IMET - International Conference on Interactive Media, Smart Systems and Emerging Technologies (2023)
N. Pelechano, F. Liarokapis, D. Rohmer, and A. Asadipour (Editors)

Short Paper

A Real-time Voice Interface for Intelligent Wheelchairs

S. Moschopoulos[2], I. Fudos[1,2], K. Koritsoglou[2], G. Tatsis[2] and D. Tzovaras[2]

[1]University of Ioannina, Computer Science & Engineering Department, Greece
[2]Information Technologies Institute, Centre for Research & Technology Hellas (CERTH), Greece

Abstract

This paper reports on the development of a real-time voice interface for navigation purposes of electric wheelchairs. To this end, we employ a convolutional neural network trained and fine-tuned using a small dataset that consists of Greek commands. Furthermore, the study explores a highly quantized version of the network to achieve computational efficiency while maintaining high accuracy on an edge device. The experimental results confirm the effectiveness of the model in accurately detecting keywords in real time with minimal misclassifications.

CCS Concepts

• Computing methodologies → ***Speech recognition;*** *Supervised learning by classification;* ***Transfer learning; • Computer systems organization*** → ***Real-time system architecture; • Hardware*** → ***Hardware accelerators;***

1. Introduction

Keyword spotting (KWS) plays a crucial role in enabling speech-based user interactions in smart devices, especially in assistive technologies such as electric wheelchairs. It requires a real-time response, high accuracy, and efficient implementation on resource-constrained edge devices. Several studies have addressed the development of voice-controlled wheelchairs using speaker-dependent voice recognition modules on microcontrollers to navigate the wheelchair [Abe15], [HYC20]. Also, speaker-independent systems using artificial intelligence have been employed, achieving high accuracy rates but necessitating significant computing resources. Consequently, these systems often utilize devices such as Android mobile phones or laptops to handle the speech recognition part [Bak22], [KSZB22]. The application of KWS in the context of IoT and edge devices with limited memory and computation resources has been studied in [Dha21], [SAL*22]. Efforts have been made to study and optimize KWS solutions for energy efficiency, considering both hardware [MW22] and neural network architectures [ZSLC18], [SZK*20], [ZT21]. Recent advances in wheelchairs have proposed obstacle detection systems [TKF*22]. Such systems can be used in conjuction with the method proposed in this paper and other interfaces for an integrated, safe, low power consumption intelligent wheelchair. This study extends existing research by introducing the integration of a Convolutional Neural Network (CNN) for real-time keyword spotting on an edge device, for navigating intelligent wheelchairs.

2. Methodology

The implementation of a real-time voice interface holds great potential in enhancing the overall user experience for individuals using electric wheelchairs. In this research, we demonstrate the efficiency of the method on voice commands from the Greek language without loss of generality. To establish a Greek dataset, a set of three words was recorded from several users. Subsequently, a keyword-spotting classifier was pre-trained using a broader set of classes from the English language. It was then fine-tuned on our dataset and quantized so that it can be deployed on a hardware accelerator integrated within the wheelchair.

2.1. Google Speech Commands

Google Speech Commands (GSC) v0.01 [War18] is a widely used and publicly available dataset designed for keyword spotting and speech recognition tasks. Version 0.01 consists of a collection of short audio recordings, each containing a single spoken word. The dataset comprises 30 different categories of words and includes a total of 64,727 files in 16-bit PCM WAV format, with a sample rate of 16 kHz and a total duration of approximately 18 hours. For this particular study, two specific classes from the dataset were selected: ‘Marvin’ which represents a name used for activation/deactivation, and ‘Stop’ which has universal usage. The remaining classes in the dataset were used to pre-train the network.

2.2. Main Dataset

Three specific words were collected as they indicate basic instructions about the direction of movement for an electric wheelchair. The collection process involved the Greek words: ‘Αριστερά’ (aristerá), for ‘left’, ‘Δεξιά’ (dexiá), for ‘right’, and ‘Πάμε’ (páme), for ‘go’. To record the corresponding audio signals from multiple users, an audio recording application was developed and

DOI: 10.2312/imet.20231250

compiled as a standalone executable to allow participants to use it without any installation process. The recording application was distributed to 25 participants, including 9 women and 16 men aged 22 to 68 years. Each participant received detailed instructions to record each word approximately 20 to 25 times, ensuring a diverse and comprehensive dataset. In total, the dataset consisted of 550 samples for the word 'aristerá', 553 samples for the word 'dexiá' and 557 samples for the word 'páme'. This distribution ensured a balanced representation of each keyword. The logging properties of the GSC dataset were retained to maintain consistency and compatibility. Each recorded sample had a duration of 1 second with a frequency of 16kHz and was stored in 16-bit PCM WAV format. This integrated set, combined with the 'Stop' and 'Marvin' classes (930 and 1,146 samples respectively), was used for the overall training and evaluation of the model. Also, a class with 'Undefined' signals was added, which contains ambient sounds [SJB14] of cafeterias, metro stations, traffic, and meetings. To enhance the network ability to detect unknown sounds in a continuous signal stream during real-time keyword spotting, the 'Undefined' category was populated with a larger number of samples (1,929 samples) as compared to the other categories (see Table 1). The test set comprises samples from users who were not part of the training set.

	Aristerá	Dexiá	Marvin	Páme	Stop	Undefined
Training	467	467	996	471	780	1794
Testing	83	86	150	86	150	135

Table 1: *Train and Test set population.*

2.3. Preprocess

During the preprocess all signals are normalized in the interval [-1, 1]. To enhance the robustness and generalization capability of the trained network, augmentation techniques were applied to the three classes derived from the Greek data collection. For each class, 100 samples were randomly selected from the training set, and one of the four augmentation procedures described below was randomly applied to each one of them.

The first technique involved adding Gaussian noise to the original sound signal. The intensity of the noise was kept small enough to avoid significant alterations to the signal of interest. The second technique was pitch shifting, where the signal was shifted upwards by 2 semitones, achieving a different tonal variation of the sound signal. The third technique was speed perturbation in the time domain, which shortened the duration of the original signal without losing any part of it. Lastly, the fourth technique involved adding random ambient sounds to the signal of interest, enriching it with additional contextual information without sacrificing valuable data.

The subsequent preprocessing step was applied to each signal in the dataset. To prepare the signals for the convolutional network, the Short-Time Fourier Transform (STFT) was employed [NNK22]. STFT breaks down the longer signal into shorter segments using overlapping windows and applies Fourier Transform to each segment. The resulting magnitude spectra from each segment are combined to form a spectrogram, which captures the frequency content variations over time. In this study, the spectrograms have final dimensions 124 and 129 on X-axis and Y-axis respectively on one channel.

2.4. Model Architecture

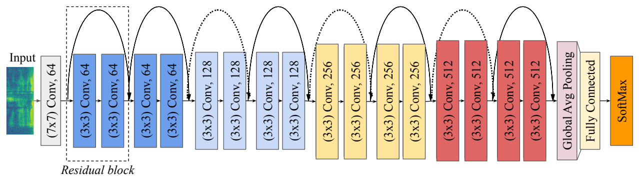

Figure 1: *ResNet 18*

ResNet18 is a convolutional neural network architecture consisting of 18 layers (Figure 1). It utilizes the concept of residual learning [HZRS16], incorporating shortcut connections to aid the learning of input information and extraction of features. The network starts with a convolutional layer with 64 filters and a kernel size of 7x7. The ReLU activation function and batch normalization are applied after each convolutional layer. The network further includes eight residual blocks consisting of two convolutional layers and a skip connection. All residual blocks use a kernel size of 3x3, while the filters are of size 2x64, 2x128, 2x256, and 2x512. A global average pooling layer reduces the dimension and passes the result to a fully connected layer. Finally, there is an output layer with SoftMax activation function.

2.5. Training and Transfer Learning

For training, we use the Adam optimizer and Categorical Crossentropy as the loss function. Initially, the ResNet18 was pre-trained on 28 out of the 30 classes from the GSC dataset for 50 epochs with a learning rate of 0.001. Thus, it learned to extract useful features from a complex and large training set, similar to ours. The accuracy of the network reached approximately 95% for the validation set, and the validation loss converged to a low value. These results indicate a well-trained network capable of accurately classifying data in the 28 categories. Subsequently, transfer learning was employed using the pre-trained model. The weights of the input layer and convolutional layers were retained, while the fully connected layer and output layer were replaced. A new fully connected layer and two dense layers were added, consisting of 128 and 6 neurons respectively, with ReLU and SoftMax activation functions. The network was then trained for an additional 100 epochs with a learning rate of 0.0001. This time, the focus was on adjusting only the weights of the newly added layers. This approach allowed us to fine-tune the network effectively for classifying correctly items in our 6-class dataset, reaching up to 99% accuracy for the validation set. For both training processes, 80% of the training set was used for training, while the remaining 20% was allocated for validation.

2.6. Hardware Accelerator

To achieve our goal of real-time keyword spotting for navigation purposes, we developed a complete system that can integrate into a wheelchair. This system combines a microcomputer with a specialized hardware accelerator, specifically the Coral USB Accelerator (`https://coral.ai/products/accelerator`) with an Edge

TPU coprocessor. Coral USB is highly efficient and offers fast execution of quantized models and is an ideal choice for convolutional networks [SAL*22]. For the microcomputer component, we selected the Raspberry Pi 4, which features an ARM Cortex-A72 processor, 4GB RAM, and low power consumption. Our hardware setup capitalizes on the strengths of the Coral USB Accelerator, offloading demanding computational tasks and thereby achieving notably low CPU and RAM usage on the Raspberry. This configuration not only ensures the efficient execution of the voice interface but also allows other operations to seamlessly run on the Raspberry.

Quantization is a technique used to reduce the requirements of models and make them suitable for resource-constrained devices [ZSLC18]. The size, memory usage, and computational requirements are reduced by representing the weights and activation functions of the network with lower precision, using 8-bit integer representations instead of 32-bit float. The reduction in the precision may lead to a slight loss in accuracy compared to the original full-precision model but the fast inference and the low power consumption make it a proper choice for real-time operations in devices with limited resources. The quantization of the ResNet18 was developed using TF lite by Tensorflow library and the Edge compiler given by Coral.

3. Results

3.1. ResNet18 Evaluation

Experiments were conducted to evaluate the network using both the unquantized version running on an Nvidia RTX 2060 Super GPU and the quantized version running on the Edge TPU. The reliability of each network was assessed individually for each class and collectively for all classes. The confusion matrices (Tables 2, 3) of the two models indicate that the quantized model may have lower accuracy compared to the original model. However, while there is a decrease in accuracy for the 'Undefined' class in the quantized model, the other critical categories show minimal or no misclassifications.

	Original ResNet18					
Aristerá	**82**	0	0	0	0	1
Dexiá	0	**86**	0	0	0	0
Marvin	0	0	**149**	1	0	0
Páme	0	0	0	**86**	0	0
Stop	0	0	0	0	**149**	1
Undefined	0	0	0	1	0	**134**
	Aristerá	**Dexiá**	**Marvin**	**Páme**	**Stop**	**Undefined**

Table 2: *Confusion Matrix of the testing set (ResNet18).*

	Quantized ResNet18					
Aristerá	**81**	0	0	0	0	2
Dexiá	0	**86**	0	0	0	0
Marvin	0	0	**148**	0	0	2
Páme	0	0	3	**82**	0	1
Stop	0	0	0	0	**148**	2
Undefined	0	0	0	1	11	**124**
	Aristerá	**Dexiá**	**Marvin**	**Páme**	**Stop**	**Undefined**

Table 3: *Confusion Matrix of the testing set (Quantized ResNet18).*

Table 4 presents the F1-score based on the confusion matrices, which is a metric that combines the Recall and Precision scores for each class and provides a balanced measure of the model's performance. Additionally, the accuracy percentage of the test set shows a small decrease of 2% for the quantized model, as expected. However, the smaller average inference time indicates fast processing in a system with real-time operations in resource-constrained devices.

	Original ResNet18	**Quantized ResNet18**
	F1 - score	
Aristerá	0.99	0.99
Dexiá	1.00	1.00
Marvin	1.00	0.98
Páme	0.99	0.98
Stop	1.00	0.96
Undefined	0.99	0.93
	Accuracy	
Total Test set	99%	97%
	Average Inference Time (seconds)	
Total Test set	0.0405	0.0149

Table 4: *F1-score of each class. Accuracy and Inference time for the total test set.*

3.2. Real-time Voice Interface

An application was developed to continuously listen to the user's audio signals and classify them using the quantized model. The output vector given by the network has integer values in [0, 255] which are normalized to [0, 1]. Each value of the output vector defines the probability that the input belongs to the corresponding class. For high accuracy and robustness, the lower bound of the classification probability is initialized to 0.9. Anything categorized with a lower probability or as 'Undefined' is considered by the application as 'Unclassified'. In real-time or near real-time signal processing, a system has to continuously analyze the incoming audio data and make immediate predictions. In this interface, a thread continuously records and sends one-second audio signals captured by the microphone to another thread that processes them. Overlapping sliding windows [PGK*09] are used to derive a decision for each new sound signal. Thus, for example, a new signal S_i is combined with the last 8,000 values of the previous S_{i-1} (Figure 2). This results in a vector of 20,000 elements where a window, spanning 16,000 elements, moves in steps of 1000 for five iterations. The network makes predictions for each window, and if a class of interest is identified in at least three out of the five windows, signal S_i is classified accordingly, and the wheelchair performs the corresponding action. If not, S_i remains unclassified, indicating that no keyword was detected, and the wheelchair does not perform any action.

The system's usage guidelines involve the utilization of the 'Marvin Páme' and 'Marvin Stop' commands, which serve to enable and disable the voice interface, respectively as well as to start and stop the wheelchair's movement. When the interface is enabled, users have the capability to employ the keywords 'Páme', 'Aristerá', 'Dexiá' and 'Stop' to control the intelligent wheelchair. From the moment that the user speaks a word of interest to the point where the wheelchair starts an action, it takes approximately 0.7 seconds (Figure 2). To assess the real-time performance of the interface, a user was instructed to generate keywords based on the mentioned usage rules. The user followed a specific routine consisting of the following steps: 'Marvin Páme' $\rightarrow$ 'Aristerá' $\rightarrow$ 'Dexiá'

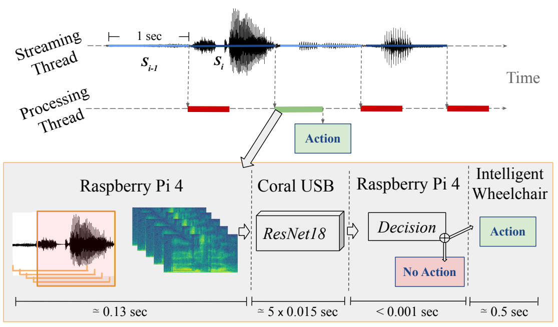

Figure 2: *Real-time work flow.*

→ 'Stop' → 'Páme' → 'Aristerá' → 'Dexiá' → 'Marvin Stop'. The procedure was repeated a total of 30 times (5 times per trial, 6 trials), and in cases where a keyword was not correctly classified, the user was asked to repeat it. The experimental results demonstrated successful real-time processing and accurate classification of the sound signals, achieving a high accuracy score for each class (Figure 3). Additionally, the findings indicated that the user gradually improved their interaction with the system over time, leading to fewer or no errors.

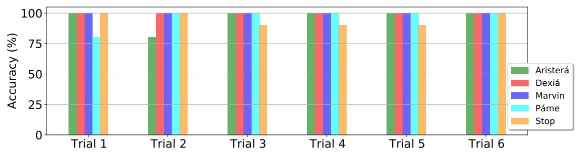

Figure 3: *Accuracy score of each class in 6 trials. Total accuracy: 'Aristerá' 96.67%, 'Dexiá' 100%, 'Marvin' 100%, 'Páme' 96.67%, 'Stop' 95%.*

4. Conclusions

This paper reports on the development and integration of a real-time voice interface to an intelligent wheelchair. The interface is designed to recognize five commands for navigation purposes. A quantized network is employed so as to facilitate the deployment of the voice interface on a hardware accelerator. The quantized model demonstrates high accuracy while maintaining low inference speed, contributing to the efficiency and effectiveness of the voice interface in real-time scenarios. In future work, the expansion of the dataset and incorporation of additional classes will be explored to enhance system accuracy and adaptability. Our goal is to seamlessly integrate the voice interface to the wheelchair controller along with the other navigation options (brain computer interface, joystick or smartphone app interface).

5. Acknowledgements

This research has been co-financed by the European Regional Development Fund of the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call Research—Create—Innovate (project code: T2EDK-02438).

References

[Abe15] ABED A.: Design of voice controlled smart wheelchair. *International Journal of Computer Applications 131* (Dec 2015), 32–38. doi:10.5120/ijca2015907235. 1

[Bak22] BAKOURI M.: Development of voice control algorithm for robotic wheelchair using min and lstm models. *Computers, Materials and Continua 73* (01 2022), 2441–2456. doi:10.32604/cmc.2022.025106. 1

[Dha21] DHAKSHINAMURTHY R.: Neural networks for keyword spotting on iot devices, 2021. arXiv:2101.00693. 1

[HYC20] HOU T. K., YAGASENA, CHELLADURAI: Arduino based voice controlled wheelchair. *Journal of Physics: Conference Series 1432*, 1 (jan 2020), 012064. doi:10.1088/1742-6596/1432/1/012064. 1

[HZRS16] HE K., ZHANG X., REN S., SUN J.: Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 770–778. doi:10.1109/CVPR.2016.90. 2

[KSZB22] KARANDE K., SOMANI S., ZOPE J., BHUSARI B.: Design and implementation of voice controlled wheelchair using matlab. *ITM Web of Conferences 44* (01 2022), 01003. doi:10.1051/itmconf/20224401003. 1

[MW22] MIAH M. N., WANG G.: Keyword spotting with deep neural network on edge devices. In *2022 IEEE 12th International Conference on Electronics Information and Emergency Communication (ICEIEC)* (07 2022), pp. 98–102. doi:10.1109/ICEIEC54567.2022.9835061. 1

[NNK22] NJOKU J., NWAKANMA C., KIM D. S.: Evaluation of spectrograms for keyword spotting in control of autonomous vehicles for the metaverse. In *Conference: Symposium of the Korean Institutes of Communications and Information Sciences (KICS)* (06 2022), vol. 78. 2

[PGK*09] PREECE S., GOULERMAS J., KENNEY L., HOWARD D., MEIJER K., CROMPTON R.: Activity identification using body-mounted sensors — a review of classification techniques. *Physiological measurement 30* (05 2009), R1–33. doi:10.1088/0967-3334/30/4/R01. 3

[SAL*22] SESHADRI K., AKIN B., LAUDON J., NARAYANASWAMI R., YAZDANBAKHSH A.: An evaluation of edge tpu accelerators for convolutional neural networks, 2022. arXiv:2102.10423. 1, 3

[SJB14] SALAMON J., JACOBY C., BELLO J.: A dataset and taxonomy for urban sound research. In *Proceedings of 22nd ACM International Conference on Multimedia* (11 2014), p. 1041–1044. doi:10.1145/2647868.2655045. 2

[SZK*20] SHUNZHI Y., ZHENG G., KAI Y., YUNGEN W., ZHENHUA H., ZHENG H.: Edgernn: A compact speech recognition network with spatio-temporal features for edge computing. *IEEE Access 8* (2020), 81468–81478. doi:10.1109/ACCESS.2020.2990974. 1

[TKF*22] TATSIS G., KORITSOGLOU K., FUDOS I., KARVOUNIS E., VOTIS K., TZOVARAS D.: Design and implementation of obstacle detection system for powered wheelchairs. In *2022 7th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)* (2022), pp. 1–4. doi:10.1109/SEEDA-CECNSM57760.2022.9932958. 1

[War18] WARDEN P.: Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition. *ArXiv e-prints* (Apr. 2018). arXiv:1804.03209. 1

[ZSLC18] ZHANG Y., SUDA N., LAI L., CHANDRA V.: Hello edge: Keyword spotting on microcontrollers, 2018. arXiv:1711.07128. 1, 3

[ZT21] ZONIOS C., TENENTES V.: Energy efficient speech command recognition for private smart home iot applications. In *2021 10th International Conference on Modern Circuits and Systems Technologies (MOCAST)* (2021), pp. 1–4. doi:10.1109/MOCAST52088.2021.9493392. 1