© Universiti Tun Hussein Onn Malaysia Publisher's Office

JEVA

http://publisher.uthm.edu.my/ojs/index.php/jeva
e-ISSN : 2716-6074

Journal of Electronic Voltage and Application

Artificial Intelligent-Based Wake Word Detection at Edge Device

Lau Khai Shen[1], Solahuddin Yusuf Fadhlullah[1*], Khadijah Kamarulazizi[1], Samihah Abdullah[2], Shabinar Abdul Hamid[2]

[1]School of Engineering,
INTI International College Penang, Pulau Pinang, 11900, MALAYSIA

[2]School of Electrical Engineering, College of Engineering,
Universiti Teknologi MARA, Cawangan Pulau Pinang, 13500, MALAYSIA

*Corresponding Author

DOI: https://doi.org/10.30880/jeva.2023.04.02.003
Received 28 August 2023; Accepted 20 October 2023; Available online 28 December 2023

Abstract: Deep Neural Network based wake word (such as Hi Alexa or Hey Siri) systems allow increasingly accurate speech communication between humans and machines. However, this setup requires high processing power or cloud services which may not be accessible by edge devices. Currently, the accuracy of machine learning methods for cloudless edge devices in voice activation hovers below 90%. This paper explores wake word implementation on edge devices using a 2-Dimensional Convolutional Neural Network (CNN) with improved and balanced accuracy and latency. The proposed CNN model is created, trained and quantized using TensorFlow on a PC and exported to a Raspberry Pi Zero 2 W. The quantization method reduces the model size by 20% and spectral gating is adopted to lower wake word inaccuracy detection in moderately noisy environment. The proposed system achieved more than 90% wake word detection accuracy across 30 to 50 dB background noise with an average of 1.03 second of response time for the intended user. The result shows low-powered edge device still offers competitive performance for detecting wake word without cloud services.

Keywords: Wake word, convolutional neural network, edge device, raspberry pi, spectral gating

1. Voice Detection Using Edge Devices

Advancements in human language detection technology make virtual voice assistants such as Siri and Alexa very popular. Activating such a system requires accurate wake word detection before the Automatic Speech Recognition (ASR) program takes over the conversation between human and machine. Wake word detection is important as it can prevent accidental activation, limit energy consumption and address privacy concerns [1]. Figure 1 shows the Speech Dialogue System architecture where the wake word is implemented at the Acoustic Front-end Speech Recognition.

Deep learning models used for speech recognition can be implemented via edge devices or the cloud. Edge devices are much more popular over cloud due to lower latency, scalability and privacy [2]. Because the edge devices are localized, the deep learning models can react faster to inputs and at the same time reduce unauthorized access. Its scalable aspect comes from the ability to be operated from a graphics processing unit (GPU) down to a small embedded device such as a Raspberry Pi. With increasing demand in smart edge devices being small and powerful, practical and solid solutions to deep learning model in wake word detection is a must.

*Corresponding author: solahuddin21@gmail.com

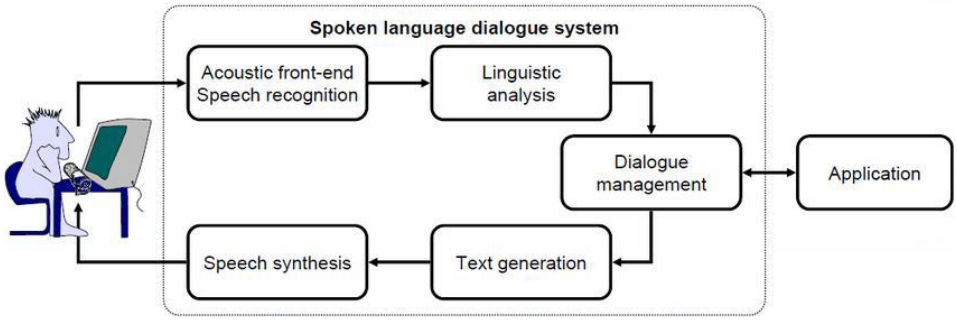


Fig. 1 - Speech Dialogue System architecture. [3]

Table 1 summarizes the related works that combined elements of wake word with machine learning (ML) in edge devices.

Table 1 - Comparison between related works on edge device

Author	Year	ML Model Type	Accuracy	Response Time	Hardware
[4]	2019	Snowboy	87%	2162ms	Raspberry Pi 3B+
[5]	2019	DWT-MFCC-SVM	80% (-10dB)	660 ms	Raspberry Pi 3
[6]	2019	WAT-MFCC-SVM	63.63% (-5dB)	n/a	Raspberry Pi 3B
[7]	2020	CNN-RNN-Softmax	80.41%	1010 ms	Raspberry Pi 3B+
[8]	2020	MobileNet- CNN-Softmax	79.2% (F1-score)	n/a	n/a
[9]	2019	MobileNet-LSTM-Softmax	86.5%	7300 ms	Raspberry Pi 4

The voice interaction system in [4] utilizes a microphone array extension board to capture audio input and an audio hard decoder extension board to provide audio feedback for matched skills. The system employs the Snowboy wake word detection engine, which is trained online and then downloaded onto the device, enabling offline wake-up. Upon detecting the wake word, the system employs cloud voice services like Baidu, Alibaba, Tencent, and IFLYTEK for recognizing input speech commands and synthesizing speech output. The authors conducted three tests to evaluate the system's performance. In an offline wake-up test within a 3-meter range, the system achieved a 95% wake-up rate in a quiet environment and 87% in a noisy environment.

The authors in [5] combined Discrete Wavelet Transform (DWT) and Mel Frequency Cepstral Coefficients (MFCC) to enhance feature extraction for real-time speech recognition system. The system also includes a Voice Activity Detector (VAD) to remove silent parts of the audio. Initially, DWT decomposes the speech signal, followed by Median Filtering (MF) to reduce noise in the decomposed signal components. Coefficients of approximation and detail are concatenated, and MFCC is applied to extract features. A Support Vector Machine (SVM) classifier is used for classification and recognition rates range from 80% to 100% in noisy settings.

The researchers in [6] utilized similar techniques as [5] in their speech recognition system, but with a different approach to the feature extraction stage. They used the technique of Wave Atoms Transform (WAT) in combination with the MFCC. The WAT is used to generate time-frequency representation of an audio signal when signal samples are decorrelated, to reduce redundancies into few coefficients.

EdgeRNN model for compact speech recognition was implemented in [7]. The system is composed of six layers: acoustic feature extraction, spatial information extraction, feature pooling, time information extraction, self-attention mechanism, and classification. Acoustic features, including Mel spectrogram, delta, and double-delta features, are processed using 1-Dimensional Convolutional Neural Network (1D-CNN) to extract spatial information, and Recurrent Neural Network (RNN) to extract temporal information, all in the frequency domain. A self-attention layer is introduced to enhance classification. Performance tests revealed that the proposed EdgeRNN achieved an average accuracy of 80.41%.

In [8], the authors devised a speech recognition system tailored for resource-constrained devices using MobileNet and 1-D CNN. Rather than relying on popular datasets, the authors utilize available audio samples and augment them through techniques like pitch shifting and speed tuning. FFT generates spectrograms, which MobileNet processes for

feature extraction while addressing gradient vanishing using ReLU activation. The SoftMax layer in MobileNet is replaced with a 1-D CNN model to enhance system robustness. After passing through the 1-D CNN, input data is reduced, flattened, and classified in the SoftMax layer. The proposed system achieved an average F1-score of 79.2% across all utilized datasets.

The researchers in [9] introduced a system that combines MobileNet (2D-CNN) and LSTM models for lip reading. This approach is most similar to [8], although their focus is not directly related to wake word or speech recognition. In this case, LSTM is used to capture sequence features from past and future input data. Dropout is employed to prevent overfitting. Test outcomes indicated an 86.5% accuracy with a response time of 7300ms. The higher response time is attributed to the inclusion of Multi-Task Cascaded Convolutional Neural Networks (MTCNN), which increased computational demands.

The project presented in this write up took off as Motorola Solutions were seeking further exploration from the academia for practical and lightweight wake word detection on edge devices, without cloud access or powerful processors. Three objectives were set for this research as follows. First, the implementation of a CNN algorithm for wake word detection, which is compatible and stable on a Raspberry Pi (edge device). Second, the detection accuracy must be at least 75% for a dedicated user in a moderately noisy environment and third, the response time of not more than 2 seconds. The algorithm is created and trained on a personal computer (PC) using TensorFlow and then exported and run on a Raspberry Pi Zero 2 W with TensorFlow Lite library installed.

2. Methodology

Figure 2 shows the block diagram of the proposed system. Unwanted background noise is first removed by passing through a finite impulse response (FIR) band-pass filter to normalize the waveform's amplitude. Noise frequencies within the band-pass are suppressed using spectral gating [10] with a set threshold. The frequency features over time of the denoised speech input are extracted using Short-Time Fourier Transform (STFT) [11] and projected onto a log scaled mel spectrogram by Mel Frequency Cepstral Coefficients (MFCC) method [12-14]. A log mel spectrogram is used as it is able to express the frequency in a range that is similar to human perceived hearing. Once the log mel spectrogram is created, the speech features presented on the spectrogram is analyzed by the Slim MobileNet (2D-CNN) and long short-term memory (LSTM) blocks. Slim MobileNet improves MobileNet-V1 model by reducing both the file size and the number of layers but with improved accuracy [15].

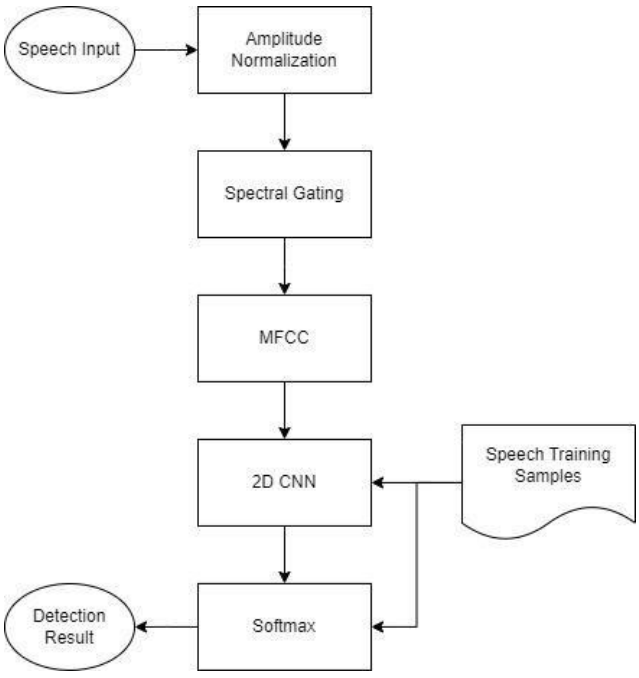


Fig. 2 - Algorithm of the proposed system

Two deep neural network (DNN) blocks are used to improve the accuracy of the feature filtering and reduction. Slim MobileNet is used to reduce the system computational power while retaining performance, whereas the LSTM is used to reduce the effects of gradient exploding and vanishing. The combined DNN blocks are responsible for filtering and reducing the speech features before passing it to the support vector machine (SVM) classifier block for result

classification. The final layer of the model uses Softmax activation to output the detection results as percentage over the classes of “wake word” and “non wake word”.

The DNN blocks of the system including the SVM classifier block are trained using speech samples as datasets. The speech samples are obtained through means of live recording in quiet and noisy environments for the dedicated user. Pitch shifting techniques are then used to produce more speech samples and reduce data overfitting.

3. Results and Discussion

3.1 Detection Accuracy

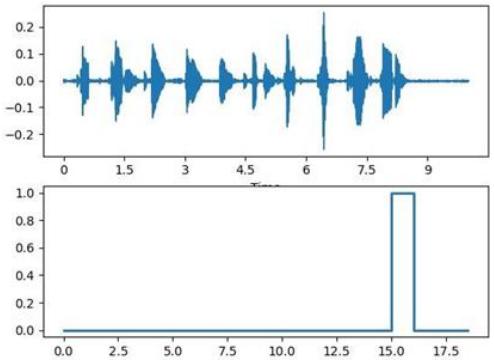
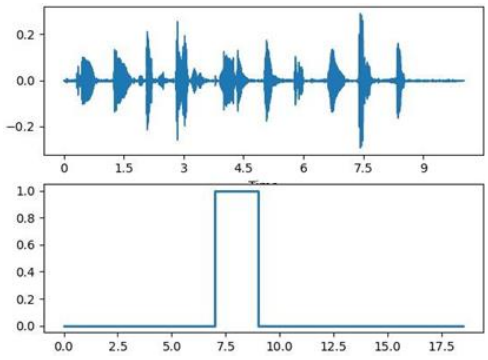
The wake word trained is ‘Marvin’ which is inserted within a sentence of ten words to trigger an activation response from the system. Non-‘Marvin’ words that do not trigger an activation are assigned as true negative. The average distance between the user to the microphone is 20 centimeters.

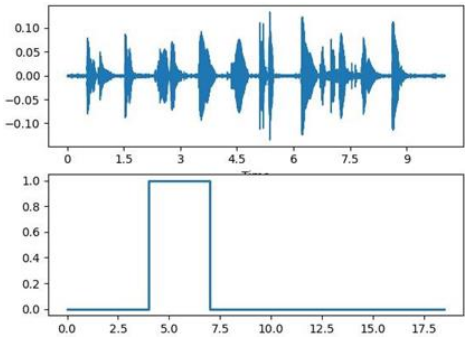
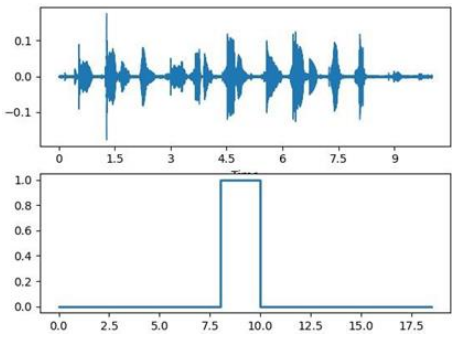
The results for detection accuracy are shown in the order of:

- Session 1 - controlled (noiseless) condition
- Session 2 – slightly noisy background between 30 to 50 dB of noise

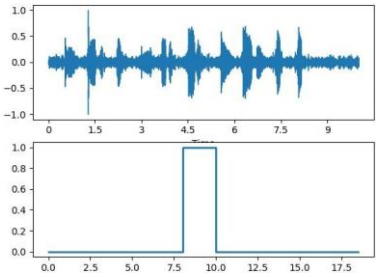
The system achieved a 98% wake word detection accuracy for 50 samples in Session 1 and several results are shown in Table 2. The pulses indicate the locations of detected wake word in each recording. As expected, the noiseless recording under controlled condition allows the system to easily identify the presence and location of the wake word in the sentences.

Table 2 - Selected sessions from controlled condition

Phrase	Detection Result
Get ready for school in fifteen minutes. Look here, Marvin.	 Correct detection
Hold the book correctly, Marvin and keep your head straight.	 Correct detection

Anything else, Marvin? Are you sick or suffering any illness?	 Incorrect detection. One false positive on the fourth word. Two words were detected as Marvin.
Be careful and be happy, Marvin. Be cheerful with that.	 Correct detection

For noisy background (Session 2), the detection is accurate for up to 50 dB for practical usage. This means that ambient noise of more than 50 dB would severely affect the detection of the wake word due to the presence of unwanted signals in both amplitudes and frequencies. The detection would improve by speaking very loud or close to the microphone. Detection results for up to 50 dB noise are shown in Figure 3 to 6. Only selected results are shown.


50 dB (correct)

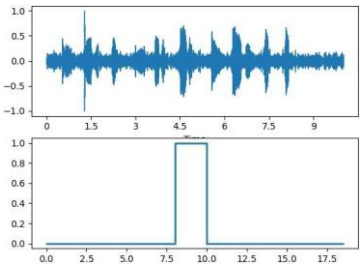
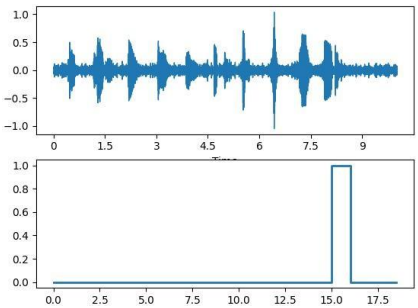

30 dB (correct)

Fig. 3 - Test sentence in a vehicle: Be careful and be happy, Marvin. Be cheerful with that


50 dB (correct)

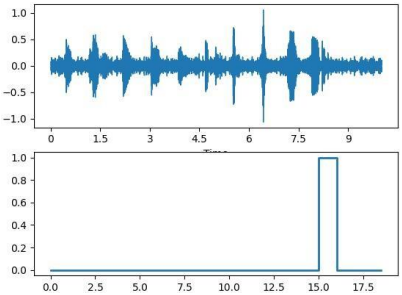
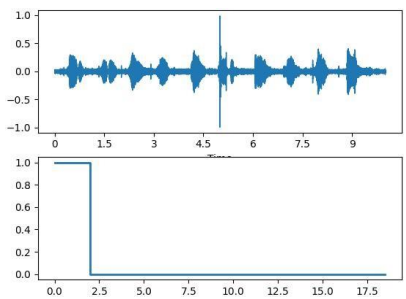
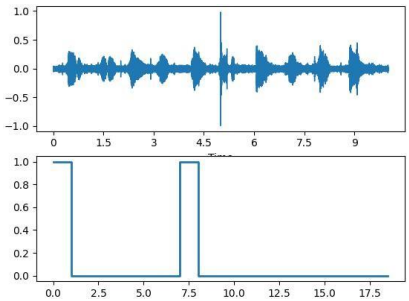

30 dB (correct)

Fig. 4 - Test sentence in a vehicle: Get ready for school in fifteen minutes. Look here, Marvin

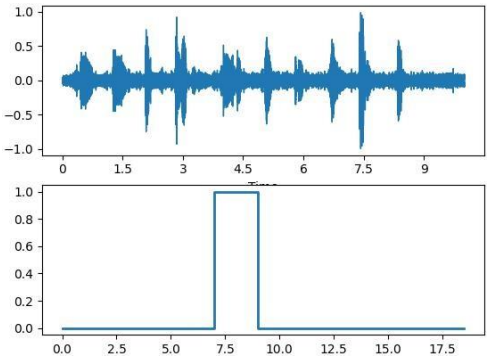


50 dB (correct)

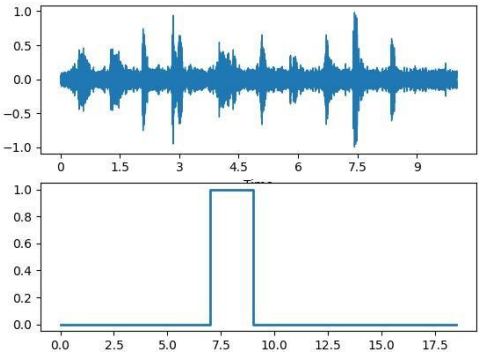


30 dB (1 false positive)

Fig. 5 - Test sentence at a food court: Marvin, cover your mouth while coughing. Do you want food?



50 dB (correct)



30 dB (correct)

Fig. 6 - Test sentence at a shopping centre: Hold the book correctly, Marvin and keep your head straight

Other factors that have been verified to affect the detection accuracy is the quality of microphone, its distance from the user and sound reflection from the placement of microphone on solid materials. The summarized result for Session 2 is given in Table 3. Overall, the prediction accuracy is 93.7 % for 150 samples. Thus, the results obtained validate the feasibility of the proposed CNN wake word detection system.

Table 3 - Detection accuracy for 30 to 50 dB ambient noise level

Noise level (dB)	True Positive	False Positive	True Negative	False Negative	Accuracy (%)
50	15	8	127	0	94.7
30	13	9	126	2	92.7

3.2 Response Time

The response time for 50 tested samples is show in Figure 7. The maximum response time taken is 9.23 seconds during the first sample detection, whereas the minimum response time taken is 0.94 seconds during 7th and 38th prediction. The response time for the first detection is the longest due the initialization of the system and resources allocation to run the CNN model. Excluding the initialization process, the average response time is 1.03 second.

The algorithm takes only 8 kB of space (with a 20 % size reduction from quantization implemented) and run in a Raspberry Pi with quad-core 64-bit ARM Cortex-A53 processor clocked at 1 GHz and 512 MB of SDRAM.

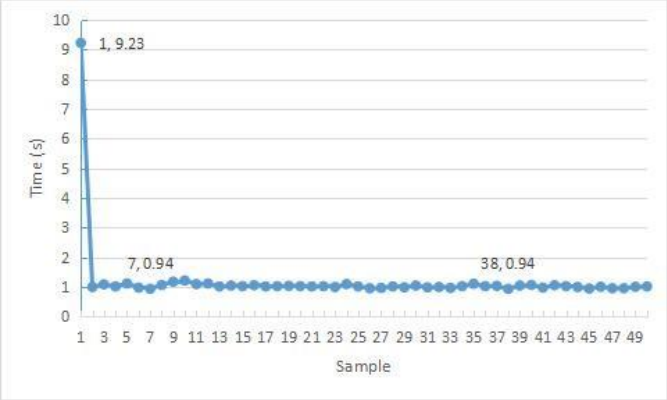


Fig. 7 - Response time graph for 50 predictions

4. Conclusion

The proposed wake word detection system was successfully implemented and run independently on an edge device without the need of cloud access or high computing power. In terms of performance, the system is able to detect correctly more than accuracy 90 % of the wake word samples albeit only for a dedicated user and in a low-level noise ambient from 30 dB to 50 dB. The average detection response time is 1.03 second after initialization process is complete. The proposed algorithm showed improvement in the accuracy and latency relative to the literature in a lightweight 8 kB file size, but challenges remain in noisy environment where limited processing power and quality of device are the major factors impeding the detection accuracy.

5. Future Recommendations

Two factors that are not being investigated but will affect the overall accuracy of the system are the wake word triggering distance and the quality of microphone used. For example, if the microphone already has a strong built-in noise filter, then the quality of waveform captured is higher that would result in a higher detection accuracy. Alternatively, a software-based noise filter can also be improved on the edge device to get better waveform quality. To further increase the detection accuracy, a dedicated personalized model training, which captures the accent and voice characteristics of the sole intended user should be done. We have pursued this approach with positive results but is not covered within the scope of this project.

Acknowledgement

Our highest gratitude to Motorola Solutions and Career Services of INTI International College Penang for facilitating and supporting this industrial employer project.

References

[1] Hossain, D., & Sato, Y. (2021). Efficient Corpus Design For Wake-Word Detection. In 2021 IEEE Spoken Language Technology Workshop (SLT), pp.1094-1100.

[2] Chen, J., & Ran, X. (2019). Deep Learning With Edge Computing: A Review. Proceedings of the IEEE, 107(8), 1655-1674.

[3] Bako, B. Z., Könings, B., Schaub, F., Wiedersheim, B., & Weber, M. (2010). Proceedings of the Seminar: Research Trends in Media Informatics. In Research Trends in Media Informatics. Universität Ulm.

[4] Yang, D., Ma, L. & Liao, F. (2019). An Intelligent Voice Interaction System Based On Raspberry Pi. 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), vol. 1, pp. 237-240.

[5] Mnassri, A., Bennasr, M. & Adnane, C. (2019). A Robust Feature Extraction Method For Real-Time Speech Recognition System On A Raspberry Pi 3 Board. In Engineering, Technology & Applied Science Research, 9(2), pp. 4066-4070.

[6] Walid, M., Bousselmi, S., Dabbabi, K. & Cherif, A. (2019). Real-Time Implementation Of Isolated-Word Speech Recognition System On Raspberry Pi 3 Using WAT-MFCC. International Journal Of Computer Science And Network Security, 19(3), pp. 42-49.

[7] Yang, S., Gong, Z., Ye, K., Wei, Y., Huang, Z. & Huang, Z. (2020). Edgernn: A Compact Speech Recognition Network With Spatio-Temporal Features For Edge Computing. IEEE Access, vol. 8, pp 81468-81478.

[8] Tan, P. S., Lim, K. M., Lee, C. P. & Tan, C. H. (2020). Acoustic Event Detection with MobileNet and 1D-Convolutional Neural Network. 2nd IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET), pp. 1-6.

[9] Wen, J., & Lu, Y. (2019). Automatic Lip Reading System Based on a Fusion Lightweight Neural Network with Raspberry Pi. Applied Sciences, 9(24), 5432.
[10] Braun, S., Gamper, H., Reddy, C. K. & Tashev, I. (2021). Towards Efficient Models For Real-Time Deep Noise Suppression. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 656-660.
[11] Krishnan, S. (2021). Advanced Analysis Of Biomedical Signals. In Biomedical Signal Analysis For Connected Healthcare, pp. 157-222.
[12] Badi, A., Ko, K. & Ko, H. (2019). Bird Sounds Classification By Combining PNCC And Robust Mel-Log Filter Bank Features. The Journal of the Acoustical Society of Korea, 38(1), pp. 39-46.
[13] Chakraborty, K., Talele, A. & Upadhya, S (2014). Voice Recognition Using MFCC Algorithm. International Journal of Innovative Research in Advanced Engineering (IJIRAE), 1(10), pp. 2349-2163.
[14] Janse, P. V., Magre, S. B., Kurzekar, P. K. & Deshmukh, R. (2014). A Comparative Study Between MFCC And DWT Feature Extraction Technique. International Journal of Engineering Research and Technology, 3(1), pp. 3124-3127.
[15] Bouguezzi, S., Faiedh, H., & Souani, C. (2021). Slim MobileNet: An Enhanced Deep Convolutional Neural Network. 18th IEEE International Multi-Conference on Systems, Signals & Devices, pp. 12-16.