CardioLive: Empowering Video Streaming with Online Cardiac Monitoring

Sheng Lyu
shenglyu@connect.hku.hk
The University of Hong Kong
Hong Kong S.A.R

Ruiming Huang
huangruiming@connect.hku.hk
The University of Hong Kong
Hong Kong S.A.R

Sijie Ji
sijieji@caltech.edu
The University of Hong Kong
Hong Kong S.A.R

Yasar Abbas Ur Rehman
yasir.abbas42@gmail.com
TCL AI Lab
Hong Kong S.A.R

Ma Lan
rubyma@tcl.com
TCL AI Lab
Hong Kong S.A.R

Chenshu Wu
chenshu@cs.hku.hk
The University of Hong Kong
Hong Kong S.A.R

ABSTRACT

Online Cardiac Monitoring (OCM) emerges as a compelling enhancement for the next-generation video streaming platforms. It enables various applications including remote health, online affective computing, and deepfake detection. Yet the physiological information encapsulated in the video streams has been long neglected. In this paper, we present the design and implementation of *CardioLive*, the first online cardiac monitoring system in video streaming platforms. We leverage the naturally co-existed video and audio streams and devise `CardioNet`, the first audio-visual network to learn the cardiac series. It incorporates multiple unique designs to extract temporal and spectral features, ensuring robust performance under realistic video streaming conditions. To enable the Service-On-Demand online cardiac monitoring, we implement *CardioLive* as a plug-and-play middleware service and develop systematic solutions to practical issues including changing FPS and unsynchronized streams. Extensive experiments have been done to demonstrate the effectiveness of our system. We achieve a Mean Square Error (MAE) of 1.79 BPM error, outperforming the video-only and audio-only solutions by 69.2% and 81.2%, respectively. Our *CardioLive* service achieves average throughputs of 115.97 and 98.16 FPS when implemented in Zoom and YouTube. We believe our work opens up new applications for video stream systems. Our code is available at https://anonymous.4open.science/r/CardioNet-4B1E/.

1 INTRODUCTION

Video streaming has exploded in recent years, and its growth shows no signs of slowing down. From social platforms like TikTok that have turned live video sharing into a global phenomenon, to Zoom, which has become synonymous with remote work and learning, video streaming has woven itself into the fabric of our daily lives. The popularity of these platforms has not waned even after the COVID-19 pandemic.

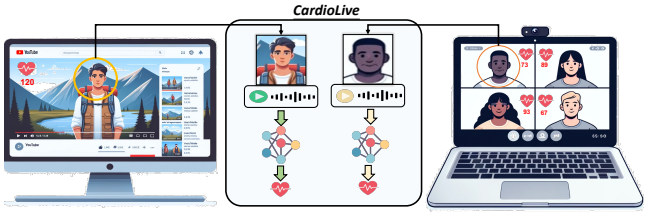


Figure 1: Online Cardiac Monitoring (OCM).

The market is booming steadily [12], reflecting our collective appetite for real-time, interactive, and accessible content.

Online Cardiac Monitoring (OCM) can be one intriguing enhancement for the next-generation video streaming platforms. The rich tapestry of video and audio in streaming not only provides the context of actions, movement, human activities, speech, etc., but it also embeds subtle cardiac events, which have been long neglected in contemporary video streaming systems. Uncovering such physiological information would bring various benefits. In the realm of remote health, physicians could remotely access real-time cardiac data without the need for specialized equipment [3]. Similarly, in video gaming, displaying a player's heart rate during live streams could add a new layer of excitement and engagement for viewers [8]. Notably, in the Paris Olympics 2024, NBC introduced heart-rate streaming to add a new "gamifying" element for creating compelling TV [6]. This technology could also play a pivotal role in online conferences or interviews, where emotional responses (including lies) inferred from cardiac data [53, 57] could enrich interactions, making them more nuanced and meaningful. Furthermore, the potential for this technology extends into security and fraud detection against digital impersonation techniques like deepfakes [37, 46]. These multifaceted applications of

OCM underscore its potential to revolutionize video streaming, making it not just a tool for communication and entertainment, but also a platform for health monitoring, affective computing, emotional intelligence, and security.

However, existing online cardiac monitoring either relies on specified hardware [7](*e.g.*, heartbeat belt) or introduces additional sensing modalities (*e.g.*, Wi-Fi [38], mmWave [70], and UWB [21] *etc.*) which are not typically available in live streaming systems. These approaches suffer from extra cost and are often misaligned with live streams. Moreover, sensing-based approaches necessitate active transmission of the sensing signals [47, 58], which is often impractical to force in live streaming applications. A video streaming system that seamlessly enables online cardiac monitoring in pervasive contexts without additional hardware still lacks.

In this paper, we ask: *Can we incorporate accurate and robust online cardiac monitoring into a video-streaming system without introducing additional hardware or modalities?* To build such a system, we answer the following key questions:

First, *what information should we take from the video streaming system to monitor the cardiac activities?* Existing works [20, 33, 39, 40, 44, 72–74, 77] on extracting heart rate from human faces focus on remote photoplethysmography (rPPG) which leverages solely video. These video-only solutions are more likely to suffer from low illumination conditions, head movement, and orientation. Recent progress in cardiac Vocal User Interfaces (VUIs) [67] inspires us to infer heart rate from human speech. However, audio signals are usually sensitive to noise interference and lack contextual background information, rendering them less robust in real-life scenarios and requiring user calibration. Conceptually, video provides detailed visual context while sound exhibits resilience to varying light conditions and body motions. Consequently, they offer complementary advantages to enhance cardiac monitoring. This motivates us to move beyond video-only or audio-only solutions, and investigate new designs to combine the naturally co-existed video and audio streams.

Second, *how to tackle real-world problems to make this system robust and accurate?* Unveiling the cardiac activity from video and audio is challenging. The information is nuanced and easy to be overshadowed by more prominent body movements, environmental dynamics, and/or ambient noise. Previous works [33, 39, 40, 44, 74, 77] primarily evaluate models on well-controlled datasets featuring static subjects under optimized light conditions and viewing angles, which simplifies the problems yet becomes unrealistic in real-world settings. The task gets even more challenging when deployed in live video streaming environments, due to the discrepancies in frame rates, degraded image quality, and presence of multiple individuals with mixed audio and video streams. To deliver an accurate and robust system in practice, novel techniques are desired to effectively discern subtle cardiac signals amidst various disturbances while combating fluctuating frame rates and drifted misalignment of the streams.

Third, *how to enable Service-On-Demand (SoD) cardiac monitoring in video streaming systems?* Despite the promise of the integration, enabling SoD for users poses significant challenges due to the complexity of modern video streaming systems. These platforms vary widely, encompassing formats such as conferences [5, 9, 11], Video-On-Demand (VoD) [4, 10], live streaming [1, 13], *etc*, each with its own technical and operational nuances. These providers must balance the demands of real-time data processing with the need for immediate accessibility and minimal latency while not interfering with the original streams. At the same time, deploying our service on edge (*e.g.*, browsers) benefits from preserving privacy, while getting access to the data yields another challenge. One naive way is to deploy our models over the WebRTC peers, but it lacks scalability and versatility. To this end, we are motivated to establish a plug-and-play service that can be seamlessly integrated into video streaming systems, whether hosted on servers or edges.

In this paper, we present *CardioLive*, the first-of-its-kind online cardiac monitoring system, that can continuously infer the heart rate in video streaming systems. At the core of *CardioLive*, we design a novel audio-video deep learning network, `CardioNet`, that can effectively learn the nuanced cardiac activities from facial regions and human voices. Specifically, we combine the temporal difference network and a frequency-aware block to model the temporal-spectrum properties from videos. We directly exploit the raw audio to capture the cardiac activities by emulating the natural filtering effects of the human body. To handle the irregularly sampled data, we integrate time embeddings to provide temporal context. Finally, we fuse audio and visual features through a multi-head temporal attention mechanism, which synergistically combines the strengths of both modalities to produce a robust and precise cardiac monitoring solution.

We further devise systematic solutions to deploy *CardioLive* as a middleware service to support the SoD online cardiac monitoring. We introduce practical techniques to handle issues like changing FPS and unsynchronized streams. Through in-depth analyses of mainstream video streaming architectures, we realize a *CardioLive* service with effective data hooks and novel packet and buffer designs, which can be easily integrated with various video streaming systems.

Extensive experiments have been done to validate the effectiveness of *CardioLive*. We have self-collected data through 8 different devices and 10 users. Our evaluation results show that *CardioLive* achieves a mean absolute error (MAE) of 1.79 BPM and root mean square error (RMSE) of 3.25 BPM, largely outperforming the video-only solutions by 69.2% in MAE and 61.4% in RMSE, and the audio-only solution by 81.2% in MAE and 76.8% in RMSE. We demonstrate *CardioLive*'s

generalizability to different environments, devices, and users. As for *CardioLive* service, we implement our system on two ends, a meeting platform (Zoom) and a content provider (YouTube), respectively. We achieve the overall throughput of 115.97 FPS and 98.16 FPS for each platform respectively, ensuring smooth updates without disrupting the original streams. These results highlight the robustness and accuracy of *CardioLive*, confirming its potential for widespread application in video streaming systems.

Contributions: We conclude our contributions as follows: ❶ To the best of our knowledge, we are the first to combine video and audio for cardiac monitoring in video streaming systems. Our solution outperforms video-only or audio-only approaches, especially under adverse conditions in practice. ❷ We develop `CardioNet`, a novel audio-video pipeline that can uncover the nuanced heart rate. Our experiments validate the robustness against different conditions. ❸ We implement *CardioLive* as a service-based plug-and-play middleware, that can seamlessly be integrated into mainstream platforms for real-time streaming.

2 DESIGN SCOPE

Application Momentum: Consider a scenario where users on platforms such as Zoom or YouTube can access real-time cardiac monitoring. With just a single click, users see their heart rate, providing immediate insights into their emotional and physiological states, including what others are thinking about, whether they are in good health, and how exciting the game is. By online cardiac monitoring, these platforms could significantly enhance user engagement and interactivity. Particularly, *CardioLive* can provide unique and compelling benefits in the downstream applications:

❶ **Accessibility:** In many video streaming scenarios, such as live product demonstrations on TikTok or Zoom interviews, using wearables or additional hardware is often impractical. OCM can overcome this problem by leveraging modalities that already exist within video streams, thereby increasing accessibility for audiences and facilitating broader engagement. It also promises wider dissemination of remote health, offering device-free cardiac monitoring compared to the latest work [19] that relies on earphones.

❷ **Enhanced Analytical Abilities:** While there exist alternative approaches for tasks including affective computing [15, 43, 45, 66] and deepfake detection [23, 69], the cardiac signal shows a strong correlation with them [45, 63], by capturing the subtle changes in heart rate. In this context, OCM provides an additional verification layer in a real-time and continuous manner, allowing experts to proceed to analyze behaviors. This analysis can help determine if someone is lying, happy, nervous, or engaging in deceptive behavior.

❸ **Entertainment:** Our work also presents a distinct chance for augmented entertainment. With the rise of live streaming, the audience can access the heart rates of celebrities, which opens up a new world for the existing viewing experiences.

Despite the potential, there are no existing solutions capable of achieving this integration without additional hardware. In this work, we focus on addressing this gap by leveraging the co-existence of audio and video signals, specifically in scenarios where a speaker is talking. This can be common in both entertainment and telehealth use cases, including affective computing, remote health, deepfake detection, *etc*. *At the core of OCM is the accurate prediction of cardiac information.* Our system should robustly detect the heart rate from the video streaming systems by hooking the video and audio chunks. Once cardiac data is acquired, it can be further analyzed for various downstream tasks, including affective computing, remote health monitoring, and deepfake detection. Yet how cardiac monitoring is used for downstream tasks (*e.g.*, emotions, lies, *etc*) is not the focus of this paper.

Audio-Video Pair: Leveraging the natural co-existence of audio and video modalities offers contemporary benefits as follows: ❶ Ubiquity: Video and audio streams are the most fundamental components in video streaming systems, while no additional hardware is needed. ❷ Feasibility: Both video and audio data contain the cardiac information (discussed in §3.1). ❸ Complementarity: Audio and video offer different strengths and weaknesses. Audio is less interfered with by motion and light but is sensitive to noises. Video is more robust to noises but will fail in various body movements and non-optimized view angles. We will elaborate the detailed analyses in §3.

To deploy such an OCM system, a straightforward way is to build a self-hosted WebRTC service, which, however, does not scale to existing video streaming systems. Therefore, for the sake of versatility, we aim to establish a microservice to host *CardioLive* for seamless integration with mainstream video streaming platforms.

Privacy Concerns: Audio and video data are inherently sensitive and vulnerable to privacy breaches. However, in our proposed scenarios, privacy concerns are mitigated for several reasons. First, the primary purpose of audio and video data in this context is for communication. Therefore, participants are already receiving this data during the meetings, regardless of whether our system is activated or not. In other words, all participants have consented to share their audio and video within the video streaming applications, without requiring extra sensitive data inputs. Additionally, our system is implemented as a middleware solution within existing video streaming systems. These contemporary systems are subject to stringent privacy regulations. *CardioLive* will operate in compliance with these established privacy frameworks.

In a nutshell, the audio-video pair appears to be an attractive choice for ubiquitous and practical OCM, yet it entails numerous challenges to build an accurate and robust multimodal algorithm and system. We will present our model design in §3 and leave the system implementation in §4.

3 CardioNet DESIGN

In this section, we will present our design of CardioNet. We will first describe the underlying fundamentals of inferring cardiac activity from video and audio. Then, we will illustrate our design of model.

3.1 Kinetics for Cardiac Learning

Principles: Since blood vessels circulate blood throughout the body, including the face, lungs, and throat, we can infer heart activity in these areas through video and audio analysis.

In video streams, when light hits the skin, subtle color changes from pulse-induced blood flow can be captured, as described by the Dichromatic Reflection Model (DRM) [50]. We define the Domain of Interest (DOI) of the facial areas as $\Pi \in \mathbb{R}^{N_v \times C \times H_f \times W_f}$, and $\Pi_{i,j} \in \Pi$ denotes the RGB pixels at the i-th row and the j-th column. To bridge the color with RGB values, we model the spectral relationship as:

$$\Psi_{\Pi_{i,j}}(f) = I(f) * \Delta(f), \tag{1}$$

where $I(f)$ is the illumination spectral components, $*$ is the convolution operation, and $\Delta(f)$ is the reflection modulator, comprising specular reflection $\Delta_s(f)$ and diffuse reflection $\Delta_d(f)$. Specular reflection occurs at the epidermis level, while diffuse reflection penetrates into the hypodermis, reflecting off capillaries and blood vessels, encapsulating physiological spectrum $H(f)$. We further decompose $I(f)$ and $\Delta_s(f)$ into static and dynamic components, where dynamic components are denoted as $\mu(H(f), O(f))$ and $\nu(H(f), O(f))$, respectively. $O(f)$ is a set of irrelevant signals. $\mu(\cdot)$ and $\nu(\cdot)$ are transfer functions without analytic expressions. Our goal is to infer $h(t)$ from Π, where $h(t)$ is the temporal counterpart of the spectral representation $H(f)$.

Speech is a complex auditory phenomenon that carries biological information. The airflow is produced from the lungs, which is then modulated by the vocal folds within the larynx to generate sound. This sound is further shaped by the movements and positions of the articulatory organs, such as the tongue and throat. Formally, the speech signal Ξ can be formulated in the frequency domain as

$$\Psi_{\Xi}(f) = L(f) \cdot R(f), \tag{2}$$

where $L(f)$ is the sound energy source. $R(f)$ is an acoustic filter creating formant, affected by the vocal tract's physical attributes. Blood flow in surrounding vessels, particularly carotid arteries, influences the acoustic properties [67].These cardiovascular dynamics are encapsulated in the model by integrating the physiological signal $\hat{H}(f)$ into $R(f)$.

Observations: Existing video-based solutions [20, 39, 40, 61, 72, 77], though many, are trained on small datasets with controlled environments, *e.g.*, PURE [51]. Their performances will degrade greatly when training and testing on more complicated datasets, *e.g.*, MMPD [54]. As can be seen from Fig. 2, the existing video-based solutions cannot effectively capture the cardiac semantics across different body movements and light conditions. These results present a grand challenge for cardiac learning. Meanwhile, different light conditions and body movements will degrade the performance from the video-based approaches, where audio can help [67]. Therefore, our goal is to design a dedicated audio-visual network to extract those motions.

3.2 Model Design

Given the underlying cardiac motions, we aim to devise a learning approach to extract $h(t)$. As shown in Fig. 3, the DOI pairs, *i.e.*, frames Π and audio clips Ξ, will be fed into video encoder E_v and audio encoder E_a, respectively, to acquire the latent representation. Then we devise a fusion network to aggregate the two modalities.

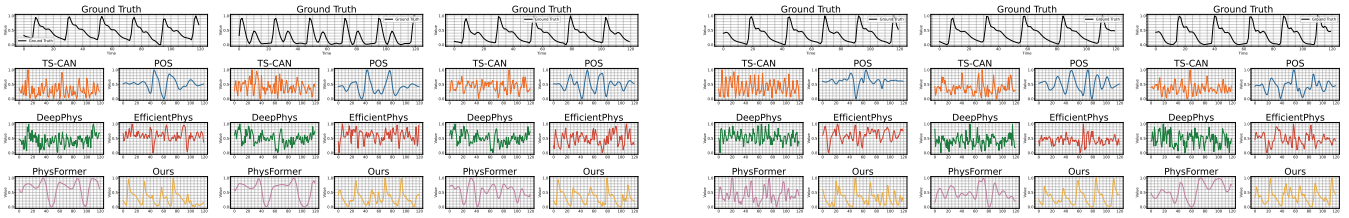
3.2.1 Video Branch Design. We will first introduce E_v.

Temporal Differential Block (TDB): The input video frames Π will first be processes as, *i.e.*,

$$\dot{\Pi}^t_{i,j} = \Pi^t_{i,j} - \Pi^{t-1}_{i,j}. \tag{3}$$

Note that in online learning, we only have past information, so we perform backward differentiation. The key idea is, we treat the psychological activities as tiny local "motions". It efficiently captures the changes between consecutive frames [60]. Furthermore, TDB plays a crucial role in isolating dynamic features while suppressing static components present in the video data, as stated in Eq. (1). Thereafter, they are fed into convolution networks and upsampled to meet the length of video features. It is also imperative to capture the static information inherent in the video frames. To this end, we integrate a parallel pathway to process the original video frames, allowing for a more comprehensive understanding of the environment. We then introduce lateral connections to facilitate fusion of static and dynamic information.

Motion-Aware Aggregation (MAA): The above design incorporates temporal information with static and dynamic modeling. After lateral fusion, we pass the intermediate latent to the bottleneck block. We recognize the importance of spatial modeling in mitigating the motion noise from head movement. Unlike video recognition tasks, where the relative location of the pixel is vital, we care more about how to track the variations of these pixels over time. To this end, we



(a) Stationary (b) Rotation (c) Talking (d) Low Light (e) Incandescence (f) Nature Light

Figure 2: The performances of video-based approaches vary under different body movements light conditions.

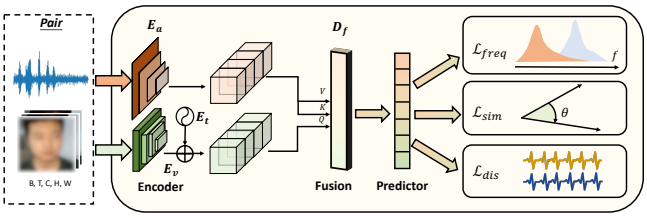


Figure 3: Overall Illustration of CardioNet

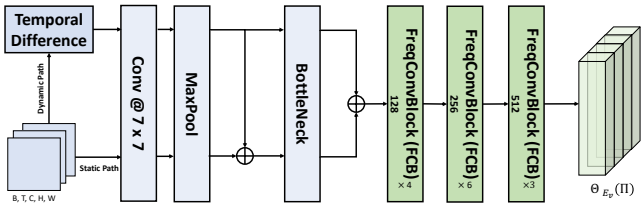


Figure 4: Video Encoder

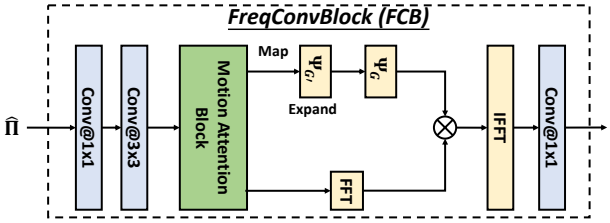


Figure 5: Frequency-Aware Convolution Block (FCB)

introduce a self-attention mechanism for frame-wise aggregation between consecutive frames. Our goal is to establish a mapping between temporal pixel variations and consecutive spatial information. Given the latent space $\hat{\mathbf{\Pi}} \in \mathbb{R}^{\hat{T}\times\hat{C}\times\hat{H}\times\hat{W}}$, we query the one pixel at time t, *i.e.*, $\hat{\Pi}^t_{i,j}$ and compute the attention with previous frame,

$$\rho^t = \text{Softmax}\left(\frac{\hat{\Pi}^t_{i,j}\cdot\left(\hat{\Pi}^{t-1}_{i\pm\Delta i,j\pm\Delta j}\right)^T}{\sqrt{d_k}}\right). \tag{4}$$

Here $\Delta i = \Delta j = k/2$, which is the perception grid size. d_k is the dimension of $\hat{\Pi}^{t-1}_{i\pm\Delta i,j\pm\Delta j}$. ρ^t captures the inter-frame pixel displacement, drawing attention to motion while enhancing temporal features between frames. Subsequently, we can get the weighted sum of temporal neighbor frames and aggregate with a query to enhance the original pixel:

$$\check{\Pi}^t_{i,j} = \hat{\Pi}^t_{i,j} + \rho^t \cdot \hat{\Pi}^{t-1}_{i\pm\Delta i,j\pm\Delta j}. \tag{5}$$

This mechanism scrutinizes pixel displacements across consecutive frames, akin to tracing the path of movement within a sequence of images. Each pixel's attention weight encapsulates its significance in depicting motion, allowing the model to recognize subtle shifts and fluctuations over time.

Frequency-Aware Block (FAB): After applying motion attention aggregation, we acquire the enhanced feature $\check{\mathbf{\Pi}} \in \mathbb{R}^{T\times\hat{C}\times\hat{H}\times\hat{W}}$. Our previous focus has been on modeling video dynamics in the temporal domain. These are very effective designs for cardiac time series learning. Moreover, given the intrinsic property of $h(t)$, which turns out to be a quasi-periodic signal, it becomes imperative to incorporate frequency features into our analysis. Here, the term "frequency" does not merely refer to the spectrum of color space within the video; rather, we aim to capture the underlying frequency variations of pixels over time. Inspired by DTF [42], we attempt to explicitly incorporate FFT in our design. For each pixel $\check{\mathbf{\Pi}}_{i,j} \in \mathbb{R}^{T\times\hat{C}}$, we apply FFT along the temporal dimension to acquire the feature spectrum $\Psi_{\check{\mathbf{\Pi}}_{i,j}}(f)$. To capture the frequency information, we introduce a learnable frequency filter $\Psi_G(f) \in \mathbb{R}^{\hat{C}\times N_f}$. We use IFFT to get the modulated temporal feature. With FCB, we can enlarge the receptive field and profile cardiac time series with frequency constraints.

Irregular Sampled Time Embedding: Another challenge of online cardiac learning is the fluctuating FPS. To this end, we introduce the timestamp feature to handle the irregular sampled time learning. Technically, we can acquire the set of timestamps $\{t_i\}_{i=1}^{N_v}$ for each frame. We incorporate a timestamp embedding E_t design and fuse it with $\mathbf{\Theta}_{E_v}(\mathbf{\Pi})$. Specifically, we employ a frequency embedding scheme, which computes triangle embedding based on a geometric progression of frequencies up to f_m. We first derive a set of frequencies with the size of embedding dimension N_{t_d}, *i.e.*,

$$\omega^k = \exp\left(\frac{2k}{N_{t_d}}\cdot\log(f_m)\right), \tag{6}$$

where $k = 1, \cdots N_{td}/2$. Then the angle for each timestamp i is given by $\theta_i^k = t_i \cdot \omega^k \cdot 2\pi$. Finally, the timestamps are embedded through trigonometric encoding by concatenating sine and cosine values for each angle.

3.2.2 Audio Branch Design. We then introduce the design for the audio encode E_a. As discussed in §3.1, human speech is modulated by the time-varying filter $R(f)$. And the cardiac series is encapsulated in $R(f)$. Inspired by this filtering process, we opt to emulate it within our design. We target directly processing raw audio in our case. We will justify the rationale first, followed by our design.

Raw Audio: Traditional audio-based learning often leverages mel-spectrogram, a common practice for tasks like speech recognition. However, this method may not be suitable for our task. Our predictions, $h(t)$, manifest as quasi-periodic signals, ideally shown as straight lines on a mel-spectrum. But because cardiac activities are variable, these lines will exhibit randomness on a temporal-frequency map. Also, the location of the "straight" line has physical meanings, rather than a simple pattern. Therefore, we resort to learning from the raw audio signals directly. The key insight is, the process of producing speed from our vocal organs is composed of several acoustic filters, as indicated in §3.1. We can simulate the effect of filters and incorporate them in our design.

Temporal-Frequency Filter (TFF): The temporal format of Eq. (2) can be rewritten as $\xi(t) = l(t) * r(t)$, where $\xi(t)$ is the speech signal. $l(t)$ represents the source of the sound while $r(t)$ is combination of source filters. To this end, we adopt the SincNet [48], which can be expressed as,

$$r_i(t, \theta) = 2f_{i,2}^{\theta} \operatorname{sinc}(2\pi f_{i,2}^{\theta} \cdot t) - 2f_{i,1}^{\theta} \operatorname{sinc}(2\pi f_{i,1}^{\theta} \cdot t). \quad (7)$$

$f_{i,2}^{\theta}$ and $f_{i,1}^{\theta}$ denotes the two cutoff frequencies. We can treat the two cutoff frequencies as learnable parameters. We then perform convolution between $r_i(t)$ and raw audio $\xi(t)$. They will be fed into 1D convolution blocks for feature extraction.

3.2.3 Fusion Block Design. Until now, we have handled the video feature $\Theta_{E_v}(\Pi)$ and audio feature $\Theta_{E_a}(\Xi)$. We now present the design of the fusion network. We opt for the late-fusion scheme, as the relationships between audio and cardiac activity, as well as video and cardiac activity, are not initially apparent. Within the fusion block, we aim to address two challenges: 1) aligning the audio and video features along the temporal domain, and 2) handling the sampling rate mismatch between the audio and video features. To do so, we propose a multi-head temporal attention fusion block. Subsequently, the fused feature will be passed through linear fully connected layers. To achieve this, we adopt a temporal attention-based fusion scheme. Technically, we exploit video features as the query, and audio features as the key and value. The fused feature $\Theta_f(\Pi, \Xi)$ will be fed to the output layer.

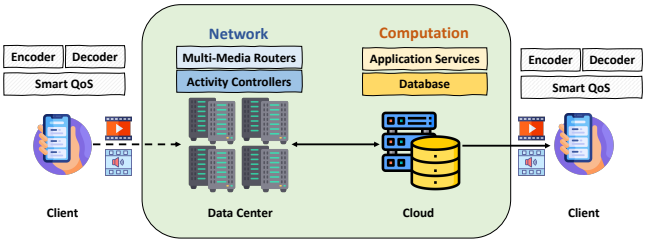


Figure 6: The architecture of a video streaming system.

3.2.4 Loss. We include three types of loss functions, *i.e.*, focal loss, frequency loss and similarity loss,

$$\mathcal{L}_{\text{all}} = \alpha \cdot \mathcal{L}_{\text{dis}} + \beta \cdot \mathcal{L}_{\text{sim}} + \gamma \cdot \mathcal{L}_{\text{freq}}, \quad (8)$$

where α, β and γ are weights to balance the loss items. The focal loss $\mathcal{L}_{\text{dis}}$ offers a more robust framework for keeping peaks in the physiological signals [49]. The similarity loss $\mathcal{L}_{\text{sim}}$ represents the extent of alignment. Additionally, as we are learning quasi-periodic signal, we incorporate spectral loss $\mathcal{L}_{\text{freq}}$ as well by calculating the MSE of FFT.

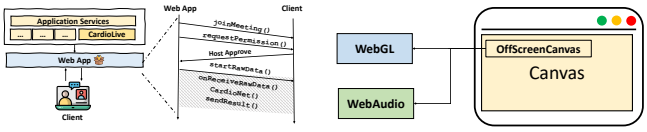
4 *CARDIOLIVE* DESIGN

In this section, we will introduce the design of *CardioLive*. We will introduce the design goal of *CardioLive* in §4.1, followed by our detailed designs of service in §4.2 and §4.3. We will introduce the preprocessing in §4.4.

4.1 Design Goal

Modern video streaming systems are complicated, and integrating OCM into them is non-trivial. As shown in Fig. 6, the content is sent through cloud servers spanning across different locations globally. Besides running the data center and cloud computing, these video streaming systems offer a range of application services, such as content summarization, transcriptions, and AI-driven interactive features. Note that for content providers like YouTube, Netflix, and many VoD providers, integrating new features is relatively straightforward because they can preload resources in their data centers. However, this does work well with streaming systems with live content generation and interactions. On the other hand, deploying cardiac monitoring on edge devices is also valuable. Users will be concerned about how the sensitive data are communicated over the network.

Therefore, to achieve SoD cardiac monitoring service, we need to both consider deploying the cardiac monitoring services on the edge ends, *e.g.*, browsers, and application services. Notably, direct access to data that manufacturers possess is often restricted by stringent privacy regulations affecting external developers. To this end, we aim to package *CardioLive* into a service, which both end users and manufacturers can readily access. At a high level, we are not



(a) In-app bot (b) Web extension

Figure 7: Data Hook Design.

concerned about specific implementations on specific platforms, but aim to develop *CardioLive* as a *microservice*. We utilize data hooks to capture video and audio streams, organizing them into buffer queues as data packets, which will be fed to the inference engines, as detailed below.

4.2 Buffer Design

Data Hook: We will first introduce data hooks to get the video and audio streams, namely `onVideoDataReceived()` and `onAudioDataReceived()`. Meeting platforms like Zoom usually support in-app bots that virtually participate in calls. We can leverage the bots to access the raw data streams, as shown in Fig. 7a. Meanwhile, increasingly more video streaming systems are based on web pages, *e.g.*, YouTube, Bilibili, *etc.* Direct accessing the video streams of this platform is rather complicated and violates the policies. To this end, we leverage WebGL and WebAudio that exist in modern browsers to get the data streams, as shown in Fig. 7b. The browsers usually provide the Document Object Model (DOM), a programming interface to manipulate the structure, style, and content of web content. Our service will first access the canvas, an element for graphics on a web page, through DOM. The canvas offers a bitmap where each pixel can be individually manipulated. We get the rendering context through WebGL, which operates as a rendering context of canvas using the underlying GPU. We create an offscreen canvas that is rendered off the main thread and read the pixels through WebGL, preventing it from interfering with the normal UI updates. Meanwhile, we capture the audio from the video element through WebAudio, a versatile framework to handle audio operations on the web. We record the timestamp of the audio and video as well. Through the data hook, we can acquire the video and audio streams. Then we will construct them into data packets and buffer queues.

Data Packet: Normally, audio and video are encoded in separate ways. In meeting platforms, the video frames are usually encoded in YUV format. They are designed for the best transmission efficiency. Encoding the data in YUV space allows fewer total bits of space in a video stream for the colors to be shown. To recover the original RGB streams, we have first to decode the YUV streams. To reduce the cost of decoding, we adapt a streaming-based decoding pipeline from GStreamer [2]. We set the `appsink` property for receiving the RGB data and assign `appsrc` for handling YUV encoding. We set the

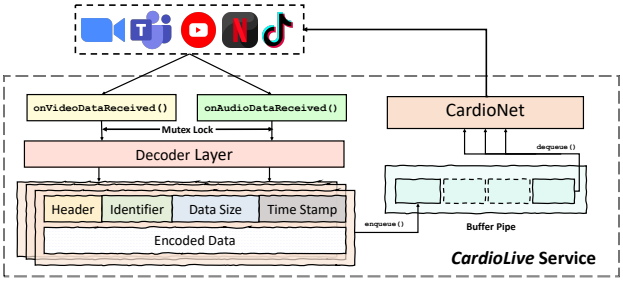


Figure 8: Packet and Buffer Design

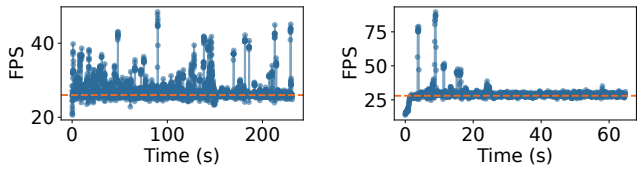
transformation in asynchronous mode so that the incoming frames will not conflict with the current operations. After that, we will construct the collected frames in buffers.

Then we feed the video-audio pair into the forwarding packets. For audio and video streams, we apply the same packet format, which contains a unique header, an identifier, the data size, timestamps, and the encoded payload data, as illustrated in Fig. 8. The unique header is designed to judge whether the packet is correctly constructed and not mixed with other packets. The identifier is assigned to indicate whether it is audio or video data packets. We embed the received timestamps to denote the sequence of the video and audio, which will be further used for synchronization.

4.3 Service Design

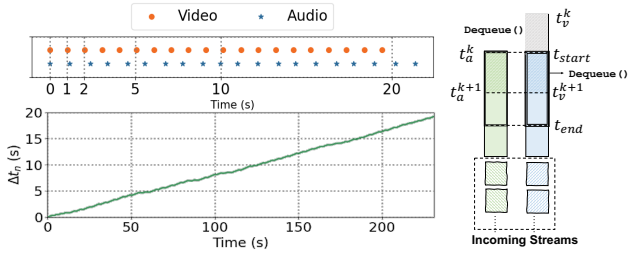
We abstract our system as a plug-and-play service. Our service first gets the hooked video and audio packets as the input. The data will be preprocessed and fed to the inference engine for output. Our design overcomes the two challenges: fluctuating FPS and unsynchronized audio and video streams.

Changing FPS: The fluctuating FPS will lead to two subproblems. Initially, the video streaming systems will ideally have 30 FPS but in reality undersampled at the receiver's end, as illustrated in Fig. 9, with some outliers present as well. Additionally, the frame rate is not constant, resulting in a varying number of frames within a given window. However, our model assumes a fixed 4-s input, with 120 frames of video (30 FPS) and 32000 samples of audio (8kHz). In other word, we have to adapt the real input size to the model. To this end, instead of padding empty frames at the end, we duplicate one single frame circularly. For instance, if the actual FPS is 25, we insert an additional identical frame after every 5 frames to approximate a smoother transition to 30 FPS. Any remaining gaps at the end of the sequence are filled by repeating the last frame. As for overlarge FPS, we downsample the frames. For the audio clips, as 8kHz is much lower than the typical sampling rates (usually 32kHz or 44.1kHz) in modern video streaming systems, we can concatenate the received audio chunks and safely downsample them to 8kHz.



(a) Chrome **(b) Zoom**

Figure 9: The FPS vary and change rapidly.



(a) The temporal drifting **(b) Sync**

Figure 10: Audio-Video Synchronization Scheme.

Audio Video Synchronization: The audio-visual misalignment is a more severe issue than changing FPS. As we are hooking audio and video from separate channels, they are likely to lose synchronization with the increase of time. As can be seen from Fig. 7b, the starting time of the audio and video will be misaligned quickly with accumulating drifts. To overcome this issue, we develop a scheme to ensure the audio and video chunks are synchronized before sending to the inference engine. Given the audio and video streams $S_a(t)$ and $S_b(t)$, they will be extended to the buffer queues $Q_a(t)$ and $Q_v(t)$, respectively. We also maintain t_a and t_v as the starting time of audio and video chunks, respectively. We denote $\Delta t_n = t_a^k - t_v^k$ as the temporal drift between audio and video streams at k-th trial. To mitigating the continuously increasing Δt_n, we align the start time at each step k as, $t_{\text{start}}^k = \max(t_a^k, t_v^k)$, when Δt_n is larger than the threshold ϵ_t. We use $\epsilon_t = 0.3$s. Then the ending time will be determined by $t_{\text{end}}^k = t_{\text{start}}^k + t_w$, where t_w is the window lengths. Note that we are adopting a sliding window scheme, with window length t_w and step length t_s. For the next window, the start time will be updated by finding the timestamp closest to, *i.e.*, $t_a^{k+1} = t_a^k + t_s$ and $t_v^{k+1} = t_v^k + t_s$. Meanwhile, we will pop the items that have been processed from the buffer queues, *i.e.*, $Q_a(t) = Q_a(t) \backslash \{S_a(t) | t < t_a^{k+1}\}$ and $Q_v(t) = Q_v(t) \backslash \{S_v(t) | t < t_v^{k+1}\}$. We then feed the synchronized pairs for inference.

4.4 Preprocessing

In this section, we will discuss the preprocessing pipelines. We use OpenCV face detector to find the faces. We also perform voice activity detection to segment the talking period.

Figure 11: Experimental Setups

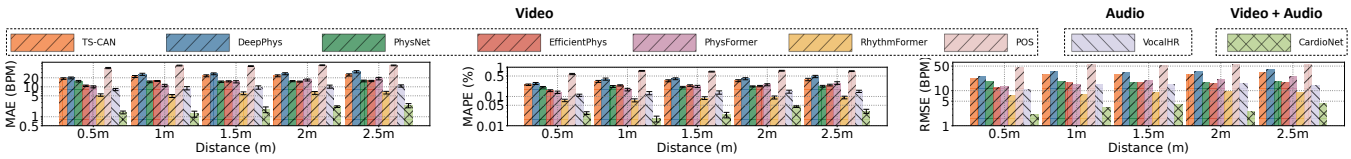
Additionally, we need to separate multiple persons, if any, and match their audio and videos.

Multi-person Separation: We deduct the more challenging multi-user case into the single-user case by separating them. Initially, face detection can determine the number of participants. To ensure facial resolution, we focus on the largest N_f faces, disregarding the others. Similarly, we will only consider N_f speech clips with the largest power spectrum when separating audio. For efficiency, we choose $N_f = 2$ in our paper. At this stage, the separated faces and speech segments may not correspond to each other. To address this mismatch, we proceed with audio-visual matching as described next.

Audio-Visual Matching: To realize the matching between speaking clips and facial hints, we adopt a cross-attention scheme [31, 56]. Specifically, after the encoders, we get two features M_a and M_v. These features are expected to encapsulate relevant speaking activities by employing temporal encoders [14, 30]. To fuse the audio and video features, The audio features M_a are integrated with the video data by treating M_v as the target for querying through an attention framework. Conversely, the video features M_v interact with Q_a, representing the audio query sequences. The outputs are concatenated together along the temporal direction.

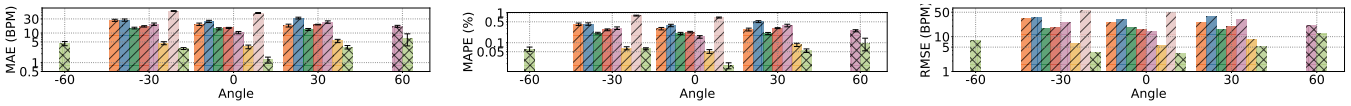
5 EVALUATION

In this section, we systematically evaluate *CardioLive*. We perform comparison studies with the state-of-the-art (SOTA) video-based solutions and audio-based solutions. We mainly leverage our self-collected dataset. We use the following metrics to evaluate the accuracy of the model: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). **Data Collection:** There is no existing dataset that can fit our requirements, with audio-visual pairs and clear heart rate ground truth. Therefore, we self-collect the dataset through 8 commodity devices which span multiple mainstream platforms (iOS, Android, Windows, Mac), major brands, and device types (smartphones, tablets, laptops, and webcams) released between 2018-2023: Logitech C930 Webcam, OPPO Reno 2Z, Redmi Note 5, Honor 20i, MacBook Air M2, iPad 2018, iPad 2023 Pro, and iPhone 14. We leverage Polar H10 [7] to collect the ground truth. Our dataset comprises recordings from 10 users of diverse genders and regions. They are requested to read 10 materials [52], counting for 2,800 words. Each round lasts for 40 minutes. They wear the heartbeat belt when they



(a) MAE **(b) MAPE** **(c) RMSE**

Figure 12: The performances for different distances.



(a) MAE **(b) MAPE** **(c) RMSE**

Figure 13: The performances for different angles.

are reading. We do not restrict users to a fixed distance from the recording devices. We leverage a tripod along with a ring light to cast different light sources on the users. We collected a total of 84,666 data clips, which are clipped into facial regions with 4-s windows. We resize the video frames to 72×72×3 and the audio is resampled to 8kHz. The missing frames will be duplicated adopting the same scheme as §4.3 mentions. We gain IRB from our university board. We also make use of two publicly available video-only datasets: PURE [51] and MMPD [54] to validate the video-only solution.

Software: We implement `CardioNet` through Pytorch 2.4.0. The model is trained via a single-card NVIDIA A100 80GB. We train the model with the learning rate of 1e-3, AdamW optimizer, batch size of 16, OneCycle scheduler. We use Pytorch JIT to compile the model. We write 2000+ lines C++ code to implement the service in Zoom and 1500+ lines JavaScript code for developing the service in the extension.

Deployment: We propose two deployment paradigms, web-based and app-based. For the web-based one, we develop a browser extension that operates *CardioLive* in the background, which continuously captures audio and video data for processing, with results displayed on a canvas within the interface. In the app-based deployment, we register a bot in compliance with the policies of the video streaming companies. This bot joins the sessions as a member, similar to other participants, with the consent of all members. The data hook extracts audio and video towards inference engines. The processed results are delivered through a notification system. Notably, the inference can be performed either on the company's cloud server or locally on the user's device. In our real-world evaluation, we perform inference on the user's device (Xiaoxin 16 Pro with AMD Ryzen 7 5800H), demonstrating the robustness and efficiency of our model.

5.1 Comparative Study

We compare our `CardioNet` with various baselines. We choose the SOTA video-only baselines: TS-CAN [39], DeepPhys [20], PhysNet [72], EfficientPhys [40], RhythmFormer [77], POS [61]. The last one represents the signal processing based rPPG approaches. We also reimplement VocalHR [67], the recent work that employs human speech for detecting heart rate. Through this study, we will justify our superior performances using both audio and video modalities.

Distances: We first experiment with different distances, ranging from 0.5m to 2.5m. We apply the logarithmic scale to each graph, with the base of 10. As shown in Fig. 12, while the error increases with distance for all methods, our approach consistently outperforms other baseline models at all tested distances. `CardioNet` achieves a MAE of just 1.40 BPM at 0.5m, significantly lower than the SOTA video-based baseline, *i.e.*, RhythmFormer, by 73.7 %, and 96.7% lower than the worst-performing model, *i.e.*, POS. Meanwhile, the audio-based model VocalHR has a MAE of 8.12 BPM at the same distance, which is 82.8% higher than ours. Even at the maximum testing distance of 2.5 meters, `CardioNet` is still 63.1% better than RhythmFormer and 77.9% better than VocalHR. This demonstrates the fusion of audio and video signals in `CardioNet` significantly enhances the overall performance. Besides, we observe the identical patterns of MAE, MAPE and RMSE, we will mainly report MAE for simplicity.

Angles: We evaluate our model across a range of angles from 0° to ±60° at a distance of 1 meter, as shown in Fig. 13. As the viewing angle increases, video-based methods suffer from significant performance degradation due to reduced visibility of facial features. However, `CardioNet`, through audio-visual fusion, maintains robust performance across all angles. While the video quality deteriorates with extreme angles, audio signals remain largely unaffected by viewing angles. Even at extreme angles like ±60°, where video signals typically falter, our model achieves up to 38.9% lower MAE compared to baseline models, highlighting its superior resilience in challenging conditions. This result underscores the critical role of the audio modality in compensating for the loss of visual information at extreme angles.

Noise Levels: We test heart rate estimation under noise levels from 30 dB to 38 dB. As in Fig. 14, increasing noise

leads to higher absolute error. Nonetheless, CardioNet consistently outperforms the SOTA audio-only model VocalHR. This can be attributed to our temporal frequency filter design and the video modality which provides complementary information that remains stable under acoustic noise. For instance, at 30 dB, our model achieves a MAE of 1.25 BPM, significantly lower than VocalHR's 8.64 BPM, and maintains this advantage even at 38 dB. The fusion network learns to adaptively reduce reliance on noisy audio features while leveraging more stable visual cues. The CDF curves show that CardioNet achieves higher cumulative probabilities at lower error thresholds, indicating its resilience to noise.

Noise Sources: We analyze the impact of noise sources such as rain, music, and TV shows in Fig. 15. CardioNet demonstrates strong noise resilience, particularly with rain noise, where it significantly outperforms VocalHR, achieving a MAE of just 1.94 BPM compared to 12.93 BPM. Even with more complex noise like music and TV shows, our model maintains lower MAEs, showcasing its robustness in diverse acoustic environments. This highlights the effectiveness of video modalities when facing the ambient noises.

Body Motions: Body motion can significantly impact the performance of heart rate detection models. To validate the robustness of our approach under different body motion scenarios, we evaluate the model in three typical body movements: walking, left-right (LR) rotation, and up-down (UD) rotation, as in Fig. 16. Despite the motion artifacts, CardioNet maintains robust performance, achieving an MAE of 1.35 BPM in the UD scenario, and consistently outperforms baselines by significant margins in all motion types. Our model benefits from the unique design of the motion-aware aggregation and temporal differentiation block. These prove the robustness of our model against body motions by effectively employing video plus audio modalities.

Video-only Solutions: We evaluate our approach on open datasets that contain only video data. As shown in Fig. 17, our method consistently ranks among the top among rPPG-based solutions. We achieve MAE errors of 2.09 and 1.12 BPM on PURE and MMPD datasets, respectively. It is important to note that during evaluation, we disable the audio branch of CardioNet. This ensures that our video encoder independently captures heart-related activities. In scenarios where no audio is available (e.g., during silent periods), our model effectively transitions into a video-only solution.

5.2 Micro Benchmarks

Different Light Conditions: We assess our model under varying lightness levels from 0.3702 to 0.3259 in Fig.18a, by adjusting the ring light. As lightness decreases, the MAE increases from 4.85 BPM to 8.16 BPM. This trend suggests that poorer conditions impact accuracy due to the reduced

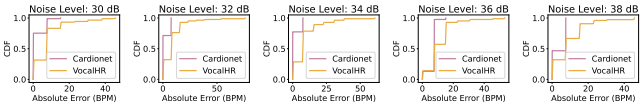


Figure 14: CDF for different noise levels.

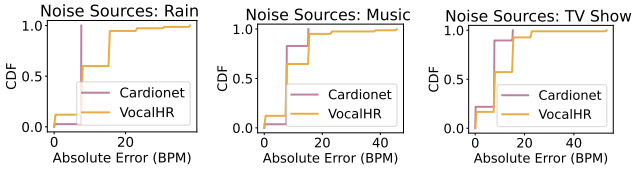


Figure 15: CDF for different noise sources.

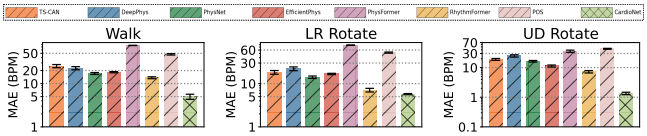
visibility of facial features. However, the model remains sufficiently robust, indicating that while lighting plays a role, the audio-visual fusion helps mitigate the negative effects.

Different FPS: We examine the model across various video frame rates (FPS), ranging from 30 FPS to 15 FPS, as shown in Fig.18b. We interpolate the frame rate by adopting the principles discussed in §4.3. The model performs best at 30 FPS with an MAE of 1.75 BPM. Even at lower frame rates, particularly 15 FPS, the MAE increases to 4.56 BPM, while still remaining in the low level. This robust performance is achieved through our frame interpolation scheme and the audio branch's ability to provide continuous cardiac information regardless of video frame rate. Also, our temporal differential block and irregular sampled time embedding block are equally vital to handle varying frame rates.

Different Quality of Image: We analyze the model's performance under different video compression qualities, from 100 (highest quality) to 40 (lowest quality), as shown in Fig.18c. Interestingly, the MAE does not consistently worsen with lower quality. At extreme compression levels, the model achieves the lowest MAE of 2.49 BPM, potentially due to smoothing effects that enhance key facial features. This suggests that while high compression degrades visual information, moderate to high levels of compression might benefit the model by reducing noise.

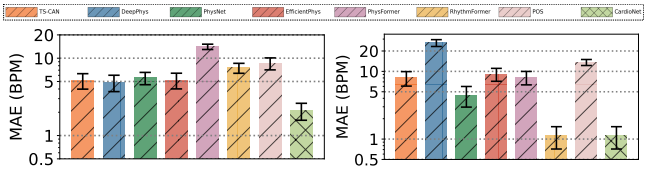
Different Environments: Our model's performance is evaluated across various environmental settings, including Office, Outdoor, Conference Room, and Laboratory, as shown in Fig.19a. The model performs best in the Office environment with a MAE of 1.40 BPM. Notably, the latter three environments are not in the training set, yet the model maintains strong performance, demonstrating that our feature extraction generalizes well to unseen conditions.

Different Face Filters: We test various facial filters, including Smooth Face, Tint Skin, Adjust Brightness, Add Contrast, and Sharpen Face, as shown in Fig.19b. The Tint Skin filter yields the best performance with an MAE of 2.38 BPM, while a more aggressive filter like Sharpen Face achieves an MAE of 8.69 BPM. It shows our model effectively handles appearance changes while maintaining accuracy.



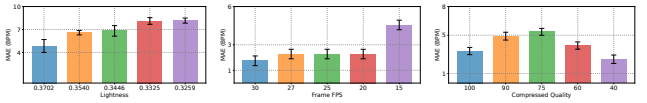
(a) Walk **(b) Left-Right** **(c) Up-Down**

Figure 16: The performances of different body motions.



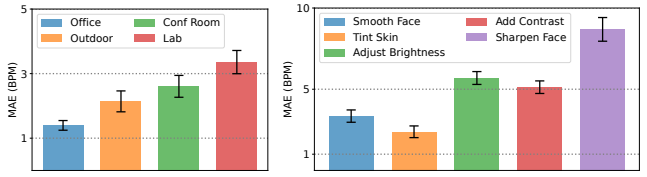
(a) PURE **(b) MMPD**

Figure 17: The performances on public datasets.



(a) Lightness **(b) FPS** **(c) Quality of Image**

Figure 18: The performances under different light, FPS and image conditions.



(a) Environment **(b) Face Beauty**

Figure 19: The performances under different environments and face beauty filters.

Different Devices: We evaluate our model on various devices under inter-device and cross-device conditions, as shown in Fig.20. For inter-device testing, the average MAE is approximately 2.95 BPM. In cross-device scenarios, the average MAE is around 8.07 BPM. While there is a drop in accuracy, the model still delivers acceptable performance across different hardware platforms. This suggests that despite some variability, the model remains robust and capable of providing reliable heart rate estimates on a wide range of devices.

Different Users: We evaluate our model's performance across a diverse set of users in Fig.21. Our model's user generalization capability stems from learning universal cardiac patterns rather than user-specific features. The temporal-spectral modeling captures fundamental physiological characteristics that are consistent across individuals. Under inter-user conditions, the average MAE is about 1.93 BPM. In cross-user scenarios, the model still performs reasonably well, with an average MAE of 7.53 BPM. Despite the diversity, the model maintains a usable level of accuracy, underscoring its generalizability across different user groups. This demonstrates that our feature extraction pipeline effectively captures device-independent cardiac patterns.

Multi-person Scenarios: We evaluate the multi-person scenarios to justify the effectiveness of our preprocessing. We set the maximum number of people to be separated as two and crop the face region to a size of 72×72 pixels. In our test, two users read materials simultaneously while sitting next to each other. We apply the facial and sound separation and match their audio and face regions. The test results show an MAE of 7.83 BPM and 8.13 BPM for each person, respectively. Although we observe some performance drops, our method still effectively distinguishes between the two individuals. Notably, the heart rates of the two people vary over time, with average heart rates of 76.17 BPM and 68.55 BPM, respectively ,showing our system can track distinct physiological states simultaneously.

5.3 *CardioLive* in the wild

In this section, we will evaluate how *CardioLive* works as a service. We assess the service on two ends: the meeting platform and the online content providers.

Meeting Platforms: We choose Zoom as one of online meeting platforms, which provides the external developers the SDK to acquire access to the raw data. The average FPS is 28.4. We exploit the data hooks to acquire the streams and leverage buffer queues to hold the packets, as described in §4.2. The model consumes on average in 850ms on CPU. We choose a step size of 1s, and a window size of 4s. It means every second, we feed the 4-s windows for inference. The overall system latency averages 1.03 seconds, as depicted in Fig. 22b. Notably, latency was primarily elevated at the start due to the initial model warm-up period [36]. This means our systems can run inference in real-time. Furthermore, we calculate the throughput of the whole system. We measure the time since the last update of heart rate. As we are feeding 4-s window of video and audio frames, the throughput is calculated as the volume of video and audio data processed per update period. As in Fig. 22b, the average throughput of the system is 115.97 FPS, which is prominently larger than the common video FPS. It means that our systems can hold the service robustly without any freezes.

Online Content Providers: Online content providers such as YouTube often host their services in the web browser. We implement such a service in a Chrome extension. We employ the data hook developed from WebGL and WebAudio to acquire the streams. The average FPS is 26.97. The overall latency of our service is 1.23s, comparable to our step size 1s, as can be observed from Fig. 22a. Meanwhile, the average

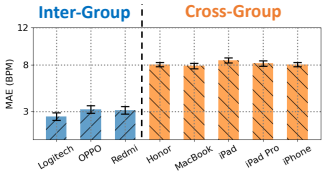


Figure 20: Different Devices

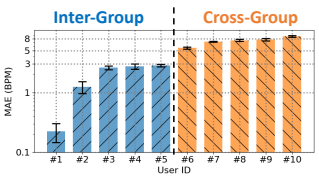
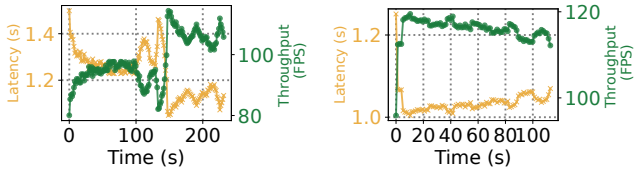


Figure 21: Different Users



(a) Chrome **(b) Zoom**

Figure 22: Latency & throughput for Zoom and Chrome

throughput is 98.16 FPS, with a maximum throughput of 114.41 FPS. These results also justify our service will run smoothly in the extensions.

6 RELATED WORK

In this section, we will summarize the existing works.

Cardiac Monitoring: Cardiac information is crucial for health monitoring, affective computing [26, 68] and deception analysis [16]. Except electrocardiograms (ECGs) and CT scans [3] that are prohibitively expensive and cumbersome for everyday use, recent advancements have focused on more portable solutions. Earable-based systems [17, 19, 27] allow earpieces to detect cardiac information, but they either need specific probing signals or custom hardware, limiting their widespread adoption. Similarly, wearable solutions necessitate constant wear, which is not practical for all users. Wireless technologies, including Wi-Fi [38], mmWave [70], and UWB [21], *etc*, are constrained by specific hardware which is not commonly available in video systems . Solutions using active acoustic sensing [47, 58, 59, 75] with smart speakers rely on pseudo-inaudible signals, which can be intrusive to human hearing and increase hardware burden. Video-based solutions use optical means to measure blood volume changes in tissues. Signal processing [22, 34, 61, 62] and deep learning [20, 33, 39, 40, 44, 72–74, 77] techniques have been developed to enhance these methods. Yet these solutions are sensitive to low light conditions, head/body movements, and typically perform poorly outside controlled environments. VocalHR [67] proves the potential of extracting heart rate from human speech. Although it leverages human speech effectively, it is limited by range, requires pre-calibration, and cannot distinguish multiple individuals. Differently, *CardioLive* is the first to combine the complementary and naturally co-existing audio and video modalities in online video streaming systems. Our video design incorporates temporal-frequency co-design and motion-aware aggregations for the first time in OCM to mitigate the light and body movement influence. The audio module employs the temporal acoustic filter for OCM. These designs are innovative and contribute to our performances.

Video Streaming System: Video streaming systems have gained immense popularity due to their vast libraries of on-demand content, user-generated videos, and live streaming capabilities, catering to diverse viewer preferences, including YouTube, TikTok, Zoom, *etc*. They can be further categorized into VoD systems, live streaming systems and video conferencing systems. Research efforts have been devoted to communication protocols [24, 29], adaptive rate streaming algorithms [35, 64, 76], online learning [28, 32, 55, 71], *etc*. None of these works explore adding cardiac monitoring into modern video streaming systems. In contrast, *CardioLive* stands out as the first work that creates a middleware service of OCM that can be seamlessly integrated into mainstream video streaming systems.

7 DISCUSSION AND FUTURE WORK

Audio-Video Pair: In our primary application scenarios (*e.g.*, live streaming, online meetings, *etc*), audio and video naturally coexist. In practice, only video data is available in some situations, where *CardioLive* can be easily adapted to a video-only solution. Such periods can be detected through mature voice activity detection techniques [65]. Our results shown in Fig. 17 have demonstrated that *CardioLive* also performs well in video-only scenarios. *CardioLive* not only introduces a novel approach to OCM by utilizing audio-visual pairs for the first time, but also integrates these capabilities into a practical system with flexibility and robustness.

Impacts on Original Streams: Integrating additional services into standard streaming platforms has been a bottleneck for many previous solutions [18, 25, 41]. In *CardioLive*, we address this challenge with a dedicated design of data hook and middleware service. Our approach ensures that these additional services are isolated from the original streams. With offscreen canvas, which operates independently in the extension, we avoid disrupting the original content. In meetings, our data hook duplicates data to the inference engine instantly, seamlessly, and without affecting the main video and audio streams. Our evaluations demonstrate that *CardioLive* operates without causing any disruptions or interference to ongoing streams.

Equality and Accessibility: *CardioLive* is designed for equality and is devised to be flexible and adaptable, allowing it to be integrated into any platform without the need for specialized hardware. This significantly increases accessibility, making the technology available to a wider audience. Moreover, while companies can promote this service on cloud

platforms, *CardioLive* is crafted to ensure democratized access, preventing any hidden biases or preferential treatment. By enabling audiences to independently initiate the service, *CardioLive* reduces the likelihood of companies manipulating the system for economic gains by altering the model.

Use of Deep Learning: The relationship between video-audio information and cardiac activity is inherently implicit and complex. We evaluate our results against signal processing approaches in Fig. 12 and Fig. 13, where our performances are significantly better. And our system evaluation validates real-time monitoring without introducing large latency. We identify the exploration of combining signal processing with increased explainability as a direction for future work.

8 CONCLUSION

In this paper, we envision the attractiveness of Online Cardiac Monitoring (OCM) in video streaming and present *CardioLive*, the first-of-its-kind system to fuse both audio and video streams for online cardiac monitoring in video streaming systems. We devise an effective audio-visual network that can robustly and accurately unveil the nuanced cardiac activities, achieving an average MAE of 1.79 BPM and outperforming the video-only and audio-only solutions by 69.2% and 81.2%, respectively. Furthermore, we design and implement *CardioLive* as a plug-and-play microservice that can seamlessly be integrated into mainstream video streaming systems. We believe our work will significantly enhance the entertainment and healthcare value of video streaming and inspire a new direction in this field.

REFERENCES

[1] . Explore - Find your favourite videos on TikTok. https://www.tiktok.com/explore

[2] . GStreamer. https://gstreamer.freedesktop.org/documentation/?gi-language=c

[3] . Heart disease - Symptoms and causes - Mayo Clinic. https://www.mayoclinic.org/diseases-conditions/heart-disease/symptoms-causes/syc-20353118

[4] . Netflix Singapore – Watch TV Programmes Online, Watch Films Online. https://www.netflix.com/sg/

[5] . One platform to connect | Zoom. https://zoom.us/

[6] . Paris 2024: Parents to be fitted with heart-rate monitors as part of Olympic Games coverage. https://olympics.com/en/news/paris-2024-parents-heart-rate-monitors-olympic-games-coverage

[7] . Polar H10 | Polar Global. https://www.polar.com/en/sensors/h10-heart-rate-sensor

[8] . Pulsoid - a real-time heart rate widget for streaming. https://pulsoid.net/

[9] . Skype | Stay connected with free video calls worldwide. https://www.skype.com/en//

[10] . Stream TV and Movies Live and Online | Hulu. https://www.hulu.com/welcome?orig_referrer=https%3A%2F%2Fwww.google.com.hk%2F

[11] . Teams and Channels | Microsoft Teams. https://teams.microsoft.com/v2/?clientexperience=t2

[12] . Video Streaming (SVoD) - Global | Statista Market Forecast. https://www.statista.com/outlook/dmo/digital-media/video-on-demand/video-streaming-svod/worldwide

[13] . YouTube. https://www.youtube.com/

[14] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. 2018. The conversation: Deep audio-visual speech enhancement. *arXiv preprint arXiv:1804.04121* (2018).

[15] Muneeb Imtiaz Ahmad, Abdullah Alzahrani, and Sunbul M Ahmad. 2024. Detecting deception in natural environments using incremental transfer learning. In *Proceedings of the 26th International Conference on Multimodal Interaction*. 66–75.

[16] Haoyu Bian, Bin Guo, Sicong Liu, Yasan Ding, Shanshan Gao, and Zhiwen Yu. 2024. UbiHR: Resource-efficient Long-range Heart Rate Sensing on Ubiquitous Devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 8, 4 (2024), 1–26.

[17] Yetong Cao, Chao Cai, Fan Li, Zhe Chen, and Jun Luo. 2023. HeartPrint: Passive heart sounds authentication exploiting in-ear microphones. In *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*. IEEE, 1–10.

[18] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. 2019. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv preprint arXiv:1906.07155* (2019).

[19] Tao Chen, Yongjie Yang, Xiaoran Fan, Xiuzhen Guo, Jie Xiong, and Longfei Shangguan. 2024. Exploring the Feasibility of Remote Cardiac Auscultation Using Earphones. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*. 357–372.

[20] Weixuan Chen and Daniel McDuff. 2018. Deepphys: Video-based physiological measurement using convolutional attention networks. In *Proceedings of the european conference on computer vision (ECCV)*. 349–365.

[21] Zhe Chen, Tianyue Zheng, Chao Cai, and Jun Luo. 2021. MoVi-Fi: Motion-robust vital signs waveform recovery via deep interpreted RF sensing. In *Proceedings of the 27th annual international conference on mobile computing and networking*. 392–405.

[22] Gerard De Haan and Vincent Jeanne. 2013. Robust pulse rate from chrominance-based rPPG. *IEEE transactions on biomedical engineering* 60, 10 (2013), 2878–2886.

[23] Ilke Demir and Umur Aybars Ciftci. 2021. Where do deep fakes look? synthetic face detection via gaze tracking. In *ACM symposium on eye tracking research and applications*. 1–11.

[24] Sandesh Dhawaskar Sathyanarayana, Kyunghan Lee, Dirk Grunwald, and Sangtae Ha. 2023. Converge: Qoe-driven multipath video conferencing over webrtc. In *Proceedings of the ACM SIGCOMM 2023 Conference*. 637–653.

[25] Kuntai Du, Ahsan Pervaiz, Xin Yuan, Aakanksha Chowdhery, Qizheng Zhang, Henry Hoffmann, and Junchen Jiang. 2020. Server-driven video streaming for deep learning inference. In *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*. 557–570.

[26] Stephen H Fairclough and Chelsea Dobbins. 2020. Personal informatics and negative emotions during commuter driving: Effects of data visualization on cardiovascular reactivity & mood. *International Journal of Human-Computer Studies* 144 (2020), 102499.

[27] Xiaoran Fan, David Pearl, Richard Howard, Longfei Shangguan, and Trausti Thormundsson. 2023. Apg: Audioplethysmography for cardiac monitoring in hearables. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*. 1–15.

[28] Yongjie Guan, Xueyu Hou, Nan Wu, Bo Han, and Tao Han. 2023. Metastream: Live volumetric content capture, creation, delivery, and rendering in real time. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*. 1–15.

[29] Pouya Hamadanian, Doug Gallatin, Mohammad Alizadeh, and Krishna Chintalapudi. 2023. Ekho: Synchronizing cloud gaming media across multiple endpoints. In *Proceedings of the ACM SIGCOMM 2023 Conference*. 533–549.

[30] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7132–7141.

[31] Yidi Jiang, Ruijie Tao, Zexu Pan, and Haizhou Li. 2023. Target Active Speaker Detection with Audio-visual Cues. In *Proc. Interspeech*.

[32] Mehrdad Khani, Ganesh Ananthanarayanan, Kevin Hsieh, Junchen Jiang, Ravi Netravali, Yuanchao Shu, Mohammad Alizadeh, and Victor Bahl. 2023. {RECL}: Responsive {Resource-Efficient} continuous learning for video analytics. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. 917–932.

[33] Jianwei Li, Zitong Yu, and Jingang Shi. 2023. Learning motion-robust remote photoplethysmography through arbitrary resolution videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 1334–1342.

[34] Xiaobai Li, Jie Chen, Guoying Zhao, and Matti Pietikainen. 2014. Remote heart rate measurement from face videos under realistic situations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4264–4271.

[35] Zhuqi Li, Yaxiong Xie, Ravi Netravali, and Kyle Jamieson. 2023. Dashlet: Taming swipe uncertainty for robust short video streaming. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. 1583–1599.

[36] David Lion, Adrian Chiu, Hailong Sun, Xin Zhuang, Nikola Grcevski, and Ding Yuan. 2016. {Don't} Get Caught in the Cold, Warm-up Your {JVM}: Understand and Eliminate {JVM} Warm-up Overhead in {Data-Parallel} Systems. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. 383–400.

[37] Jian Liu, Cong Shi, Yingying Chen, Hongbo Liu, and Marco Gruteser. 2019. Cardiocam: Leveraging camera on mobile devices to verify users while their heart is pumping. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*. 249–261.

[38] Jian Liu, Yan Wang, Yingying Chen, Jie Yang, Xu Chen, and Jerry Cheng. 2015. Tracking vital signs during sleep leveraging off-the-shelf wifi. In *Proceedings of the 16th ACM international symposium on mobile ad hoc networking and computing*. 267–276.

[39] Xin Liu, Josh Fromm, Shwetak Patel, and Daniel McDuff. 2020. Multi-task temporal shift attention networks for on-device contactless vitals measurement. *Advances in Neural Information Processing Systems* 33 (2020), 19400–19411.

[40] Xin Liu, Brian Hill, Ziheng Jiang, Shwetak Patel, and Daniel McDuff. 2023. Efficientphys: Enabling simple, fast and accurate camera-based cardiac measurement. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 5008–5017.

[41] Yunzhuo Liu, Bo Jiang, Tian Guo, Ramesh K Sitaraman, Don Towsley, and Xinbing Wang. 2020. Grad: Learning for overhead-aware adaptive video streaming with scalable video coding. In *Proceedings of the 28th ACM International Conference on Multimedia*. 349–357.

[42] Fuchen Long, Zhaofan Qiu, Yingwei Pan, Ting Yao, Chong-Wah Ngo, and Tao Mei. 2022. Dynamic temporal filtering in video models. In *European Conference on Computer Vision*. Springer, 475–492.

[43] Aske Mottelson and Kasper Hornbæk. 2016. An affect detection technique using mobile commodity sensors in the wild. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 781–792.

[44] Xuesong Niu, Zitong Yu, Hu Han, Xiaobai Li, Shiguang Shan, and Guoying Zhao. 2020. Video-based remote physiological measurement via cross-verified feature disentangling. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 295–310.

[45] M Prajwal, Ayush Raj, Sougata Sen, Snehanshu Saha, and Surjya Ghosh. 2023. Towards Efficient Emotion Self-report Collection Using Human-AI Collaboration: A Case Study on Smartphone Keyboard Interaction. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 7, 2 (2023), 1–23.

[46] Hua Qi, Qing Guo, Felix Juefei-Xu, Xiaofei Xie, Lei Ma, Wei Feng, Yang Liu, and Jianjun Zhao. 2020. Deeprhythm: Exposing deepfakes with attentional visual heartbeat rhythms. In *Proceedings of the 28th ACM international conference on multimedia*. 4318–4327.

[47] Kun Qian, Chenshu Wu, Fu Xiao, Yue Zheng, Yi Zhang, Zheng Yang, and Yunhao Liu. 2018. Acousticcardiogram: Monitoring heartbeats using acoustic signals on smart devices. In *IEEE INFOCOM 2018-IEEE conference on computer communications*. IEEE, 1574–1582.

[48] Mirco Ravanelli and Yoshua Bengio. 2018. Speaker recognition from raw waveform with sincnet. In *2018 IEEE spoken language technology workshop (SLT)*. IEEE, 1021–1028.

[49] T-YLPG Ross and GKHP Dollár. 2017. Focal loss for dense object detection. In *proceedings of the IEEE conference on computer vision and pattern recognition*. 2980–2988.

[50] Steven A Shafer. 1985. Using color to separate reflection components. *Color Research & Application* 10, 4 (1985), 210–218.

[51] Ronny Stricker, Steffen Müller, and Horst-Michael Gross. 2014. Non-contact video-based pulse rate measurement on a mobile service robot. In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, 1056–1062.

[52] Ke Sun and Xinyu Zhang. 2021. UltraSE: single-channel speech enhancement using ultrasound. In *Proceedings of the 27th annual international conference on mobile computing and networking*. 160–173.

[53] Zhaodong Sun, Alexander Vedernikov, Virpi-Liisa Kykyri, Mikko Pohjola, Miriam Nokia, and Xiaobai Li. 2022. Estimating stress in online meetings by remote physiological signal and behavioral features. In *Adjunct Proceedings of the 2022 ACM International Joint Conference on Pervasive and Ubiquitous Computing and the 2022 ACM International Symposium on Wearable Computers*. 216–220.

[54] Jiankai Tang, Kequan Chen, Yuntao Wang, Yuanchun Shi, Shwetak Patel, Daniel McDuff, and Xin Liu. 2023. MMPD: Multi-Domain Mobile Video Physiology Dataset. In *2023 45th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. 1–5. https://doi.org/10.1109/EMBC40787.2023.10340857

[55] Xiaohu Tang, Yang Wang, Ting Cao, Li Lyna Zhang, Qi Chen, Deng Cai, Yunxin Liu, and Mao Yang. 2023. Lut-nn: Empower efficient neural network inference with centroid learning and table lookup. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*. 1–15.

[56] Ruijie Tao, Zexu Pan, Rohan Kumar Das, Xinyuan Qian, Mike Zheng Shou, and Haizhou Li. 2021. Is Someone Speaking? Exploring Long-term Temporal Features for Audio-visual Active Speaker Detection. In *Proceedings of the 29th ACM International Conference on Multimedia*. 3927–3935.

[57] Dan Wang, Haibo Lei, Haozhi Dong, Yunshu Wang, Yongpan Zou, and Kaishun Wu. 2020. What you wear know how you feel: An emotion inference system with multi-modal wearable devices. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. 1–3.

[58] Lei Wang, Tao Gu, Wei Li, Haipeng Dai, Yong Zhang, Dongxiao Yu, Chenren Xu, and Daqing Zhang. 2023. Df-sense: Multi-user acoustic

sensing for heartbeat monitoring with dualforming. In *Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services*. 1–13.

[59] Lei Wang, Wei Li, Ke Sun, Fusang Zhang, Tao Gu, Chenren Xu, and Daqing Zhang. 2022. LoEar: Push the range limit of acoustic sensing for vital sign monitoring. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 3 (2022), 1–24.

[60] Limin Wang, Zhan Tong, Bin Ji, and Gangshan Wu. 2021. Tdn: Temporal difference networks for efficient action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 1895–1904.

[61] Wenjin Wang, Albertus C Den Brinker, Sander Stuijk, and Gerard De Haan. 2016. Algorithmic principles of remote PPG. *IEEE Transactions on Biomedical Engineering* 64, 7 (2016), 1479–1491.

[62] Wenjin Wang, Sander Stuijk, and Gerard De Haan. 2015. A novel algorithm for remote photoplethysmography: Spatial subspace rotation. *IEEE transactions on biomedical engineering* 63, 9 (2015), 1974–1984.

[63] Claudia AF Wascher. 2021. Heart rate as a measure of emotional arousal in evolutionary biology. *Philosophical Transactions of the Royal Society B* 376, 1831 (2021), 20200479.

[64] Hao Wen, Yuanchun Li, Zunshuai Zhang, Shiqi Jiang, Xiaozhou Ye, Ye Ouyang, Yaqin Zhang, and Yunxin Liu. 2023. Adaptivenet: Post-deployment neural architecture adaptation for diverse edge environments. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*. 1–17.

[65] John Wiseman and Ivan Yu Bondarenko. 2016. Python interface to the webrtc voice activity detector. *Python interface to the WebRTC voice activity detector* (2016).

[66] Hao Wu, Jinghao Feng, Xuejin Tian, Edward Sun, Yunxin Liu, Bo Dong, Fengyuan Xu, and Sheng Zhong. 2020. EMO: Real-time emotion recognition from single-eye images for resource-constrained eyewear devices. In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*. 448–461.

[67] Chenhan Xu, Tianyu Chen, Huining Li, Alexander Gherardi, Michelle Weng, Zhengxiong Li, and Wenyao Xu. 2022. Hearing heartbeat from voice: Towards next generation voice-user interfaces with cardiac sensing functions. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*. 149–163.

[68] Kangning Yang, Benjamin Tag, Chaofan Wang, Yue Gu, Zhanna Sarsenbayeva, Tilman Dingler, Greg Wadley, and Jorge Goncalves. 2022. Survey on emotion sensing using mobile devices. *IEEE Transactions on Affective Computing* 14, 4 (2022), 2678–2696.

[69] Wenyuan Yang, Xiaoyu Zhou, Zhikai Chen, Bofei Guo, Zhongjie Ba, Zhihua Xia, Xiaochun Cao, and Kui Ren. 2023. Avoid-df: Audio-visual joint learning for detecting deepfake. *IEEE Transactions on Information Forensics and Security* 18 (2023), 2015–2029.

[70] Zhicheng Yang, Parth H Pathak, Yunze Zeng, Xixi Liran, and Prasant Mohapatra. 2016. Monitoring vital signs using millimeter wave. In *Proceedings of the 17th ACM international symposium on mobile ad hoc networking and computing*. 211–220.

[71] Rongjie Yi, Ting Cao, Ao Zhou, Xiao Ma, Shangguang Wang, and Mengwei Xu. 2023. Boosting dnn cold inference on edge devices. In *Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services*. 516–529.

[72] Zitong Yu, Xiaobai Li, and Guoying Zhao. 2019. Remote photoplethysmograph signal measurement from facial videos using spatio-temporal networks. *arXiv preprint arXiv:1905.02419* (2019).

[73] Zitong Yu, Wei Peng, Xiaobai Li, Xiaopeng Hong, and Guoying Zhao. 2019. Remote heart rate measurement from highly compressed facial videos: an end-to-end deep learning solution with video enhancement. In *Proceedings of the IEEE/CVF international conference on computer vision*. 151–160.

[74] Zitong Yu, Yuming Shen, Jingang Shi, Hengshuang Zhao, Yawen Cui, Jiehua Zhang, Philip Torr, and Guoying Zhao. 2023. Physformer++: Facial video-based physiological measurement with slowfast temporal difference transformer. *International Journal of Computer Vision* 131, 6 (2023), 1307–1330.

[75] Fusang Zhang, Zhi Wang, Beihong Jin, Jie Xiong, and Daqing Zhang. 2020. Your Smart Speaker Can" Hear" Your Heartbeat! *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 4 (2020), 1–24.

[76] Anfu Zhou, Huanhuan Zhang, Guangyuan Su, Leilei Wu, Ruoxuan Ma, Zhen Meng, Xinyu Zhang, Xiufeng Xie, Huadong Ma, and Xiaojiang Chen. 2019. Learning to coordinate video codec with transport protocol for mobile video telephony. In *The 25th Annual International Conference on Mobile Computing and Networking*. 1–16.

[77] Bochao Zou, Zizheng Guo, Jiansheng Chen, and Huimin Ma. 2024. Rhythmformer: Extracting rppg signals based on hierarchical temporal periodic transformer. *arXiv preprint arXiv:2402.12788* (2024).