CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Enhancing a Deep Learning Camera-Based Approach for Heart Rate Detection in Vehicles Using Non-Functional Requirements

Bachelor of Science Thesis in Software Engineering and Management

Anton Golubenko
Akuen Akoi Deng

Department of Computer Science and Engineering
UNIVERSITY OF GOTHENBURG
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024

CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Enhancing a Deep Learning Camera-Based Approach for Heart Rate Detection in Vehicles Using Non-Functional Requirements
Exploring how enhancing a state-of-the-art convolutional neural network model by using non-functional requirements can be achieved to potentially improve heart rate detection in a vehicle environment

Supervisor: Tayssir Bouraffa
Examiner: Christian Berger

University of Gothenburg
Chalmers University of Technology
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering
UNIVERSITY OF GOTHENBURG
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024

Enhancing a Deep Learning Camera-Based Approach for Heart Rate Detection in Vehicles Using Non-Functional Requirements

1st Anton Golubenko
Department of Computer Science and Engineering
University of Gothenburg
Gothenburg, Sweden
gusgoluan@student.gu.se

2nd Akuen Akoi Deng
Department of Computer Science and Engineering
University of Gothenburg
Gothenburg, Sweden
gusdengak@student.gu.se

***Abstract*—Remote detection and monitoring of vital signs, such as heart rate, using remote photoplethysmography in a vehicle environment is a researched method to detect drivers' sudden illnesses and improve road safety. Deep learning models have shown their efficiency in detecting heart rate in a vehicle environment. However, improving their performance and evaluating them becomes a challenge due to the complexity of such models. Moreover, using non-functional requirements to evaluate the quality of these models presents a challenge since there is limited understanding and research on how they can be applied to deep learning models. In this study, we explore how a state-of-the-art camera-based deep learning model can be enhanced by identifying relevant non-functional requirements, such as explainability, reliability, and robustness, based on the challenges posed by detecting heart rate in the vehicle environment. Experiment research method was chosen for this study to compare and evaluate pre-enhanced and enhanced versions of the same model. We improved the feature extraction of the model based on the signal-to-noise ratio evaluation metric, though the model's generalizability decreased. Based on the process of enhancement and evaluation, we provide recommendations to software engineers concerning what to potentially expect when working with such models. This study presents a different approach to enhancing a deep learning model, particularly using non-functional requirements to evaluate the model's quality and performance, highlighting their importance in the evaluation and enhancement process.**

***Index Terms*—Remote photoplethysmography, software engineering, requirement engineering, advanced driver assistance systems, driver monitoring systems, non-functional requirements.**

I. Introduction

Each year, road accidents, including car crashes, occur due to various factors, such as driver fatigue, drowsiness, or health-related problems that could have been detected earlier [1]. A study by Skyving et al. showed that one-third of all the car crashes of drivers aged 50 and above in the period of 2010 to 2019 in Sweden were caused by sudden acute disease incidents, the majority of which are cardiovascular-related [2]. In 2022, there were a total of 227 deaths in only road traffic-related accidents in Sweden, of which 51% were drivers of four-wheeled vehicles [3].

Advanced systems are being developed to help drivers drive more safely, improving road safety and potentially saving more lives. Advanced Driver-Assistance Systems (ADAS) and Driver Monitoring Systems (DMS) are being researched and implemented to aid in reducing road fatalities [4], [5]. Due to rising demand from both car manufacturers and regulators, a key focus area in developing these systems is developing a software solution that detects and monitors drivers' vital signs remotely [6].

As a result, deep learning camera-based methods are being researched and used to improve the performance of contactless methods for remote heart rate detection in vehicle environment [7]. Chen & McDuff have introduced a deep convolutional neural network-based method to detect a driver's heart rate [8]. The results showed that the technique was superior to the compared state-of-the-art methods in the study and demonstrated promising potential as an alternative to conventional methods researched in the remote heart rate monitoring field [8], [9].

With the growing presence of machine learning (ML) in software development, it becomes relevant to know when an ML model, such as a deep learning camera-based model, can be integrated into a software system [10]. By their design, deep learning models are black-box algorithms that are automated to train and learn on a provided dataset. In contrast, traditional software applications are coded and tested by software developers. Therefore, traditional software testing techniques and code review cannot be applied to testing and evaluating deep learning models similarly [11]. Due to the black-box characteristic of deep learning models, where the process of inference from provided data lacks transparency, there is a question about the trustworthiness of such models. To what extent such ML models can be trusted, especially when it comes to the model being applied in a vehicle environment, is still an active research topic [11].

Non-functional requirements (NFRs), also known as quality attributes that apply to a software application, may differ from those of deep learning models, where, for example, validation accuracy often measures the model's success, and other quality

attributes are frequently overlooked [12]. Different quality attributes and their challenges for ML are researched, such as fairness, security, safety, and privacy, to name a few [13]. Furthermore, the researched quality attributes relevant to understanding black-box ML models and their application challenges in the automotive industry are explainability, robustness, and reliability [10], [14].

The explainability quality attribute of an ML model is the degree to which the processes inside a model and its predictions can be followed and understood by its observers. Understanding the model's inner workings, including any erroneous performance of the model and ways of improving it, can be insightful and could help understand its predictions [15]. Although the definition of robustness of the model can vary based on the ML domain and application, it generally measures its ability to withstand unexpected input, such as unexpected noise in the input frames in the case of heart rate monitoring in vehicles [10], [16]. Meanwhile, reliability can be defined as the predictability of model behavior across the variation of data it is trained on [11]. The reliability would also indicate how generalizable and consistent its prediction results are based on different training data.

Quality attributes can aid in developing deep learning models since the data they are trained on dictate their behavior. However, there is currently a lack of understanding and techniques of how quality attributes can contribute to the field of ML, which also concerns developing and enhancing deep learning models in the context of quality attributes [13]. This introduces a gap in the application of NFRs for ML model evaluation and enhancement, which needs to be addressed to be able to understand how different quality attributes can be used to enhance ML models.

This study was conducted as part of pre-study for developing Video-based Driver Condition Monitoring for Safe Driving (ViDCoM). The purpose of this study is to explore the use of quality attributes such as explainability, robustness, and reliability in evaluating and enhancing a deep learning camera-based model. Our motivation for choosing these quality attributes for the study is based on the unpredictable conditions of the vehicle environment and the nature of the model to be enhanced. Firstly, the degree of the explainability of the model and our understanding of it will determine to what extent we can enhance it. Secondly, enhancing the model in terms of its robustness to excessive noise that can appear in a vehicle environment could result in better confidence in the model prediction under unexpected circumstances. Lastly, the reliability of the model could indicate how consistent the model prediction is under varied training setups and how generalizable the model predictions are.

We will train and test our deep learning model on the MR-NIRP Car dataset collected in [17]. Using the results from our model, we will compare the performance of our approach with the pre-existing solutions in the literature by conducting an experiment and using evaluation metrics, such as mean absolute error (MAE), root mean square error (RMSE), mean absolute percentage error (MAPE) and signal-to-noise ratio (SNR). Additionally, we will use a literature review to collect current knowledge of the field.

The sections of this thesis work are structured as follows: Section I states the problem and purpose of the research; Section II presents the research in the field of ML and RE, as well as contactless heart rate detection using deep learning models; Section III defines the research questions, research methodology and the plan to conduct enhancement and evaluation of a deep learning model; Section IV presents the results from this study in form of answers to the research questions; Section V discusses the achieved results, how they are connected to related research and future work; Section VI summarizes the findings and concludes the research.

II. Related work

A. Background

1) Remote photoplethysmography and types of cameras: Different methods have been used to detect and monitor the driver's heart rate. Rouast et al. discuss the use of remote photoplethysmography (rPPG), which is a non-invasive method used to detect blood volume changes in the reflected light from the skin using different cameras such as RGB or Near-Infrared (NIR) [9]. An RGB camera uses three channels: red, green, and blue, which represent the face of a person in the RGB spectrum, with changes in the blood flow in the areas of the face. As the name suggests, an NIR camera uses a Near-Infrared spectrum, in which the light from the camera can go deeper into the skin than the visible light emitted from an RGB camera [9]. After extracting the rPPG signals, the signals are then converted to heart rate based on the peaks of impulses in the signals, as is shown in Fig 1.

2) Requirements engineering and machine learning: The integration of machine learning in most aspects today has led to disruption and changes in the traditional software development landscape [10]. In the requirements engineering (RE) field, researchers have raised concerns over the need for a new approach when it comes to applying requirement engineering practices such as requirement elicitation, verification, validation, and traceability in the context of machine learning software development [13], [18], [19].

Machine learning-based software systems, like other software systems, are required to fulfill requirements in terms of functionality and quality. However, identifying which quality attributes are relevant for machine learning model development and the lack of a clear understanding of NFRs for ML presents challenges to machine learning and software developers. RE methods could elicit quality attributes for ML that are concerned not only with the model's accuracy but also other qualities that would satisfy the requirements of relevant stakeholders and further enhance the understanding of the model and how it is trained on data [13].

B. NFRs at the forefront towards improving deep learning approach

Villamizar et al. showed that there is a lack of accepted methods in the field of applying RE in the context of devel-

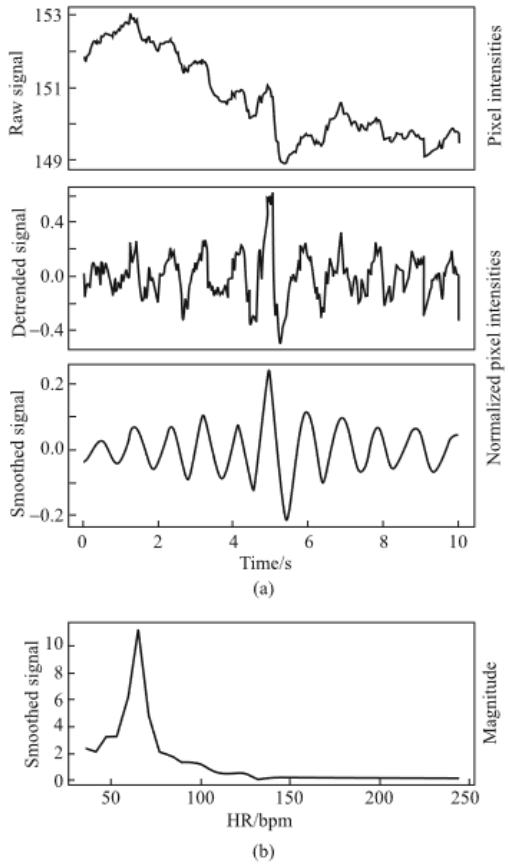

Fig. 1. Illustration of (a) rPPG signals extracted from a face using the green channel of RGB (b) heart rate derived from the same signals as illustrated in [9]

oping ML-based software solutions according to quality attributes and how the quality attributes should be used properly in the context of ML [13]. In the same systematic mapping study, Villamizar et al. also identified quality attributes such as explainability, data quality, fairness, transparency, and reliability as some of the most frequently considered for machine learning models [13]. Despite the lack of accepted guidelines on when and how NFRs should be elicited, documented, and validated, they are still considered essential and critical for a successful software system [19].

Horkoff discusses that ML software's quality attributes, or in other words, NFRs, differ from those used to evaluate conventional software [20]. A novel type of quality attributes that would be more relevant for ML-based software, such as fairness and transparency may arise. According to Horkoff, it is important to bring NFRs for ML to the forefront, facilitate early consideration, definition, and trade-off analysis, and, more importantly, raise awareness of ML performance regarding such qualities [20].

C. Machine learning model evaluation

When evaluating deep learning models, the focus is placed on validation accuracy as the general measure for testing and validation. However, Hyatt & Lee argue that higher accuracy of a neural network model validation does not necessarily indicate an absence of defect nature in the classification process of the data it is trained on [12]. They propose that other performance measures should be added to the requirements for developing robust deep learning models [12]. Closely similar to Hyatt & Lee proposal, in a different study, Trenquier et al. take a holistic approach to validate a model using quality attributes by evaluating the impact of each attribute on the model's performance [21]. By explicitly capturing the aspects in data regarding requirements and environmental assumptions, Trenquier et al. could conduct quality evaluation and potential improvement granularly [21]. However, the core challenge in this approach lies in the lack of adequate knowledge about the effect of the attributes on the performance of an ML model before evaluation [21].

D. Deep learning camera-based heart rate detection methods

Some deep learning methods that predict heart rate from rPPG signals in a laboratory setting are PhysNet, DeepPhys, and rPPGNet [7]. Although It was shown that DeepPhys outperformed conventional methods in the past [8], as of more recent research, Ni et al. and Yang et al. demonstrated that PhysNet has the most accurate performance overall among the deep learning methods when it comes to measuring heart rate in low, medium and high-light variations [7], [23]. PhysNet uses a 3D convolutional neural network (CNN) that also takes spatial and temporal information for segmentation of skin from an image and has a better performance than a more common 2-dimensional CNN architecture that is, as an example, implemented in DeepPhys [23].

Compared with conventional methods of detecting heart rate, like CHROM, ICA, and POS [17], which are not deep learning-based, deep learning methods perform with varying reliability when determining heart rate. Wang et al. proposed a physiological-based approach using a deep learning method, which was tested on the MR-NIRP Car dataset [24]. As a result, there is still room for improvement in both physiological and deep learning approaches regarding cleaning noise from the rPPG data [24].

A CNN model was suggested by Chiu et al. to predict heart rate from rPPG based on five different datasets [25]. The results show that the model performs better on heart rate prediction on most datasets than the conventional methods, based on both RGB and NIR types of recordings [25]. Lee et al. showed that deep learning methods are superior to conventional methods due to their ability to adaptively recognize and learn patterns and generalize features compared to conventional methods [26].

The related research introduces a gap, such as inconsistency in the performance of the deep learning methods that detect heart rate in a vehicle environment, as compared to each other and conventional methods. Key factors such as camera type, lighting, head movement, and dataset diversity significantly affect the methods' reliability. In addition, we need to consider system requirements, such as quality attributes, when designing ML solutions. Further evaluation and testing

of deep learning models beyond accuracy is necessary. More research is needed on software-based deep learning camera methods for heart rate detection in vehicle environments, using RE techniques for ML systems, and evaluating quality-based ML model models. Our contribution aims to address this gap by introducing an enhanced deep learning model for more accurate and robust heart rate detection in vehicle environments by identifying and focusing on relevant NFRs to evaluate and enhance the performance of the deep learning model for detecting heart rate.

III. Research Methodology

This research will involve conducting an experiment. According to Stol & Fitzgerald, an experiment aims to investigate, evaluate, or compare techniques, practices, processes, or approaches within certain conditions or settings, for example, by controlling and changing their variables and observing how that process affects the results [27]. The main reason for choosing this research method to tackle this research problem is to investigate, evaluate, and compare the outcome of using NFRs with a deep learning approach to detect and monitor heart rate in a vehicle environment compared to other deep learning approaches. The research can act as a roadmap for software engineers in identifying relevant challenges to an ML model, defining relevant quality attributes depending on the end goal, and enhancing the model based on these quality attributes. Additionally, we will use a literature review to learn about current research and challenges in using deep learning camera-based models for detecting heart rate in vehicle environments and the current state of requirements engineering for developing ML-based software systems.

A. Research questions and hypotheses

Research questions

To achieve the aims of this research, we will address four research questions.

RQ1: What are the challenges associated with consistently monitoring heart rate in diverse driving conditions?

With this research question, we aim to identify the challenges of monitoring heart rate in a vehicle environment, and see how these challenges can affect the ability to detect consistent, reliable heart rate from a subject or how they introduce different anomalies and noise in the input data affecting the reliability and robustness of the system. This, in turn, makes it necessary for the system to require some degree of adaptability and robustness to handle the variations. This will help demonstrate how these challenges affect the overall quality of heart rate detection and, therefore, can be linked to and reinforce our justification for choosing explainability, robustness, and reliability to enhance the deep learning model based on these quality attributes.

To answer this research question, we will conduct a literature review of the current research in remote heart rate detection in a vehicle environment, by using a keyword search of the relevant literature.

RQ2: How can explainability contribute to understanding and enhancement of a deep learning approach?

Using this research question, we want to investigate how the explainability of a model can be used to help understand the inner workings of the deep learning model that we will work with. We will explore common methods used to increase the explainability of the model, and based on the results we get, we will assess how a better understanding of the model can contribute to the enhancement of the model.

RQ3: How do robustness and reliability quality attributes play a role in systematically evaluating and enhancing a deep learning approach?

In RQ3, we aim to understand how quality attributes such as robustness and reliability can be used to evaluate deep learning models and enhance their robustness to noise and reliability of the predicted results based on the variation of the training data. Our goal is to establish the evaluation of an ML solution using other equally significant NFRs beyond accuracy. It is worth mentioning that the definition of robustness can differ based on the application of ML [28]. However, robustness can be defined by feeding unexpected visual input to the model to test its resilience to the newly introduced noise. Moreover, evaluating the model based on reliability quality attribute could give more insight into how predictable and consistent the model's performance is based on different subjects in the dataset compared to other deep learning models.

RQ4: What recommendations can be provided to software engineers based on the process of enhancing a deep learning model?

By exploring which insights can be derived from enhancing an ML model based on NFRs, we aim to reflect on our study's findings and provide recommendations to software engineers working with ML models, particularly in the automotive domain. Moreover, we will cross-check the recommendations with the recommendations/conclusions from available studies in the fields of RE and ML.

Main hypothesis

We hypothesize that using NFRs such as explainability, robustness, and reliability to evaluate and enhance a deep learning approach to monitor heart rate in a vehicle environment will lead to a more transparent, consistent, and robust heart rate detection.

Null Hypothesis ($\mathbf{H}_0$): *The performance of the enhanced deep learning model based on robustness and reliability* is worse compared to the deep learning model before the enhancement.

Alternative hypothesis ($\mathbf{H}_1$): *The performance of the enhanced deep learning model based on robustness and reliability is better than the deep learning model before the enhancement.*

B. Research methodology to be used

In this experiment, we will use PhysNet, a CNN deep learning model [7], which will serve as our independent variable throughout the experiment. We will use quality attributes

robustness, reliability, and explainability as experimental treatments (levels) to guide us in the iterative process of systematically enhancing and evaluating the model. Eventually, using the enhanced model's results based on reliability and robustness as our dependent variables, we will compare these results to the results of the pre-enhanced PhysNet.

As part of systematically evaluating the PhysNet model, we will perform activities, the steps of which are stated below and in Fig. 3, each with a specific purpose for the whole process:

Data pre-processing - We will split the training dataset MR-NIRP into five different parts (folds), as shown in Fig. 2. Each fold will also be divided into training, validation, and test sets.

Model pre-evaluation - At this step, we will evaluate the PhysNet model before performing any enhancement. This evaluation will explore explainability methods currently available in the field of ML [29] that can provide additional information and understanding of the inner workings of PhysNet. After choosing a suitable explainability tool and technique for our case, we will focus on the explainability quality attribute to analyze and understand the architecture of PhysNet. This understanding will guide us when performing the enhancement.

Model training - With some understanding of the model's architecture as a result of using explainability in the pre-evaluation step above, in this stage, we will perform some enhancements on the model guided by what was learned from the explainability. We will proceed with the model's training as we continuously make iterative enhancements. The model will be trained on the five folds of the dataset created previously in the **Data pre-processing** step to facilitate the evaluation of the model on reliability in the next stage.

Model evaluation - After successfully training the model on all five dataset folds, we will evaluate the model based on reliability and robustness quality attributes and for any overfitting, inconsistent prediction, and level of generalization based on validation loss.

- *Reliability* - To evaluate the model based on reliability, we would test for model performance consistency across different subsets of the dataset with the k-fold cross-validation approach [30]. We will observe how the model performs on each of the created folds, and later, we will average the performance of all five folds to get a consistent and reliable overview of the results.
- *Robustness* - To test the model's robustness, we will create a sixth fold containing a test group of five subjects with only large head movements both driving and in the garage, testing the models based on the significant presence of noise. This will be used for testing how robust the model is and give us an understanding of how it adapts and handles datasets with significant noise.
- *Evaluation Metrics* - To evaluate the relationship between the NFRs and ML model, specifically measuring and evaluating the model's performance based on robustness and reliability, evaluation metrics will be used, such as MAE, RMSE, MAPE, and SNR, further described in Section III-B2.

Comparative analysis - At this final step, we will evaluate and compare the enhanced PhysNet with the pre-enhanced PhysNet, along with other state-of-the-art models [31] and observe how the enhancement process has affected the selected quality attributes. This step will help us evaluate and compare the reliability and robustness quality attributes beyond the performance accuracy of the models.

Our methodology's iterative enhancement process is a deliberate strategy for exploring how NFRs can be used to enhance a deep learning model and address its challenges. It will guide us in understanding how the selected NFRs can affect the ML model and, in turn, how the challenges of the specific ML model can affect the selected NFRs. This research context supports our research questions by providing the basis for learning the importance of using NFRs as a guide to working with an ML-based model.

1) Data collection: We will use the MERL-Rice Near-Infrared Pulse (MR-NIRP) dataset for our study, which was collected in [17]. A novel approach was used in collecting the dataset in a vehicle environment, particularly using a pulse oximeter as the ground truth, which is recorded together with the video recording of the participants. It is collected in this way to be compared with newly developed methods and to facilitate other research in discovering how photoplethysmography signals are affected by multiple noise origins [17].

The qualitative data, such as videos in different vehicle locations, weather conditions, types of head movement, gender of participants, presence of facial hair, and types of cameras, were collected to provide context and generalizability of the dataset. The different conditions where the data was collected had the following labels: driving large motion, driving small motion, driving still, garage large motion, garage small motion, and garage still. Both garage and driving large motion datasets present the most diverse challenges in a vehicle environment, such as large head movement and varying illumination when driving.

The quantitative data included the number of participants, the duration of the video recordings, pulse oximeter readings, and camera setup specifications. In our study, we will test and train our deep learning camera-based model on the videos where the vehicle is being driven.

The drivers were recorded in a garage with an idling vehicle and while driving around the city. Two different head movement scenarios were recorded in these driving conditions: small head movements and additional head movements, where participants moved their heads in a natural driving condition. Two individuals were female, and five had facial hair [17].

18 healthy individuals, ages 25 to 60, each with different skin tones, participated in the data gathering. One individual was recorded twice, during both day and night driving, resulting in 19 city driving recordings and 18 garage recordings. Four videos were taken at night and 14 during the day, with eight in sunny weather and six in overcast conditions. In total,

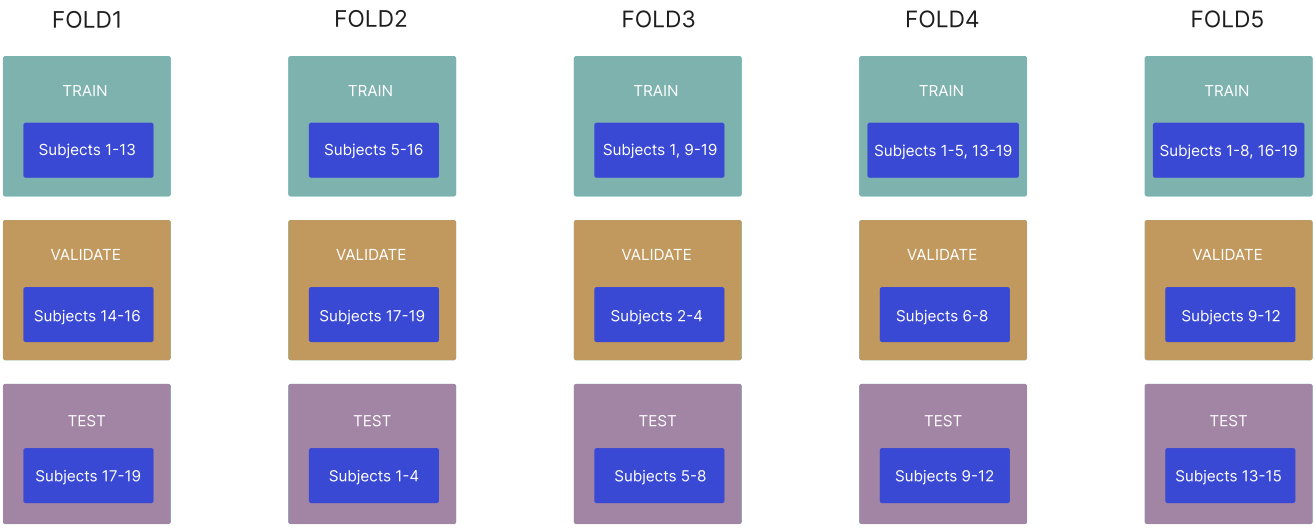

Fig. 2. Illustration of the five folds and how the subject groups are distributed

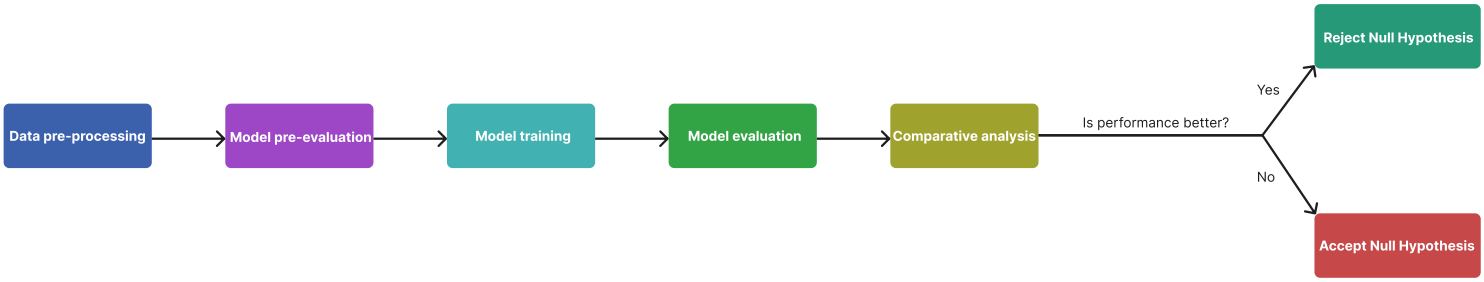

Fig. 3. Representation of how the experiment will be conducted

there are 190 video recordings in the dataset and 19 folders of subjects (individuals) [17].

Two cameras were mounted on the dashboard inside the vehicle: RGB and Near-Infrared (NIR). The RGB camera had 4.5 mm focal lenses, and NIR had 8 mm lenses. Additionally, the NIR camera had a 940 mm bandpass filter with a passband of 10 mm. Four illuminators were used on each side of a participant's face. Natural sunlight was used as an illumination on the RGB camera. 10-bit raw images with a resolution of 640 x 640 were captured at a 30 frames per second rate. Lastly, the garage recordings were two minutes long, and the city driving recordings were two to five minutes long [17].

2) Data analysis: To analyze data and compare our model's performance to other models and methods in our research, we will use evaluation metrics, such as MAE, RMSE, MAPE, and SNR. These metrics were chosen because they are used in the performance comparison and evaluation of camera-based state-of-the-art methods in heart rate detection in a vehicle environment [17].

The MAE metric will measure the mean absolute error between the predicted heart rate by our deep learning model and the ground truth heart rate reading. The MAE is suitable for our research since all errors are assigned an equal weight despite their magnitude [31]. This ensures that outliers or noise do not affect the evaluation, providing a more reliable measure of overall model performance. A lower MAE value indicates that model's predictions are closer to the ground truth.

The RMSE measures the average difference between the model's predicted heart rate values and the ground-truth heart rate values [32]. Since it closely resembles the normal standard deviation, it gives us the standard deviation of the residuals (prediction errors), which measures how far from the prediction's ground truth it is. This easily and intuitively shows how the model is performing. Low RMSE values indicate that the model fits the data well and has more precise predictions.

The MAPE metric measures the mean absolute percentage error between the predicted heart rate and the actual one. The MAPE is suitable for our research analysis since it expresses

the mean absolute error as a percentage regardless of whether the deviation from the ground truth was a positive or negative value [32]. The percentage makes it easier to compare different models' mean absolute error. The lower the MAPE, the better the model's performance.

The SNR metric would measure the signal-to-noise ratio obtained from the captured images. The higher the positive SNR values, the cleaner the reading of an rPPG signal, which can facilitate a more accurate reading of the heart rate [33]. This metric will be used as a base for our deep learning model, which will predict the heart rate from the images captured with the camera.

IV. Results

In this section, we will present the challenges that we have discovered from the literature review (RQ1), findings from enhancing and evaluating the PhysNet model based on NFRs (RQ2 and RQ3), and what recommendations we could give to software engineers working with ML models based on our findings (RQ4).

A. RQ1: What are the challenges associated with consistently monitoring heart rate in diverse driving conditions?

To identify and understand the key main challenges associated with consistently monitoring heart rate in a vehicle environment, we conducted a literature review of the currently available research on the remote measurement of vital signs in a vehicle environment. We searched for the available literature using the following keywords: rPPG in-vehicle environment, remote heart rate monitoring of drivers, deep learning and rPPG in-vehicle environment. We identified and reviewed ten different research papers, as seen in Table I, and the findings were consistent in this context. From these studies, we identified that varying illumination and motion artifacts were the main challenges encountered when remotely detecting and monitoring vital signs, such as heart rate, in a vehicle environment. Apart from the main identified challenges, different skin tones and vibration were also identified as a hindrance for heart rate monitoring, although they occurred in fewer research papers. When it comes to the understanding of how these challenges affect the deep learning models during the monitoring of heart rate, it is important to look at these challenges in the context of how relevant NFRs can be used to enhance the deep learning model to address these challenges of monitoring heart rate in such environment.

As was written in Section I, explainability, robustness, and reliability were defined for the study and can be seen in Table I alongside the identified challenges. From the point of explainability of the deep learning model, it is important to understand how the model works and which features it extracts so that it is more clear what needs to be done to address the identified challenges, such as varying illumination and excessive head motion. The robustness quality attribute is used to understand how well the model handles the unpredictable noise received when monitoring heart rate in the vehicle environment. It is especially relevant in this case since the varying illumination and excessive head movements introduce significant noise to the input of the model, and it is crucial to enhance the ability of the model to filter out that noise better. Lastly, the model's reliability is crucial when the model's predictability should be considered. The heart rate detection and monitoring should be consistent across different types of data being fed into it for training, especially when the model is used in the automotive domain and as part of a safety-critical system. The more reliable the prediction is, the more trust can be potentially put in the prediction of the model.

To conclude, it is crucial to use SE elicitation practices and domain analysis in the context of using NFRs for ML to identify relevant NFRs that could improve understanding and design of ML solutions to address relevant challenges. The importance of NFRs for ML lies in a systematic approach, whereby not only an ML model but also the problem it solves is analyzed. Thus, deriving the relevant quality attributes helps in a broader assessment of the model's performance.

B. RQ2: How can explainability contribute to understanding and enhancement of a deep learning approach?

Before commencing to enhance the model, we need to understand the architecture of the model and how the image data is represented within each convolutional layer and block. Fig. 4 represents the architecture of the PhysNet model before its enhancement. By analyzing the code of the model and using a model summary function from PyTorch [35], we got a more in-depth understanding of the model structure. The model takes an image as an input, which is represented as a PyTorch tensor, and feeds it through sequential layers. Each layer has a three-dimensional convolutional operation on the five-dimensional input tensor, which calculates pre-processed image features based on spatial and temporal data from the image, as briefly described in Section II-D. Each convolution is achieved by multiplying the input tensor with a smaller three-dimensional matrix tensor called a kernel. This operation reduces the tensor's dimensionality, which is standard practice when using CNNs on more complex and memory-intensive input data [36].

The five dimensions of the pre-processed image tensor are represented as ([batch, channels, length, width, height]). Batch is the number of image samples passed to the network; the channels are an RGB representation of the image, which has three channels in this case before the first layer, and the length, width, and height are the image's dimensions in three-dimensional space. An example of the passed image in the tensor form is the torch.Size([4, 3, 512, 72, 72]). The features extracted from the tensor are normalized with the three-dimensional batch normalization (BatchNorm3d) function, which uses mean and variance calculations on the input tensor to minimize the amount of distributed activations in the model, thus reducing the training time of the model [37]. Rectified Linear Unit (ReLU) function takes in the summed weighted output from nodes of the CNN and returns either zero if the output is negative or the output value itself if it is positive [38].

TABLE I
RESEARCH PAPERS (RP) USED, THE CHALLENGES IDENTIFIED AS PART OF OUR LITERATURE REVIEW, AND THE DERIVED QUALITY ATTRIBUTES

Challenges	Research Papers	Derived Quality attributes
Varying illumination and head movements	RP1 [5], RP2 [6], RP3 [17], RP4 [22], RP5 [23], RP6 [24], RP7 [25], RP8 [26], RP9 [33], RP10 [34].	**Explainability** - learning about the inner workings and how explainable are the components of the model to understand how it can be enhanced to address the challenges
Different skin tones	RP9 [33], RP10 [34].	**Robustness** - evaluating the model based on high-noise scenarios may bring insights of how well it performs in these scenarios, and enhancing it further to address the constantly changing conditions in the vehicle environment
Vibration caused by driving	RP1 [5].	**Reliability** - assessing the consistency of the model predictions may show how well it adapts to different data fed to it, which can be reflected in to what extent the predictions can be trusted

The output of the first layer is then propagated to a maximum pooling (MaxPool3d) layer of the network. Maximum pooling further reduces the dimensionality of the passed tensor and selects only the highest values in the feature matrix [36]. The layer sequence is repeated three times until it reaches layers eight and nine, followed by upsampling layers. The upsampling layers increase the spatial dimensionality of the image by using the convolutional transpose (ConvTranspose3d) function [39]. After the upsampling, the tensor is forwarded to an adaptive average pooling function (AdaptiveAvgPool3d), which adaptively (automatically adjusting the stride window) averages extracted features as compared to extracting the highest values in maximum pooling [40]. Lastly, the tensor is sent through the last layer, which consists of a convolutional block.

The analysis of the visualized PhysNet architecture gave us a better understanding of how the model is structured and which output shape each layer has. With this information, we could deduce how the input tensor's dimensionality changes throughout the CNN's forward pass. However, to further understand how the features are represented within the model, we have generated feature maps from each convolutional block [41]. Feature maps, in the case of the PhysNet model, represent the mapped features that the model sees at each convolutional block [42]. The feature maps are displayed from RGB channels and represent the facial features of a subject sitting in front of the camera, which are extracted to calculate rPPG signals. As shown in Fig. 5, the facial features and background noise are more prominent in the first convolutional block than in the other blocks. The model extracts rPPG signals based on the features of the face edges. Furthermore, with each layer of the PhysNet model, the complexity of the model increases due to the increase in the number of filters, which consequently capture more features from the input tensor. Thus, with each layer, there is less noise captured from the features, and the input tensor is smaller due to maximum pooling and convolutional operations [43].

This observation encouraged us to perform hyperparameter tuning by experimenting with the model's hyperparameters, such as the channel values of each convolutional block and the number of convolutional layers. Hyperparameters in a convolutional layer are convolutional filter size, padding, channel, and stride values, which define the structure of the convolutional layer [44]. We started tuning the channel values and observed how the evaluation metrics would change after the model was trained on all five folds. After increasing the channel values from 64 to 256 and adding a convolutional block at the end of the model, the model's performance degraded on all five folds. We assumed that by increasing the number of channels, the model would extract more features from an image and thus learn better based on the increased number of features. However, this was not the case. After conducting the experiment further, we have identified that implementing the Swish activation function:

$$f(x) = x \cdot \text{sigmoid}(\beta x) \tag{1}$$

which is proposed by Ramachandran et al., in combination with the ReLU activation function:

$$f(x) = \max(0, x) \tag{2}$$

increases the overall performance of the PhysNet model on folds one and five, as shown in Table IV [45].

Using feature maps to increase the model's explainability gave us insights into how the model perceives a subject's face. It also gave us ground for experimenting with channel values to try to extract more features from the image passed through the network. However, analyzing the different components in the model's architecture and researching different hyperparameters of the model has given us more valuable insights into enhancing the model. The factors that affect the explainability quality of the model are shown in Table II.

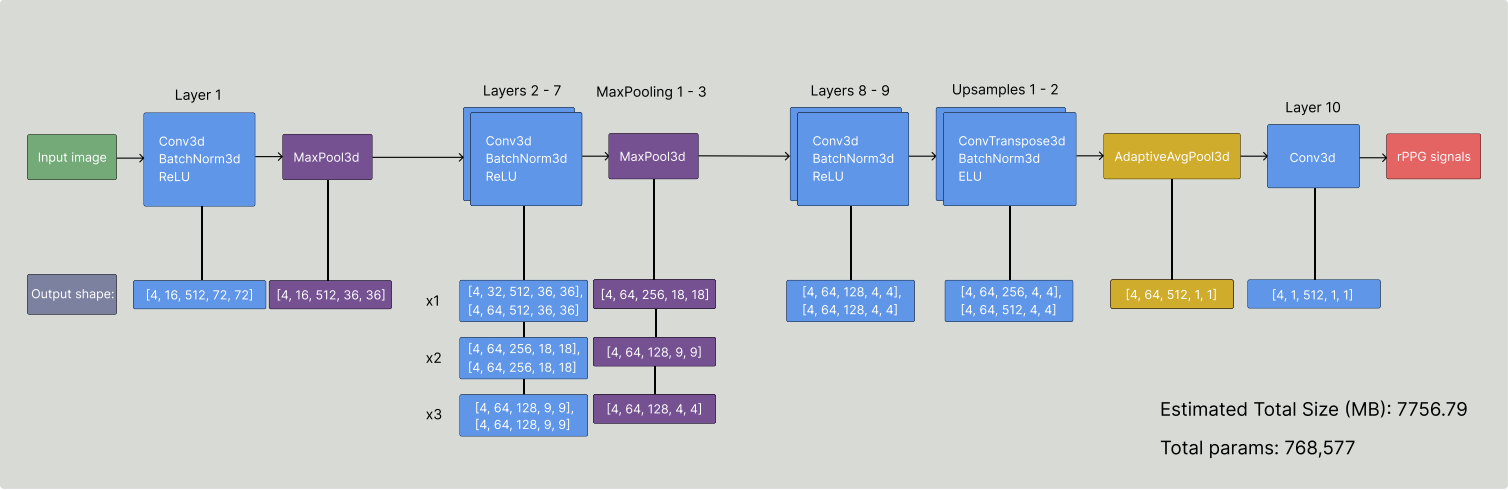

Fig. 4. Summary of the pre-enhanced PhysNet model architecture

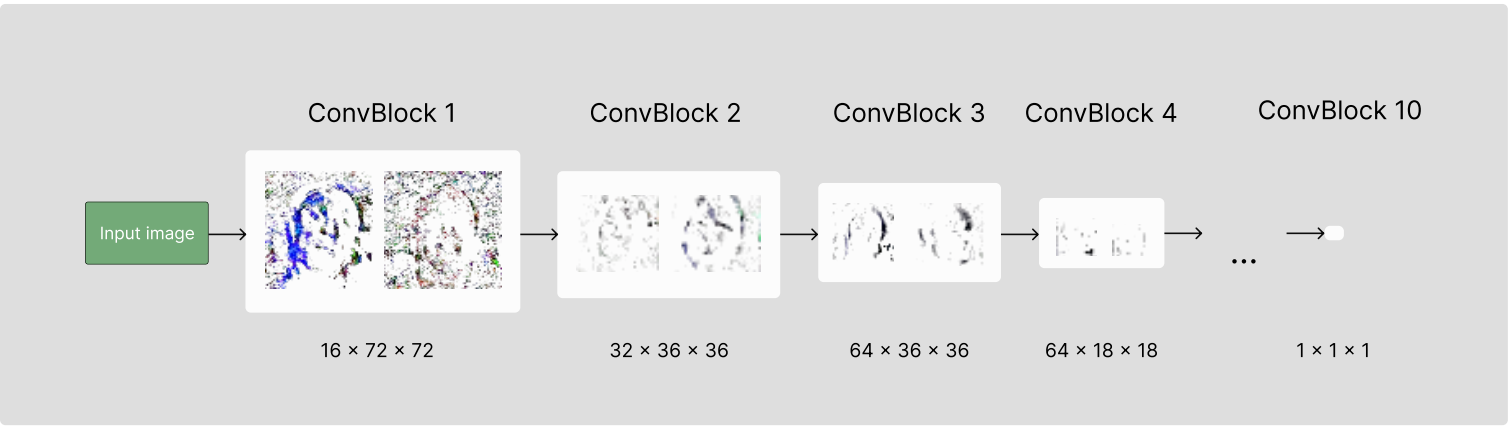

Fig. 5. Feature maps of convolutional blocks of pre-enhanced PhysNet model, with the dimensions of [channels, width, height]

C. RQ3: How do robustness and reliability quality attributes play a role in systematically evaluating and enhancing a deep learning approach?

To understand how the reliability quality attribute plays a role in systematically evaluating and enhancing a deep learning model approach, we performed k-fold cross-validation across five different folds from our dataset. Each fold was divided into training, validation, and testing sets; we thoroughly trained and tested the model and then averaged the performance across all folds. The average results of evaluation metrics of the newly enhanced model across five-folds, compared to pre-enhanced and other deep learning models such as PhysFormer, DeepPhys, and TS-CAN, can be seen in Table III.

Despite not getting consistent improvement results on all four metrics (MAE, RMSE, MAPE, and SNR) on average, the newly enhanced model outperforms the pre-enhanced model in two folds, the first and fifth, on almost all of these metrics, except SNR in fold five. These results are illustrated in Table IV.

In those folds where the enhanced PhysNet model performed worse compared to the pre-enhanced version (folds two, three, and four), we observed that the enhanced model showed some slight signs of overfitting on those folds, which led to a small decrease in performance when presented with the test set. This can be seen in the training and validation graph using fold four as the example as shown in Fig. 6. There, we can see that the validation graph slightly takes an upscale turn compared to the pre-enhanced version of the graph.

As a way of testing the model's robustness, we tested the model on the robustness fold, fold six. We tested folds four and five of the pre-enhanced model compared to folds four and five of the enhanced model. These folds were chosen because they performed worse on average than folds one, two, and three. The results from both models across these two folds were averaged, and the newly enhanced model outperforms the pre-enhanced model as shown in Table V. After our enhancements, the new model has a higher SNR than the pre-enhanced one, as shown in Table III and Table V. This could indicate that it extracts heart rate signals better than the pre-enhanced model on average. Based on robustness and reliability testing, we could evaluate the enhanced model's performance against the pre-enhanced model across different

TABLE II
FACTORS THAT CONTRIBUTE POSITIVELY/NEGATIVELY TO THE EXPLAINABILITY OF THE ML MODEL

Factors	Positive Effect	Negative Effect
Feature Maps	Helps in visualizing the features that are learned by an ML model	Can be ambiguous and complex to comprehend, depending on the ML-based knowledge of the developer
Hyperparameter Tuning	Aids in the understanding of how each hyperparameter contributes to the learning of an ML model	Involves time-consuming experimentation
Architecture Analysis	Helps getting a detailed look into the components of an ML model and how they function	Based on the complexity of the model, it would require time and research to understand what each component does; it does not show how each component contributes to the prediction of the model

folds and observe how well the enhanced model can generalize across different variations of the same dataset. The factors affecting the model's robustness and reliability are summarized in Table VI.

D. RQ4: What recommendations can be provided to software engineers based on the process of enhancing a deep learning model?

After taking into account what we have learned from the experiment that we have conducted to compare our findings to the current research within the field of ML and RE interconnection, we have come up with the main takeaways that software engineers can follow when working with ML model enhancement:

1) Identify the relevant NFRs: Hyatt & Lee posit that focusing only on validation accuracy would not suffice to improve the classification ability of a neural network, as it does not reflect the ability of the model to generalize on unforeseen scenarios [12]. In the case of our experiment, we focused on the quality attributes that are relevant to our case. We have analyzed the literature we collected as part of the RQ1 in Section IV-A and identified explainability, reliability, and robustness as the quality attributes to focus on in our study. Reliability and robustness are also considered important quality attributes in automotive ML systems [46].

As a recommendation for the software engineers who are working with ML models, identifying the relevant NFRs, apart from validation accuracy or performance, is worth considering for achieving an improved overall performance of the model. The improved performance may manifest on different operation levels in the solution space, which are also relevant to stakeholders. This implies that, for example, in the automotive space, it could be that better ML model performance in safety and robustness are considered more crucial than in other quality attributes. As discussed by Habibulah et al., NFRs represent the quality of the addressed system, and to assess the overall quality of an ML system, it is important to facilitate the evaluation of ML systems by applying RE [47].

2) Analyze the tradeoffs: When developing or/and enhancing an ML model, the choice for the value of each hyperparameter may be influenced by many factors, and it becomes a tradeoff between which to settle on and which to use. Determining the optimal values for epoch, batch size, iterations, learning rate, choice of activation function, and the type of

TABLE III
MEAN AVERAGE OF RESULTS OF PRE-ENHANCED AND ENHANCED PHYSNET COMPARED TO OTHER DEEP LEARNING MODELS

Models	MAE	RMSE	MAPE	SNR
PhysNet(Pre-enhanced)	4.2827	9.0170	5.6655	2.0243
PhysNet(Enhanced)	**4.2966**	**9.2360**	**5.7370**	**2.4500**
PhysFormer	6.4194	11.0054	8.3306	-0.3872
DeepPhys	7.4537	12.6894	9.9498	-4.3735
TS-CAN	6.7798	11.7558	9.1654	-4.1959

TABLE IV
ENHANCED AND PRE-ENHANCED PHYSNET MODELS PERFORMANCE COMPARISON PER FOLD

Fold	Models	MAE	RMSE	MAPE	SNR
1	PhysNet(Pre-enhanced)	4.7251	8.7611	6.8127	-0.0911
	PhysNet(Enhanced)	**4.0094**	**8.5693**	**6.1740**	**2.1791**
2	PhysNet(Pre-enhanced)	3.6425	8.4647	5.6647	2.2839
	PhysNet(Enhanced)	**3.7695**	**8.5475**	**5.9132**	**2.7771**
3	PhysNet(Pre-enhanced)	2.7244	5.8102	3.9590	3.1817
	PhysNet(Enhanced)	**3.0690**	**6.3595**	**4.5107**	**2.9616**
4	PhysNet(Pre-enhanced)	5.7843	13.1602	6.1664	1.5952
	PhysNet(Enhanced)	**6.7007**	**14.7606**	**7.1017**	**1.4591**
5	PhysNet(Pre-enhanced)	4.5375	8.8888	5.7248	3.1519
	PhysNet(Enhanced)	**3.9351**	**7.9442**	**4.9859**	**2.8733**

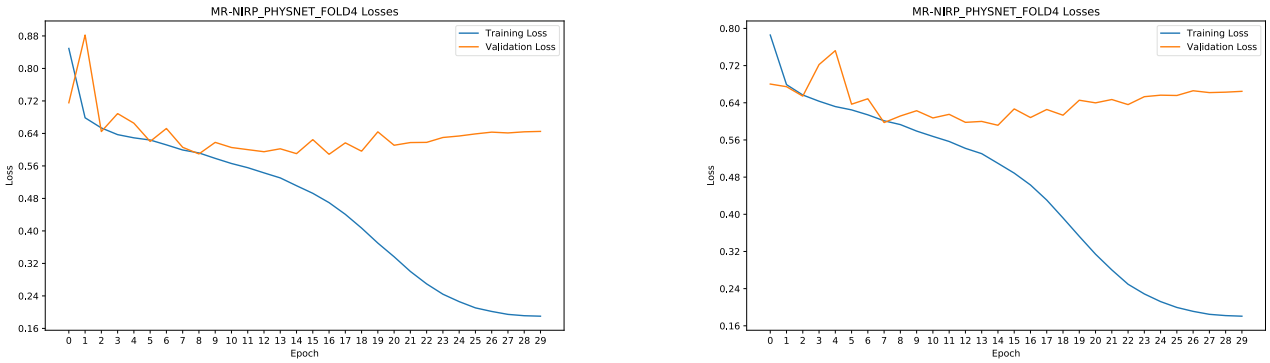

Fig. 6. Comparison of the training and validation loss graphs of pre-enhanced (left) and enhanced (right) models fold four illustrating signs of overfitting

TABLE V
MEAN AVERAGE RESULTS OF RUNNING TESTING ON THE PRE-ENHANCED AND ENHANCED MODEL BOTH TRAINED ON FOLD FOUR AND FOLD FIVE AGAINST FOLD SIX FOR ROBUSTNESS

Models	MAE	RMSE	MAPE	SNR
PhysNet(Enhanced)	**0.5530**	**2.2765**	**0.7830**	**9.8815**
PhysNet(Pre-enhanced)	0.5715	2.3915	0.8070	9.6955

regularization techniques to use may all depend on the size of your dataset, the task at hand, and the complexity of your model. However, experimenting with various hyperparameter configurations to find the optimal settings is crucial and may result in developing a more robust ML model [48], [49].

Finding a unique combination for the hyperparameter configuration usually becomes a process of trial and error until you compromise on the preferred settings. This process can be time-consuming, so it is important to know and manage your time to get the most out of your limited resources. Hyperparameter tuning can lead to an improvement in the model's performance based on quality attributes such as robustness. However, at times, it can increase the complexity of the model, leading to the model's overfitting and the usage of more computational power, resources, and time. In this scenario, one has to understand the tradeoffs between computational efficiency and the improvement you are willing to accept.

As a recommendation to software engineers performing any hyperparameter tuning during the model's enhancement, we suggest careful planning and consideration of the goals and objectives of the process. It is important to have a clear objective for the improvements you are willing to accept and settle on. This can be achieved by understanding the tradeoffs you might face and the choices you have to make and, hence, ensuring proper efficiency by managing your available resources carefully. An example of these tradeoffs would be whether the improvement percentage achieved is worth the computational resources and power the model needs and the time spent to achieve it. Similar tradeoffs can be seen in cases when engineers or companies weigh between cost and accuracy, as discussed by Baughman et al. by pointing out that the stochastic nature of many data science and machine learning techniques leads to tradeoffs between costs and accuracy or perhaps percentage margin of improvement in this case [50].

3) Choose explainability tools relevant to end goal: There are multiple explainability tools available for ML models for interpreting them [29]. However, the relevance of an explainability tool for a particular model should be considered. We have tried to implement the SHAP explainability tool, which is commonly used to represent the features and their contribution to the overall result of the model's performance [29]. However, the tool was incompatible with our model due to the dimensions of the output of our model and the complexity of the model's architecture. Instead, for our model, we analyzed the summary of the model's architecture and visualized the feature maps that are extracted from each convolutional layer as described in Section IV-B.

The results of an executed explainability tool on an ML model are limited to human cognition and understanding of those results [51], [52]. In other words, it is crucial to use an explainability tool that is comprehensible enough for a developer and relevant to explaining the ML model. Our recommendation is to analyze the architecture of the ML model before using any explainability tool to, first and foremost, understand which components the model consists of and which type of architecture the model has, as also suggested by Leslie in the guidelines for ML explainability [51]. Based on that, determine which explainability tool best suits the specific ML model being used [51].

V. DISCUSSION

In this section, we discuss the key findings from our study, how they relate to current research in the field, the strengths and weaknesses of our research, and finally, future work that could be done to extend the research.

TABLE VI
FACTORS THAT CONTRIBUTE POSITIVELY/NEGATIVELY TO ROBUSTNESS AND RELIABILITY OF THE ML MODEL

Factors	**Positive Effect**	**Negative Effect**
K-fold Cross-Validation (Reliability)	• Using the results from all five folds helped provide a more reliable estimate of the model's performance compared to using a single data train and test set split group • Dividing and arranging the dataset in five different folds helped in ensuring efficient data usage; it ensured that we could train, test, and validate all parts of the whole dataset	• Using the k-fold technique on more complex ML models and bigger datasets can be both time-consuming and computationally costly • May negatively affect the generalizability of the model due to potential overfitting on validation sets, thus affecting the reliability of the model when applied to different datasets
Noise Intensive Data Testing (Robustness)	• Helped elaborate and validate that an improvement in the ability to reduce noise in the input data is closely related to the robustness of the model. When the SNR value of the model improved, its robustness to high-noise data improved	• It may not present some of the corner cases that may arise during real-life driving scenarios and, therefore, cannot be relied upon as a complete method for testing the robustness of an ML model

A. *Analysis of the findings*

1) RQ1: According to the current literature, the key challenges identified repeatedly include varying illumination and head motion, different skin tones, and vibrations caused by driving. Our study focused on the challenges that are specific to monitoring heart rate in a vehicle environment. Based on these challenges, explainability, robustness, and reliability NFRs were appropriate to choose in the context of enhancing a black-box deep learning model since these NFRs help us to test the model beyond the performance in accuracy to reveal any inconsistencies in the other qualities of it. When identifying challenges that can affect the predictions of an ML model, it is crucial to see how NFRs can be applied to help resolve or mitigate the challenges. Understanding the challenges and qualities of the ML system being enhanced or developed could show a broader perspective on the system and indicate which qualities need improvement based on the identified challenges.

Apart from the challenges that should be addressed by choosing the relevant quality attributes specific to our study, other qualities of an ML-based system may be relevant to the system's operation. Improving an ML-based system based on qualities such as, for example, interoperability and fairness may address different challenges connected to the integration of the system to software systems and trust in the system itself, as is also discussed in [20]. Which may not directly affect the prediction of an ML-based system, such as heart rate detection as in our study, though it would affect the understanding of compatibility of the system with other software components and bias in the system to certain features.

2) RQ2: The findings from RQ2 emphasize the importance of understanding how an ML model works before performing any action to further enhance or develop the model. Due to the black-box characteristic of deep learning models [11], to improve them, it becomes essential for developers to understand them better [15], as their process of learning and predicting data is complex and inconspicuous.

An increased explainability of an ML model can contribute to a better understanding of its architecture and how each function contributes to the model learning process. This becomes even more relevant when working with a complex CNN model, with numerous operations on input tensors hidden from the developer's comprehension surface. The more convolutional layers and pooling operations the model has, the more complex it can get in terms of the extent to which it can be interpreted. Gaining a better understanding of the inner workings of a black-box ML model has helped us enhance the model and gave us an idea of what areas of the model's architecture we could look into to experiment with enhancing the model.

The experimental part of the study was based on visualizing feature maps of each convolutional block and the architecture of the PhysNet model. However, understanding and using the explainability tool or method that seems appropriate for the purpose depends on the type and complexity of the model, as well as the developer's cognitive abilities.

After performing hyperparameter tuning, we discovered that using the Swish activation function together with ReLU allows us to leverage the unique properties of each activation function to capture different aspects of the data and introduce diversity in the non-linear transformations of the data. This could help the model learn more complex patterns in the data at each convolutional block.

3) RQ3: According to the results from RQ3, the enhanced model performed better than the pre-enhanced model in some of the evaluation metrics, such as SNR. This implies that the enhanced PhysNet got rid of the noise better when extracting facial features from the dataset images than the pre-enhanced PhysNet. However, The enhanced model decreases in perfor-

mance in other metrics such as MAE, RMSE, and MAPE based on the average results of Table III. This shows that despite its ability to reduce more noise, as indicated by higher SNR results, the model still has higher loss in some cases. These differences are negligible and in the lower percentage. These cases might be unique and hard to pinpoint unless there is more data to test since the enhanced model outperforms the pre-enhanced model when tested on the robustness fold six. The results across all five folds of the pre-enhanced and the enhanced model show that the enhanced model outperforms the pre-enhanced Physnet in folds one and five. This shows that the model can learn better about some of the dataset arrangements in the fold.

Using k-fold cross-validation to reinforce the reliability of the model's performance as per the reliability quality attribute we chose in RQ3, we see that, on average, the enhanced model does not outperform the pre-enhanced model on all the metrics. Therefore, we failed to reject the null hypothesis stated in Section III-A. Using k-fold cross-validation provided a more reliable estimation of the model's performance by ensuring that the model's performance is consistent across different parts of the dataset. It also reinforced the model's reliability by providing a more robust performance and generalization capability evaluation.

According to our robustness test on both models using fold six, the newly enhanced PhysNet outperforms the initial pre-enhanced PhysNet. These results mean that despite the slight decline in performance on metrics such as MAE, RMSE, and MAPE on average, the enhanced PhysNet predicts better on large-motion datasets. This could be due to the small improvement in its ability to remove noise, as shown by the SNR scores, and capture more accurate rPPG signals. The combination of using the Swish and ReLU activation functions became the factor of the model improvement on folds one, five, and six. This could be attributed to the sigmoid function present in the Swish activation function, which calculates the output from neurons in the network more evenly than the ReLU function due to the inclusion of negative output and potentially better convergence when training of the model takes place [45].

These findings show that testing on a model based on defined quality attributes is important. They also show that the more robust the model is, the more its SNR performance increases. This assumption can be seen as true since, in Table IV, it can be observed that the lower the SNR score of a model, the more unnecessary noise it picks up from the facial features. This implies a less accurate prediction of the heart rate.

B. Connection to related work & strengths and weaknesses

As argued by Hyatt & Lee and Trenquier et al., evaluating an ML model using relevant NFRs beyond accuracy and performing a quality evaluation of the model based on quality attributes may improve it by making it more consistent and reliable [12], [21]. Horkoff discusses that it is important to evaluate the performance of an ML system from different quality attributes other than accuracy [20]. By prioritizing and using reliability and robustness NFRs, using k-fold cross-validation and fold six for testing and evaluation, we could evaluate the model from a different perspective rather than focusing on its accuracy. Without k-fold cross-validation, we would have had one way of comparing the two different model versions. For example, if we had used fold five only, it would have appeared that the enhanced version outperformed the pre-enhanced PhysNet entirely. However, after averaging the results across five different folds, we could achieve consistent results on two of the folds, thus indicating that our model still needs improvement.

Some weaknesses of our study include the diversity of the data we experimented with. Fold six that we have created consists of five subjects, with only large head movements. However, this may not be sufficient for testing the robustness of the model since robustness would entail how well the model predicts heart rate during corner cases, such as blurred frames due to potential fogging of the camera. It is also worth mentioning that light could come from different angles, and depending on the intensity of light or its color, the model can have varying performance. However, the strengths of this study include using NFRs with practical application in evaluating a deep learning model. We have used relevant NFRs to evaluate our model and achieved a broader evaluation by focusing on more NFRs than by using accuracy as an indicator of the model's improved or worsened performance in detecting heart rate. The study has also shown that performing tasks such as k-fold cross-validation is significant in obtaining credible, consistent, and reliable model performance.

C. Validity threats

Internal validity: A possible threat to internal validity is that to unravel the black-box aspect of the ML model and understand its inner workings and architecture, we used only a few methods for explainability. The function model.summary() and feature maps were the only techniques applied that might limit the extent to which explainability was covered. In addition, an improvement of explainability of the model may not directly be connected to using the model.summary() function along with the feature maps. External factors, such as researching what each layer does by looking at the descriptions of the functions in PyTorch, could have also contributed to our understanding of the model. Although first analyzing the architecture of the model and how each convolutional block represents the features served as our starting point for enhancing the explainability of the model overall.

Construct validity: When working with the model, we used k-fold cross-validation to test for reliability and part of the dataset with a large head motion for testing for robustness. However, that may not be enough to determine if the reliability and robustness of the model increased based on these factors alone, which could be a potential threat to the construct validity. Therefore, we cannot state with absolute certainty that the overall robustness of the model increased based on fold six alone and SNR values or that the model became

less reliable overall based on five folds compared to its pre-enhanced version. To ensure that the right extent of robustness and reliability is satisfied, the relevant stakeholders of an ML system have to be involved in determining to what extent an ML-based system has to be tested to reach a satisfactory level of robustness and reliability.

External validity: A potential external validity threat to our study is our NFR elicitation method. Some common ways of eliciting NFRs in RE are by conducting stakeholder interviews or surveys [53]. However, our method of eliciting encompassed researching and analyzing domain knowledge of the topic of our study. It can be argued that since there is limited research on the requirement elicitation for ML-based systems, we have used an alternative method by using a literature review of the available research to elicit the most common NFRs in working with ML, such as explainability, robustness, and reliability, with the second and third being common in the automotive field and safety-critical ML systems. Moreover, the methods used to better understand the model may not be appropriate and applicable for working with other ML-based systems, since we have worked with a CNN based model. In the case of our study, we increased the explainability of the model by using the model.summary() function and feature maps to see how the frames are represented within each layer. These methods helped us understand the model better. However, our understanding can be subject to our past knowledge before working with the model and our cognitive ability. In our study, the complexity of the code infrastructure and scope contributed to our decision for choosing these methods for explainability.

D. Future work

Firstly, the effectiveness of using NFRs on ML models should be explored. Some NFRs may be more suitable for certain models than others; however, assessing the impact that each quality attribute brings to the model is crucial to understanding which quality attributes contribute the most to the development and enhancement of the model.

Secondly, when it comes to hyperparameter tuning, it is worth exploring how pooling layers can be altered to improve the quality of extracting the features from images. For example, a customized pooling layer called T-Max-Avg shows promising results in improving the prediction accuracy of CNN models, which could also be implemented in the PhysNet model and potentially improve its heart rate monitoring accuracy [54]. Additionally, hyperparameter tuning algorithms such as Grid Search, Bayesian Optimization, or Random Search could be used to automate hyperparameter tuning to potentially save time when enhancing the model [55].

Lastly, the relevance of the identified quality attributes in RQ1 in Section IV-A could be assessed further by getting an opinion from software engineers working in the automotive industry in the ML field to validate our findings. Moreover, robustness testing could be performed on more varied data, including augmented data with varying skin tones, simulation of vibration from the car, slightly blurred camera lens, and other available vehicle environment datasets, to increase the generalization of the training data fed to the model.

VI. Conclusion

In conclusion, it is crucial to identify challenges in the problem space and apply requirement engineering to elicit vital quality attributes from them. Using these quality attributes as a guide in enhancing black-box ML models may change the way these ML models are tested and evaluated. In our study, we have identified explainability, reliability, and robustness as relevant NFRs and as points of focus to base the enhancement and evaluation of the PhysNet CNN model. We have seen an average improvement in the SNR evaluation metric compared to the pre-enhanced PhysNet model and better performance in predicting heart rate based on the large head movements dataset. However, the pre-enhanced PhysNet model outperforms the enhanced version by predicting the heart rate more accurately overall.

It is important to understand the trade-offs when enhancing an ML model. Specifically, when enhancing a model based on hyperparameter tuning, more complexity can be introduced into its architecture, thus compromising its explainability. Different methods can be used to interpret the model's explainability. However, it depends on what can be understood from using these methods, and therefore, it is subject to the purpose of model enhancement.

VII. Acknowledgement

We would like to extend our gratitude to our supervisor, Tayssir Bouraffa, for guiding and supporting us throughout this thesis work.

References

[1] E. Petridou and M. Moustaki, "Human factors in the causation of road traffic crashes," *Eur J Epidemiol*, vol. 16, pp. 819–826, 2000.

[2] M. Skyving, J. Möller, and L. Laflamme, "What triggers road traffic fatalities among older adult drivers? An investigation based on the Swedish register for in-depth studies of fatal crashes," *Accident Analysis & Prevention*, vol. 190, p. 107149, Sep. 2023.

[3] Department of Statistics, "Road Traffic Injuries," Traffic Analysis, http://www.trafa.se, 2022. [Online]. Available: https://www.trafa.se/en/road-traffic/road-traffic-injuries/ [Accessed: Feb. 28, 2024].

[4] M. Staubach, "Factors correlated with traffic accidents as a basis for evaluating Advanced Driver Assistance Systems," *Accident Analysis & Prevention*, vol. 41, no. 5, pp. 1025-1033, 2009.

[5] Z. Gong, X. Yang, R. Song, X. Han, C. Ren, H. Shi, J. Niu and W. Li, "Heart Rate Estimation in Driver Monitoring System Using Quality-Guided Spectrum Peak Screening," in *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1-14, 2024.

[6] G. Kaiwen, T. Zhai, M. H. Purushothama, A. Dobre, S. Meah, E. Pashollari, A. Vaish, C. DeWilde and M. N. Islam, "Contactless vital sign monitoring system for in-vehicle driver monitoring using a near-infrared time-of-flight camera," *Applied Sciences*, vol. 12, no. 9, p. 4416, 2022.

[7] A. Ni, A. Azarang and N. Kehtarnavaz. "A Review of Deep Learning-Based Contactless Heart Rate Measurement Methods." *Sensors*, Basel, Switzerland, vol. 21,11 3719. 27 May. 2021, doi:10.3390/s21113719

[8] W. Chen and D. McDuff, "DeepPhys: Video-Based Physiological Measurement Using Convolutional Attention Networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 349-365, 2018.

[9] P. V. Rouast, M. T. P. Adam, R. Chiong, D. Cornforth and E. Lux, "Remote heart rate measurement using low-cost RGB face video: a technical literature review," *Frontiers of Computer Science*, vol. 12, pp. 858–872, 2018.

[10] S. Amershi , A. Begel, C. Bird, R. Deline, H. Gall, E. Kamar, N. Nagappan, B. Nushi and T. Zimmermann, "Software Engineering for Machine Learning: A Case Study," in *IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, Montreal, QC, Canada, pp. 291-300, 2019, doi: 10.1109/ICSE-SEIP.2019.00042.

[11] M. Borg ,C. Englund , K. Wnuk , B. Duran , C. Levandowski ,S. Gao, Y. Tan , H. Kaijser , H. Lonn and J. Tornqvist, "Safely Entering the Deep: A Review of Verification and Validation for Machine Learning and a Challenge Elicitation in the Automotive Industry ", *Journal of Automotive Software Engineering*, Volume 1, Issue 1, Pages 1 - 19, Dec. 2019.

[12] J. S. Hyatt and M. S. Lee, "Requirements for Developing Robust Neural Networks", at *AAAI FSS-19: Artificial Intelligence in Government and Public Sector*, Arlington, Virginia, USA, Oct. 2019.

[13] H. Villamizar, T. Escovedo and M. Kalinowski, "Requirements Engineering for Machine Learning: A Systematic Mapping Study," *47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, Palermo, Italy, pp. 29-36, 2021, doi: 10.1109/SEAA53835.2021.00013.

[14] M. T. Ribeiro, S. Singh and C. Guestrin, "Why Should I Trust You?": Explaining the Predictions of Any Classifier ", in *KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug 2016 , pp 1135–1144. https://doi.org/10.1145/2939672.2939778

[15] T. Li and L. Han, "Dealing with Explainability Requirements for Machine Learning Systems," in *IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, Torino, Italy, pp. 1203-1208, 2023, doi: 10.1109/COMPSAC57700.2023.00182. –¿ old b13

[16] H. B. Braiek and F. Khomh, "Machine Learning Robustness: A Primer", *arXiv preprint*, 2024, doi: 10.48550/arXiv:2404.00897.

[17] E. M. Nowara, T. K. Marks, H. Mansour and A. Veeraraghavan, "Near-Infrared Imaging Photoplethysmography During Driving," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 4, pp. 3589-3600, Apr. 2022.

[18] Z. Pei, L. Liu, C. Wang and J. Wang, "Requirements Engineering for Machine Learning: A Review and Reflection," in *IEEE 30th International Requirements Engineering Conference Workshops (REW)* , Melbourne, Australia, pp. 166-175, 2022, doi: 10.1109/REW56159.2022.00039.

[19] K. M. Habibullah and J. Horkoff, "Non-functional Requirements for Machine Learning: Understanding Current Use and Challenges in Industry," in *IEEE 29th International Requirements Engineering Conference (RE)*, Notre Dame, IN, USA, pp. 13-23, 2021, doi: 10.1109/RE51729.2021.00009.

[20] J. Horkoff, "Non-Functional Requirements for Machine Learning: Challenges and New Directions," in *IEEE 27th International Requirements Engineering Conference (RE)*, Jeju, Korea (South), pp. 386-391, 2019, doi: 10.1109/RE.2019.00050.

[21] H. Trenquier, F. Ishikawa and S. Tokumoto, "Attribute-based Granular Evaluation for Performance of Machine Learning Models," in *IEEE International Conference On Artificial Intelligence Testing (AITest)*, Oxford, UK, pp. 125-132, 2020, doi: 10.1109/AITEST49225.2020.00026.

[22] E. M. Nowara, T. K. Marks, H. Mansour and A. Veeraraghavan, "SparsePPG: Towards Driver Monitoring Using Camera-Based Vital Signs Estimation in Near-Infrared," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Salt Lake City, UT, USA, pp. 1353-135309, 2018.

[23] Z. Yang, H. Wang and F. Lu, "Assessment of Deep Learning-Based Heart Rate Estimation Using Remote Photoplethysmography Under Different Illuminations," in *IEEE Transactions on Human-Machine Systems*, vol. 52, no. 6, pp. 1236-1246, Dec. 2022.

[24] Z. Wang, X. Yang, H. Lu, C. Shan and W. Wang, "Benchmark of Physiological Model Based and Deep Learning Based Remote Photoplethysmography in Automotive Applications," *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Rhodes Island, Greece, pp. 1-5, 2023.

[25] L.-W. Chiu, Y.-R. Chou, Y.-C. Wu and B.-F. Wu, "Deep-Learning-Based Remote Photoplethysmography Measurement in Driving Scenarios With Color and Near-Infrared Images," in *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1-12, 2023.

[26] R. J. Lee, S. Sivakumar and K. H. Lim, "Review on remote heart rate measurements using photoplethysmography," *Multimed Tools Appl,* 2023, doi: 10.1007/s11042-023-16794-9

[27] K. J. Stol and B. Fitzgerald, "The ABC of Software Engineering Research," *ACM Transactions on Software Engineering and Methodology*, vol. 27, Issue 3, Article No.: 11pp 1–51, Sept. 2018.

[28] T. Freiesleben and T. Grote, "Beyond Generalization: a Theory of Robustness in Machine Learning," in *Synthese*, Vol 202,no. 109, Sept. 2023. https://doi.org/10.1007/s11229-023-04334-9

[29] L. V. Haar, T. Elvira and O. Ochoa, "An analysis of explainability methods for convolutional neural networks", in *Engineering Applications of Artificial Intelligence*, Volume 117, Part A, 105606, Jan. 2023.

[30] T. -T. Wong and P. -Y. Yeh, "Reliable Accuracy Estimates from k-Fold Cross Validation," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 8, pp. 1586-1594, 1 Aug. 2020, doi: 10.1109/TKDE.2019.2912815.

[31] Z. Yu, Y. Shen, J. Shi, H. Zhao, P. H. S. Torr, G. Zhao, "PhysFormer: Facial Video-Based Physiological Measurement With Temporal Difference Transformer", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4186-4196, 2022.

[32] M. V. Shcherbakov, A. Brebels, N. L. Shcherbakova, A. P. Tyukov, T. A. Janovsky, and V. A. E. Kamaev, "A survey of forecast error measures," *World Applied Sciences Journal*, vol. 24, no. 24, pp. 171-176, 2013.

[33] Y.-C. Lin and Y.-H. Lin, "A study of color illumination effect on the SNR of rPPG signals," *39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Jeju, Korea (South), pp. 4301-4304, 2017, doi: 10.1109/EMBC.2017.8037807.

[34] S. Chen, S. K. Ho, J. W. Chin, K. H. Luo, T. T. Chan, R. H.Y. So and K. L. Wong, "Deep learning-based image enhancement for robust remote photoplethysmography in various illumination scenarios," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Vancouver, BC, Canada, pp. 6077-6085, 2023. doi: 10.1109/CVPRW59228.2023.00647.

[35] T. Yep, Torchinfo; on *GitHub.* [Online]. Available: https://github.com/TylerYep/torchinfo [Accessed: May. 14, 2024].

[36] H. H. Aghdam and E. J. Heravi, "Convolutional Neural Networks, " in: *Guide to Convolutional Neural Networks. Springer, Cham.*, pp 85-130, May. 2017.

[37] S. Ioffe and C. Szegedy, " Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Proceedings of the 32nd International Conference on Machine Learning*, PMLR 37:448-456, vol 37, 2015.

[38] A. F. Agarap, " Deep Learning using Rectified Linear Units (ReLU)," *Cornell University; Computer Science, Neural and Evolutionary Computing*, 2018. https://doi.org/10.48550/arXiv.1803.08375.

[39] PyTorch Documentation, "ConvTransposed3d," [Online]. Available: https://pytorch.org/docs/stable/generated/torch.nn.ConvTranspose3d.html [Accessed: May. 14, 2024].

[40] PyTorch Documentation, "AdaptiveAvgPool3d [Online]. Available: https://pytorch.org/docs/stable/generated/torch.nn.AdaptiveAvgPool3d.html [Accessed: May. 14, 2024].

[41] B. Athiwaratkun and K. Kang, "Feature Representation in Convolutional Neural Networks," in *Cornell University; Computer Science, Computer Vision and Pattern Recognition*, 2015. https://doi.org/10.48550/arXiv.1507.02313.

[42] S. Chakraborty, S. Paul, R. Sarkar and M. Nasipuri, "Feature Map Reduction in CNN for Handwritten Digit Recognition," in *Machine Learning and Data Analytics. Advances in Intelligent Systems and Computing*, vol 740. Springer, Singapore, 2018.

[43] N. Gowdra, R. Sinha and S. MacDonell, "Examining convolutional feature extraction using Maximum Entropy (ME) and Signal-to-Noise Ratio (SNR) for image classification," *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, Singaporeb, pp. 471-476, 2020. doi: 10.1109/IECON43393.2020.9254346.

[44] W.-Y. Lee, S.-M. Park and K.-B. Sim, "Optimal hyperparameter tuning of convolutional neural networks based on the parameter-setting-free harmony search algorithm," in *Optik - International Journal for Light and Electron Optics*, 2018. https://doi.org/10.1016/j.ijleo.2018.07.044

[45] P. Ramachandran, B. Zoph and Q. V. Le, "Swish: A Self-gated Activation Function," *Technical report*, 2017.

[46] K. M. Habibullah, H. -M. Heyn, G. Gay, J. Horkoff, E. Knauss, M. Borg, A. Knauss, H. Sivencrona and P. J. Li, "Requirements

and software engineering for automotive perception systems: an interview study," *Requirements Engineering*, Vol 29, pp 25–48, 2024. https://doi.org/10.1007/s00766-023-00410-1

[47] K. M. Habibullah, G. Gay and J. Horkoff, "Non-Functional Requirements for Machine Learning: An Exploration of System Scope and Interest," in *IEEE/ACM 1st International Workshop on Software Engineering for Responsible Artificial Intelligence (SE4RAI)*, Pittsburgh, PA, USA, 2022, pp. 29-36, doi: 10.1145/3526073.3527589.

[48] K. E. Hoque and H. Aljamaan, "Impact of Hyperparameter Tuning on Machine Learning Models in Stock Price Forecasting," in *IEEE Access*, vol. 9, pp. 163815-163830, 2021, doi: 10.1109/ACCESS.2021.3134138.

[49] P. Schratz, J. Muenchow, E. Iturritxa, J. Richter and A. Brenning, "Hyperparameter tuning and performance assessment of statistical and machine-learning algorithms using spatial data," *Ecological Modelling*, Vol 406, pp- 109-120, 2019. https://doi.org/10.1016/j.ecolmodel.2019.06.002.

[50] M. Baughman, N. Chakubaji, H. -L. Truong, K. Kreics, K. Chard and I. Foster, "Measuring, Quantifying, and Predicting the Cost-Accuracy Tradeoff," in *IEEE International Conference on Big Data (Big Data)*, Los Angeles, CA, USA, 2019, pp. 3616-3622, doi: 10.1109/BigData47090.2019.9006370.

[51] D. Leslie, "Understanding artificial intelligence ethics and safety, A guide for the responsible design and implementation of AI systems in the public sector", *The Alan Turing Institute*, 2019.

[52] A. B. Arrieta, N. D-. Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila and F. Herrera, "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, Vol 58, Pages 82-115, 2020. https://doi.org/10.1016/j.inffus.2019.12.012

[53] "Guide to the Software Engineering Body of Knowledge (SWEBOK)," *IEEE Computer Society*, Version 3.0, 2014. [Online]. Available: https://www.computer.org/education/bodies-of-knowledge/software-engineering/v3. [Accessed: Jun. 16, 2024].

[54] L. Zhao and Z. Zhang, "A improved pooling method for convolutional neural networks" in *Scientific Reports*, Vol 14, 2024. https://doi.org/10.1038/s41598-024-51258-6

[55] H. Alibrahim and S. A. Ludwig, "Hyperparameter Optimization: Comparing Genetic Algorithm against Grid Search and Bayesian Optimization," *2021 IEEE Congress on Evolutionary Computation (CEC)*, Kraków, Poland, 2021, pp. 1551-1559, doi: 10.1109/CEC45853.2021.9504761.