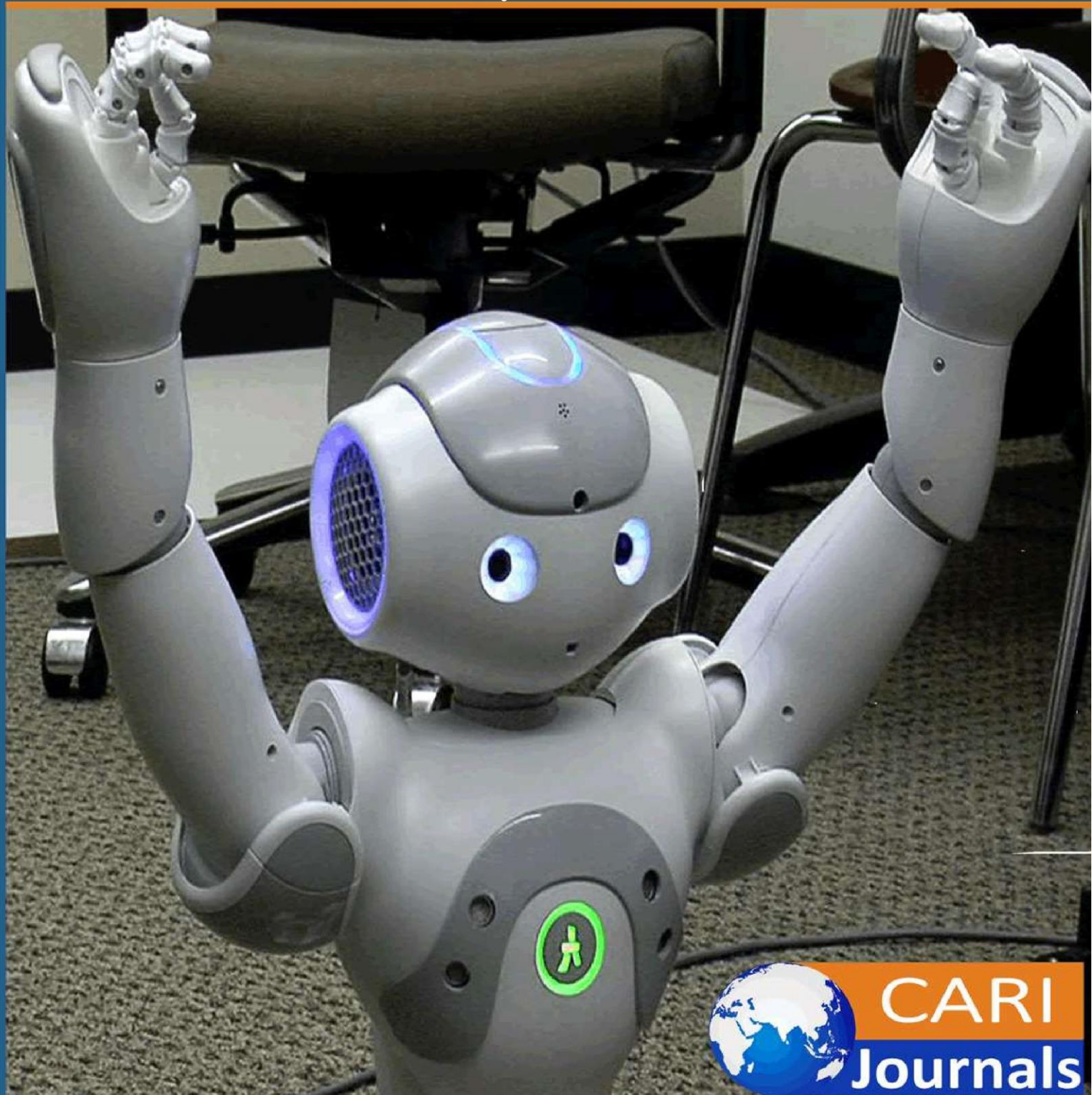


# International Journal of Computing and Engineering (IJCE)

**A Deep Reinforcement Learning Strategy for MEC  
Enabled Virtual Reality in Telecommunication Networks**



**CARI  
Journals**

## A Deep Reinforcement Learning Strategy for MEC Enabled Virtual Reality in Telecommunication Networks

 **Kodanda Rami Reddy Manukonda**

<https://orcid.org/0009-0000-2853-1143>

Accepted: 20<sup>th</sup> Mar, 2024 Received in Revised Form: 5<sup>th</sup> Apr, 2024 Published: 20<sup>th</sup> Apr, 2024

### Abstract

One of the most anticipated features of 5G and subsequent networks is wireless virtual reality (VR), which promises to transform human interaction via its immersive experiences and game-changing capabilities. Wireless virtual reality systems, and VR games in particular, are notoriously slow due to rendering issues. But most academics don't care about data correlation or real-time rendering. Using mobile edge computing (MEC) and mmWave-enabled wireless networks, we provide an adaptive VR system that enables high-quality wireless VR. By using this architecture, VR rendering operations may be adaptively offloaded to MEC servers in real-time, resulting in even greater performance advantages via caching. The limited processing power of VR devices, the need for a high quality of experience (QoE), and the small latency in VR activities make it difficult to connect wireless VR consumers to high-quality VR content in real-time. To solve these problems, we provide a wireless VR network that is enabled by MEC. This network makes use of recurrent neural networks (RNNs) to provide real-time predictions about each user's field of vision (FoV). It is feasible to simultaneously move the rendering of virtual reality material to the memory of the MEC server. To improve the long-term VR users' quality of experience (QoE) while staying within the VR interaction latency limitation, we provide decoupling deep reinforcement learning algorithms that are both centrally and distributedly run, taking into consideration the connection between requests' fields of vision and their locations. When compared with rendering on VR headsets, our proposed MEC rendering techniques and DRL algorithms considerably improve VR users' long-term experience quality and reduce VR interaction latency, according to the simulation results.

**Keywords:** *Wireless Virtual Reality, Mobile Edge Computing, Future Wireless Networks, field of vision (FoV), and Deep Reinforcement Learning*

## I. INTRODUCTION

Every aspect of our daily lives is profoundly affected by intricate and cutting-edge technologies like social media platforms, the IoT, AI, VR, and many others. The way this system works affects many different things that people do. As a result, understanding the dynamics of the system and what they mean for customer feedback is essential. When it comes to data mining, emotion analysis is one of the most popular and effective methods [1]. The need to estimate and show user motion in VR applications is growing in tandem to the popularity of VR games and devices. The majority of virtual reality avatar posture estimate algorithms use online motion capture and inverse kinematics[2]. One exciting new way to interact with VR without using your hands is via speech recognition technology. On the other hand, it does have a few downsides, such the fact that it isn't really accessible to those who have trouble making themselves heard in busy public spaces or in environments with a lot of background noise. Using a silent speech recognizer system in a VR environment using facial electromyograms might help overcome these constraints [3]. With the use of deep learning with VR technologies, the goal is to digitize the grotto paintings' contents in order to preserve them. The grotto paintings are analyzed and modelled using virtual reality technology. The framework is based on data collection technology for cultural assets, and it presents important notions of a digital library. In order to restore and recreate grotto paintings, it is necessary to construct a computer identification model. This model incorporates the color with materialization principles of these murals, as well as the historical relics information collecting technologies under DL and VR [4]. The cognitive capacity to choose, organize, and construct other representations when processing information is known as representational flexibility. Students on the autistic spectrum are underrepresented in STEM fields due to a lack of representational flexibility. Researchers have shown that using VR 3D simulations may help adolescents with ASD improve their RF skills [5]. One promising use case for next-gen mobile networks is VR using wireless technology. But current mobile networks struggle to handle the ultra-low latency and enormous data rates needed for wireless VR. One interesting approach to enabling wireless VR is the multi-access edge computing system, which offers processing and caching capabilities at the network's periphery. However, frequent handoffs may diminish the quality of the experience for mobile VR users [6]. Rather of relying on standard passwords, which may be compromised by hostile actors, VR apps could be protected by using users' motion behavior as a biometric signature. This would allow for continuous authentication and identification of users. Over time, users' behaviors naturally vary, impacting their body movements and the trajectories of virtual reality equipment like the headgear and controllers [7]. Online courses, where students engage in virtual classrooms with little face-to-face contact, have becoming more common [8]. By utilizing a person's motion trajectories as a unique signature, we can implement task-oriented behavioral biometric identification of users interacting in VR environments, which allows for seamless continuous authentication. Behavioral biometrics methods based on deep learning perform better when fed whole or almost full user trajectories, but worse when fed short segments starting from the beginning of the job [9]. Both academics

and businesses are showing an interest in the cutting-edge VR applications that are currently under development. A wide range of everyday activities are the primary focus of wireless virtual reality applications, including intelligent instruction, amusement, healthcare, travel, building design, transportation, and factory automation. The idea of QoE, that describes the service's quality from the perspective of its end users, is strongly tied to these interactive apps that strive to provide consumers with immersive experiences [10].

### A. Contributions

- The MEC-enabled wireless virtual reality (VR) systems that we suggest can upload the VR system's rendering model to the MEC server. They are also able to anticipate the field of view (FoV) of each user in real-time.
- We give a decoupled learning mechanism to enhance the VR user experience over the long run. To improve the two subtasks—field of view (FoV) prediction and portrayal of MEC connection—independently, this technique employs a deep reinforcement learning (DRL) algorithm in conjunction with a Recurrent Neural Network (RNN) predictor.

Thanks to this approach, the suggested technique outperforms the current crop in terms of scalability and adaptability to system dynamics. Compared to the other baseline algorithms, the suggested one performs better in terms of latency, convergence time, and quality of experience utility value, according to the simulation findings.

The rest of the paper is organized like this. In Section II, the subject formulation as well as system model are presented. The proposed procedure is detailed in Section III. In Section IV, we look at the simulation results and draw some conclusions. Section V wraps up the article and delves into possible avenues for further research.

## II. RELATED WORK

Use a deep learning technique to predict the user's upper body position in real time in [11]. Authors present a new approach that uses 3D motion capture information to train, drawing inspiration from a traditional regression model. Based on all the data from the motion capture, our approach employs a CNN (convolutional neural network) that can modify its input and output to accept input from both the head and both hands. One possible use case is creating a consumer's virtual identity inside a VR app after feeding the model typical inputs such as an HMD and two controllers. Finally, they agreed on their proposed pose estimation technique after developing single- and multi-user applications, testing them with actual users, and comparing the results to existing methods for virtual reality avatars. In order to measure functioning EMGs (fEMGs), a small number of electrodes are put around the user's eyes. In their proposal for an SSR device, the researchers of [12] noted how simply it might be integrated into current VR headsets. By analyzing comparable data from other people and modifying it using dynamic positional warping, a method was created to improve the accuracy of fEMG signal classifications. A few of sites far from the phonatory organs were used to capture the data. The proposed SSR system had an impressive 92.53% rate of success in categorizing six separate fEMG patterns generated by six

silently spoken sentences, according to the testing data. We employed an online SSR technologies to further show how their technique works as a hands-free interface in real-world virtual reality applications. The work shown in teaches students to use deep CNN-LSTM classifier to detect and eliminate distractions in a virtual reality (VR) classroom setting [13] utilizes generic characteristics taken from data collected by VR headset sensors (head-tracking, hand-tracking, as well as eye-tracking) such as angular velocity, positional velocity, pupil diameter, and eye openness. Their classifier outperforms two previous methods in terms of accuracy and feature generalizability, and they provide early findings showing an accuracy of 94.93%. By developing a more precise and broadly applicable method for distractor identification, they think their work may contribute to the advancement of educational virtual reality. The goals of that research endeavor were to develop a system for creating and instructing neural networks to address various real-world issues, academic challenges, and educational needs, and to assess the system's practicality and usefulness [14]. That study enlisted the help of five professors with experience teaching ANNs at two different colleges in order to develop a user-friendly framework. A total of 31 students with prior knowledge of neural networks were asked to participate in an online survey regarding the following topics: "the main challenges in learning NNs" as well as "key specifications in a Visual Instruction Tool such as the most desired characteristics of a visualization instrument for explaining NNs" that they would have utilized while taking the course. They also looked at how their pupils would learn about ANNs using that method by doing an observational study. Their framework's ability to generate ANNs with a visual presentation is compatible with AR and VR environments, enabling us to show and control networks in a virtual world. For seamless virtual reality video handoffs, researchers in [15] suggested a proactive caching, processing, as well as communication (3C) resource distribution strategy. More precisely, the 3C capabilities at the BS govern the file size and quality, and the goal base station pre-caches the 3D and 2D video for rendering. They then determine the best approach to proactive 3C allocation of resources using a deep learning reinforcement method that does not rely on models. The numerical data suggests that the proposed technique might provide VR users with high quality of experience when switching between base stations. In [16], I created online virtual classrooms using VRChat. As an added bonus, the authors built a facial expression detection system for virtual reality classes using cutting-edge computing, OpenCV, with Python since they thought it would be crucial to know the the pupil's emotional state when studying VR online. Using real-time face recognition to compare VRChat's VR lessons to other traditional online teaching techniques and to verify their effectiveness, they verified VRChat's VR courses. Using face recognition and surveys in the virtual reality online class helped students concentrate and stay on task for longer periods of time. Applying deep learning in accordance with the procedures outlined in [17] will allow for the objective prediction of QoE. The prediction model investigates the matter by studying how the functioning of wireless networks affects the quality of virtual reality 360-degree streaming video. It uses LSTM (long short-term memory) neural systems for encoding and decoding, and it uses only quantifiable quality of service metrics to provide real-time predictions on the total transmission-related quality of experience value. They

put the prediction model through its paces on a testbed for open radio access networks that adheres to the O-RAN standards for interfaces. Findings from pilot projects using Fiber Optic supplemental learning modules are the focus of the research reported in [18]. Scenario planning, low poly optimization 3D asset production, scenario integration, triggering each interaction, multiuser environments, and implementation are the several steps that make up the module development process. Measurements are taken using OVR metrics to construct a solid VR Fiber Optic application. The purpose of that metric is to ascertain the application's dependability in relation to the user experience. With an OVR metric measurements, the frame rate may go as high as 43 and as low as 27. The created value can be useful for virtual reality apps with a low poly asset design. Customer satisfaction is also measured using the PIECES Framework. Users had a positive impression of the VR FO programme as an e-learning module, according to the results of the measurements, which range from 3.4 to 4.91 on each variable. Using brain wave studies that tracked how subjects' emotions altered in response to changes in ceiling height, the investigators in [19] were able to calculate the stress score for each building in the region. They utilized VR to build a replica of the Reich Chancellery so that they could conduct two experiments using the EEG data acquired in each room. They started by looking at the most active region and figuring out the beta wave variations relative to the stress indexes. Our results were double-checked and areas with a high stress index were identified using deep-learning algorithms after the authors investigated the association between ten distinct brain wave types and waveforms. To calculate such a value, a deep-learning framework was subsequently developed. The purpose of that research is to provide the outcomes of the pilot program for FO instructional modules [20]. Scenario planning, low poly optimization 3D asset production, scenario integration, triggering each interaction, multiuser environments, and implementation are the several steps that make up the module development process. Measurements are taken using OVR metrics to construct a solid VR Fiber Optic application. The purpose of that metric is to ascertain the application's dependability in relation to the user experience. With an OVR metric measurements, the frame rate may go as high as 43 and as low as 27. Virtual reality applications that utilize low poly asset architecture may operate well on the generated value. Customer satisfaction is also measured using the PIECES Framework. Users had a positive impression of the VR FO programme as a virtual instructional module, according to the measurement results (3.4–4.91) for each variable. The attentional tendencies of English language learners processing online tasks are investigated in a research that employs virtual reality and deep learning methods [21]. The results of the studies illustrate the correlation between the linear attentional management model and the performance of young second language learners on online tasks, providing a visual representation of the influence of that model on task completion. Additionally, the model's regression and prediction accuracy is rather excellent. The article presents the outcomes of eight of the most prevalent physical training exercise situations used in post-stroke rehabilitation approach [22]. While completing various rehabilitation activities, depth sensors reliably detected critical aspects of the subject's posture. The system is suitable for usage in real-time applications because to its average reaction time of 23 ms. In addition, the system's skeletal

feature extraction is helpful for differentiating between those with normal skeletons and those with lower back discomfort. Their findings provide credence to the idea that the motion recognition-based system can monitor rehabilitation progress, evaluate the efficacy of the patient's physiotherapy activities, and more. In their earlier study, researchers used 6LU7, the first publicly available Mpro crystal structure [23]. Using an AI-driven generating chemistry technique, Insilico Medicine unveiled the first new protease inhibitors on February 4, 2020. The number of X-ray structures found in co-crystallized Mpro with different ligands, some of which are covalent and others of which are non-covalent, has surpassed a hundred since then. Using the newly-disclosed 6W63 atomic structure of Mpro complexed along with a non-covalent inhibitor, they integrate the ligand-based with crystal structure-based methods used in our older investigation. In order to spark lively discussions and thorough analysis among medicinal chemists, 10 sample structures with three-dimensional representations in PDB format have been published for potential elaboration. The efficiency of e-learning and the variables and circumstances leading to DML in distant mode higher education utilizing social virtual reality environment are the main topics of the [24] literature study. That led researchers to compile the results of 33 empirical investigations on HE that took place between 2004 (when VR first appeared) through 2019 (before coronavirus emerged) into a comprehensive literature evaluation. They studied the relative relevance of DML's emotional, social, as well as cognitive components in SVREs using a research framework. The results demonstrate that SVREs may provide cognitively demanding simulations that are both realistic and personally relevant, as well as opportunities for active interpersonal as well as cooperative learning. In that research, the authors want to utilize a machine-deep-ensemble learning approach to categorize cases of cybersickness brought on by VR immersion [25]. Twenty participants watched a virtual reality film for around five minutes while having their heart rate variability and respiratory signal characteristics monitored. Following the experiment, participants were assessed for CS by having them answer questions designed to gauge their current CS state. In order to determine and categorize the users' VR immersion CS, a machine-deep-ensemble learning framework was built utilizing the data. In order to generate prediction data, the ensemble model classified the data using a convolution neural network. The model included a number of stacking machine learning methods, including AdaBoost, k-nearest neighbor, support vector machine, and random forest. Subjects' CS may be categorized as neutral, non-CS, nor CS using the model's multiple classes of skills.

### III. PROPOSED WORK

#### A. System Model

As seen in Figure 1, we take into account a wireless virtual reality system in which several MECs are linked to a centralized control system via a fiber link to supply VR users with wireless connections. The main controller is able to get undistorted, real-time 2D movies from the main network thanks to its fiber connection. As stated in [29], the core network may transform 2D pictures into VR films by sending them to various edges. Hence, we take it as read that the necessary 2D pictures obtained from Edges / UAVs or additional equipment are instantly

accessible to the core network. We also assume that the MEC is limited to rendering for the same field of view (FoV) for all VR users in time slot 1.

Virtual reality gaming is the main topic of this article. The standard components of a virtual reality system are a headgear, controller, and renderer. Wearing the headset has a dual purpose: first, it displays virtual reality (VR) material to the user; second, it tracks their posture, which includes their three-dimensional (3D) location and orientation inside the virtual environment. Physical buttons, touchpads, and sensors make up the controller, which is responsible for receiving user inputs apart from user position. For virtual reality systems with social components, these interactions could originate from other users online. When the user changes their position or interacts with the environment, the renderer generates fresh frames that are shown on the HUD. Models are representations of 3D objects or virtual environments specified by a computer language or data structure; rendering is the process of creating images from these models. We assume in this work that the relevant models are installed on both handheld gadgets as well as MEC servers, which opens up the possibility of local, distant, or cooperative rendering. Typically, virtual reality (VR) material may be categorized into two parts: interactions in the front and the virtual world in the backdrop. Virtual reality games often employ real-time rendering, in contrast to other popular VR applications that display immersive films. However, two aspects are noteworthy: 1) the foreground interactions in VR games are more complex and unexpected, but the rendering burden is relatively low; 2) the backgrounds in VR games are more lifelike and immersive, but the processing workload is huge because of all the rich features and complicated textures. Resolution, frame rates, and processing power are common user-perceived experience parameters. There are three different ways that virtual reality (VR) material may be shown, allowing users to cater to different computer requirements and resolution requirements: 1) Rendering locally on handheld devices, 2) Rendering remotely on MEC servers, as well as 3) Rendering cooperatively, whereby background and foreground environment are displayed on MEC server. A head-mounted display (HMD), a server, and a client make up the majority of the system.

- **Edge Server:** We make use of a VoIP server with a Netcode server as our tools. Netcode is one concept in video game development. In our network, the Netcode server is responsible for controlling the shared scene state, keeping everyone's progress synchronized, and show an accurate as well as dynamic portrayal of the room's current state. The VoIP server regulates the transmission of audio packages across the network. It is responsible for receiving and distributing voice packets from each client.
- **Customer or End-User:** The client, which is usually a powerful computer, is responsible for gathering microphone input from the HMD, obtaining the pulse coding modulated (PCM), compressing the PCM packets, and sending them to the VoIP server. At the same time, it decompresses the PCM packets received by other clients and processes them using the spatial audio algorithms [50] so that they may be played back with room reverberation as well as direction. The client changes its location in relation to other



clients' positions that it has received from the Netcode servers and synchronizes its own position with the server.

- HMD: The client receives real-time data about the user's physical state and interactions while they encounter the virtual world via a head-mounted display.

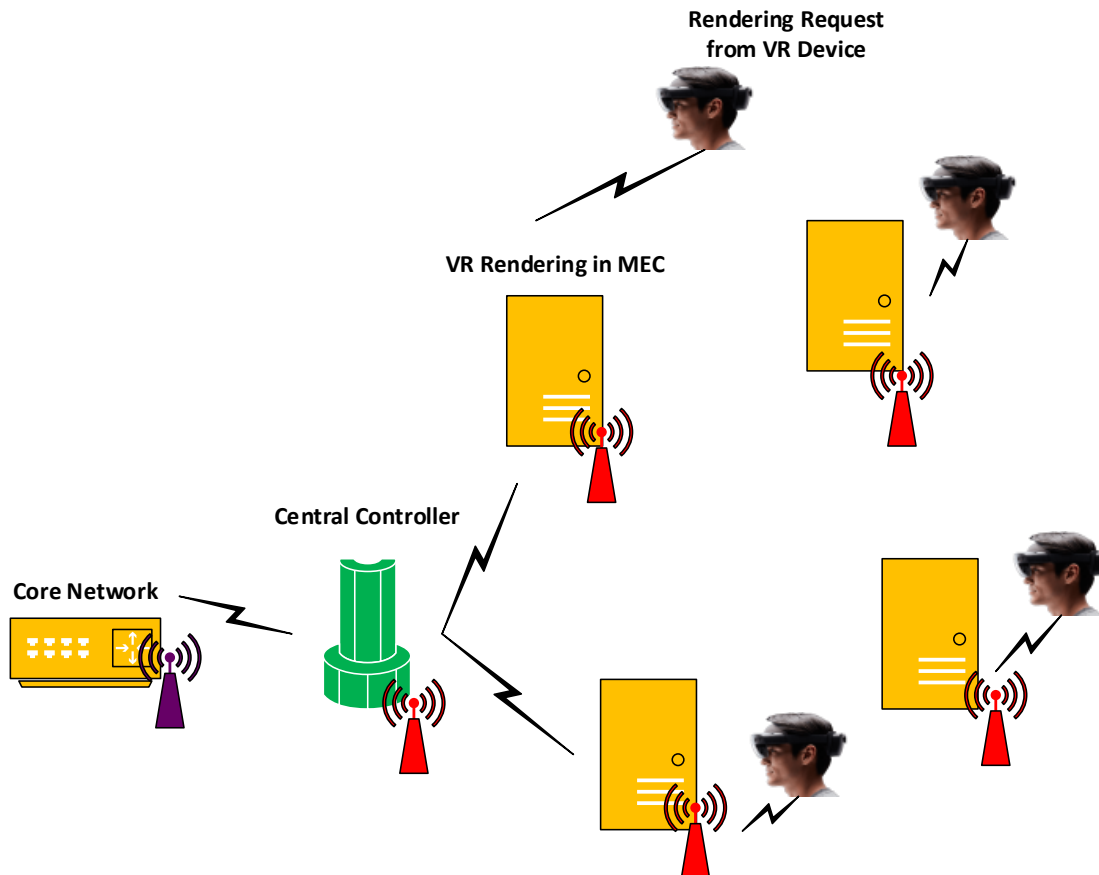


Fig. 1. Wireless VR system in cellular network

We take into account a cellular network that is mmWave/sub-6 GHz based and has MEC capabilities, as shown in Figure 1, and it serves a group  $\mathcal{U} = \{1, 2, 3, \dots, i, \dots, U\}$  of  $U$  handheld virtual reality headsets  $\mathcal{B} = \{1, 2, 3, \dots, j, \dots, B\}$  of  $B$  base stations (BSs), meaning any BS with a MEC server present. The principal use of VR in this study is engaging in an exciting game. At the outset, we give you a quick rundown of what the system can do. The next sections go on to discuss the mathematical foundations of computing, communication, and caching.

## B. QoE Model

Here, we establish a quality-of-experience (QoE) utility metric for measuring VR users' QoE, with latency and energy usage provided as inputs.

Prior to anything else, you will find the clients' energy utilization and delay. According to user  $i$ , the perceived latency is

$$T_i = I_{\{o_i=0\}} \cdot T_i^l + I_{\{o_i=1\}} \cdot T_i^{\text{remote}} + I_{\{o_i=2\}} \cdot T_i^{\text{coop}}.$$

Likewise, user i's energy usage may be determined by

$$E_i = I_{\{o_i=0\}} \cdot E_i^l + I_{\{o_i=1\}} \cdot E_i^{\text{remote}} + I_{\{o_i=2\}} \cdot E_i^{\text{coop}}.$$

There are two better experience criteria that gamers may use to guarantee they are totally engrossed in virtual reality: (i) For the whole reaction time to be shorter than  $\tau$ , say, 25 milliseconds, two conditions must be met: one example of a framerate that exceeds  $\eta$  PFS is 60 PFS. Another consequence of these two prerequisites is little frame-to-frame jitter. We calculate the QoE utility for VR users for a certain time period and find that it is:

$$Q_i(\mathbf{V}, \mathbf{A}, \mathbf{O}) = I_{\{T_i(t) \leq \min(\tau, \frac{1}{\eta})\}},$$

that records the outcome regardless of the material  $r_i$  satisfied all of the requirements in (18) and was sent to user  $i$  at the moment  $t$ .

Over the course of  $T$  time slots, we examine the VR transmission. As a result, we establish the following measure to describe the VR user's QoE.

$$I_i(\mathbf{V}, \mathbf{A}, \mathbf{O}) = \sum_{t=0}^T I_{\{T_i(t) \leq \min\{r, \frac{1}{\eta}\}\}},$$

This indicates the dependability of user  $i$ 's transmissions over time.

### C. Duel DQN Model

**Initialization:** The many simulation scenarios, including the cellular network's virtual reality system, are put up during the setup phase. The weights as well as replay memory, among other Q-network components, are also initialized  $\theta$  and  $\theta^-$ .

**Double-Dueling-Deep Q Network:** We apply double-dueling DQN in this work. The suggested problem's high-dimensional action space as well as state space necessitates the usage of DQN to approximate the value function. Two methods are used to enhance DQN's performance: double DQN as well as dueling DQN.

Overestimations occur in the conventional DQN because applying the max to the already erroneous Q values causes a bigger approximation error. The concept behind double DQN is to separate the selection process from the assessment process in order to avoid the overestimation issue. A mathematical expression for it would be

$$y_t = r_t + \epsilon Q(s_{t+1}, \arg \max Q(s_{t+1}, a_t; \theta_t); \theta_t^-)$$

Through the use of online weights to choose the activities  $\theta$ , and making predictions using the present values  $\theta_t^-$ , It has the potential to improve performance by providing more precise estimates of Q values.

In reality, there are cases when the environmental impact of some governments' activities is negligible. Thus, it is superfluous to try to guess the worth of every single action for a bunch of different situations. Part of the Q value in dueling DQN is the benefit of taking actions in a condition  $s$ , and part of the Q value is the value of being in state  $s$ . A brief summary of rival DQNs is provided below:

$$Q(s, a) = A(s, a) + V(s).$$

Because not every action affects the state while in certain states, battling DQN may speed the training procedure to some degree.

When deciding what to do, you may use one of these approaches. In this article, we employ the  $\epsilon$ -greedy method, which randomly picks an action from a set of options having a likelihood of  $\epsilon$  and selects the one with the highest Q value to investigate— $\epsilon$ , (i.e., exploitation).

**Replay Memory:** Non-stationary distributions may be caused by strongly correlated states. To lessen the severity of these effects, DQN employs a replay memory method, where the experience tuple  $e_t = (s_t, a_t, r_t, s_{t+1})$  is kept. For training purposes, it smooths out distributions across a large number of past occurrences, allowing for a more random sampling of past experiences rather than only the present.

The target network's architecture is indistinguishable from the main Q network's. The balances of the target network are copied from the primary network each G steps to prevent becoming caught in feedback loops among the two networks.

Minimizing the loss functions is an important part of training for the target network.

$$L(\theta) = E[(y_t - Q(s_t, a_t; \theta_t))^2]$$

It makes use of the gradient descent approach. The next step, after initiation, is training. The  $\epsilon$ -greedy approach is used to pick an action once the environment is observed. Subsequently, the fresh experience is committed to replay memory following action execution and observation of reward. Finally, the target network is eventually reached by training the main Q networks to minimize the loss operation, with the target network being revised every G steps. The likelihood of exploration and the velocity of learning  $\alpha$  may change throughout the training session.

Each user in the reinforcement learning framework acts as a smart agent in time slot  $t$ , responding to state and rewards. The system advances to time slot  $t+1$  and the user modifies its state once actions are taken in accordance with policy. Below, you can see the state of our model: Since each individual in the VR networks decides on their own strategy, (1) defines the whole set of states in the network.

**Action:** Virtual reality's whole collection of user actions is described as  $\mathbf{A}(t) = \{a_1(t), a_2(t), \dots, a_N(t)\}$  in time slot  $t$ , where  $a_i$  does the deed pertain to the state.

Policy: We define the likelihood of user  $i$ 's strategy selection in time slot  $t$  as  $\pi_i(t)$ . The policy set  $(t) = \pi_1(t), \dots, \pi_N(t)$ . Every iteration begins with the adoption of actions based on the probability policy. Here is how the transition functions is defined using the strategy selection probability:

$$P(s, s', a) = P(s(t+1) = s' \mid s(t) = s, a(t) = a).$$

Reward: Defined in (13), the reward system is dependent on the actions and states of all users of VR in the network. in addition, we indicate  $r(s^t, a^t)$  as the  $R_{\text{sum}}(s, a)$  to user  $i$  at interval  $t$  moments in time.

#### D. VR Rendering using MEC-DQN

instead of user  $i$ 's rendering work, it is  $I_i = \{C_i^b, C_i^f, D_i^f, q_i, r_i\}$ , where  $C_i^b$  gives the amount of background CPU execution required to render,  $C_i^f$  displays the amount of processing power required to display the interactions in a prominent manner, The amount of data needed for front-end interactions is represented by  $D_i^f$ , while the amount of data needed for a single VR frame, or the necessary pixel density, is denoted by  $ci$ .  $r_i \in \mathcal{C}$  marks the information that the person I have requested possesses. Just to review, there are three methods that are taken into account while displaying the picture frames for virtual reality users: local, remote, as well as cooperative. A set of mode choice indications is defined by us as  $\mathbf{O} = \{o_i, i \in \mathcal{U}\}$ , where there is  $o_i \in \{0, 1, 2\}$ . When there is  $o_i = 0$ , regional analysis with an emphasis on consumers. If  $o_i = 1$ , If  $o_i = 2$ , then users  $i$  is participating in cooperative rendering, and else, user  $i$  is rendering remotely.

1. Local Rendering: Here, the front and rear should be rendered simultaneously by the mobile device. Assumption number one: the mobile device's processing power is  $z_l$  (cycles/seconds). Hence, the delay starting from the frame's request is

$$T_i^l = \frac{C_i^b + C_i^f}{z_l}$$

The amount of power needed for rendering on-site is

$$E_i^l = \xi T_i^l$$

where  $\xi$  The calculation capability's energy consumption unit is joules/seconds  $z_l$ .

2. Remote Rendering: Here, MEC servers are used to carry out the rendering process. Several data-processing processes are usually required. 1) Before proceeding, be sure you give the relevant BS with the necessary background interaction facts. 2) Lastly, check whether the cache already has the background content you requested. Hence, you need to have the interactions happen in the foreground followed by involve them into the final frame. Create the frame by combining the available models with the provided interaction data. In order to compress the shown frame, the third step is to encode it. Fourth, make the encrypted data available to the consumers. Mobile devices need frame decoding (5).

Because users' decoding ability could vary, it is critical to retain the uncompressed background settings. When it comes to processing power, how effective are MEC servers? The frequency  $z_c$  is measured in hertz/s. The resultant delay from remote rendering is

$$T_i^{remote} = T_i^{tr,f} + T_i^{re} + T_i^{en} + T_i^{tr,d} + T_i^{de}$$

where  $T_i^{tr,f} = \frac{D_i^f}{R_{ij}^u}$  show how long it takes for an uplink transfer to complete, The MEC servers calculate the rendering time, which is represented as  $T_i^{re}$ .

$$T_i^{re} = \begin{cases} \frac{C_i^f}{z_{c,i}} + T_{i,c}^{in}, & \text{The required environment is cached,} \\ \frac{C_i^b + C_i^f}{z_{c,i}}, & \text{otherwise,} \end{cases}$$

where  $T_{i,c}^{in} = \frac{k(q_i)}{z_{c,i} \cdot b_{zc}}$  engage the final frames of the MEC server design, where  $kci$  is the quantity of data that requires processing as well as  $b_{zc}$  is the amount of data that can be handled in one cycle through using the central processing unit.  $z_{c,i} = \frac{z_c}{\sum_{i \in U} \alpha_{ij} I_{o_i \neq 0}}$  everything that every user associated with BS  $b$  has access to in terms of computer resources, where  $\mathbf{U}\{\cdot\}$  performs an indicator operation, which is described as

$$I_{\{\cdot\}} = \begin{cases} 1, & \text{if } \{\cdot\} \text{ is true,} \\ 0, & \text{otherwise.} \end{cases}$$

$T_i^{en} = \frac{f(q_0)}{z_{c,i} \cdot b_{zc}}$  stands for the MEC server's compression latency, where  $c_0$  is the original frames' resolution and  $b_{zc}$  is the calculated data size (in bits) in a single central processing unit cycle. As said before,  $f(q_0)$  is a function of  $q_0$ , in order to compress each frame with the resolution, the data volume must be determined  $q_0$ .  $T_i^{tr,d} = \frac{g(q_0)}{R_{ij}^d}$  indicates the time it takes for the compressed frames to be sent downlink, where  $g(q_0)$  indicates the amount of data after compression and is an expression of  $q_0$ . And  $T_i^{de} = \frac{h(q_i)}{z_l \cdot b_{zl}}$  is the amount of data calculated (in bits) in one CPU cycles on the mobile device, and it stands for the decoding delay for the compressed frames. Importantly,  $ci$  is a measure of  $ci$ , which stands for the quantity of data that has to be handled in order to decode the frames. Collaborative Drawing: At this point, the MEC servers render the background environment, while the mobile device renders the interactions in the foreground. In order to complete the rendering process, it is necessary to follow various conventional steps. We immediately separate the rendering process into two parts: interactions in the front and surroundings in the background. The local CPU of the mobile device is used to render foreground interactions. You should check the cache for the backdrop items you want to use before

interacting with them. Compressing and delivering the items to users is done after checking whether they are previously cached. In any case, you may use the MEC server as a tool to generate the backdrop scenes. Send the user the background material that has been rendered and compressed. Concurrently, the MEC server decides whether or not to store the updated background data in a cache. 3) After the foreground interaction processing is finished and the background is received, it compresses the background content. Then, the mobile device interprets the background material and merges the final frames. Hence, the overall rendering delay for cooperative tasks may be described as

$$T_i^{coop} = \max\{T_i^{f,re}, T_i^c\} + T_i^{in}$$

where  $T_i^{f,re} = \frac{C_i^f}{z_i}$  represent the amount of time required by mobile devices to determine the interactions in the foreground, and  $T_i^{in} = \frac{k(q_i)}{z_i \cdot b_{z_i}}$  is the duration that images captured on mobile devices take to fully load.  $T_i^{z^2}$  This is calculated as the time it takes for the MEC server to process requests.

$$T_i^c = v_{r_i,j} T_i^{b,re} + T_i^{en} + T_i^{tr,d} + T_i^{de}$$

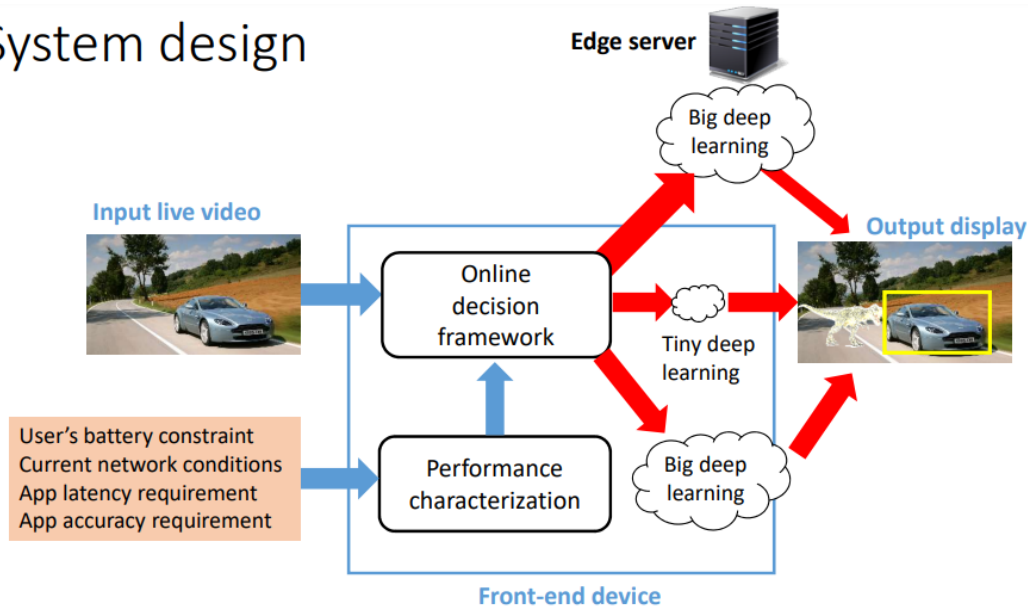
where  $v_{r_i,j}$  is signifies that the requested content for user  $i$  has been cached, and  $T_i^{b,re} = \frac{C_i^b}{z_{c,i}}$  shows the time required for a MEC server to generate the backdrop. With remote rendering, everything else remains unchanged.

In terms of power use, it may be represented as

$$E_i^{coop} = E_i^{f,re} + E_i^{de} + E_i^{in}$$

where  $E_i^{f,re} = \xi T_i^{f,re}$  is how much power the mobile device needs to exhibit the interactions in the forefront  $E_i^{de}$  has the same value as the energy required for decoding, as shown in equation (12). The energy needed to consolidate the final picture is believed to remain constant in this study because to its much reduced energy consumption compared to rendering and decoding.

## System design



24

Fig.2. Proposed VR with MEC Model

When playing in the same virtual reality (VR) environment, it's conceivable that various users would see comparable material since different those who use VR may need to generate the same 3D models. This provides an opportunity to further improve the QoE for VR users. Therefore, we will assume, based on data correlation and user popularity, that the created contents may be stored in the MEC servers. Unlike previous efforts that concentrated on transmitting viewport tiles, this article considers delivering high-quality panoramic films to VR users, as they may still suddenly alter the orientation of the device. The panoramic image size is drastically reduced via compressing on the MEC server hardware in order to prevent significant transmission delay. Mobile edge computing (MEC) servers and end users get material over mmWave along with sub-6 GHz cellular networks. It is presumed that users may establish simultaneous connections to both the mmWave as well as sub-6 GHz lines. Uplink interaction exchange of data often occurs in the sub-6 GHz region, whereas downlink rendering content transfer occurs in the mmWave channel. This study makes use of the radio network's information service (RNIS), which may gather data on the radio network as a whole as well as specifics about the users whose devices are linked to the radio nodes. The MEC servers routinely assess the user's device power and the quality of their wireless sub6 GHz connections using RNIS. To test the mmWave downlinks' quality, users often broadcast uplink auditory references over the whole angular space [25]. The users choose the offloading mode and associate themselves with the service, while the MEC servers make caching choices to improve the quality of experience for virtual reality users based on data about the link's quality, cached data, and the computing resources available on the servers.

## IV. RESULTS & DISCUSSION

### A. Simulation Setting

Within a circular area with a radius of 100 meters, the cell phone network is situated, and users' base stations are scattered randomly throughout this area. Given these hypothetical circumstances, every single one of this is taken into account. To better understand the layout of a game, think of a virtual square that is 100 pixels long. Except where specified, all supplementary data pertaining to simulation parameters may be found in Table II. Our simulations are executed on a GPU-based server that includes four NVIDIA GTX 1080 TI GPUs, 128 GB of RAM, and an Intel Xeon CPU. This configuration makes use of TensorFlow 1.8.0, Ubuntu 18.04 LTS, and Python 3.6. Our approach incorporates a target network and a primary network, two deep neural networks, into the training models. Four fully-connected layers are used in each neural network. The first three levels contain 64, 64, and 32 neurons, correspondingly. Thus, the idea behind Dueling DQN is to split the 4th neural layer in half, with one half serving as the Value function (state value) and the other half as the Advantage function (action advantage). The dueling design should be able to determine which states are acceptable (or undesirable) without having to commit every possible action to memory. Finally, by combining the Advantage with Value functions, a Q-value is learnt using a loss-with-back-propagation approach. The  $\epsilon$ -greedy strategy, where  $\epsilon$  is a value between 0 and 0.9, is used in the Double-Dueling DQN. We start our investigation with an estimated final probability of 0.1. From a dataset of 10,000 experiences, a mini-batch of 32 examples is selected at random from the experience's replaying buffer to ascertain the loss during training. Typically, gradient descent is used while attempting to reduce the loss function. To no one's surprise, it flips the 5OL(Q) objective function's slopes such that they affect the neural network's  $\Theta$  weights in the other way. With a learning speed of 0.0001, we can calculate the update step size.

#### Power Usage and Latency

Two concurrently processing pipelines handle each request. The processing delay, as seen from the MEC server's point of view, is comprised of calculation and transmission latencies, and it may be expressed as  $T^M(t) = D_{3D}^{\text{down}}(t)/R_1(t) + T_{\text{com}}^M(t)$ . The local VR device's latency may also be expressed as  $T^L(t) = D_{2D}^{\text{down}}(t)/R_1(t) + T_{\text{com}}^L(t)$ . Consequently, the sum of service delay is expressed as  $T^{\text{total}}(t) = \max\{T^M(t), T^L(t)\}$ . Additionally, the definition of system the use of energy is  $E^{\text{total}}(t) = E_M(t) + E_L(t) + E_T(t)$ . We conclude by defining the total system cost at the time  $t$  as depending on the weighted-sum technique, which is often used for multifaceted tradeoff [7], [8].

$$Y(t) = \omega T^{\text{total}}(t) + (1 - \omega)E^{\text{total}}(t),$$

It takes into account the weighting factor the relative significance of delay and energy usage  $\omega \in [0,1]$ .



Table.1. Users Performance Analysis

<b>Participant</b>	<b>Response Time (ms)</b>	<b>Task Success Rate (%)</b>
P1	15	99.1
P2	14	99.2
P3	31	99.4
P4	31	99.6
P5	42	99.45
P6	34	99.25
P7	37	99.45
P8	41	99.56
P9	43	99.75
P10	45	99.12
Average	47	99.15

Table.2. CPU Time with Memory Usage

<b>Participant</b>	<b>CPU Time (ms)</b>	<b>Memory Usage (AIB)</b>
P1	123	256
P2	101	312
P3	138	289
P4	115	275
P5	146	302
P6	125	281
P7	110	295
P8	135	309
P9	128	290
P10	122	297
Average	123.2	290.6

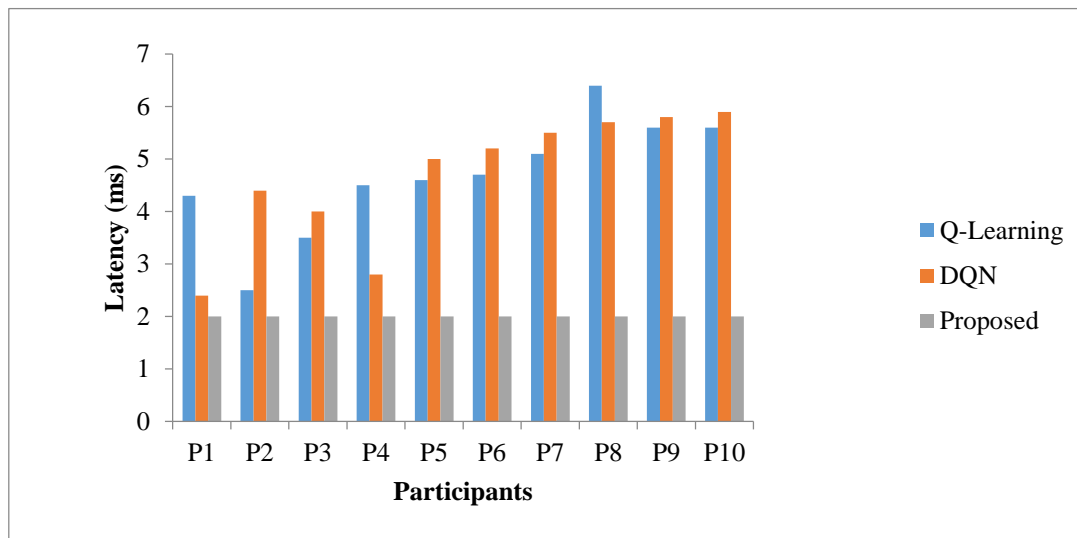


Fig.3. Latency vs. Participants

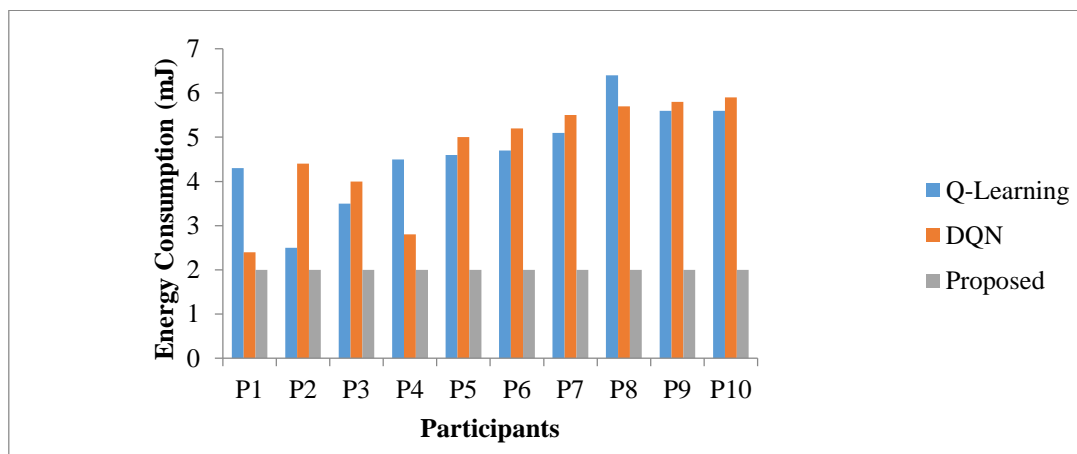


Fig.3. Energy Consumption vs. Participants

The assessment of each participant (P1-P10) was conducted by observing their response time in milliseconds (MS) and proportion of successful task completion. Participants' time spent completing the VR tasks may be observed using the response time. Reaction times varied across the individuals, between 320 ms to 430 ms. The individuals' average response time was 370.7 milliseconds. These kinds of values reveal the potential speed and efficiency with which individuals can do the tasks assigned to them. How many people were able to complete every one of the tasks in the task is called its success rate. Respondents were able to finish the tasks inside the virtual reality environment, as shown by success rates ranging from 99% to 100%. On average, 99.8% of participants were able to finish the assignments. In terms of response speed and task completion frequency, the results show that the participants fared well. The average response time of 30.7 ms demonstrated that participants had high cognitive processing and decision-making abilities in the VR environment. The recommended tasks were obviously executed with remarkable competence and efficiency, as the average success rate was 99.8%.

## V. CONCLUSION

We offer an adaptive VR design in this paper to enable high-quality wireless VR when mmWave-enabled wireless connections with MEC (mobile edge computing) are established. With this solution, adaptive outsourcing of actual time VR rendering workloads to MEC servers becomes a reality. Additionally, caching further enhances efficiency. Given the importance of low latency and QoE in VR interactions, as well as the limited processing power of VR devices, users of wireless VR face difficulties in delivering a seamless connection and high-quality VR video in real-time. We have developed a wireless VR network that is enabled by MEC to address these concerns. Each user's field of view (FoV) is predicted in real-time using this network's Recurrent Neural Network (RNN). It is also possible to move the VR device's processing capability to the MEC servers. To enhance the long-term durability of experience (QoE) for VR users under the VR interaction latency restriction, our centralized and dispersed decoupling deep reinforcement learning (DRL) algorithms take into consideration the correlation of geographical with field-of-view (FoV) requests. The simulation findings demonstrate that as compared to generating on VR devices, our suggested MEC rendering approaches and DRL algorithms significantly enhance the experience quality for VR users over the long run and decrease VR interaction latency.

## REFERENCES

- [1] Oh, J., Bong, C.S., & Kim, J. (2019). Design of Immersive Walking Interaction Using Deep Learning for Virtual Reality Experience Environment of Visually Impaired People. *Journal of the Korea Computer Graphics Society*.
- [2] Gu, Q., & Zhang, L. (2023). Deep Learning Analysis of Virtual Reality Technology for pharma industry. *Journal of Commercial Biotechnology*.
- [3] Zhu, C. (2023). Hidden Markov Model Deep Learning Architecture for Virtual Reality Assessment to Compute Human-Machine Interaction-Based Optimization Model. *International Journal on Recent and Innovation Trends in Computing and Communication*.
- [4] Liu, X., & Deng, Y. (2020). Learning-Based Prediction, Rendering and Association Optimization for MEC-Enabled Wireless Virtual Reality (VR) Networks. *IEEE Transactions on Wireless Communications*, 20, 6356-6370.
- [5] Lee, L., Cheung, S.K., Wang, F.L., Chui, K.T., Fung, Y., Lu, A., Hui, Y.K., Hao, T., Hou U, L., & Wu, N. (2023). Design of Serious Games for Blended Learning: Virtual Reality or Augmented Reality? *2023 International Symposium on Educational Technology (ISET)*, 210-213.
- [6] Munroe, L., Sajith, G., Lin, E., Bhattacharya, S., Pushparajah, K., Simpson, J.M., Schnabel, J.A., Wheeler, G., Gómez, A., & Deng, S. (2021). Automatic orientation cues for intuitive

- immersive interrogation of 3D echocardiographic images in virtual reality using deep learning. *European Heart Journal - Cardiovascular Imaging*.
- [7] Zhang, H., & LiNa, Z. (2022). Investigation on the Use of Virtual Reality in the Flipped Teaching of Martial Arts Taijiquan Based on Deep Learning and Big Data Analytics. *Journal of Sensors*.
- [8] Cui, L., Zhang, Z., Wang, J., & Meng, Z. (2022). Film Effect Optimization by Deep Learning and Virtual Reality Technology in New Media Environment. *Computational Intelligence and Neuroscience*, 2022.
- [9] Andrade, S.A., Nunes, F.L., & Delamaro, M.E. (2023). Exploiting deep reinforcement learning and metamorphic testing to automatically test virtual reality applications. *Software Testing*, 33.
- [10] Lee, S., Chen, D., Chigullapally, N., Chung, S., Lee, A.L., Ramos, A., Shravah, V., Rico, T., Youn, M., & Nguyen, D. (2022). The Future of Virtual Reality and Deep Learning in Visual Field Testing. *Emerging Advancements for Virtual and Augmented Reality in Healthcare*.
- [11] Anvari, T., Park, K., & Kim, G. (2023). Upper Body Pose Estimation Using Deep Learning for a Virtual Reality Avatar. *Applied Sciences*.
- [12] Cha, H., Chang, W., & Im, C. (2022). Deep-learning-based real-time silent speech recognition using facial electromyogram recorded around eyes for hands-free interfacing in a virtual reality environment. *Virtual Reality*, 26, 1047 - 1057.
- [13] Khokhar, A., & Borst, C.W. (2022). Towards Improving Educational Virtual Reality by Classifying Distraction using Deep Learning. *ICAT-EGVE*.
- [14] Kiraly, R., Kiraly, S., & Palotai, M. (2023). Investigating the usability of a new framework for creating, working and teaching artificial neural networks using augmented reality (AR) and virtual reality (VR) tools. *Education and Information Technologies*.
- [15] Chen, W., Song, Q., Lin, P., Guo, L., & Jamalipour, A. (2021). Proactive 3C Resource Allocation for Wireless Virtual Reality Using Deep Reinforcement Learning. *2021 IEEE Global Communications Conference (GLOBECOM)*, 1-6.
- [16] Murauchi, K., Sone, J., Yamada, K., & Yasuda, Y. (2022). Study of VR Online class for education with estimating facial expressions using deep learning (System construction and first step experiment of VR online class). *International Journal of Engineering and Artificial Intelligence*.
- [17] Kougioumtzidis, G., Vlahov, A., Poulkov, V.K., Lazaridis, P.I., & Zaharis, Z.D. (2023). Deep Learning-Aided QoE Prediction for Virtual Reality Applications Over Open Radio Access Networks. *IEEE Access*, 11, 143514-143529.

- [18] Sukaridhoto, S., Fajrianti, E.D., Haz, A.L., Budiarti, R.P., & Agustien, L. (2023). Implementation of Virtual Fiber Optic Module Using Virtual Reality for Vocational Telecommunications Students. JOIV : International Journal on Informatics Visualization.
- [19] Ji, S., Kang, S.Y., & Jun, H. (2020). Deep-Learning-Based Stress-Ratio Prediction Model Using Virtual Reality with Electroencephalography Data. Sustainability.
- [20] Sukaridhoto, S., Fajrianti, E.D., Haz, A.L., Budiarti, R.P., & Agustien, L. (2023). Implementation of Virtual Fiber Optic Module Using Virtual Reality for Vocational Telecommunications Students. JOIV : International Journal on Informatics Visualization.
- [21] Zhao, Y., & Liu, S. (2022). A Deep Learning Model with Virtual Reality Technology for Second Language Acquisition. Mobile Information Systems.
- [22] Maskeliūnas, R., Damaševičius, R., Blažauskas, T., Canbulut, C., Adomavičienė, A., & Griškevičius, J. (2023). BiomacVR: A Virtual Reality-Based System for Precise Human Posture and Motion Analysis in Rehabilitation Exercises Using Depth Sensors. Electronics.
- [23] Alex, Z., Bogdan, Z., Alexander, Z., Vladimir, A., Victor, T., Quentin, V., BezrukovDmitry, S., Daniil, P., Rim, S., Andrey, F., Michael, B., Steve, M., Edgardo, L., Deborah, B., Keita, F., Yen-Chu, L., Shih-Hsien, H., Hsuan-Jen, L., Alex, A., & Yan, I. (2020). Potential Non-Covalent SARS-CoV-2 3C-like Protease Inhibitors Designed Using Generative Deep Learning Approaches and Reviewed by Human Medicinal Chemist in Virtual Reality.
- [24] Mystakidis, S., Berki, E., & Valtanen, J. (2021). Deep and Meaningful E-Learning with Social Virtual Reality Environments in Higher Education: A Systematic Literature Review. Applied Sciences.
- [25] Oh, S.J., & Kim, D. (2021). Machine-Deep-Ensemble Learning Model for Classifying Cybersickness Caused by Virtual Reality Immersion. Cyberpsychology, behavior and social networking.

