engineering proceedings

MDPI

Proceeding Paper

A TinyML Approach to Real-Time Snoring Detection in Resource-Constrained Wearables Devices [†]

Timothy Malche [1], Sumegh Tharewal [2,*] and Priti Maheshwary [3]

[1] Department of Computer Applications, Manipal University Jaipur, Jaipur 303007, Rajasthan, India; timothy.malche@jaipur.manipal.edu
[2] School of Advance Computing, DBS Global University, Dehradun 248001, Uttarakhand, India
[3] Department of Computer Science & Engineering, Rabindranath Tagore University, Bhopal 464993, Madhya Pradesh, India; pritimaheshwary@gmail.com
* Correspondence: sumegh.tharewal@dgu.ac.in or sumeghtharewal@gmail.com
[†] 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 study proposes a health monitoring system for snoring detection utilizing Tiny Machine Learning (TinyML) models, specifically designed for resource-constrained wearable Internet of Things (IoT) devices. This research addresses significant constraints associated with running machine learning models on IoT devices, such as latency, limited memory, and low computational resources. These parameters are essential for real-time monitoring in healthcare applications, where prompt response is critical. The research focuses on developing a TinyML model capable of identifying specific audio patterns related to snoring during sleep. Experimental evaluations conducted in real-world sleep environments with the TinyML model deployed on resource-constrained wearable IoT devices. The evaluation results show that the proposed model achieves high accuracy while utilizing minimal computational resources and without introducing latency issues. Through the integration of audio (Syntiant) and advanced audio preprocessing techniques, the proposed system improves the efficiency of the TinyML model on wearable devices. The quantized TinyML model achieved an accuracy of 95.85% with a low latency of 48 ms, utilizing only 17.0K of RAM and 34.07K of flash memory for real-time snoring classification. This study highlights the benefits of practical deployment of the TinyML model for snoring detection on resource-constrained wearable IoT devices, demonstrating that such models can operate effectively within the constraints of current wearable technology.

Keywords: wearable sensors; healthcare monitoring; Internet of Things; TinyML

check for updates

Citation: Malche, T.; Tharewal, S.; Maheshwary, P. A TinyML Approach to Real-Time Snoring Detection in Resource-Constrained Wearables Devices. *Eng. Proc.* **2024**, *82*, 55. https://doi.org/10.3390/ecsa-11-20352

Academic Editor: Stefano Mariani

Published: 25 November 2024

1. Introduction

Studies conducted on human lives reveal that close relationships are very important for sustaining happiness, more than money or anything else [1]. Snoring not only disrupts the snorer's sleep but can also foster bitterness and resentment between couples. Studies have also shown that almost 30–40% humans are habitual snorers [2]. When muscles around the throat relax during sleep, it narrows the airway, which results in vibration, due to which snoring happens [3]. Snoring is often related to symptoms of a sleep disorder. This emphasizes the importance of detecting and addressing snoring to enhance people's quality of life. This study aims to build a TinyML (Tiny Machine Learning)-based device dedicated to detecting and alerting when a person is snoring [4].

Building a snoring detection device has some challenges, ranging from technical hurdles to user-related considerations. Snoring patterns can also vary from person to person. Therefore, designing a device that can accurately capture and interpret different snoring voices is a big challenge. Environmental noise, such as ambient noise in the bedroom or external disturbances, can also impact accurate snore detection. Ensuring that the snoring device is comfortable for users to wear during sleep is also a main challenge. Building a machine learning model for resource-constrained devices, such as wearable devices, is also a major issue. The size and complexity of machine learning models can also directly impact the storage limitation of such wearable devices. Achieving real-time inference on resource-constrained devices is also challenging; audio signals have to be processed for snoring detection. On the other hand, the use of TinyML in building a wearable device for snoring detection offers many benefits, as follows:

- TinyML models are specifically designed for resource-constrained environments.
- TinyML models are optimized for real-time inference. They enable on-device inference and eliminate the need for continuous data transmission to external servers for analysis.
- TinyML models ensure real-time response because of low latency. In the context of snore detection, this capability is critical for providing timely alerts.
- TinyML models are characterized by their compact size, making them suitable for deployment in wearable devices.

2. Related Work

The study in [5] presents a novel snore detection algorithm which uses convolutional recurrent neural networks. The model is evaluated on audio data from 38 users while they slept. The system uses microphones installed in different places, and the algorithm achieved high accuracy (95.3%) in detecting snore events, with 92.2% sensitivity and 97.7% specificity. The performance of the algorithm remained robust across different microphone positions, which indicates the reliability of the system for snore detection in different sleep environments. Another study [6], focuses on the crucial need for a reliable snoring detection system for monitoring and diagnosing obstructive sleep apnea (OSA) to improve the quality of life for those with the disorder. This research proposes a hybrid convolutional neural network (CNN) model for detecting snores. In OSA monitoring in real-world situations, the model achieved an 89.3% average classification accuracy, 89.7% sensitivity, and 88.5% specificity. The research in [7] discusses the health hazards associated with obstructive sleep apnea hypopnea syndrome (OSAHS) and suggests a novel strategy for monitoring and identification to avoid treatment delays. The system classifies the snoring voices of normal individuals and those with OSAHS. Mel-frequency cepstral coefficients (MFCCs) are used in the study, along with CNN and LSTM models for feature extraction. The method has the greatest accuracy rate of 87% for binary categorization of snoring data. The suggested approach can also estimate the severity of OSAHS using the determined AHI value, providing useful information for clinical diagnosis and therapy. In [8], the authors discuss the challenges of detecting OSAHS by developing a snore detection system that enables at-home screening. The study suggests an approach for utilizing the hardware limitation of smartphones. The developed system detects snoring and environmental noises with using a real-time snore detector (RTSD) for sleep sound recordings. The RTSD serves as a valuable standalone tool for analyzing quality of sleep.

The research in [9] explores snoring issues and highlight their impact on health. The research aimed to develop a deep learning model implemented as an Android smartphone app for snore detection. The app analyzes real-time audio captured by the on-device microphone and classifies snore and non-snore voice and noise with an accuracy of 98%. The authors in [10] discussed an efficient method for snoring detection to diagnose OSA and health complications related to it. The study employs a CNN to distinguish between snoring and non-snoring voices and noises based on audio inputs. The raw data is preprocessed using MFCC, and multi-scale features are extracted from the frequency domain using a multi-branch CNN (MBCNN). The developed model achieves a snoring detection accuracy of 99.5%. The research in [11] addresses the issue of poor sleep quality in modern society and proposed a method to detect snoring and coughing episodes during sleep. The study discusses a three-stage method, consisting of the segmentation of nightly sound data into individual events, extraction of features from snoring and coughing episodes using

Fourier transforms, and recognition of these events using the Support Vector Machine (SVM) and the Hidden Markov Model (HMM). Experimental results demonstrate the effectiveness of the method in accurately detecting snoring and coughing events. The study presented in [12] proposes a low-cost alternative to polysomnography (PSG) for detecting OSA. A repository of snoring audio recordings is processed using multi-threshold endpoint detection and feature extraction to obtain distinctive information. Machine learning models are then trained to predict feature categories. A real-time system for an embedded device is developed to detect snoring and OSA. A multi-classification temporal convolutional network (TCN) is trained to distinguish between non-snoring, snoring noises, and OSA-related snoring. The model achieved a high detection accuracy of 96.7% regarding OSA-related snoring.

Based on the analysis of the research review, it is concluded that there is a need for the development of an efficient, real-time wearable device and system capable of accurately detecting snoring. The system should address the limitations of resource-constrained devices and eliminate dependency on cloud servers for inferencing. Through the creation of a wearable device that can accurately detect snoring in real-time without relying on external servers, people with OSA can benefit from continuous monitoring and timely intervention.

3. Methods

3.1. System Design

To detect snoring when a person is sleeping, a wearable device was designed using Arduino Nicla Voice [13]. The device contains built-in microphone to receive snoring sound and provide input to TinyML model that was built. The main features of the device are given in Table 1:

Table 1. Device configuration.

Microprocessor	Syntiant® NDP120 Neural Decision Processor™ (NDP) (Syntiant Corp., Irvine, CA, USA)
Microcontroller	nRF52832 64 MHz (Arm Cortex M4) (Nordic Semiconductor ASA, Trondheim, Norway)
Sensor	Microphone IM69D130 (Infineon Technologies AG, Neubiberg, Germany)
Power	3.7 V Li-po battery
Memory	512 KB Flash, 64 KB SRAM 16 MB SPI Flash, 48 KB SRAM
Connectivity	Bluetooth® Low Energy (ANNA-B112)

The device is powered by a 3.7 V Li-po battery, as illustrated in Figure 1.

The system works by detecting the sound of snoring using a wearable sensor. If snoring is detected by the TinyML model running on the wearable device, it sends a signal to the nearby gateway device which in turn generates an alarm. The gateway also stores the snore detection data with timestamps locally and further sends it to a cloud server for permanent storage and analysis. In this way, the system not only detects and alerts users to snoring but also keeps track of historical data which may be shared with medical practitioners for further analysis and diagnosis.

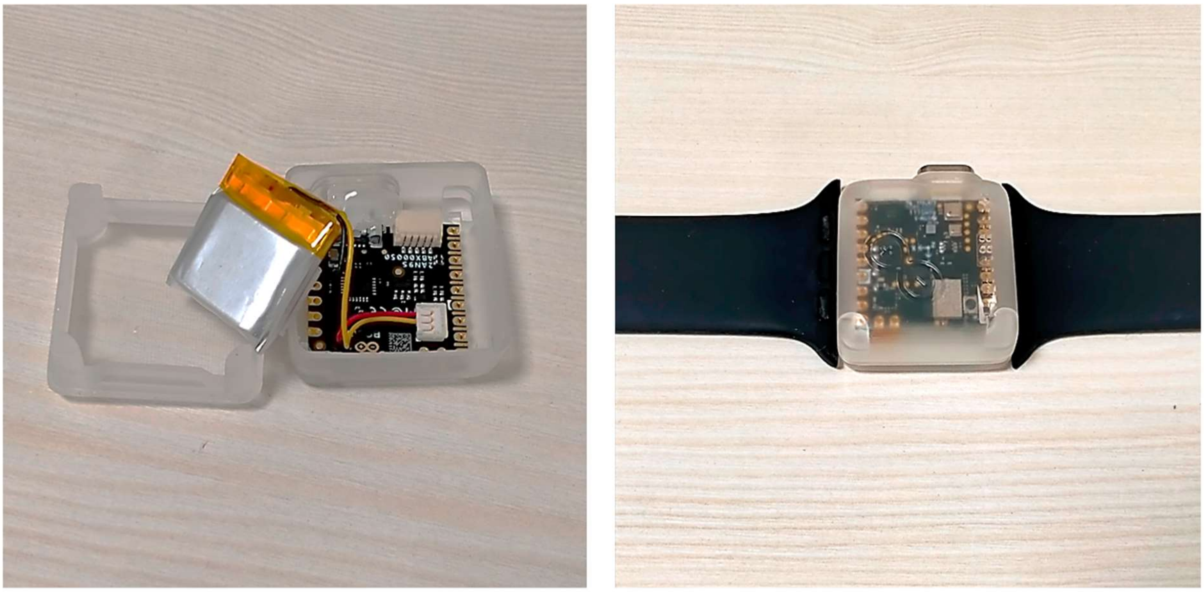

Figure 1. Wearable sensor design.

As shown in Figure 1, the device can be comfortably worn on the wrist by the user during sleep and is used to detect snoring and send alerts to the user. The system architecture is shown in Figure 2.

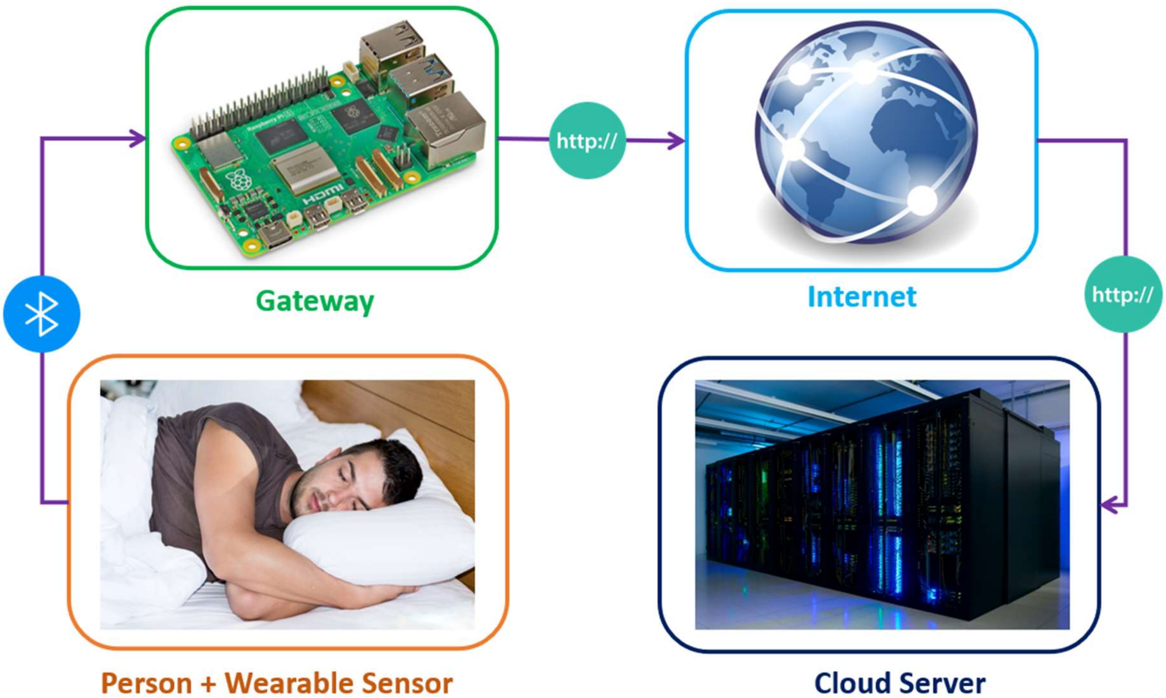

Figure 2. System architecture.

3.2. Dataset

The dataset consists of two distinct classes, snoring and non-snoring. The snoring class has 1000 samples of snoring voices, each lasting for 1 s. This collection includes the snoring voices of adult men and women and of children. The dataset contains snoring and non-snoring voices and noises with and without background.

The non-snoring class consists of 500 samples of non-snoring voices and noises, each also lasting for 1 s. These samples contain varied background sounds. The dataset consists of 50 samples from each category of non-snoring voices and sounds. The dataset used for non-snoring sounds consists of audio recordings from various categories, including doors opening and closing, silence with motor vibrations, clocks ticking, toilets flushing, vehicle sirens, rain and thunderstorms, streetcar sounds, human speech, and television news. These datasets were obtained from Kaggle [14]. Figure 3 provides a visual representation of the dataset used.

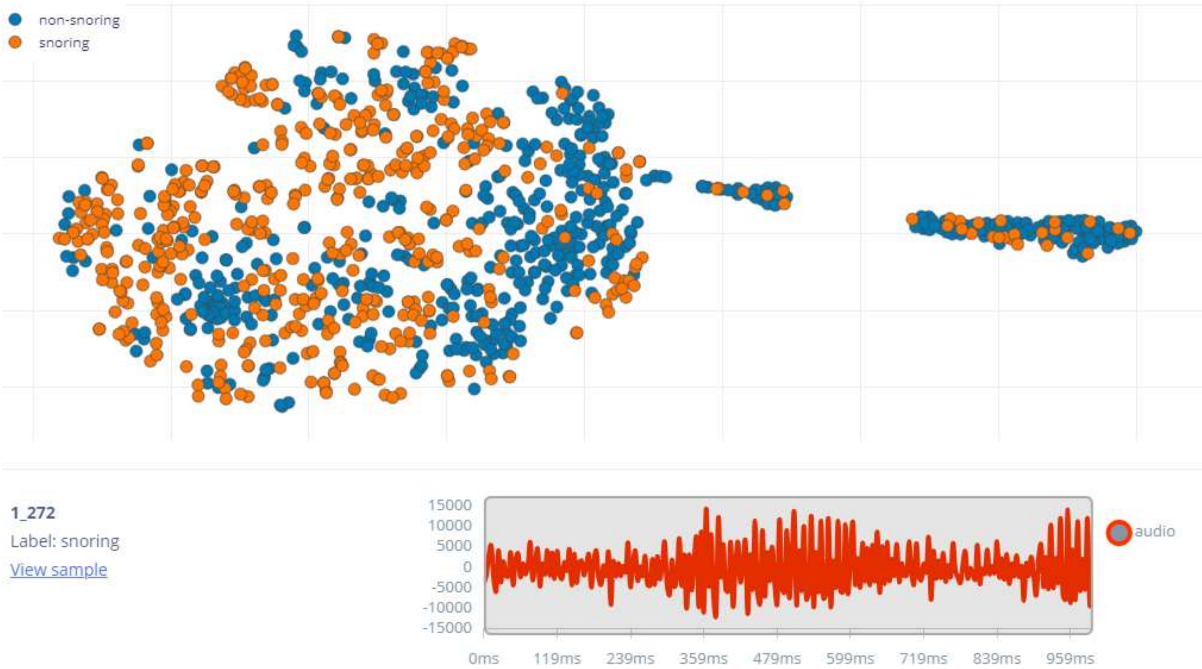

Figure 3. Dataset for snoring and non-snoring sounds.

3.3. Processing

In this study, Audio Syntiant [15] processing is utilized to classify snoring from non-snoring voices and noises. Audio Syntiant is used to extract time and frequency information from signals. The Syntiant audio processing pipeline is a specialized version of Audio MFE, with additional steps for the Syntiant chip's unique characteristics. Key parameters include frame length, frame stride, filter number, FFT length, low frequency, and high frequency, which define the spectrogram features extracted using Mel-filterbank energy. Pre-emphasis is applied with a specified coefficient. The chip-specific feature extractor is chosen based on the particular Syntiant chip used, ensuring optimal feature generation for the given hardware.

Syntiant's feature extraction process begins with a pre-emphasis step to amplify high-frequency components. The audio signal is divided into overlapping segments, with the frame length and stride determining the size and spacing of these segments. These parameters influence the extracted speech features. Table 2 provides the specific values used for Audio Syntiant.

Table 2. Audio Syntiant Parameters.

Log Mel filterbank energy features	
Frame length:	0.032
Frame stride:	0.024
Filter number:	40
FFT length:	512
Low frequency:	0
High frequency:	0
Preemphasis	
Coefficient:	0.96875
Chip	
Feature extractor:	log-bin (NDP1 20/200)

Figure 4 shows the DSP results as a Syntiant spectrogram of snoring sounds, and Figure 5 visualizes the features generated for snoring and non-snoring voices and noises.

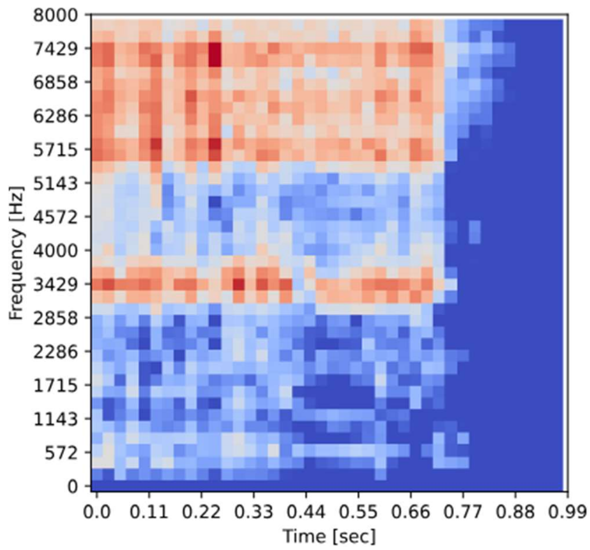

Figure 4. Syntiant spectrogram of snoring sounds.

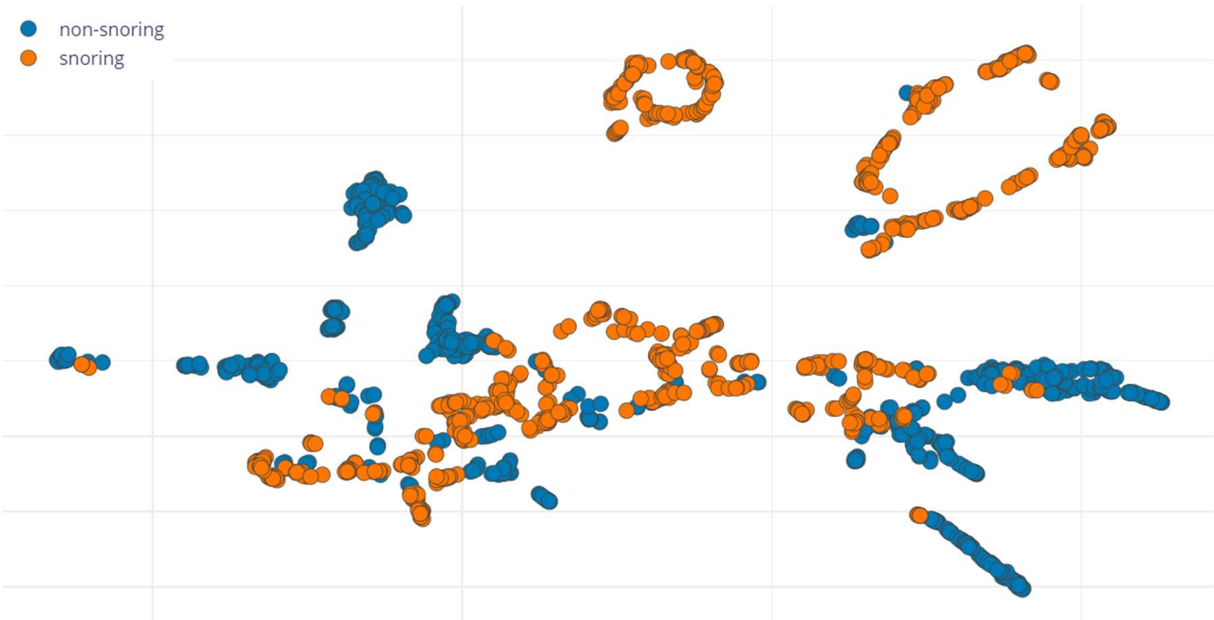

Figure 5. Generated features for non-snoring and snoring sounds.

3.4. Model Architecture

The neural network architecture, as shown in Figure 6, is built using a sequential model. The input to the network consists of a 2D array of 40 × 40 × 1 representing audio features. The first layer is 2D convolutional layer which has 8 filters, 3 × 3 in size. The filters are constrained using max-norm regularization with a maximum norm of 1. The ReLU activation function is used in this layer. After this, a max pooling layer with a pool size of 2 × 2 and a stride 2 is applied. Another 2D convolutional layer having 8 filters, 3 × 3 in size, is applied. A 'valid' padding, max-norm constraint, and ReLU activation are added to capture more complex features in this layer. Another similar max pooling layer follows the second convolutional layer. A dropout layer with a rate of 0.25 is included after the pooling layers to prevent overfitting. An average pooling layer with a dynamically determined pool size is used to aggregate features spatially before flattening. A reshape layer converts the 2D pooled feature maps into a 1D vector, preparing them for the fully connected layers. A fully connected layer having 16 neurons and ReLU activation is used, with L1 regularization to encourage sparsity in the learned weights. A second fully connected layer with 8 neurons and ReLU activation is added, also with L1 regularization and a dropout layer. The final fully connected layer has 2 neurons, representing the number of classes, with a softmax activation function to provide class probabilities.

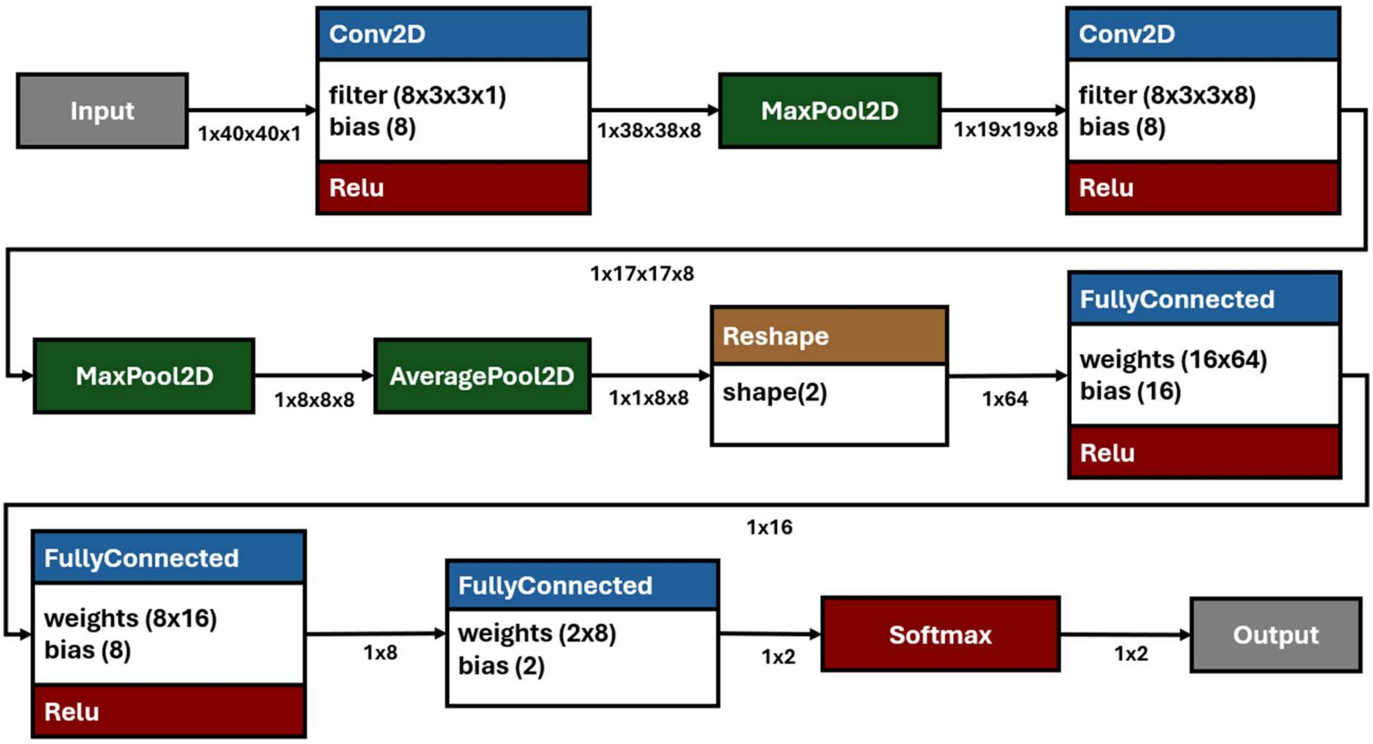

Figure 6. Model architecture.

This architecture is designed to meet the specific requirements of the snore detection problem. For training the model, 100 epochs, 0.0005 learning rate, and a batch size of 32 was used. After training, the model achieved a total accuracy of 96.6% with 0.11 loss. Finally, the TinyML model is quantized from float32 to int8 to fit in the requirements of a low-resource device, making it work efficiently in wearable devices. Table 3 shows the confusion matrix for the training dataset.

Table 3. Confusion matrix (training dataset).

	Non-Snoring	Snoring
Non-Snoring	98.8%	1.2%
Snoring	5.7%	94.3%
F1 Scores	0.97	0.96

The graph in Figure 7 shows that the model has accurately identified snoring and non-snoring voices and noises during training. The model architecture is shown in Figure 6.

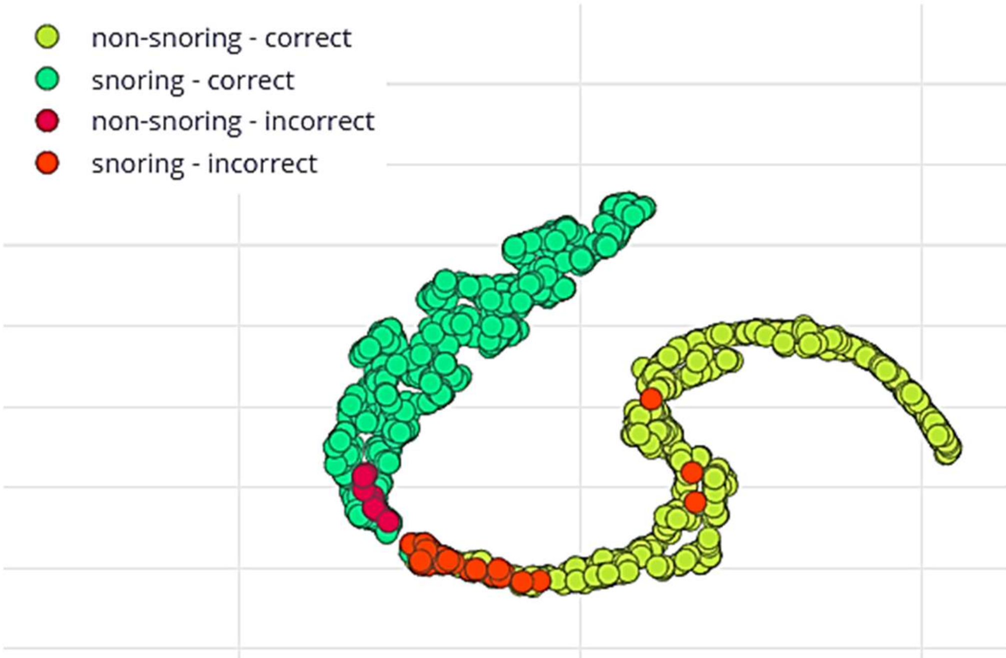

Figure 7. Model accuracy during training for snoring (94.3%) and non-snoring (98.8%).

4. Results and Discussion

The trained model is evaluated on a test dataset. The accuracy of the model, as well as the memory it takes and its processing speed, is also tested on resource-constrained IoT devices. The confusion matrix for the test dataset is shown in Table 4. For non-snoring predictions, the model correctly classified 96.8% of non-snoring instances as non-snoring and incorrectly classified 1.1% of non-snoring instances as snoring. It also classified 2.1% of non-snoring instances as uncertain, indicating that the model could not confidently classify the instance as either snoring or non-snoring. Similarly, the model correctly classified 96.4% of snoring instances as snoring and incorrectly classified 0.5% of snoring instances as non-snoring. It also classified 3.1% of snoring instances as uncertain. F1 scores are measures of the accuracy of the model, and the F1 scores for 'non-Snoring' and 'snoring' voices were 0.98, which indicates that the model achieved high accuracy in classifying both classes in new data. The graph in Figure 8 shows the model performance on the test dataset.

Table 4. Confusion matrix (test dataset).

	Non-Snoring	Snoring	Uncertain
Non-Snoring	96.8%	1.1%	2.1%
Snoring	0.5%	96.4%	3.1%
F1 Scores	0.98	0.98	

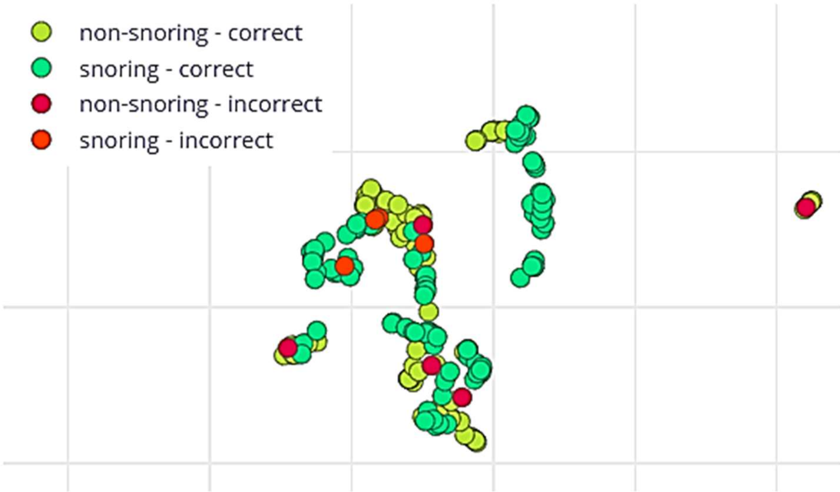

Figure 8. Model performance on the test dataset.

The main objective of this research is to find a suitable model architecture that can be deployed in the resource-constrained environment of the target wearable device. This requires analyzing the hardware requirements of the target device and accordingly designing and selecting the best TinyML model to achieve maximum performance and accuracy. Analysis of input data, signal processing, and neural network structures was conducted to build efficient model architecture per the computing power and memory requirements of the device. The following architecture and configurations were used to deploy a TinyML model:

- Dataset category: voice events
- Target device: Cortex M4
- Time per inference: 100 ms
- Target RAM: 340 KB
- Target ROM: 1024 KB

After a thorough investigation, the best model for the device was selected, and the model was quantized to be deployed in the target device. The following Table 5 provides a comparison of the converted TensorFlow Lite model (float32) and the quantized model (int8) in terms of latency, memory, and accuracy.

Table 5. Model comparison.

	Unoptimized Model (FLOAT32)		**Quantized Model (INT8)**	
	Classifier	**Total**	**Classifier**	**Total**
Latency	955 ms	955 ms	48 ms	48 ms
RAM	58.6 K	58.6 K	17.0 K	17.0 K
Flash	32.8 K		34.7 K	
Accuracy	**96.63%**		**95.85%**	

5. Conclusions

This study proposes a health monitoring system specifically developed for snoring detection using Tiny Machine Learning (TinyML) models. The primary objective of the research is to develop a TinyML-based snoring detection model to accurately identify specific audio patterns associated with snoring during sleep. The research is also aimed at designing TinyML models for resource-constrained Internet of Things (IoT) devices. To test the efficacy of the proposed model, the experiments were conducted in real-world sleep environments by deploying the TinyML model to resource-constrained wearable IoT devices. The results show that the model achieved high accuracy while using minimal computational resources on the target wearable device. The quantized model achieved an accuracy of 95.85% with a low latency of 48 milliseconds, RAM 17.0 K, and Flash

34.7 K, making the model an ideal choice for wearable devices. This research demonstrates the feasibility and practicality of implementing snoring detection TinyML models on resource-constrained IoT devices and provides a non-intrusive method of sleep monitoring. The proposed research makes an important contribution to the advancement of TinyML applications in health monitoring.

Author Contributions: T.M. designed the methodology and built the TinyML model. S.T. conceptualized the idea for this manuscript, supervised and administered the work, prepared the original draft, and validated the data and results. P.M. designed hardware, curated data, and validated the data and results. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the first author.

Acknowledgments: The TinyML model and the graphs for this study were generated using the Edgeimpulse [16] platform.

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

References

1. Mineo, L. Good Genes Are Nice, But Joy Is Better. The Harvard Gazette. Available online: https://news.harvard.edu/gazette/story/2017/04/over-nearly-80-years-harvard-study-has-been-showing-how-to-live-a-healthy-and-happy-life/ (accessed on 23 February 2024).
2. Snoring—Overview and Facts. Available online: http://sleepeducation.org/essentials-in-sleep/snoring/overview-and-facts (accessed on 23 February 2024).
3. Khan, T. A deep learning model for snoring detection and vibration notification using a smart wearable gadget. *Electronics* **2019**, *8*, 987. [CrossRef]
4. Shin, H.; Cho, J. Unconstrained snoring detection using a smartphone during ordinary sleep. *Biomed. Eng. Online* **2014**, *13*, 116. [CrossRef] [PubMed]
5. Xie, J.; Aubert, X.; Long, X.; van Dijk, J.; Arsenali, B.; Fonseca, P.; Overeem, S. Audio-based snore detection using deep neural networks. *Comput. Methods Programs Biomed.* **2021**, *200*, 105917. [CrossRef]
6. Li, R.; Li, W.; Yue, K.; Zhang, R.; Li, Y. Automatic snoring detection using a hybrid 1D–2D convolutional neural network. *Sci. Rep.* **2023**, *13*, 14009. [CrossRef] [PubMed]
7. Shen, F.; Cheng, S.; Li, Z.; Yue, K.; Li, W.; Dai, L. Detection of snore from OSAHS patients based on deep learning. *J. Healthc. Eng.* **2020**, 8864863. [CrossRef] [PubMed]
8. Mitilineos, S.A.; Tatlas, N.A.; Korompili, G.; Kokkalas, L.; Potirakis, S.M. A real-time snore detector using neural networks and selected sound features. *Eng. Proc.* **2021**, *11*, 8. [CrossRef]
9. Ansari, M.W.; Rajak, A.; Basak, R. A Deep Learning Model to Snore Detection Using Smart Phone. In Proceedings of the 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 6–8 July 2021; pp. 1–5.
10. Dong, H.; Wu, H.; Yang, G.; Zhang, J.; Wan, K. A multi-branch convolutional neural network for snoring detection based on audio. *Comput. Methods Biomech. Biomed. Eng.* **2024**, 1–12. [CrossRef] [PubMed]
11. Yang, C.H.; Kuo, Y.M.; Chen, I.C.; Lin, F.M.; Chung, P.C. A Machine-Learning-Based Detection Method for Snoring and Coughing. *J. Internet Technol.* **2022**, *23*, 1233–1244. [CrossRef]
12. Luo, H.; Li, H.; Lu, Y.; Lin, X.; Zhou, L.; Wang, M. Design of embedded real-time system for snoring and OSA detection based on machine learning. *Measurement* **2023**, *214*, 112802. [CrossRef]
13. Nicla Voice. Available online: https://store-usa.arduino.cc/products/nicla-voice (accessed on 3 March 2024).
14. Snoring Dataset. Available online: https://www.kaggle.com/datasets/tareqkhanemu/snoring (accessed on 5 May 2024).
15. Audio Syntiant. Available online: https://docs.edgeimpulse.com/docs/edge-impulse-studio/processing-blocks/audio-syntiant (accessed on 10 May 2024).
16. EdgeImpulse. Available online: https://edgeimpulse.com/ (accessed on 11 May 2024).