



Article Super-Resolution-Empowered Adaptive Medical Video Streaming in Telemedicine Systems

Hangcheng Han * and Jian Lv

School of Integrated Circuits and Electronics, Beijing Institute of Technology, Beijing 100081, China * Correspondence: hanhangcheng@bit.edu.cn

Abstract: Due to influence of COVID-19, telemedicine is becoming more and more important. Highquality medical videos can provide a physician with a better visual experience and increase the accuracy of disease diagnosis, but this requires a dramatic increase in bandwidth compared to that required by regular videos. Existing adaptive video-streaming approaches cannot successfully provide high-resolution video-streaming services under poor or fluctuating network conditions with limited bandwidth. In this paper, we propose a super-resolution-empowered adaptive medical video streaming in telemedicine system (named SR-Telemedicine) to provide high quality of experience (QoE) videos for the physician while saving the network bandwidth. In SR-Telemedicine, very low-resolution video chunks are first transmitted from the patient to an edge computing node near the physician. Then, a video super-resolution (VSR) model is employed at the edge to reconstruct the low-resolution video chunks into high-resolution ones with an appropriate high-resolution level (such as 720p or 1080p). Furthermore, the neural network of VSR model is designed to be scalable and can be determined dynamically. Based on the time-varying computational capability of the edge computing node and the network condition, a double deep Q-Network (DDQN)-based algorithm is proposed to jointly select the optimal reconstructed high-resolution level and the scale of the VSR model. Finally, extensive experiments based on real-world traces are carried out, and the experimental results illustrate that the proposed SR-Telemedicine system can improve the QoE of medical videos by 17–79% compared to three baseline algorithms.

Keywords: medical video streaming; telemedicine system; video super resolution; deep reinforcement learning

1. Introduction

With the development of eligible consumer devices, broadband access mobile networks, as well as emerging video technologies such as high-definition (HD) and ultrahighdefinition (UHD) videos and virtual reality (VR), medical videos have been widely applied in telemedicine or remote healthcare systems. In particular, under the influence of COVID-19, telemedicine is becoming more and more important. In telemedicine systems, real-time streaming video of the patients can be transmitted to a central hospital or a physician expert for diagnosis of diseases (e.g., neurological eye examinations [1], heart disease diagnosis [2–4], and kidney disease care [5]). However, on the one hand, low video quality (e.g., caused by low video resolution: 240p, 360p, 480p) will severely degrade the accuracy of the diagnosis of the disease. On the other hand, HD/UHD videos (e.g., 4k, 8k) will bring huge data volume and higher overall bit rate, which imposes great transmission pressure on the networks.

It is challenging to provide HD/UHD medical video services with low network overhead or under poor or fluctuating network conditions. The traditional dynamic adaptive streaming scheme (e.g., DASH [6], LEAN [7]) highly depends on network conditions and can only offer low-resolution videos when the network condition is poor. Video superresolution (VSR) [8] is a promising technique for breaking this dependence, because it



Citation: Han, H.; Lv, J. Super-Resolution-Empowered Adaptive Medical Video Streaming in Telemedicine Systems. *Electronics* 2022, 11, 2944. https://doi.org/ 10.3390/electronics11182944

Academic Editor: Stefanos Kollias

Received: 1 August 2022 Accepted: 13 September 2022 Published: 16 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). can reconstruct low-resolution videos as high-resolution videos near the end-user side. However, VSR requires powerful processing capability, whereas most of the existing VSR models [8] cannot achieve line-speed VSR processing on mobile devices. An alternative method is to reconstruct the low-resolution videos at an edge node. In this case, the patient only needs to output the video streaming with lower resolution, and the bandwidth from the patient to the edge node near the physician will be largely saved. Nonetheless, the time complexity of the VSR model should dynamically match the available computational capability of the edge node, which is time-varying because it is shared by diverse applications. At the same time, the bit rates of the reconstructed high-resolution videos should match the dynamic network conditions. Otherwise, additional processing delay or transmission delay will be introduced.

In order to break the strong dependence of video transmission quality on network bandwidth, many works have investigated the enhancement of HD/UHD video transmission via VSR. NAS [8] and SRAVS [9] introduce a new DASH-based video-delivery framework that promotes the video quality by employing VSR on the client device. Nonetheless, these two approaches require the client to have powerful computational capability (e.g., consumer GPUs), which is not practical for mobile devices. To address this problem, VISCA [10] presents an edge-assisted adaptive video-streaming solution, which enhances the low-quality video chunks by VSR at the edge. Although this approach can provide high-quality videos even when the network resources are limited, it does not consider the dynamic state of the available computational capability on the edge node. An additional super-resolution processing delay will be introduced when the available computational capability cannot satisfy the requirement of VSR, which will lead to more rebuffering and degrade the users' quality of experience (QoE). Therefore, in addition to selecting an optimal reconstructed high-resolution level, the model of VSR also needs to be well designed to adapt to the available computational capability of the edge node. In addition, in the field of telemedicine or healthcare, a super-resolution (SR) technique has already been applied to promote the quality of medical images or videos [11–19], and many efforts also have been devoted to designing adaptive medical video-streaming strategies [7,20–24], but there is still a lack of research on how to employ VSR to enhance the medical video transmission.

In this paper, we propose a super-resolution-empowered adaptive medical video streaming in telemedicine (named SR-Telemedicine) system to improve the physician's QoE while saving the network bandwidth. The SR-Telemedicine system can reconstruct the low-resolution video chunks into high-resolution ones with several different high-resolution levels (such as 720p, 1080p, 4k) at the edge node near the physician. At the same time, in order to match the time-varying computational capability of the edge node and the network condition, the VSR neural network model is designed to be scalable. The scalability of the model size is implemented by introducing the early exit mechanism [25] (i.e., the number of ResBlocks used for inference is optional). Finally, to jointly select the optimal reconstructed high-resolution level and VSR model scale, a deep reinforcement learning (DRL) variant algorithm—the double deep Q-Network (DDQN) [26]—is employed with the aim of maximizing QoE. Our main contributions are summarized as follows.

- We design the SR-Telemedicine scheme to improve physician's QoE while saving the network bandwidth. In SR-Telemedicine, VSR is used to reconstruct the low-resolution video chunks into high-resolution ones with an appropriate high-resolution level at the edge node.
- To decrease the VSR processing delay, the scale of the VSR neural network is designed to be scalable and can be determined dynamically according to the computational capability of the edge node and the network condition.
- To successfully catch the dynamic states of the SR-Telemedicine system and the network, a DRL-based adaptation algorithm is proposed to jointly optimize the reconstructed high-resolution level and the scale of VSR neural network.

Extensive experiments based on real-world network traces are conducted, and the

17–79% compared to three baseline algorithms. The remainder of the paper is structured as follows. Related literature is reviewed in Section 2. Section 3 presents the system models and problem formulation. In Section 4, the DRL-based joint optimization algorithm is proposed to improve the QoE of our proposed SR-Telemedicine system. The evaluation results are detailed within Section 5. Some

results illustrate that our proposed SR-Telemedicine system can improve QoE by

2. Related Work

In this section, the relevant literature about adaptive video streaming for healthcare and SR for healthcare are reviewed.

remaining issues are discussed in Section 6. Finally, Section 7 concludes the paper.

2.1. Adaptive Video Streaming for Healthcare

Dynamic adaptive streaming over HTTP (DASH) [6] has emerged as a key technology by which to enhance bandwidth utilization for video delivery. In the DASH system, videos are encoded into multiple chunks along with several bit rate levels, and chunks across bit rates are aligned to support seamless quality transitions. The adaptive bit rate (ABR) algorithm is an essential part of the DASH system, which integrates the client side in order to dynamically select the most appropriate bit rate for the current state. The solution for the ABR algorithm can be roughly divided into four categories: (1) rate-based algorithms (e.g., RB [27]) select the highest possible bit rate based on the estimated available throughput; (2) buffer-based algorithms (e.g., BOLA [28]), make bit rate decisions according to the buffer occupancy level of video players; (3) rate and buffer joint algorithms (e.g., MPCbased algorithm [29]), select the most appropriate bit rate by solving a QoE minimization problem; and (4) learning-based algorithms (e.g., Pensieve [30]) train a neural network model that selects bit rates for future video chunks based on observations collected by client players. The rate-based and buffer-based algorithms are simple and heuristic, but find it difficult to respond to the time-varying changes of the network in a timely manner. The rate and buffer joint algorithms can perform better than approaches that use fixed thresholds. However, the MPC's performance relies on an accurate model of the system dynamics—in particular, a forecast of future network throughput. The learning-based algorithms can learn a control policy for bit rate adaptation purely through experience without using any pre-programmed control rules or explicit assumptions about the operating environment. However, the native DRL algorithm (e.g., DQN) directly calculates the target Q value through the greedy algorithm. Although the Q value can quickly be closer to the possible optimization target in this way, the target Q value will be overestimated, and the deviation will be large.

With the advent of the Internet of Things (IoT) and 5G/6G communication technologies, medical video streaming is widely applied from remote patients to centralized hospitals for better diagnosis and treatment. Lavanya et al. [7] propose a new hybrid predictive model, long effective adaptive networks (LEAN), an efficient medical video transmission, which works on the principle of integrating the time-predictive long shortterm memory (LSTM) with boosting machine learning algorithms. Usman et al. [20] present a complete framework for a 5G-enabled connected ambulance that focuses on two-way data communication including audio-visual multimedia flow between ambulances and hospitals. Rajavel et al. [21] propose an IoT-based smart healthcare video surveillance system by using edge computing to reduce the network bandwidth and response time and significantly maximize the fall behavior prediction accuracy. In order to achieve realtime adaptation to time-varying constraints for medical video communications, Antoniou et al. [22] propose an adaptive video-encoding framework based on multi-objective optimization that jointly maximizes the encoded video's quality and encoding rate (in frames per second) while minimizing bit rate demands. Ghimire et al. [23] propose a system that consists of an enhanced video quality and distortion minimization (EVQDM) algorithm to

achieve guaranteed quality, minimum distortion, and the minimum delay in telemedicine real-time video transmission. This system guarantees the video quality by using the adaptive video-encoding technique and minimizes the distortion by considering the truncating distortion in the enhanced distortion-minimization algorithm. Delhaye et al. [24] address the challenges of video compression for telemedicine applications in a very low bandwidth environment, such as air-to-ground transmission channels. Their method compresses the region of interest (ROI) with higher quality by using, for a specific available bit rate, different quantizers for blocs inside and outside the ROI.

Most of the existing approaches highly depend on network bandwidth conditions and can only offer low-resolution videos when the network condition is poor. In this paper, we introduce the VSR technique into medical video transmission, which requires that only very low-resolution videos need to be transmitted from the patient to an edge node near the physician.

2.2. Super-Resolution for Healthcare

The SR technique is able to reconstruct low-resolution images as high-resolution images or video frames, and modern SR techniques using deep learning methods significantly improve the reconstruction quality. NAS [8] and SRAVS [9] integrate SR into adaptive video streaming at the client side to provide high-quality videos while minimizing the impact of dynamic network conditions on users' QoE. NAS is proposed to improve the video quality by modified MDSR for devices with strong capabilities. Considering the limited computing capacity of users' devices, SRAVS adapts SRCNN, a simple, lightweight SR model with only three convolutional layers. Beyond the aforementioned systems, an edge platform can provide more powerful computing capacity to reduce SR elapsed time while supporting a consistent, high-quality server for multiple clients. VISCA [10] produces an edge-assisted, adaptive video-streaming solution that integrates SR with edge caching to improve QoE. However, it is not aware of the available computing capacity at the edge.

The SR technique also has been applied to enhance the medical images or videos. Zhang et al. [11] present a fast medical image super-resolution (FMISR) method whereby the three hidden layers complete feature extraction in the same way as the SR convolution neural network. In [12,13], an efficient medical video SR method based on deep backprojection networks is investigated. Shi et al. [14] use local residual block and global residual network to extend SRCNN to solve a 2D magnetic resonance imaging (MRI) SR problem. Farias et al. [15] propose a generative adversarial network (GAN)-based lesion-focused framework for computed tomography (CT) image SR. For the lesion (i.e., cancer) patchfocused training, spatial pyramid pooling (SPP) is incorporated into GAN-constrained by the identical, residual, and cycle learning ensemble (GAN-CIRCLE). Qiu et al. [16] propose the multiple improved residual network (MIRN) SR reconstruction method. MIRN designs the residual blocks connected by multi-level skips to build multiple improved residual block (MIRB) modules. Zhao et al. [17] propose an information distillation and multi-scale attention network (IDMAN) for medical CT image SR reconstruction. Jebadurai et al. [18] propose a hybrid architecture for IoT healthcare to process the retinal images captured by using smartphone fundoscopy. The proposed SR algorithm for retinal images uses multi-kernel support vector regression (SVR) to improve the quality of the captured images. Deeba et al. [19] propose a wavelet-based mini-grid network medical image super-resolution (WMSR) method, which is similar to the three-layer hidden-layer-based SRCNN method.

The above approaches with SR only use a fixed VSR model to reconstruct video blocks from low resolution to a certain high resolution. In this paper, we will jointly optimize the reconstructed high-resolution level and the model scale of VSR to match the time-varying computational capability of the edge node and the network condition.

2.3. Edge Computing for Healthcare

VSR can reconstruct low-resolution videos to high-level resolution videos but it requires powerful processing capability. Most of the existing VSR models cannot achieve a line-speed VSR processing on the mobile devices. An alternative method is to reconstruct the low-resolution videos at an edge node. In recent years, edge computing has been widely used to drive healthcare advances due to the following advantages. (1) Edge computing provides a robust infrastructure to keep continuous processing without disruption, even during network outages. (2) Processing data at the edge provides near-instantaneous feedback. (3) Keeping data within the device and inference at the edge means that patient health information stays secure. (4) AI processing at the edge reduces the need to send high-bandwidth data. (5) Domain experts are able to create a highly adaptive and outcome-focused VSR solution by controlling the data processing AI parameters at the edge.

Many efforts have been devoted to investigating edge computing-assisted telemedicine [21,31–34]. Dong et al. [31] designed an edge computing-based healthcare system to schedule transmission and computation resources for cost-efficient monitoring in the Internet of medical things (IoMT). Wang et al. [32] proposed a multi-layer 5G mobile edge computing (MEC)-centered telemedicine design that dynamically integrates wearable devices with an OpenEMR electronic health records system. Prabhu et al. [33] presented a healthcare framework that incorporates a promising edge–IoT ecosystem—the EdgeX Foundry—for the telehealth use case of blood pressure monitoring. Oueida et al. [34] proposed a resource preservation net (RPN) framework by using Petri net, integrated with custom cloud and edge computing suitable for emergency department (ED) systems. RPN is applicable to a real-life scenario in which key performance indicators, such as patient length of stay (LoS), resource utilization rate, and average patient waiting time are modeled and optimized. In order to make a robust video surveillance system, Rajavel et al. [21] proposed a cloud-based object tracking and behavior identification system (COTBIS) that can incorporate the edge computing capability framework at the gateway level.

Most of the existing approaches only leverage edge computing to analyze the patients' health data and assist doctors in diagnosis. In this paper, we mainly employ edge computing to conduct VSR, and the enhanced patients' video streaming can make the physician more clearly observe small lesions and increase the accuracy of the diagnosis.

3. System Model and Problem Formulation

In this section, we first introduce the reference architecture of VSR-enabled telemedicine systems. Then, the system model is described. Finally, we present the problem formulation.

3.1. System Model

The reference architecture of VSR-enabled telemedicine systems is illustrated in Figure 1, which mainly consists of three parts: patient video monitoring, VSR-enabled edge computing, and physician video playing.

The patient video-monitoring part can monitor the real-time status of the patient via video cameras. The captured video chunks can be encoded into different levels of resolutions, denoted as $\mathcal{K} = \{1, 2, ..., K\}$. Let $q_i \in \mathcal{K}$ represent the transmitted resolution of the *i*-th chunks, and $b(i, q_i)$ denote the actual size of this chunk. Each chunk has the same time duration, denoted by *T*. Then, video chunks will be streamed and transmitted to the edge computing node near the physician via Internet. The VSR-enabled edge computing node is comprised of three modules: VSR processing buffer, VSR processor, streaming proxy, and DRL agent. The VSR processing buffer is employed to cache the video chunks transmitted from the patient, the VSR processor is used to promote the resolution of the video chunks (e.g., from 480p to 1080p), the DRL agent will collect the states of the telemedicine system and determine the optimal resolutions of the source video chunks (on the patient's side) and the promoted video chunks (at the edge node near the physician), and the streaming proxy is used to restream the promoted high-definition video chunks



and transmit them to the physician's side. Finally, the physician video-playing part will play the streaming video to the physician for disease diagnosis.

Figure 1. The reference architecture of VSR-enabled telemedicine systems—SR-Telemedicine, which mainly consists of three parts: patient video monitoring, VSR-enabled edge computing, and physician video playing.

3.1.1. Multi-Scale Neural Network for Video Super-Resolution

The low-resolution video chunks transmitted from the patient will be cached into the VSR processing buffer B_V at the edge node. By following the principle of first-in-first-out (FIFO), the VSR processor will sequentially reconstruct the cached chunks and output the high-resolution chunks. For each low-resolution video chunk *i* in the VSR processing buffer, a DRL agent will make a reconstruction decision (q_i, q'_i) for the VSR processor. Then, the VSR processor will promote the resolution of the video chunk *i* from the low resolution q_i into the high resolution q'_i ($q'_i \ge q_i$). Let $\mathcal{D} = \{(q_i, q'_i) | \forall q_i, q'_i \in \mathcal{K}, q'_i \ge q_i\}$ denote the all possible reconstruction pairs.

For a certain amount of computing power, the VSR processing time will partially depend on the scales of the VSR model. A more complex VSR model will cause a longer processing time and increase the end-to-end delay, but can gain a better quality of the reconstructed video chunks. Inspired by NAS [8], VSR pre-configures a network model for each possible $(q_i, q'_i) \in D$. At the same time, in order to achieve a good tradeoff between the processing time and the video quality, we propose a multi-scale mechanism for each VSR model which consists of total $|\mathcal{L}|$ ResBlocks [25]. The scalability of the VSR model can be implemented by regulating the output path of the early exit mechanism [25] (i.e., the number of ResBlocks used for inference). For example, the scale can be set as low, medium, high, and ultrahigh with different number of ResBlocks in the neural network as shown in Figure 2. However, it is worth noting that even a small number of ResBlocks are used to reconstruct low-resolution videos, and QoE can still be improved compared to watching the original low-resolution videos.



Figure 2. Multi-scale video SR models. The scalability of the SVSR model size can be achieved by the output path of the early exit mechanism [25] (i.e., the number of ResNet blocks used for inference).

3.1.2. Available Edge Computing Capability

In general, VSR is a computation-intensive task. Due to sharing with diverse applications, the available computational capability at the edge node is time-varying, and thus the VSR reconstruction time is also not fixed. In order to select an optimal VSR model to reconstruct the resolution of video chunk *i* from q_i to q'_i , we need to estimate the available computational capability at the edge node. In this paper, we will use the historical VSR reconstruction time to estimate the VSR capability of the edge node.

For the last δ chunks, the reconstructed resolution pairs can be denoted as

$$\mathbf{q} = \langle (q_{i-1}, q'_{i-1}), (q_{i-2}, q'_{i-2}), \cdots, (q_{i-\delta}, q'_{i-\delta}) \rangle .$$
(1)

Let $h(q_i, q'_i)$ denote the reconstruction time of chunk *i*. The reconstruction time of the last δ chunks can be represented as a vector:

$$\mathbf{h} = < h(q_{i-1}, q'_{i-1}), h(q_{i-2}, q'_{i-2}), \cdots, h(q_{i-\delta}, q'_{i-\delta}) > .$$
⁽²⁾

Then, we can use **h** to implicitly measure the current available computational capability of the edge node.

3.1.3. VSR Processing Model

Assume that the maximum length of the VSR processing buffer B_V is B_V^{max} , and t_i^{SR} denote the time of starting to reconstruct the *i*-th chunk. t_i^{SR} should satisfy

$$t_i^{VSR} \triangleq \max\{t_{i-1}^{VSR} + h(q_{i-1}, q_{i-1}'), t_i^V\},\tag{3}$$

where $h(q_{i-1}, q'_{i-1})$ denotes the time for reconstructing the resolution of the (i - 1)-th chunk from q_{i-1} to q'_{i-1} , and t^V_i indicates the time when the edge node finishes receiving the *i*-th video chunk. The evolution of VSR processing buffer can be expressed as

$$B_V(t) = \max\{j|t_j^V \le t\} - \max\{j|t_j^{VSR} + h(q_j, q_j') \le t\}.$$
(4)

3.1.4. Video Playback Model

Let B_P^{max} denote the max length of the playback buffer B_P . The *i*-th chunk will be transmitted from the edge node, and after that (1) the (i - 1)-th chunk has already been transmitted into the playback buffer, and (2) the *i*-th chunk has already been reconstructed at the edge node. Therefore, t_i^D , the time when the physician side starts to download the *i*-th chunk, should satisfy

$$t_{i}^{D} \triangleq \max\{t_{i-1}^{D} + d_{i-1}, t_{i}^{VSR} + d_{i}, t_{i-B_{p}^{max}}^{play}\},$$
(5)

where d_{i-1} indicates the download time of the (i - 1)-th chunk from the edge node to the physician, and t_j^{play} (here $j = i - B_P^{max}$) denotes the time when the physician side begins to play the *j*-th chunk.

On the physician's side, the video chunks will be decoded and rendered. The *i*-th chunk can be played only when it has already been downloaded in the playback buffer and the (i - 1)-th chunk had already been played completely. Thus, the playing time t_i^{play} of the *i*-th chunk can be calculated as

$$t_i^{play} = \max\{t_{i-1}^{play} + T, t_i^D + d_i\}.$$
(6)

Then, the evolution of playback buffer can be expressed as

$$B_P(t) = \max\{j \mid t_i^D + d_j \le t\} - \max\{j \mid t_i^{play} \le t\}.$$
(7)

In particular, when the playback buffer is exhausted, the rebuffering process will happen, which results in an extreme degradation of the user's QoE. The rebuffering time when playing video chunk *i* can be depicted as τ_i :

$$\tau_i = t_i^{play} - (t_{i-1}^{play} + T).$$
(8)

3.1.5. Transmission Model

In our proposed SR-Telemedicine system, there exist two transmission stages: (1) from the patient to the edge node, and (2) from the edge node to the physician. For the first state, we suppose that the patient side begins to transmit the *i*-th video chunk at the time t_i^S whereas the edge node finishes receiving the *i*-th video chunk at time t_i^V , and then the transmission rate for the *i*-th video chunk can be expressed as $c_i^1 = b(i, q_i) / (t_i^V - t_i^S)$. For the last δ video chunks, we have

$$\mathbf{c_1} = < c_{i-1}^1, c_{i-2}^1, \dots, c_{i-\delta}^1 > .$$
(9)

Similarly, for the second stage, we assume that the edge node begins to transmit the *i*-th video chunk at the time t_i^D whereas the physician finishes receiving the *i*-th video chunk at time t_i^P . Then the transmission rate for the *i*-th video chunk can be expressed as $c_i^2 = b(i, q_i')/(t_i^D - t_i^P)$. For the last δ video chunks, we have

$$\mathbf{c_2} = < c_{i-1}^2, c_{i-2}^2, \dots, c_{i-\delta}^2 > .$$
(10)

3.2. Problem Formulation

In order to improve the users' QoE, the DRL agent needs to jointly determine the reconstructed high-resolution level $(q_i, q'_i) \in \mathcal{D}$ for each chunk *i*, and the number of ResBlocks $l_i \in \mathcal{L}$ utilized in the VSR neural network model. The objective is to maximize the users' QoE, which consists of three aspects: video bit rate, rebuffering time, and video-quality jitter.

3.2.1. Video Bit Rate

In this paper, the average video bit rate is employed to evaluate the video quality, which is also used in Pensieve [30],

$$Q_{quality} = \frac{1}{N_{chunk}} \sum_{i=1}^{N_{chunk}} R'_i, \tag{11}$$

where R'_i represents the bit rate of the reconstructed video chunk *i*, and N_{chunk} is the total number of video chunks during the system runtime.

3.2.2. Rebuffering Time

Rebuffering time occurs when the playback buffer becomes empty. The video playing will be interrupted until a new chunk has been cached. The average rebuffering time can be calculated as

$$Q_{rebuf} = \frac{1}{N_{chunk}} \sum_{i=1}^{N_{chunk}} \tau_i.$$
(12)

It is worth noting that if rebuffering does not occur when playing video chunk *i*, the rebuffering time is 0.

3.2.3. Video Quality Jitter

The frequent jitter of video quality between two adjacent chunks may cause physiological symptoms for users such as dizziness and headaches. The jitter can be defined as the average changes in bit rate between two consecutive reconstructed chunks:

$$Q_{jitter} = \frac{1}{N_{chunk} - 1} \sum_{i=2}^{N_{chunk}} |R'_i - R'_{i-1}|.$$
(13)

To sum up, we express the total QoE as

$$QoE = \beta_1 Q_{quality} - \beta_2 Q_{rebuf} - \beta_3 Q_{jitter}, \tag{14}$$

where β_1 , β_2 and β_3 are non-negative weight factors. Therefore, the problem of maximizing user's QoE can be formulated as

$$\max_{\substack{(q_i,q_i') \in \mathcal{D}, l_i \in \mathcal{L} \\ s.t. \quad (3) \ - \ (8).}} QoE$$
(15)

4. The Proposed DRL-Based Adaptation Algorithm

In this section, to derive the optimal policy for our proposed SR-Telemedicine system, we employ a variant of a DRL–DDQN algorithm to solve the above optimization problem.

4.1. The Elements of DRL

In this paper, a time-slotted system is assumed, and the time slot is indexed by *t*. The duration of each time slot is set to *T*, which is identical to the length of the video chunk. Here we define the state space, action space, and reward function for the DDQN algorithm as follows.

4.1.1. State Space

The states of our proposed SR-telemedicine system at the time t, denoted by s_t , can be expressed as

$$\mathbf{s}_t = \{B_V(t), B_P(t), \mathbf{q}(t), \mathbf{h}(t), \mathbf{c_1}(t), \mathbf{c_2}(t)\},\tag{16}$$

where the $B_V(t)$, $B_P(t)$, $\mathbf{q}(t)$, $\mathbf{h}(t)$, $\mathbf{c_1}(t)$, and $\mathbf{c_2}(t)$ are defined in Formulae (4), (7), (1), (2) (9) and (10), respectively. All the possible values of \mathbf{s} construct the state space S.

4.1.2. Action Space

At the beginning of each time slot t, the agent jointly determines the reconstructed high-resolution level $q'(t) \in \mathcal{K}$, and the model size of VSR (i.e., the number of ResBlocks $l(t) \in \mathcal{L}$) for all video chunks i which are processed during the time slot t. Therefore, the action vector is

$$\mathbf{a}_t = \{q'(t), l(t)\}, \forall q'(t) \in \mathcal{K}, l(t) \in \mathcal{L}.$$
(17)

All the possible values of **a** construct the state space A.

4.1.3. Reward Function

The objective of our proposed algorithm is to improve QoE; thus, the reward function can be designed according to Formula (14),

$$r_t = \beta_1 \overline{R'}(t) - \beta_2 \overline{\tau}(t) - \beta_3 |\overline{R'}(t) - \overline{R'}(t-1)|,$$
(18)

where R' denotes the average bit rate of the reconstructed video chunks during the time slot t, $\overline{\tau}$ indicates the average rebuffering time that occurs during during the time slot t, $|\overline{R'}(t) - \overline{R'}(t-1)|$ is the average jitter in bit rate between the time slot t and t-1, and β_1 , β_2 and β_3 are non-negative weight factors.

4.2. The Preliminary of DQN

First, the basic background knowledge related to the deep Q-Network [35] is introduced. In the DQN algorithm, the input of Q network is the state **s** observed by the DRL agent. The output is the estimated value of action value $Q(\mathbf{s}, \mathbf{a})$ of the state when all available actions $\mathbf{a} \in A$ are executed in the same state. Two networks with the same structure, namely, critic network Q and target network Q^- , were used in the DQN algorithm. Critic network Q was used to select actions and update parameters, whereas target network $Q^$ was used to calculate the target Q value. The parameters of critic network Q were updated through iterations, whereas the parameters of target network θ^- are slowly updated with the parameters of DQN θ . A parameter τ can be applied to control the update rate, i.e., $\theta^- \leftarrow \tau \theta + (1 - \tau)\theta^-$.

In the *t*-th iteration, the parameters θ can be updated via the gradient of the least squares loss function

$$L_t(\theta) = [y_t - Q(\mathbf{s}_t, \mathbf{a}_t; \theta)]^2,$$
(19)

where y_t is the target value that can be estimated by

$$y_t = r(\mathbf{s}_t, \mathbf{a}_t) + \gamma \max_{\mathbf{a}'} Q^-(\mathbf{s}_{t+1}, \mathbf{a}'; \theta^-),$$
(20)

where θ^- indicates the weight parameters of the target network. In practice, DQN applies experience replay and target network for its training [35–37]. The state transition samples, which are collected when the DRL agent interacts with the environment, are stored in the experience replay buffer. The DQN is updated with a mini-batch sampled from this replay buffer. The utilization of experience replay is able to avoid oscillations or divergence and smooth out the training.

4.3. The Training of DDQN Algorithm

In the DQN algorithm, the target value can be directly calculated through the greedy algorithm. Although the *Q* value can quickly be closer to the possible optimization target in this way, the target *Q* value will be overestimated and the deviation will be large.

In the DDQN algorithm [26], overestimation can be eliminated by decoupling the action selection and calculation of the target Q value. When updating the parameters of the target Q network, the critic network Q will be employed first to choose an optimal action $\mathbf{a}_{max}(\mathbf{s}_{t+1})$, which can be expressed as

$$\mathbf{a}_{max}(\mathbf{s}_{t+1}, \theta) = \arg\max Q(\mathbf{s}_{t+1}, \mathbf{a}; \theta).$$
(21)

Then, the target value of DDQN that can be estimated by

$$y_t^{ddqn} = r(\mathbf{s}_t, \mathbf{a}_t) + \gamma Q(\mathbf{s}_{t+1}, \mathbf{a}_{max}(\mathbf{s}_{t+1}, \theta); \theta^-).$$
(22)

The pseudocode of training the DDQN algorithm is shown in Algorithm 1.

Algorithm 1 The Training of the DDQN Algorithm in SR-telemedicine

Input: Initialize the weights θ of the Q-network $Q(\mathbf{s}, k, \mathbf{x}_k; \theta)$ with random values. Initialize the weights θ^- of the target network Q' with θ .

Initialize replay buffer \mathcal{R} , and minibatch size B.

- 1: **for** Episode = 1 to T_{end} **do**
- 2: Initialization: Initialize the first state **s**
- 3: Update the basic exploration factor ϵ based on the number of episodes
- 4: **for** the each initialized first state **s do**
- 5: With probability 1ϵ , select action $\mathbf{a} = \arg \max Q(\mathbf{s}, \mathbf{a}; \theta)$,
- 6: or randomly select an action **a**
- 7: Take action **a**, observe reward r_t and the next state **s**'
- 8: Store the transition $\{\mathbf{s}, \mathbf{a}, r, \mathbf{s}'\}$ into \mathcal{R}
- 9: Sample a random minibatch of *B* transitions
- 10: $\{\mathbf{s}_{j}, \mathbf{a}_{j}, r_{j}, \mathbf{s}_{j+1}\}_{1 \le j \le |B|}$
- 11: Set $y_i^{ddqn} = r(\mathbf{s}_i, \mathbf{a}_i) + \gamma Q(\mathbf{s}_{i+1}, \arg \max Q(\mathbf{s}_{i+1}, \mathbf{a}; \theta); \theta^-)$

12: Use the mean square loss function (19) to update the network weight θ via the stochastic gradient method

- 13: Update the parameters for the network $Q^{-}(\theta^{-})$ every *C* steps
- 14: $\theta^- \leftarrow \tau \theta + (1 \tau) \theta^-$
- 15: Until **s** ends in the "Terminated" state
- 16: **end for**
- 17: Until reaching the maximum number of iteration episodes T_{end}
- 18: end for

5. Evaluation

In this section, extensive experiments based on real-world traces are carried out to evaluate the performance of our proposed SR-Telemedicine system, and the experimental results are illustrated and analyzed.

5.1. Dataset

To drive our experiments, 27 videos (resolution: 1080p) are collected from Youtube with the length ranging from 40 s to 180 s. The chunk size, frame rate, and bit rates are set to 4 s, 24 fps and {400 kbps, 800 kbps, 1200 kbps, 2400 kbps, and 4800 kbps} (the corresponding resolutions are {240p, 360p, 480p, 720p, 1080p}), respectively. To evaluate the performance of our proposed SR-Telemedicine system under limited network bandwidth, we here assume that the resolution of the video sent from the patient is fixed to 240p; thus, $D = \{(240p, 720p), (240p, 1080p)\}$.

To simulate the two-stage transmission in SR-Telemedicine, a real-world bandwidth dataset consisting of 508 throughput traces from Norway's 3G network and 421 traces from U.S. broadband are utilized, which is compiled by the authors of Pensieve [30].

5.2. Setup

Because $\mathcal{D} = \{(240p, 720p), (240p, 1080p)\}$, the 240p video chunks transmitted from the patient to the physician can be reconstructed to 720p and 1080p. We will train one neural network model for each pair in \mathcal{D} with 4 levels of scales (different number of ResBlocks), respectively. The 4 levels of scales can be denoted as {low: {ResNet blocks: 2}, medium: {ResNet blocks: 4}, high: {ResNet blocks: 6}, and ultrahigh: {ResNet blocks: 8}}. Some other parameters are set as follows: the max length of the VSR processing buffer B_V^{max} and playback buffer B_P^{max} are both set as 60 s (the same as in Pensieve [30]), and the number of historical video chunks δ is set as 8. For the DDQN algorithm, the discount factor $\gamma = 0.95$, ϵ ranges from 1 to 0.001, and the initial learning rate is set as 0.0003.

5.3. Baseline Algorithms

To verify the performance improvement of our proposed approach, we compare it with the following three baseline algorithms.

- Buffer-Based (BB) [38]: This approach selects the proper bit rate to keep the length of buffer larger than 5 s, and chooses the highest available bit rate when the buffer length exceeds 15 s.
- robustMPC [29]: This approach optimally combines buffer occupancy and throughput prediction, and employs model predictive control (MPC) to maximize the user's QoE.
- Pensieve [30]: This approach uses DRL to maximize QoE without VSR, which trains a neural network model that selects bit rates for future video chunks based on observations collected by client video players.

5.4. Convergence Analysis

We compare the training process of our proposed SR-Telemedicine with that of Pensieve. As illustrated in Figure 3, it can be seen that both algorithms can be trained to converge, but SR-Telemedicine outperforms Pensieve in average QoE; this implies that SR-Telemedicine can achieve better QoE under the same network condition.



Figure 3. The training processes of our proposed SR-Telemedicine approach and Pensieve. It can be seen that both algorithms can be trained to converge.

5.5. Performance Comparison

After training the proposed and learning-based baseline algorithms, we conducted experiments to compare our proposed SR-Telemedicine with baseline algorithms in terms of average QoE, average bit rate, average rebuffering time, and average jitter.

First, from Figure 4a, we can see that SR-Telemedicine can improve the average QoE by 17–79% compared with the three baseline algorithms. Specifically,

• **BB vs. robustMPC:** robustMPC can improve the average QoE by 20% compared to BB. This is because robustMPC can make an accurate prediction on the bandwidth of the network and achieve a better match between the video resolution and the network dynamic condition.

- robustMPC vs. Pensieve: Pensieve improves the average QoE over robustMPC by 30%. The reason is that the DRL-based algorithm can better capture the network dynamics and select a more appropriate resolution with a better QoE.
- Pensieve vs. SR-Telemedicine: SR-Telemedicine outperforms Pensieve, and can increase the average QoE by 17%. This is because VSR can improve users' QoE by reconstructing the low-resolution video chunk into high-resolution chunks, even over a network with limited bandwidth.



Figure 4. The comparison results between our proposed SR-telemedicine and baseline algorithms. (a) The average QoE. SR-Telemedicine can improve the average QoE by 17 \sim 79%. (b) The average bit rate. SR-Telemedicine can largely increase the average video bit rate due to VSR processing on the edge node. (c) The average rebuffering time. SR-Telemedicine is worse than Pensieve because the reconstruction of the low-resolution video chunks at the edge node would increase the amount of transmitted data and thus increase the probability of rebuffering. (d) The average jitter. SR-Telemedicine can always promote the low resolution to a higher resolution, and thus video quality switching can be effectively avoided.

Furthermore, from Figure 4b, it can be observed that SR-Telemedicine can largely increase the average video bit rate compared to the baseline algorithms. This is due to VSR processing on the edge node. The proposed SR-Telemedicine system can provide video chunks with higher quality under the same network condition as that of the baseline algorithms. Even when the network condition is poor, SR-Telemedicine can still obtain low-resolution videos from the patient, which can be then reconstructed into high-resolution videos at the edge node by a VSR processor before sending it to the physician. However, the three baseline algorithms can only choose to deliver low-quality videos to the physician under poor network conditions.

In Figure 4c, we can see that Pensieve can obtain a better rebuffering time than the other algorithms because of the advantages of the DRL technique. Our proposed SR-Telemedicine is also worse than Pensieve; this is because SR-Telemedicine increases the amount of data transmitted from the edge node to the physician due to the reconstruction of the low-resolution video chunks, thus increasing the probability of rebuffering. Although

robustMPC performs best in rebuffering time, low-resolution videos are always delivered to the physician's side and would result in a poor visual experience. In Figure 4d, it can be seen that SR-Telemedicine can achieve the best performance with regard to the videoquality jitter. SR-Telemedicine can always promote the low resolution to a higher resolution, and thus video quality switching can be effectively avoided.

In addition, we conducted experiments to analyze the cumulative distribution function (CDF) of QoE for all employed algorithms in this paper. From Figure 5, we can see that our proposed SR-Telemedicine outperforms the baseline algorithms in the average QoE and can achieve a better QoE with a higher probability. At the same time, the maximum QoE that SR-Telemedicine can achieve is also better than other baseline algorithms. This performance gain mostly comes from the VSR processing on the edge node near the physician.



Figure 5. Cumulative distribution function (CDF) of QoE. It can be seen that our proposed SR-Telemedicine outperforms the baseline algorithms in the average QoE and can achieve a better QoE with a higher probability.

5.6. Scalability Analysis for SR-Telemedicine

In order to verify the validity and feasibility of the multi-scale VSR model, we recorded the reconstructed resolution level and the corresponding selected VSR model as illustrated in Figure 6. When SR-Telemedicine reconstructs the resolution of the video chunks from 240p to 720p or 1080p, it actually chooses different model scales to adapt to the time-varying computing resources. In this case, we may maintain a continuous and stable resolution level to reduce the video-quality jitter between adjacent video chunks even though the computational capability changed.



Figure 6. The model level and the number of ResNet block for different chunks. It can be found that our proposed algorithm actually chooses different model scales adapt to the time-varying computing resources.

6. Discussion

In this paper, patient medical videos are reconstructed adaptively via VSR at the edge node near the physician side. This paradigm can break the dependence of video transmission on the network condition, save network bandwidth and enhance the medical video quality received by the physician. However, VSR may introduce "fake" information to the physician when the VSR model or its generalization ability is not good enough. To mitigate this issue, (1) a more powerful VSR model could be trained to improve the quality of the enhanced videos, and (2) some new VSR algorithms could be utilized to improve the VSR performance. For example, Khani et al. [39] proposed SRVC, which encodes video into two bit streams: (i) a content stream, produced by compressing downsampled lowresolution video with the existing codec, and (ii) a model stream, which encodes periodic updates to a lightweight SR neural network customized for short segments of the video. SRVC decodes the video by passing the decompressed low-resolution video frames through the (time-varying) SR model to reconstruct high-resolution video frames. At the same time, to avoid the risk caused by the fake information, some real-time and quality-sensitive telemedicine applications, such as telesurgery, are not suitable for applying VSR at the current stage.

7. Conclusions

In this paper, we investigated the problem of providing high QoE medical video service in telemedicine systems. A super-resolution-empowered adaptive medical video streaming in telemedicine system (named SR-Telemedicine) is designed to achieve this purpose. SR-Telemedicine can reconstruct the low-resolution video chunks into high-resolution ones with several different high-resolution levels at the edge node near the physician. In addition, the neural network of the VSR model employed in the SR-Telemedicine system is designed to be scalable. The scale of the VSR model can be dynamically determined according to the computational capability of the edge node and the network conditions by applying the early exit mechanism. To jointly select the optimal reconstructed highresolution level and the scale of the VSR model, a DDQN-based algorithm is proposed with the aim of maximizing QoE. Extensive experiments based on real-world traces verified that the proposed SR-Telemedicine system can improve the QoE of medical video streaming by 17–79% compared to three baseline algorithms. **Author Contributions:** Conceptualization, H.H.; methodology, H.H. and J.L.; data curation, J.L.; simulation, H.H. and J.L.; writing—original draft preparation, H.H. and J.L.; supervision, H.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Hassan, M.A.; Yin, X.; Zhuang, Y.; Aldridge, C.M.; McMurry, T.; Southerland, A.M.; Rohde, G.K. A Pilot Study on Video-based Eye Movement Assessment of the NeuroEye Examination. In Proceedings of the 2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI), Virtual Conference, 27–30 July 2021; pp. 1–4.
- Suriani, N.S.; Jumain, N.A.; Ali, A.A.; Mohd, N.H. Facial Video based Heart Rate Estimation for Physical Exercise. In Proceedings of the 2021 IEEE Symposium on Industrial Electronics & Applications (ISIEA), Virtual Conference, 10–11 July 2021. pp. 1–5.
- Oviyaa, M.; Renvitha, P.; Swathika, R. Real Time Tracking of Heart Rate from Facial Video Using Webcam. In Proceedings of the 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 15–17 July 2020; pp. 1–7.
- Doshi, M.; Fafadia, M.; Oza, S.; Deshmukh, A.; Pistolwala, S. Remote diagnosis of heart disease using telemedicine. In Proceedings of the 2019 International Conference on Nascent Technologies in Engineering (ICNTE), Navi Mumbai, India, 4–5 January 2019; pp. 1–5.
- 5. Young, A.; Orchanian-Cheff, A.; Chan, C.T.; Wald, R.; Ong, S.W. Video-based telemedicine for kidney disease care: A scoping review. *Clin. J. Am. Soc. Nephrol.* 2021, *16*, 1813–1823. [CrossRef]
- 6. Bentaleb, A.; Taani, B.; Begen, A.C.; Timmerer, C.; Zimmermann, R. A Survey on bit rate Adaptation Schemes for Streaming Media Over HTTP. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 562–585. [CrossRef]
- Lavanya, K.; Vimala Devi, K.; Subramaniam, M. Quality of experience (QOE) content aware hybrid lean predictive models for medical video transmission over Internet of things (IOT) networks. *Int. J. Commun. Syst.* 2021, e4833. [CrossRef]
- Yeo, H.; Jung, Y.; Kim, J.; Shin, J.; Han, D. Neural adaptive content-aware internet video delivery. In Proceedings of the 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18), Carlsbad, CA, USA, 8–10 October 2018; pp. 645–661.
- Zhang, Y.; Zhang, Y.; Wu, Y.; Tao, Y.; Bian, K.; Zhou, P.; Song, L.; Tuo, H. Improving quality of experience by adaptive video streaming with super-resolution. In Proceedings of the IEEE INFOCOM 2020-IEEE Conference on Computer Communications, Virtual Conference, 6–9 July 2020; pp. 1957–1966.
- 10. Zhang, A.; Li, Q.; Chen, Y.; Ma, X.; Zou, L.; Jiang, Y.; Xu, Z.; Muntean, G.M. Video super-resolution and caching—An edge-assisted adaptive video streaming solution. *IEEE Trans. Broadcast.* 2021, *67*, 799–812. [CrossRef]
- 11. Zhang, S.; Liang, G.; Pan, S.; Zheng, L. A fast medical image super resolution method based on deep learning network. *IEEE Access* **2018**, *7*, 12319–12327. [CrossRef]
- Ren, S.; Guo, H.; Guo, K. Towards efficient medical video super-resolution based on deep back-projection networks. In Proceedings of the 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Atlanta, GA, USA, 14–17 July 2019; pp. 682–686.
- Ren, S.; Li, J.; Guo, K.; Li, F. Medical video super-resolution based on asymmetric back-projection network with multilevel error feedback. *IEEE Access* 2021, 9, 17909–17920. [CrossRef]
- Shi, J.; Liu, Q.; Wang, C.; Zhang, Q.; Ying, S.; Xu, H. Super-resolution reconstruction of MR image with a novel residual learning network algorithm. *Phys. Med. Biol.* 2018, 63, 085011. [CrossRef] [PubMed]
- 15. de Farias, E.C.; Di Noia, C.; Han, C.; Sala, E.; Castelli, M.; Rundo, L. Impact of GAN-based lesion-focused medical image super-resolution on the robustness of radiomic features. *Sci. Rep.* **2021**, *11*, 1–12. [CrossRef]
- Qiu, D.; Zheng, L.; Zhu, J.; Huang, D. Multiple improved residual networks for medical image super-resolution. *Future Gener. Comput. Syst.* 2021, 116, 200–208. [CrossRef]
- 17. Zhao, T.; Hu, L.; Zhang, Y.; Fang, J. Super-resolution network with information distillation and multi-scale attention for medical CT image. *Sensors* **2021**, *21*, 6870. [CrossRef]
- 18. Jebadurai, J.; Peter, J.D. Super-resolution of retinal images using multi-kernel SVR for IoT healthcare applications. *Future Gener. Comput. Syst.* **2018**, *83*, 338–346. [CrossRef]
- 19. Deeba, F.; Kun, S.; Dharejo, F.A.; Zhou, Y. Wavelet-based enhanced medical image super resolution. *IEEE Access* 2020, *8*, 37035–37044. [CrossRef]

- Usman, M.A.; Philip, N.Y.; Politis, C. 5G enabled mobile healthcare for ambulances. In Proceedings of the 2019 IEEE Globecom Workshops (GC Wkshps), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6.
- 21. Rajavel, R.; Ravichandran, S.K.; et al. IoT-based smart healthcare video surveillance system using edge computing. *J. Ambient. Intell. Humaniz. Comput.* **2022**, *13*, 3195–3207. [CrossRef]
- 22. Antoniou, Z.C.; Panayides, A.S.; Pantzaris, M.; Constantinides, A.G.; Pattichis, C.S.; Pattichis, M.S. Real-time adaptation to time-varying constraints for medical video communications. *IEEE J. Biomed. Health Inform.* 2017, 22, 1177–1188. [CrossRef]
- Ghimire, A.; Alsadoon, A.; Prasad, P.; Giweli, N.; Jerew, O.D.; Alsadoon, G. Enhanced the Quality of telemedicine Real-Time Video Transmission and Distortion Minimization in Wireless Network. In Proceedings of the 2020 5th International Conference on Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA), Sydney, Australia, 25–27 November 2020; pp. 1–10.
- 24. Delhaye, R.; Noumeir, R.; Kaddoum, G.; Jouvet, P. Compression of patient's video for transmission over low bandwidth network. *IEEE Access* **2019**, *7*, 24029–24040. [CrossRef]
- Teerapittayanon, S.; McDanel, B.; Kung, H.T. Branchynet: Fast inference via early exiting from deep neural networks. In Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016; pp. 2464–2469.
- Van Hasselt, H.; Guez, A.; Silver, D. Deep reinforcement learning with double q-learning. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February; Volume 30.
- Izima, O.; de Fréin, R.; Malik, A. A Survey of Machine Learning Techniques for Video Quality Prediction from Quality of Delivery Metrics. *Electronics* 2021, 10, 2851. [CrossRef]
- Spiteri, K.; Urgaonkar, R.; Sitaraman, R.K. BOLA: Near-optimal bit rate adaptation for online videos. *IEEE/ACM Trans. Netw.* 2020, 28, 1698–1711. [CrossRef]
- Yin, X.; Jindal, A.; Sekar, V.; Sinopoli, B. A control-theoretic approach for dynamic adaptive video streaming over HTTP. In Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication, London, UK, 17–21 August 2015; pp. 325–338.
- Mao, H.; Netravali, R.; Alizadeh, M. Neural adaptive video streaming with pensieve. In Proceedings of the 2017 ACM Special Interest Group on Data Communication, Los Angeles, CA, USA, 21–25 August 2017; pp. 197–210.
- 31. Dong, P.; Ning, Z.; Obaidat, M.S.; Jiang, X.; Guo, Y.; Hu, X.; Hu, B.; Sadoun, B. Edge Computing Based Healthcare Systems: Enabling Decentralized Health Monitoring in Internet of Medical Things. *IEEE Netw.* **2020**, *34*, 254–261. [CrossRef]
- Wang, Y.; Tran, P.; Wojtusiak, J. From Wearable Device to OpenEMR: 5G Edge Centered telemedicine and Decision Support System. In Proceedings of the 2022 HEALTHINF, Virtual Conference, 9–11 February 2022; pp. 491–498.
- Prabhu, M.; Hanumanthaiah, A. Edge Computing-Enabled Healthcare Framework to Provide Telehealth Services. In Proceedings of the 2022 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET), Virtual Conference, 24–26 March 2022; pp. 349–353. [CrossRef]
- 34. Oueida, S.; Kotb, Y.; Aloqaily, M.; Jararweh, Y.; Baker, T. An Edge Computing Based Smart Healthcare Framework for Resource Management. *Sensors* 2018, *18*, 4307. [CrossRef]
- Mnih, V.; Kavukcuoglu, K.; Silver, D.; et al. Human-level control through deep reinforcement learning. *Nature* 2015, 518, 529–533. [CrossRef]
- Ran, Y.; Zhou, X.; Hu, H.; Wen, Y. Optimizing Data Centre Energy Efficiency via Event-Driven Deep Reinforcement Learning. IEEE Trans. Serv. Comput. 2022, [CrossRef]
- Ran, Y.; Hu, H.; Wen, Y.; Zhou, X. Optimizing Energy Efficiency for Data Center Via Parameterized Deep Reinforcement Learning. *IEEE Trans. Serv. Comput.* 2022, [CrossRef]
- Huang, T.Y.; Johari, R.; McKeown, N.; Trunnell, M.; Watson, M. A buffer-based approach to rate adaptation: Evidence from a 561 large video streaming service. In Proceedings of the 2014 ACM Conference on SIGCOMM, Chicago, Il, USA, 17–22 August 2014; pp. 187–198.
- Khani, M.; Sivaraman, V.; Alizadeh, M. Efficient Video Compression via Content-Adaptive Super-Resolution. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Virtual Conference, 11–17 October 2021; pp. 4521–4530.