Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2024 SCPE. Volume 25, Issues 2, pp. 1191–1198, DOI 10.12694/scpe.v25i2.2624

SECURITY ENABLED NEW TERM WEIGHT MEASURE TECHNIQUE WITH DATA DRIVEN FOR NEXT GENERATION MOBILE COMPUTING NETWORKS

ANIL KUMAR BUDATI[*] SHAYLA ISLAM[†] SHAIK MOHAMMAD RAFEE[‡] CHENGAMMA CHITTETI[§] AND T. LAKSHMI NARAYANA[¶]

Abstract. In the field of ASIC and FPGA, Machine Learning (ML) techniques play a major role and become predominant for accurate results for different applications like big data analysis and automotive electronics, and driverless vehicles which are required speed and power savings. Due to increasing the demand for higher accuracy, low power, low area consumption, and higher throughput for the complexity of the designs in the latest technology, the proposed system is fulfilling these demands in ASIC and FPGA domains, reconfigurable hardware architecture has been proposed it consists of an ML-based Support Vector Machine (SVM), high-speed AHB protocol and Floating point (FP) operations and also the system has the flexibility to communicate with I2C and I2S protocols. In order to increase throughput with minimal latency, the proposed architecture with AHB protocol and AHB to APB bridge is incorporated between the fabric dynamically reconfigurable multi-processor (FDPM) and peripherals along with security algorithms using SHA-256bits and AES. In order to perform ML-based applications, the proposed system is incorporated double-precision floating point (DPFP) arithmetic operations. The overall proposed architecture is developed in Verilog HDL and quality checking using the LINT tool and Clock Domain Crossing (CDC) using Spyglass tool and synthesized using DC compiler for ASIC and Vivado Design Suite 2018.1 for FPGA implementation and verification. The entire design is interfaced with the Zynq processor and SDK tool to verify data transfer between hardware and software. The obtained results show the generated custom accelerator is able to compute any complex ML classifiers for a larger amount of data. The obtained results are compared with existing state-of-art results and found that 18 % improvement in throughput, a 21 % improvement in power consumption savings, and a 34 % reduction in latency.

Key words: Speech recognition, Human-machine interface system, CNN, ASR.

1. Introduction. Spoken language serves as a natural means of human communication. Despite our continued reliance on hardware interfaces like keyboards and mice to engage with computers, there's a growing need for a software interface that emulates natural communication—enter the automatic speech recognition (ASR) system [1]. ASR, also known as speech recognition (SR), translates spoken words into written text or symbols [2]. The popularity of ASR has surged alongside the rise of new technologies such as robots, autonomous cars, cell phones, and smartwatches, and it has remained a dynamic field of research in human-computer interaction (HCI) in recent years. Its applications span various domains, including security and surveillance systems, automated credit card activations, and voice-controlled functionalities.

Additionally, isolated word recognition systems offer diverse applications in banking automation, voice-activated dialing, PIN code-operated devices, and data entry automation [3]. The extensive range of applications necessitates the development of ASR systems for all languages. However, crafting a universal ASR system to accommodate the approximately 6900 languages worldwide is unfeasible [3-4]. While some languages boast standardized SR systems, others must improve such technological advancements, particularly under-resourced ones.

SR poses a formidable challenge due to the extensive variability in speaker attributes encompassing different languages, varied vocabularies, diverse speaking styles, and unpredictable environmental noises [5]. The speech patterns of multilingual individuals, different genders, and individuals with distinct social types or di-

[*]Department of ECE, Koneru Lakshmaiah Education Foundation, Hyderabad, India and ICSDI, UCSI University, Malaysia. (anilbudati@gmail.com)
[†]ICSDI, UCSI University, Kuala Lumpur, Malaysia. (shayla@ucsiuniversity.edu.my)
[‡]AI&ML Department, Sasi Institute of Technology & Engineering, Tadepalligudem, India (mdrafee1980@gmail.com)
[§]Department of Data Science, School of Computing, Mohan Babu University, Andhra Pradesh, India (Sailusrav@gmail.com)
[¶]Department of Electronics and Communication Engineering, Kandula Lakshumma Memorial College of Engineering for Women, Kadapa, Andhra Pradesh, India (lakshmi.svuniversity@gmail.com)

alects exhibit substantial variations [6-7]. Over recent decades, researchers have consistently embraced new technologies and methodologies to confront these hurdles and devise improved solutions. Radha et al. [8] have specifically addressed some of these challenges by categorizing them into three main groups: types of speech utterance, speaker model variations, and vocabulary types. These speech utterance categories cover isolated words, connected words, continuous speech, and spontaneous speech. Speaker model variations include both speaker-independent and speaker-dependent models. A significant obstacle for local languages lies in the need for more resources for adequately training the model, particularly in acquiring a corpus with substantial and relevant data.

Ali et al. [4] constructed a database involving 50 speakers (25 male and 25 female) reciting digits from zero to nine. In this study, the dataset used by us, suffers from significant noise interference and mispronunciation issues. Moreover, their proposed model achieves a classification accuracy of only 76.8 %, indicating room for improvement. Similarly, Ahmad et al. [9] created a Pashto corpus comprising 161 words articulated by 50 speakers (25 male and 25 female). They employed a linear discriminant analysis (LDA) classifier for digit classification, with the word error rate reaching up to 60 % for the initial ten words. Nisar et al. [10] developed another database for Pashto digits, incorporating 150 speakers (75 males and 75 females). They used k-nearest neighbor (k-NN) and support vector machine (SVM) classifiers for classification. The SVM achieved an overall accuracy of 91.5 %, while the k-NN reached 87.75 %, demonstrating satisfactory performance but leaving room for enhancement.

Veisi et al. [11] proposed an automatic speech recognition (ASR) system for a local language, Persian. They utilized the Farsdat dataset, comprising 6080 Persian audio signals sampled at 16 kHz, to train the model and evaluate its performance. They employed Mel-frequency cepstral coefficients (MFCC) algorithms to extract features from each signal, which were then inputted into a deep belief network (DBN) to enhance the model's performance. Within the DBN architecture, a DBN autoencoder was utilized for feature extraction. For training the Acoustic Model (AM), the authors experimented with four network models: long short-term memory (LSTM), bidirectional long short-term memory (BLSTM), Deep Long short-term memory (DLSTM), and deep bidirectional long short-term memory (DBLSTM). The results obtained from these models were compared against a Hidden Markov Model (HMM), which the author and Kaldi-DNN implemented. The test accuracy %ages of different models were as follows: 75.2 % for HMM, 77 % for LSTM, 78 % for LSTM-DBN, 79.3 % for BLSTM, 80.3 % for DLSTM, and 82.9 % for DBLSTM.

In [12], an innovative system for recognizing isolated Persian digits was introduced. The primary emphasis was on effectively classifying Persian digits with notably similar phonetic and spectral characteristics. 450 Persian speech samples were gathered, covering digits from one to nine. Among these, 330 samples were allocated for training the model, while 120 were used for validation. The dataset comprised 400 male and 50 female speech samples, encompassing various age groups. Thirteen Mel-frequency cepstral coefficients (MFCC) were extracted for each digit and were input for the HMM-SVM (Hidden Markov Model-Support Vector Machine) model. The model's performance was assessed in both noisy and clean environments, achieving an accuracy of 98.59 % in a clean environment and 68.45 % in a noisy one.

Social media platforms like Twitter and Facebook provide an accessible outlet for individuals to share their thoughts using text, images, and videos on a multitude of topics. Users across different age groups frequent these platforms, flooding them with vast data. Regrettably, the lack of efficient tools to manage abusive comments has facilitated the propagation of false information and hateful speech, causing harm to individuals' standing in society. It's incumbent upon both these platforms and governments to collaborate and enforce strategies that curtail the dissemination of such harmful content before it gains traction among a broader audience. In the current landscape dominated by social media platforms, identifying and preventing hate speech has become increasingly crucial. It's imperative to detect and stop harmful or deceptive language from spreading across users to prevent adverse impacts on society. My research focus centers on reducing the presence of hate speech in mobile data exchanges. Although machine learning methods have been used in past studies to identify hate speech, there's a recognized necessity for higher accuracy. To tackle this issue, researchers have developed a novel approach, aiming to significantly improve the precision of detecting hate speech in mobile communication.

2. Material and methods. This section illustrates the materials employed and the methods for devising the detection system targeting abusive or foul language in speech transmission within mobile networks. The

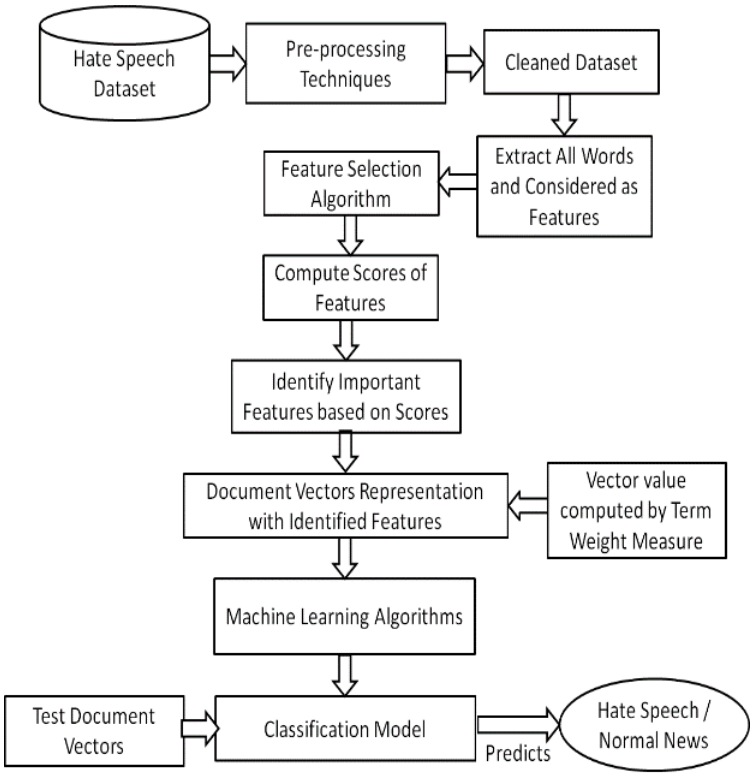

Fig. 2.1: The proposed ASR methodology

proposed ASR methodology is depicted in Figure 2.1, encompassing several key components and steps, notably involving the utilization of a refined dataset. Obtaining and preprocessing data are crucial steps in developing a database. Once the data from speakers is collected and preprocessing steps are completed, datasets were created at sampling frequencies of 16,000 Hz and 44,100 Hz. Using the MFCC algorithm, each digit's audio file is transformed into a 2-dimensional array, serving as input for the proposed CNN. A portion of the dataset is allocated for training the CNN model, while the rest is used for model validation. The CNN model is a digit classifier, discerning digits based on shared features. This study employed the English dataset sourced from subtask A of the 2019 FIRE competition [13]. This specific subtask aimed to identify whether the given text contained hate speech or offensive language.

ASR encounters a major challenge due to the considerable variability in speech signals, prompting the highly recommended practice of feature extraction to mitigate this variability [13]. Feature extraction is pivotal in ASR, aiming to eliminate unnecessary details from speech and facilitate speaker-independent recognition by converting the speech signal into a digital form and evaluating its attributes [14][15]. Various techniques such as Linear Predictive Code (LPC), Perceptual Linear Prediction (PLP), and Mel-Frequency Cepstral Coefficient (MFCC) serve as feature extraction methods [17], with MFCC being particularly prevalent in automatic speech recognition research [18]. In this research, MFCC is prioritized over others due to its ability to encompass the signal's temporal and frequency aspects. Moreover, it is favored for its adeptness in handling dynamic features and extracting linear and non-linear elements. While diverse MFCC coefficients were removed, the CNN model exhibited optimal performance using 20 coefficients for the current datasets. The process of MFCC feature extraction involves pre-emphasis, framing, windowing, Fast Fourier Transform (FFT), Mel filter bank application, and Discrete Cosine Transform (DCT) computation [19].

Step:1 During this phase, every digit sample undergoes filtration that accentuates higher frequencies, amplifying the signal's energy. If Y(n) represents the output emphasis signal and X(n) denotes the input

signal, their relationship can be expressed through Equation (2.1).

$$Y[n] = X[n] - \alpha X[n-1] \tag{2.1}$$

Step:2 During this stage, the speech signal undergoes segmentation into smaller segments typically lasting between 20 to 40 milliseconds, known as frames. The voice signal is partitioned into N samples, with consecutive frames separated by a M margin (where M<N). Commonly, the values chosen for N and M are 256 and 100, respectively.

Step:3 To maintain the signal's continuity, every frame derived from the preceding step undergoes multiplication by the Hanning window. The resulting output of the Hanning window is denoted as Y(n), obtained by multiplying the input X(n) with the Hanning window W(n) as expressed in Equation (2.2).

$$Y(n) = X(n)xW(n) \tag{2.2}$$

The Hanning window W (n) is defined by Eq. (2.3)

$$W(n) = 0.24 - 0.16\cos(\frac{3\pi n}{N_2}), where 0 \leq n \geq N-1 \tag{2.3}$$

Step:4 During this stage, each frame transforms from the time domain to the frequency domain to obtain the magnitude frequency responses for each frame. The result of the Fourier Transform is denoted as Y(n) and is defined by Equation (2.4).

$$Y(n) = \sum_{n=-a}^{a} X(n)\exp^{-iwn} \tag{2.4}$$

Step:5 During this phase, a multi-filter bank is employed to compute the average energies within each block and subsequently derive the logarithm of all filter banks, as specified in Equation 2.(5). This transformation is necessary because the voice signal's behavior doesn't adhere to a linear scale due to the Fourier Transform's wide frequency range.

$$F(Mul) = 1024X\ln(2.4 + \frac{f}{800}) \tag{2.5}$$

A Convolutional Neural Network (CNN) is a type of feed-forward artificial neural network that distinguishes itself by employing specialized convolutional layers instead of traditional fully connected hidden layers [5][26]. Unlike a standard feed-forward neural network, each neuron within the convolutional layer connects exclusively to a small region of the preceding layer, termed the local receptive field [27]. While convolutional and fully connected layers possess parameters, pooling and non-linearity layers do not. Due to their exceptional performance, CNNs have gained widespread popularity across various machine learning domains such as computer vision, pattern recognition, and natural language processing (NLP) [26]. Hence, we opted for CNNs to perform digit classification. Within the convolutional layer, the input data interacts with multiple filters sliding over this data to extract features. A filter, also known as a convolutional kernel, consists of elements within a matrix that undergo training via backward propagation. This layer's outcome results from the sum of the products derived by multiplying each input element with its respective filter element. The designed CNN employs three convolutional layers [28], with the initial layer as the input and the second and third layers as hidden layers.

3. Results and Discussions. The research entails using a CNN-ASR and five different TWMs to evaluate term values in document vectors. The process involved identifying the top 8000 terms, starting with 1000 terms initially and increasing by 1000 in each step. The CNN-ASR classifier was utilized to train the classification model. Table 1 outlines the assumptions created in the simulated setting, which helped train and test the dataset with the proposed CNN-ASR classifier.

Table 3.2 presents the experiment's results, encompassing training and testing 8000 samples to detect hate speech words. The evaluation of the research's effectiveness included running previous methodologies

Table 3.1: Assumption

Machine Learining Algorithm	CNN-ASR
Top Scored terms	800
Experiment initiated terms for every cycle	900
Enhanced words for every cycle	900
Language	English
HOF Trained data set	2151
NOT Trained data set	3271

Table 3.2: Table with 7 Columns and 9 Rows

TWMs number of feature samples	TF	TFIDF	TFIE	TFRF	TF PROB	CNN ASR Proposed 7
1000	0.590	0.671	0.695	0.753	0.761	0.779
2000	0.603	0.685	0.707	0.764	0.773	0.787
3000	0.615	0.689	0.725	0.770	0.795	0.806
4000	0.630	0.696	0.738	0.778	0.802	0.811
5000	0.647	0.702	0.749	0.791	0.809	0.818
6000	0.659	0.716	0.752	0.817	0.824	0.834
7000	0.671	0.727	0.758	0.821	0.831	0.839
8000	0.663	0.733	0.769	0.824	0.846	0.855

on identical feature datasets. As indicated in Table 3.2, the proposed TWM exhibited a notable accuracy of 0.855 in detecting hate speech, surpassing other TWMs when working with 8000 samples. The findings highlighted improved accuracy as the number of terms used to represent document vectors increased. However, the accuracy of hate speech detection declined when experiments utilized more than 8000 terms for document vector representation.

Figure 3.1 depicts the relationship between probability and the quantity of samples for both the current and proposed models. With an increase in sample size, there is a concurrent elevation in accuracy. Notably, the proposed CNN-ASR model demonstrates superior performance compared to the existing model as the sample size progresses from 1000 to 8000. This advancement results in an 18 % enhancement across all samples

Figure 3.2 displays the correlation between probability and the quantity of features in both the current and proposed models. With an increase in samples from 1000 to 8000, the proposed PTWM model outperforms the existing TFIDF and TFIEF models. In particular, 5 % and 12 % enhancement are observed across all samples when comparing the proposed CNN-ASR model to the existing TFIDF and TFIEF models, respectively.

Figure 3.3 presents the relationship between probability and the quantity of features in both the current and proposed models. With an increase in the number of samples, there is a corresponding increase in accuracy. Specifically, the proposed model showcases better performance than the existing models as the number of samples increases from 1000 to 8000. A noticeable 6 % and 3 % improvement is evident across all samples when comparing the proposed CNN-ASR model to the existing TFRF and TF-PROB models, respectively.

Figure 3.4 demonstrates the correlation between probability and the quantity of train and test samples about the sensitivity parameter. At a sample size of 1000, the proposed method achieves a sensitivity of approximately 93 %, surpassing the sensitivities obtained by the existing TF and TFIEF methods, which are 83 % and 89 %, respectively. This heightened sensitivity of the proposed method remains consistent even when the sample size is expanded to 2000. Consequently, the proposed method consistently displays higher sensitivity than the two existing methods.

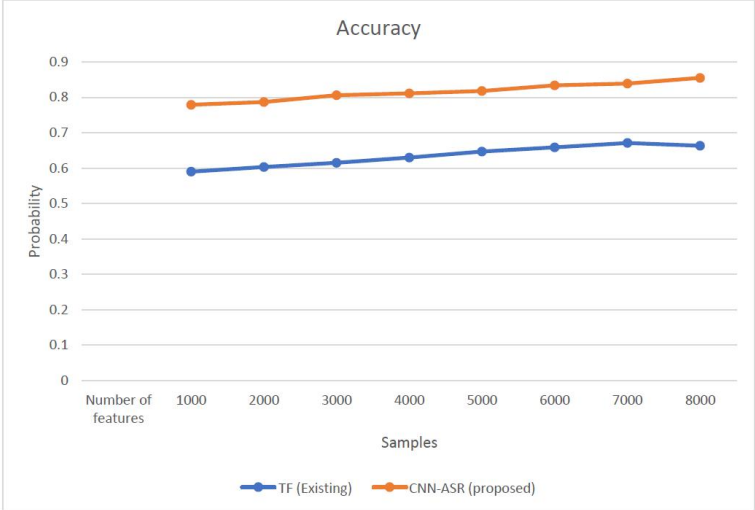

Fig. 3.1: TF and CNN-ASR accuracy comparison

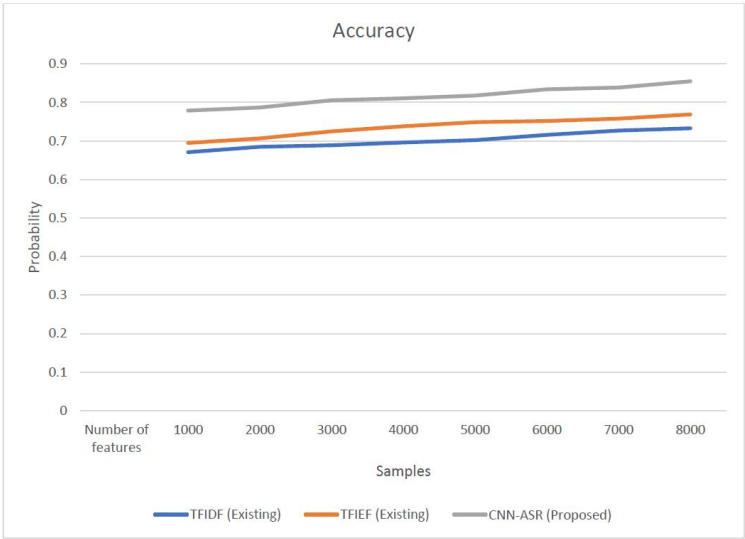

Fig. 3.2: TFIDF, TFIEF and CNN-ASR accuracies comparison

4. Conclusion. Speech recognition is one of the dominant ways of human-machine interaction and become advanced for different languages. However, unfortunately, this field is still immature for detecting foul/abused words in mobile networks. The experimental results of our proposed study are compared with already existing work, which is much better than the earlier approaches, as shown in Table 3.2. The CNN-ASR model performs better in the case of a large amount of data, but unfortunately, there is a limited amount of data in our case. The study involved implementing a CNN-ASR model and five established term weight measures. It was observed that the proposed CNN-ASR attained the highest accuracy of 85.5 % in detecting hate speech (abuse) within mobile networks. Additionally, the experiment evaluated the sensitivity parameter, highlighting that the proposed CNN-ASR classifier outperformed the existing methods in terms of performance.

Fig. 3.3: TFRF and TF-PROB and CNN-ASR accuracies comparison

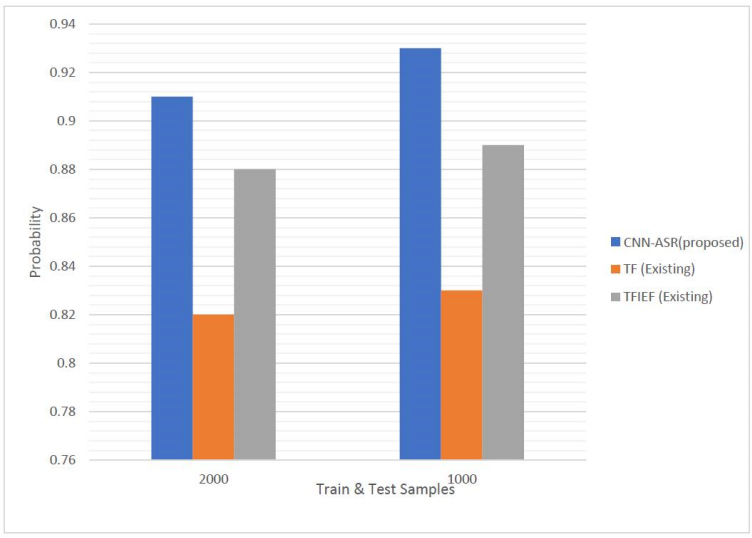

Fig. 3.4: TF, TFIEF and CNN-ASR comparison for Sensitivity parameter

REFERENCES

[1] M. Dua, R. Aggarwal, V. Kadyan, and S. Dua, "Punjabi automatic speech recognition using HTK," International Journal of Computer Science Issues (IJCSI), vol. 9, p. 359, 2012.

[2] M. Joshi and S. Srivastava, "Human Computer Interaction Using Speech Recognition Technology," in National Conference on Mathematical Analysis and Computation (NCMAC), 2015.

[3] Ali, H., Jianwei, A. and Iqbal, K., 2015. Automatic speech recognition of Urdu digits with optimal classification approach. International Journal of Computer Applications, 118(9), pp.1-5.

[4] L. Besacier, E. Barnard, A. Karpov, and T. Schultz, "Automatic speech recognition for under-resourced languages: A survey," Speech communication, vol. 56, pp. 85-100, 2014.

[5] Z. Ali, A. W. Abbas, T. Thasleema, B. Uddin, T. Raaz, and S. A. R. Abid, "Database development and automatic speech recognition of isolated Pashto spoken digits using MFCC and K-NN," International Journal of Speech Technology, vol. 18, pp. 271-275, 2015.

[6] O. Abdel-Hamid, "rahman Mohamed," A., Jiang, H., Deng, L., Penn, G., and Yu, D, pp. 1533-1545, 2014.

[7] M. Malik, M. K. Malik, K. Mehmood, and I. Makhdoom, "Automatic speech recognition: a survey," Multimedia Tools and Applications, vol. 80, pp. 9411-9457, 2021.
[8] A. Y. Vadwala, K. A. Suthar, Y. A. Karmakar, and N. Pandya, "Survey paper on different speech recognition algorithm: challenges and techniques," Int J Comput Appl, vol. 175, pp. 31-36, 2017.
[9] V. Radha and C. Vimala, "A review on speech recognition challenges and approaches," doaj. org, vol. 2, pp. 1-7, 2012.
[10] I. Ahmed, N. Ahmad, H. Ali, and G. Ahmad, "The development of isolated words Pashto automatic speech recognition system," in 18th International Conference on Automation and Computing (ICAC), 2012, pp. 1-4.
[11] S. Nisar, I. Shahzad, M. A. Khan, and M. Tariq, "Pashto spoken digits recognition using spectral and prosodic based feature extraction," in 2017 Ninth International Conference on Advanced Computational Intelligence (ICACI), 2017, pp. 74-78.
[12] H. Veisi and A. H. Mani, "Persian speech recognition using deep learning," International Journal of Speech Technology, vol. 23, pp. 893-905, 2020.
[13] S. Hejazi, R. Kazemi, and S. Ghaemmaghami, "Isolated Persian digit recognition using a hybrid HMM-SVM," in 2008 International Symposium on Intelligent Signal Processing and Communications Systems, 2009, pp. 1-4.
[14] J. Ashraf, N. Iqbal, N. S. Khattak, and A. M. Zaidi, "Speaker independent Urdu speech recognition using HMM," in 2010 The 7th International Conference on Informatics and Systems (INFOS), 2010, pp. 1-5.
[15] P. V. Janse, S. B. Magre, P. K. Kurzekar, and R. Deshmukh, "A comparative study between MFCC and DWT feature extraction technique," International Journal of Engineering Research and Technology, vol. 3, pp. 3124-3127, 2014.
[16] K. R. Ghule and R. Deshmukh, "Feature extraction techniques for speech recognition: A review," International Journal of Scientific & Engineering Research, vol. 6, pp. 2229-5518, 2015.
[17] A. Stolcke, E. Shriberg, L. Ferrer, S. Kajarekar, K. Sonmez, and G. Tur, "Speech recognition as feature extraction for speaker recognition," in 2007 IEEE Workshop on Signal Processing Applications for Public Security and Forensics, 2007, pp. 1-5.
[18] N. Desai, K. Dhameliya, and V. Desai, "Feature extraction and classification techniques for speech recognition: A review," International Journal of Emerging Technology and Advanced Engineering, vol. 3, pp. 367-371, 2013.
[19] N. Dave, "Feature extraction methods LPC, PLP and MFCC in speech recognition," International journal for advance research in engineering and technology, vol. 1, pp. 1-4, 2013.
[20] P. P. Singh and P. Rani, "An approach to extract feature using MFCC," IOSR Journal of Engineering, vol. 4, pp. 21-25, 2014.

Edited by: S.B.Goyal
Special issue on: Soft Computing and Artificial Intelligence for wire/wireless Human-Machine Interface
Received: Sep 26, 2023
Accepted: Jan 8, 2024