

Original Research

Received 06 July 2026

Revised 07 Aug 2026

Accepted 20 Aug 2026

Natural Resources for Human Health 2026, Vol. 6 (17s): 146–164

KEYWORDS:

EMI, FPGA, HDL code generation, audio signals, machine learning, SVM, signal processing

Design and prototype development of real time AI enabled audio controlled assistive wheelchair mechanism

Dasari Lakshmi Prasanna¹, Suman Lata Tripathi^{2*}, Deepika Ghai¹, Inung Wijayanto³

¹VLSI Design Lab, Lovely Professional University, Punjab, India

²Department of Electronics and Communication Engineering, Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune, India

³School of Electrical Engineering, Telkom University, Bandung, Indonesia

Email: dlakshmi525@gmail.com, tri.suman78@gmail.com, deepika.21507@lpu.co.in, iwijayanto@telkomuniversity.ac.id

Corresponding Author: Suman Lata Tripathi

ABSTRACT: Most assistive technologies use audio as the input for controlling electronic devices. Speech is one of the easiest forms of data to control for both disabled and non-disabled people to access systems. Audio data consists of noise, which can be challenging to classify or extract information from. In this paper, an assistive system is designed for audio classification and implemented on a Field-Programmable Gate Array (FPGA) board. Electromagnetic attenuators are designed to filter out noise signals from the useful information or signal. The Electromagnetic Interference (EMI) filtered signals are further processed to classify commands, assisting devices in calculating the performance of all multiclass SVM classifiers. The classifiers are designed and optimized in MATLAB R2021a, and the work is extended to implement on Vivado 2020.1. The Hardware Description Language (HDL) code, generated from MATLAB, is used for model development on the Nexys 4 Artix 7 board. The built-in microphone on the mentioned FPGA board is used to collect audio signals as input data for the learning modules. The accuracy of the machine learning module depends on the training dataset. The larger training data set will ensure the accuracy of the proposed module. The accuracy of the proposed audio signal-based assistive device is 87% for eight different commands to control the wheelchair.

1. INTRODUCTION

The World Health Organization (WHO) reports that about 2.5 billion individuals globally require one or more assistive devices (AD) to perform their daily activities or mobility. With an aging population and rising disease rates, it is estimated that by 2050, about 3.5 billion people will need ADs [1]. Many individuals with impairments have difficulty performing simple tasks independently, particularly when their mobility or limb function is impaired. In such situations, individuals depend on others for small tasks. As technology is developing, it has made it easy for researchers working on assistive technology (AT) for individuals to do daily activities. Different ATs are available for individuals with disabilities, for example, mobility devices such as wheelchairs and exoskeletons, communication aids such as speech-generating devices and fluency assistance devices, artificial larynx, sensory devices for blind people such as screen readers, hearing aids, and Braille displays, cognitive support systems like reminder applications and memory aids [2]. Among all other ATs, speech-based ATs are intact and best for disabled people. Speech is the most natural and accessible mode of communication, which has the potential to greatly enhance the inclusion and quality of life for people with special needs, such as those with cerebral palsy or elderly people living alone [3]. In many cases of severe motor impairment (spinal cord injury and limb paralysis), vocal ability is preserved because vocal cords and speech

muscles are controlled by cranial nerves. Designing AT using audio helps motor impairments, makes it easy to access, and helps in their daily life.

An audio module can include a few technologies, like automatic speech recognition (ASR), instrument retrieval, auditory scene analysis, and glitch detection. Classifying and searching audio according to its characteristics would be useful for many audio and multimedia applications [4-5]. Audio documents are compiled based on similar content by using Machine Learning Classifier models (MLC). MLCs are commonly used for distinguishing speech from other sounds in mixed audio, segmenting clips by class, or identifying similar audio content [6]. Electronic device applications rely on ASR for things like converting voice into patterns, identifying abnormal spikes in sounds, identifying desired sounds from sound signals, and listing audio recordings. The presence of learning classifiers can be seen in several applications, such as medical, finance, academic, military, medical field diagnosis, automatic detection, rehabilitation, or assistive devices [7]. A variety of classifiers are designed differently based on how the input data is extracted. Multiclass, binary classifications, and multilabel classifications are different kinds of classification. In multi-label classification, the samples are mapped to the target labels if there are two possible outcomes. Binomial classification involves two possible outcomes, and multiclass classification involves two classes or more classes [8].

Speech-based control is regarded as the best AT, as speech often remains intact despite functionality of other body parts. Voice commands provide a natural, hands-free, and unobtrusive method of interaction. Numerous studies have demonstrated the feasibility of speech-controlled wheelchairs and assistive technology. In paper [9], the author developed an economical Raspberry Pi prototype that employed on-board speech recognition using Python and open-source toolkits to facilitate basic directional control. Which is affordable and easily assembled, but it reveals that general-purpose single-board computers encounter issues with responsiveness and real-time reliability. The author [10] developed an intelligent voice-controlled wheelchair utilizing a microcontroller (MCU) Arduino, which integrates a voice recognition module utilizing cloud-based ASR such as Google Speech-to-Text, and used ultrasonic sensors for obstacle detection to enhance safety. Their design emphasizes cost-effectiveness and sensor fusion.

Audio or speech can be classified using different approaches of supervised machine learning classifiers that are SVM, k-Nearest Neighbors (KNN), Naïve Bayes (NB), Random Forests (RF) and Decision Trees (DT), Neural Networks (Artificial, convolution, and Recurrent NN), Gradient Boosting, Hidden Markov Models (HMM) and more [11]. In real-world applications, multiple classes must be considered and SVM frameworks need to be tailored accordingly. A trained model is developed by processing data provided in a comma-separated values (CSV) format [12].

A lightweight ASR system is proposed for individuals with visual impairments, especially for educational accessibility. It can be adapted to diverse platforms. Depending on the user, the accuracy of the system varied from 68% to 100%. The system was sensitive to word pronunciation, clarity, and tone. And it was not recommended for real-world noisy environments [13]. In paper [14], the author developed a voice-operated smart robotic wheelchair (SRW). It is employed by Network-in-Network (NIN) convolutional neural networks and Long Short-Term Memory (LSTM) models for speech recognition. The system is trained with five English words pronounced by 10 native Arabic speakers. It is low-cost, effective, and reliable and works in both indoor and outdoor environments, and includes an emergency button. Raspberry Pi and DC motors are used for implementation. In paper [15], Alexa telehealth was developed for people who have cognitive issues. Voice assistive technology (VAT) is trained on specifically Amazon Alexa, to support adults with cognitive-communication disorders (CCDs) stemming from traumatic brain injury, aphasia, stroke, and other conditions in managing activities of daily living (ADL), including scheduling, self-care, meal preparation, and communication. The innovation lies not in the Alexa device itself, but in a telemedicine training program designed to instruct individuals with CCDs and their caretakers on the effective use of VAT to achieve independence. In paper [16], the author compared the speech enhancement and voice based assistive device, and ASR with text-to-speech (TTS) frameworks for dysarthric speech. This demonstrated that various voice-driven assistive techniques are more effective for distinct forms of impaired speech, highlighting the necessity for these strategies to function optimally amidst atypical voices and ambient noise.

The tongue is an ideal tool for developing wearable AT systems for people with severe paralysis, it can control various movements and is connected to the brain through a cranial nerve. Also, it is non-repetitive and can be worn all inside the mouth. It is possible to get into the tongue without any interference from the surrounding body. It is also very comfortable and does not cause any problems. A dual-band transmitter operates at 27 and 432 MHz, which allows the user to switch the external interface [17]. The transmitter streams the digitized samples to a custom-designed TDS to transfer the data to other devices like

smartphones, powered wheelchairs, and personal computers. The iTDS system on chip is the power management integrated circuit (PMIC), which gives 1.8 V supply for analog front-end, digital control blocks, Sensors, and transistors. The PMIC charges 50 mAh Li-ion battery with a constant current of up to 4.2 V, and flourishes data and clock to inform its configuration register through a 13.56 MHz inductive link. The iTDS SoC had been implemented in a 0.5 μm cmos process and consumes 3.7 mW power [18].

The iTDS is a specially designed arch-shaped dental retainer positioned in the buccal shelf at the front of the mouth [19]. It utilizes a pattern recognition algorithm to interpret data from magnetic sensors. Commands are transmitted in 10-bit packets to an external receiver (Rx) via a custom-built, high-gain intraoral antenna operating at a frequency of 2.4 MHz. The standalone iTDS was tested by four participants, each performing tasks based on over five random tongue commands. The system successfully completed 99.2% of the tasks, achieving an information transfer rate of 150.1 bits per minute. Additionally, a novel typing method tailored to the iTDS enabled a typing speed of 3.76 words per minute with an error rate of 2.23%. Machine learning classifier's performance is compared, analyzed and categorized for image and text data [20]. This review paper concludes that the more datasets will give better performance. In paper [21], it is proposed that using magnetic sensors is non-hygienic. Instead, it suggests using an FPGA's onboard microphone for detecting audio signals. By considering on board microphone there is no chance of being unhygienic and uncomfortable, the data collected very easily. And it proposes MATLAB HDL coder that are used to convert the code to HDL language.

In this paper, an SVM classifier designed for a larger training dataset, which is 1400 audio clips, is used. Eight words or classes are used to classify the audio signal. On-board microphones are used to collect the command for real-time audio processing. MATLAB HDL coder generates the HDL code for the learning modules. The performance metrics of SVM are calculated and implemented on the FPGA Artix 7 board.

2. METHODOLOGY

The Artificial Intelligence (AI) enabled assistive wheelchair is designed to assist individuals with severe motor disabilities by enabling hands-free control using speech commands. The methodology involves multiple stages, from signal acquisition to wheelchair movement execution. For implementing TDS on FPGA involves multiple blocks. Which is mentioned in Figure 1, signal acquisition, EMI filters, feature extraction, MLC module and Pulse Width Modulation (PWM) control signal. Designed in MATLAB, converted to HDL codes separately using MATLAB HDL coder and implemented in Vivado on Nexys 4 Artix 7 FPGA board.

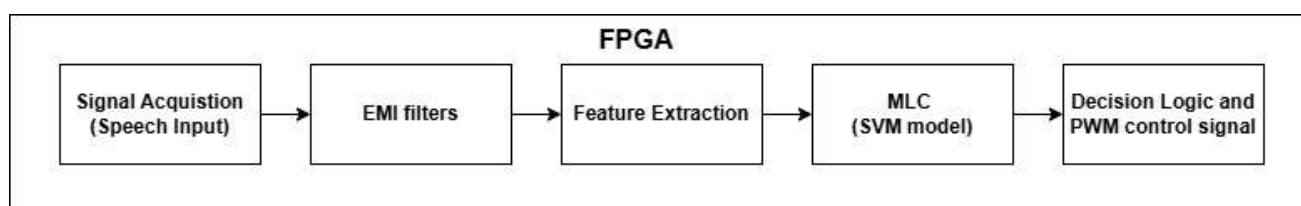


Figure 1 Block Diagram of TDS on FPGA

2.1 Signal Acquisition

In signal acquisition, audio signal is converted in electrical signal by microphone and its level is upscaled by amplifiers. The corresponding electrical signal is sampled and quantized to be converted into digital signal for further processing. The TDS wheelchair utilizes the onboard MEMS (Micro-Electro-Mechanical System) microphone on the Nexys 4 Artix-7 FPGA board to capture speech commands. The microphone is designed to pick up voice signals effectively, ensuring clear command recognition. Artix 7 has 61dBa signal to noise ratio specifically for human ear and -26dBFS sensitivity (full-scale) for pulse density modulation (PDM) of omni-directional microphone. The characteristic of microphone is omnidirectional ADMP421 MEMS microphone, which captures speech commands spoken by the user and optimized for speech frequency range (~300 Hz to 3.4 kHz). The microphone records speech at a sampling rate of 48 kHz, ensuring high-fidelity signal acquisition. The signal is encoded using 16-bit Pulse Code Modulation (PCM) for accurate representation. The system processes single-channel (mono) audio, reducing computational complexity and processing overhead [22]. Figure 2 shows the block diagram of ADMP421 MEMS omni directional microphone.

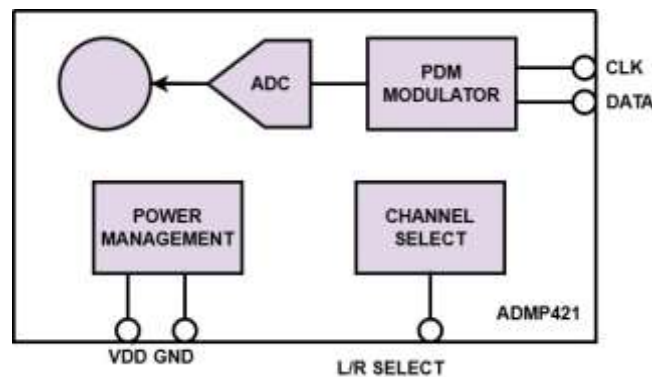


Figure 2 ADMP421 MEMS Microphone Omni Directional Microphone with Bottom Port and Digital Output [22]

2.2 EMI filters

Electromagnetic Interference (EMI) filters are critical for audio classifiers in machine learning, especially in environments where electronic noise can affect signal quality and lead to misclassifications. EMI filtering reduces the burden on the classifier to differentiate between noise and signal, helping the model generalize better to real-world conditions. Cleaner signals can reduce false positives and false negatives, leading to a more robust classifier that performs consistently across varied environments. Electromagnetic Interference (EMI) filters in a Tongue Drive System (TDS) are essential for ensuring signal integrity by minimizing noise and interference from external electromagnetic sources. EMI filter suppresses high frequencies and passes low frequencies and protect circuit from external noise.

An EMI filter performs operation similar to low-pass filter (LPF) where LPF transfer function follows:

$$|H(f)| \approx \begin{cases} 1, & f \ll f_c \\ 0, & f \gg f_c \end{cases} \quad (1)$$

Where, cutoff frequency, f_c depends on R & C values of the circuit.

Since the TDS relies on speech processing using an onboard microphone on the Nexys 4 Artix-7 FPGA board, unwanted noise from power supplies, electronic devices, or radio frequency interference (RFI) can degrade system performance. EMI filters, typically consisting of passive components like capacitors, inductors, and resistors, help attenuate high-frequency noise while allowing the desired speech signals to pass through. Commonly used configurations include low-pass filters, which suppress high-frequency EMI, and shielding techniques to prevent external noise from affecting signal acquisition. Proper EMI filtering improves the reliability and accuracy of the speech recognition system, ensuring robust operation of the TDS for controlling the wheelchair. A compressive sensing (CS) technique was used to compress EEG signals at a reduced sampling rates than the Nyquist rate for ECG signals [23].

2.3 Feature Extraction

Feature extraction in speech processing involves transforming raw audio signals into a set of numerical representations that highlight important spectral, temporal, and statistical characteristics of the speech input. Since audio signals are time-series data with varying amplitude and frequency components, extracting robust features ensures improved classification performance. The extracted features should be noise-resistant, computationally efficient, and discriminative enough to separate different commands reliably.

i) Time-Domain Features

Energy-Measures the intensity of the speech signal, helping to differentiate between voiced and unvoiced sounds. Zero-Crossing Rate (ZCR)-Counts the number of times the signal crosses zero amplitude, useful for detecting different phonemes in speech [24]. Short-Time Energy (STE)-Computes energy over small frames, aiding in segmenting speech from noise [25].

ii) Frequency-Domain Features

Mel-Frequency Cepstral Coefficients (MFCCs)- Extracts spectral features by mimicking the human auditory system, commonly used in speech recognition [26]. Spectrogram Based Features- Provides a time-frequency representation of the signal using Short-Time Fourier Transform (STFT), capturing variations in speech [27]. Linear Predictive Coding (LPC)- Models the vocal tract by estimating filter coefficients, enhancing speech intelligibility for classification [28-29]. Keras framework in CNN is applied using a hyperparameter tuning technique to achieve high accuracy (99%) in classification of patients [30-31].

iii) Wavelet Transform Features

Discrete Wavelet Transform (DWT)- Decomposes speech signals into different frequency bands, aiding in capturing transient components and reducing noise sensitivity [32].

2.4 Machine Learning Classifier

Machine learning classifiers play important role in AT, especially controlled by commands. It classifies the command and helps the device to know which command it is and activate the device according to the command. Therefore, choosing the good classifier is crucial in developing the assistive device. MLC is an algorithm that learns from labeled input data to assign an input feature vector to one of several predefined classes or categories. In paper [37], designed learning module for speech detection. The learning classifiers SVM, NB, RF and Stochastic Gradient Descent (SGD) are designed using MATLAB. By looking at the performance metrics of learning classifiers, we can say that SVM may give better performance compared to other MLCs. Binary classification tasks are generally carried out using Support Vector Machines (SVMs). After careful consideration and comparing different binary classifier model's performance metrics for speech detection models, we decided to use SVM classifier for our research.

The SVM classification approach represents each data sample as a point in an n -dimensional feature space, where classification is performed based on the values of the extracted features. The Support Vector Machine (SVM) determines an optimal decision boundary by identifying the maximal-margin hyperplane, which is defined using the closest positive and negative training samples (support vectors) [6]. Kernel functions $f(x)$ enables the projection of data into a higher-dimensional space that allows a linear separator in space to model complex.

$$f(x) = \text{sign}(\sum \alpha_i \gamma_i K(s_i, x) + b) \quad (2)$$

Samples are classified using above function. Where, x = test vector, s_i = support vector and $K(s_i, x)$ =kernel function, α_i =weight, γ_i =label, and b =bias.

Table 1 shows the comparative analysis for the existing machine learning classifier with their application. Learning classifiers are designed and optimized using MATLAB, HDL code was extracted from MATLAB HDL code for FPGA implementation [33]. HDL code conversion, verification and simulation done in MATLAB. The data collected from the microphone is given to MLC module for extracting the required command/ operation. Eight words are used as commands for TDS, which are back, call, down, help, left, right, straight and up. A total of 1400 audio clips are collected from 175 members. Male and female of different ages are collected. The accent of the people is different because they are from different parts of India. The audio clips are merged to ~0.2 sec and saved in .wav files. Audio files are mono channel with 48000 Hz. Audacity application is used to trim the unnecessary audio signals and maintain audio duration for all audio files. These audio clips are divided into 8 classes, which means each class consists of 175 words. The dataset set is divided into training and testing datasets in a ratio of 80-20%, which means 140 samples in every word consider as training dataset and 35 samples as testing dataset. Figure 3 shows the step followed in the proposed MLC model for wheelchair.

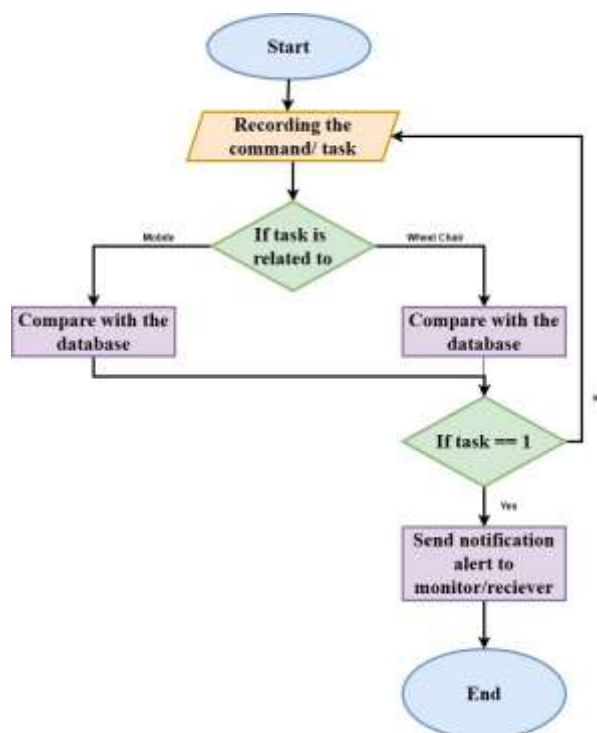


Figure 3 Flow chart of MLC Module

Table 1 Comparative analysis for classifier [14]

Authors	Year	Audio Classifiers	Results
Khalid Zaman et al. [34]	2023	CNN, RNN, Auto encoders, Transformers and Hybrid deep learning	All the models are applied on large dataset of voice records. Hybrid neural network has better accuracy compared to other learning models.
Zvi Kons, Orith Toledo-Ronen [35]	2013	DNN	Applied on small dataset and the audio is classified by deep neural network.
Feng Rong [36]	2016	SVM classifier with Gaussian kernel	Two datasets are considered GS and AS, both database has accuracy of 87.6 and 86.3 respectively.
D. L. Prasanna and S. L. Tripathi [37]	2022	NB, SVM, SGD, and RF	The results of models and comparing the performance metrics SGD has high accuracy of 93%. SVM has high sensitivity and F1 score compared with NB, RF and SGD classifiers.
J. Kim et al [38]	2023	KNN, SVC, Logistic Regression, Random Forest, Decision Tree Classifier and Gaussian NB	Here 20 audio files and 2 classes are considered. The performance accuracy of the models achieved approximately 95%.
A. Kulkarni et al [39]	2020	Author proposed the use of a Long Short-Term Memory (LSTM) network for classifying environment sounds	The researchers evaluate the model using various metrics. The results show an accuracy of 0.86, precision of 0.87, recall of 0.87, support value of 1747, and an f1 score of 0.87 achieved by the proposed model.

2.5 Verilog HDL model

The MATLAB HDL Coder is a powerful tool for designing, simulating, and implementing Hardware Description Languages (HDL) for digital systems. The program allows developers to generate synthesizable VHDL or Verilog code directly from MATLAB algorithms or Simulink models. In FPGA and ASIC design, this capability is particularly useful since high-performance, hardware-optimized implementations are a vital component of the design process.

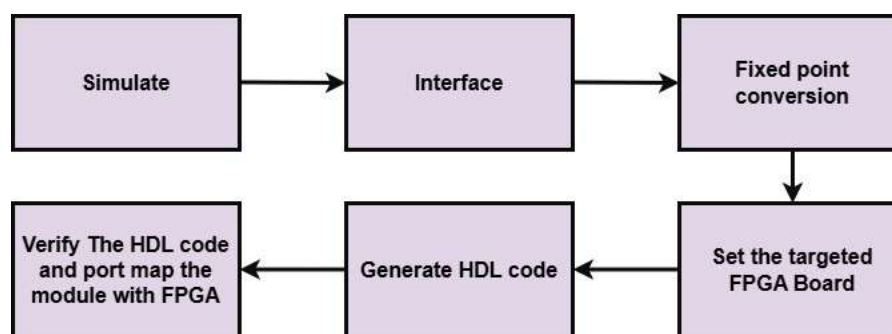


Figure 4 HDL code generation

Figure 4 explains the process of HDL code conversion from MATLAB programming using MATLAB HDL code generation. In this conversion the programming code converts to HDL code automatically, this code generation saves a lot of time. The steps involved here are that the MATLAB code has to be simulated first; MATLAB tool has to be interfaced with synthesis tool (Vivado); In the next step we need to convert the data to fixed point for that datatype has to be defined. Set the targeted FPGA board and their specification and synthesis tool. Set the preferable language for generating HDL code and the final step which involved in code generation is verification [40, 41]. Nexys 4 Artix 7 FPGA board is used for implementing the MLC module for audio classifiers. The HDL code extracted from the MATLAB is implemented using Vivado Xilinx 2020.1. For real-time audio signals on board microphones are used, which use ADMP421 analog devices. Microphones have a signal to noise ratio of 61dB and -26dBFS sensitivity. The board has 100Hz and 15kHz of frequency response range. The audio signal is collected from on board microphone is .wav file. This file is used for classification.

3. IMPLEMENTATION OF TDS SPEECH BASED ASSISTIVE DEVICE ON FPGA

3.1 Dataset Description:

In this research, we collected 1400 audio clips for eight commands- “back”, “call”, “down”, “help”, “left”, “right”, “straight” and “up” from 175 volunteers of different age groups, including both male and female. Volunteers are from different states of India, this introduced natural variation in pronunciation and accent, which strengths the robustness of model during training. To find the variability, audio clips are collected in different environments- quiet places like rooms without background disturbance and moderate noise environment like open area, inside a classroom with bit background noise. This ensured the balancing the real-world conditions. The pronunciation, accent and the data collecting environment are different for every participant. Each volunteer recorded the commands separately using mobile microphone and shared the audio recordings through mobile applications. To ensure that the dataset remained de-identified, audio recordings were stored without personal identifiers.

3.2 Dataset organization and labeling:

The dataset is systematically organized and labeled according to the command words. The audio clips are labelled corresponding to their command word (e.g. back_1, back_2, etc.) and each command word as identified class. All the audio files are renamed and organized according to classes. Classes are well balanced and it made easy to load and test the data consistency easy.

3.3 Data pre-processing:

Before feeding the data to training module the audio files undergone preprocessing for quality processing. To normalize, trim, change the format and reduce the noise and making it ideal for classifier AUDACITY application is used. AUDACITY

application is free, open-source digital audio editing software [42]. All the audio clips are normalized and saved in a proper .wav format. Speed of the audio files are standardized, starting and ending silence was removed and employed mono channel with 48000 Hz and encoded to 16-bit signed PCM. The quality of the dataset is improved by applying these pre-processing techniques.

3.4 Audio Acquisition by accessing the Microphone from FPGA board

Accessing audio input from an FPGA involves using digital microphones like the ADMP421, a high-performance, low-power MEMS microphone ideal for clear and accurate audio capture. The ADMP421 is equipped with a MEMS sensor, a preamplifier, and a Sigma-Delta modulator that produces a Pulse Density Modulated (PDM) output signal. This microphone is compact in size (3 mm × 4 mm × 1 mm) and its surface-mount design makes it suitable for embedded applications. An ADMP421 provides high audio quality and low power consumption while maintaining 61 dBA Signal-to-Noise Ratio (SNR), 26 dBFS Sensitivity, and wide frequency response, making it the ideal solution for real-time systems like FPGA-based tongue drive systems (TDS) [22]. The ADMP421 microphone requires special connections to be attached to an FPGA. The Nexys 4 Artix 7 board has one ADMP421 microphone mounted to an FPGA.

The desired sampling rate of the audio clips is 48000 Hz, mono channel and 16-bit signed PCM. The microphone nexys 4 uses a frequency of a PDM signal in the range of 1MHz to 3MHz. The microphone clock frequency is 3.07 MHz, and the audio acquisition works on this frequency, which is calculated as below:

$$f_{CLK} = f_s \times \text{Over Sampling Ratio} \quad (1)$$

Where $f_s = 47\text{kHz}$ and $\text{Over Sampling Ratio} = 64x$.

The majority of machine learning models and audio processing algorithms, such as the SVM classifier, are designed to operate with PCM data. This data signifies genuine audio amplitudes inside a discrete temporal series. PDM data, conversely, is a high frequency bitstream that represents the density of 1s and 0s rather than direct amplitude values. It is infeasible to extract significant features for subsequent analysis without converting PDM to PCM [43]. In signal processing, cascaded integrator-comb (CIC) filters are specialized digital filters that are frequently employed, particularly when there are notable variations in sample rates, such as when the rate is increased (interpolation) or decreased (decimation). These filters are constructed with integrator stages and comb stages as their constituent parts. Integrators sum up the signal over time, functioning as low-pass filters, whereas comb stages calculate the difference between delayed input samples, acting as high-pass filters [44]. Without the need of multipliers, this configuration enables CIC filters to carry out effective low-pass filtering using just addition and subtraction.

The microphone transmits a continuous flow of 1-bit PDM data with each rising edge of the clock signal (micClk). A greater quantity of 1s indicates an elevated sound level, whereas an increased number of 0s signifies a diminished sound level. The PDM data is thereafter transmitted to the CIC filter for processing and conversion to PCM. The high-frequency PDM signals from the ADMP421 microphone are converted to lower-frequency PCM signals via a Cascaded Integrator-Comb (CIC) filter. It employs only adders, subtractors, and delay components, making it ideal for hardware implementation. The filter comprises two primary stages: integrators and combs. Integrators execute cumulative summing to refine PDM data, functioning as a low-pass filter. Combs compute the disparity between delayed samples, eliminating high-frequency noise and minimizing mistakes. The decimation procedure decreases the sampling rate by a factor corresponding to the OSR. The CIC filter output signifies the genuine audio waveform in PCM format.

$$G = (R \cdot M)^N \quad (2)$$

Where R = Decimation Factor, M = Differential Delay and N = Number of integrator-comb stages

The input is 1 bit 16 PDM from MEMS ADMP421 microphone, the output will be 16-bit PCM audio. CIC filter designed with 3 stages of integer-comb ($N=3$), R which is over sampling ratio, and it is 64 and M represents time interval for comb. For audio applications, lower time interval is the best choice i.e. $M=1$. Which minimizes the hardware complexity and beneficial to use FPGA resources utilization. The module uses a 3-stage CIC filter to change the ADMP421 microphone's 1-bit PDM signal into a 16-bit PCM signal. Integrators are employed to capture and extract low-frequency audio data from the PDM signal, which

is sampled at a high rate of 3.008 MHz. The integrator outputs are then routed through comb filter stages, which eliminate unwanted noise and reduce signal drift. In order to get the optimal sample rate for audio processing, a decimation factor of 64 is utilized, which brings the sample rate down to around 47 kHz.

3.5 EMI filters for removing unwanted electromagnetic noise

The microphone and other electronic components might pick up electromagnetic noise from surrounding devices, particularly on an FPGA with numerous high-speed digital circuits. These undesirable noise signals from the microphone are lessened by the EMI filter [45]. The system makes sure that only the crucial audio frequencies are preserved by removing high-frequency noise. As a result, voice commands are more accurately recognized by the SVM classifier. The SVM model may make errors or classify audio data incorrectly due to noise, but the filter helps by supplying cleaner data. In applications like medical or assistive equipment, including TDS, adherence to Electromagnetic Compatibility (EMC) standards is essential [46-49]. EMI filters facilitate compliance with these rules.

The ADMP421 microphone's voice signal has a frequency range of 100 Hz to 15 kHz. The PDM stream is sampled at a high frequency (3.008 MHz) to encode the audio using a noise-shaping algorithm. In order to obtain useable audio data, we compress the signal to a tolerable sample rate of 47 kHz and eliminate high-frequency noise using a low-pass filter. The EMI filter for the Tongue Drive System (TDS) is implemented after converting the PDM data to 16-bit PCM using a CIC filter. We chose a low-pass filter with a cutoff frequency of around 20 kHz to effectively suppress noise while preserving important voice data. The filter was designed using MATLAB because of its strong simulation and filter design capabilities. We selected a 3rd-order Butterworth filter for its maximally flat frequency response in the passband, ensuring minimal distortion of the voice signal. The resulting in a value of approximately 0.851 for a sampling rate of 47 kHz. Using MATLAB's butter function, the filter coefficients are generated and visualized using the fvtool to confirm the frequency response.

$$f_c = \frac{\text{Cutoff frequency}}{\text{Sampling frequency}/2} \quad (3)$$

3.6 Support Vector Machine Classifier

The learning classifier uses eight commands to activate wheelchair, those commands are back, call, down, help, left, right, straight and up. The audio clips are collected from 175 people, a total of 1400 mono channel, 16-bit signed PCM audio clips are used to train the SVM. A One-vs-All (OvA) SVM classifier is a method employed for multi-class classification. It operates by partitioning the issue into many binary classification tasks. Each classifier is trained to differentiate one class from all the others, employing several Support Vector Machines. The decision function for i^{th} SVM represents

$$f_i(x) = w_i^T x + b_i \quad (4)$$

Where w_i is weight vector for i^{th} SVM, b_i is bias term for the i^{th} SVM and x is input feature vector.

The objective function for each SVM

$$\min_{w_i, b_i} \frac{1}{2} \|w_i\|^2 + C \sum_{j=1}^m \xi_j \quad (5)$$

$$\begin{aligned} y_i(w_i^T x_j + b_i) &\geq 1 - \xi_j & \text{for all } j \\ \xi_j &\geq 0 \end{aligned} \quad (6)$$

Where C = Regularization parameter, ξ_j = slack variable, $y_j \in \{-1, 1\}$ = class label for training sample j , and m = number of training samples.

$$\hat{y} = \arg \max_i f_i(x) \quad (7)$$

For the prediction of an audio command using one vs all above equation (Eq. (7)) is used, which computes all the decision functions for each class. The class with highest score is considered as the predicted class,

3.7 PWM Generation

A verilog module of PWM generation is used for generating the output signal for controlling motor driver according to the commands or audio signals.

DC Motor mapping: The L298N controls the two DC motors of the wheelchair. Each motor moves independently to achieve different motions. Table 2 give details of motor commands and the action.

Table 2 Motor commands and actions

Command	Action	Motor 1 (Left Wheel)	Motor 2 (Right Wheel)	Speed (PWM)
back	Move Backward	Reverse (0,1)	Reverse (0,1)	Medium/High
call	Move Forward & Activate Buzzer	Forward (1,0)	Forward (1,0)	Medium/High
down	Decrease Speed	No change	No change	Reduce PWM
help	Stop & Activate Buzzer	Stop (0,0)	Stop (0,0)	0 (stop)
left	Turn Left	Stop (0,0)	Forward (1,0)	Medium
right	Turn Right	Forward (1,0)	Stop (0,0)	Medium
straight	Move Forward	Forward (1,0)	Forward (1,0)	High
up	Increase Speed	No change	No change	Increase PWM

4. RESULT AND DISCUSSION

SVM multi-class classifier is designed for eight words to control the wheelchair. The performance metrics are calculated for SVM multiclass classifier. Obtained 8x8 confusion matrix and Receiver Operating Characteristic (ROC) of SVM multiclass classifier. For real-time audio signals on board microphone is used and utilization summary is calculated. For SVM implementation MATLAB HDL coder is used to convert the MATLAB code to HDL language- Verilog. Comparative analysis of implementation of different blocks is mentioned.

4.1 SVM multi-class classifier optimization

SVM classifiers classify audio signals according to eight pre-defined categories, recognized as classes. Systems of multi-class classifiers are commonly employed in the recognition of speech commands, the detection of audio events, and the classification of music genres. The SVM classifier is designed for eight classes, 1400 audio clips are used to design and test the SVM multiclass classifier. The classifier has 8 words, which means eight classes. The classifier is trained and tested in 80%-20% split. Means 1120 clips used to train and 280 as test. Each class has 175 audio clips, in that 140 audio clips are used to train and 35 clips are given as testing to calculate the performance metrics. The predicted and true values of classes are mentioned in the matrix. 20 clips are given as testing samples/ dataset to the SVM module. All the audio clips are adjusted and saved in .wav format using Audacity software. Standardizing the audio format ensures consistency across all data, making it easier to process and analyze. With all clips at the same frequency and duration, it reduces variability that could affect model training and classification accuracy. Additionally, using a common format like .wav guarantees compatibility with various audio processing tools and libraries.

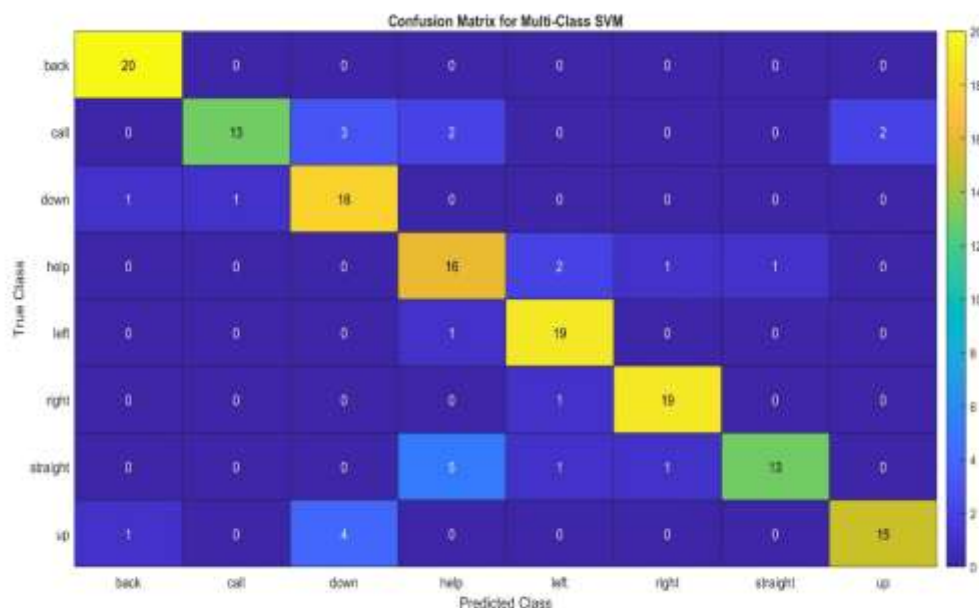


Figure 5 8x8 SVM multi-class classifier Confusion matrix

Above figure 5, explains an 8x8 confusion matrix has rows and columns corresponding to the 8 different classes (words). Each entry in the matrix represents the number of instances where the classifier predicted one class when the true class was another. Specifically: Rows correspond to the actual (true) classes. and Columns correspond to the predicted classes. Each element at position (i, j) in the matrix tells you how many instances of class i (actual class) were predicted as class j (predicted class). For an 8x8 matrix, you have 8 possible classes for both the actual and predicted values, so the matrix will contain 64 entries in total.

Table 3 Performance parameters

Performance Metrics	Formula [36]	Percentage
Overall accuracy	$\frac{\sum_{i=1}^8 TP_i}{\sum_{i=1}^8 (TP_i + FP_i + FN_i + TN_i)}$	85
Precision	$\frac{1}{8} \sum_{i=1}^8 \frac{TP_i}{TP_i + FN_i}$	84.625
Sensitivity/ Recall	$\frac{1}{8} \sum_{i=1}^8 \frac{TP_i}{TP_i + FN_i}$	83.125
Specificity	$\frac{1}{8} \sum_{i=1}^8 \frac{TN_i}{TN_i + FP_i}$	97.625
F1 score	$\frac{1}{8} \sum_{i=1}^8 F1_i$ $[F1 = \frac{2 * Precision * Recall}{Precision + Recall}]$	82.75

In an 8x8 multi-class classifier, performance metrics such as accuracy, precision, recall, and F1-score provide insight into how the classifier distinguishes between the 8 different classes (e.g., words). Accuracy is the overall percentage of correct predictions

across all classes and is calculated by dividing the sum of true positives (diagonal elements) by the total number of predictions. Precision measures how many of the instances predicted as a certain class were actually correct, focusing on minimizing false positives for that class. Recall (or sensitivity) evaluates how well the classifier identifies all true instances of a class, minimizing false negatives. The F1-score is the harmonic mean of precision and recall, providing a balanced measure when precision and recall differ. For each class, these metrics are computed individually (using a One-vs-Rest approach) and can also be averaged across all classes using macro-averaging (treating each class equally) or micro-averaging (weighted by the number of instances per class). These metrics offer a comprehensive evaluation of the classifier's ability to distinguish between all 8 words in the dataset, revealing strengths and weaknesses in both general accuracy and class-specific performance.

```
Optimization completed.
MaxObjectiveEvaluations of 30 reached.
Total function evaluations: 30
Total elapsed time: 66.5431 seconds
Total objective function evaluation time: 52.1873
```

Figure 5 Simulation report

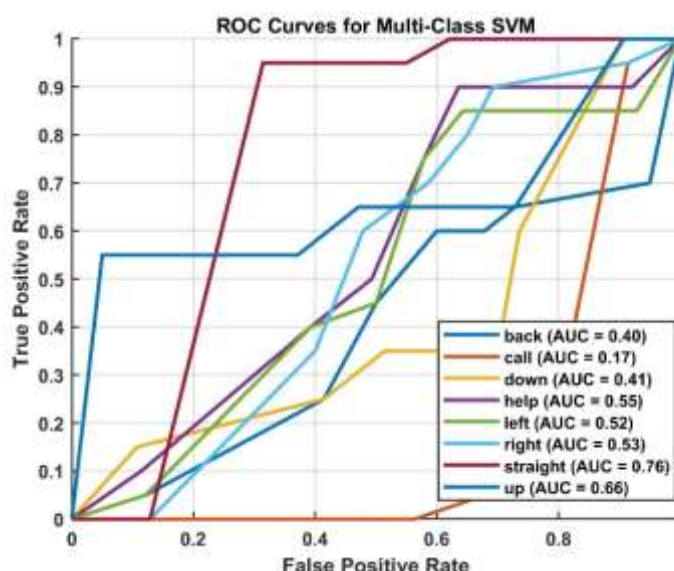


Figure 6 ROC curve of 8x8 SVM audio classifier

ROC curves are commonly used in medical diagnostics to evaluate the performance of tests and determine optimal thresholds for disease detection. In machine learning, they are used to assess the discriminatory power of classification models. Additionally, ROC curves are applied in credit scoring to analyze the ability of models to distinguish between good and bad credit risks. ROC curves are constructed by plotting True Positive Rate (TPR) against False Positive Rate (FPR) for each class i and varying the decision threshold as shown in figure 6. As a result of different thresholds, different pairs of TPR and FPR will result, which is plotted to form the ROC curve. Figure 7 shows the response of minimum objective with respect to function evolution. The proposed audio assistive module is simulated and verified using Artix7 Nexys 4 board. Table 4 shows the FPGA utilization report by indicating the total number of LUTs (look up tables), slice registers, DSP (Digital Signal Processing) blocks and buffers. Table 5 shows the timing report of the FPGA based assistive model, indicating timing constraints values under limit without any violation of timing check.

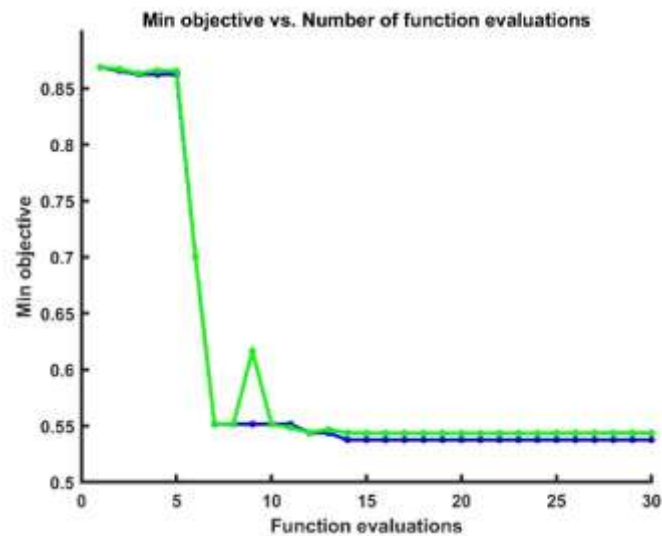


Figure 7 Number of function evaluation of SVM multi-class classifier

Table 4 MEMs microphone ADMP421 of Nexys 4 Artix 7 FPGA board utilization report

Name	Slice LUTS (63400)	Slice Registers (126800)	Slice (15850)	LUT as logic (63400)	LUT as Memory (19000)	Block Ram (135)	DSP (240)	Bonded IOB (210)	Buffers (30)
Ethernet Top	414	716	201	337	77	0.5	2	14	3
Ethernet reset generator	9	26	11	9	0	0	0	0	3
Packet generator	215	306	91	215	0	0.5	0	0	0
Microphone	143	317	74	66	77	0	2	0	0
Reset	3	3	4	3	0	0	0	0	0

Table 5 Design Timing Summary

Timing Constraints	Setup	Hold	Pulse Width
Worst Negative Slack (WNS)	0.796ns	0.087ns	3.000ns
Total Negative Slack (TNS)	0.000ns	0.000ns	0.000ns
Number of Falling Endpoints	0	0	0
Total Number of Endpoints	2043	2043	2043

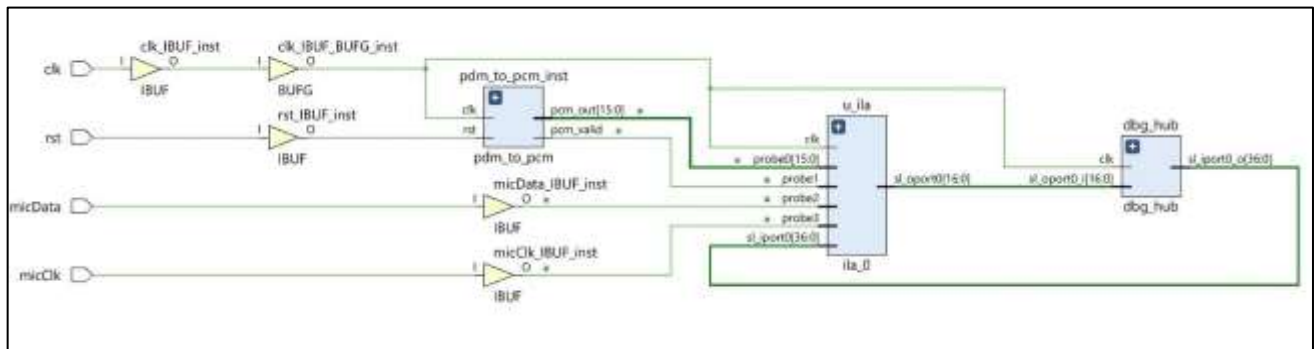


Figure 8 Schematic of audio acquisition



Figure 9 Simulation waveform of audio model of assistive device on FPGA

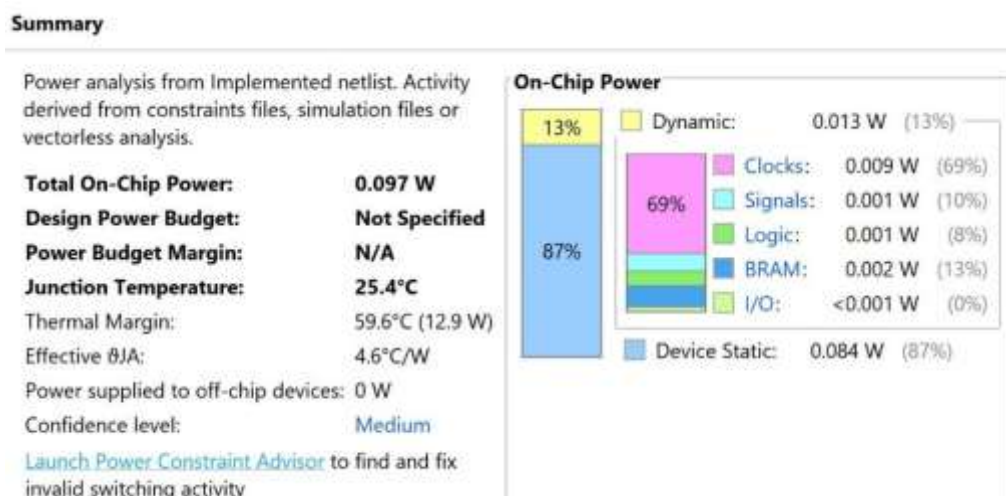


Figure 10 Power summary of using on-board microphone for implementation of real-time TDS

A schematic of audio acquisition model for assistive device is shown in figure 8 that is obtained in Vivado simulation. In this model, the PDM microphone signal is converted in a suitable PCM signals and ILA is used for debug monitoring. Figure 9 shows the simulation waveform that verifies the output based on audio command. A power summary is shown in figure 10 that gives details of dynamic power of 87%, static of 13% and signal power consumptions on Artix 7 Nexys 4 board. Power calculations help in deciding the battery requirements in real time scenario.

4.2 EMI filter Implementation

In paper [49], EMI filters designed, low-pass, high-pass and band pass filters. Table 6 shows the utilization device report of filter model used in the proposed voice controlled assistive device. The proposed filter design utilize 0.15% of LUT, 0.09% of FF, 1.67% of DSP and 15.24% of IO blocks on FPGA board. Figure 11 the design block of EMI filter module developed

FPGA. The power summary as shown in figure 12, demonstrates 50% dynamic and 50% static power consumptions with total power consumption of 0.182 W at junction temperature 25.8°C.

Table 6 Utilization summary of EMI filter FPGA implementation

Resource	Utilisation	Available	Utilisation (%)
LUT	98	63400	0.15
FF	112	126800	0.09
DSP	4	240	1.67
IO	32	210	15.24

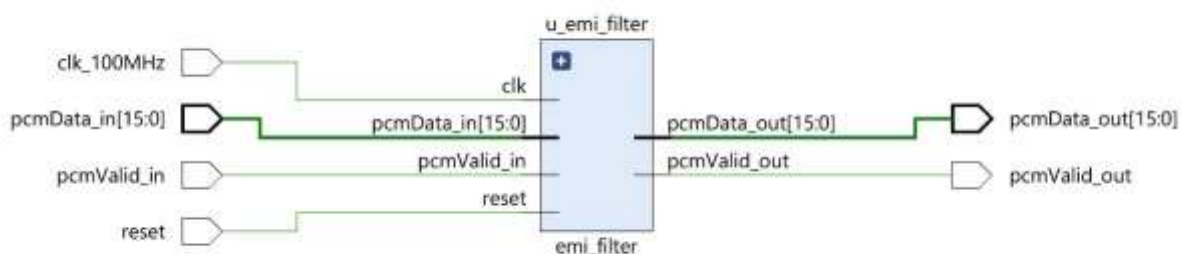


Figure 11 EMI filter design block on Xilinx Vivado

Summary

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power: 0.182 W
Design Power Budget: Not Specified
Power Budget Margin: N/A
Junction Temperature: 25.8°C
 Thermal Margin: 59.2°C (12.8 W)
 Effective θ_{JA} : 4.6°C/W
 Power supplied to off-chip devices: 0 W
 Confidence level: Low
[Launch Power Constraint Advisor](#) to find and fix invalid switching activity



Figure 12 Power analysis of EMI filter on Artix 7

5. COMPARATIVE ANALYSIS

A comparative device utilization report is shown table 7 including the Microphone, EMI filters and MLC modules. The report shows the major contribution by MLC block for number LUTs while Microphone block uses a greater number of FF and registers on FPGA for the proposed audio model. EMI filter requires highest number of block RAM compared to the other two blocks.

Table 7 Performance comparison of device utilization of different blocks on Artix 7 FPGA.

Module	LUTs	Registers	Slices	Block RAM	DSP	FFs	IO
Microphone	414	716	201	0.5	2	716	14
EMI Filter	98	112	41	112	4	112	32
MLC	822	504	718	90	10	504	32

After evaluating performance and power analysis of all the required module, a voice controlled assistive device is proposed to help disabled person. A prototype model is shown in figure 13 including FPGA based wheelchair module controlled through the voice command. The proposed assistive device architecture consists of an FPGA-based processor configured using a bitstream file to execute an audio-controlled signal processing framework. The system integrates an EMI filter, a microphone for audio capture, and a Machine Learning Core (MLC) for intelligent signal classification and processing. The audio command will be processed through FPGA that send signal to move the wheelchair in forward, backward left and right. Here, FPGA-based motor control system for a wheelchair platform, where a Nexys A7 FPGA board generates low-voltage control signals (direction and PWM speed control) that are fed into an L298N dual H-bridge motor driver. The motor driver acts as a power interface, amplifying these logic signals to drive two DC geared motors connected to the wheelchair wheels. A 12 V battery supplies the required power for the motor driver and motors, while a shared ground ensures proper signal reference between the FPGA and driver. Overall, the system demonstrates how audio signal or speech fed through an FPGA can implement digital control logic to drive the motor, enabling controlled movement of the wheelchair.

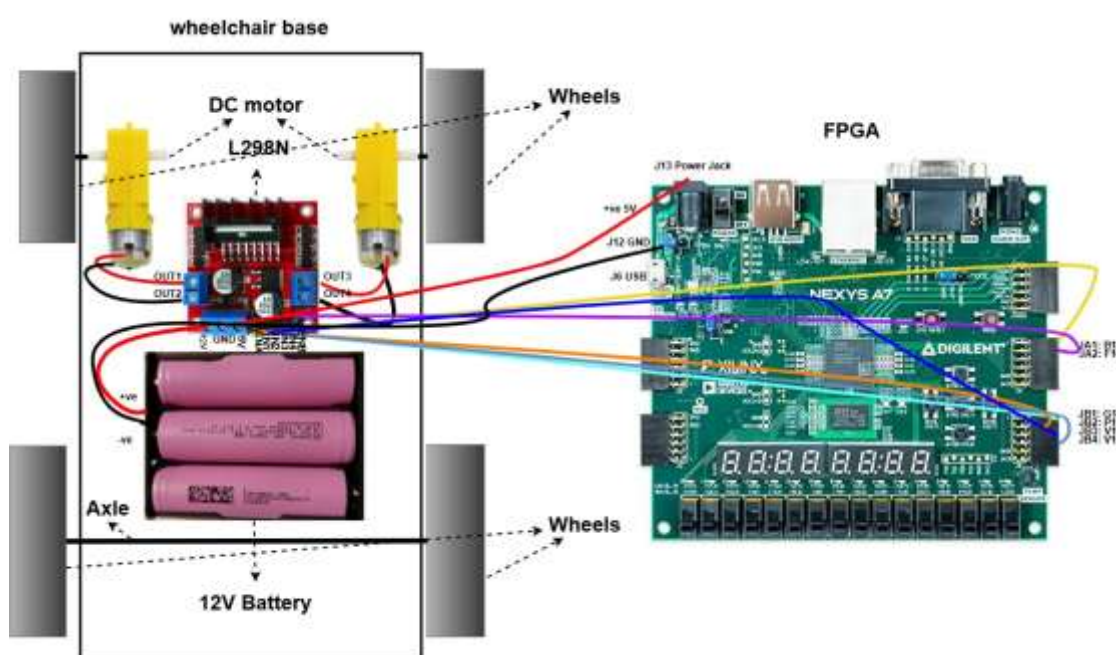


Figure 13 Wire connections of FPGA and L298N motor driver to two DC motors of wheelchair

6. CONCLUSION

The voice or speech controlled assistive device is designed to activate wheelchairs including three major blocks as EMI filter, MLC and input from microphone. Eight words are used as commands to control the wheelchair. The speech data has been collected from 175 members. A total of 1400 audio clips are utilized to train and test the model. Audio signals are classified based on SVM one vs rest multiclass classifier. An 8x8 confusion matrix is calculated for the multiclass classifier for TDS wheelchair. The performance metrics of the multiclass classifier for eight words are calculated. Obtained 87% of overall accuracy

of the multiclass SVM classifier. ROC of the classifier is calculated for multiclass SVM. Power, timing and utilization of the microphone are calculated. SVM is implemented on the FPGA board using MATLAB HDL coder. Overall, the multiclass SVM classifier proved to be an efficient and accurate tool for classifying audio signals, with power, timing and utilization of the microphone considered. After design and simulation of the proposed model, a prototype is modelled to assist the disable person through voice command. For future development the assistive wheelchair can be worked on specker-independence.

AUTHORS' CONTRIBUTIONS

The research work is performed by **Dasari Lakshmi Prasanna**, introduction and methodology are written by **Suman Lata Tripathi**, discussion is written by **Deepika Ghai** and concluding remarks is written by **Inung Wijayanto**.

DECLARATION

FUNDING

No funding.

CONFLICTS OF INTEREST/COMPETING INTERESTS

Authors do not have any conflict of interest.

DATA AVAILABILITY

The audio dataset collected and analyzed during this work is not publicly available because it is part of ongoing PhD research. Subject to institutional consent, the first author may make de-identified data available upon reasonable request.

CODE AVAILABILITY

Custom code is developed for simulation and classification is not publicly available at this time as part of ongoing PhD research. Subject to institutional consent, the first author may provide code upon reasonable request.

ETHICS APPROVAL NA

The research was non-clinical, simulation-based design and prototyping with FPGA. The study did not involve any medical interventions and did not produce environmental hazards. According to the institutional guidelines of Lovely Professional University, formal ethics committee approval was not required for non-clinical study. Informed consent was obtained from all participants prior to recording.

AUTHORS' CONTRIBUTIONS

The research work is performed by **Dasari Lakshmi Prasanna**, introduction and methodology are written by **Suman Lata Tripathi**, discussion is written by **Deepika Ghai** and concluding remarks is written by **Inung Wijayanto**.

REFERENCES

- [1] World Health Organization: WHO, *Assistive technology*, Jan. 02, 2024. <https://www.who.int/news-room/fact-sheets/detail/assistive-technology>
- [2] *Types of AT / Guide to Assistive Technology*. (n.d.). Guide to Assistive Technology. <https://mn.gov/admin/at/getting-started/understanding-at/types/>
- [3] Wold, E., Blum, T., Keislar, D., & Wheaton, J. (1996). *Content-based classification, search, and retrieval of audio*. IEEE multimedia, 3(3), 27-36.
- [4] Rabiner L. and Juang B. H. (2003) *Fundamentals of Speech Recognition*, Person Education, 2003.
- [5] Ghai, D., Saxena, S., Dhingra, G., Tripathi, S. L. (2024) *A comprehensive review on performance-based comparative analysis, categorization, classification and mapping of text extraction system techniques for images*. Multimed Tools Appl. 84, 2327–2484. <https://doi.org/10.1007/s11042-024-20257-0>
- [6] Dasari L. Prasanna; Suman Lata Tripathi, *Machine and Deep-Learning Techniques for Text and Speech Processing*, in Machine Learning Algorithms for Signal and Image Processing , IEEE, 2023, pp.115-128, doi: 10.1002/9781119861850.ch7.
- [7] Weston, J. & Watkins, C. (1999). *Multi-Class Support Vector Machine*.

- [8] Meng Joo Er, Rajasekar Venkatesan, and Ning Wang (2016) *An online universal classifier for binary, multi-class and multi-label classification*. In 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE Press, 003701–003706. <https://doi.org/10.1109/SMC.2016.7844809>
- [9] P. Vashistha, J. P. Singh, P. Jain and J. Kumar, *Raspberry Pi based voice-operated personal assistant (Neobot)*, 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2019, pp. 974-978, doi: 10.1109/ICECA.2019.8821892.
- [10] S. Umchid, V. Sutthipibul, A. Vorapantrakool, P. Vipattipumiprates and T. Wangkham, *Development of Voice Controlled Wheelchair for Persons with Physical Disabilities*, 2024 9th International Conference on Automation, Control and Robotics Engineering (CACRE), Jeju Island, Korea, Republic of, 2024, pp. 112-116, doi: 10.1109/CACRE62362.2024.10635015.
- [11] Kotsiantis, S. B. & Department of Computer Science and Technology, University of Peloponnese, Greece. (2007). *Supervised Machine Learning: A review of classification techniques*. In Informatica (Vol. 31, pp. 249–268). [https://datajobs.com/data-science-repo/Supervised-Learning-\[SB-Kotsiantis\].pdf](https://datajobs.com/data-science-repo/Supervised-Learning-[SB-Kotsiantis].pdf)
- [12] Aggarwal, S., Aggarwal, N., U.I.E.T., Panjab University, Chandigarh, & Dept of C.S.E., U.I.E.T., Panjab University, Chandigarh. (2011). *Classification of audio data using support vector machine*. International Journal of Computer Science and Technology, Vol. 2, Issue 3, September 2011, 398. <https://www.ijcst.com/vol23/2/shruti.pdf>
- [13] H. E. Semary, Khamis A. Al-Karawi and Mahmoud M. Abdelwahab. *Using Voice Technologies to Support Disabled People*. JDR 2024. Vol. 3(1). DOI: 10.57197/JDR-2023-0063
- [14] Mohsen Bakouri, *Development of voice control algorithm for robotic wheelchair using NIN and LSTM models*, Computers, Materials & Continua, vol. 73, no. 2, pp. 2441–2456, 2022, doi: 10.32604/cmc.2022.025106.
- [15] Y. Du, C. O'Connor, G. Byun, L. H. Kim, S. Amrgousian, and P. Vora, *Voice Assistive Technology for activities of Daily living: Developing an Alexa Telehealth training for adults with Cognitive-Communication Disorders*, CHI '24: Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems, pp. 1–15, May 2024, doi: 10.1145/3613904.3642788.
- [16] W.-Z. Zheng, J.-Y. Han, H.-L. Cheng, W.-C. Chu, K.-C. Chen, and Y.-H. Lai, *Comparing the performance of classic voice-driven assistive systems for dysarthric speech*, Biomedical Signal Processing and Control, vol. 81, p. 104447, Dec. 2022, doi: 10.1016/j.bspc.2022.104447.
- [17] Huo, X., Park, H., Kim, J., & Ghovanloo, M. (2013). *A dual-mode human computer interface combining speech and tongue motion for people with severe disabilities*. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 21(6), 979-991.
- [18] A. Jafari et al., *An embedded FPGA accelerator for a stand-alone dual mode assistive device*, in Proc. IEEE Biomed. Circuits Syst. Conf., Oct. 2017, pp. 1–4.
- [19] Sahadat, M. N., Alreja, A., Srikrishnan, P., & Ghovanloo, M. (2015, October). *A multimodal human computer interface combining head movement, speech and tongue motion for people with severe disabilities*. In 2015 IEEE biomedical circuits and systems conference (BioCAS) (pp. 1-4). IEEE.
- [20] H. Park and M. Ghovanloo (2014) *An Arch-Shaped Intraoral Tongue Drive System with Built-in Tongue-Computer Interfacing SoC*, Sensors, 14(11), 21565–21587, Nov. 2014, doi: 10.3390/s141121565.
- [21] Prasanna, D.L., Tripathi, S.L. (2024) *A Review on Tongue Based Assistive Technology, Devices and FPGA Processors Using Machine Learning Module*. Wireless Pers Commun 134, 151–170. <https://doi.org/10.1007/s11277-024-10897-8>
- [22] <https://www.analog.com/media/en/technical-documentation/obsolete-data-sheets/ADMP421.pdf>
- [23] Hassan, A.M.A., Mohsen, S. *Compression of electrocardiogram signals using compressive sensing technique based on curvelet transform toward medical applications*. Multimed Tools Appl 84, 11203–11219 (2025). <https://doi.org/10.1007/s11042-024-19328-z>
- [24] Do, D.-H., Chau, T.-D., & Tran, T.-S. (2023). *Speech feature enhancement based on time–frequency analysis*. ACM Transactions on Asian and Low-Resource Language Information Processing, 22(8), Article 219. <https://doi.org/10.1145/3605549>
- [25] Deshmukh, Ratnadeep & Kurzekar, Pratik & Waghmare, Dr. Vishal & Shrishrimal, Pukhraj. (2014). *A Comparative Study of Feature Extraction Techniques for Speech Recognition System*. International Journal of Innovative Research in Science, Engineering and Technology. 3. 18006-18016. 10.15680/IJIRSET.2014.0312034.
- [26] A. Ghosal, R. Chakraborty, R. Chakraborty, S. Haty, B. C. Dhara and S. K. Saha, *Speech/Music Classification Using Occurrence Pattern of ZCR and STE*, 2009 Third International Symposium on Intelligent Information Technology Application, Nanchang, China, 2009, pp. 435-438, doi: 10.1109/IITA.2009.427.
- [27] Joy, S., & Upadhyay, S. S. (2018). *Speech analysis in time and frequency domain*. International Journal of Engineering Research and Technology, 3. <https://api.semanticscholar.org/CorpusID:86769901>

- [28] Vimal, B. & Surya, Muthyam & Darshan, & Sridhar, V.S. & Ashok, Asha. (2021). *MFCC Based Audio Classification Using Machine Learning*. 1-4. 10.1109/ICCCNT51525.2021.9579881.
- [29] Gourisaria, M.K., Agrawal, R., Sahni, M. *et al. Comparative analysis of audio classification with MFCC and STFT features using machine learning techniques*. *Discov Internet Things* **4**, 1 (2024). <https://doi.org/10.1007/s43926-023-00049-y>
- [30] W&B. (n.d.). *Weights & biases*. W&B. <https://wandb.ai/mostafaibrahim17/ml-articles/reports/An-Introduction-to-Audio-Classification-with-Keras--Vmldzo0MDQzNDUy>
- [31] Khamparia, Aditya & Gupta, Deepak & Nhu, Nguyen & Khanna, Ashish & Pandey, Babita & Tiwari, Prayag. (2019). *Sound Classification Using Convolutional Neural Network and Tensor Deep Stacking Network*. *IEEE Access*. PP. 1-1. 10.1109/ACCESS.2018.2888882.
- [32] Pintér, I. (1996). *Perceptual wavelet-representation of speech signals and its application to speech enhancement*. *Computer Speech & Language*, 10(1), 1–22. <https://doi.org/10.1006/csla.1996.0001>
- [33] Kayed, Somaya & Elsayed, Ghada. (2023). *An optimizing technique for using MATLAB HDL coder*. *Bulletin of the National Research Centre*. 47. 10.1186/s42269-023-01066-1.
- [34] Khalid Zaman, Melike Sah, Cem Direkoglu and Masashi Unoki (2023) *A Survey of Audio Classification Using Deep Learning*, *IEEE Access*, Oct 2023, doi: 10.1109/ACCESS.2023.3318015
- [35] Zvi Kons and Orith Toledo- Ronen (2013) *Audio event classification using deep neural network*, *INTERSPEECH*, doi: 10.21437/Interspeech.2013-384A.
- [36] F. Rong (2016) *Audio Classification Method Based on Machine Learning*, *2016 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*, Changsha, China, pp. 81–84, doi: 10.1109/ICITBS.2016.98.
- [37] Prasanna D. L. and Tripathi S. L. (2022) *Machine Learning Classifiers for Speech Detection*, *2022 IEEE VLSI Device Circuit and System (VLSI DCS)*, pp. 143–147, doi: 10.1109/VLSIDCS53788.2022.9811452.
- [38] J. Kim et al. (2015) *Assessment of the Tongue-Drive system using a computer, a smartphone, and a Powered-Wheelchair by people with tetraplegia*, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 1, pp. 68–78, doi: 10.1109/tnsre.2015.2405072.
- [39] Kulkarni, Y. Pino, M. French, and T. Mohsenin (2016) *Real-time anomaly detection framework for many-core router through machine-learning techniques*, *ACM Journal on Emerging Technologies in Computer Systems*, 13(1), 1–22.
- [40] R. Sayed, H. Azmi, A. M. Nassar and H. Shawkey (2020) *Design Automation and Implementation of Machine Learning Classifier Chips*. *IEEE Access*, 8, 192155–192164, doi: 10.1109/ACCESS.2020.3032658.
- [41] Tripathi, S. L., Prasanna, D. L., & Mahmud, M. (2023). *A Low-Power audio processing using machine learning module on FPGA and applications*. In *Explainable Machine Learning Models and Architectures* (pp. 49–63). <https://doi.org/10.1002/9781394186570.ch4>
- [42] M. Papadonikolakis and C. -S. Bouganis (2012) *Novel Cascade FPGA Accelerator for Support Vector Machines Classification*, *IEEE Transactions on Neural Networks and Learning Systems*, 23(7), 1040–1052. doi: 10.1109/TNNLS.2012.2196446.
- [43] Audacity (Version 3.7.3). Software. Retrieved from Audacity website URL. For example: Audacity (Version 3.7.3) (2024). Audacity. Software. Retrieved from <https://www.audacityteam.org>.
- [44] Liu, Y., Lanier, J., Fu, K., & Sample, A. (2021). PDM-to-PCM microphone signal conversion on FPGA using CIC and FIR filters. *University of Michigan*. https://yatian-liu.github.io/public/PDM_PCM_Signal_Conversion_FPGA.pdf
- [45] Jinglei, Zhou & Chengliang, Li & Bo, Qi & Yanhui, Wei. (2010). CIC interpolation filter design in the audio decoder. 10.1109/CCTAE.2010.5544510.
- [46] Mohsen, S., Scholz, S.G. & Elkaseer, A. *Detection of COVID-19 in Chest X-Ray Images Using a CNN Model toward Medical Applications*. *Wireless Pers Commun* **137**, 69–87 (2024). <https://doi.org/10.1007/s11277-024-11309-7>
- [47] M. M. Zayed, S. Mohsen, A. Alghuried, H. Hijry and M. Shokair, *IoUT-Oriented an Efficient CNN Model for Modulation Schemes Recognition in Optical Wireless Communication Systems*, in *IEEE Access*, vol. 12, pp. 186836–186855, 2024, doi: 10.1109/ACCESS.2024.3515895
- [48] Chatterjee, S., Majumdar, A., & Roy, S. (2025). *A New Control Approach for a Multi-Controlled Wheelchair Utilizing Deep Learning Framework and Raspberry Pi*. *IETE Journal of Research*, 71(4), 1338–1355. <https://doi.org/10.1080/03772063.2025.2453880>
- [49] Prasanna, D. L., Tripathi, S. L., Mahmud, M., & Wijayanto, I. (2024). *High attenuation electromagnetic interface filter for effective processing of audio signals*. *Engineering Research Express*, 6(3), 035309. DOI: 10.1088/2631-8695/ad5fdf