Article

Deep Reinforcement Learning-Based Video Offloading and Resource Allocation in NOMA-Enabled Networks

Siyu Gao , Yuchen Wang , Nan Feng , Zhongcheng Wei , Jijun Zhao *

Posted Date: 25 April 2023

doi: 10.20944/preprints202304.0891.v1

Keywords: mobile edge computing (MEC); non-orthogonal multiple access (NOMA); video offloading; resource allocation; deep reinforcement learning (DRL)

Article

Deep Reinforcement Learning-Based Video Offloading and Resource Allocation in NOMA-Enabled Networks

Siyu Gao [1,2], Yuchen Wang [1,2], Nan Feng [1,2], Zhongcheng Wei [1,2] and Jijun Zhao [1,2,*]

[1] School of Information and Electrical Engineering, Hebei University of Engineering, Handan 056038, China; yayaye1216@163.com (S.G.); ssdlhy123450@163.com (Y.W.); fengnan3128@163.com (N.F.); weizhongcheng@hebeu.edu.cn (Z.W.)

[2] Hebei Key Laboratory of Security and Protection Information Sensing and Processing, Handan 056038, China

* Correspondence: zjijun@hebeu.edu.cn

Abstract: With the proliferating of video surveillance system deployment and related applications, real-time video analysis is very critical to achieve intelligent monitoring, autonomous driving, etc. It is non-trivial to achieve high accuracy and low latency video stream analysis through the traditional cloud computing. In this paper, we propose a non-orthogonal multiple access (NOMA) based edge real-time video analysis framework with one edge server (ES) and multiple user equipments (UEs). A cost minimization problem composed of delay, energy and accuracy is formulated to improve the QoE of UEs. In order to efficiently solve this problem, we propose the joint video frame resolution scaling, task offloading, and resource allocation algorithm based on the Deep Q-Learning Network (JVFRS-TO-RA-DQN), which effectively overcomes the sparsity of the single-layer reward function and accelerates the training convergence speed. JVFRS-TO-RA-DQN consists of two DQN networks to reduce the curse of dimension, which respectively select the offloading and resource allocation action, the resolution scaling action. Experimental results show that JVFRS-TO-RA-DQN can effectively reduce the cost of transmission and computation, and have better performance in convergence compared to other baseline.

Keywords: mobile edge computing (MEC); non-orthogonal multiple access (NOMA); video offloading; resource allocation; deep reinforcement learning (DRL)

1. Introduction

Along with the development of the communication infrastructure and embedded systems, a number of cameras have been deployed around the world to collect environmental information, including traffic monitoring, electronic health care, object tracking, and smart robotics [1]. According to Cisco's forecast, video streaming will account for 80% of total Internet traffic by 2023 [2]. In particular, surveillance cameras can produce video transmissions of nearly 25-30 frames per second. Low frame rate (1.25 Hz) moving image sequences generate more than 100M data per second. However, camera sensors have limited computing power and only support low-complexity recognition algorithms, which means that video recognition accuracy is limited. In addition, deploying a network that only uses cameras to meet computing requirements is too costly for the system. To obtain the information in video, it is necessary to send the video frame taken by the UEs to the data center assigned by calculation resources, and then deal with the desired scene information. However, the bandwidth needed to efficiently transmit and accurately analyze the video stream is prohibitive. In addition, video analytic is also computationally intensive, so it cannot meet the resource and latency requirements of video analytic services only based on the terminal and cloud data centers. Although many researchers have tried to solve this problem, there is still a huge challenge of video sensitivity to latency due to the lack of effective performance upgrades of traditional terminal and data center analysis solutions. Video processing that supports MEC is the only feasible method to satisfy the demand for lots of video streams in real time.

As an emerging computing architecture, MEC can decentralize computing power from centralized cloud centers to users and mobile devices, which are close to the edge of network [3]. Performing all or part of the computing at the network edge can not only reduce latency and the bandwidth load of the network, but also reduce the risk of privacy leakage and improve data security. Because the real-time video queries (such as object tracking, license plate readers, etc.) are closely related to their specific location, it can improve up and down perception and delay compared to offloading all video frames to the cloud by addressing these issues on edge servers closer to the terminal. Most of the existing articles on task offloading adopt orthogonal frequency division multiple access technology (OFDMA) [4], which can only provide a single channel resource for UEs and has low bandwidth resource utilization. To improve the utilization of bandwidth resources, NOMA has been proposed [5]. Unlike orthogonal multiple access technology (OMA), NOMA can accommodate more users by non-orthogonal resource allocation, that is, it can provide services to multiple users on the same subchannel, at the same frequency and time, thus improving spectral efficiency [6,7]. Applying NOMA technology to the video offloading process can effectively increase the capacity of bandwidth resources, reduce the delay of video stream offloading, and improve the user's QoE [8].

However, it is not easy to analyze real-time video offloading to the edge for analysis is also time-consuming and resource-intensive, which needs to require dynamic trade-offs between edge computing and video frame transmission. Specifically, video analysis usually has a high demand for resources, and computing on edge devices can easily lead to long latency [9], due to the relatively large amount of computing to analyze the video frame by frame [10,11], and the complex network structure will consume several GBs of memory. Because of the limitation of the computational resources of the edge, it is necessary to make a reasonable allocation of resources in order to satisfy the demand for time and precision. Generally, the offloading decision optimization problem is combined with resource allocation, which results in a non-convex NP problem. Moreover, it is a very difficult problem to determine the best way to allocate resources in a distributed environment in a time-dependent manner.

Recently, DRL has been widely used in many applications in the field of mobile communication [12]. The application of MEC in large-scale dynamic analysis of video, and DRL for the improvement of the performance in the solution of the problem, is quite interesting and stimulating. When the video analytic system lacks an effective mechanism to implement adaptive tuning configuration to compensate for the performance gap caused by differences in network dynamics, accuracy, and latency requirements. Specifically, we use the resolution as an example to illustrate the impact of video configuration on the accuracy and latency of video analysis. In this paper, we divide the total time into several equal time slots and define fps as the number of frames for each time slot. Images with high resolution are also more accurate, but at the same fps, there is bound to be a longer transmission delay. When the network bandwidth is time-varying, translating video with high resolution may cause resource consumption and high latency, and translating video with low configuration can reduce network utilization and affect analysis accuracy.

Based on the above discussion, this paper propose JVFRS-TO-RA-DQN algorithm to optimize video edge offloading and resource allocation decision. JVFRS-TO-RA-DQN algorithm contains two DQN networks, one is used to select offloading and resource allocation action, and the other is used to select video frame resolution scaling action, which effectively overcomes the sparsity of the single-layer reward function and accelerates the training convergence speed. Experimental results show that JVFRS-TO-RA-DQN performs better in terms of the convergence and training effificiency. The key contributions of this paper are as follows.

- Propose a three-layer NOMA-assisted video edge scheduling architecture, where UEs are divided into different clusters of NOMA, and tasks generated by UEs in the same cluster are offloaded over a common subchannel to improve efficiency of offloading.
- Aiming to optimize the QoE of UEs, formulate a cost minimization problem composed of delay, energy and accuracy, to weigh the relationship between the three parameters.

- Propose JVFRS-TO-RA-DQN algorithm to solve the joint optimization problem. Carry out comparative experiments and prove JVFRS-TO-RA-DQN algorithm can achieve better performance gains in improving video analysis accuracy, reducing total delay and energy consumption compared to other baseline.

The rest of this article is organized as follows. In section 2, we review the relevant work done in other studies. Then, section 3 deals with the problem description and the scheduling model. Section 4 depicts the details of our algorithm's implementation. Then, in section 5, the results of the simulation are analyzed. Finally, we summarize our work in section 6.

2. Related Works

2.1. NOMA-Enabled Task Offloading in MEC Scenarios

The video surveillance system has been used extensively in various industries, and it has been increasingly intelligent, for example, face recognition and object detection [13,14], etc. Along with the rapidly growing number of video monitoring applications, there is necessary to transfer and analyze a lot of video data. Because of its intelligence in computing and caching, MEC has great potential in terms of reducing delay [15–17] and network burden [18–20]. It is possible to reduce the transmission burden and latency experienced by IoT users by offloading a computation-intensive and latency-sensitive task to an adjacent MEC server. In order to further optimize the allocation of spectrum resources, achieve high speed transmission and wide coverage, most studies have combined NOMA and MEC. The authors in [21] considered the partial offloading and binary offloading problems under time division multiple access (TDMA) and NOMA, and tried to maximize the computing efficiency of the system. The authors in [22] proposed the UDHN based NOMA-MEC system, and studied the resource allocation problem of multi-SBS and multi-users to minimize user energy consumption and task delay. The authors in [23] focus on downlink NOMA based MEC system to reduce the transmission delay and optimize workload offloading allocation. Then, they designed an algorithm based on channel quality ranking to obtain the optimal offloading decision. The authors in [24] considered the task random arrival and uncertainty of channel conditions, and proposed a decentralized DRL framework to solve the problem of power allocation, where the state was based on local observations.

2.2. Video Analysis in MEC Scenarios

Due to the increasing demand for video surveillance applications for resources [25], offloading them to edge devices for processing has been widely studied by industry scholars. And these studies focus on optimizing the target through intelligent offloading or adaptive configuration. The authors in [26] observed ROI changes from the terminal equipment's perspective, decoupled the rendering part and the offloading part, and the fast object tracking way was used locally to solve this problem. The authors in [27] investigated edge-end cloud collaboration for real-time video analytic and designed an online algorithm to achieve near-optimal utility while meeting low delay requirements by adjusting the quality of video frames generated on terminals and querying the edge resources allocated to each video. The authors in [28] proposed a new approach to designing a video configuration decision-making system, which examined the influence of video content on the basis of both the frame rate and the resolution of the video stream. The model did not rely on any precoding procedure and specific assumptions about the environment and adaptively made configuration decisions based only on learning from observing the resultant performance of historical decisions. The authors in [29] used Lyapunov and Markov approximation, to optimize video configuration and network bandwidth resource allocation, which solved the problems of capacity limitation and network dynamic changes in edge-based video analysis systems. However, for real surveillance cameras, the resolution can only be selected downward, and when the network resources are sufficient, we can consider selecting the higher resolution using super-resolution techniques to obtain higher video frame resolution and video analysis accuracy.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 April 2023 doi:10.20944/preprints202304.0891.v1

2.3. Video Offloading Based on DRL

In the last few years, the development of artificial intelligence (AI) technologies has led to rapid progress in the application of reinforcement learning in the case of modeling, routing, and resource management in a model-free environment. An adaptive video configuration network was pursued in [30], which was based on a black-box approach independent of the detailed analytical performance modeling. the authors presented and designed an intelligent system named Cuttlefish, which was a kind of smart coder that can adapt to the needs of the user and did not use any pre-programmed models or specific assumptions. The authors in [31] dealt with the problem of joint configuration adaptation and bandwidth allocation in an edge-assisted real-time video analysis system. They presented a novel approach that can be used to select the right configurations for multiple video streams based on the state of the network and the content of the video immediately. To work out the issue of collaboration in the MEC network, an AI-based task allocation algorithm was presented in [32], which was trained using a self-play strategy. The algorithm could detect the change in the network environment, and adjust the distribution policy of resources simultaneously. The authors in [33] proposed a two-layer learning model based on DQN and back propagation neural network to decrease the curse of a dimension of reinforcement learning, realize the joint decision of task offloading, wireless channel allocation, and image compression ratio selection in video analysis, and balance the accuracy of image recognition and processing delay. The authors in [34] presented a new approach to allocating resources in MEC networks using RM and DRL. Then, they presented a Collaborative offloading and resource Allocation (ACJORA), which was used to solve the problem of reducing system latency and energy consumption. However, using a DQN network to train the multi-parameter problem needs to calculate the probability of various actions and select the action with the maximum probability, which leads to high training delay and poor effect. Therefore, it is worthwhile to study how to increase the training efficiency and guarantee the precision of the network.

System Model

The video edge scheduling model is mainly split into three parts, the data generation layer, the aggregation layer, and the service layer, as shown in Figure 1. All the UEs and ES have data processing capabilities. Let's consider an edge server that assists M UEs for computation, and the set of UEs is represented by $\mathcal{M} = \{1, 2, \cdots, M\}$. $\mathcal{N} = \{1, 2, \cdots, N\}$ represents the set of the cluster of NOMA, $\mathcal{K} = \{1, 2, \cdots, K\}$ denotes the set of subchannel, all UEs in a NOMA cluster share one subchannel at the same time for offloading, with each subchannel has an equal bandwidth. We suppose the UEs adopt a binary offloading rule, that is, in each time range, the raw data must be processed locally on the terminal or remotely on ES. The UEs continuously transmit video analysis tasks to the ES.

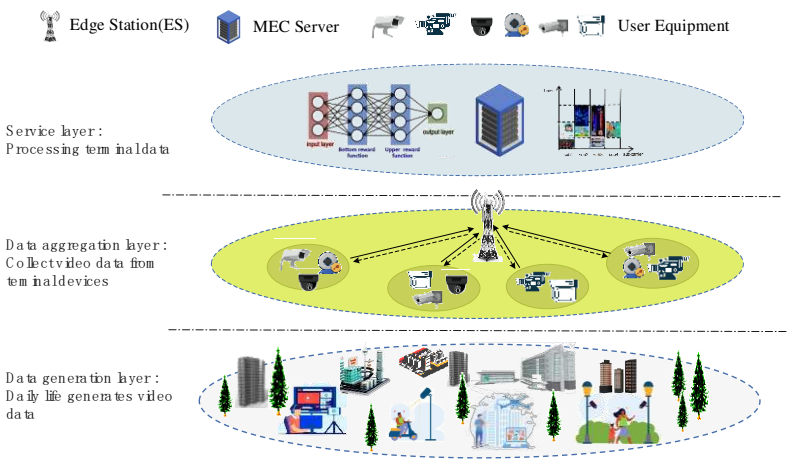

Figure 1. The video edge scheduling architecture.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 April 2023 doi:10.20944/preprints202304.0891.v1

The video stream calculation task of the UE m is expressed as D_m= {l_m, c_m, t_m^{max}}, where l_m represents the data size of the input video stream task D_m, c_m represents the total CPU cycles required for the computing task D_m, where we use t_m^{max} denotes the maximum tolerance time for task D_m. After t_m^{max}, the task D_m will be declared the process ended in failure. The data size l_m of the task D_m can be expressed as

$$l_m = \tau \cdot \eta_m^2, \tag{1}$$

where τ represents the number of bits required for a pixel to carry information. η_m^2 indicates the video frame resolution of the task D_m.

Although the edge environment is constantly changing, the network state and video data can be considered stable within a short time slot range, so we split the time into discrete time slots. Each time slot has a duration, and at the start of every time slot, the resources are reconfigured according to the present status and historical trend, so as to obtain the best resource distribution status for the overall and long-term results. In the rest of this section, we provide the NOMA-enabled transmission model (Chapter 3.1) and the edge computing model (Chapter 3.2).

3.1. NOMA-Enabled Transmission Model

We assume that each UE can only be grouped into one NOMA cluster [35]. We define $x_{m,n}$=1 represents UE m is assigned to the NOMA cluster n. While $x_{m,n}$=0 represents this assignment doesn't occur. Then

$$\sum_{n=1}^{N} x_{m,n} = \begin{cases} 1, & \text{task } D_m \text{ transmit to ES through NOMA cluster } n, \\ 0, & \text{task } D_m \text{ is computed locally.} \end{cases}. \tag{2}$$

Since the result obtained after video processing is very small, we do not consider the process of sending the result back to the UEs, and only consider the process of offloading the video stream to the ES. The uplink transmission rate of the UEs for the NOMA scheme is

$$R_m = \frac{1}{K} W \sum_{n=1}^{N} x_{m,n} \log_2 \left(1 + \frac{p_{m,k} h_{m,k}}{\sigma_k^2 + \sum_{j \neq m} p_{j,k} h_{j,k}} \right), \tag{3}$$

where W denotes the total transmission bandwidth, which may be bisected by K subchannels, $p_{m,k}$ represents the transmit power of the UE m to ES on subchannel k. $h_{m,k}$ represents the channel gain between the UE m and ES on subchannel k. σ_k^2 indicates the noise power on the subchannel k. Then the translation delay of the UE m can be expressed as

$$t_m^U = \frac{l_m}{R_m} \cdot \frac{1}{\rho_m}, \tag{4}$$

where ρ_m is the compression ratio of the video frame for UE m, which is determined by video resolution and bit rate [36].

When the video stream task D_m is offloaded to the edge for computing, The energy consumption generated by the UE m during transmission is

$$e_m^U = p_{m,k} \cdot t_m^U. \tag{5}$$

3.2. Edge Computation Model

1)Edge Computing: We denote F as the total computing capacity of the ES, and κ_m is the ratio of computing capacity allocated by the ES to the UE m. Thus, the computation delay of the task D_m offloaded to the ES is

$$t_m^C = \frac{c_m}{\kappa_m F}. \tag{6}$$

The energy consumption generated by the UE m at the edge processing is

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 April 2023 doi:10.20944/preprints202304.0891.v1

$$e_m^C = \upsilon \cdot \left(\kappa_m F\right)^2 \cdot c_m, \tag{7}$$

where υ=10^{-27} is the effective switched capacitance of the CPU, determined by the CPU hardware architecture.

2)Local Computing: For the task D_m computed locally, we use F_m^{loc} to present the computing capacity of the UE m. Thus, the latency of the task D_m processed locally is

$$t_m^{loc} = \frac{c_m}{F_m^{loc}}, \tag{8}$$

Thus, the energy consumption of the task processed locally is

$$e_m^{loc} = \upsilon \cdot \left(F_m^{loc}\right)^2 \cdot c_m, \tag{9}$$

3.3. Problem Formulation

In order to optimize multiple conflicting goals equally, a common approach is to give different weights to these conflicting goals, and then to weigh and sum the goals. In this article, improving user accuracy and reducing processing latency are the basic goals. According to the literature [37], the analytical accuracy of the task φ_m is expressed as the ratio of the number of objects which are correctly identified to the total number of objects in a video frame, which can be expressed as

$$\varphi_m = 1 - 1.578e^{-6.5\times10^{-3}\eta_m}, \tag{10}$$

Combined with formula (4) and formula (6), the total latency generated by the UE m offloaded to the ES is composed of transmission delay and computation delay, which can be expressed as

$$t_m^{off} = t_m^U + t_m^C = \frac{l_m}{R_m} \cdot \frac{1}{\gamma_m} + \frac{c_m}{\kappa_m F}, \tag{11}$$

Combined with formula (5) and formula (7), the total energy consumption generated by the UE m offloaded to the ES is composed of transmission energy consumption and computation energy consumption, which can be expressed as

$$e_m^{off} = e_m^U + e_m^C = p_{m,k} \cdot t_m^U + \upsilon \cdot \left(\kappa_m F\right)^2 \cdot c_m, \tag{12}$$

According to the assigned calculation model and communication model, the total latency required for the task D_m to be processed at t is expressed as

$$T_m = \left(1 - \sum_{n=1}^{N} x_{m,n}\right) t_m^{loc} + \sum_{n=1}^{N} x_{m,n} t_m^{off}, \tag{13}$$

At the same time, the total energy consumption of the task D_m at t can be expressed as

$$E_m = \left(1 - \sum_{n=1}^{N} x_{m,n}\right) e_m^{loc} + \sum_{n=1}^{N} x_{m,n} e_m^{off}, \tag{14}$$

The states and video content are constantly changing, causing the offloading decision and resource allocation strategies to need to be constantly adjusted to accommodate the dynamics of our environment. When designing adaptive algorithms, our goal is to optimize the cost function consisting of time delay, energy consumption, and video analysis accuracy under long-term resource constraints. Based on the design of the utility function in [38], the optimization problem can be modeled as

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 April 2023 doi:10.20944/preprints202304.0891.v1

$$\begin{aligned}
&\min \frac{1}{m}\sum_{m=1}^{M}\left[\omega_t T_m + \omega_e E_m + \left(1-\omega_t-\omega_e\right)\varphi_m\right] \\
&s.t.\ C_1: x_{m,n} \in \{0,1\}, \forall m \in \mathcal{M}, n \in \mathcal{N} \\
&\quad C_2: \sum_{n=1}^{N} x_{m,n} \le 1, \forall m \in \mathcal{M}, n \in \mathcal{N} \\
&\quad C_3: T_m \le T_m^{\max}, E_m \le E_m^{\max}, \forall m \in \mathcal{M} \\
&\quad C_4: 0 \le p_{m,k} \le P_m^{\max}, \forall m \in \mathcal{M}, k \in \mathcal{K} \\
&\quad C_5: \sum_{m=1}^{M} \kappa_m \le 1, \forall m \in \mathcal{M} \\
&\quad C_6: \eta_m \ge \eta_m^{\min}, \forall m \in \mathcal{M}
\end{aligned} \quad . \tag{15}$$

In the cost function, ω_t is the weight for the delay, and ω_e is the weight for energy consumption, where $\omega_t + \omega_e < 1$. Constraint C_1 in the objective function guarantees that a UE can only be assigned to a NOMA cluster. Constraint C_2 denotes that the UEs can only select computed locally or offloaded to ES. Constraint C_3 ensures that the maximum total delay of UE m must be less than the tolerance time $T_m^{\max}$ of the task D_m, and the total energy consumption of UE m cannot exceed the threshold $E_m^{\max}$. Constraint C_4 guarantees that the transmit power of the UE m cannot exceed the threshold $p_m^{\max}$. Constraint C_5 ensures that the allocated computation capacity cannot exceed the total capacity of the ES. Constraint C_6 guarantees that the minimum video frame resolution of task D_m must be higher than the threshold $\eta_m^{\min}$.

Two important challenges to solving this problem are the difficulty of the problem itself and the prediction of future network status, video content, and other information. Since edge nodes typically run for months or years, in order to deal with the problem of unpredictable future information, it is necessary to relax the constraints in each time slot of the objective function to the average over a long period of time. In addition, the optimization problem is a mixed integer nonlinear program which is hard to resolve even if the future information is known. To address these two challenges, we need to design an algorithm that provides the best offloading and resource allocation for video streaming without foreseeing future information.

4. Deep Reinforcement Learning-Based Algorithm

Based on the optimization target and constraints, the DRL-based algorithm is adopted. In the rest of this chapter, we first define the state space, the action space, and the reward function. What's more, we present a more detailed description of the participant critic algorithm framework.

4.1. Deep Reinforcement Learning Model

The deep reinforcement learning process reformulates the computational offloading problem as a Markov Decision Process (MDP) model. A typical MDP model consists of a tuple $\{S, A, P, R, \gamma\}$ with five elements. where S represents the state space, A represents the finite action space, P is the state transfer probability, R represents the reward function, and $\gamma \in [0,1]$ is the discount factor for future rewards. Each element of the MDP model tuple corresponds to the following meaning.

4.1.1. State Space

At time slot t, the state of the UEs includes basic information about the computational task. The state space $S_{m,t} \in S_t$ can be expressed as

$$S_{m,t} = \{s_{m,t} \mid s_{m,t} = (l_m, \eta_m, h_{m,k}, T_m^{\max})\}, \tag{16}$$

where $S_{m,t}$ denotes the state space at time slot t.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 April 2023 doi:10.20944/preprints202304.0891.v1

4.1.2. Action Space

At time slot t, the action of UE i is represented as

$$A_{m,t} = \{a_{m,t} \mid a_{m,t} = (\alpha_m, \beta_m)\}, \quad (17)$$

It consists of two vectors: the task offloading and resource allocation vector α_m, and the video frame resolution scale vector β_m. The vector α_m contains two actions: the resource allocation action $x_{m,n}$, and the offloading decision action $\sum_{n=1}^{N} x_{m,n}$. $x_{m,n}$ represents whether UE m is assigned to the NOMA cluster n, and $\sum_{n=1}^{N} x_{m,n}$ represents whether the task D_m needs to be offloaded to the ES. The vector β_m represents the action for video frame resolution compression ratio selection, where $\beta_m \in [0.5,1.5]$.

In the MEC system network proposed in this paper, the ES distributes the offloading and resource allocation policy to the UEs, and the selection of the video frames resolution also should be determined.

4.1.3. Reward Function

The formulaic question in this article contains multiple factors. Specifically, our goal is to reduce latency and energy consumption while improving video analysis accuracy as much as possible while meeting the constraints of resources. Therefore, reward functions can be designed based on the optimization problem.

We propose JVFRS-TO-RA-DQN, which contains two DQN networks, the first DQN network selects the optimal offloading and resource allocation strategy, and the second DQN network selects the appropriate video frame resolution scaling factor to ensure the maximum accuracy of video analysis, reduce the system delay and energy consumption. A detailed description of the two reward functions is given below.

After performing the action $A_{m,t}$, a reward $r_{m,t}$ will be obtained for the action $A_{m,t+1}$, that the edge server chooses to perform. The reward function is generally related to an objective function, which aims at minimizing the latency of the system in the context of task offloading and resource allocation. However, the aim of reinforcement learning training is to obtain the maximum long-term accumulation of rewards. Thus, the offloading reward function of UEs at time t can be designed as

$$\xi_{m,t} = -\left(\omega_1 T_m + \omega_2 E_m\right), \quad (18)$$

For the UEs, it should be penalized if the accuracy of the next state is not within the threshold after taking the action $a_{m,t}$. Therefore, the resolution scaling reward function is designed as

$$\zeta_{m,t} = \left(\omega_t + \omega_e - 1\right)\varphi_m, \quad (19)$$

And finally, we use $r_{m,t}=\xi_{m,t}+\zeta_{m,t}$ to represent the total reward of the system. By maximizing the long-term cumulative reward $r_{m,t}$, an efficient joint video frame resolution, task offloading, and resource allocation strategy, which we abbreviate as JVFRS-CO-RA-DQN in this paper, can be determined to achieve the minimization of system delay and energy consumption while improving the video analysis accuracy.

4.2. JVFRS-TO-RA-DQN algorithm

In this section, we propose JVFRS-TO-RA-DQN to solve the problem of jointing video frame resolution scaling, task offloading, and resource allocation. The proposed algorithm is based on DQN, which can study from the offline historical data through the experience of the simulation without requiring full environmental knowledge. The detailed algorithm is shown in Figure 2.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 April 2023 doi:10.20944/preprints202304.0891.v1

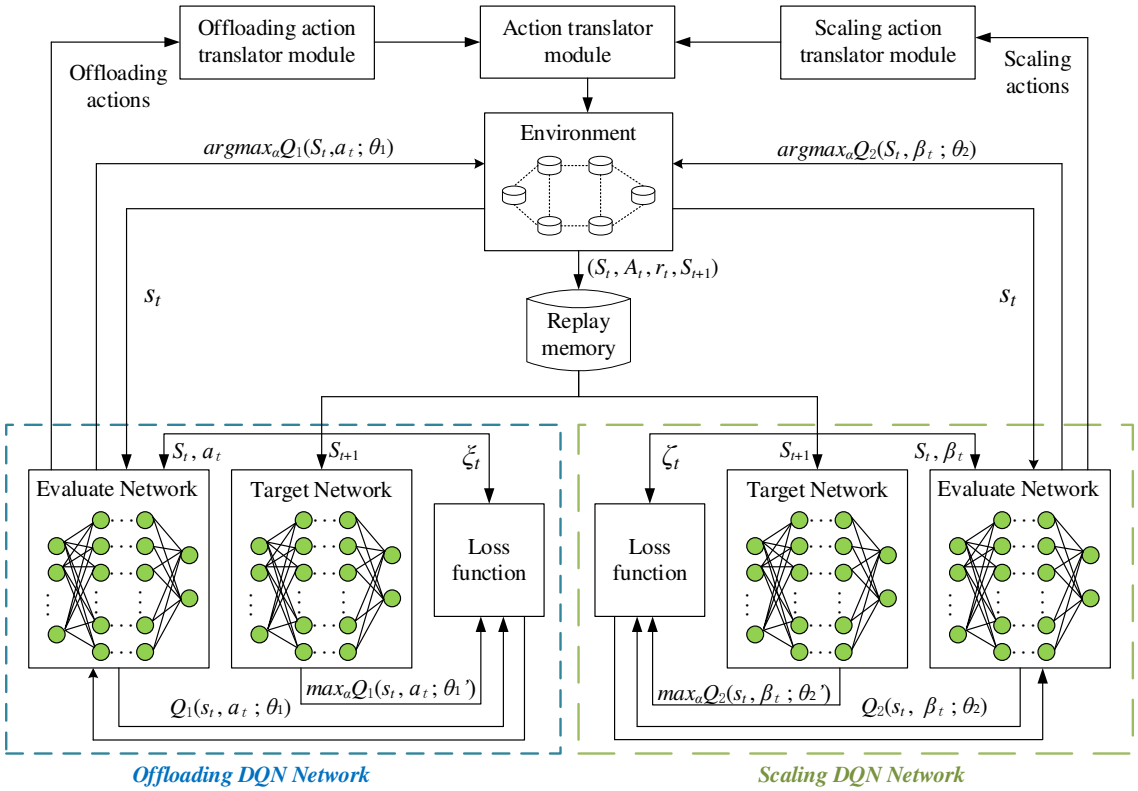

Figure 2. An illustration of the JVFRS-TO-RA-DQN architecture.

According to the state of the system at present, the DQN algorithm maximizes the predefined reward function by choosing an $A_{m,t}$ from a limited sum of actions.

In the process of training, apart from state $S_{m,t}$, action $A_{m,t}$, policy π, and reward function $r_{m,t}$, the state-action value function $Q^{\pi}(S_{m,t}, A_{m,t})$ determines the action $A_{m,t}$ of the state $S_{m,t}$ through a mapping function $\pi(A_{m,t} | S_{m,t})$. If $Q^{\pi}(S_{m,t}, A_{m,t})$ is updated at each time step, then it is assumed that it will converge to the optimum state-action value function $Q^{\pi'}(S_{m,t}, A_{m,t})$. According to the Bellman equation, the evaluation of the quality of a specific action in a given specific state can be expressed as

$$Q^{\pi}(S_{m,t}, A_{m,t}) = (1-\lambda) Q^{\pi}(S_{m,t}, A_{m,t}) + \lambda (r_{m,t} + \gamma \max_{A'} Q(S_{m,t+1}, A')), \tag{20}$$

where λ represents the learning rate which reflects the rate of the algorithm adapting to a new environment, where $\lambda \in (0,1]$. Since the complexity of formula (20) is exponentially related to the number of state-action pairs, it is more difficult to solve Q values when the state-action pairs increase. In order to accurately calculate the Q values, predicting the values of Q between different state-action pairs is significant, that is also the hinge of the DQN algorithm.

In contrast to prediction using traditional tabular Q-learning, DQN has a special replay memory structure to store the data generated after each step, including for every step. When the network trains, the network extracts some memory from the replay memory for experiential learning. The replay memory has enough training data to fit Q values of different state-action pairs, which leads to

$$Q^{\pi}(S, A) \approx \hat{Q}(S, A, \theta), \tag{21}$$

where $\hat{Q}(\cdot; \theta)$ is a deep learning network function denoted by θ. $Q^{\pi}(S, A)$ is updated by minimizing the loss function, which is defined as

$$L(\theta) = \mathbb{F}\left[\left(r_{i,t} + \gamma \max_{A'} Q(S_{i,t+1}, A') - Q(S, A, \theta)\right)^2\right], \tag{22}$$

Actually, the gradient descent algorithm is utilized to minimize the Loss in (22) and therefore to update the weight θ, so as to make it possible to minimize the error between the evaluation and the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 April 2023 doi:10.20944/preprints202304.0891.v1

target. As a result, the neural network can predict more accurately as the training process goes on. The JVFRS-CO-RA-DQN algorithm is explained in Algorithm 1.

Algorithm 1: JVFRS-CO-RA-DQN algorithm
Input: D_m, w, F, γ.
Output: α_m, β_m.

Initialize the evaluate network with random weights as θ
Initialize the target networks as a copy of the evaluate network with random weights as θ'
Initialize replay memory D
Initialize an empty state set S_Set
for episode=1 to Max **do**
  Initialize state $S_{m,t}$ in equation (16)
  for $t<T$ **do**
    With probability ε to select a random offloading and resource allocation decision $\alpha_{m,t}$; With probability δ to select a random resolution $\beta_{m,t}$
    Execute action $\alpha_{m,t}$, receive a reward $\xi_{m,t}$; execute action $\beta_{m,t}$, receive a reward $\zeta_{m,t}$
    Combine $\alpha_{m,t}$ and $\beta_{m,t}$ as $A_{m,t}$, calculate $r_{m,t}$ with $\xi_{m,t}$ and $\zeta_{m,t}$, and observe the next state $S_{m,t+1}$
    Store interaction tuple $\{S_{m,t}, A_{m,t}, r_{m,t}, S_{m,t+1}\}$ in D
    Sample a random tuple $\{S_{m,t}, A_{m,t}, r_{m,t}, S_{m,t+1}\}$ from D
    Compute the offloading target Q value and the scaling target Q value
    Train the offloading target Q value and the scaling target Q value
    Perform gradient descent with respect to θ
    Update the evaluate Q-network and target Q-network
  end for
end for

5. Results and Analysis

5.1. Parameter Setting

To verify the effectiveness of the JVFRS-CO-RA-DQN proposed in this paper, a network consisting of one MEC, four NOMA cluster and ten UEs are considered for experiments, with UEs randomly distributed within [0, 200] meters from the MEC. The total communication bandwidth W is 12 MHz. We define c_m is correlates with the data size of task as $c_m=c_m^{bit} l_m$, where c_m^{bit} is 100 cycles/bit. The computational capacity of the MEC F is 12 GHz, the computational capacity of the UEs F_m^{loc} is in [0.4, 2] GHz, the average energy consumption threshold E_m^{max} is 15 J, and the number of bits required to carry information per unit pixel of video τ is 24. The minimum transmit power p of the UEs is 0.5 W, the maximum tolerance time t_m^{max} of task D_m is 30 ms. Based on the experimental data from the literature [37], we adopt $\varphi_m = 1 - 1.578e^{-6.5\times10^{-3}\times\eta_m}$ as the analytics accuracy function on both edge servers and UEs, and the video frame resolution is higher than 40000 pixels (200 × 200). The parameters are shown in Table 1.

Table 1. Simulation parameters.

Parameters	Value
Number of UEs, M	10
Number of NOMA cluster, k	4
The distance between ES and UEs	[0, 200] m
The total communication bandwidth, W	12 MHz
Required CPU cycles for unit bit task, c_m^{bit}	100 cycles/bit
The computational capacity of the MEC server, F	12 GHz
The computational capacity of UE, F_m^{loc}	[0.4, 2] GHz
Average energy consumption threshold, E_m^{max}	15 J
Required bits representing one pixel, τ	24
Maximum transmission power, p	0.5 W

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 April 2023 doi:10.20944/preprints202304.0891.v1

Maximum tolerance time for task, t_m^{max}	30 ms
Minimum video frame resolution	40000 pixels (200 × 200)
Constant about IoT device, υ	1×10^{-27}
Compression ratio of the video frame for UE, ρ_m	74
Discount factor, γ	[0,1]
Batch size, Z	32
Replay buffer, B	100

5.2. Result Analysis

To assess performance fairly, the proposed scheme was compared with four baseline schemes:

(1) Local Computing Only (LCO): the video streams are processed totally at UEs with $\sum_N^{n=1} x_{m,n}= 0$, $\forall m\in$M, which has a fixed video frame resolution.

(2) Edge Computing Only via OMA (ECO-OMA): the video streams are totally offloaded to and processed at the MEC with $x_{m,n} = 1$, $\forall m\in$M, $n\in$N, which has a fixed video frame resolution.

(3) JVFRS-TO-RA-DQN via OMA (JVFRS-TO-RA-DQN-OMA): Unlike JVFRS-TO-RA-DQN, the task D_m generated by the UE m are offloaded through OMA. Each UE has an independent subchannel. We use y_m to denote whether the task D_m offloaded to ES, $y_m = 1$ denotes the task D_m offloaded to ES, otherwise, $y_m = 0$.

(4) Task offloading and resource allocation algorithm based on DQN via NOMA (TO-RA-DQN-NOMA) [39]: Compared with JVFRS-TO-RA-DQN, TO-RA-DQN-NOMA doesn't consider the change in video frame resolution, which means it has a fixed video frame resolution.

We first take the experiment in a specific scene. There is one MEC, four NOMA clusters and ten UEs in this scenario, with the computing capacity of the MEC server is 5 GHz, and the total communication bandwidth is 12 MHz. Then we record the system delay under five different schemes. We take 3000 experiments and average the experimental data, which is exhibited in Table 2. The average latency of the JVFRS-TO-RA-DQN scheme is 167.71 ms. That's about 74.18% less than the JVFRS-TO-RA-DQN-OMA scheme, and about 33.38% less than the TO-RA-DQN-NOMA scheme.

Table 2. The average latency of five schemes.

	LCO	ECO-OMA	JVFRS-TO-RA-DQN-OMA	TO-RA-DQN-NOMA	Proposed
The bandwidth of subchannel (*MHz*)	1.2	1.2	1.2	3	3
Average delay (*ms*)	644.54	801.49	649.66	251.77	167.71

The influence of different network environments on cost is shown in Figure 3. Figure 3 exhibits the effect of different communication bandwidths on the cost in this scenario under five schemes. First of all, the cost of all schemes decreases except for LCO scheme as the communication bandwidth increases, this is because LCO scheme transmits only at the UEs, and the change of network communication bandwidth does not affect it. And the average cost of LCO schemes does not change with the bandwidth at W = 0.6. Then the cost in other schemes decreases as the communication bandwidth increases since every UE is able to allocate more bandwidth, and the cost of communication transmission is also reduced. The cost of the proposed algorithm reduces about 50.05% compared with the JVFRS-TO-RA-DQN-OMA scheme at W=6 MHz, and about 34.32% compared with the TO-RA-DQN-NOMA scheme at W=6 MHz.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 April 2023 doi:10.20944/preprints202304.0891.v1

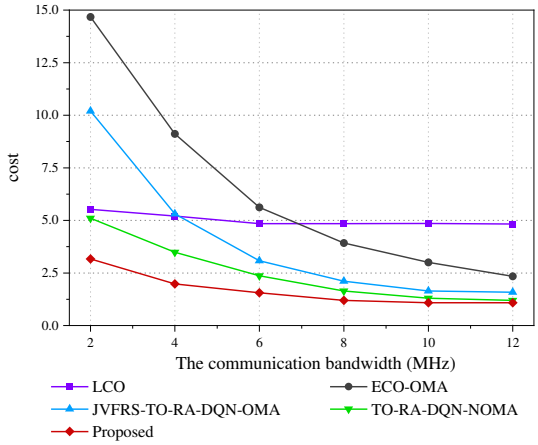

Figure 3. The effect of communication bandwidth on the cost.

Figure 4 depicts the effect of the different computational capacities of the ES on the cost in this scenario under ten schemes. Due to the fact that the UEs don't utilize the computing resources of the ES, LCO curves will not change as the computational capacity of the ES grow. The other curves are reduced as the computational capacity of the ES increases, since more server computer source is allocated for MEC, the computation time will but shortened accordingly. it is obvious that the average cost is affected mostly by some other elements when the computational capacity of the ES is much bigger than the computational capacity of UEs.

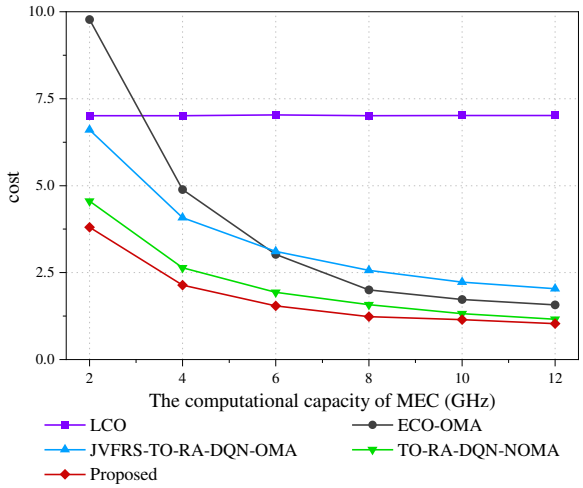

Figure 4. The effect of computational capacity of MEC on the cost.

Except for the translation latency and computation latency, the video analytic accuracy is also affected by the video frame resolution. We modify the smallest video frame resolutions in this experiment to estimate the influence of resolution on accuracy. We assume the video frame resolution is lower than 700×700 pixels, and the optimized resolution is between the smallest and the largest video frame resolutions. The video frame resolution of the LCO scheme, ECO-OMA scheme and TO-RA-DQN-NOMA scheme is identified as the average of the smallest and the largest video frame resolutions.

Figure 5 shows the effects of different smallest frame resolutions on average delay. As illustrated in Figure 5, all algorithms increase in regard to the smallest video frame resolution. The proposed algorithm keeps the system performance (both the average delay and the analytical precision), that is because the proposed algorithm can adjust the resolution of video frame adaptively to reduce the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 April 2023 doi:10.20944/preprints202304.0891.v1

system delay. The average delay of JVFRS-TO-RA-DQN-OMA scheme is higher than proposed algorithm, which means the delay of NOMA based MEC system is superior to that of OMA.

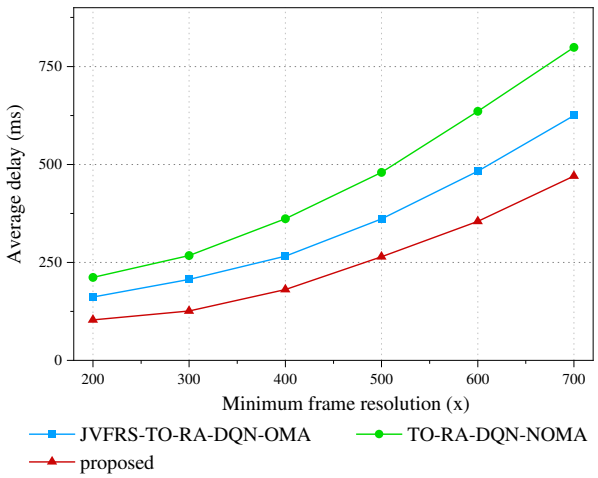

Figure 5. The effect of minimum frame resolution on average delay.

Figure 6 shows the effects of different smallest frame resolutions on video analytic accuracy. It is demonstrated that JVFRS-TO-RA-DQN-OMA algorithm is always Maintain the highest accuracy. The proposed algorithm achieves a lower latency at the expense of video analytic accuracy when the smallest video frame resolution is lower than 500×500 pixels. Nevertheless, the influence of delay on the system is gradually reduced when the smallest video frame resolution is larger than 500×500 pixels, the network resources are adequate, and the video analytic accuracy is increased continuously.

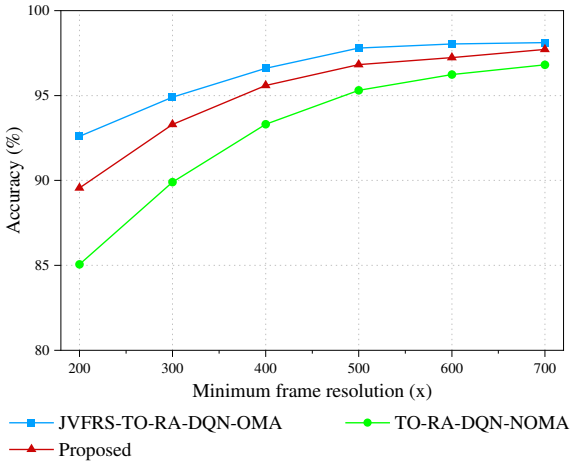

Figure 6. The effect of minimum frame resolution on video analysis accuracy.

We then depict the process of convergence with TO-RA-DQN-NOMA algorithm and the proposed algorithm. Figure 7 depicts the performance differences between TO-RA-DQN-NOMA algorithm and the proposed algorithm in different learning rates. On the one hand, the proposed algorithm obtains correspondingly higher rewards than TO-RA-DQN-NOMA scheme. This is owing to that different algorithms lead to different offloading decisions, which leads to the video frames offloaded to MEC being different, and the average delay and video analysis accuracy are also different. And then, from the simulation data in Table 3, Compared with TO-RA-DQN-NOMA algorithm, the simulation data indicate that the proposed algorithm has better performance than TO-RA-DQN-NOMA algorithm in video analysis accuracy. Moreover, the convergence rate of the proposed algorithm is higher than that of TO-RA-DQN-NOMA algorithm. What's more, we can discover that after training about 200 epochs, the proposed algorithm with both learning rates is 10^{-6}

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 April 2023 doi:10.20944/preprints202304.0891.v1

and 10^{-7} converges to a reward value, and after training more than 400 epochs, TO-RA-DQN-NOMA algorithm with both learning rates is 10^{-6} and 10^{-7} converges to a reward value.

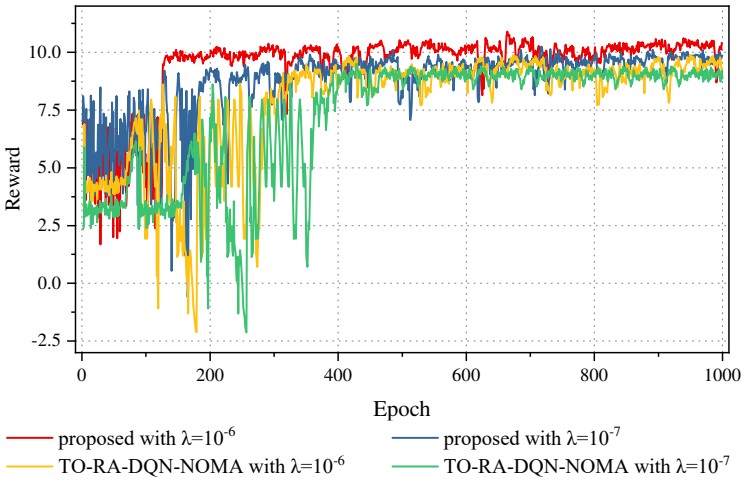

Figure 7. Convergence performance of the proposed algorithm and TO-RA-DQN-NOMA algorithm under different learning rates.

Table 3. The video analysis accuracy of two schemes.

	TO-RA-DQN-NOMA	Proposed
Learning rate 10^{-6}	95.87%	98.74%
Learning rate 10^{-7}	95.96%	98.82%

Figure 8 shows the convergence performance of our proposed algorithm with different numbers of UEs. The algorithm can converge fastly and steadily no matter how many UEs there are. What's more, the average reward of 10 UEs apparently exceeds that of 15 UEs. The average reward of 10 UEs converges to a reward for about training at 120 epochs, the average reward of 15 UEs converges to a reward for about training at 270 epochs, and the average reward of 20 UEs converges to a reward value for more than 400 epoch training. This is due to the fact that more UEs are able to offload their computing tasks at a higher rate when there are fewer UEs, which will reduce the energy consumption of UEs while enhancing the users' QoE on the basis of latency, energy consumption and video analytic accuracy.

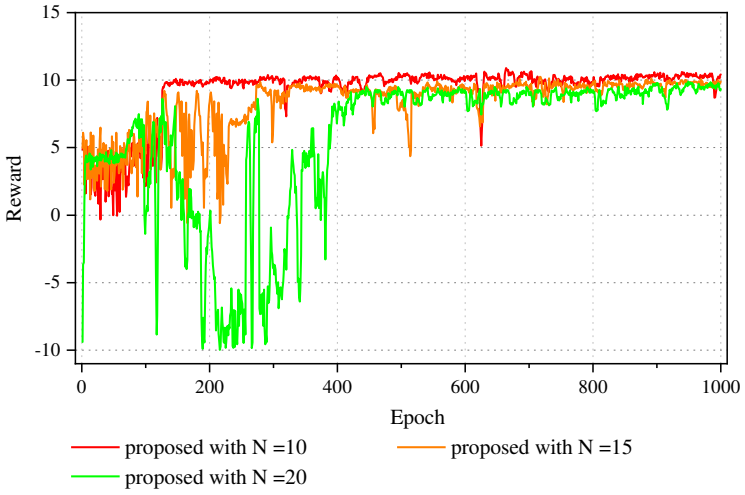

Figure 8. Convergence performance of the proposed algorithm under different IoT devices.

6. Conclusions

With the popularization of video surveillance applications and the diversification of functional applications, real-time video stream analysis is of great value in intelligent monitoring, smart cities, autonomous driving, and other scenarios. In this paper, we focused on the design of a NOMA based multiple UEs smart video analysis system for the purpose of improving the video analysis accuracy, reducing average delay and energy consumption in the system. Aiming to optimize the QoE of UEs, we formulated a cost minimization problem composed of delay, energy and accuracy, to weigh the relationship between the three parameters. The optimization function was a NP-hard problem, which was difficult to calculate the optimal solution because of it was a high dimensional nonlinear mixed integer programming problem. we proposed the JVFRS-TO-RA-DQN algorithm to solve the above problem. The proposed algorithm contains two DQN networks, one was used to select offloading and resource allocation action, and the other was used to select video frame resolution scaling action, which effectively overcame the sparsity of the single-layer reward function and accelerated the training convergence speed. A large number of simulation experiments showed that JVFRS-TO-RA-DQN algorithm can achieve better performance in improving video analysis accuracy, reducing total delay and energy consumption compared to other baseline.

Author Contributions: Conceptualization, S.G. and J.Z.; methodology, S.G.; software, S.G.; validation, S.G.; formal analysis, S.G. and Y.W.; investigation, S.G. and N.F.; resources, S.G.; data curation, S.G.; writing—original draft preparation, S.G. and N.F.; writing—review and editing, S.G. and Y.W.; visualization, S.G. and Y.W.; supervision, Z.W.; project administration, Z.W.; funding acquisition, J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Science and Technology Research Project of Higher Education Institutions of Hebei Province under Grant (NO. QN2020193) and (NO. ZD2020171), Hebei Province Innovation Capability Improvement Plan Program (22567624H), the Handan Science and Technology Research and Development Program under Grant (NO. 21422031288), and the Provincial Innovation Funding Project for Graduate Students of Hebei Province under Grant (NO. CXZZSS2023120).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data in this paper is not publicly available at this time.

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

References

1. Wan, S.; Ding, S.; Chen, C., Edge computing enabled video segmentation for real-time traffic monitoring in internet of vehicles. Pattern Recognition 2022, 121, 108146.
2. Cisco. Annual Internet Report (2018–2023) White Paper [Online]. Available online: http://www.cisco.com (accessed on 22 March 2023).
3. Luo Q, Hu S, Li C, et al. Resource scheduling in edge computing: A survey. IEEE Communications Surveys & Tutorials, 2021, 23(4): 2131-2165.
4. Wu Y, Wang Y, Zhou F, et al. Computation efficiency maximization in OFDMA-based mobile edge computing networks. IEEE Communications Letters, 2019, 24(1): 159-163.
5. Dai L, Wang B, Ding Z, et al. A survey of non-orthogonal multiple access for 5G. IEEE communications surveys & tutorials, 2018, 20(3): 2294-2323.
6. Kiani A, Ansari N. Edge computing aware NOMA for 5G networks. IEEE Internet of Things Journal, 2018, 5(2): 1299-1306.
7. Maraqa O, Rajasekaran A S, Al-Ahmadi S, et al. A survey of rate-optimal power domain NOMA with enabling technologies of future wireless networks. IEEE Communications Surveys & Tutorials, 2020, 22(4): 2192-2235.
8. Yan J, Bi S, Zhang Y J A. Offloading and resource allocation with general task graph in mobile edge computing: A deep reinforcement learning approach. IEEE Transactions on Wireless Communications, 2020, 19(8): 5404-5419.

9. Zhang, H.; Ananthanarayanan, G.; Bodik, P, et al. In Live video analytics at scale with approximation and delay-tolerance, 14th USENIX Symposium on Networked Systems Design and Implementation, 2017; 2017.
10. Zhao, Y.; Yang, Z.; He, X, et al. Trine: Cloud-edge-device cooperated real-time video analysis for household applications. IEEE Transactions on Mobile Computing 2022:1-13.
11. Ran, X.; Chen, H.; Zhu, X, et al. In Deepdecision: A mobile deep learning framework for edge video analytics, IEEE INFOCOM 2018-IEEE conference on computer communications, 2018; IEEE: 2018; pp 1421-1429.
12. Wang X, Han Y, Leung V C M, et al. Convergence of edge computing and deep learning: A comprehensive survey. IEEE Communications Surveys & Tutorials, 2020, 22(2): 869-904.
13. Ren, S.; He, K.; Girshick, R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems 2015, 28.
14. Zaidi, S. S. A.; Ansari, M. S.; Aslam, A, et al. A survey of modern deep learning based object detection models. Digital Signal Processing 2022, 103514.
15. Zhang, H.; Yang, Y.; Huang, X, et al. Ultra-low latency multi-task offloading in mobile edge computing. IEEE Access 2021, 9, 32569-32581.
16. Li, Y.; Wang, T.; Wu, Y, et al. Optimal dynamic spectrum allocation-assisted latency minimization for multiuser mobile edge computing. Digital Communications Networks 2022, 8, (3), 247-256.
17. Kuang, Z.; Ma, Z.; Li, Z, et al. Cooperative computation offloading and resource allocation for delay minimization in mobile edge computing. Journal of Systems Architecture 2021, 118, 102167.
18. Sun, Y.; Zhou, S.; Xu, J., EMM: Energy-aware mobility management for mobile edge computing in ultra dense networks. IEEE Journal on Selected Areas in Communications 2017, 35, (11), 2637-2646.
19. You, C.; Huang, K.; Chae, H, et al. Energy-efficient resource allocation for mobile-edge computation offloading. IEEE Transactions on Wireless Communications 2016, 16, (3), 1397-1411.
20. Zhou, H.; Jiang, K.; Liu, X, et al. Deep reinforcement learning for energy-efficient computation offloading in mobile-edge computing. IEEE Internet of Things Journal 2021, 9, (2), 1517-1530.
21. Zhou F, Hu R Q. Computation efficiency maximization in wireless-powered mobile edge computing networks. IEEE Transactions on Wireless Communications, 2020, 19(5): 3170-3184.
22. Cheng Q, Li L, Sun Y, et al. Efficient resource allocation for NOMA-MEC system in ultra-dense network: A mean field game approach//2020 IEEE International Conference on Communications Workshops (ICC Workshops). IEEE, 2020: 1-6.
23. Zhu B, Chi K, Liu J, et al. Efficient offloading for minimizing task computation delay of NOMA-based multiaccess edge computing[J]. IEEE Transactions on Communications, 2022, 70(5): 3186-3203.
24. Zhu H, Wu Q, Wu X J, et al. Decentralized power allocation for MIMO-NOMA vehicular edge computing based on deep reinforcement learning. IEEE Internet of Things Journal, 2021, 9(14): 12770-12782.
25. Naouri, A.; Wu, H.; Nouri, N. A, et al. A novel framework for mobile-edge computing by optimizing task offloading. IEEE Internet of Things Journal 2021, 8, (16), 13065-13076.
26. Hanyao, M.; Jin, Y.; Qian, Z, et al. In Edge-assisted online on-device object detection for real-time video analytics, IEEE INFOCOM 2021-IEEE Conference on Computer Communications, 2021; IEEE: 2021; pp 1-10.
27. Wang, Y.; Wang, W.; Liu, D, et al. Enabling edge-cloud video analytics for robotics applications. IEEE Transactions on Cloud Computing 2022.
28. Liu, L.; Li, H.; Gruteser, M. In Edge assisted real-time object detection for mobile augmented reality, The 25th annual international conference on mobile computing and networking, 2019; 2019; pp 1-16.
29. Yang, P.; Lyu, F.; Wu, W, et al. Edge coordinated query configuration for low-latency and accurate video analytics. IEEE Transactions on Industrial Informatics 2019, 16, (7), 4855-4864.
30. Chen, N.; Quan, S.; Zhang, S, et al. Cuttlefish: Neural configuration adaptation for video analysis in live augmented reality. IEEE Transactions on Parallel Distributed Systems 2020, 32, (4), 830-841.
31. Wang, C.; Zhang, S.; Chen, Y, et al. In Joint configuration adaptation and bandwidth allocation for edge-based real-time video analytics, IEEE INFOCOM 2020-IEEE Conference on Computer Communications, 2020; IEEE: 2020; pp 257-266.
32. Chen, J.; Chen, S.; Wang, Q, et al. iRAF: A deep reinforcement learning approach for collaborative mobile edge computing IoT networks. IEEE Internet of Things Journal 2019, 6, (4), 7011-7024.

33. Yan, K.; Shan, H.; Sun, T, et al. Quek, T. Q., Reinforcement learning-based mobile edge computing and transmission scheduling for video surveillance. IEEE Transactions on Emerging Topics in Computing 2021, 10, (2), 1142-1156.
34. Liu, X.; Zhou, L.; Zhang, X, et al. Joint Radio Map Construction and Dissemination in MEC Networks: A Deep Reinforcement Learning Approach. Wireless Communications Mobile Computing 2022, 2022, 4621440.
35. Liu B, Liu C, Peng M. Resource allocation for energy-efficient MEC in NOMA-enabled massive IoT networks. IEEE Journal on Selected Areas in Communications, 2020, 39(4): 1015-1027.
36. Kimura T, Kimura T, Matsumoto A, et al. Balancing quality of experience and traffic volume in adaptive bitrate streaming. IEEE Access, 2021, 9: 15530-15547.
37. Liu, Q.; Huang, S.; Opadere, J, et al. In An edge network orchestrator for mobile augmented reality, IEEE INFOCOM 2018-IEEE conference on computer communications, 2018; IEEE: 2018; pp 756-764.
38. Zhao T, He L, Huang X, et al. DRL-Based Secure Video Offloading in MEC-Enabled IoT Networks. IEEE Internet of Things Journal, 2022, 9(19): 18710-18724.
39. Wu, Z.; Yan, D., Deep reinforcement learning-based computation offloading for 5G vehicle-aware multi-access edge computing network. China Communications 2021, 18, (11), 26-41.