engineering proceedings

MDPI

Proceeding Paper

On-Device Automatic Speech Recognition for IIoT and Extended Reality Industrial Metaverse Applications †

Antón Valladares-Poncela [1,2,3,*], Paula Fraga-Lamas [1,2,3] and Tiago M. Fernández-Caramés [1,2,3]

[1] Department of Computer Engineering, Faculty of Computer Science, Universidade da Coruña, 15071 A Coruña, Spain; paula.fraga@udc.es (P.F.-L.); tiago.fernandez@udc.es (T.M.F.-C.)
[2] Centro Mixto de Investigación UDC-Navantia, Universidade da Coruña, Edificio de Batallones, s/n, 15403 Ferrol, Spain
[3] Centro de Investigación en Tecnologías de la Información y las Comunicaciones (CITIC), Universidade da Coruña, 15701 A Coruña, Spain
* Correspondence: anton.valladares@udc.es
† Presented at The 11th International Electronic Conference on Sensors and Applications (ECSA-11), 26–28 November 2024; Available online: https://sciforum.net/event/ecsa-11.

Abstract: This paper presents a comprehensive study on enhancing Industrial Internet of Things (IIoT) and Industrial Metaverse applications through the integration of On-Device Automatic Speech Recognition (ASR) using Microsoft HoloLens 2 smart glasses. Specifically, this paper focuses on the utilization of the HoloLens 2 microphone array and sound capture APIs to benchmark the performance and accuracy of on-device ASR models. The evaluation of these models includes metrics such as Character Error Rate (CER), Word Error Rate (WER) and latency. In addition, this paper explores various optimization techniques, including quantization tools and model refinement strategies, aimed at minimizing latency while maintaining high accuracy. This study also emphasizes the importance of supporting low-resource languages, using Galician—a language spoken by less than 3 million people worldwide—as a case study. By benchmarking different variations of a Wav2Vec2.0-based ASR model fine-tuned for Galician, the most effective models are identified, as well as their optimal runtime configurations. This work underscores the critical role of low-latency on-device ASR systems in real-time IIoT and Industrial Metaverse applications, highlighting how these technologies can enhance operational efficiency, privacy and user experience in industrial environments. The findings demonstrate the significant potential of the on-device ASR system developed to enhance voice interactions in emerging Metaverse applications, specially for low-resource languages.

Keywords: Automatic Speech Recognition; ASR; Internet of Things; IIoT; Industrial Metaverse; Microsoft HoloLens 2; Extended Reality

check for updates

Citation: Valladares-Poncela, A.; Fraga-Lamas, P.; Fernández-Caramés, T.M. On-Device Automatic Speech Recognition for IIoT and Extended Reality Industrial Metaverse Applications. *Eng. Proc.* **2024**, *82*, 3. https://doi.org/10.3390/ecsa-11-20466

Academic Editor: Francisco Falcone

Published: 26 November 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The emergence of Extended Reality (XR) technologies has revolutionized various sectors by integrating devices like the Microsoft HoloLens 2 smart glasses into professional environments, ranging from education to industrial applications [1–4]. These advancements are crucial for the development of the Industrial Metaverse [5], a collective virtual space where users interact via XR devices, combining digital twins and Industrial Internet of Things (IIoT) devices to optimize industrial processes. This paper presents results from a research project with Navantia, a Spanish naval company specialized in designing and constructing military and civilian ships. The objective of the work is to develop an Industrial Metaverse application utilizing XR technology to enhance the speed and efficiency of operators during the placement and installation of electrical boilermaking components during the shipbuilding process. In order to enhance the user experience of the Industrial Metaverse application, the optimization of On-Device Automated Speech Recognition

(ASR) is essential [6]. The developed application makes use of Microsoft HoloLens 2 smart glasses, which embed a sophisticated microphone array that facilitates voice interactions, making the optimization of audio capture crucial for achieving a clear communication and an effective command execution [7]. Specifically, this paper focuses on leveraging the HoloLens 2 capabilities to deploy a fine-tuned Wav2Vec2.0-based ASR model directly on the device [8]. Thus, the proposed approach is aimed at minimizing latency while maintaining privacy (the ASR model is executed locally on the smart glasses, without communicating with external servers), which is key for performing real-time interactions in the Industrial Metaverse application developed for Navantia. At the same time, by focusing on a low-resource language like Galician, spoken by less than 3 million people [9], this paper highlights both the lack and importance of ASR systems for minority languages in IIoT and Industrial Metaverse applications.

2. State of the Art

ASR systems have become an essential component in various applications within the domains of IoT and IIoT. As it can be seen in Figure 1, these systems facilitate voice-based interaction, which is especially useful for environments where manual control is impractical. For instance, ASR has been integrated into smart home devices and industrial settings to improve user interaction with machines, enabling voice-based commands and control [10].

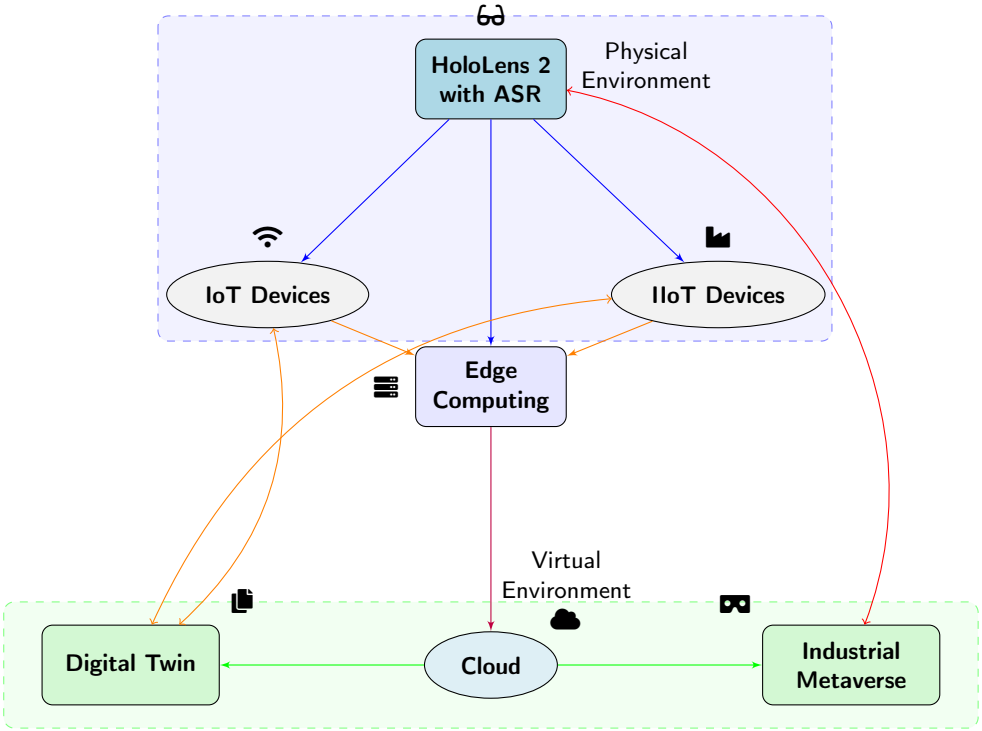

Figure 1. Representation of the integration of the on-device ASR system in the Industrial Metaverse as well as the interaction with IoT and IIoT devices.

Recent advancements have also extended ASR technologies into the realm of the Metaverse with the integration of ASR in virtual environments, particularly in collaborative workspaces which allow users to interact seamlessly with digital objects and avatars through voice commands. However, the development of ASR for the Metaverse faces challenges like real-time processing, speech recognition in noisy environments, and the need for high accuracy in voice command recognition [11].

Developing ASR systems for low-resource languages such as Galician poses several challenges due to the scarcity of annotated speech data. Traditional ASR models, which typically require large datasets, often struggle with languages that lack sufficient linguistic resources. Recent research has focused on developing ASR models based on multilingual

frameworks that can be fine-tuned with smaller datasets, enabling the deployment of ASR systems for low-resource languages [8].

3. Methodology

3.1. HoloLens 2 Microphone Array and APIs

As it can be seen in Figure 2, the Microsoft HoloLens 2 smart glasses feature an array of five microphones designed for optimal audio capture. Three are located on top of the visor, while another two are inside, aimed towards the user's mouth, thus facilitating effective voice isolation and background noise cancellation. Three main Application Programming Interfaces (APIs) are currently available for capturing sound with HoloLens 2: Unity API [12], MRTK's MicStreamSelector.dll [13], and Windows Runtime (WinRT) API [14]. Each API provides different levels of control and processing capabilities.

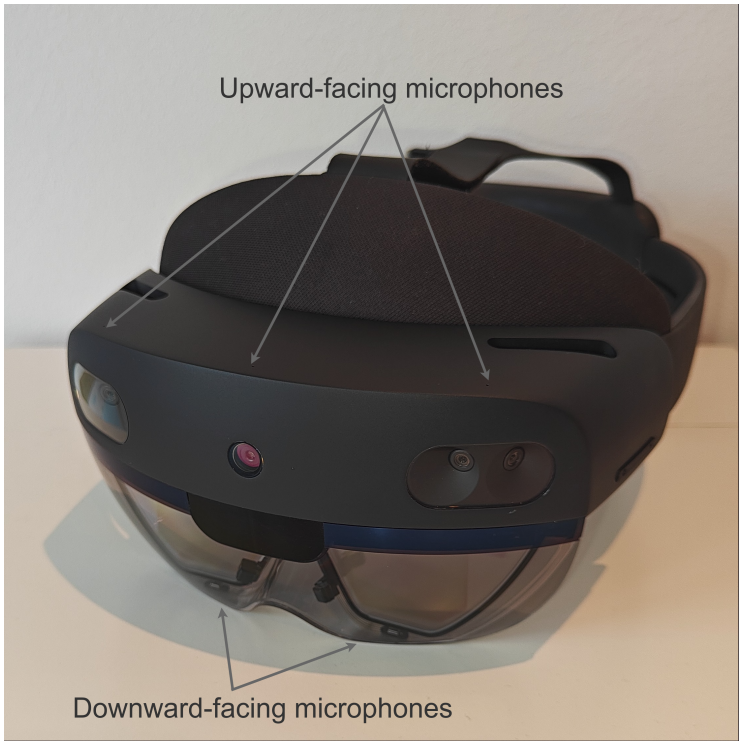

Figure 2. Microsoft HoloLens 2 microphone array.

3.2. Dataset and Benchmarking Procedure

The dataset used for benchmarking in this paper comprises 60 clips, with 10 clips representing a male voice and 10 clips representing a female voice recorded for each of the 3 available sound capture APIs. Each clip represents a single voice command in Galician of up to five seconds and is based on phrases related to IIoT, sensor-based and voice-controlled applications (e.g., 'apaga a luz', which means 'turn off the lights').

The benchmarking process uses ONNX Runtime's CPU Execution Provider (EP) [15] and iterates over the 60 clips and over every variation of the ASR model, trying every possible combination of runtime parameters in the CPU EP to achieve the best results for that single clip. Every inference result for each combination is stored for further analysis. The benchmarking process is *single-pass*, which means that the model is not warmed up before the benchmarking process, so it has not seen any of the data it will be fed, and will only see each sample from the dataset once during the benchmarking process. This is performed in order to simulate a real-world scenario where the model is loaded and executed without any prior knowledge of the input data.

3.3. ASR Model and Optimization Techniques

The base ASR model used in this paper is a Wav2Vec2.0-based model fine-tuned for Galician. This model is based on the Wav2Vec 2.0 architecture which is a state-of-the-art ASR model that uses a self-supervised learning approach to learn speech representations from raw audio data [16]. Additionally, a larger version of the model was also used in the benchmarking process.

In addition to the base and large models, this paper also tests a distilled version. Distillation is a technique that consists of training a smaller model to mimic the behavior of a larger model. The distilled model strives to maintain the accuracy of the base model while reducing its size and computational requirements.

Quantization is another optimization technique used in this paper. Quantization reduces the precision of a model's weights and activations, thus reducing the model's size and computational requirements. In this paper, two quantization tools are used: Optimum 1.23.3 [17] and PyTorch 2.5.1 [18].

PyTorch is one of the most popular deep learning frameworks, and its quantization tool is widely used for quantizing models; however, it is not optimized for hardware-aware quantization. Optimum, on the other hand, is a quantization tool developed by Hugging Face that takes into account the target hardware when quantizing the model, thus exploiting the hardware capabilities to achieve better performance.

3.4. Evaluation Metrics: CER, WER and Latency

To evaluate the performance of the developed on-device ASR system, several metrics were considered: Character Error Rate (CER), Word Error Rate (WER) and latency.

The CER and WER metrics measure the percentage of units (characters for CER, words for WER) that are incorrect in a transcribed output, as follows:

$$\text{Error Rate} = \left(\frac{\text{Substitutions} + \text{Insertions} + \text{Deletions}}{\text{Total units in reference}} \right) \times 100 \quad (1)$$

Since in an IIoT or Industrial Metaverse ASR-based application, the developed system has to deal with short voice commands, the use of CER is commonly prioritized over WER. This is due to the fact that in very short commands, WER can be misleading, as a single character error can significantly affect the WER value, while the CER value remains more stable and representative of the actual accuracy of the ASR system.

Latency is a challenging aspect to address when performing the processing and transcription on a computationally constrained battery-powered device. Achieving low latencies in an on-device ASR system requires precise tweaking, optimization and efficient resource utilization.

In Metaverse and IIoT applications, low latency is crucial for providing a seamless user experience as it allows for real-time interactions both with the application and with the environment [19]. In the context of ASR, low latency is essential for providing immediate feedback, ensuring that the system feels responsive to the user.

Latency measured in this paper is the time taken by the ASR system to process a single 5 s voice command and return the transcription. As represented in Figure 3, latency is measured from the moment the audio is encoded into the input tensor until the moment the Connectionist Temporal Classification (CTC) decoding process is completed and the transcription is returned, allowing the system to execute the desired command.

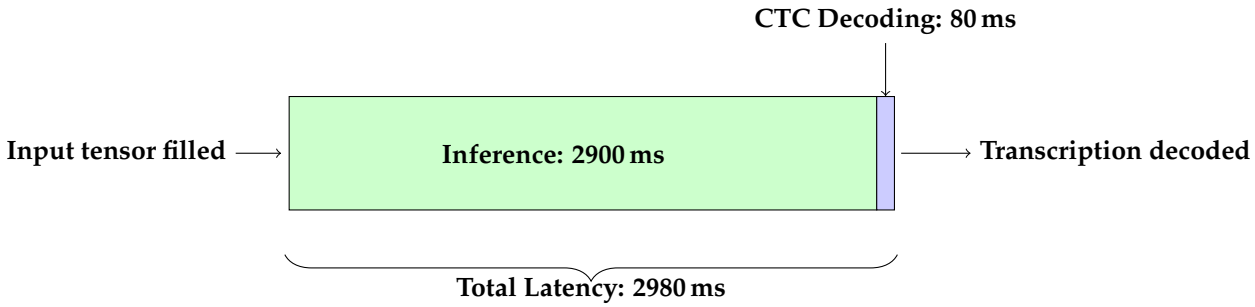

Figure 3. Latency breakdown of the on-device ASR system.

4. Benchmarking and Results

As it was mentioned in Section 3.2, the benchmarking process involved iterating over 60 clips, each representing a unique voice command that allows for interacting with an IIoT system through an Industrial Metaverse application for Microsoft HoloLens 2. For each clip, every possible combination of runtime parameters for ONNX Runtime's CPU EP was tested. This comprehensive approach allowed us to identify the optimal runtime settings for each ASR model variation, depending on whether the executed task required higher accuracy, lower latency or a balance between both.

Figure 4 shows the results of the benchmarking process, where each point represents a specific ASR model variation with a specific set of runtime parameters. The points have been clustered based on the used quantization tool and on whether the model was distilled or not (Base vs. Distilled).

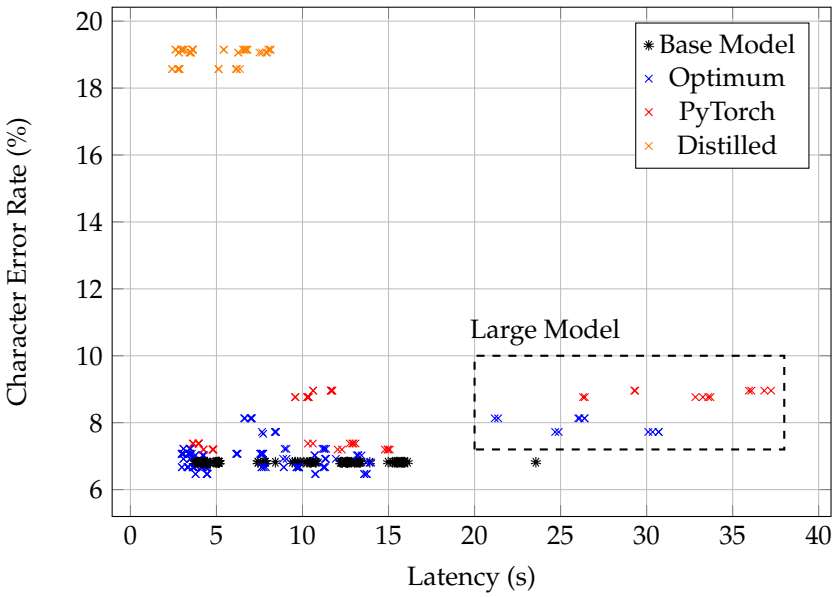

Figure 4. CER vs. Latency for different ASR model variations, clustered by quantization tool, model base or distillation process.

As it can be observed, non-quantized *wav2vec2-base-gl* based models provide a reasonable trade-off between accuracy and latency, having a mean CER of 6.8 % and a mean latency of 4.5 s when using the optimal runtime parameters. Regarding the *wav2vec2-large-gl* based models, they incur a high latency (of more than 20 s) and do not deliver a significant improvement in CER. Lastly, the *wav2vec2-distilled-gl* based models obtain the lowest latencies, but at the cost of a high CER, which averages to 19.1 %.

On the other hand, the models quantized dynamically using Optimum for ARM64 provide a significant improvement in latency when running them under optimal runtime parameters, with latencies as low as 2.98 s, which is approximately a 40 % improvement

over the base models. Moreover, the lowest CER obtained by these models under optimal runtime parameters was 6.46 %, thus slightly improving the base models. Lastly, the models quantized using PyTorch tend to have a lower latency in comparison to the base model, averaging a 5 % decrease in latency, but at the cost of increasing CER by approximately 6 %.

Table 1 shows a subset of the best performing ASR model variations, runtime parameters and quantization mechanisms for minimizing CER. The table shows that for the *wav2vec2-base-gl* model with the lowest CER, the optimal runtime parameters consist of setting *Graph Optimization* to *Disabled* or *Basic*, and *Execution Mode* to *Sequential*. This is because higher levels of graph optimization can introduce less precise computations in certain nodes. Moreover, in this specific case, the ASR model's computational graph does not benefit significantly from parallel execution due to its limited branching structure.

Table 1. Optimal combinations of ASR models, runtime parameters and quantization mechanisms for minimizing CER.

Model	Quantization					Runtime Parameters		CER (%)	Latency (s)
	Tool	Type	Data Type	Sym. Act.	Per Channel	Graph Optimization	Execution Mode		
wav2vec2-base-gl	Optimum	Dynamic	QUInt8/QInt8	No	No	Disabled	Sequential	**6.46**	3.78
wav2vec2-base-gl	Optimum	Dynamic	QUInt8/QInt8	No	No	Basic	Sequential	**6.66**	3.61
wav2vec2-base-gl	Optimum	Dynamic	QUInt8/QInt8	Yes	No	Basic	Sequential	**6.66**	3.65
wav2vec2-base-gl	Optimum	Dynamic	QUInt8/QInt8	Yes	Yes	Extended	Sequential	**6.67**	2.98
wav2vec2-base-gl	Optimum	Dynamic	QUInt8/QInt8	Yes	Yes	Disabled	Sequential	**6.70**	3.80
wav2vec2-base-gl	–	–	–	–	–	Disabled	Sequential	**6.76**	4.47
wav2vec2-base-gl	PyTorch	Dynamic	QInt8	No	No	Basic	Sequential	**7.19**	4.24
wav2vec2-large-gl	Optimum	Dynamic	QUInt8/QInt8	Yes	Yes	Disabled	Sequential	**7.66**	7.72
wav2vec2-large-gl	PyTorch	Dynamic	QInt8	No	No	All	Sequential	**8.76**	9.56
wav2vec2-distilled-gl	Optimum	Dynamic	QUInt8/QInt8	Yes	Yes	All	Sequential	**18.56**	2.42
wav2vec2-distilled-gl	–	–	–	–	–	All	Sequential	**19.14**	2.61

Regarding quantization mechanisms, the best-performing models in terms of CER are quantized dynamically using Optimum with the ARM64 preset and with *Data Type* set to *QUInt8/QInt8*, which means that activations are quantized using 8-bit unsigned integers, while weights use 8-bit signed integers. Weights are always quantized symmetrically in order to ease the computation of dot products.

Finally, Table 2 presents a subset of the best-performing ASR model variations, runtime parameters and quantization mechanisms for minimizing latency. The table shows that distilled models provide the lowest latency, but their CER is significantly higher than the one obtained by base models and, therefore, they are not recommended for scenarios where accuracy is important. In addition, Table 2 also indicates that base models quantized dynamically using Optimum for ARM64 provide the best trade-off between CER and latency, with latencies as low as 2.98 s and CERs of only 6.67 %. It can also be observed in this case that higher levels of graph optimization usually lead to lower latencies without impacting the CER significantly.

Table 2. Optimal combinations of ASR models, runtime parameters, and quantization mechanisms for minimizing latency.

Model	Quantization					Runtime Parameters		CER (%)	Latency (s)
	Tool	Type	Data Type	Sym. Act.	Per Channel	Graph Optimization	Execution Mode		
wav2vec2-distilled-gl	Optimum	Dynamic	QUInt8/QInt8	Yes	Yes	All	Sequential	18.57	**2.42**
wav2vec2-distilled-gl	–	–	–		–	All	Sequential	19.15	**2.61**
wav2vec2-base-gl	Optimum	Dynamic	QUInt8/QInt8	Yes	No	Extended	Sequential	7.07	**2.98**
wav2vec2-base-gl	Optimum	Dynamic	QUInt8/QInt8	Yes	Yes	Extended	Sequential	6.67	**2.98**
wav2vec2-base-gl	Optimum	Dynamic	QUInt8/QInt8	No	No	Extended	Sequential	6.92	**3.07**
wav2vec2-base-gl	PyTorch	Dynamic	QInt8	No	No	Extended	Sequential	7.38	**3.61**
wav2vec2-base-gl	–	–	–	–	–	Extended	Sequential	6.82	**3.77**
wav2vec2-base-gl	–	–	–	–	–	All	Sequential	6.82	**3.86**
wav2vec2-large-gl	Optimum	Dynamic	QUInt8/QInt8	Yes	Yes	Extended	Sequential	8.13	**6.61**
wav2vec2-large-gl	PyTorch	Dynamic	QInt8	No	No	All	Sequential	8.76	**9.56**

5. Conclusions

This paper analyzed the potential and viability of using on-device ASR models to enhance future sensor-based IIoT and Industrial Metaverse applications. Different ASR models for Galician, a low-resource language, were optimized to achieve low latency and high accuracy, making them ideal for real-time voice-controlled applications on mobile devices like Microsoft HoloLens 2 smart glasses.

The experiments showed that base models optimized for ARM64 provide an effective balance between latency and accuracy, achieving latencies as low as 2.98 s with a CER of 6.67 %. However, while distilled models offer lower latency, they result in higher error rates, making them unsuitable for scenarios where accuracy is crucial. This suggests that well-optimized base models are the most suitable for real-time processing in industrial environments.

Moreover, the implementation of ASR models for a minority language like Galician demonstrates the feasibility of developing such speech recognition solutions for low-resource languages. The results of this paper also highlight how on-device processing, independent of external servers, enhances privacy and also enables voice-controlled Metaverse applications in environments with limited or no Internet connectivity.

Author Contributions: Conceptualization, A.V.-P., P.F.-L. and T.M.F.-C.; methodology, A.V.-P., P.F.-L. and T.M.F.-C.; investigation, A.V.-P., P.F.-L. and T.M.F.-C.; writing—original draft preparation, A.V.-P., P.F.-L. and T.M.F.-C.; writing—review and editing, A.V.-P., P.F.-L. and T.M.F.-C.; supervision, P.F.-L. and T.M.F.-C.; project administration, T.M.F.-C.; funding acquisition, T.M.F.-C. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by Centro Mixto de Investigación UDC-NAVANTIA (IN853C 2022/01), funded by GAIN (Xunta de Galicia) and ERDF Galicia 2021-2027 and TED2021-129433A-C22 (HELENE) funded by MCIN/AEI/10.13039/501100011033 and the European Union NextGenerationEU/PRTR.

Institutional Review Board Statement: Not applicable

Informed Consent Statement: Not applicable

Acknowledgments: The authors would like to thank Aida Vidal-Balea for technical support in the development of the Extended Reality environment and Iván Froiz-Míguez for the initial training of the Galician ASR models.

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

Abbreviations

The following abbreviations are used in this manuscript:

IIoT	Industrial Internet of Things
IoT	Internet of Things
XR	Extended Reality
ASR	Automatic Speech Recognition
CER	Character Error Rate
WER	Word Error Rate
ONNX	Open Neural Network Exchange
API	Application Programming Interface
EP	Execution Provider
CTC	Connectionist Temporal Classification
QUInt8	Quantized Unsigned 8-bit Integer
QInt8	Quantized Signed 8-bit Integer

References

1. Choi, G.; Lee, S.; Roh, B.; Kang, J.; Kim, S. A design of safety and disaster response system with XR, IoT and LBS convergence. In Proceedings of the 2019 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 5–7 December 2019; pp. 1558–1559.
2. Jaivignesh, R.; Janarthanan, R.D.; Gnanalakshmi, V. Smart home automation using Augmented Reality and Internet of Things. *J. Phys. Conf. Ser.* **2022**, *2325*, 012003. [CrossRef]
3. Fraga-Lamas, P.; Fernández-Caramés, T.M.; Blanco-Novoa, Ó.; Vilar-Montesinos, M.A. A Review on Industrial Augmented Reality Systems for the Industry 4.0 Shipyard. *IEEE Access* **2018**, *6*, 13358–13375. [CrossRef]
4. Adebowale, O.; Agumba, J. Applications of Augmented Reality for construction productivity improvement: A systematic review. *Smart Sustain. Built Environ.* **2022**, *13*, 479–495. [CrossRef]
5. Fernández-Caramés, T.M.; Fraga-Lamas, P. Forging the Industrial Metaverse-Where Industry 5.0, Augmented and Mixed Reality, IIoT, Opportunistic Edge Computing and Digital Twins Meet. *arXiv* **2024**, arXiv:2403.11312.
6. Choudhary, T.; Mishra, V.; Goswami, A.; Jagannathan, S. A comprehensive survey on model compression and acceleration. *Artif. Intell. Rev.* **2020**, *53*, 5113–5155. [CrossRef]
7. Atal, B. Automatic recognition of speakers from their voices. *Proc. IEEE* **1976**, *64*, 460–475. [CrossRef]
8. Froiz-Míguez, I.; Fraga-Lamas, P.; Fernández-Caramés, T.M. Design, Implementation, and Practical Evaluation of a Voice Recognition Based IoT Home Automation System for Low-Resource Languages and Resource-Constrained Edge IoT Devices: A System for Galician and Mobile Opportunistic Scenarios. *IEEE Access* **2023**, *11*, 63623–63649. [CrossRef]
9. Census on the Galician Language. Available online: https://www.lingua.gal/to-know/basic-data-on-galician-language (accessed on 12 June 2024).
10. Zembrzuski, M.; Jeon, H.; Marhula, J.; Beksa, K.; Sikorski, S.; Latkowski, T.; Bujnowski, P. *Automatic Speech Recognition Adaptation to the IoT Domain Dialogue System*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 215–226. Available online: https://link.springer.com/chapter/\10.1007/978-3-319-60438-1_22 (accessed on 25 September 2024).
11. Fernando, S.; Moore, R.K.; Cameron, D.; Collins, E.C.; Millings, A.; Sharkey, A.; Prescott, T. Automatic Recognition of Child Speech for Robotic Applications in Noisy Environments. *arXiv* **2016**, abs/1611.02695. Available online: https://arxiv.org/abs/1611.02695 (accessed on 25 September 2024).
12. Unity Microphone Scripting API, Unity Technologies. Available online: https://docs.unity3d.com/ScriptReference/Microphone.html (accessed on 12 June 2024).
13. Mixed Reality ToolKit, Microsoft Corporation. Available online: https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk3-overview/ (accessed on 12 June 2024).
14. Windows RunTime (WinRT) MediaCapture Class, Microsoft Corporation. Available online: https://learn.microsoft.com/en-us/uwp/api/windows.media.capture.mediacapture (accessed on 12 June 2024).
15. ONNX Runtime, Microsoft Corporation. Available online: https://onnxruntime.ai/ (accessed on 12 June 2024).
16. Baevski, A.; Zhou, H.; Mohamed, A.; Auli, M. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 12449–12460.
17. Huggingface Optimum, Huggingface. Available online: https://huggingface.co/docs/optimum/en/index (accessed on 12 June 2024).
18. PyTorch Quantization, PyTorch. Available online: https://pytorch.org/docs/stable/quantization.html (accessed on 12 June 2024).
19. Attig, C.; Rauh, N.; Franke, T.; Krems, J. *System Latency Guidelines Then and Now–Is Zero Latency Really Considered Necessary*? Springer: Berlin/Heidelberg, Germany, 2017; pp. 3–14.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.