World Journal of Advanced Engineering Technology and Sciences

eISSN: 2582-8266

Cross Ref DOI: 10.30574/wjaets

Journal homepage: https://wjaets.com/

(REVIEW ARTICLE)

Check for updates

Convolution neural networks with hybrid feature extraction methods for classification of voice sound signals

Pratibha Rashmi * and Manu Pratap Singh

Department of Computer Science, Dr. Bhimrao Ambedkar University, Khandari Campus, Agra, India.

World Journal of Advanced Engineering Technology and Sciences, 2023, 08(02), 110–125

Publication history: Received on 07 February 2023; revised on 16 March 2023; accepted on 19 March 2023

Article DOI: https://doi.org/10.30574/wjaets.2023.8.2.0083

Abstract

The convolutional neural networks (CNNs) lead in the domain of Sound Recognition due to its flexibility and ability with different adjusting parameters. The recognition of spoken English Alphabets by different people with deep learning techniques attracted the research community. In this paper, we are exploring the use of convolutional neural network (CNN), a deep learner that can automatically learn features directly from the dataset while training for the classification of sounds signals of English alphabets. In this proposed work, we consider two CNN architectures. In first architecture, we propose MFCC based features for pretrained two convolutional layer CNN architecture. In the second architecture, we propose a hybrid feature extraction method to train a block-based CNN architecture. The proposed systems consist of two components namely hybrid feature extraction and CNN classifier. The five auditory features log-Mel spectrogram (LM), MFCC, chroma, spectral contrast and Tonnetz features are extracted and then LM & MFCC are combined as one feature set. LM, MFCC, and CST features are aggregated as another for training to the proposed two CNNs, respectively. The different sound samples of English alphabets are collected from different people of different age groups. The feature sets collected from the hybrid feature extraction methods are presented to both the proposed CNNs and the experimental results are collected. The experimental results indicate that the taxonomic accuracy of the proposed architectures can surpass the existing methods of CNNs with single feature extraction methods. The proposed second architecture performs more effectively over the proposed first CNN architecture.

Keywords: Deep Neural Network; Convolutional Neural Networks; Sound Recognition; MFCC; Classification

1. Introduction

The basic idea of any automatic speech recognition (ASR) is to paraphrase the mortal speech into spoken words. This task is veritably grueling due to variability in human speech signals. These variabilities are due to different speaker attributes, different speaking styles, and uncertain overlapped environmental noises. The system for automatic speech recognition needs to collude variable-length speech signals into variable length sequences of words or phonetic symbols. It has been observed that the Hidden Markov Models (HMMs) have been applied for handling variable length sequences as well as modeling the temporal behavior of speech signals using a sequence of states, each of which is associated with a particular probability distribution of observations [1]. A high-performance alphabet recognition system based on context-dependent phoneme HMM is proposed [2]. It used E-set letters consisting of the letters B, C, D, E, G, P, T, V and Z to perform the recognition experiment. The English Alphabet Recognizer (EAR) system has been proposed to perform recognition of isolated alphabets [3]. In this system, a rule-based segmented was used to segment the alphabet into four broad phonetic categories. Then, features were extracted from these broad phonetic categories and features were the input of back propagation neural network (BPNN) for classification. Gaussian mixture models (GMMs) have been also regarded as the most powerful model for estimating the probabilistic distribution of speech signals associated with each of these HMM states [4]. Meanwhile, the generative training methods of GMM-HMMs based

* Corresponding author: Pratibha Rashmi

on the expectation maximization (EM) algorithm have been considered for speech recognition [5]. Further, a plethora of discriminative training methods is considered to further improve HMMs to yield the state-of-the-art ASR systems [6]. Recently, HMM models considered with artificial neural networks (ANNs) created the significant resurgence of research interest for the automatic sound recognition System [7]. It considered first the TIMIT phone recognition task with mono-phone HMMs for MFCC features [8] and further on several large vocabulary tasks [9]. Very recently the deep learning methods are employed for the performance improvements in automatic sound recognition system. The deep learning techniques have been considered as the extension of Artificial neural networks architectures to accomplish the patten recognition tasks with increased number of hidden layers and to automate the feature extraction process with filters of receptive filed. The deep neural network replaced the conventional neural networks for pattern recognition tasks though the ANNs have been used for speech recognition for more than two decades. Early trials with neural network techniques worked on static and limited speech inputs where a fixed-sized buffer was used to hold enough information to classify a word in an isolated speech recognition scheme [10]. They have been also used in continuous speech recognition as feature extractors in TANDEM approach [11] and in bottleneck feature methods [12], and as nonlinear predictors to aid the recognition of speech units [13]. In an approach of artificial neural network, a combination of HMM and Restricted Boltzmann machine is used for the sound recognition [14]. In most of the architectures the Mel-Log Frequency spectral coefficient was used as an input for the system. Hybrid ANN-HMMs also now often directly use log Mel-frequency spectral coefficients without a decorrelating discrete cosine transform (DCT) and all of these factors have had a significant impact upon performance [15]. Further a deep neural network as a part of a hybrid Deep neural network-HMM model is applied on a small – scale speech task [16]. Later, a pretrained DNN-HMM is considered on acoustic modeling with varying depths of networks [17]. Further, the Deep neural network is used for speech recognition for large vocabulary speech tasks [18].

The convolutional neural networks (CNNs) lead in this domain due to its flexibility and ability with different adjusting parameters [19]. CNNs have been applied to acoustic modeling in which convolution was applied over windows of acoustic frames that overlap in time in order to learn more stable acoustic features for classes such as phone, speaker and gender [20]. The weight sharing with CNNs has been proposed to improve the performance of sound classification system to improve the learned acoustic features with a tolerance to small shifts in frequency such as those may arise from differing vocal tract lengths and has led to a significant improvement over Deep neural networks of similar complexity on speaker-independent phone recognition. Convolution neural networks is used for continuous speech recognition using raw speech signal [21]. The approach has been extended for large vocabulary speech recognition problem and compared the CNN-based approach against the conventional ANN-based approach on Wall Street Journal corpus [22]. Deep architectures have remarkable merit to enables a model to handle many types of variability in the speech signal. It has been shown, that the feature representations used in the hidden layers of DNNs are more invariant to small perturbations in the input, regardless of their deep structural insight or abstraction so, in this manner its leads to better model for generalization and to improve recognition performance, especially under speaker variations [23]. Although primarily CNN is used in visual recognition contexts but is has been also successfully applied in speech and music analysis [24]. Recently the Convolution neural network architectures are used for the classification of environmental sounds. Earlier classification of environmental sounds is still predominantly based on applying general classifiers like Gaussian mixture models, support vector machines and hidden Markov models. In all these classifiers the manually extracted features, such as Mel- frequency cepstral coefficients are used [25]. Mel-Frequency Cepstral Coefficients (MFCC) is one of the voice feature extraction techniques that are often used to distinguish one sound from other sounds [26]. The MFCC features can be classified from voice input to a specified class. However, the many variations of sound can cause the classification process to look for non-linear correlations. Therefore, to solve non-linear correlations problem many researchers tried to use machine learning techniques [27] specifically the classification using deep learning techniques. Among the different deep learning techniques, the convolution neural networks are considered as the most prominent model for the classification of sound samples. The convolution neural network used in different types of sound classification tasks due to its ability to automatically learn from the dataset while training. In some of the cases the CNN used for feature extraction followed by the SVM for the classification in the domain of sound classification and in some other work the handcrafted features are used with CNN to classify the sound of marine animals [28]. Further the use of combining deep learning (using CNN) and shallow learning for the problem of sound recognition with MFCC approach as baseline for both [29]. Thus, lots of attempt have been made to implement the classifications for different types of sound signals like environmental sound, animal sounds and human voice samples. The recognition of voice of human by its sound is still an open area of research in which the sound of English alphabets recognition is an important aspect. English alphabets sound signals of different people taxonomy generally consist of two basic components i.e., acoustic features and classifiers. To extract acoustic features, sound signals are first separated into frames with a cosine window function and then, features are extracted from each frame and this set of features is used as one instance of training or testing [30]. The classification result of one sound is the summation of probabilities predicted for each segment. Features derived from Mel filters: Mel Frequency Cepstral Coefficients (MFCC) and Log-Mel Spectrogram (LM) are two widely used features in automatic sound recognition [31]. Moreover, a considerable number

of research works indicated that combined features performed better than use only one feature set in Automatic sound classification tasks. While adding more conventional features cannot improve the performance. Hence, a suitable feature aggregate scheme is an essential part of sound taxonomy. Support-vector machines (SVM), Gaussian mixture model (GMM) extreme learning machine (ELM) are widely used classifiers in sound related classification tasks [32]. However, these conventional classifiers are designed to model small variations which result in the lack of time and frequency invariance. Recently, the convolutional neural network (CNN) is identified as one of the most used architectures of deep learning models, which could address the former limitations by learning filters that are shifted in both time and frequency [33]. The CNN is designed to process data that come in the form of multiple arrays of 1D for various kinds of sound signals, such as speech and music, and 2D for audio spectrograms. Even though CNN can solve the limitations of conventional classifiers, the longer temporal context information still cannot be captured by this method. Hence, several works proposed to use merged neural networks architectures [34]. In this approach, one or more CNNs are used to extract the spatial information with different acoustic features firstly. Then, the outputs are merged by concatenation and feed to recurrent neural network (RNN) layers or another CNN layers for temporal information extraction [35]. The decision level fusion framework method is used to integrate the different deep neural network architectures to obtain the taxonomy of sound signals for classification. This approach fused the softmax values acquired from different neural networks through mean calculation, or uncertainty reasoning algorithms such as Dempster—Shafer evidence theory (DS theory) and Bayesian Theory [36]. Even though some research works attempted to use deeper neural networks or stacked deep architectures to improve the taxonomic accuracy, however, the performance is still unsatisfactory. Hence, there is need to develop appropriate auditory features and novel neural network models to achieve high categorization accuracy for spoken English alphabets sound classification.

In this paper, we explore the use of deep learning approaches, specifically approaches based on the convolutional neural network (CNN), a deep learner that can automatically learn features directly from the dataset while training for the classification of sounds signals of English alphabets. In this proposed work, we consider two convolution neural network architectures. In first architecture, we propose a two-layer CNN architecture based on two combined auditory features and in the second architecture we propose a Block based CNN model which consist with different number of filters and size of the filters. The proposed systems consist of two components namely hybrid feature extraction and CNN classifier. The five auditory features log-Mel spectrogram (LM), MFCC, chroma, spectral contrast and Tonnetz are extracted and then LM & MFCC are combined as one feature set. LM, MFCC, Chroma, spectral contrast and Tonnetz are aggregated as another for training to the proposed two CNNs, respectively. The different sound samples of English alphabets are collected from different people of different age groups. The feature sets collected from the hybrid feature extraction methods are presented to both the proposed CNNs and the experimental results are collected. The experimental results indicate that the taxonomic accuracy of the proposed architectures can surpass the existing methods of CNNs with single feature extraction methods. The proposed second architecture performs more effectively over the proposed first CNN architecture.

The remaining structure of this paper is organized as follows. Section 2 presents in brief about the Convolution Neural networks and sound signals of the spoken English alphabets by different people. Section 3 describes the feature extraction and the architecture of the proposed models. The experiment results and analysis are shown in Section 4. In Section 5, the conclusion of our work is presented followed by the references.

2. Convolutional neural networks

Convolutional neural networks are an extension of the multilayer feed forward neural network model with more hidden layers. Thus, a deep neural network (DNN) refers to a feedforward neural network with many hidden layers. Each hidden layer consists with number of units (or neurons), each of which considers all outputs of the previous layer as input, multiplies with associated weight vector, sums the result, and passes it through a non-linear but continuous activation function. A typical convolutional neural network consists of a number of different layers stacked together in a deep architecture as an input layer, a group of convolutional and pooling layers, a fully connected hidden layers, and an output (loss) layer as shown in figure 1.

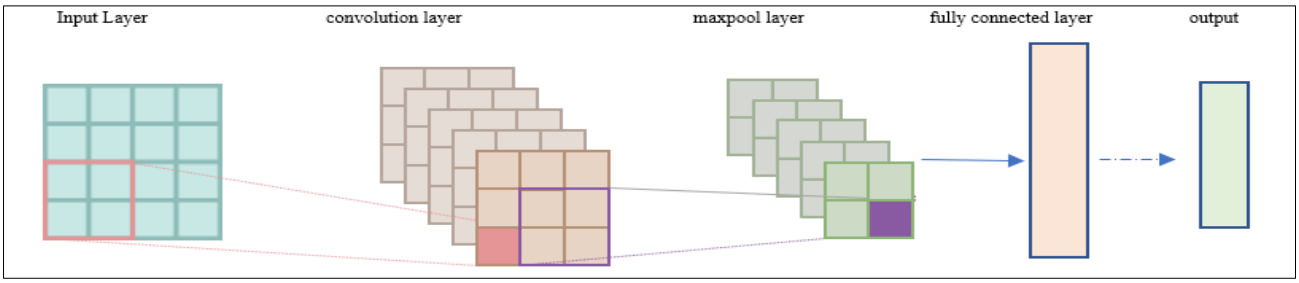

Figure 1 Convolution pooling operation performed on the input data

A convolutional layer introduces a special way of organizing hidden units which aims to take advantage of the local structure present in the two-dimensional input data. Each hidden unit, instead of being connected to all the inputs coming from the previous layer, is limited to processing only a tiny part of the whole input space considers as receptive filed. The weights of such a hidden unit create a convolutional kernel or filter which is applied to the whole input space, resulting in a feature map. A typical convolutional layer will consist of numerous filters and respectively the features maps. Let us consider a typical output for the i^{th} unit of the network as [37]:

$$S_i^l = f\sum_i (S_j^{l-1} w_{ij}^l + b_i^l) \quad .....(1)$$

Here, S_i^l denotes the output of the i^{th} unit in the l^{th} layer, represents the weight from the j^{th} unit in the $(l\text{-}1)^{th}$ layer to the i^{th} unit in the l^{th} layer, b is a bias added to the i^{th} unit and f is the nonlinear but continuous activation function. The Convolutional neural network considers the 2-dimensional input and passes it to the filters of receptive field. In the case of sound signal generally the input is presented in the form of Time frequency patches (TF). Let us consider a two-dimensional input X applies to the first convolution layer $(H1)$ which consists with m channels (filters) of size $n \times n$, where $n = 1, 2, \ldots \ldots \ldots . . N$. Let $[X]_{i,j}$ and $[H_1]_{i,j}$ denote the TF patch value at location (i, j) in the 2D representation of the Audio signal. Hence, this input data is processed through several trainable convolution layers for an appropriate representation of the input. Since the neurones in a layer are connected only to a small region of the previous layer so that, each of the hidden units receives input from each of the input pixels through the parameter weight tensor W. Let U contains biases, so that we can express the layer output as:

$$[H]_{i,j} = F\left[[U]_{i,j} + \sum_k \sum_j [W]_{i,j,k,l}[X]_{k,l}\right] \quad \ldots\ldots\ldots (2)$$

$$\text{Or, } [H]_{i,j} = F\left[[U]_{i,j} + \sum_a \sum_b [V]_{i,j,a,b}[X]_{i+a,j+b}\right] \ldots \ldots \ldots (3)$$

Such that $k = i + a$ and $l = j + b$

Here V represents the convolution filter or kernel of the convolution layer and F is a non-linear output function. Therefore, the deep convolution network is designed to learn the set of parameters V of convolutional layers and of the dense (fully connected) layers to map the input to the predicted output T. Generally, with the hierarchy of layers and to use the equation 3, we can express the predicted output T in the terms of unknown parameters (W) for fully connected network with non-linear output function F as:

$$T_1 = \emptyset[F(X|V)]$$

$$\text{or, } T_1 = \emptyset[F_L(\ldots \ldots \ldots F_2(F_1[V + V_1 \otimes X_1]|V_2)|V_L)] \ldots\ldots\ldots\ldots (4)$$

$$\text{and, } T = F(T_1|W) = F_o(F_H(T_1 * W_H + b_H) * W_o + b_o) \ldots\ldots\ldots\ldots (5)$$

Where, $\otimes$ represents the convolution operation or tensor product and * represents the dot product of the vectors, $\emptyset$ represents the max pool operator, T_1 is the final feature map obtained from the max pool layer inserted after the last convolution layer and b is a bias vector used by the layers of fully connected network, L is the number of convolution (hidden) layers of the network, X_1 is the 2 dimensional input matrix of N features maps, and V is a collection of the two dimensional filters. The output of the final convolution layer (after max-pooling) is flattered and used as input to the first layer of dense network. In the case of multiclass classification, the number of neurons in the output layer is considered according to the number of classes. Hence, for the output layer of the classification layer, the softmax activation function is used. The network is trained for the input samples and the parameters of the network are

optimized using mini-batch stochastic gradient learning and regularization methods to minimize the error or cross entropy. Normally, the pooling layers are used after the convolution layers to minimize the dimensionality of the feature map. The max pool or average pooling (local or global) are the most common pooling operations performed are taking the max or mean of the input cells. This down sampling further improves invariance to translations. The CNN also supports the mechanism to deal with the overfitting during the training of patterns. The dropout learning is used to tackle this problem. In this learning after each iteration every hidden unit is randomly removed with a predefined probability and the learning procedure continues normally. This random perturbation prevents the network from learning spurious dependencies and creates complex co-adaptions between hidden units. Architecture averaging introduced by dropout tries to ensure that each hidden unit learns feature representations that are generally favorable in producing the correct classification answer.

In the case of spoken English alphabets classification, the spoken alphabet is considered as the sound signal. The captured sound signals of English alphabets from different people are transform it into a visual image. There are various methods can be used to create the image namely spectrograms, harmonic-percussive spectrogram, and scattergrams. These visual representations of the audio signals are fed directly to the pretrained convolutional neural networks, that automatically learns the feature vectors. Audio signals are converted into spectrogram images that shows the spectrum of frequencies along the vertical axis as they vary in time along the horizontal axis as shown in figure 2 as the Spectrogram images of the English alphabet A from four different speakers and figure 3 as the spectrogram images of the English alphabets from A to D from the same speaker.

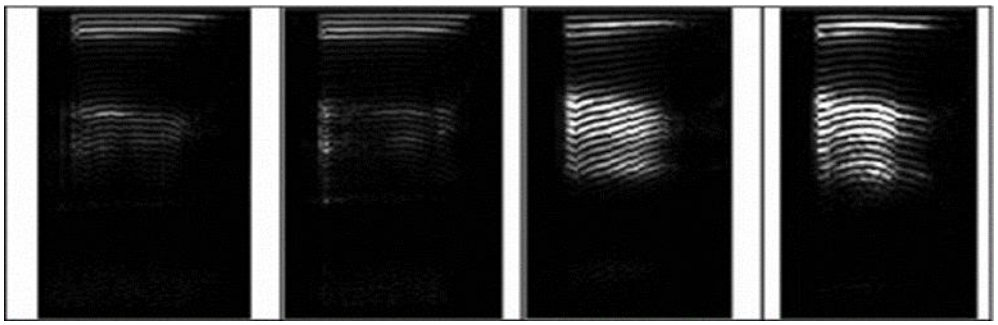

Figure 2 Spectrogram images of the English alphabet A from four different speakers

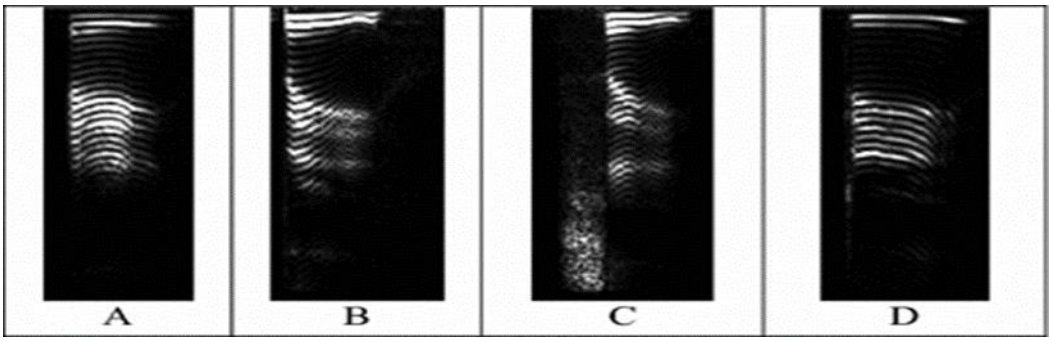

Figure 3 Spectrogram images of the English alphabet A–D from the same speaker

The input image of the sound samples can loosely be thought of as a spectrogram, with static, delta and delta-delta features. The spectrogram images as needed for inputs preserves locality in both axes of frequency and time. In the Convolutional neural networks, we normally consider a single window of input to network. It consists with a wide range of context i.e., 9 to 15 frames. The frequency can handle with the conventional network with the use of MFCCs. The frequency actually does present a major problem due to the discrete cosine transforms. It projects the spectral energies into a new basis that may not maintain locality. Thus, in this paper, we are using the computation of Spectrogram spectrum directly from the Mel-frequency coefficients without applying DCT and we consider it as the Mel-Frequency Spectral Coefficients (MFSC) features. These will be used to represent each speech frame, along with their deltas and delta-deltas, to describe the acoustic energy distribution in each of several different frequency bands. There exist several different alternatives to organizing these MFSC features into maps for the CNN. First, as shown in Fig. 4, they can be

arranged as three 2-D feature maps, each of which represents MFSC features (static, delta and delta-delta) distributed along both frequency using the frequency band index and time using the frame number within each context window. Therefore, a two-dimensional convolution is performed to normalize both frequency and temporal variations at the same time. On the other way, we may only consider normalizing frequency variations. In this case, the same MFSC features are organized as several one-dimensional (1-D) feature maps along the frequency band index. Thus, in this case a one-dimensional convolution will be applied along the frequency axis. In this paper, we are only focusing on this last arrangement as shown in Fig. 4 i.e., a one-dimensional convolution along frequency [37].

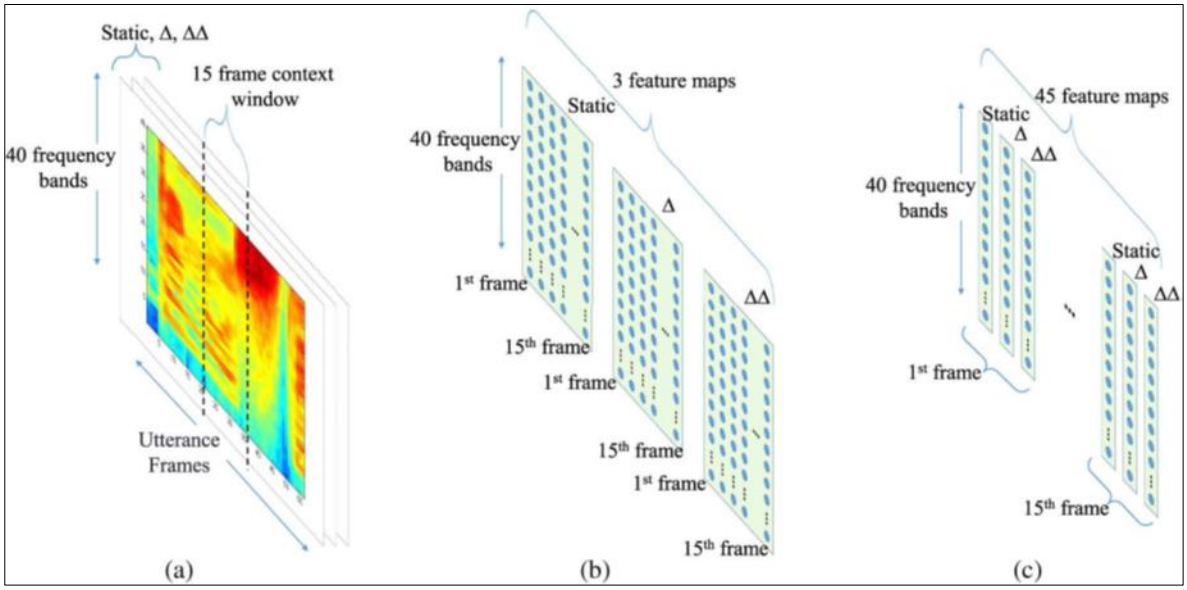

Figure 4 Two different ways to organize speech input features to a CNN

Once the input feature maps are formed, the convolution and pooling layers apply their respective operations to generate the activations of the units in those layers, in sequence, as shown in figure 5. Like those of the input layer, the units of the convolution and pooling layers can also be organized into maps. A deep CNN thus consists of two or more of these pairs in succession [37].

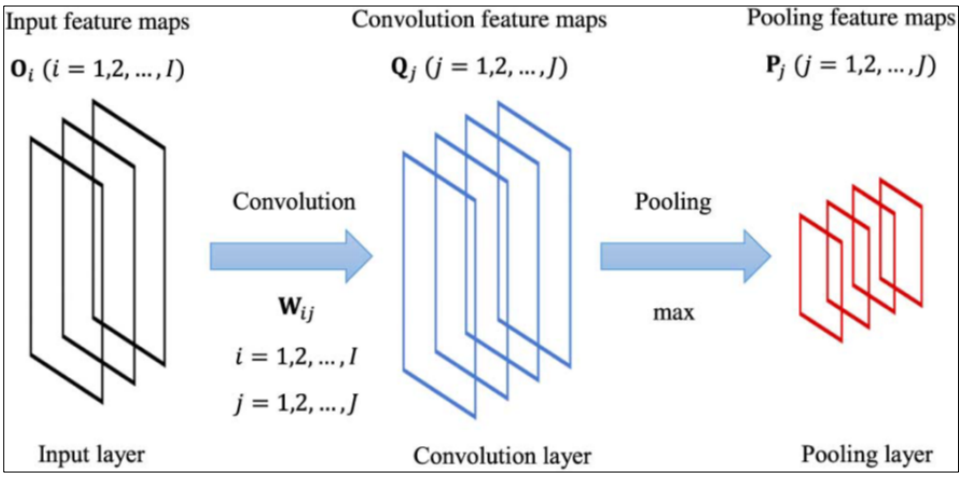

Figure 5 An illustration of one CNN "layer" consisting of a pair of a convolution and a pooling layer in succession.

Therefore, every input feature map $I_i(\mathrm{i} = 1, \ldots\ldots \mathrm{I})$ is connected to many feature maps i.e., $J_j(\mathrm{j} = 1, \ldots\ldots \mathrm{J})$ with connection strength $W_{ij}(i = 1, \ldots\ldots I; j = 1, \ldots\ldots J)$. The mapping can be represented as the well-known convolution operation in signal processing. The I D features maps of all inputs can be computed in the convolution layer as:

$$S_{i,j} = f(\sum_{i=1}^{I} \sum_{n=1}^{M} I_{i,n+m-1} W_{i,j,n} + b_{0,j}); j(= 1, \ldots\ldots J) \ldots (6)$$

Or, in vector form, $S_j = f(\sum_{i=1}^{I} I_i \otimes \boldsymbol{W}_{i,j})$ ..... (7)

Here, $I_{i,n+m-1}$is the *m-th* unit of the *i-th* input feature map, $J_j$is the m-th unit of the feature map and $W_{i,j,n}$ is the *nth* element of the weight vector. M is considered as the size of the filter which determines the number of frequency bands in each input feature map that each unit in the convolution layer receives as input to consider the issue of locality that arises from the choice of MFSC features. Thus, these feature maps are confined to a limited frequency range of the speech signal. The output of the convolution layer is presented to the pooling layer to reduce the dimensionality of the feature maps. Thus, the units of pooling layer serve as generalization over the features of the previous convolution layer and due to this the generalization will again be spatially localized in frequency. Generally, the max or average pooling are used in the CNN as:

$$p_{i,m} = max_1^G S_{i,(m-1)\times s+n} \text{ ..... (8)}$$

$$\text{And, } p_{i,m} = r \sum_{n=1}^{G} S_{i,(m-1)\times s+n} \text{ ..... (9)}$$

Here *r* is a scaling factor, *G* is the pooling size and *s is the shift size* of the pooling window which determine the overlap of adjacent pooling windows. In the case of sound signal classification, we considered *G* and *s* independently.

In the process of learning for the network, all weights in the convolution layers can be learned using the mini batch stochastic gradient descent method. In this process of learning we consider the input and convolution feature maps in the vector form $\dot{p}$ and $\dot{S}$ respectively. The input feature map can be expressed as:

$$\dot{p} = [v_1|v_2| \dots \dots \dots |v_m] \text{.... (10)}$$

Here, v_m is a row vector containing the values of the m^{th} frequency band along all *I* feature maps and *M* is the number of frequency bands in the input layer. Thus, the weight change in the convolution layers can be computed as:

$$\triangle \widehat{W} = \mathfrak{y}.\hat{S}'.E(l) \text{..... (11)}$$

$$E(l) = (E(l-1)(W(l+1)') * \boldsymbol{S(t)} * (\boldsymbol{1} - \boldsymbol{S(t)}) \text{.... (12)}$$

$$where, l = L-1, \dots \dots 2,1)$$

In the case of convolution neural network, the total weight update for the network can be expressed as: $\triangle w_{i,j,n} = \sum_m \triangle \widehat{\boldsymbol{W}}\, i + (m+n-2) \times I, j + (m-1) \times J$..... (13)

Here, I and J are the number of feature maps in the input layer and convolution layers. There is no weight in pooling layers so there will be no change occurs in the pooling layers.

3. Implementation and Experiment Design

In our proposed system, we considered the sound samples of English alphabets spoken by different people. The dataset of CSLU is used to collect the samples. The CSLU: ISOLET Spoken Letter Database version 1.3 datasets includes 7800 spoken letters each by 150 speakers labelled the English alphabets sounds (the length is less than or equal to 4 s) collected from the real-world, totaling 9.7 h. The dataset is separated into 26 audio event classes. These samples were transformed in the Mel-Log spectrum. Initially, all sound clips are converted to the single channel wave files with the frequency of 22,050 Hz. Then, divided into 40 frames with an overlap of 50% where each frame is about 23 milli second. We use the pre-setting channels of Librosa to extract the Chroma, Spectral Contrast and Tonnetz features. For the MFCC extraction, the value of first twenty channels with their first and second order derivatives are used, resulting in 60-dimensional feature vectors. The channels of Log-Mel Spectrogram are set to 60, to make the dimension to be equal to the MFCC. Then, all the spectrograms are represented as a matrix with a size of 40 X 60. The feature size of chroma, tonnetz and spectral contrast is 40 X 7, 40 X 6 and 40 X 12, separately. The speech recognition model that we used in this paper is a Convolutional Neural Network (CNN), CNN is a form of artificial neural network that has a 3-dimensional input type. Because the sound input that has been extracted using the MFCC feature only leaves a 1-dimensional vector shape, we must make a few changes to our feature vector as input for CNN. We use MFCC feature vectors with size 40, we make it constant because CNN cannot process vectors of varying sizes although MFCC vectors might vary in size for different audio input. For that we must make an MFCC feature vector that uniforms in size. In this preprocessing task if after the MFCC process is obtained more than 40 elements in the extracted feature vector then the excess is removed,

whereas if less than 40 features are padded by filling in the remaining vectors with the number 0. The sound input is sampled with several 60 samples per sound, and we extracted each sample with MFCC and give us 40 features, so now we have a two- dimensional matrix that represents the number of features and sound sampling. Therefore, our input feature map is a matrix of the size 40X60 as shown in figure 6.

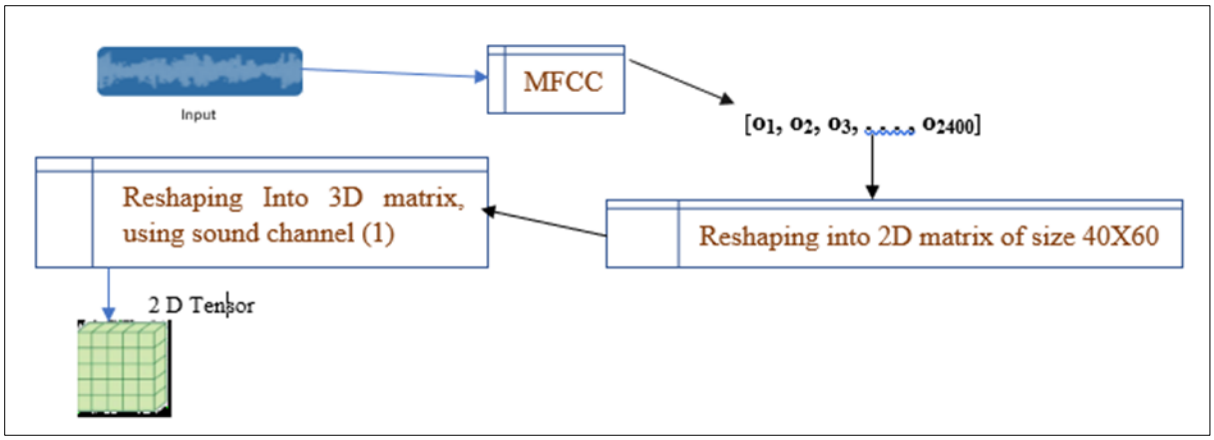

Figure 6 Sound signal pre-processing to make an input feature map

After constructing the input feature maps of sound samples, we considered the two CNN models. The training set and test data sets are constructed from the samples. Hence out of 7500 samples we considered 6500 samples for the training set and 1000 samples were used for the testing. We considered the two different types of convolution neural networks for the classification of sound samples. The first architecture of CNN has two 2D convolutional layers with 64 filters followed by a max-pool layer. The second last layer is a fully connected layer with 500 neurons. It considered the rectified linear activation function, and the last layer is the classification layer with 26 neurons and the Softmax activation function. Training is performed using mini batch stochastic gradient decent method with 50 epochs as shown in figure 7.

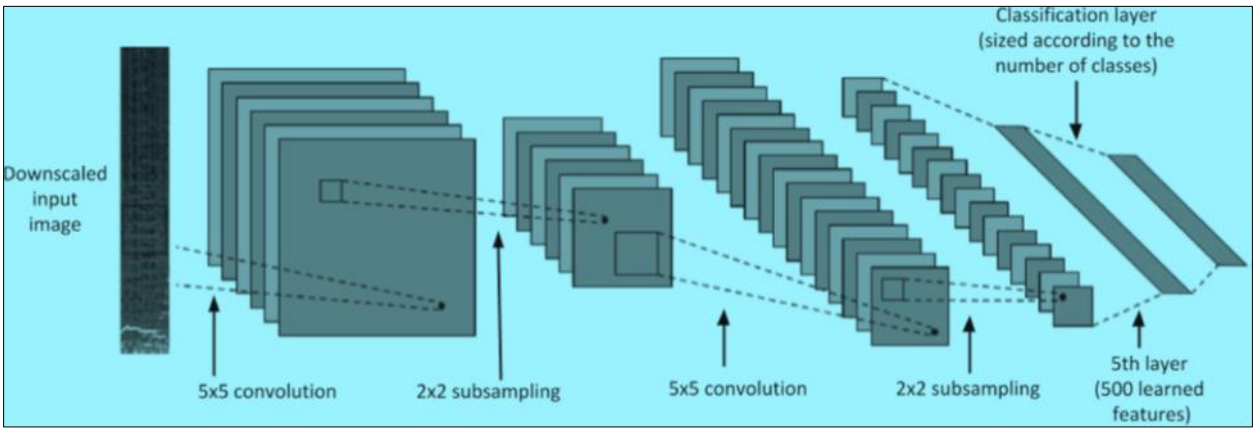

Figure 7 The proposed CNN architecture for sound classification

Further, the second model of CNN is used for the sound classification. In this model, we combined LM, MFCC and CST together to form a new feature set called MLMC as shown in figure 8, to make a further investigation of the influence of various feature combination strategies in spoken sound classification tasks.

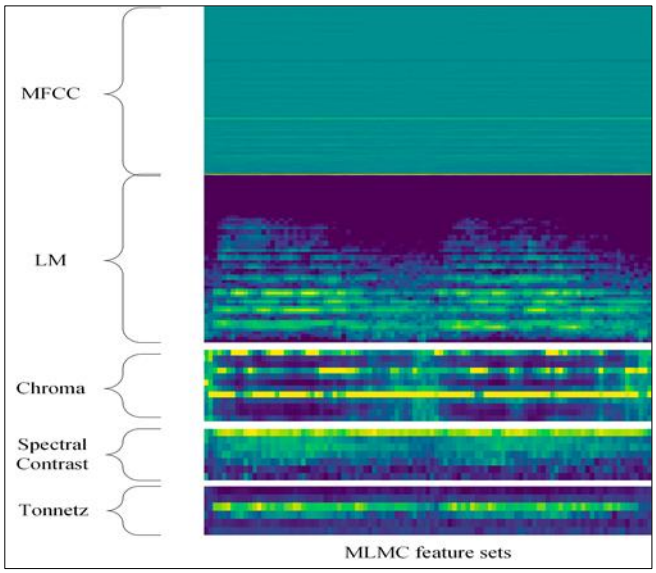

Figure 8 The spectrogram of MLMC feature sets

In this proposed architecture, we considered the 5 convolutional blocks followed by dropout regularization and fully connected neural network. The fully connected neural networks is followed by softmax layer and the classification output. In each convolutional block we considered a convolution layer, batch normalization with rectified liner activation function followed by the Max-pool layer. The input feature map size of the MLMC is 40X175 and the first convolutional block used 35 kernels of size 3X3 and max pool layer with stride 2. The same pattern is followed in next two convolutional blocks. The 4th and 5th convolutional blocks used the 64 kernels of size 5X5. Again, the same max pooling method is applied and the output the last max pooling is passed to the dropout unit where almost 50% units are dropped on random basis. The second last layer is a fully connected layer with 300 neurons. It considered the rectified linear activation function, and the last layer is the classification layer with 26 neurons and the Softmax activation function. The proposed architecture can be presented as shown in figure 9.

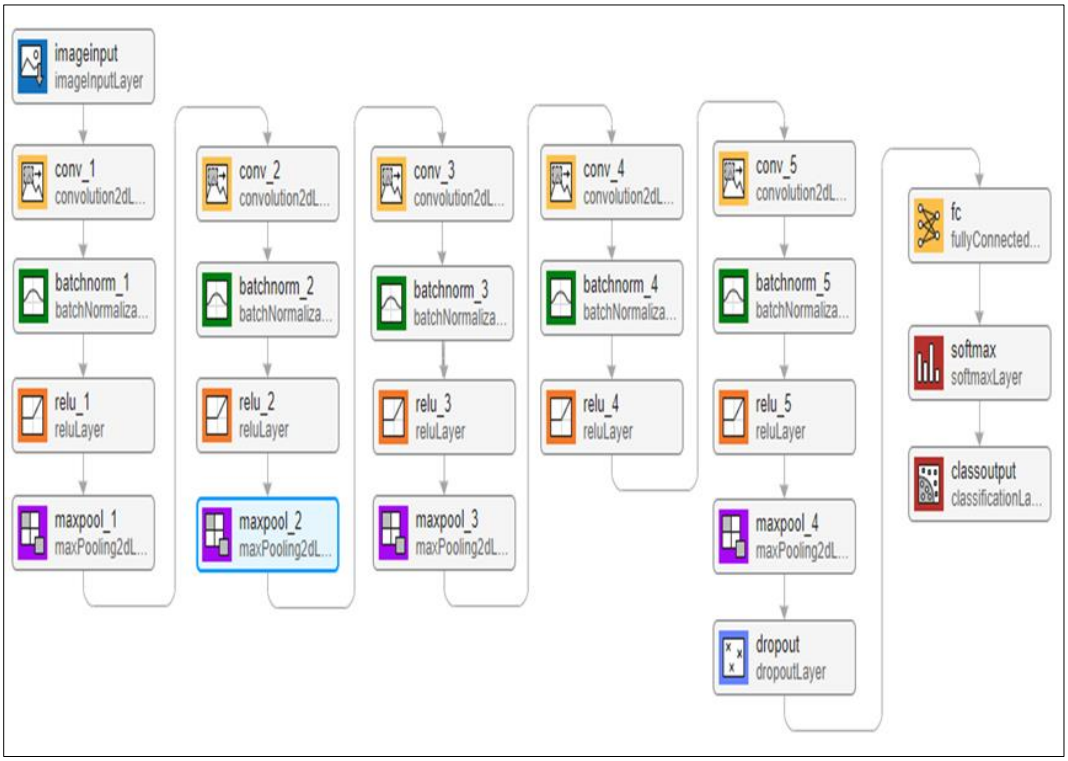

Figure 9 The proposed CNN with 5 convolutional blocks

The simulation results from both proposed models are obtained on the CSLU: ISOLET Spoken Letter Database version 1.3 dataset with different epochs parameters of networks for training. Parameters for both proposed CNN architectures are presented in table 1 and table 2 as:

Table 1 Parameters for the first Proposed CNN Architecture

No	Layer Type	No. of Filter/Units	Kernel size	Pool Size	Activation
1	CONV 1	64	5X5		RELU
2	POOL	Max Pooling		2X2	
3	CONV2	64	5 X 5		RELU
4	POOL	Max Pooling		2X2	
5	DROPOUT	50%			FLATTEN
6	DENSE	500			RELU
7	CLASSIFICATION	26			SOFTMAX

Table 2 Parameters for the Second Proposed CNN Architecture

No	Layer Type	No. of Filter/Units	Kernel size	Pool Size	Activation
1	CONV 1	35	3X3		RELU
2	POOL1	Max Pooling		2X2	
3	CONV2	35	3X3		RELU
4	POOL2	Max Pooling		2X2	
5	CONV3	35	3X3		RELU
6	POOL3	Max Pooling		2X2	
7	CONV4	64	5X5		RELU
8	POOL4	Max Pooling		2X2	
9	CONV5	64	5X5		RELU
10	POOL5	Max Pooling		2X2	
11	DROPOUT	50%			
12	DENSE	300			RELU
13	CLASSIFICATION	26			SOFTMAX

The simulations result of training and testing are obtained. Both the proposed CNN architectures were trained for 50 epochs for a batch and total 10 batches were constructed. Thus, the total epochs are considered 500 for the training.

4. Results and Discussion

Two experiments are conducted to obtain the results. In the first experiment Convolution neural networks is used with two convolution layers and two pooling layers. The pooling is used after each convolution layer. The pooling down sampled the feature map obtained after each convolutional layer. The number of filters and the size of filters were set same for both the layers. The two-phase regularization and normalization are used to avoid the overtraining and two layers of fully connected neural network is used before the classification layer. The input feature map of is constructed with MFCC spectrum and the 2D tensor of size 40 X 60 is used for the training. We conduct training with as many as 500 epochs of the training sound dataset that we have separated with sound validation data. We get pretty good accuracy

with training accuracy reaching 100% and training losses close to 0. Then we use the model generated from the training results to validate the 26 validation sound data. The training accuracy versus loss, plot result can be seen in figure 10. It can be seen in figure 10 that before the 250th epoch the accuracy value of the training is already close to 100% but the loss value is still not convergent, but after the 300th epoch the accuracy and loss values begin to converge, and the model becomes stable in doing classification. The model itself only cost around 6 MB in size and can use to predict 3 second length input sound with just around 1 second waiting time.

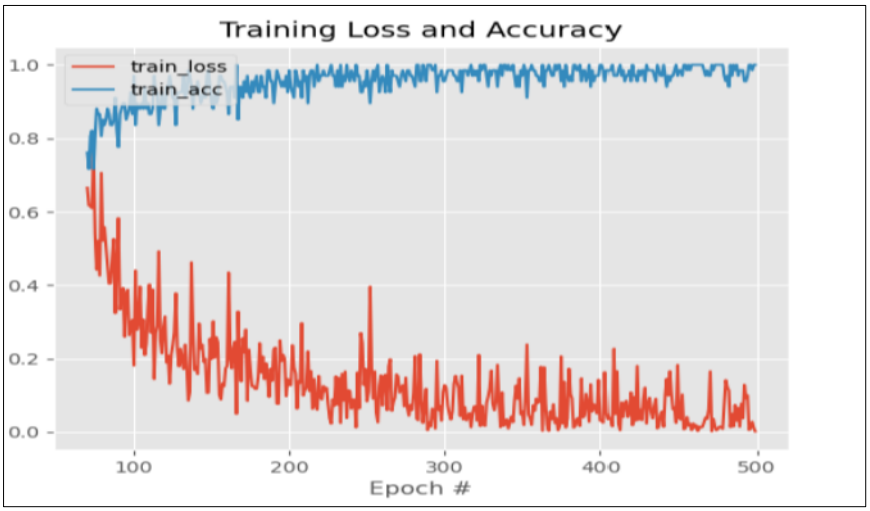

Figure 10 Training plot for speech classification with first proposed CNN architecture

After the model formulation and training process the validation process starts. It is using 26 validation data that had been prepared. Because the classification is expected to classify into 26 classes so that the accuracy, precision, and recall of the model is calculated that is produced through validation data. The confusion matrix for the classification of English alphabets van be seen in figure 11. Therefore, from the confusion matrix the accuracy, precision and recall can be calculated to measure the accuracy of the model for the classification. The exactness of the classifier is how much forecast of a specific class coordinates with the genuine worth. Exactness is the main boundary which shows the level of accurately ordered subjects and is determined utilizing equation 14, 15 and 16 as:

$$Accurcy = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{TP(i)+FN(i)}{TN(i)+TP(i)+FP(i)+FN(i)}\right) \quad \ldots\ldots\ldots\ldots (14)$$

$$Precision = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{TP(i)}{TP(i)+FP(i)}\right) \quad \ldots\ldots\ldots\ldots\ldots\ldots (15)$$

Another parameter is F-Score. It is likewise a huge proportion of the order execution which depicts the connection between affectability and accuracy of the classifier. It can be determined as:

$$F - Score = 2 \times (Precision \times Recall)/(Precision + Recall) \qquad (16)$$

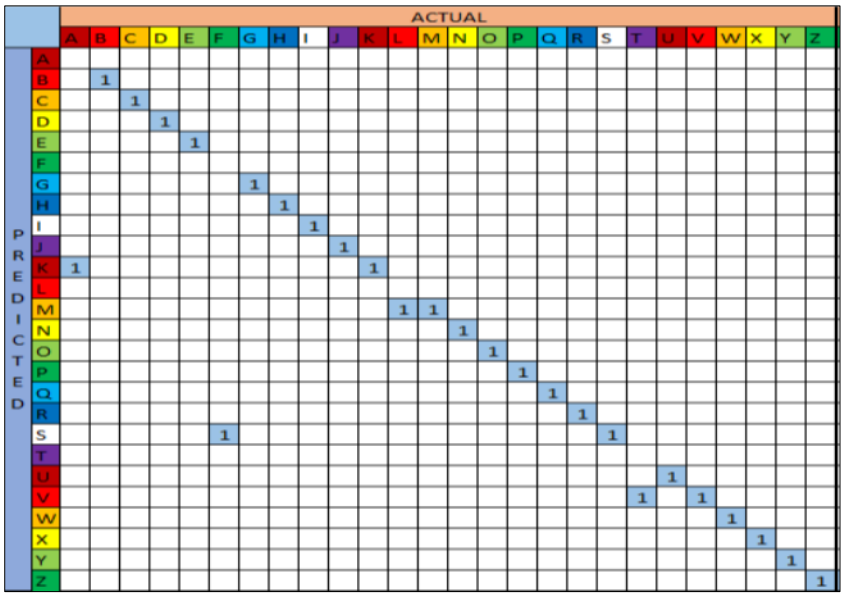

Figure 11 Confusion matrix of the validation data for the first proposed CNN

Where, Recall is defined as: $Recall = (\frac{TP}{(TP+FN)}) \times 100\%$.....(17)

Therefore from the confusion matrix as shown in figure 11 and with the equations 14, 15, 16 and 17 the accuracy, precision, recall and be calculated as follows:

$Accuracy$= ($correctly\ classified\ data\ /\ number\ of\ data\ to\ be\ classify$) ×100%

Hence, $Accuracy$= (22 / 26) ×100%=84%

$Precision$= (TP / (TP+FP)) ×100%

Hence, $Precision$= (22 / 26) ×100%=84%

The Recall can be obtained with equation 17 as:

$Recall$= (22/22) ×100%=100%

The F-Measure can be obtained from equation 16 as:

$F-Measure$= 2 × (0.84 × 1)/ (0.84+1) =0.91

Thus, from the $F-Measure$ it can be seen that the classification performance is pretty good. It is 91%. It shows that the proposed architecture of CNN can classify the 90 out of 100 spoken English alphabets correctly its respective classes.

In the second experiment, Convolution neural networks is used with five convolutional blocks followed by fully connected layer and a classification layer. The first, second and third convolution blocks consist with convolutional layer with 35 filters of size 3X3 followed by the max pooling of size 2 X 2 with stride of 2. The batch normalization is used in each block. The fourth and fifth convolutional blocks consist with 64 filters of size 5 X 5 followed by the max pooling of size 2 X 2 with stride of 1. The 50% dropout rate is used after the fifth convolutional block. The fully connected neural network is used with 300 neurons followed by the classification layer. The input feature map of is constructed with the combined spectrum of LM, MFCC and CST together to form a feature map MLMC. The MLMC feature map of size 40 X 175 is used as for the training. We conduct training with 500 epochs of the training sound dataset that we have separated with sound validation data. We get pretty good accuracy with training accuracy reaching 100% and training losses close to 0. Then we use the model generated from the training results to validate the 26 validation sound data. The training accuracy versus loss, plot result can be seen in figure 12. It can be seen in figure 12 that the convergent rate is fast with respect to the first architecture.

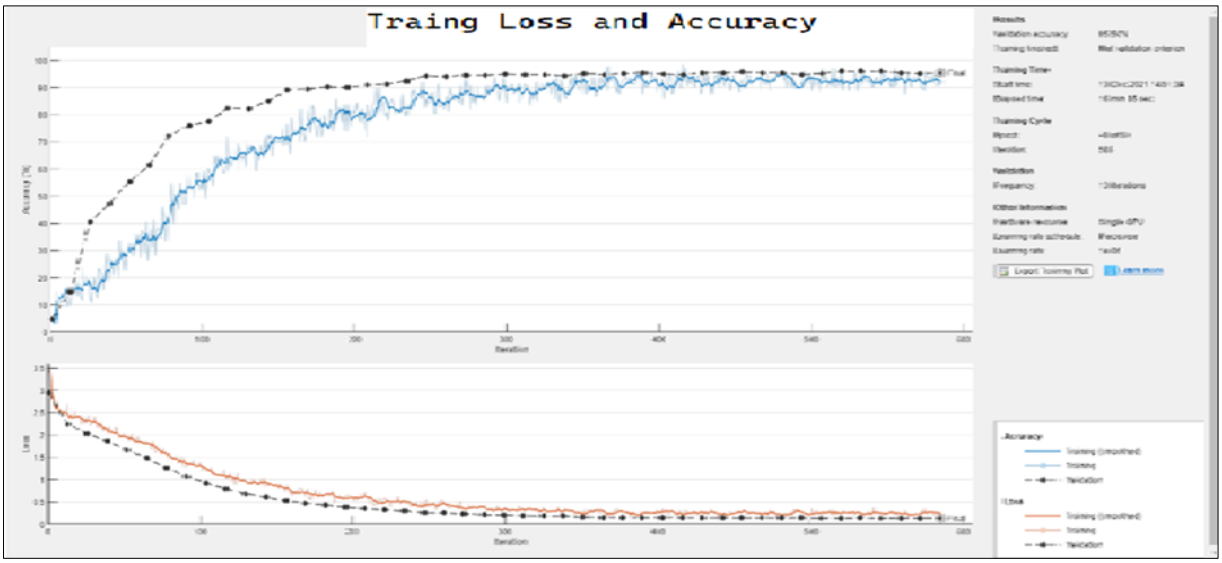

Figure 12 Training plot for speech classification with Second proposed CNN architecture

The confusion matrix for the classification of English alphabets can be seen in fig 13. Therefore, from the confusion matrix the accuracy, precision and recall can be calculated by using the equations 14,15,16 and 17 to measure the accuracy of the model for classification as:

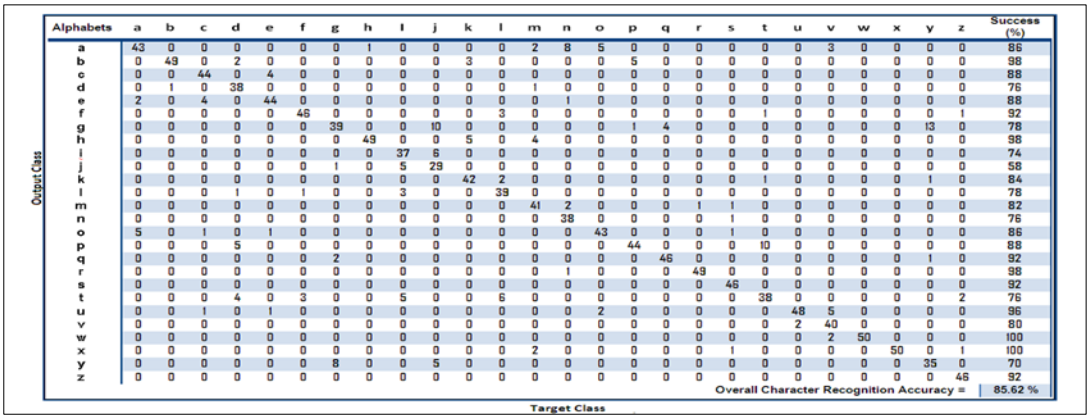

Figure 13 Confusion matrix of the validation data for the Second proposed CNN

$Accuracy$= ($correctly\ classified\ data$ / $number\ of\ data\ to\ be\ classify$) ×100%

Hence, $Accuracy$= (23 / 26) ×100%=86%

$Precision$= (TP / (TP+FP)) ×100%

Hence, $Precision$= (23 / 26) ×100%=86%

The *Recall* can be obtained with equation 17 as:

$Recall$= (23/23) ×100%=100%

The *F-Measure* can be obtained from equation 16 as:

$F-Measure$= 2 × (0.85 × 1)/ (0.85+1) =0.92

Thus, from the $F-Measure$ it can be seen that the classification performance is better than the first proposed CNN architecture. It is 92%. It shows that the proposed architecture of CNN can classify the 92 out of 100 spoken English alphabets correctly its respective classes. Therefore, the second proposed architecture preforms slightly better than first architecture. The rate of convergence is also fast for the second architecture. Hence, the feature extraction method which has been selected for the second approach i.e., combination of LM, MFCC and CST together to form a feature map MLMC spectrum provides better feature maps for training and more relevant features are available for the sound samples to train them with the Convolution neural networks.

5. Conclusion

In this paper, we proposed the two different methods of feature extractions from the sound's samples of English Alphabets and beside this the two CNNs models were proposed to measure the accuracy in the classification of these samples. The sound samples were classified in 26 classes i.e., one class for each alphabet. Input feature maps were constructed with MFCC spectrum and MLMC spectrum. These two preprocessing techniques to transform the sound signals into the 2D tensor form exhibits interesting observations. The MFCC is using the matrix of size 40X60 whereas the MLMC used matrix of size 40X175. Hence, more information was found available in MLMC spectrum for training. The feature map obtained from MFCC is presented to the first proposed CNN model which consist only two convolution layers and two pooling layers. The single dense layer is used with 500 neurons. On the other hand, the feature map obtained from MLMC spectrum is presented to the second CNN architecture which consist with 5 convolution blocks. In each block we had a convolution layer and a max pooling layer. The different size and numbers of filters are used in first three blocks and in last two blocks. The single layer fully connected network is used with 300 units. The second model is more complex but it took less time in convergence with respect to the first model. The classification accuracy of second model is also found better than the first one. Therefore, the MLMC is providing more relevant feature during training and testing over simple MFCC spectrum feature map. Beside this, the different size of filters and varying the number of filters in different convolution layers improves the performance in classification for validation data and improves the convergence rate during training. This study shows that the proposed CNN models with both the preprocessing techniques are capable of recognizing speech, with data in the form of sound. Although, the presented approaches and models have a good success in recognizing sounds, but it needs to be more improve. Beside this, the model cannot used for the recognition of words. This opens the opportunities for further research in this direction.

Compliance with ethical standards

Acknowledgments

This research is funded by Uttar Pradesh Government, Lucknow, India in the form of a major research project: 47/2021/606/seventy/4-2020-4(56)/2021.

Disclosure of conflict of interest

The authors whose names are listed immediately below certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

References

[1] Chu S, Narayanan S, and Kuo CC. (2009). Environmental sound recognition with time-frequency audio features. IEEE Transactions on Audio, Speech, and Language Processing, 17(6), 1142–1158.

[2] Jiang H. (2010). Discriminative training for automatic speech recognition: A survey. Computer speech, Language, 24(4), 589–608.

[3] Loizou PC, Spanias AS. (1996). High-Performance Alphabet Recognition. IEEE Transactions on Speech and Audio Processing, 4, 430-445.

[4] Nguyen QT, & Bui T D. (2016). Speech classification using SIFT features on spectrogram images. Vietnam Journal of Computer Science, 3, 247-257.

[5] Morgan N. (2012). Deep and wide: Multiple layers in automatic speech recognition. IEEE Trans. Audio, Speech Lang. Process, 20 (1), 7-13

[6] Deng L, Li X. (2013). Machine learning paradigms for speech recognition: An overview. IEEE Trans. Audio, Speech, Lang. Process., 21 (5), 1060-1089

[7] Ali H, Tran SN, Benetos E, Gareez AS (2018). Speaker recognition with hybrid feature from a deep belief network, Neural Computing & Application, 29, 13-19.

[8] Hagen A, Morris A. (2005). Recent advances in the multi-stream HMM/ANN hybrid approach to noise robust ASR. Computer Speech & Language, 19 (1), 3-30.

[9] Mohamed A, Dahl G, and Hinton G. (2009). Deep belief networks for phone recognition. Proceedings of NIPS Workshop Deep Learning Speech Recognition Related Applications,

[10] Pan J, Liu C, Wang Z, Hu Y, Jiang H. (2012). Investigation of Deep Neural Networks (DNN) for large vocabulary continuous speech recognition. Proceedings of ISCSLP.

[11] Salam MSH, Mohamad D, Salleh S. (2011). Malay Isolated Speech Recognition Using Neural Network: A Work in Finding Number of Hidden Nodes and Learning Parameters. International Arab Journal of Information Technology, 8 (4), 364-371.

[12] Zhu Q, Chen B, Morgan N, Stolcke A. (2005). Tandem connectionist feature extraction for conversational speech recognition. Machine Learning for Multimodal Interaction: Springer, 3361, 223–231.

[13] Bao Y, Jiang H, Dai LR, Liu C. (2013). Incoherent training of deep neural networks to de-correlate bottleneck features for speech recognition. Proceeding IEEE international conference Acoust., Speech, Single Process, 6980-6984.

[14] Deng L, Hassanein K, Elmasry M. (1994). Analysis of Correlation structure for a neural predictive model with applications to speech recognition. neural network, 7 (2), 331-339.

[15] Hinton G. (2002). Training products of experts by minimizing contrastive divergence. Neural Computing, 14, 1771-1800.

[16] Ji J, Dai W, Metze F, Qu S, Das S. (2017). A comparison of Deep learning methods for environmental sound detection. Proceedings of 2017 IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP), 126-130.

[17] Li Q, Zhang C, Woodland PC. (2023). Combining hybrid DNN-HMM ASR System with attention-based models using lattice rescoring. Speech Communication, 147, 12-21.

[18] Zhao T, Zhao Y, Chen X. (2016). Ensemble Acoustic Modeling for CD-DNN-HMM using Random Forests of Phonetic Decision Trees. Journal of Signal Processing System, 82, 187-196.

[19] Nanni L, Costa YMG, Lucio DR, Silla Jr CN, Brahnam S. (2017). Combining visual and acoustic features for audio classification tasks. Pattern Recognition Lett., 88, 49-56.

[20] Li S, Yao Y, Hu J, Liu G, Yao X, Hu J. (2018). An Ensemble Stacked Convolutional Neural Network Model for Environment Event Sound Recognition. Appl. Sci. 8, 1152-1171.

[21] Feng Y, Yang J. (2021). A Deep Neural Network Model for Speaker identification. Appl. Sci., 11(8), 3603.

[22] Kubanek M, Bobulski J, Kulawik J. (2019). A Method of Speech Coding for Speech Recognition Using a Convolutional Neural network. Symmetry, 11(9), 1185.

[23] Siniscalchi M, Svendsen T, Lee CH. (2014). An artificial neural network approach to automatic speech processing. Neurocomputing, 140, 326-338.

[24] Deng L, Hinton G and Kingsbury B. (2013). New types of deep neural network learning for speech recognition and related applications: an overview. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 8599-8603.

[25] Korvel G, Treigys P, Tamulevicus G, Bernataviciene J, and Kostek B. (2018). Analysis of 2D Feature Spaces for Deep Learning-Based Speech Recognition. J. Audio Eng. Soc., 66(12), 1072-1081.

[26] Piczak KJ. (2015). Environmental sound classification with convolutional neural networks. 2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP), 1-6.

[27] Bezoui M, Elmoutaouakkil A and Beni-hssane A. (2016). Feature extraction of some Quranic recitation using Mel-Frequency Cepstral Coeficients (MFCC). 5th International Conference on Multimedia Computing and Systems (ICMCS), 127-13.

[28] Khrisne DC, Hendrawati T. (2020). Indonesian Alphabet Speech Recognition for Early Literacy using Convolutional Neural Network Approach. Journal of Electrical Electronics and Informatics, 4(1), 34.

[29] Pineda XA, Ricci E, Yan Y, Sebe N. (2016). Recognizing Emotions from Abstract Paintings Using Non-Linear Matrix Completion. Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 5240-5248.

[30] Ko K, Park S and Ko H. (2018). Convolutional Feature Vectors and Support Vector Machine for Animal Sound Classification. 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 376-379.

[31] Puig P, Jordi, and Serra X. (2019). Randomly weighted CNNs for (music) audio classification. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

[32] Rajasekhar A and Hota MK. (2018). A Study of Speech, Speaker and Emotion Recognition Using Mel Frequency Cepstrum Coefficients and Support Vector Machines. International Conference on Communication and Signal Processing (ICCSP), 0114-0118.

[33] Lee CH, Han CC and Chuang CC. (2018). Automatic Classification of Bird Species from Their Sounds Using Two-Dimensional Cepstral Coefficients. IEEE Transactions on Audio, Speech, and Language Processing, 16(8), 1541-1550.

[34] Su Y, Zhang K, Wang J, Madani K. (2019). Environment Sound Classification Using a Two-Stream CNN Based on Decision-Level Fusion. Sensors 19(7), 1733.

[35] Parascandolo G, Heittola T, Huttunen H, Virtanen T. (2017). Convolutional recurrent neural networks for polyphonic sound event detection. IEEE/ACM Trans. Audio Speech Lang. Process. 25, 1291–1303.

[36] Li J, Qiu T, Wen C, Xie K, Wen FQ. (2018). Robust Face Recognition Using the Deep C2D-CNN Model Based on Decision-Level Fusion. Sensors, 18, 2080.

[37] Hamid OA, Mohamed AR, Jiang H. Deng L, Gerald P. Yu D. (2014). Convolutional Neural Networks for Speech Recognition. IEEE/ACM Transactions on Audio, Speech and Language Processing, 22(10),1533-1545.