International Journal of Scientific Research in Science and Technology
Print ISSN: 2395-6011 | Online ISSN: 2395-602X (www.ijsrst.com)
doi : https://doi.org/10.32628/IJSRST207543

# Efficient Dual-tone Multi-frequency Signal Detection using a KNN Classifier

**Arunit Maity, Sarthak Bhargava, Prakasam P***
School of Electronics Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India
*prakasamp@gmail.com

**Article Info**
Volume 7, Issue 5
Page Number: 208-224
Publication Issue :
September-October-2020

**Article History**
Accepted : 01 Oct 2020
Published : 10 Oct 2020

## ABSTRACT

The requirement for an efficient method for noise-robust detection of Dual-tone Multi-frequency (DTMF) signals keeping in mind the continuous evolution of telecommunication equipment is conspicuous. A machine learning based approach has been proposed in this research article to detect DTMF tones under the influence of various noises and frequency variations by employing the K-Nearest Neighbor (KNN) Algorithm. In order to meet accurate classification/detection requirements for various real-world requirements, a total of four KNN models have been created and compared, and the best one proposed for real-time deployment. Two datasets have been amassed, a clean dataset without noise and a noisy augmented dataset with perturbations that are observed in telecommunication channels such as additive white gaussian noise (AWGN), amplitude attenuation, time shift/stretch etc. Mel-Frequency Cepstral Coefficients (MFCC) and Goertzel's Algorithm (used to estimate the absolute Discrete Fourier Transform (DFT) values for the fundamental DTMF frequencies) are employed to calculate features to be fed to the KNN models. The four models differ in being trained with and without the augmented data and using either of the two aforementioned feature extraction algorithms. The proposed models have been verified and validated with unseen noisy testing data and it was found that the proposed KNN model D, which was trained on the augmented dataset and uses the Goertzel's algorithm to extract the absolute DFT values as features, outperformed all the other models with a macro recall, macro precision and macro F1 classification score of 97.7, 97.70625 and 97.70046 respectively. The proposed model is also computationally inexpensive and showcases relatively low time complexity with an average detection time of 20ms.

**Keywords :** Dual-tone multifrequency, K-nearest neighbors, Mel-Frequency Cepstral Coefficients, Goertzel's algorithm, Machine Learning

## I. INTRODUCTION

While transmission receivers have gotten exceedingly better with time, there is still a degree of unreliability associated with analog detectors. This is due to various factors such as the circulation of outdated transmitters and the infusion of noise and frequency variations during transmission through the telecommunication channel. In olden days single tone frequency line pulsing were used in telephone equipment [1]. Due to technological advancements in the recent past, wireless networks have moved towards digitization wherein all signals are considered equally. Hence, the requirement for digital DTMF detection is recommended to reduce the extra investment required to deploy analog detection. It is therefore more financially prudent to replace analog receivers with their digital equivalents which are far less susceptible to noise as well as more reliable and cost effective. Even though the best classifiers of DTMF signals have performed with a high accuracy (95%), a more robust hybrid method is required whose performance remains excellent even in non-ideal conditions. The procedures involved in designing such a model has been outlined in this research article.

This paper aims to implement a machine learning based DTMF detection tool for identifying DTMF signals that have been subjected to frequency variation as well as additive white gaussian noise (AWGN). The proposed DTMF detector uses the KNN algorithm for the effective detection of DTMF signals. The proposed KNN models have been modelled using Goertzel's algorithm to compute the absolute DFT values of the fundamental DTMF frequencies as well as the MFCC algorithm. The proposed models have been trained with both an augmented and non-augmented data set. The recall, precision and F1 classification scores and confusion matrices have been computed to measure the efficacy of the model. The entire simulation, computation and analysis has been carried out using MATLAB R2019.

## II. LITERATURE REVIEW

A short review of specifications for DTMF and R2 signalization, suggested by ITU-T has been analyzed and followed by a short review of decoding techniques which are applied in practice [2]. Finally, a new computationally efficient method has been proposed and experimentally verified where the DTMF detection can be carried out using FFT methods which consume more power and involve more hardware [3][4]. They have developed a DTMF discovery model involving lesser area and power in FPGA using the Split Goertzel process. It was concluded that the suggested resource allocation process consumes lower power and may still detect DTMF signals efficiently.

A new method for performance evaluation for DTMF receivers employing Quick Fourier transform (QFT) has been proposed in [5]. The symmetric properties of the QFT method leads to better performance in terms of lower memory occupation, good real-time deployment as compared with DFT, FFT and Goertzel method with few floating-point process [6]. Cheng-Yu Yeh, Shaw-Hwa Hwang [7] investigated a multi-frequency detecting (MFD) method instead of traditional single point method. It is a suitable way to reduce computational load even further for DTMF detection. Dabbabi Karim et al. [8] presented a new method to optimize the audio classification and segmentation utilizing GASOM algorithm for multimedia data set. The audio coding using Empirical Mode Decomposition has been presented [9]. In this method, the audio signal has been broken into intrinsic oscillatory components and encoded.

The advancements made in machine learning algorithms and the technology to implement these algorithms is a tool that is being implemented everywhere [15][16][17]. Recognition of DTMF tones is yet another avenue where ML can be employed. There is a need to make the recognition system immune to the presence of unwanted elements like speech, noise and frequency variation. These reasons have led to the employment of artificial intelligence (AI) for the reception of DTMF signals in the most efficient way possible. Nagi et al., [18] proposed the AI based method to detect DTMF which is contaminated by White Gaussian Noise (WGN) using Support Vector Machines (SVM). Pao et al., [19] assigned three weighting functions and compared the performance of weighted KNN, weighted D-KNN and conventional KNN to identify ten digits in Mandarin Database. To retrieve the content-based audio, Genetic algorithm with KNN based approach has been suggested [20]. The fed audio files at the input of the system have been ordered by their similarity at the output of the system using the different features. Ali et al., [21] investigated the performance of KNN classifier to classify the heterogeneous data by measuring the resemblance between the distance for binary and numerical data.

Daponte et al., [22] depicted a new method to decode DTMF tones efficiently using an artificial neural net (ANN) and implemented the same on a DSP processor. After a suitable training phase, the ANN can learn to detect an output pertaining to a DTMF signal. Salamon et. al [23] has proposed a CNN based architecture to classify environmental sounds. Also, they have demonstrated the merits of data augmentation to avoid data scarcity issue and investigated the impact of various augmentations on the execution of the suggested architecture.

## III. MATERIALS AND METHODS

### *A. DTMF Tones*

DTMF tones are created by adding two sinusoidal signals from the predefined eight fixed frequencies set. The predefined set contains both low and high frequency groups which are mutually exclusive. The four low and high frequency tones are for the respective rows and columns of the table. When a DTMF tone is generated for an element from the table, the frequencies corresponding to the row-column intercept are generated and summed. The formula for the generation of a pure DTMF signal is given by:

$$x(t) = A_m cos(2\pi f_L T + \theta) + A_m cos(2\pi f_H T + \theta) \quad (1)$$

Here, $A_m$ is the amplitude for each DTMF waveform. The higher and lower frequencies are given by $f_H$ and $f_L$ respectively. T is the sample rate. The keypad formed by this combination of frequencies is given in Table 1.

Table 1: DTMF Signals – Touch keypad

| **Frequency (Hz)** | **High frequency group (f > 1kHz)** | | | |
|---|---|---|---|---|
| **Low frequency group (f < 1kHz)** | 1209 | 1336 | 1477 | 1633 |
| 697 | 1 | 2 | 3 | A |
| 770 | 4 | 5 | 6 | B |
| 852 | 7 | 8 | 9 | C |
| 941 | * | 0 | # | D |

The Bell System Inc, US developed the standards for DTMF signals. The standards have been specified in the ITU-T Recommendation Q.23 [24] and have been tabulated in Table 2.

Table 2: ITU Standards for DTMF Signals [24]

| Parameters | AT&T |
|---|---|
| Low frequency signals | 697, 770, 852, 941 Hz |
| High frequency signals | 1209, 1336, 1477, 1633 Hz |
| Frequency Tolerance (Operation) | ≤ 1.5% |
| Frequency Tolerance (Non-operation) | ≥ 3.5% |
| Power Levels (Operation) | 0 to -25 dBm |
| Power Levels (Non-operation) | ≤ -55 dBm |
| Power Level Difference | +4 dB to -8 dB |
| Signal Duration (Operation) | ≥ 40 ms |
| Signal Duration (Operation) | ≤ 23 ms |
| Pause duration | ≥ 40 ms |
| Signal interruption | ≤ 10 ms |
| Signalling velocity | ≥ 93 ms/digit |

These standards ensure the proper generation of DTMF signals based on certain essential technical configurations. This ensures uniformity and proper decoding of the signals.

*B. Mel-Frequency Cepstral Coefficients (MFCC)*

A widely used method to obtain spectral characteristics is to calculate the MFCCs which are purely used for audio and speech identification using the Mel scale. Since MFCCs are based on spectral domain characteristics, they are much more precise than spatial domain characteristics. MFCC is a real cepstral which is computed by windowing the short-time signal obtained through FFT of the given signal [10]. Additionally, these coefficients are very robust to variations. MFCC represents an audio feature abstraction method which obtains metrics from an audio signal and enhances the desired features while de-emphasizing the rest. The signal is broken up as several frames comprising of a certain count of samples. In many systems there exists an overlap to smoothen the transition between samples. In order to eliminate gaps at the boundaries, every frame is windowed using a Hamming window. The frequency components of the windowed signal are extracted using the Mel-scale filter bank based FFT computation. The Mel-scaled frequency can be expressed as:

$$Frequency(scaled) = 2595log\left(1 + \frac{f}{700}\right) \quad (2)$$

Here, f is the frequency of audio/speech signal. MFCCs employ the Mel-scale filter bank in which high frequency filters occupy more bandwidth than low frequency filters while time-based resolutions remain the same. The final process is to compute Discrete Cosine Transformation (DCT) for the filter bank output. The $0^{th}$ coefficient is deemed redundant owing to its undependability.

*C. Computation of DFT Coefficients using Goertzel's Algorithm*

In the Goertzel's algorithm, the DFT Coefficients are computed using a second order recursive digital resonance system [6] and it is shown in Fig 1. In place of solving for all N-point DFT values, this algorithm obtains the DTMF using a bank of eight filters. The index value $k$ for the DFT is defined as $k=N*f/f_s$, where $f$, $N$ and $f_s$ are frequency of DTMF signal, length of the block and sampling frequency, respectively.

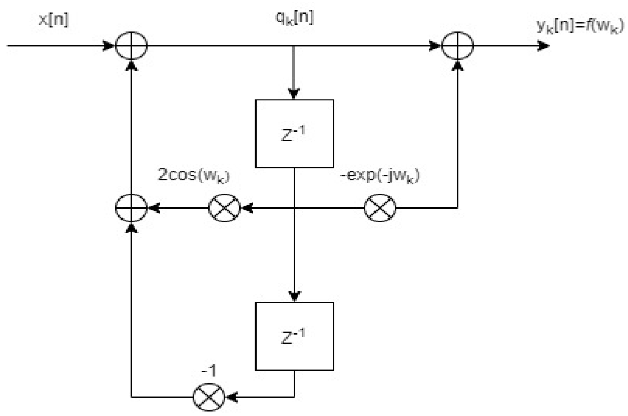


Fig 1: Second order digital resonance system - Goertzel's Algorithm

The process of this algorithm is depicted using the following equations:

$$q_k(n) = x[n] + 2cos(w_k)q_k[n-1] - q_k[n-2] \quad (3)$$

$$y_k(n) = q_k[n] - q_k[n-1]e^{-jw_k} \quad (4)$$

Here, n, x[n] and $w_k$ denotes the number of samples in input, input signal and $k^{th}$ DFT coefficient value. The non-recursive and recursive process of this algorithm are represented in eqns. (4) and (3) respectively where the recursive process is executed for each input sample whereas the non-recursive action is performed at 1/N times of the sampling rate.

*D. KNN Classifier*

In recent days, the KNN classification model is mostly used to classify, identify or recognize patterns amongst other applications. It can compete with the most accurate models due to its highly accurate predictions. It can be used for applications where a high accuracy is required but not a human-readable model. It is a non-parametric method which employs classification and regression [21].

In the KNN algorithm, the real-time test data set is classified by computing the similarity with the stored training set and this similarity sample is called the nearest neighbor. It works by considering the closest k neighbors and utilizing the vote for majority policy for the exact class. The nearest neighbors can be computed by assessing the distance between the test data set with every pattern in the training data set. The distance can be found by various distance measures such as Euclidean, City Block, Cosine, Correlation, Minkowski, Chebyshev, etc. The formulae for the distance measures being employed in this paper have been tabulated in table 3.

Table 3: Distance Measure Methods for KNN

| **Distance measure** | **Formula for distance measure** |
|---|---|
| Minkowski | $\left(\sum_{i=1}^{k}(\lvert x_i - y_i\rvert)^q\right)^{\frac{1}{q}}$ |
| Cosine | $\frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2 \sqrt{\sum_i y_i^2}}}$ |
| City Block | $d_{ij} = min(\lvert i-x\rvert + \lvert j-y\rvert), 0 \le i,j \le n-1$ |
| Correlation | $\frac{N\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{N\sum x_i^2 - (\sum x_i)^2 \sqrt{N\sum y_i^2 - (\sum y_i)^2}}}$ |

The algorithm for the proposed KNN classifier is represented in Algorithm 1.

**Algorithm 1 - KNN Classifier for Audio Signal**

**Begin**

*Load* DTMF Audio data set

**for j = 1 to N** (N - number of audio files in dataset)

*Initialize:* *K* - The number of nearest neighbors that want to take the vote of

***Calculate the distance*** $d_{ij}$ – Distance between the test data (i) and each row of the training data (j) using the formula given in Table 3

*Arrange* $d_{ij}$ in an *increasing order*

*Obtain* *max{* $d_{ij}$ *}*

*Identify* *K row* pertained to *max{*$d_{ij}$*}*

*Find* the *frequent* occurring class

*Predict* based on *highest frequency*

**End**

**End**

The finest value of K is the one which leads to the lowest test error rate. Therefore, the test error for several values of K has been calculated. In a way the test set is used as a training set. A better approach is to hold out a subset of the training data. This is termed as the validation set. This will help in selecting the appropriate level of flexibility of the algorithm.

The 5-fold stratified cross validation has been used in this research work as the validation approach. It involves dividing the training set in terms of 5 number of groups of roughly the same size. The first fold is treated as a validation data set and remaining are employed as the training data set. The error rate is calculated 5 times on the held-out fold. For each iteration, a distinct group is maintained as validation data set. During this iterative process, test error obtained 5 times are averaged for final validation.

## IV. METHODOLOGY

The process flow diagram of the proposed machine learning based KNN classifier for audio signal is illustrated in Fig 2. The operation carried out in each block is explained in the following subsections.

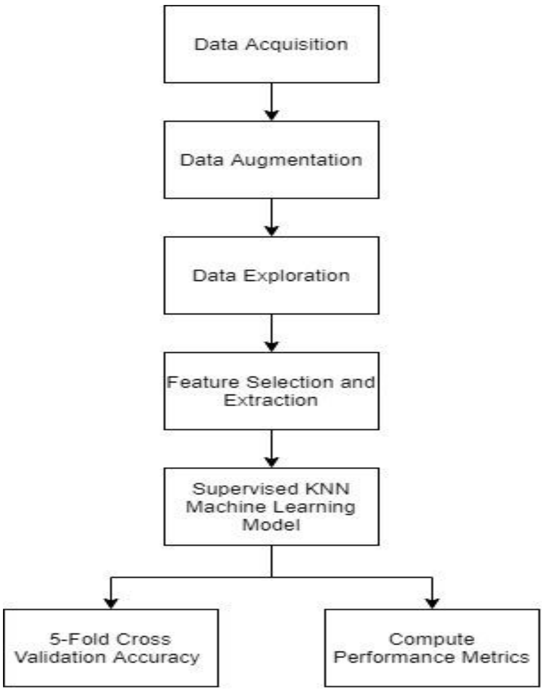


Fig 2: Process flow diagram of the proposed methodology

### A. *Data Acquisition*

The final dataset used for training, validation and testing of the KNN classifier model consists of 20,320 DTMF audio files with a tone duration of 100ms and a sampling frequency of 8000 Hz since those are the standard audio specifications of DTMF tones used for modelling generators and decoders. The initial dataset consisted of 2032 DTMF signal audio files which were downloaded from the online Dual Frequency Audio Tone Generator tool offered by the website "audiocheck.net" by creating multi-frequency tones whose low and high frequency components were chosen as specified by the ITU-T Q.23 recommendation [17].

To eliminate the process of filling out the form at audiocheck.net and downloading each of the 2032 audio files, a python-based Selenium (an open source web-browser automation tool) script was written and deployed. These audio files were then respectively annotated to create a total of 16 different categorical classes, one for each keypad character/DTMF tone. Since there exist a total of 16 DTMF tones in the ITU-T Q.23 recommendation, each tone had 127 audio files. The details of audio files are represented in table 4.

These 127 audio files for each tone were generated by varying the individual power levels of the high and low frequency components in accordance with the values of multi-frequency push-button receiving parameters adopted by AT&T as seen in table A-1/Q.24 of the ITU-T Q.24 recommendation as shown in Table 5.

A digital domain estimate conversion of dBm to dBFS was done using the equation (5) given below:

$$P_{out(dBm)} = P_{out(dBFS)} + 10 \qquad (5)$$

After conversion from dBm to dBFS, it is understood that for a given pair of tones, the maximum permissible signal strength should not exceed -9 dBFS. The minimum signal strength for low and high frequency clusters should be -20.5 dBFS and -18.5 dBFS respectively. Apart from the tone's corresponding low and high frequency values, the energy from the remaining frequency bands should be either inexistent or lesser than -45 dBFS. An ideal detector will decode the DTMF signal properly when the low frequency (row) signal is 8 dB more than the high frequency (column) signal or the high frequency (column) signal is 4 dB more than the low frequency (row) signal.

Table 4: Details of Original Dataset

| Low Frequency Component | High Frequency Component | Key | No. of audio files |
|---|---|---|---|
| 697 | 1209 | 1 | 127 |
| 697 | 1336 | 2 | 127 |
| 697 | 1477 | 3 | 127 |
| 697 | 1633 | A | 127 |
| 770 | 1209 | 4 | 127 |
| 770 | 1336 | 5 | 127 |
| 770 | 1477 | 6 | 127 |
| 770 | 1633 | B | 127 |
| 852 | 1209 | 7 | 127 |
| 852 | 1336 | 8 | 127 |
| 852 | 1477 | 9 | 127 |
| 852 | 1633 | C | 127 |
| 941 | 1209 | * | 127 |
| 941 | 1336 | 0 | 127 |
| 941 | 1477 | # | 127 |
| 941 | 1633 | D | 127 |
| | **Total Number of audio files** | | 2032 |

Table 5: Specifications of Audio Files in Dataset

| Parameters | Value |
|---|---|
| Tone Duration | 100 ms |
| Sampling Frequency | 8000 KHz |
| Number of Samples | 800 |
| Bitrate | 128 kbps |
| Number of Files | 2032 |
| Number of Files after augmentation | 20320 |

Table 6: Details of Augmented Dataset

| Low Frequency Component | High Frequency Component | Key | No. of audio files |
|---|---|---|---|
| 697 | 1209 | 1 | 1270 |
| 697 | 1336 | 2 | 1270 |
| 697 | 1477 | 3 | 1270 |
| 697 | 1633 | A | 1270 |
| 770 | 1209 | 4 | 1270 |
| 770 | 1336 | 5 | 1270 |
| 770 | 1477 | 6 | 1270 |
| 770 | 1633 | B | 1270 |
| 852 | 1209 | 7 | 1270 |
| 852 | 1336 | 8 | 1270 |
| 852 | 1477 | 9 | 1270 |
| 852 | 1633 | C | 1270 |
| 941 | 1209 | * | 1270 |
| 941 | 1336 | 0 | 1270 |
| 941 | 1477 | # | 1270 |
| 941 | 1633 | D | 1270 |
| | **Total Number of audio files** | | 20320 |

### *B. Data Augmentation*

Currently, the sheer lack of data is one of the most prevalent issues in data science problems. Data augmentation assists in the generation of synthetic data from existing data sets such that generalization capability of the classifier model can be enhanced. The proposed method for detection of DTMF tones is designed to be robust to noise and other possible errors corrupting the tones during the transmission of these DTMF tones over a telecommunication network channel. In order to incorporate this, the dataset comprising 2032 audio files was augmented into a total of 20,320 audio files. The augmentation is done such that each audio file present in the initial dataset is made into 10 corrupted audio files with different random errors. The details of audio files using augmentation is tabulated in table 6. Audio file corruption was done by addition of Additive White Gaussian Noise (AWGN), time-stretch, time-shift and volume control.

*1) Background Noise*

This perturbation adds noise to the input DTMF audio signal at a random signal to noise ratio (SNR). Additive White-Gaussian Noise is added to each audio file with an 95% probability. The noise addition SNR in dB is randomly chosen within the range of 5 to 10 dB.

*2) Time-Stretching*

This augmentation process changes the time duration of the input audio signal. The underlying equation is as follows:

$$x_c[n] = x[an] \qquad (6)$$

Here, the audio signal is stretched or compressed without changing the pitch of the audio. Time-Stretching is applied with an 95% probability to all audio files and the speed-up factor, 'a' is chosen at random within a range of 0.8 to 1.2.

*3) Time Shifting*

The concept of shifting time is very elementary. It simply shifts the audio to the left or right by a random amount of time. Here, a circular shift is applied on the time-domain audio data to simulate transmission error that occurs during audio data transmission. The underlying equation is as follows:

$$x_c[n] = \begin{cases} x[n-n_0] & n_0 \le n \le N-1 \\ x[N-n_0+n] & 0 \le n \le n_0 \end{cases} \qquad (7)$$

*4) Simple Gain/Volume Control*

This augmentation process increases or decreases the loudness of an input audio signal by a specified gain (in dB). It is implemented by reading the audio file and carrying out simple array manipulations. Volume gain corruption is implemented at a 95% probability with a volume gain chosen randomly within a range of -3 to +3 dB.

*C. Data Exploration*

In any automated audio recognition/classification system, an important step is to extract features which can be used to train the classifier model. These features must be unique and should be useful to identify and differentiate the spectral content in the audio file while ignoring redundant information such as background noise etc. This is done in the form of visual analysis. In order to find the most appropriate feature to be extracted from the audio files for training the KNN classifier, two different approaches were used during preliminary analysis. In the first approach, the MFCCs have been calculated for all audio files and the Mel-spectrogram has been plotted. Few sample Mel-spectrum plots have been shown in Fig 3. From the Mel-spectrum plot, comparisons have been drawn between audio files of different classes i.e. the multi-tone corresponding to a certain telephone keypad button. The second approach involved the extraction of the absolute value of the frequency components of the 8 frequencies, namely the 4 higher frequencies and the 4 lower frequencies used in

DTMF tones. Therefore, a total of 8 computations are required. Goertzel's algorithm has been employed to compute the magnitude of these 8-individual DFT coefficients. The frequency spectrogram plot pertaining to only these 8 frequencies have been plotted for all audio files as shown in Fig 4 and compared with each other to identify stark differences.

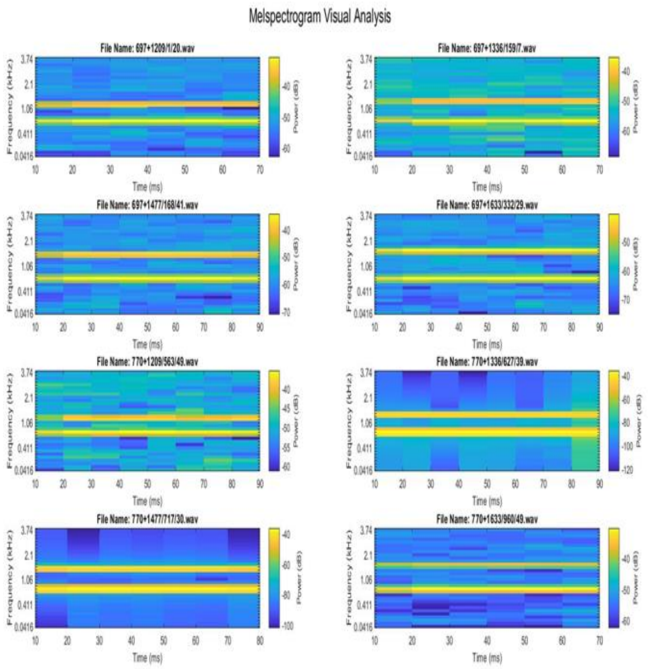


Fig 3: Mel-spectrogram Visual Analysis

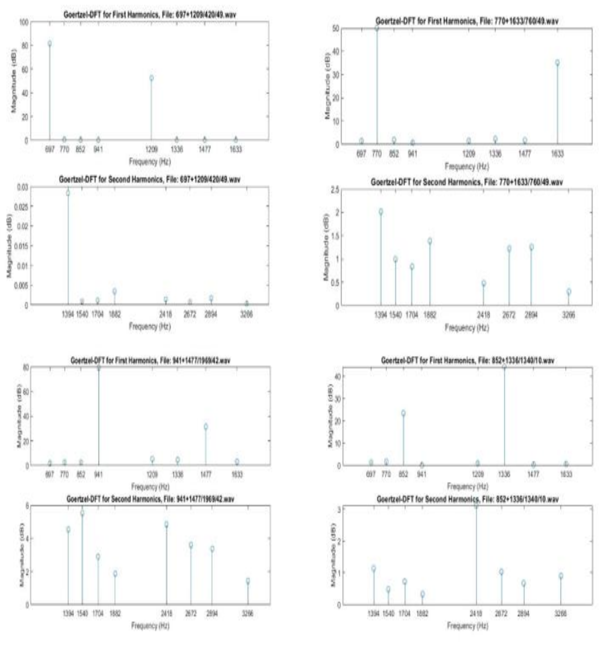

Fig 4: Goertzel's coefficients

*D. Feature Selection*

After visual analysis and data exploration, it has been inferred that both MFCCs and Goertzel Algorithm's coefficients computation can serve as viable features for training the model to produce correct/accurate predictions. Thus, these two different approaches were employed towards selection and extraction of features from the dataset.

*Approach 1:* The first 13 MFCCs along with the overall pitch of the audio signal were calculated from the dataset and used for training the model. The first 13 MFCCs are used to explain the shape of the spectrum of the audio signal.

*Approach 2:* The selective magnitude response of the audio files was calculated using Goertzel's algorithm for efficient determination of the magnitude of the DFT at only the eight fundamental frequencies associated with DTMF tones.

Therefore, a total of four different models have been trained using the above-mentioned feature selection approaches. Out of the them, two models were trained on the initial clean dataset consisting of 2032 audio files using the above-mentioned two feature extraction approaches, whereas the others were trained on the augmented dataset consisting of 20,320 audio files using the same feature-selection approaches but trained with the intuition that they shall be more robust to noise, time-shifts and other channel discrepancies.

*E. Data Standardization*

In machine learning, data is prepared mostly using the Data Standardization technique. The main objective of standardization is to centre and scale the values of columns with numerical data in the dataset such that the range of values is not disturbed. This normalises the impact of each feature/attribute. After the features were extracted from the audio files dataset, each column of the entire table was centred and scaled by the column mean and standard deviation, respectively. This has been done by subtracting each cell ($x$) of every column with its column-mean ($\mu$) and dividing each column by its column-standard deviation ($\sigma$) as can be seen in the following equation:

$$z = \frac{x - \mu}{\sigma} \tag{8}$$

*F. Architecture of the Proposed Classifier Model*

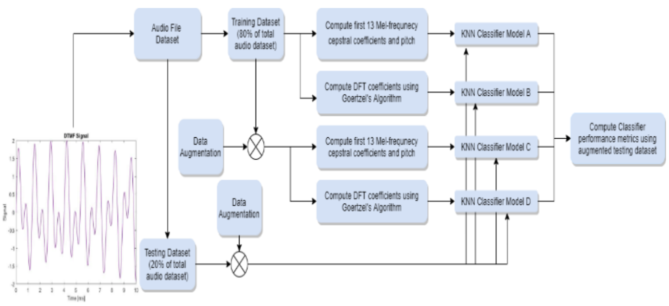


Fig 5: Architectures of the Hybrid KNN Classifier Models

The detailed architecture of the proposed machine learning based KNN classifier is shown in Fig 5. As discussed in section 2, the dataset has been acquired by Web Scraping in accordance with the prevalent technical specifications for DTMF tones. Next, the dataset has been augmented with noise and other channel discrepancies to make the model more robust to noise and other interferences. After augmentation, the data has been explored through visual analysis and viable features selected for training the model with either MFCCs or magnitude of DFT coefficients computed using Goertzel's algorithm. In order for training and testing the KNN classifier model, the dataset has been split randomly into a ratio of 4:1. 80% of the dataset has been used for training the data and the balance 20% has been used to test the model accuracy. Over or under-fitting of the model has been avoided this way. The two approaches discussed previously have been used to create four models whose characteristics are discussed below-

*KNN Model A:*

Feature selection approach 1 has been used and the model has been trained with the dataset that has not been augmented. The Validation and Test Confusion matrices are obtained. The model is tested with the augmented dataset and the confusion matrix is plotted and performance metrics are computed.

*KNN Model B:*

Feature selection approach 2 has been used and the model has been trained with the clean/non-augmented dataset. However, it is tested with the augmented dataset and the confusion matrix/chart for this is obtained and the performance metrics are computed.

*KNN Model C:*

Feature selection approach 1 has been used and the model has been trained with the dataset that has been augmented. The Validation and Test Confusion matrices are obtained. The performance metrics are obtained from the confusion chart plotted for the Test Accuracy using the testing data.

*KNN Model D:*

Feature selection approach 2 has been used and the model has been trained and tested with the augmented dataset. The testing accuracy confusion chart is used to compute the performance metrics.

*1) Hyperparameter Tuning*

The final hyperparameters for the KNN model were obtained using the 'automatic hyperparameter optimization' function in MATLAB. The function works by employing the Bayesian optimization algorithm to solve the minimization problem of the 5-fold cross-validation classification loss/error by optimally varying these hyperparameters. The Bayesian optimization algorithm aims at minimizing a scalar objective function f(x) for x in a bounded domain. In our case, the objective function is the five-fold stratified cross-validation loss/error. The value for $y_i = f(x_i)$ is evaluated for 4 values of $x_i$, taken at random within the variable bounds. In the event of evaluation errors, more random points are taken until there are 4 successful evaluations. The probability distribution of each component is either uniform or log scaled. The acquisition function calculates the expected amount of improvement in the objective function, while ignoring values that cause an increase in the objective. The expected improvement (acquisition function) is defined as:

$$EI(x, Q) = E_Q\, max[\left(0, u_Q(x_{best}) - f(x)\right)] \tag{9}$$

Here, Q represents the posterior distribution over function obtained after updating the Gaussian process model of f(x), $x_{best}$ is the location of the lowest posterior mean and $\mu_Q(x_{best})$ is the lowest value of the posterior mean. To avoid a local objective function minimum, the acquisition function modifies its behaviour when it estimates that it is *overexploiting* an area. If $\sigma_F(x)$ is the standard deviation of the posterior objective function at $x$ and $\sigma$ the posterior standard deviation of the additive noise, then:

$$\sigma^2{}_Q(x) = \sigma^2{}_F(x) + \sigma^2 \tag{10}$$

Let $t_\sigma$ be the value of the exploration ratio (0.5). After each iteration, the acquisition function evaluates whether the next point $x$ satisfies:

$$\sigma_F(x) < t_\sigma \sigma \tag{11}$$

If so, the algorithm deems that $x$ is overexploiting. The exploration ratio therefore controls a trade-off between exploring new points for a better global solution, versus concentrating near points that have been examined already. Optimizing and fine-tuning the hyperparameters for the KNN models ensures that we extract the best performance out of these models. The resulting optimal hyperparameter values for each KNN model have been shown in table 7.

Table 7: KNN Classifier Model Parameter Values

| KNN Classifier Model | Number of neighbours (K) | Distance measure | Distance weight |
|---|---|---|---|
| A | 9 | Minkowski | Inverse |
| B | 31 | Cosine | Equal |
| C | 25 | City Block | Squared inverse |
| D | 8040 | Correlation | Squared inverse |

*2) Evaluation Setup*

The evaluation of different KNN based machine learning models is done with the help of certain statistical measures which can help us to understand how good a multi-class classifier is performing. These statistical measures are called performance metrics. We created a confusion matrix for the best model to visualize its performance. We can clearly see the occasional errors in classification and evaluate the models using the frequency of such errors. The performance metrics used for validating and confirming our research are given in the Table 8. The higher the value of precision and recall, the lesser the number of false positives and negatives and better the model. The F1 score takes both the recall and precision into consideration and assigns the model with a performance evaluation that is a single number. A high F1 score would imply that most tones are correctly classified by the model.

Table 8: Performance Metrics

| Performance Metrics | Formula |
|---|---|
| *Precision* | $\frac{True\ Positive}{True\ Positive + False\ Positive}$ |
| *Recall* | $\frac{True\ Positive}{True\ Positive + False\ Negative}$ |
| *F1 Score* | $\frac{2 * Precision * Recall}{Precision + Recall}$ |

## V. RESULTS AND DISCUSSION

The four different KNN classifier models discussed in the previous sections were created and compared based on various performance metrics. The results were evaluated to choose the best model amongst them. To judge the performance of the models, the following measures were analyzed:

- 5-fold stratified Cross Validation Accuracy – The training dataset was shuffled and split into 5 groups and one-by-one, each group was used as testing data while the other groups were used as training data to fit the KNN model. The model was evaluated on the test set and the evaluation score was calculated. The average of the 5-fold evaluation score was computed and called the 5-fold cross-validation accuracy.
- Training Accuracy – The same training dataset which was used to train the model was given as an input to the model and the predicted responses were then used to plot a confusion matrix/chart.
- Testing Accuracy - The testing dataset which was created when the dataset was split into testing (20%) and training (80%) was given as an input to the model and the predicted responses were then used to plot a confusion matrix/chart. This confusion chart was used for computation of performance metrics as it shows how the model will perform on unseen and new data.

The experiments have been carried out using a desktop with the following specifications:

- CPU: Intel Core i5-3570K @ 3.80 GHz
- RAM: 8GB DDR4 RAM @ 1600 MHz
- GPU: Nvidia GTX 760 with 2GB GDDR4 memory

### *A. KNN Classifier Model A*

The KNN model A has been tested with the augmented dataset. This dataset has been designed to imitate real-world scenarios as closely as possible. The test accuracy confusion matrix chart with augmented data set is shown in Fig 6. Also the individual categorical class recall, precision and F1 classification scores are tabulated in table 9.

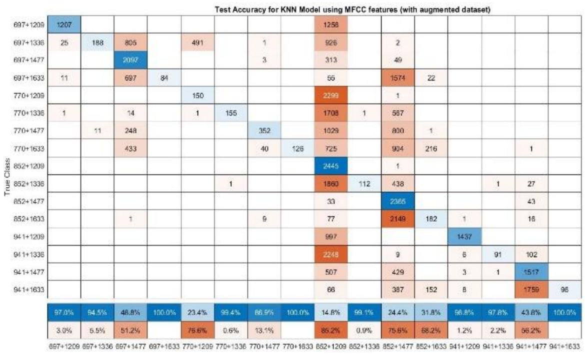


Fig 6: Confusion chart for Test Accuracy for KNN Model A with augmented dataset

Pertaining to the training accuracy, it was observed that for all the cases, the proposed KNN Model A had classified the clean DTMF tones correctly, which implies that the value of both precision and recall for every class is 100%. However, this is not a representation of the real-world data available. Noise and power discrepancies have not been considered yet. Looking at the confusion chart in figure 6 we can see that the performance is poor when it comes to noisy augmented data. Therefore, this model's effectiveness is quite severely limited.

Table 9: Performance Metrics for KNN Model A

| Label/Class | Recall | Precision | F1 Score |
|---|---|---|---|
| 667+1209 | 49 | 97 | 65.10959 |
| 697+1336 | 7.7 | 94.5 | 14.23973 |
| 697+1477 | 85.2 | 48.8 | 62.05612 |
| 697+1633 | 3.4 | 100 | 6.576402 |
| 770+1209 | 6.1 | 23.4 | 9.677288 |
| 771+1336 | 6.3 | 99.4 | 11.84901 |
| 770+1477 | 14.4 | 86.9 | 24.70602 |
| 770+1633 | 5.2 | 100 | 9.885932 |
| 852+1209 | 100 | 14.8 | 25.78397 |
| 852+1336 | 4.6 | 99.1 | 8.7919 |
| 852+1477 | 96.9 | 24.4 | 38.98368 |
| 852+1633 | 7.5 | 31.8 | 12.1374 |
| 941+1209 | 59 | 98.8 | 73.88086 |
| 941+1336 | 3.7 | 97.8 | 7.130246 |
| 941+1477 | 61.7 | 43.8 | 51.23147 |
| 941+1633 | 3.9 | 100 | 7.507218 |

From Table 9, the values for macro-recall, macro-precision and macro-F1 score were found to be 32.1625, 72.5313 and 26.846675, respectively.

*B. KNN Classifier Model B*

Similarly, the KNN model B has been tested with the augmented dataset. The test accuracy confusion matrix chart with augmented data set is shown in Fig 7.

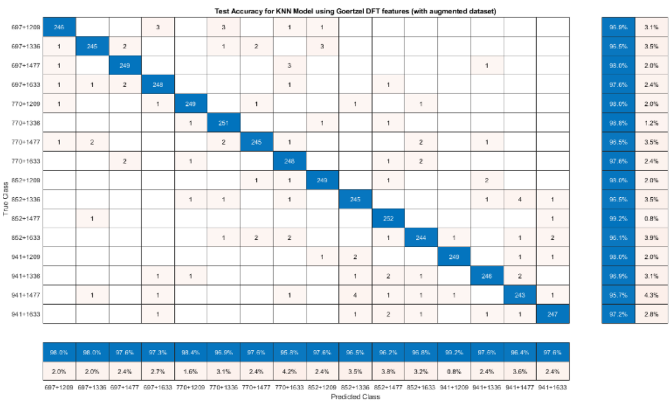


Fig 7: Confusion chart for Test Accuracy for KNN Model B with augmented dataset

Also the computed individual categorical class recall, precision and F1 classification scores are tabulated in table 10. This model can be considered a suitable model for the identification of DTMF tones even when input data is relatively imperfect and contains noise and other channel discrepancies.

Table 10: Performance Metrics for KNN Model B

| Label/Class | Recall | Precision | F1 Score |
|---|---|---|---|
| 667+1209 | 96.9 | 98 | 97.4469 |
| 697+1336 | 96.5 | 98 | 97.24422 |
| 697+1477 | 98 | 97.6 | 97.79959 |
| 697+1633 | 97.6 | 97.3 | 97.44977 |
| 770+1209 | 98 | 98.4 | 98.19959 |
| 771+1336 | 98.8 | 96.9 | 97.84078 |
| 770+1477 | 96.5 | 97.6 | 97.04688 |
| 770+1633 | 97.6 | 95.8 | 96.69162 |
| 852+1209 | 98 | 97.6 | 97.79959 |
| 852+1336 | 96.5 | 96.5 | 96.5 |
| 852+1477 | 99.2 | 96.2 | 97.67697 |
| 852+1633 | 96.1 | 96.8 | 96.44873 |
| 941+1209 | 98 | 99.2 | 98.59635 |
| 941+1336 | 96.9 | 97.6 | 97.24874 |
| 941+1477 | 95.7 | 96.4 | 96.04872 |
| 941+1633 | 97.2 | 97.6 | 97.39959 |

From Table 10, the values for macro-recall, macro-precision and macro-F1 score were found to be 97.34375, 97.34375 and 97.33988 respectively.

*C. KNN Classifier Model C*

The next two models have been configured with data augmentation. The test accuracy confusion matrix of the KNN Model C which incorporates approach 1 for feature selection with augmented data set for training has been computed and depicted in Fig. 8. The computed individual categorical class recall, precision and F1 classification scores are tabulated in table 11.

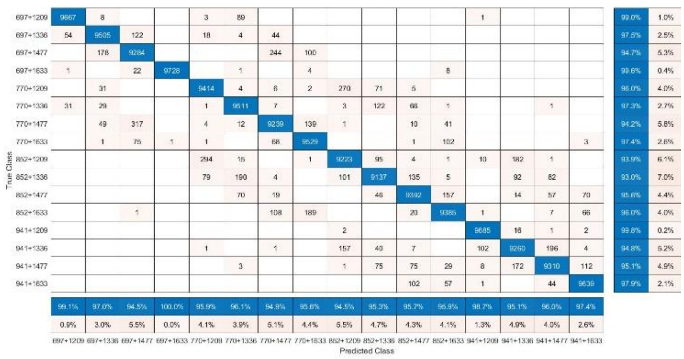

Fig 8: Confusion chart for 5-fold stratified Cross Validation Accuracy for KNN Model C (trained with augmented dataset)

This provides a vast improvement as compared with KNN Model A which had been trained without the augmented data. Thus, it can be concluded that it is essential to train the KNN classifier model using an augmented dataset to achieve a better accuracy in terms of predicted results.

Table 11: Performance Metrics for KNN Model C

| Frequencies | Recall | Precision | F1 Score |
|---|---|---|---|
| 667+1209 | 99 | 99.1 | 99.04997 |
| 697+1336 | 97.5 | 97 | 97.24936 |
| 697+1477 | 94.7 | 94.5 | 94.59989 |
| 697+1633 | 99.6 | 100 | 99.7996 |
| 770+1209 | 96 | 95.9 | 95.94997 |
| 771+1336 | 97.3 | 96.1 | 96.69628 |
| 770+1477 | 94.2 | 94.9 | 94.5487 |
| 770+1633 | 97.4 | 95.6 | 96.49161 |
| 852+1209 | 93.9 | 94.5 | 94.19904 |
| 852+1336 | 93 | 95.3 | 94.13595 |
| 852+1477 | 95.6 | 95.7 | 95.64997 |
| 852+1633 | 96 | 95.9 | 95.94997 |
| 941+1209 | 99.8 | 96.7 | 98.22555 |
| 941+1336 | 94.8 | 95.1 | 94.94976 |
| 941+1477 | 95.1 | 96 | 95.54788 |
| 941+1633 | 97.9 | 97.4 | 97.64936 |

From table 11, the values for macro-recall, macro-precision and macro-F1 score were found to be 96.3625, 96.23125 and 96.29331 respectively.

### *D. KNN Classifier Model D*

The proposed KNN Model D was trained with the augmented dataset and employes the second approach of feature extraction. The trained model was validated using the testing data. The confusion chart for the test accuracy of the KNN model D was generated and has been depicted in Fig. 9. The computed individual categorical class recall values, precision score and F1 score have been tabulated in table 12.

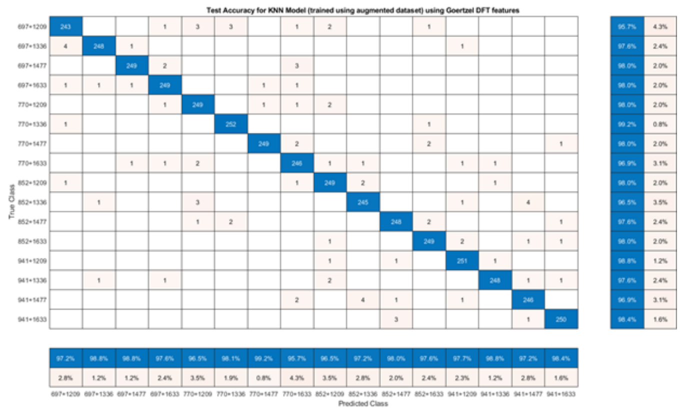

Fig 9 : Confusion chart for Test Accuracy for KNN Model D

Table 12: Performance Metrics for KNN Model D

| Frequencies | Recall | Precision | F1 Score |
|---|---|---|---|
| 667+1209 | 95.7 | 97.2 | 96.44417 |
| 697+1336 | 97.6 | 98.8 | 98.19633 |
| 697+1477 | 98 | 98.8 | 98.39837 |
| 697+1633 | 98 | 97.6 | 97.79959 |
| 770+1209 | 98 | 96.5 | 97.24422 |
| 771+1336 | 99.2 | 98.1 | 98.64693 |
| 770+1477 | 98 | 99.2 | 98.59635 |
| 770+1633 | 96.9 | 95.7 | 96.29626 |
| 852+1209 | 98 | 96.5 | 97.24422 |
| 852+1336 | 96.5 | 97.2 | 96.84874 |
| 852+1477 | 97.6 | 98 | 97.79959 |
| 852+1633 | 98 | 97.6 | 97.79959 |
| 941+1209 | 98.8 | 97.7 | 98.24692 |
| 941+1336 | 97.6 | 98.8 | 98.19633 |
| 941+1477 | 96.9 | 97.2 | 97.04977 |
| 941+1633 | 98.4 | 98.4 | 98.4 |

From Table 12, the value for macro-recall, macro-precision and macro-F1 score has been obtained as 97.7, 97.70625 and 97.70046, respectively. The scores are marginally higher than all the KNN models, signifying best performance.

### *E. Performance Comparison for the KNN Models*

The performance metrics obtained for each of the four proposed KNN models have been tabulated in Table 13. From the reported literature, it can be understood that the KNN Classifier D with the highest macro-F1 classification score is the most optimum model to decode and detect DTMF signals.

Table 13: Performance Comparison

| Proposed KNN Classifier | Metrics values with augmented data set | | |
|---|---|---|---|
| | Macro-Recall | Macro-Precision | Macro-F1 Score |
| KNN Model A | 32.1625 | 72.5312 | 26.846675 |
| KNN Model B | 97.3475 | 97.3475 | 97.33988 |
| KNN Model C | 96.3625 | 96.23125 | 96.29331 |
| KNN Model D | 97.7 | 97.70625 | 97.70046 |

## VI. CONCLUSION

A machine learning based approach for the detection of noisy/corrupted DTMF tones has been presented in this paper. Four KNN models were designed out of which the first two models had been trained without augmented data set and the remaining two had been trained with the augmented data set. Similarly, two approaches have been proposed for feature selection and extraction which utilized either MFCCs or DFT coefficients using Goertzel's algorithm. All the proposed KNN models were validated with the help of the confusion matrix, by computing the individual categorical class recall, precision and F1 classification scores. From the experimental analysis it has been concluded that the proposed KNN Model D, which was trained using the augmented dataset and employs the Goertzel's algorithm to estimate the selective magnitude response of the DFT coefficients at the eight fundamental DTMF frequencies, attained the highest F1 score of 97.70046 which is appropriate for real-time DTMF detection. The average computing time for detection of a single DTMF tone using the KNN model D is approximately 20ms. In accordance with the internal features of machine learning models, the obtained solution allows the possibility of obtaining a significantly faster decoder which is much less affected by noise and sound interference compared to the traditional approaches. Also, the experimental results have shown that noise has a negligible effect on the model's accuracy.

## VII. REFERENCES


[1] Trittler, S. Hamprecht, FA. Near optimum sampling design and an efficient algorithm for single tone frequency estimation, Digital Signal Processing, 19(4): 628-639, (2009).

[2] Popovic, Miodrag. Efficient decoding of digital DTMF and R2 tone signalization. Facta universitatis - series: Electronics and Energetics. 16. 2013, doi:10.2298/FUEE0303389P.

[3] S. N. Bhavanam, P. Siddaiah and P. R. Reddy, "FPGA based efficient DTMF detection using Split Goertzel algorithm with optimized resource sharing approach," 2014 Eleventh International Conference on Wireless and Optical Communications Networks (WOCN), Vijayawada, 1-8, 2014..

[4] Bhavanam. Goertzel Algorithm based DTMF Detection. American International Journal of Research in Science, Technology, Engineering & Mathematics. 1. 6-12, 2014.

[5] Park, Min & Lee, Sang & Yoon, Dal. Signal detection and analysis of DTMF receiver with

quick Fourier transform. Proceedings of 30th Annual Conference of IEEE Industrial Electronics Society, 3, 2058 – 2064, 2004.

[6] T. Joseph, K. Tyagi and D. R. Kumbhare, "Quantitative Analysis of DTMF Tone Detection using DFT, FFT and Goertzel Algorithm," Proceedings of 2019 Global Conference for Advancement in Technology, 1-4, 2019.

[7] Yeh, Cheng-Yu & Hwang, Shaw-Hwa. Efficient Detection Approach for DTMF Signal Detection. Applied Sciences. 9(3). 2019, doi:10.3390/app9030422.

[8] Dabbabi Karim, Cherif Adnen and Hajji Salah. An Optimisation of Audio Classification and segmentation using GASOM Algorithm. International Journal of Advanced Computer Science and Applications, 09(04), 143-157, 2018.

[9] Boudraa, AO. Khaldi, K. Chonavel, T. Hadj-Alouane, MT. Komaty, A. Audio coding via EMD, Digital Signal Processing, 104, 2020, DOI: 10.1016/j.dsp.2020.102770.

[10] Dave, Namrata. Feature Extraction Methods LPC, PLP and MFCC In Speech Recognition. 2013, Corpus id : 212563590.

[11] Thiruvengadam. Speech/Music Classification using MFCC and KNN. International Journal of Computational Intelligence Research, 13(10), 2449-2452, 2017.

[12] Q. Li et al., "MSP-MFCC: Energy-Efficient MFCC Feature Extraction Method With Mixed-Signal Processing Architecture for Wearable Speech Recognition Applications," in IEEE Access, vol. 8, pp. 48720-48730, 2020.

[13] Zhao, X., Zhang, S. & Lei, B. Robust emotion recognition in noisy speech via sparse representation. Neural Comput & Applic 24, 1539–1553 (2014).

[14] Demircan, S., Kahramanli, H. Application of fuzzy C-means clustering algorithm to spectral features for emotion classification from speech. Neural Computing and Applications, 29, 59–66 (2018).

[15] Yang, XK. Qu, D. Zhang, WL. Zhang, WQ. An adapted data selection for deep learning-based audio segmentation in multi-genre broadcast channel, Digital Signal Processing, 81, 08-15, (2018).

[16] M. Esmaeilpour, P. Cardinal and A. Lameiras Koerich. A Robust Approach for Securing Audio Classification Against Adversarial Attacks. IEEE Transactions on Information Forensics and Security, 15, 2147-2159, 2020.

[17] Rahmani, MH. Almasganj, F. Seyyedsalehi, SA. Audio-visual feature fusion via deep neural networks for automatic speech recognition, Digital Signal Processing, 82, 54-63, (2018).

[18] J. Nagi, S. K. Tiong, K. S. Yap and S. K. Ahmed, "Dual-tone Multifrequency Signal Detection using Support Vector Machines," Proceedings of 6th National Conference on Telecommunication Technologies and 2nd Malaysia Conference on Photonics, Putrajaya, 350-355, 2008.

[19] T. Pao, W. Liao and Y. Chen, "Audio-Visual Speech Recognition with Weighted KNN-based Classification in Mandarin Database," Proceedings of 3rd International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 39-42. 2007.

[20] Hussein Toman, S., Ghazi Abdul Sahib, M., & Hussein Toman. Content-Based Audio Retrieval by using Elitism GA-KNN Approach. Journal of Al-Qadisiyah for Computer Science and Mathematics, 9(1), 153-168, 2017.

[21] Ali, N., Neagu, D. & Trundle, P. Evaluation of k-nearest neighbour classifier performance for heterogeneous data sets. SN Applied Sciences. 1, 1559 (2019). https://doi.org/10.1007/s42452-019-1356-9

[22] P. Daponte, D. Grimaldi and L. Michaeli. Neural network and DSP based decoder for DTMF signals. Proceedings of IEEE international Conference on Science, Measurement and Technology, 147(1), 34-40, 2000.

[23] J. Salamon and J. P. Bello. Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification. IEEE Signal Processing Letters, 24(3), 279-283, 2017.

[24] International Telecommunication Union, ITU-T Recommendation Q.23, Technical features of push-button telephone sets, Fascicle VI.1, Blue Book, 1993.

**Cite this article as :**

Arunit Maity, Sarthak Bhargava, Prakasam P, "Efficient Dual-tone Multi-frequency Signal Detection using a KNN Classifier", International Journal of Scientific Research in Science and Technology (IJSRST), Online ISSN : 2395-602X, Print ISSN : 2395-6011, Volume 7 Issue 5, pp. 208-224, September-October 2020. Available at doi : https://doi.org/10.32628/IJSRST207543

Journal URL : http://ijsrst.com/IJSRST207543