5G Edge Vision: Wearable Assistive Technology for People with Blindness and Low Vision

Tommy Azzino*, Marco Mezzavilla*, Sundeep Rangan*, Yao Wang*, John-Ross Rizzo* †
*NYU Tandon School of Engineering, Brooklyn, NY, USA - {ta1731, mezzavilla, srangan, yw523}@nyu.edu
†Department of Rehabilitation Medicine, NYU Langone, New York, NY, USA - {johnrossrizzo}@gmail.com

2024 IEEE Wireless Communications and Networking Conference (WCNC) | 979-8-3503-0358-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/WCNC57260.2024.10570607

***Abstract*—In an increasingly visual world, people with blindness and low vision (pBLV) face substantial challenges in navigating their surroundings and interpreting visual information. From our previous work, VIS⁴ION is a smart wearable that helps pBLV in their daily challenges. It enables multiple microservices based on artificial intelligence (AI), such as visual scene processing, navigation, and vision-language inference. These microservices require powerful computational resources and, in some cases, stringent inference times, hence the need to offload computation to edge servers. This paper introduces a novel video streaming platform that improves the capabilities of VIS⁴ION by providing real-time support of the microservices at the network edge. When video is offloaded wirelessly to the edge, the time-varying nature of the wireless network requires adaptation strategies for a seamless video service. We demonstrate the performance of our adaptive real-time video streaming platform through experimentation with an open-source 5G deployment based on open air interface (OAI). The experiments demonstrate the ability to provide microservices robustly in time-varying network conditions.**

***Index Terms*—5G, testbed, AI, assistive technology, e-health, wearable, edge computing, video streaming.**

I. Introduction

There are 39 million blind and 246 million people with low vision worldwide, according to the world health organization (WHO) [1]. While therapeutic advances are being developed for a handful of conditions, there are a multitude of etiologies that result in severe visual disability [2], and the prevalence of many of these conditions is increasing. Impaired vision constrains mobility, inevitably leading to problems with unemployment and quality of life [3], both of which limit psychosocial well-being.

To address the challenges of people with blindness and low vision (pBLV), we have developed VIS⁴ION (visually impaired smart service system for spatial intelligence and onboard navigation) [4], [5] – a discreet and ergonomic wearable equipped with miniaturized sensors, including cameras, microphones, global navigation satellite system (GNSS) receivers, and inertial measurement units (IMUs). A binaural bone conduction headset and an optional reconfigurable waist strap turned haptic interface provide real-time feedback, as depicted on the left of Fig. 1.

VIS⁴ION offers multiple microservices based on artificial intelligence (AI) for pBLV with visual data as input. These include object detection for obstacle avoidance, large language model (LLM)-based audio assistance with visual data, and vision-based localization and navigation. Ideally, these microservices run on the edge, where powerful computational resources enable faster inference and deployment of larger models for optimal performance. However, edge offloading requires seamless wireless connectivity to upload and process video and audio data continuously.

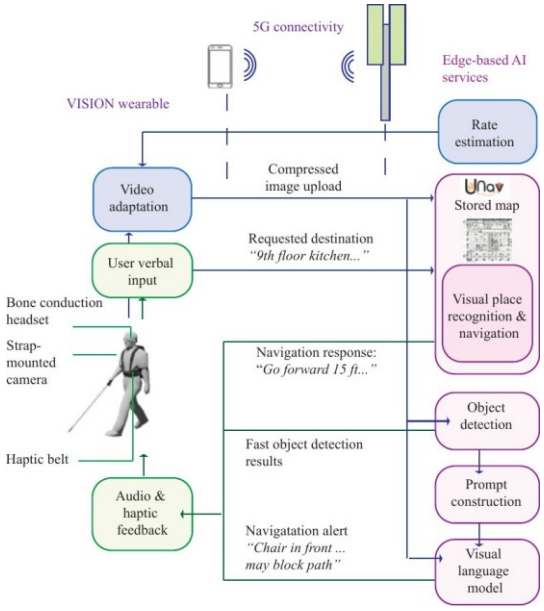


Fig. 1: The proposed VIS⁴ION platform for people with blindness and low vision provides several powerful edge-based AI microservices. These services all use video captured on a strap mounted camera on the wearable.

A critical challenge in developing VIS⁴ION with edge offloading – and the focus of this work – is wireless connectivity. Wireless deployment of vision-based edge services suffers from the time-varying nature of network conditions [6]. Wireless connectivity will vary with the user's location relative to the access point or base station, as well as with the network load. Therefore, the bitrate and processing frequency of the video frames will need to be adjusted based on such conditions. Most previous work on video adaption for edge-based services has focused on object detection [7]. In contrast, the VIS⁴ION system requires adapting video that caters to multiple services with different quality of service (QoS) requirements.

The paper presents several important contributions to address these challenges and advancing the set of features supported by VIS⁴ION:

- We develop a complete video streaming platform to support the real-time fruition of the envisioned microservices for pBLV. The implementation supports concurrent

979-8-3503-0358-2/24/$31.00 ©2024 IEEE
Authorized licensed use limited to: New York University. Downloaded on September 10,2024 at 16:42:20 UTC from IEEE Xplore. Restrictions apply.

execution of the AI-based microservices over a pool of available graphics processing unit (GPU) resources.

- We implement a method to estimate the available link rate using our REBERA algorithm [8]. Based on the link rate estimate, a heuristic algorithm is proposed to adapt the video across different services.
- We evaluate the proposed adaptive video streaming platform using an open-source 5G testbed based on open air interface (OAI) in an indoor lab environment. 5G link conditions are manipulated by changing the available bandwidth to emulate a network under time-varying load.

II. VIS4ION with 5G Connectivity and Video Adaptation

VIS4ION microservices architecture

VIS4ION wearable is wirelessly connected to a set of powerful edge microservices based on AI, including visual scene processing, real-time localization and navigation, and audio assistance through vision-language models (VLMs). The architecture for these services is schematically depicted in Fig. 1. The primary input for all services is the video captured from the wearable strap-mounted cameras for image stability. The captured video is compressed and uploaded wirelessly to the edge. In this work, we explore 5G connectivity, although any wireless connectivity can be used, including Wi-Fi. As we explain below, the uplink rate can vary, and adaptation of the video compression is critical.

Several edge microservices use the uploaded video data. The first service is **object detection** using *YoLo*. Building on our previous work in [6], our video streaming platform allows for real-time high frame rate object detection using one of the most recent YoLo models, YoLoV7 [9]. In particular, we use *YoLoV7-w6*, which supports inference on higher-resolution images compared to the basic YoLoV7 model. In order to support fast responses to immediate and dangerous obstacles, the object detection results are generally reported at a fast frame rate and then rendered to the user via haptic or audio feedback. Our analysis in [6] suggested that object detection should be as fast as 30 fps to allow a response time of 100 ms when including video upload, inference, and feedback time.

The second microservice offered is **UNav** [10]. UNav is an AI-based software that creates infrastructure-free, camera-based digital twins or 3D maps of complex indoor and outdoor environments, supporting wayfinding to close or far-range destinations through audio and haptic user commands. Note that the commonly used global positioning system (GPS) would not work indoors and has a relatively low accuracy. UNav relies on *visual place recognition* [11], where the features of the query image are compared against a library of pre-stored features to locate the user. The location information can then be used to provide navigation instructions. For example, as shown in Fig. 1, the user could verbally request a destination such as a kitchen, and UNav first finds the user's current location from the current video frame and then provides a text-based navigation suggestion such as "go forward" a certain distance. The navigation suggestions are then converted to audio using text-to-speech tools and delivered to the user through the headset.

Lastly, our platform supports real-time audio assistance using a **VLM** called *InstructBLIP* [12]. InstructBLIP is an LLM-based VLM for comprehensive scene understanding and textual descriptions. It can generate descriptive output text based on the input prompt and image. Specifically, InstructBLIP begins by capturing high-level visual representations of the image. Then, the input prompt and visual features are used to generate contextualized information through an LLM. Specifically, InstructBLIP works in association with an LLM known as Vicuna-13B [13] to generate the final output text.

Each service has different computational and feedback frequency requirements. As described above, object detection inference must support a high frame rate to capture suddenly appearing objects that could pose a risk to pBLV. Therefore, feedback must be received within a strict delay budget. On the other hand, navigation and audio assistance require complex models and can often be executed with lower inference frequency. For example, localization and navigation may be sufficient at one feedback per second. The same applies to audio assistive on the general scene composition and relatively far obstacles.

Adaptive video streaming platform

The real-time video streaming platform consists of a *client* and a *server*. The client runs on an NVIDIA Jetson board, which connects to a 5G-enabled smartphone for cellular connectivity, as shown in Fig. 2. Primarily, the client initiates an uplink video stream to the server using real-time transport protocol (RTP) and receives a sequence of outputs from the microservices introduced above. The reception of such information is essential to unleash the full potential of VIS4ION and assist pBLV through haptic and audio feedback. Importantly, to ensure the seamless performance of these assistive microservices, video streaming must consider link rate fluctuations that are related but not limited to network congestion, poor wireless channel quality, and the number of users simultaneously accessing the same wireless resources. As such, our video streaming client implements an adaptive mechanism that enables video encoder adjustments based on channel quality. Specifically, the client periodically receives link rate estimates from the server and then uses the REBERA algorithm [8] to predict the link rate in the next time step based on previous observations and accordingly set the appropriate encoder rate. In addition, the client adapts the video streaming resolution according to the set video rate. As we found in [6], lowering the video resolution when the target video rate is below a certain threshold improves object detection performance.

Navigation and VLM-based scene description do not need to run at high frame rates due to their high computational load and relaxed latency requirements. A single high-quality video frame per second should suffice. Hence, we support multiple video-based services at different frame rates and spatial resolutions. In particular, when the predicted link rate is high,

Authorized licensed use limited to: New York University. Downloaded on September 10,2024 at 16:42:20 UTC from IEEE Xplore. Restrictions apply.

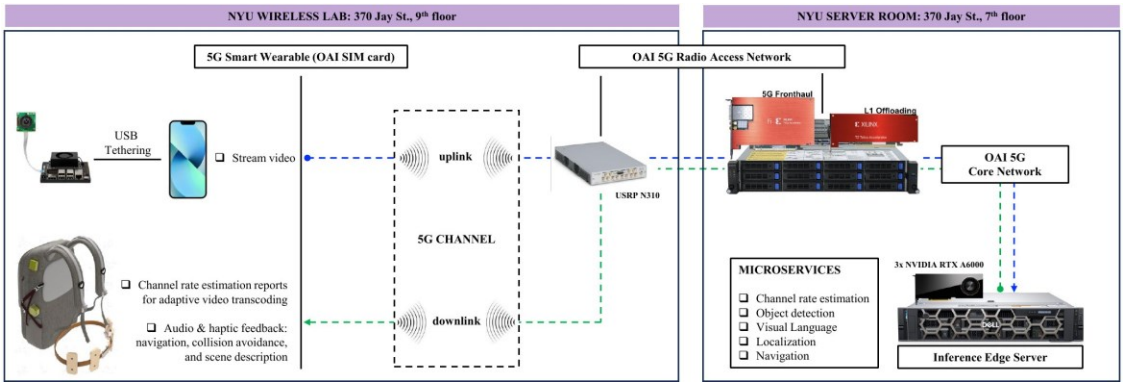


Fig. 2: End-to-end 5G real-time adaptive video streaming platform.

we send a single video stream with a high frame rate (30 fps) and high spatial resolution (1920×1080). In this case, object detection will be performed for every frame, but navigation and VLM will be performed only once every second, hence every 30 frames are received. When the predicted link quality degrades, we reduce the video resolution but keep the same frame rate so that object detection can still be performed frequently but with reduced accuracy. We further enable a secondary video stream with a low frame rate (e.g., 1 fps) but high spatial resolution to perform navigation and VLM inference on high-quality and resolution images. As outlined in [6], high-resolution images enable the detection of objects far away (which would occupy a small region in an image and are more difficult to detect from low-resolution images). Sending high-resolution video when the network connectivity is robust allows the user to "see" objects far away and plan their trajectory properly. When the network link is weak, despite the reduced image resolution, we can still accurately detect nearby objects (which have relatively larger image size and can still be easily detected in low-resolution images) that could threaten pBLV. We choose to perform navigation and VLM services only in high-resolution frames because our separate studies have found that increasing spatial resolution can significantly improve navigation accuracy [14]. Effectively, the presence of a secondary video stream adds a second layer of adaptation based on the QoS requirement for a given microservice.

As depicted in Fig. 2, the server runs on any computer system equipped with GPUs. It receives the uplink video stream from the client and performs a series of services instrumental in supporting the full set of features expected for VIS4ION. Similarly to [8], the server periodically computes an estimate of the link rate based on the incoming stream of video packets. As previously discussed, each estimate is sent back to the client for video adaptation. The server implements a pipeline to support real-time, high-frame rate object detection on the received video frames. In addition, the server hosts microservices for navigation and localization, as well as vision-language inference on high-resolution frames at a low frame rate. All these services run on parallel threads to distribute their computational requirements over different resources on the same machine. Furthermore, when enabled by the client, the server supports the reception of a secondary low-frame rate, high-resolution video stream. If this stream is present, microservices with a lower frame rate requirement (i.e., navigation and localization, and VLM instruction) enjoy high-quality video frames. If this stream is not running, the mentioned microservices use frames from the primary high-frame-rate video stream.

Finally, the client and server have been developed using GStreamer libraries [15] to stream real-time video, and DeepStream [16] for real-time object detection with YoLo running on NVIDIA's GPUs. Socket and thread programming in Python have been used to enable parallel execution of microservices and link rate estimation.

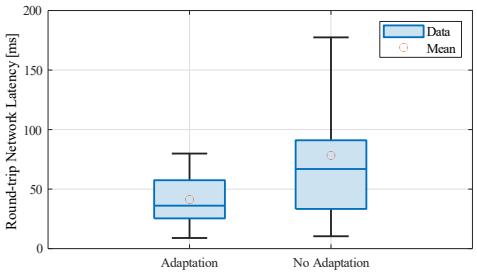
III. 5G Experimental Platform

5G cellular stack

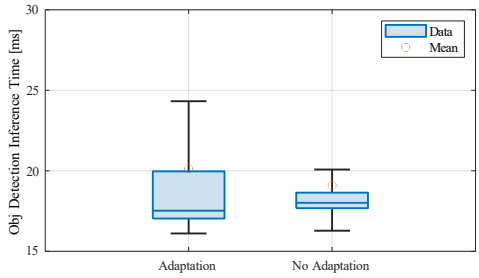
A programmable 5G network was deployed using the widely adopted OAI stack [17]. OAI is an open-source project that implements the 3rd generation partnership project (3GPP) technology on general-purpose x86 computing hardware and commercial off-the-shelf (COTS) software-defined radio (SDR) cards such as universal software radio peripherals (USRPs). Notably, the OAI framework includes both radio access network (RAN) and 5G core (5GC), enabling end-to-end 5G cellular functionalities. As shown in Fig. 2, a commercial phone gets programmable 5G connectivity using an OAI subscriber identity module (SIM) card. The 5G RAN components are compiled on the host machine, whereas the 5GC is loaded on the same machine by launching several docker images that will jointly verify, authenticate, and subscribe each preregistered OAI device. Once connected to 5G, the user will be able to communicate with any accessible servers on the external data network. In our case, as discussed in further detail in the next section, we deployed a powerful inference server on the same dedicated network as 5GC, thus mimicking the performance of an edge server. Finally, a significant benefit of using open-source cellular software is that critical parameters can be logged and tracked at any network layer, allowing us to accurately profile the overall performance of our experiments.

OAI code modification – variable BWPs: Critically, to assess the advantages of our adaptive video streaming technology, we

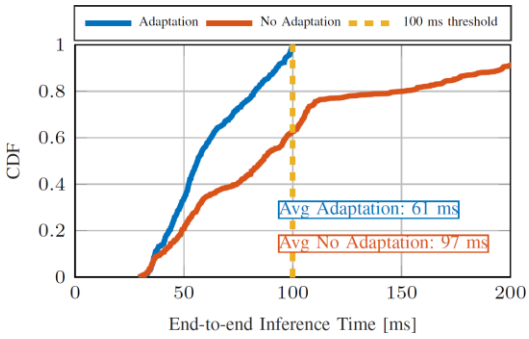
Authorized licensed use limited to: New York University. Downloaded on September 10,2024 at 16:42:20 UTC from IEEE Xplore. Restrictions apply.



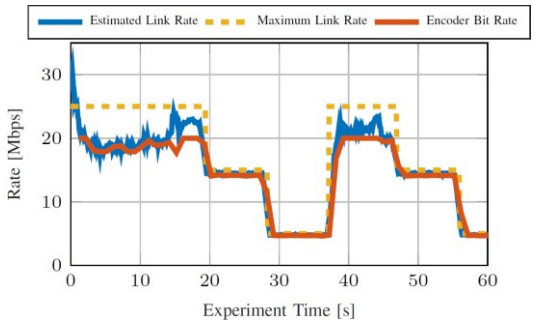
(a) Box-plots for the round-trip 5G network latency.



(b) Box-plots for the object detection inference time.



(c) CDF of the end-to-end inference latency. It includes round-trip networking and object detection inference times.



(d) Maximum link rate, estimated link rate, and encoder bitrate over the duration of the experiment.

Fig. 3: Results for the video streaming experiment on our platform. The blue box in the box plots represents the interquartile range between the 25th and 75th percentiles. The blue line represents the median.

modified to the OAI next-generation node base (gNB) scheduler code to trigger specific variations in channel capacity. To do so, we leverage a new feature in 5G, bandwidth part (BWP), introduced in 3GPP Release 15, to dynamically adapt each user's carrier bandwidth and associated numerology [18]. We programmed the scheduler so that gNB will modify the allocated BWP configuration at specific time intervals.

Hardware

5G Network: As shown in Fig. 2, the 5G network infrastructure comprises the following hardware: the gNB *Radio Unit* runs on the USRP N310, which connects to the Proxicast 4G/5G Aerial Antennas; the gNB *Baseband Unit* and the 5GC run on the *Nautilus* machine, which is a high-performance computing (HPC) server equipped with Xilinx T1 and T2 boards for L1 acceleration.

Client: The client includes a Jetson Orin NX 16 GB that connects to a wide-angle camera. This device has 5G OAI connectivity through USB tethering via a Google Pixel 5A.

Server: We use a powerful Dell Precision 7920 Rack Workstation with 3x NVIDIA RTX A6000 (48GB each) to host the video streaming server with AI edge services for our platform. This machine shares the same rack as the Nautilus server to mimic a realistic edge computing environment.

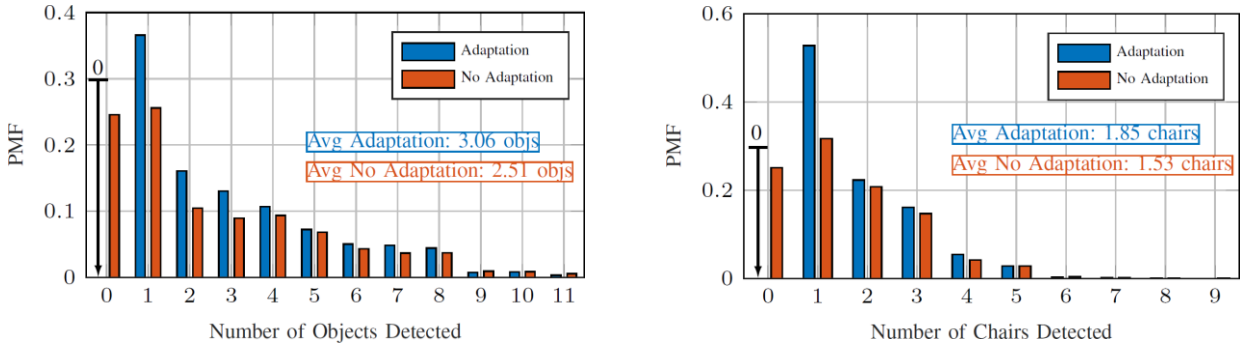
IV. Results

This section discusses the experimental results obtained using the 5G platform introduced above. Our goal is to test our adaptive video streaming platform along with the microservices enabled for pBLV and to measure its overall performance. Critically, we compared the results of each microservice with and without video adaptation enabled. Without video adaptation, we fix the encoder bitrate at 20 Mbps, the frame rate to 30 fps, and the video resolution to 1920×1080.

We recorded a high-bitrate and resolution video where we walked from our wireless lab to the kitchen on the same floor. We placed a set of potentially hazardous obstacles, such as office chairs, along the route. Then, we walked from the lab room to the kitchen, recording approximately 60 seconds of video. We then emulated the streaming of this video under time-varying network conditions and invoked the three AI services on the decoded video on the inference edge server. A pre-recorded video allowed us to compare the performance of the multiple AI services with video adaptation vs. without video adaptation under the same controlled network conditions and using the exact same video frames. We tethered the Jetson board to the Google Pixel to gain 5G OAI connectivity. The Google Pixel, which was wirelessly connected to gNB, did not move due to coverage constraints in our 5G frontend and, therefore, experienced stable channel quality throughout the duration of the experiment. To mimic 5G link rate fluctuations, we modified BWP every ≈ 10 seconds, as reported in Table I, and as shown in Fig. 3d by the dashed yellow line representing the maximum link rate.[1]

A collection of experimental results is illustrated in Fig. 3. In particular, Fig. 3a shows box plots of the round-trip 5G network latency. With adaptation, the average and median are approximately 40 ms, while the interquartile range is ≈ 30 ms. On the other hand, without adaptation, the network latency is consistently higher, with an average of 80 ms and a

Authorized licensed use limited to: New York University. Downloaded on September 10,2024 at 16:42:20 UTC from IEEE Xplore. Restrictions apply.



(a) Bars plot with the PMF of the total number of objects detected per frame.

(b) Bars plot with the PMF of the number of *chairs* detected per frame.

Fig. 4: Object detection results for the video streaming experiment on our platform.

TABLE I: Main parameters for the 5G OAI experiments. Relatively to the TDD configuration, **D** indicates a downlink slot, **U** stands for uplink, and **S** is a special slot with a mixture of 6 downlink symbols and 4 uplink symbols.

Parameter	Value	Parameter	Value
Frequency carrier	2593.350 MHz	BWP 1 channel bw	40 MHz
TDD configuration	DSUUU	BWP 2 channel bw	20 MHz
Max encoder bitrate	20 Mbps	BWP 3 channel bw	10 MHz
Video resolution	1920x1080	BWP 1, 2, 3 SCS	30 KHz

median of ≈ 70 ms. Besides, the interquartile range and the maximum round-trip are higher. This indicates that without adaptation, latency becomes unpredictable, leading to higher jitter values and worse video quality. Fig. 3b suggests that the object detection inference time does not vary significantly in the two cases, as both average values are close (i.e., roughly 20 ms). The average inference time is slightly lower without adaptation because fewer candidates are detected given the lower average video quality. Fig. 3c shows the cumulative distribution function (CDF) of the end-to-end inference latency computed as the sum of the round-trip network latency and the average object detection inference time. As we can see, with adaptation, we can clearly satisfy the 100 ms QoS requirement (as in [6]) for object detection all the time. Instead, without adaptation, 40% of the time, we cannot meet the requirement. The results show that our adaptive video streaming platform can provide better object detection performance compared to an implementation without adaptation.

Fig. 3d shows the predicted channel link rate and the adapted encoder bitrate during the video adaptation experiment. Note that the encoder rate without adaptation would be fixed to the maximum value of 20 Mbps. We could transmit the video at a lower bitrate (i.e., underestimating the channel condition); nonetheless, this will result in consistently degraded performance for all the microservices due to poor image quality. The figure shows that the encoder bitrate closely follows the predicted link rate. In addition, the proximity between the prediction and the actual channel link rate depends on the amount of data sent on the uplink. As such, when the encoder bitrate is capped at its maximum value of 20 Mbps, the predicted link rate is consistently lower compared to the actual maximum channel link rate.

YoLoV7 [9] can track objects from 80 classes; however, in this experiment, we considered a subset of eight typical objects that can be found in an office environment. From the probability mass function (PMF) of Fig. 4a, we can see that a higher number of objects per frame is detected with adaptation. Moreover, from Fig. 4b, more chairs can be detected with adaptation. Without adaptation, there is a non-negligible probability that no objects or chairs are detected. Therefore, video adaptation helps to increase our VIS⁴ION wearable object detection capabilities.

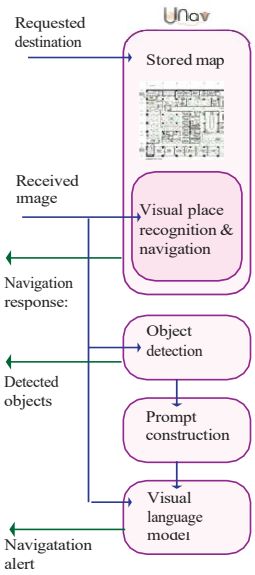
We mapped the entire 9th floor at 370 Jay Street for navigation and localization. We deployed InstructBLIP as in [12] for vision-language processing based on VLM. Table II shows the importance of video adaptation to allow for precise inference in each microservice. In the case of adaptation, the video frame depicted is received from the low-frame rate, high-resolution secondary video stream, which is activated when the predicted link rate falls to 5 Mbps. Highlighted in yellow in the table, we have the output of UNav with and without adaptation based on the corresponding video frame as input. As can be seen, with adaptation, the output corresponds precisely to the correct navigation information to reach the destination; without adaptation, due to poor image quality, UNav cannot localize the user, thus not finding any path to the destination.

Highlighted in cyan in Table II, we show the results of the VLM inference. In the case of adaptation, YoLo recognizes the *chair* in the frame; hence, we can use object detection results to create a specific prompt to input into the model, as reported in the table. With adaptation, InstructBLIP can assess the risks associated with the detected object along the path of pBLV. However, without adaptation, we are forced to ask a more generic question to the model since the chair is not detected, which ultimately cannot capture the risk associated with the obstacle along the route. This is just one possible usage of the VLM assistance microservice (assessing environmental risks and complementing object detection). Other possible usages include, as an example, scene description.

[1]Note: the maximum link rate is lower than the capacity expected for each BWP because the uplink performance is limited by the computational resources of our Nautilus server. To overcome this limitation, we are integrating FPGA-based LDPC acceleration enabled by the Xilinx T1 card.

Authorized licensed use limited to: New York University. Downloaded on September 10,2024 at 16:42:20 UTC from IEEE Xplore. Restrictions apply.

TABLE II: Navigation and localization and VLM inference results.

Computational flow at edge	Item	Adaptation	No Adaptation
	Requested destination	"Kitchen room of 9th floor at 370 Jay St."	
	Received image at the edge along with detected objects by Yolo running at the edge		
	Detected objects	Chair	None
	Navigation response to user	"Go straight to 12 o'clock and walk 15 feet. Then turn right."	"No path to destination"
	Engineered prompt (automatically generated, not provided by the user)	"There appears to be a chair in the scene. Is the chair a threat to a user with blindness who wants to walk forward? If yes, in which direction should the user go to avoid it? (avoid mentioning blindness in your answer)"	"Is there any object that can be a threat to a user with blindness who wants to walk forward? If yes, in which direction should the user go to avoid it?" (avoid mentioning blindness in your answer)
	Alert given to user	"Yes, the chair is a threat to a user who wants to walk forward. To avoid the chair, the user should go to the left side of the hallway. The chair is located on the right side of the hallway."	"No, there is no object that can be a threat to a user who wants to walk forward in this image. The user can simply walk forward without any obstacles."

V. Conclusions

The adaptive video streaming platform presented in this paper aims to increase the performance of an ensemble of microservices based on AI offered to VIS4ION, a smart wearable that helps pBLV in their daily challenges. To achieve that, we propose to (1) leverage edge computing resources to increase performance and minimize latency along with battery consumption, and (2) utilize adaptive video streaming strategies that maximize the quality of the video frames fed into the AI-based microservices. This framework has been validated using real-world experiments. To do so, we built a testbed using OAI, which is an open-source implementation of 5G. This platform allowed us to derive real-world performance metrics such as end-to-end latency and visual processing accuracy. Overall, our results show the benefits of our proposed architecture both in terms of accuracy and aggregate latency – network plus inference–, which are particularly critical in the context of assistive technology.

References

[1] WHO, "Visual impairment and blindness. WHO fact sheet, 282. Geneva." 2010.

[2] R. R. Bourne, G. A. Stevens, R. A. White, J. L. Smith, S. R. Flaxman, H. Price, J. B. Jonas, J. Keeffe, J. Leasher, K. Naidoo *et al.*, "Causes of vision loss worldwide, 1990–2010: a systematic analysis," *The lancet global health*, vol. 1, no. 6, pp. e339–e349, 2013.

[3] C. E. Sherrod, S. Vitale, K. D. Frick, and P. Y. Ramulu, "Association of vision loss and work status in the united states," *Jama Ophthalmology*, vol. 132, no. 10, pp. 1239–1242, 2014.

[4] A. Boldini, A. L. Garcia, M. Sorrentino, M. Beheshti, O. Ogedegbe, Y. Fang, M. Porfiri, and J.-R. Rizzo, "An inconspicuous, integrated electronic travel aid for visual impairment," *ASME Letters in Dynamic Systems and Control*, vol. 1, no. 4, p. 041004, 2021.

[5] A. Boldini, J.-R. Rizzo, and M. Porfiri, "A piezoelectric-based advanced wearable: obstacle avoidance for the visually impaired built into a backpack," in *Nano-, Bio-, Info-Tech Sensors, and 3D Systems IV*, vol. 11378. International Society for Optics and Photonics, 2020, p. 1137806.

[6] Z. Yuan, T. Azzino, Y. Hao, Y. Lyu, H. Pei, A. Boldini, M. Mezzavilla, M. Beheshti, M. Porfiri, T. E. Hudson, W. Seiple, Y. Fang, S. Rangan, Y. Wang, and J.-R. Rizzo, "Network-aware 5g edge computing for object detection: Augmenting wearables to "see" more, farther and faster," *IEEE Access*, vol. 10, pp. 29 612–29 632, 2022.

[7] L. Liu, H. Li, and M. Gruteser, "Edge assisted real-time object detection for mobile augmented reality," in *The 25th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '19, 2019.

[8] E. Kurdoglu, Y. Liu, Y. Wang, Y. Shi, C. Gu, and J. Lyu, "Real-time bandwidth prediction and rate adaptation for video calls over cellular networks," in *Proceedings of the 7th International Conference on Multimedia Systems*, ser. MMSys '16, 2016.

[9] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv preprint arXiv:2207.02696*, 2022.

[10] A. Yang, M. Beheshti, T. E. Hudson, R. Vedanthan, W. Riewpaiboon, P. Mongkolwat, C. Feng, and J.-R. Rizzo, "UNav: An Infrastructure-Independent Vision-Based Navigation System for People with Blindness and Low Vision," *Sensors*, vol. 22, no. 22, p. 8894, 2022.

[11] S. Lowry, N. Su¨nderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual place recognition: A survey," *ieee transactions on robotics*, vol. 32, no. 1, pp. 1–19, 2015.

[12] W. Dai, J. Li, D. Li, A. M. H. Tiong, J. Zhao, W. Wang, B. Li, P. Fung, and S. Hoi, "Instructblip: Towards general-purpose vision-language models with instruction tuning," 2023.

[13] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, and E. P. Xing, "Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality," March 2023. [Online]. Available: https://lmsys.org/blog/2023-03-30-vicuna/

[14] A. Yang, Y. Wang, J.-R. Rizzo, and C. Feng, "Distillation improves visual place recognition for low-quality queries," 2023.

[15] GStreamer, "GStreamer Library," https://gstreamer.freedesktop.org/.

[16] NVIDIA, "DeepStream SDK," https://developer.nvidia.com/deepstream-sdk.

[17] F. Kaltenberger, A. P. Silva, A. Gosain, L. Wang, and T.-T. Nguyen, "Openairinterface: Democratizing innovation in the 5g era," *Computer Networks*, vol. 176, p. 107284, 2020.

[18] 3GPP, "Release 15," https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3389.

Authorized licensed use limited to: New York University. Downloaded on September 10,2024 at 16:42:20 UTC from IEEE Xplore. Restrictions apply.