H.264 Video Encoding-based Edge-assisted Mobile AR Systems: *Network and Energy Issues*

Anik Mallik and Jiang Xie
The University of North Carolina at Charlotte, Charlotte, NC, USA
Email: amallik@uncc.edu, linda.xie@uncc.edu

ICC 2022 - IEEE International Conference on Communications | 978-1-5386-8347-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICC45855.2022.9838862

***Abstract*—Edge-assisted mobile augmented reality (Edge-MAR) systems have emerged as effective ways to support computation-intensive and latency-sensitive applications for mobile devices due to the offloading capability of heavy computational burdens. However, the network- and energy-resource utilization of such systems is high. Video encoding schemes like H.264 can help Edge-MAR systems reduce latency and bandwidth utilization but at the cost of increased energy consumption. In this paper, we present a comprehensive study of Edge-MAR using H.264 video encoding with a focus on network condition, resource utilization, detection accuracy, and energy consumption of various mobile devices. We collect latency, energy, transmitted data size, and accuracy data for each segment of an object detection pipeline measured through experiments with testbeds, and analyze the non-linear behaviors of Edge-MAR. Following this, we demonstrate the challenges associated with the experiments conducted to test the system as well as the ways to overcome them. Finally, we propose regression-based models to analytically compute different Edge-MAR parameters to achieve desired outcomes. This extensive study provides essential guidelines to network- and energy-aware H.264 video encoding-based Edge-MAR system design.**

***Index Terms*—Energy measurement, mobile augmented reality, edge computing, computation offloading, wireless network**

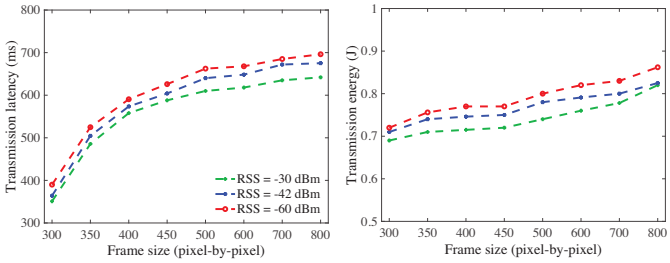
I. Introduction

Augmented reality (AR) is an interactive experience of a real-world environment where objects in the real world are enhanced by computer-generated perceptual information. The introduction of deep learning techniques may add more intelligence to mobile AR (MAR) applications. However, sizeable deep learning models are not proven effective for mobile devices, especially smartphones and other portable devices, due to their limited computation capabilities. Additionally, these models also involve high energy consumption, which causes shorter battery support for mobile devices. In this case, taking the assistance of cloud, edge, or fog computing may relieve mobile devices from these issues.

Edge computing is a distributed computing paradigm that brings computation and data storage closer to the sources of data where heavy computational tasks are offloaded to a nearby edge server from a mobile device. The server completes the computation and sends the result back to the mobile [1]. In this way, mobile devices can save battery life. Mobile vision applications, including AR, can now be provided to mobile users being edge-assisted [2]. However, there is an increasing concern about the network resources, such as bandwidth, used to support edge-assisted MAR or Edge-MAR applications.

A straightforward way to reduce the network resource utilization is to reduce the data size transmitted to and from the edge server over the wireless network. From our study, we discover that a 400×400 image frame sent to an edge server usually takes about 1.2 MB of data, which results in 36 MB per second on average if frames are being sent at a rate of 30 frames per second (fps). With the increase in the number of Edge-MAR users, the large amount of data being transmitted may cause severe delays, which is in contrast to the requirements of real-time applications. Video frame encoding schemes like H.264 help compress the source data and reduce the data size [3]. H.264 encoding compresses the redundant background while keeping the objects of interest in consecutive frames intact. Therefore, there is an opportunity to explore this encoding scheme in MAR.

This work was supported in part by the US National Science Foundation (NSF) under Grant No. 1910667, 1910891, and 2025284 and funds from Toyota Motor North America.



(a) (b)

Fig. 1. Transmission (a) latency and (b) energy consumption at different RSS levels at CPU freq.=1 GHz.

Motivations: Offloading computational tasks to edge servers reduces energy consumption on mobile devices but brings challenges such as increased latency and congested networks. Many resource allocation techniques are proposed that can reduce latency for real-time applications. Nevertheless, to mitigate network congestion or alleviate the impact of poor wireless link quality is under-explored for MAR applications. Our measurement shows that transmission latency from mobile devices to an edge server increases over 10% and energy consumption rises by around 5% due to decrease in received signal strength (RSS) as shown in Fig. 1. In such a case, reducing the data size through compression can be a potential solution. Moreover, moving object detections in smartphones deal with a high volume of data which can be further reduced using video compression techniques.

To reduce the data size of video frames, compression is a well-researched method [4]. However, compression techniques are mainly studied for powerful machines, not smartphones or other mobile devices. Consequently, there is a need for a trade-off study among latency, energy consumption, data size, and inference accuracy. To propose a model capable of balancing such trade-offs, we need to understand how MAR applications behave in terms of latency and energy consumption under different network conditions with or without encoding.

Challenges: The latency and energy consumption of mobile devices in different MAR situations are not linear at all. Through extensive studies, experiments, and measurements, these non-linear traits can be understood. However, setting up an experimental testbed that resembles real-world difficulties involves many challenges.

Moreover, the testbed setup for energy measurement adds further challenges. For Android OS-based smartphones, the energy consumption can be obtained from the on-device log files. However, due to the low sampling rate of this method, it does not provide a precise measurement. Hence, an external power measurement device must be introduced in the testbed that can provide more accurate data with a better sampling rate. However, the input terminal of a smartphone for energy supply is nowadays designed so delicately that accessing these power input terminals and connecting those to an external measuring instrument becomes difficult. Additionally, measuring and collecting the latency data during an MAR activity poses further challenges. While collecting the latency data for each element of the MAR pipeline, each task needs to be organized properly so that the time can be calculated accurately for every event, which is very sensitive in our experiment.

Our contributions: Our contributions in this paper are summarized as follows:

- **Latency and energy consumption of Edge-MAR and Edge-MAR with H.264 encoding (§V-B):** We carry out experiments with different smartphones running an Edge-MAR application (object detection). Then we, for the first time, implement another Edge-MAR system with H.264 video encoding at the client-side (mobile devices) and run the experiments for a similar setup. We use this encoding scheme due to its high compression ratio and compatibility with object recognition systems. Our measurements of the total latency and energy consumption show the necessity of applying compression in data transmission.
- **Impact of H.264 encoding on transmitted data size and inference accuracy (§V-C):** We measure and collect the transmitted data size and inference accuracy for an Edge-MAR application with H.264 encoding and compare with those for an existing Edge-MAR application. Our study shows that there is a trade-off between data size and accuracy in order to use any of the systems.
- **Behavior of Edge-MAR in different network conditions (§V-D):** We emulate three different wireless transmission conditions in our testbed. We analyze the latency and energy data for both systems. It is evident that only the data transmission part of an Edge-MAR pipeline gets affected due to worse wireless signal strength.
- **Impact of encoding on latency and energy consumption (§V-E):** Introducing encoding in an Edge-MAR system helps reduce the latency and transmitted data size, but at the cost of additional energy consumption. We investigate the latency and energy data for encoding only, which further proceeds to regression-based models.
- **Regression model of Edge-MAR parameters (§V-F):** We, finally, develop regression-based models for different Edge-MAR parameters, i.e., latency, accuracy, data size, frame size, and energy consumption, from the large datasets obtained from experiments. These models can help design network- and energy-aware Edge-MAR systems with H.264 encoding.

II. Related Work

Computation offloading and Edge-MAR: Numerous research works are done on computation offloading from mobile devices to servers, especially while using deep learning-based applications [2]. These works are on offloading decisions, efficient resource allocation, service placement [5], and saving energy of mobile devices [6]. However, none of the existing works make the system energy-aware which can process data to be offloaded according to different network conditions.

H.264 encoding for object detection: H.264/AVC encoding scheme is a popular standard video coding technology in streaming applications and video file generation [3]. It is also investigated for object detection in video surveillance applications [7]. Existing papers involve the use of H.264 as feature descriptors [8] or for reducing latency [9]. Dynamic video encoding is also used to improve the streaming latency [10]. Nevertheless, none of the existing works describe the changes in MAR behaviors due to encoding in terms of latency and energy, making it difficult to design adaptive systems.

Network and energy resource utilization by MAR: Making MAR applications network-aware still remains a research problem. It is shown in [11] that the radio network is accountable for around 33% of the total latency of an MAR system, which infuses a need for network-aware MAR systems. Moreover, preparing an energy-aware MAR system is a dormant research issue since energy modeling is very challenging. Analytical models are developed for smartphones' energy consumption [12]. Some papers use on-device logging to measure the energy consumption of smartphones [13], as well as third-party applications, which do not provide precise measurements. Unlike these methods, recent research works prefer to use external energy consumption measuring instrument [14]. New energy models for MAR and an energy-aware MAR system are proposed in [15], [16]. However, none of the existing systems considers latency and energy consumption along with network resource usage altogether.

III. H.264-based Edge-MAR: System Description

We propose an Edge-MAR system based on the H.264 video encoding scheme to detect and recognize objects. This system includes an encoder at the client-side (mobile devices) and a decoder at the server-side. Like other Edge-MAR pipelines, our system does not include a "frame conversion" segment since the encoder can process raw frames with YUV color formats. The system workflow is shown in Fig. 2. First, a frame is generated by the client's camera capturing the intended AR object with the available background. Second, the raw frame is previewed on the client's output display. Third, the raw frame is sent to the encoder of the client and further encoded using the H.264 scheme. Fourth, the encoded frame is transmitted to the edge server over the wireless network. This communication

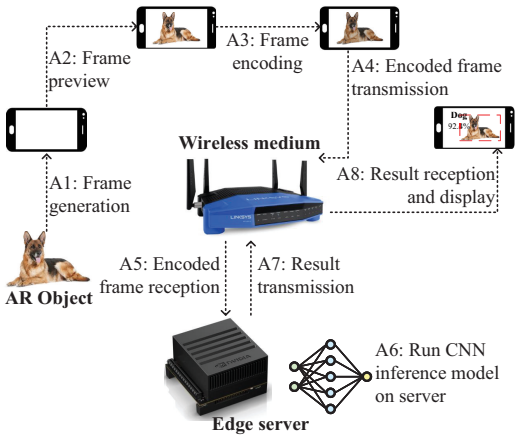


Fig. 2. System description of Edge-MAR with H.264.

takes place by creating a TCP (transmission control protocol) socket at the server end. Fifth, the server receives the encoded frame and then decodes it. If running convolutional neural netwok (CNN) on each frame takes longer than the frame reception time, all the received and decoded frames remain in the queue (buffer). Sixth, the server runs the CNN inference model on each decoded frame and gets the result of the object detection and recognition. The result is then transmitted back to the client using the TCP socket. Finally, the client receives the result for each frame and displays it with a bounding box and inference accuracy. This workflow is repeated while the MAR application is running.

IV. Experimental Setup and Methodology

Testbed: We perform vast experiments using our implemented testbeds consisting of different smartphones (as mobile client devices), an edge server, and a WiFi router. In order to make the proposed system usable for most of the commercially available Android OS-based smartphones, we select phones released in different years, having different specifications, which are listed in Table I. We use Jetson AGX Xavier as the edge server, which has an 8-core ARM 64-bit CPU with 32GB 256-bit LPDDR4x 137GB/s RAM and 512-core Volta GPU with Tensor Cores. As the WiFi access point, we use a Linksys dual-band router, which is connected to the edge server. For energy consumption measurement purposes, we connect an external instrument – "Monsoon Power Monitor" to the smartphones.

Methodology: For "Edge-MAR with H.264", we use Android's "MediaCodec" library to encode the generated frames from the mobile device's camera. The camera is moved at a constant speed and angle to capture the objects to be detected. MediaCodec provides a compression ratio of around 92% (1:12.5) in our experiment. For ease of development, we save 300 frames into a video file, then encode the video and transmit it. We use 30 fps, I-frame interval 5, and maximum video bitrate 30 MB/s as configuration parameters for encoding. For remote execution in the edge server, we adopt a version of the famous CNN model, YOLOv3 [17] that uses COCO dataset having 80 classes of objects. We use the 2.4 GHz band of the router to access the Wi-Fi network. To produce different network conditions with different RSS, we use different distances with line-of-sight considerations from the Wi-Fi access point to the mobile devices while keeping the transceivers directional.

The energy consumption of smartphones is measured by the power monitor that is connected to the smartphones via the battery terminals. In the case of the latest smartphones, the batteries need to be removed from the back panel of the phones by applying heat from a heat gun. Then the terminals are soldered to extended wires, which are then connected to the input/output terminals of the power monitor — the power monitor powers up the phones.

This external power monitor provides voltage, current, and energy consumption data for every 2 ms. Before measuring the energy consumption, all the irrelevant features and background applications are turned off in the smartphones to understand the behavior of the object detection application properly. The latency data for different segments of the Edge-MAR pipeline, on the other hand, are logged in separate files. The application is run for 300 frames each time. Then the energy and latency are considered for a single frame by taking the average of all the measurements for 300 frames. To compare our experimental results with an existing MAR system, we implemented the work in [18], which we name here as "Edge-MAR" only. Finally, using multiple linear regression, new models are developed for different parameters of the Edge-MAR system taking all the experimental data as inputs.

V. Results and Discussion

A. Key parameters

1) Performance metrics: In any edge-based AR system, latency is the most important performance metric, which defines whether a system is suited for real-time or other sensitive applications. The inference accuracy describes the system's ability to recognize any object correctly. Moreover, for Edge-MAR systems, energy consumption is another crucial metric to determine a mobile device's stability in terms of battery health. Lastly, the transmitted data size dictates how much network resources are consumed.

2) Control factors: Our experimental testbed consists of an H.264 encoder, where the encoding configuration regulates the encoding latency and energy consumption due to encoding. Additionally, smartphones' CPU frequency governs the way frames are processed and encoded. The size of the captured frames does not necessarily control the compression, but the data size depends on it heavily. However, no matter what the data size is, the transmission latency and energy vary on different signal strengths of the wireless medium.

B. Latency and energy consumption of Edge-MAR and Edge-MAR with H.264 encoding

We conduct experiments on both Edge-MAR and Edge-MAR with H.264 for 8 different sizes of frame resolution (300×300, 350×350, 400×400, 450×450, 500×500, 600×600, 700×700, and 800×800) and for 3 different CPU frequencies (1, 2, and 3 GHz). The main difference between the pipelines of these two systems is the presence and absence of frame conversion and frame encoding, and vice-versa. The latency and energy measurements are shown in Fig. 3.

TABLE I
BRIEF SPECIFICATIONS OF THE SMARTPHONES USED IN THE EXPERIMENTS

Manufacturer	Samsung	Asus	Motorola	Vivo	Google
Model	Galaxy S5	ZenFone AR	One Macro (XT2016-2)	IQOO Z1	Pixel 4a
OS	Android 6.0.1	Android 7.0	Android 9.0	Android 10.0	Android 10.0
SoC	Snapdragon 801 (28nm)	Snapdragon 821 (14nm)	MediaTek (12nm)	Mediatek (7nm)	Snapdragon 730G (8nm)
CPU	32-bit 4-core 2.5GHz Krait 400	64-bit 4-core 2.4GHz Kryo	64-bit Octa-core 2x2GHz ARM Cortex	Octa-core (4x2.6GHz & 4x2GHz Cortex)	Octa-core (2x2.2GHz & 6x1.8GHz Kryo)
GPU	Adreno 330	Adreno 530	Mali-G72	Mali-G77	Adreno 618
RAM	2GB	6GB	4GB	6GB	6GB
WiFi	802.11n/ac	802.11n/ac/ad	802.11 b/g/n	802.11 a/b/g/n/ac	802.11 a/b/g/n/ac
Release date	April, 2014	July, 2017	October, 2019	May, 2020	August, 2020

The overall latency and energy consumption for Edge-MAR varies from 677.6 ms to 1156.35 ms and 5.87 J to 7.55 J for 1 GHz, 662.84 ms to 1144 ms and 6.45 J to 7.81 J for 2 GHz, and 610.96 ms to 1115.5 ms and 6.58 J to 8.6 J for 3 GHz CPU frequency for the above-mentioned frame sizes. The major latency is caused by the transmission, and most of the energy is consumed by the frame generation. We find that *for frame sizes from 350×350, the latency does not increase drastically for Edge-MAR till the frame size of 500×500. After that, the change in latency is steeper.* Similar trend goes for *energy consumption* also in Edge-MAR.

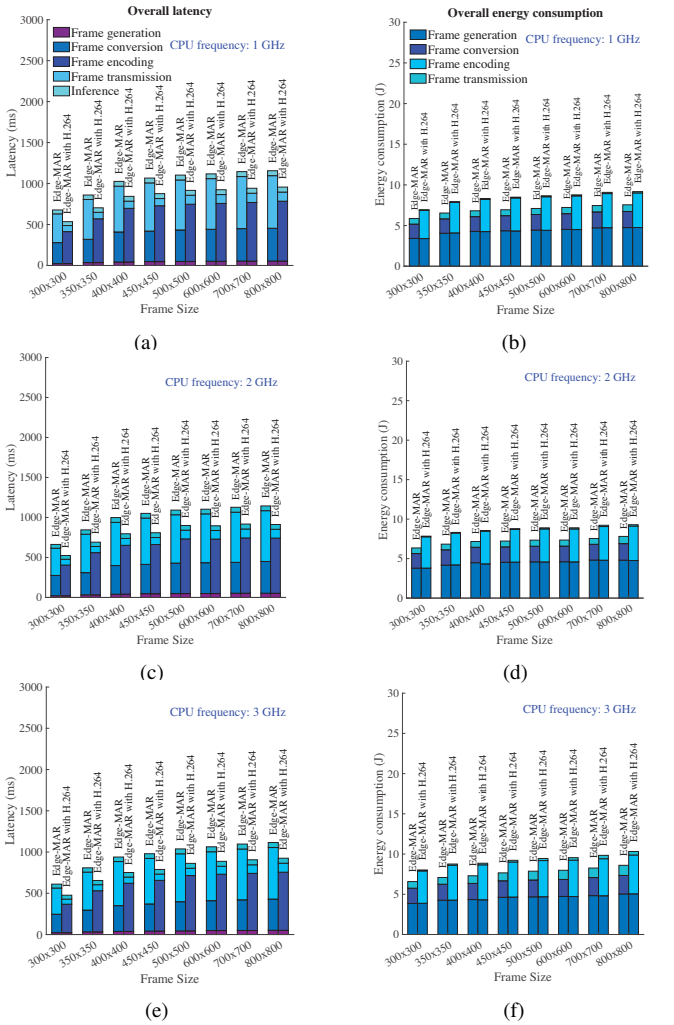


Fig. 3. Overall latency at CPU frequency (a) 1 GHz, (c) 2 GHz, and (e) 3 GHz, and energy consumption at CPU frequency (b) 1 GHz, (d) 2 GHz, and (f) 3 GHz for Edge-MAR and Edge-MAR with H.264 respectively.

For Edge-MAR with H.264, we find that *the overall latency is reduced by around 80%, but at the cost of energy consumption increase of around 20%. Most of the latency and energy consumption is here caused by the encoding.* Similar to Edge-MAR, in this system, from frame size 350×350 to 500×500, latency and energy consumption do not increase significantly.

Insight: With the increase in CPU frequency, latency decreases, and energy consumption increases. However, for the increase in frame resolution, both latency and energy consumption rise. Edge-MAR with H.264 provides less latency but at the cost of an apparent increase in energy consumption. Frame size 350×350 to 500×500 is observed to be an optimal range for MAR applications in terms of latency and energy consumption due to the hardware limitations such as sensor size and frame rates of mobile devices.

C. Data size and accuracy

Our measurement shows that with the increase in frame sizes, the size of transmitted data per frame rises, as shown in Fig. 4. This is due to the increase in frame information with the larger frame size. From frame size 400×400 to 450×450 and from 600×600 to 700×700, there is a sharp rise of data size due to the sudden introduction of additional information. *The increment in data size is more stable from frame sizes 450×450 to 500×500, because of a more minor increase in background information with the movement of the camera or the object.* This is true for the encoded data size as well. Another interesting finding is that *there is a slight decline in compression at frame sizes 600×600 and 800×800, because of small information added to the frames compared to the other sizes.* The data size for Edge-MAR varies from 0.95 MB to 1.8 MB per frame, and for Edge-MAR with H.264 from 0.072 MB to 0.13 MB per frame with an increase in frame sizes, i.e., additional information. Furthermore, the inference accuracy varies a little across different frame sizes. It ranges from 84.50% to 89.7% per frame, with YoloV3 running at the server. Another *However, due to encoding, this accuracy drops slightly by around 0.1% to 0.5%, because of the lossy compression of the frames*, as depicted in Fig. 5.

Insight: Edge-MAR with H.264 provides a considerable reduction in data size in our experiment, but at the cost of reduced inference accuracy by 0.5% implying that using H.264, an Edge-MAR system can save around 92% of the allocated bandwidth with slightly reduced accuracy. Neither the data size nor the inference accuracy depends on the CPU frequency. Though frame resolution does not influence

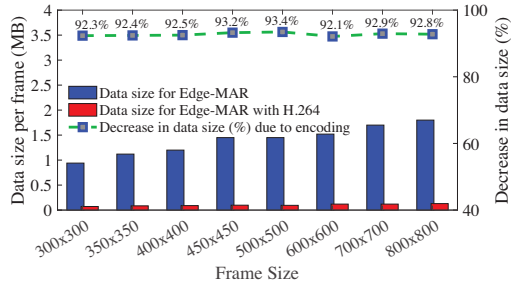


Fig. 4. Transmitted data size and decrease due to encoding for different frame sizes and CPU frequencies.

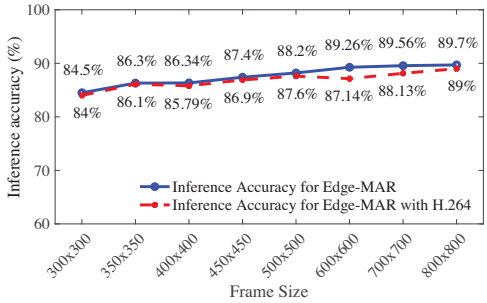


Fig. 5. Inference accuracy in percentage for different frame sizes and CPU frequencies.

encoding, with additional information in a frame, encoded data size and compression ratio increase.

D. Impact of RSS on transmission latency and energy

The impact of the signal strength of the wireless medium is pivotal in the transmission of data from mobile devices to the server. Both latency and energy consumption due to transmission increase with the decrease in RSS, which in turn increases the overall latency and energy consumption. We generate 3 RSS levels for our experiment: -30 dBm, -42 dBm, and -60 dBm. Fig. 6 and Fig. 7 show the transmission latency and energy, respectively, for different RSS at different CPU freq. for both Edge-MAR and Edge-MAR with H.264. For RSS=-60 dBm, in Edge-MAR and Edge-MAR with H.64, transmission latency increases on average by 44.41 ms and 7.73 ms at 1 GHz, 32.13 ms and 6.38 ms at 2 GHz, and 25.36

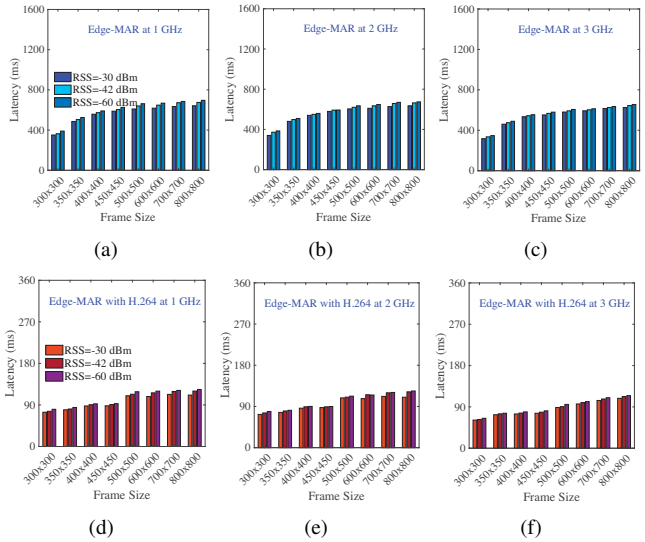


Fig. 6. Transmission latency for different RSS for Edge-MAR at CPU frequency (a) 1 GHz, (b) 2 GHz, and (c) 3 GHz, and for Edge-MAR with H.264 at (d) 1 GHz, (e) 2 GHz, and (f) 3 GHz.

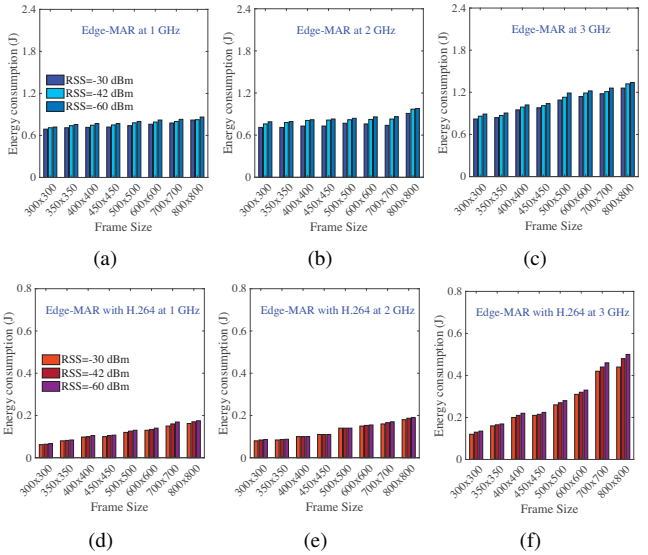


Fig. 7. Energy consumption for frame transmission for different RSS for Edge-MAR at CPU frequency (a) 1 GHz, (b) 2 GHz, and (c) 3 GHz, and for Edge-MAR with H.264 at (d) 1 GHz, (e) 2 GHz, and (f) 3 GHz.

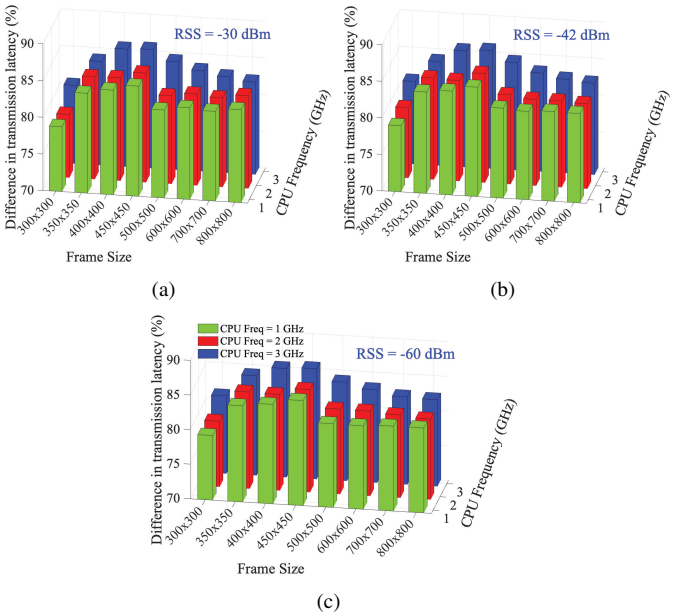


Fig. 8. Difference in transmission latency due to encoding in percentage for RSS levels (a) -30 dBm, (b) -42 dBm, and (c) -60 dBm.

ms and 5.08 ms at 3 GHz of CPU freq. sequentially from RSS=-30 dBm. At the same RSS, a corresponding increase in energy consumption for transmission for these systems are 49.38 mJ and 9.38 mJ at 1 GHz, 89.75 mJ and 4.31 mJ at 2 GHz, and 75.63 mJ and 24.75 mJ at 3 GHz of CPU freq. from -30 dBm. Fig. 8 illustrates the difference in transmission latency for encoding from Edge-MAR only for different RSS.

Insight: It is evident that with the rise in CPU frequencies, transmission latency decreases. However, at 2 GHz CPU frequency, the transmission energy does not increase sharply, compared to the other frequencies, due to smartphone architectures' high compatibility with the 2 GHz range. Moreover, due to encoding, transmission latency reduces by almost 80% from that of only Edge-MAR.

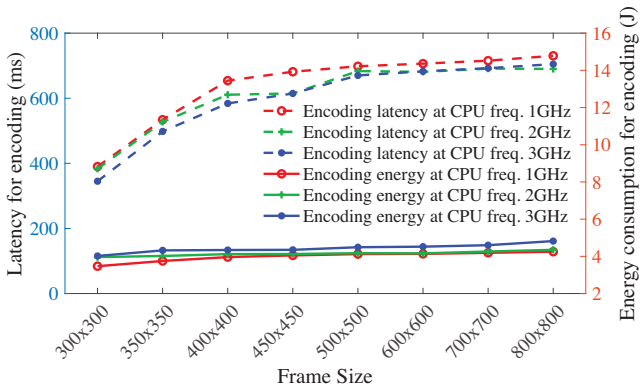


Fig. 9. Latency and energy consumption due to H.264 encoding.

E. Impact of encoding on latency and energy consumption

Since our Edge-MAR system involves H.264 at the client-side, encoding has an impact on both latency and energy consumption of mobile devices, shown in Fig. 9. With the increase in CPU frequency, the encoding latency decreases, but the encoding energy increases. However, at 2 GHz CPU frequency, the reduction in latency is high, and the increase in energy consumption is much lower than that at 3 GHz.

Insight: At 2 GHz of CPU frequency, smartphones provide higher efficiency in terms of both encoding latency and energy consumption due to encoding. For frame size 450×450, it gives the optimal latency and energy consumption.

F. Regression model

To develop regression-based models, we denote frame size as S_{frame}, data size as S_{data}, accuracy for encoded frames as Acc_{encode}, difference in overall and transmission latency from Edge-MAR to Edge-MAR with H.264 as $\triangle t_{all}$ and $\triangle t_{transm}$ respectively, encoding latency as t_{encode}, the difference in overall energy consumption from Edge-MAR to Edge-MAR with H.264 as $\triangle E_{all}$, RSS levels as S_{RSS}, and finally CPU frequency as f. The proposed models for Acc_{encode}, t_{encode}, $\triangle t_{transm}$, and S_{frame} are summarized in Table II. The R^2 values show the strength of the relationship between the model and the dependent variables, implying a good fit of the model. This model can be used to further design network- and energy-aware H.264-based Edge-MAR systems where developers can choose the independent variables to achieve desired values of dependent variables within the 95% confidence boundary.

VI. Conclusion

In this paper, we presented a detailed, comprehensive experimental study of network- and energy-resource utilization by an Edge-MAR with H.264 in different wireless network conditions for a variety of mobile devices. Our measurement showed that the use of H.264 in Edge-MAR can substantially reduce latency, but at the cost of slightly increased energy consumption – especially in worse wireless network conditions. We observed that with the increase in CPU frequency and frame size, the overall transmission and encoding latency and energy consumption varies to a great extent, but at some specific frequencies and frame sizes, the variations are different due to smartphones' efficiency issues. The study showed the necessity of trade-offs among Edge-MAR parameters to achieve desired outcomes. Finally, we proposed regression-based models to design Edge-MAR systems with H.264 encoding. Any MAR system involving video transmission can leverage the benefits of this proposed model. In short, we believe that the findings from this paper will provide great insights to further designs of latency- and energy-aware Edge-MAR pipelines with H.264.

TABLE II
Proposed Linear Regression-Based Models

Parameters	Proposed models	R^2 value
Acc_{encode}	$0.81 + 2.94\times10^{-5}S_{frame} + 0.34f$	0.73
t_{encode}	$382.77 + 0.53S_{frame} - 19.89f$	0.64
$\triangle t_{transm}$	$23.08 - 0.21S_{frame} + 5316.3S_{data} - 11.8f - 0.89S_{RSS}$	0.71
S_{frame}	$-5306.5 - 2.33f - 1.18S_{RSS} - 39.38\triangle E_{all} + 1.2\triangle t_{all} + 0.96t_{encode} - 2.43t_{transm} + 6075.7Acc_{encode} + 7508.2S_{data}$	0.97

References

[1] F. Jalali, K. Hinton, R. Ayre, T. Alpcan, and R. S. Tucker, "Fog computing may help to save energy in cloud computing," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, 2016.

[2] X. Ran, H. Chen, Z. Liu, and J. Chen, "Delivering deep learning to mobile devices via offloading," in *Proc. of ACM Workshop on Virtual Reality and Augmented Reality Network*, 2017, pp. 42–47.

[3] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

[4] S. Huang, J. Xie, and M. Muslam, "A cloud computing based deep compression framework for UHD video delivery," *IEEE Transactions on Cloud Computing*, 2022.

[5] B. Gao, Z. Zhou, F. Liu, and F. Xu, "Winning at the starting line: Joint network selection and service placement for mobile edge computing," in *Proc. of IEEE INFOCOM*, 2019, pp. 1459–1467.

[6] Q. Wang, S. Guo, J. Liu, and Y. Yang, "Energy-efficient computation offloading and resource allocation for delay-sensitive mobile edge computing," *Sustainable Computing: Informatics and Systems*, vol. 21, pp. 154–164, 2019.

[7] P. Dong, Y. Xia, L. Zhuo, and D. Feng, "Real-time moving object segmentation and tracking for H.264/AVC surveillance videos," in *Proc. of IEEE International Conference on Image Processing*, 2011.

[8] M. Makar, V. Chandrasekhar, S. S. Tsai, D. Chen, and B. Girod, "Interframe coding of feature descriptors for mobile augmented reality," *IEEE Transactions on Image Processing*, vol. 23, no. 8, 2014.

[9] L. Liu, H. Li, and M. Gruteser, "Edge assisted real-time object detection for mobile augmented reality," in *Proc. of ACM International Conference on Mobile Computing and Networking (MobiCom)*, 2019, pp. 1–16.

[10] S. Huang and J. Xie, "DAVE: Dynamic adaptive video encoding for real-time video streaming applications," in *Proc. of IEEE SECON*, 2021.

[11] K. Apicharttrisorn, B. Balasubramaniany, J. Chen, R. Sivarajz, Y.-Z. Tsai, R. Janay, S. Krishnamurthy, T. Trany, and Y. Zhouy, "Characterization of multi-user augmented reality over cellular networks," in *In Proc. of IEEE SECON*, 2020, pp. 1–9.

[12] J. M. Vatjus-Anttila, T. Koskela, and S. Hickey, "Power consumption model of a mobile GPU based on rendering complexity," in *Proc. of IEEE International Conference on Next Generation Mobile Apps, Services and Technologies*, 2013, pp. 210–215.

[13] K. Chen, T. Li, H.-S. Kim, D. E. Culler, and R. H. Katz, "MARVEL: Enabling mobile augmented reality with low energy and low latency," in *Proc. of ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2018, pp. 292–304.

[14] S. Aggarwal, M. Ghoshal, P. Banerjee, D. Koutsonikolas, and J. Widmer, "802.11ad in smartphones: energy efficiency, spatial reuse, and impact on applications," in *Proc. of IEEE INFOCOM*, 2021, pp. 1–10.

[15] H. Wang and J. Xie, "User preference based energy-aware mobile AR system with edge computing," in *Proc. of IEEE INFOCOM*, 2020.

[16] H. Wang, B. Kim, J. Xie, and Z. Han, "How is energy consumed in smartphone deep learning apps?" in *Proc. of IEEE GLOBECOM*, 2019.

[17] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[18] H. Wang, B. Kim, J. Xie, and Z. Han, "Energy drain of the object detection processing pipeline for mobile devices: analysis and implications," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 1, pp. 41–60, 2021.