spectrogram has widely been used for tone classification and speech synthesis and so on. For its internal advantage of finding the frequency trend for time series, I think indeed this would be an excellent tool for our project.

According to the fact that human heart rate would typically at the range of 30 and 150, we can set the observed frequency range is between 0.5Hz~2.5Hz, then for all 518 samples, we use 8s STFT(Short Time Fourier transform) to compute and draw their spectrogram and store it into the local computer so that we can directly use them for convenience.

Then Dr.Liang said that the Coordinate axis and value may make most of the training of the model look for these two rules, so she provides us with a set of cut pictures without coordinate axis and value. We only need to match the pictures with ECG signals in the data set to make a corresponding label.



**FIGURE 32.** Examples of the spectrogram(the x-axis should be time and y-axis should be frequency). The frequency distribution is illustrated by color, brighter color means more power of this frequency during this short period of time. (a) is of good quality, (b) is relatively bad.

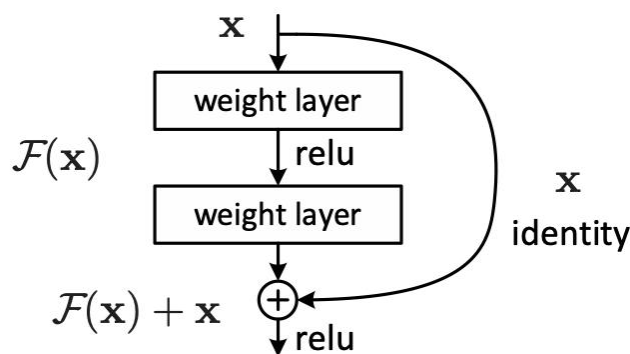
Thus, when having proper input data, three classical models, Resnet, Inception v3, Inception-Resnet v2 are experimented to figure out whether they can predict the heart rate from the spectrogram.



---

### 5.2.1 ResNet

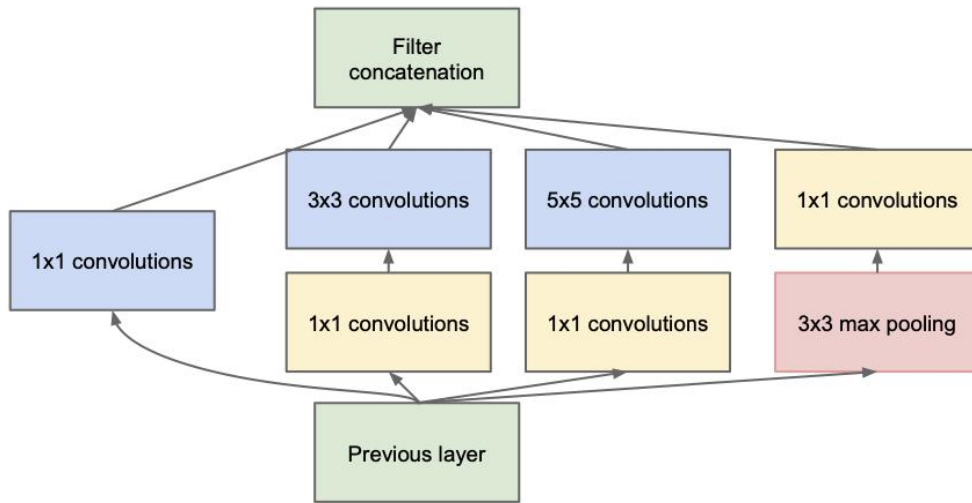
ResNet was proposed by He, *et al.* in 2015 [23]. In early image classification networks, the number of layers of the neural network is usually not too many, and it often shows that the more layers, the better the model classification effect. However, as the number of layers increases, the network degradation occurs: as the number of layers increases, the training set loss gradually decreases and then becomes saturated, and if the depth of the network continues to increase, the training set loss will increase. To solve this problem, ResNet introduces inter-layer residual skipping, which means that one or more layers are skipped, thus passing information to deeper layers of the neural network.



**FIGURE 33.** The residual block from [23], skip connection.

### 5.2.2 Inception v3

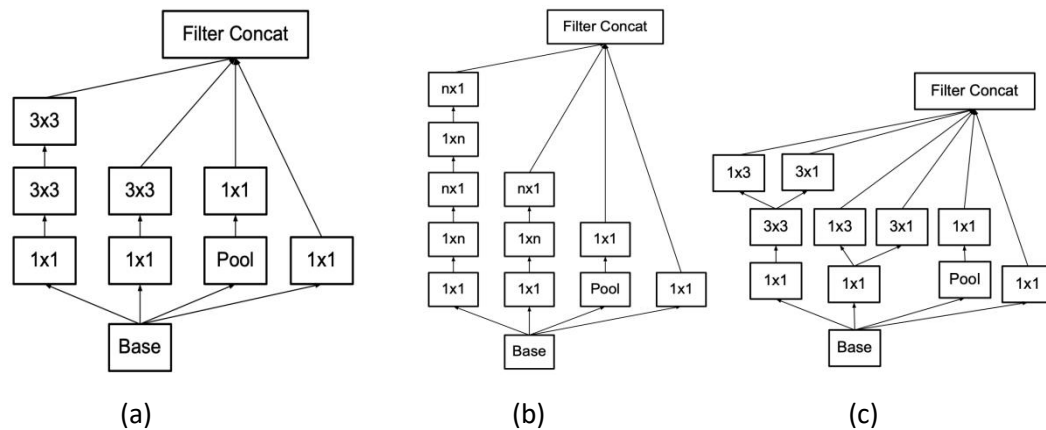
Inception v3 is an advanced CNN model proposed and modified by Google researchers. In image classification, the part that has a great influence on classification accuracy is called salient part. The positions and sizes of salient parts in different images are different. So this makes it very difficult to choose a suitable kernel size, given that Because the larger convolution kernel is good at learning global features, the smaller convolution kernel is good at learning local features. Improving network size used to be a very safe way to improve the model performance, but a larger size model might lead to over-fitting due to the lack of data and also the limited computing power of modern computers compel researchers to make their network moving from fully connected to sparsely connected architectures. To address the drawback of modern architecture, in the Inception network, they built what they called inception block[24]



**FIGURE 34.** Inception block from [26]

And the idea of the inception block is that if filters with multiple sizes are at the same level, the network will be slightly wider in nature, rather than deeper.

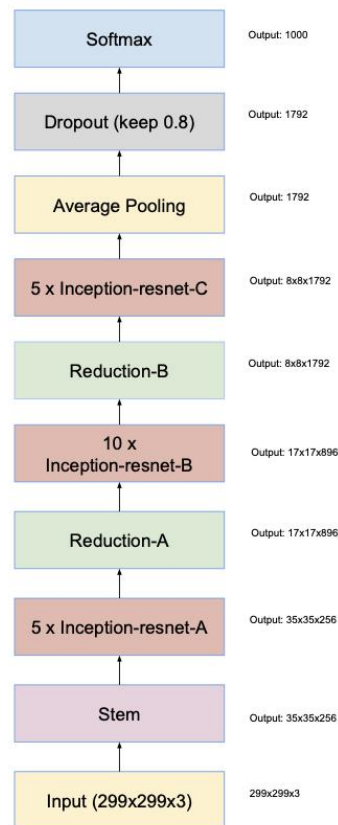
As for inception v3 that we used in our project, it's an improved version of the previous model. To further reduce the number of parameters[25], They proposed factorization. Change  $5 \times 5$  filter with 2  $3 \times 3$  filters or  $n \times n$  with  $1 \times n$  and  $n \times 1$ .



**FIGURE 35.** Three inception module with factorization. (a) is Inception modules where each  $5 \times 5$  convolution is re- placed by two  $3 \times 3$  convolution (b) is Inception modules after the factorization of the  $n \times n$  convolutions. (c) is Inception modules with expanded the filter bank outputs.[25]

### 5.2.3 Inception-ResNet v2

After trying the inception v3, Another further improved model, Inception-ResNet v2 comes to our eyes. Compared with inception v3, this model refers to the ResNet, which is a good network architecture that allows our neural network to go deeper without hurting the performance and also a feasible method to prevent our model to be too complicated to do the work. The Inception-ResNet v2 actually incorporates the idea of ResNet with inception network to further improve the speed of training and performance of classification[26].

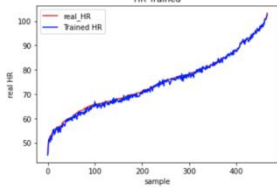
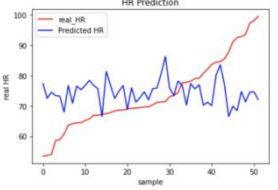
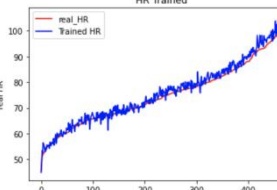
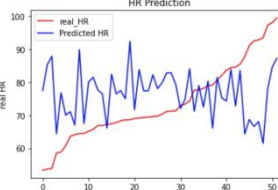

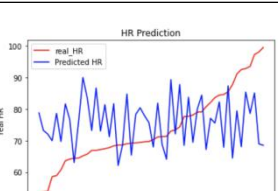


**FIGURE 36.** The architecture of Inception-ResNet. Inception-ResNet block A, B, C are different designed block. (details in [26]).

After fully understanding these models and built it as a regression problem with Keras, we adopted a method to only train top layers ( other layers only used to extract features so the pre-trained result is reserved, we just load the pre-trained weights), but the result is not so good, so we then open one or two blocks in the model for training to fine-tune the model. After the fine-tuning, the model can fit well with the training set.

From the result of using these three classical CNN models (Table 5), we found that all these three networks have a quite good fit on the training set, but the common problem is they all perform badly on the test set. This result indicates our model turns into an over-fitting state and its generalization ability is low. After discussion with our supervisor and Dr. Liang, the main problem here still lies in the data set. Only 518 images are not sufficient

for these three complicated to learn all the features. Typically, to predict one label for a classification problem might need thousands of images for the model to learn thoroughly.

model	Performance on train set	Performance on test set	Data set	MSE test	MAE test
ResNet			All	160.4 1	10.23
Inception V3			All	216.2 7	11.84
Inception-ResNet V2			All	189.8 7	11.48

**Table 5.** result of different image network

Luckily, Dr. Liang and her team are working on generating more image data from the original video data set. According to their description, if the whole program runs off, about 20000 images can be generated. So our future work can be done by using a more sufficient data set. This could be a game-changing improvement for our project. The problem is we might need to purchase some cloud computing servers to help us with training the model due to the limitation of our local computers.

## 6 Conclusion and future work

### 6.1 Conclusion

The algorithms for each of the three partial optimal results are:

1. extraction of physiological signals from the face: luv (3.3.3) with MAE of 8.6bpm.
2. Traditional signal processing methods: wavelet transform (4.1.3) MAE of with 7.3bpm.

---

3. Machine-learning methods: modified Cornet (5.1) with MAE of 4.42bpm.

Machine learning algorithms outperform traditional signal processing algorithms. This is because, for each of the three parts, each part is based on the best work of the previous part. For the traditional approach, we have referred to the literature and tested a large number of filters and smoothing algorithms in an attempt to find the optimal one; this part of the work is still relatively complete and adequate, and the results obtained are more consistent with the literature results. For the machine learning approach, which we discuss separately in the one-dimension and two-dimension cases, this part of the work still has greater potential.

## **6.2 Further work**

1. From the testing process, we have found that despite the performance improvements of advanced methods, the quality of the dataset contributes more importantly to the results, and adequate data can often alleviate the problem of undertraining. In the paper we use data from 5s to 35s in the original dataset, aiming to be consistent with the standards prevalent in academia. However, in engineering practice, we can also expand the dataset by two types of methods: a. splitting the 30s data into 8s windows, b. splitting more 30s data in videos of 2 minutes in length. Both of these methods involve the problem of overlapping datasets and can usually be considered as a class of migration learning problems, which require a more detailed investigation of the project, as well as more time spent on model tuning.

2. We refer to other literature and find that machine learning can also play an important role in the conversion of image signals into physiological signals, e.g. DeepPhys. often the signals in different regions are not identical and therefore a criterion for filtering the quality of the signals in different regions is needed. In addition, information about the motion between adjacent frames can also help to improve the prediction results. We anticipate that this new technical approach will further improve model performance.

---

## Acknowledgment

We would like to thank Professor John for his guidance and Dr. Zhao's team in Wuhan for their support, without which it would be tough to push forward our final project.

We would also like to thank each other for our teammates who have shown an excellent sense of responsibility during the project, which has made this collaboration impressive.

---

## References

- [1] Poh, Ming-Zher, Daniel J. McDuff, and Rosalind W. Picard. "Non-contact, automated cardiac pulse measurements using video imaging and blind source separation." *Optics express* 18.10 (2010): 10762-10774.
- [2] Noulas, Athanasios K., and Ben JA Kröse. "EM detection of common origin of multi-modal cues." *Proceedings of the 8th International Conference on Multimodal interfaces*. 2006.
- [3] Viola, Paul, and Michael Jones. "Rapid object detection using a boosted cascade of simple features." *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*. Vol. 1. IEEE, 2001.
- [4] Kamshilin, Alexei A., *et al.* "Accurate measurement of the pulse wave delay with imaging photoplethysmography." *Biomedical optics express* 7.12 (2016): 5138-5147.
- [5] Balakrishnan, Guha, Fredo Durand, and John Guttag. "Detecting pulse from head motions in video." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013.
- [6] Goudarzi, Reza Heydari, Seyedeh Somayyeh Mousavi, and Mostafa Charmi. "Using imaging Photoplethysmography (iPPG) Signal for Blood Pressure Estimation." *2020 International Conference on Machine Vision and Image Processing (MVIP)*. IEEE, 2020.
- [7] Verkruysse, Wim, Lars O. Svaasand, and J. Stuart Nelson. "Remote plethysmographic imaging using ambient light." *Optics express* 16.26 (2008): 21434-21445.
- [8] Jonsson, Annika. *Pressure sore etiology-highlighted with optical measurements of the blood flow*. Diss. Mälardalens högskola, 2006.
- [9] W. Wang, A. C. den Brinker, S. Stuijk and G. de Haan, "Algorithmic Principles of Remote PPG," in *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 7, pp. 1479-1491, July 2017, doi: 10.1109/TBME.2016.2609282.
- [10] Bousefsaf, Frederic, Choubeila Maaoui, and Alain Pruski. "Automatic selection of webcam photoplethysmographic pixels based on lightness criteria." *Journal of Medical and Biological Engineering* 37.3 (2017): 374-385.
- [11] Yu, Yong-Poh, *et al.* "Dynamic heart rate estimation using principal component analysis." *Biomedical optics express* 6.11 (2015): 4610-4618.
- [12] Yu, Yong-Poh, P. Raveendran, and Chern-Loon Lim. "Dynamic heart rate measurements from video sequences." *Biomedical Optics Express* 6.7 (2015): 2466-2480.
- [13] Wang, Dingliang, *et al.* "Detail-preserving pulse wave extraction from facial videos using consume-level camera." *Biomedical optics express* 11.4 (2020): 1876-1891.
- [14] Bal, Ufuk. "Non-contact estimation of heart rate and oxygen saturation using ambient light." *Biomedical optics express* 6.1 (2015): 86-97.
- [15] Zhan, Qi, Wenjin Wang, and Gerard de Haan. "Analysis of CNN-based remote-PPG to understand limitations and sensitivities." *Biomedical optics express* 11.3 (2020): 1268-1283.
- [16] Biswas, Dwaipayan, *et al.* "CorNET: Deep learning framework for PPG-based heart rate estimation and biometric identification in ambulant environment." *IEEE transactions on*

---

*biomedical circuits and systems* 13.2 (2019): 282-291.

[16] Viola, Paul, and Michael Jones. "Rapid object detection using a boosted cascade of simple features." *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*. Vol. 1. IEEE, 2001.

[17] Lienhart, Rainer, and Jochen Maydt. "An extended set of haar-like features for rapid object detection." *Proceedings. international conference on image processing*. Vol. 1. IEEE, 2002.

[18] Kumar, Mayank, Ashok Veeraraghavan, and Ashutosh Sabharwal. "DistancePPG: Robust non-contact vital signs monitoring using a camera." *Biomedical optics express* 6.5 (2015): 1565-1588.

[19] Biswas, D. , *et al.* "CorNET: Deep Learning framework for PPG based Heart Rate Estimation and Biometric Identification in Ambulant Environment." *IEEE Transactions on Biomedical Circuits and Systems* PP.2(2019):1-1.

[20] Chen, Weixuan & McDuff, Daniel. (2018). DeepPhys: Video-Based Physiological Measurement Using Convolutional Attention Networks.

[21] Yu, Z. , X. Li , and G. Zhao . "Remote Photoplethysmograph Signal Measurement from Facial Videos Using Spatio-Temporal Networks." (2019).

[22] D. Biswas *et al.*, "CorNET: Deep Learning Framework for PPG-Based Heart Rate Estimation and Biometric Identification in Ambulant Environment," in *IEEE Transactions on Biomedical Circuits and Systems*, vol. 13, no. 2, pp. 282-291, April 2019, doi: 10.1109/TBCAS.2019.2892297.

[23] He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[24] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A., 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

[25] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z., 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818-2826).

[26] Szegedy, C., Ioffe, S., Vanhoucke, V. and Alemi, A., 2017, February. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 31, No. 1).

[27] Hayashi, Takahiro, and Tatsuya Ooi. "Estimation of heart rate during exercise from a photoplethysmographic signal considering exercise intensity." *Sens Mater* 28.4 (2016): 341-348.

[28] Fukunishi, Munenori, *et al.* "Non-contact video-based estimation of heart rate variability spectrogram from hemoglobin composition." *Artificial Life and Robotics* 22.4 (2017): 457-463.

[29] <https://github.com/habom2310/Heart-rate-measurement-using-camera>