Mathematical Statistician and Engineering Applications
ISSN: 2094-0343
2326-9865

Enhancing Real-Time Speech Recognition with hybrid system by using Adam Optimization, CNNs and SVM on GPU

Bhosale Rajkumar S.

Amrutvahini College of Engineering, Sangamne,

Department of Information Technology,

Maharashtra, India, bhos_raj@rediffmail.com

Panhalkar Archana R.

Amrutvahini College of Engineering, Sangamne,

Department of Information Technology,archana10bhosale@rediffmail.com

Article Info
Page Number: 12468-12488
Publication Issue:
Vol. 71 No. 4 (2022)

***Abstract*—** ASR systems can be used for a wide range of applications, including virtual assistants, voice search, dictation, and voice-controlled devices. They can also be integrated with other technologies such as natural language processing (NLP) and machine learning to provide even more advanced functionalities, such as sentiment analysis and personalized recommendations. However, it is important to note that ASR technology is not without its challenges, such as dealing with variations in accents, background noise, and speech disorders. Nonetheless, ongoing research and development in this field is expected to lead to further improvements in ASR technology and its applications.

Real-time speech recognition is an important technology that allows machines to transcribe spoken words into written text in real-time. In recent years, hybrid systems that combine multiple approaches, such as deep neural networks (DNNs) and support vector machines (SVMs), have shown promising results in improving the accuracy of speech recognition. In this approach, the convolutional neural network (CNN) is used to extract features from the speech signal, which are then fed into a SVM for classification.

To further enhance the performance of the system, Adam optimization is employed as an algorithm for training the hybrid system. Adam optimization is a stochastic gradient descent (SGD) optimization algorithm that has been shown to perform well in optimizing deep neural networks. To accelerate the processing speed of the system, GPU is utilized for parallel processing. This allows for faster computation and thus enables the system to perform real-time speech recognition.

Overall, this hybrid system using Adam optimization, CNNs and SVM on GPU shows promise in achieving high accuracy and real-time performance in speech recognition. The previous system outperformed for 11 labels with Google TebsorFlow and AIY teams, it contains 105,000 wave audio files and five layer model which achieve accuracy of 94.9% in less training time of 4.5116 sec using GPU. Beyond this our system also work for real time command used form user like ON, OFF, LogOff, Shutdown, Open, Close using this it is possible to operate computer system by blind or any handicap person fluently. The previous system work for deep neural network classification system applied on 65000

WAVE Google's Tens or flow dataset and AIY commands also it is possible to apply TIMIT slandered dataset of Speech with real time user data also. For feature extraction use hybrid system Mel Spectrogram and LPC extract from the input speech and Adam optimization algorithm perform training on Convolutional neural network (CNN) and Support vector machine. Both SVM and Convolutional Neural Network and SVM system proves to outperform than other models and can achieve accuracy of 98.2% for 6 labels of data as well as real time data recognition with best and accurate accuracy.

Article History
Article Received: 25 August 2022
Revised: 30 September 2022
Accepted: 15 October 2022

***Keywords*—**Hybrid Model, Graphics processing unit (GPU), SVM, Mel Spectrogram, Adam Optimization, Convolutional Neural Network (CNN), linear discriminant analysis algorithm (LDAA).

1. INTRODUCTION

Very deep convolutional networks (VDCNs) are neural networks that consist of multiple convolutional layers stacked on top of each other. These networks are used primarily for image recognition tasks, such as object detection and image classification.

The idea behind VDCNs is to increase the depth of the network in order to capture more complex features of the input image. This is achieved by stacking more and more convolutional layers on top of each other, which allows the network to learn hierarchical representations of the input image [1].

Voice-activated technology is gaining significant traction as an effective means of interacting with machines and mobile devices using spoken commands. Google has developed one of the most advanced real-time speech recognition technologies, enabling users to search for information by simply speaking. Most Android smartphones now come equipped with hands-free functionality, making it extremely convenient and effortless for users to perform tasks without having to type. Real-time speech recognition technology can run smoothly on a range of devices, including smartphones, tablets, and small gadgets. Hidden Markov models (HMMs) and Gaussian mixture models (GMMs) are two popular techniques used to recognize speech with high levels of accuracy [24]. Now days, from deep learning method found promising result as well accuracy in less time. [24, 31].

The goal of our project is to utilize cutting-edge techniques in speech processing, such as Convolutional Neural Network (CNN) architecture, to detect various predetermined speech commands in real-time applications [25]. We have employed ADAM optimization and GPU acceleration to improve the performance of our model, which was trained using a massive dataset of 105,000 wave audio files provided by Google Tensor Flow and AIY team [3]. This dataset, comprising 11 distinct classes, has not been previously used in speech recognition research [6]. Our experimentations on GPU have enabled us to implement this large dataset recognition and increase our model's accuracy significantly.

The ability to detect and respond to students' emotions in real-time can be crucial in creating a positive learning environment and helping students stay engaged and motivated. Automatic recognition of student emotions can also provide valuable feedback to teachers, enabling them to tailor their teaching methods to meet the individual needs of each student [2].

The goal of histogram distance metric learning is to find a distance metric that can accurately distinguish between different distributions. This is achieved by first representing the distributions as histograms and then learning a distance metric that minimizes the distance between histograms of similar distributions while maximizing the distance between histograms of dissimilar distributions [26].

Our paper presents a three-stage approach, which involves dividing the data into Training, Validation, and Testing sets, followed by the computation of Speech Spectrograms through various phases. The spectrograms are then converted to log-Mel spectrograms to facilitate efficient training of a convolutional neural network. This conversion is performed for all the datasets, including the training, validation, and test sets. To ensure a smoother distribution, we take the logarithm of the spectrograms. Finally, we evaluate the system's performance and predict unknown speech words using a Confusion Matrix.

In Section 2, we present a comprehensive review of the existing literature on speech classification. In Section 3, we elaborate on the preprocessing and spectrogram details used in our proposed work. The algorithms used in the implementation of the proposed work are specified in Section 4. Finally, in Section 5, we discuss the results of our experiments and their implications.

2. Review work

Transfer learning is a popular technique in machine learning where a model trained on one task is used as a starting point for training a model on a different but related task. Transfer learning can save time and resources by leveraging the pre-trained model's knowledge, especially when the new task has a limited amount of training data [4].

The production of speech sounds involves the movement of air through the vocal tract, which includes the lips, tongue, teeth, and vocal cords. These movements produce a range of sounds that are perceived as speech. However, there are several sources of variability that can affect the recognition of speech sounds: Speaker Variability: The same word can be pronounced differently by different speakers, due to differences in accent, dialect, voice quality, and speech rate. This makes it challenging for ASR systems to accurately recognize spoken words. Environmental Variability: The acoustic properties of speech can be affected by the surrounding environment, such as background noise, reverberation, and other sounds. This can make it difficult for ASR systems to separate the speech signal from the background noise. Contextual Variability: The meaning of a word can be influenced by the words that come before or after it in a sentence. This is known as contextual variability, and it can make it challenging for ASR systems to accurately recognize spoken words in context [30].

In the context of speech recognition, HMMs are used to model the acoustic properties of speech sounds, such as the spectral and temporal characteristics of the speech signal. The HMMs are trained on large datasets of speech recordings, and they learn to recognize patterns in the speech sounds [29].

Neural networks can be used as acoustic models for HMM-based speech recognition, and this approach was first introduced more than 20 years ago. This approach, known as the hybrid approach, combines the strengths of HMMs and neural networks, and has been shown to achieve state-of-the-art performance on several speech recognition benchmarks [11, 12]. Key-word spotting (KWS) is a technique used in speech recognition to detect specific keywords or phrases in a stream of speech. KWS is often used in voice assistants, such as Amazon Alexa or Google Home, to trigger a particular action or command when a user says a specific keyword or phrase.

The KWS technique involves detecting a specific keyword or phrase within a stream of speech, which can then be used to trigger a particular action or command. KWS systems typically use a combination of acoustic models and language models to recognize the target keyword or phrase [28].

In addition to recognizing specific keywords, KWS systems may also detect filler words, such as "um" or "ah," that are commonly used in natural speech. Filler word detection can be useful for improving the accuracy and naturalness of speech recognition systems [13, 14, 15, 16].

LSTM networks are a type of RNN that are specifically designed to handle long-term dependencies in the input data. LSTMs use a memory cell and several gates to control the flow of information within the network, allowing them to selectively remember or forget information from previous inputs [27].

By using neural networks with memory capabilities, speech recognition systems can more accurately capture the complex temporal relationships between different parts of a speech signal. This can lead to improved recognition accuracy, especially for more complex speech tasks, such as continuous speech recognition [31, 17, 18]. While the HMM-based approach has been widely used in speech recognition, it can be computationally expensive, especially for large datasets or complex speech tasks. The Viterbi decoding algorithm can require a significant amount of computation to find the most likely sequence of hidden states.

The 2020 Data Science Bowl challenged participants to map the ocean floor using sonar data collected by the National Oceanic and Atmospheric Administration (NOAA). The competition attracted over 2,000 participants, and the winning solution used a combination of deep learning models, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs) [15].

Histogram of Oriented Gradients (HOG). HOG is a popular feature descriptor used for object detection and recognition. It represents the local shape and gradient information of an image by dividing it into small rectangular regions called cells. The gradients are then computed for each pixel in the cell and are quantized into a set of discrete orientations. The histogram of these orientations is then computed for each cell, and the final feature vector is obtained by concatenating all the histograms [18].

A novel hybrid CNN-SVM model for the recognition of fMRI images has been proposed recently. This model combines the strengths of CNNs and SVMs to achieve superior performance in fMRI analysis. The model consists of two main parts: a CNN feature extractor and an SVM classifier.

The CNN feature extractor is trained to learn relevant features from the fMRI images. This process involves multiple convolutional and pooling layers that are optimized to extract the most discriminative features from the input images. The extracted features are then fed into the SVM classifier, which learns to classify the images into different classes based on the extracted features [23].

To address this issue, researchers have developed other methods for speech recognition that do not rely on the HMM-based approach. One such method is the large-margin formulation model, which is a discriminative model that directly learns the mapping from speech signals to text labels.The large-margin formulation model is based on support vector machines (SVMs), a popular machine learning algorithm. The model learns a hyperplane that separates the feature vectors for each label in the dataset. During testing, the model calculates the distance between the test feature vector and the hyperplane for each label, and selects the label with the maximum distance [19, 20, 21, 22].

Recently, deep neural networks (DNNs) have been applied to keyword spotting (KWS) systems with promising results. DNN-based KWS systems can be very accurate and less time-consuming than traditional HMM-based systems. Google's KWS system is an example of a successful DNN-based KWS system that has replaced the traditional HMM system [32].

In addition to DNNs, convolutional neural networks (CNNs) have also been shown to be effective for speech recognition and keyword spotting tasks. CNNs can capture local patterns and spectral features in the speech signal, and can be more robust to noise and other distortions than other neural network architectures.

Several studies have shown that CNN models can outperform DNN models in a variety of small and large vocabulary tasks. For example, a study by Abdel-Hamid et al. (2014) found that a CNN-based system outperformed a DNN-based system for large vocabulary speech recognition tasks [5, 10, 23].

In paper [3], the authors evaluated three different models, namely CNN, DNN, and Vanilla, for keyword spotting (KWS) commands using MFCC feature extraction. They found that the CNN model performed better than the other two models. However, the second convolutional layer of the CNN model used a large number of multiplies, which was identified as a drawback. Thus, the authors suggest that a new CNN-based model is required to reduce the number of multiplies while improving performance.

In paper [26], the authors proposed a fast and simple deep KWS system that was trained only on keyword (KW) data. They found that their proposed system outperformed other systems, but noted that their work was limited to small data.

In paper [5], the authors proposed a small footprint keyword spotting classification system that used limited multiplies of CNNs and DNNs. The system was designed to be efficient while still achieving high accuracy.

Here we can say that CNNs model cross over two model DNNs for KWS task for mainly two advantages. First, DNNs applies concept of column vector instead input topology. However, for input speech signals, the spectrum representations show very strong correlations in time and frequency. Due to this modeling local correlations with CNNs get improved and these cause much better performance than DNNs. Second, recognizing parameter is sharing quality of CNNs, CNNs support very few parameters compared to DNNs for the same task, means reducing memory footprint and computational requirement. Thus observation conclude for that CNNs have enhanced routine and summary model size over DNNs and is thus the state-of-the-art technique for KWS task.

3.Preprocessing And Generating Speech Spectrograms For A Dataset

A. Speech Dataset

Here are the steps you can follow:

1. **Loading the dataset**: Load the audio dataset which you want to preprocess and generate spectrograms for. You can use libraries such as librosa, soundfile, or scipy for loading the audio files.

2. **Preprocessing**: Preprocessing involves cleaning and processing the audio data to remove noise, normalize the amplitude, and adjust the sampling rate. You can use various preprocessing techniques such as noise reduction, bandpass filtering, and normalization to improve the quality of the audio data.

3. **Extracting Features**: Extracting features is an important step in preprocessing the audio data. You can extract features such as Mel Frequency Cepstral Coefficients (MFCCs), log Mel-spectrograms, or spectral features such as Spectral Centroid, Spectral Contrast, and Spectral Rolloff. These features capture important information about the audio signal and can be used for speech recognition, speaker identification, and other audio analysis tasks.

4. **Generating Spectrograms**: Spectrograms are visual representations of the frequency content of an audio signal over time. You can generate spectrograms from the preprocessed audio data using libraries such as matplotlib or librosa. You can adjust the spectrogram parameters such as the window size, overlap, and the number of frequency bins to generate spectrograms that capture the relevant information.

5. **Saving the Spectrograms**: Once you have generated the spectrograms, you can save them in a suitable format such as JPEG or PNG for further analysis or use in machine learning models.

In this research work, a deep learning model was developed to detect the presence of specific words in audio files using the Speech Dataset developed by Google. The dataset contained 105,000 wave audio files, which were split into Training, Validation, and Test Sets. The dataset consisted of 11 different labels, including "Down," "Go," "Left," "No," "Off," "On," "Right," "Stop," "Up," "Yes," and "Unknown" keywords. Each label was considered for a count of 2300 to 2400 times, while the "Unknown" keyword had a count of around 8193.

To prepare the data for efficient training of a convolutional neural network, the speech waveforms were converted to log-Mel spectrograms. The log-Mel spectrograms were computed for the training, validation, and test sets, and the logarithm of the spectrograms was taken to obtain data with a smoother distribution.

To make the system more robust and able to work in noisy environments, the original signal was supplemented with robust noise, which was proportional to the training set. The noisy environment audio signal was then used for recognition, and a CNN was used along with Adam optimization techniques to achieve accurate results. Throughout the system, supplementary noise of volume and frequency was inserted in proper ratio to make the system more effective in noisy environments.

Finally, the pre-processing network was designed to not only recognize different spoken words but also to detect if the input contained silence or background noise, making the system more robust and effective in real-world scenarios.

To prevent co-adaptation, a technique called dropout can be used. Dropout involves randomly dropping out (i.e., setting to zero) some of the activations in a layer during training. This forces the network to learn more robust features that are less dependent on any single feature detector. Dropout has been shown to be an effective regularization technique that can improve the generalization performance of neural networks [6-new to 10].

Another technique that can help prevent co-adaptation is weight decay or L2 regularization. Weight decay involves adding a penalty term to the loss function that encourages the weights of the network to be small. This can prevent the network from overfitting by limiting the complexity of the model and encouraging it to learn simpler, more generalizable features.

B. VISUAL REPRESENTATION OF THE FREQUENCY USING SPECTROGRAM

In order to efficiently train a convolutional neural network, it is common to convert speech waveforms to log-Mel spectrograms. These spectrograms provide a visual representation of how the frequency content of the speech signal changes over time. The spectrograms are computed using a mathematical transformation called a Fourier transform, which breaks down the speech signal into its component frequencies. The resulting spectrogram is a two-dimensional image that displays frequency on the y-axis, time on the x-axis, and amplitude or energy as color or shading.

In this paper, the authors perform three different phases of training, validation, and testing using the log-Mel spectrograms as input to output classes. The logarithm of the spectrograms

is taken to obtain data with a smoother distribution. To illustrate the process, the authors plotted the waveforms and corresponding spectrograms of a few training samples based on the recorded input signal. The spectrogram was plotted for the actual dataset used in the paper, allowing for a visual representation of how the frequency content of the speech signal changes over time. This approach can help improve the accuracy and efficiency of training a convolutional neural network for speech recognition tasks.

The spectrogram and playing of particular audio clips file for few dataset has been shown on sample basis as example shown in figure 2.1.

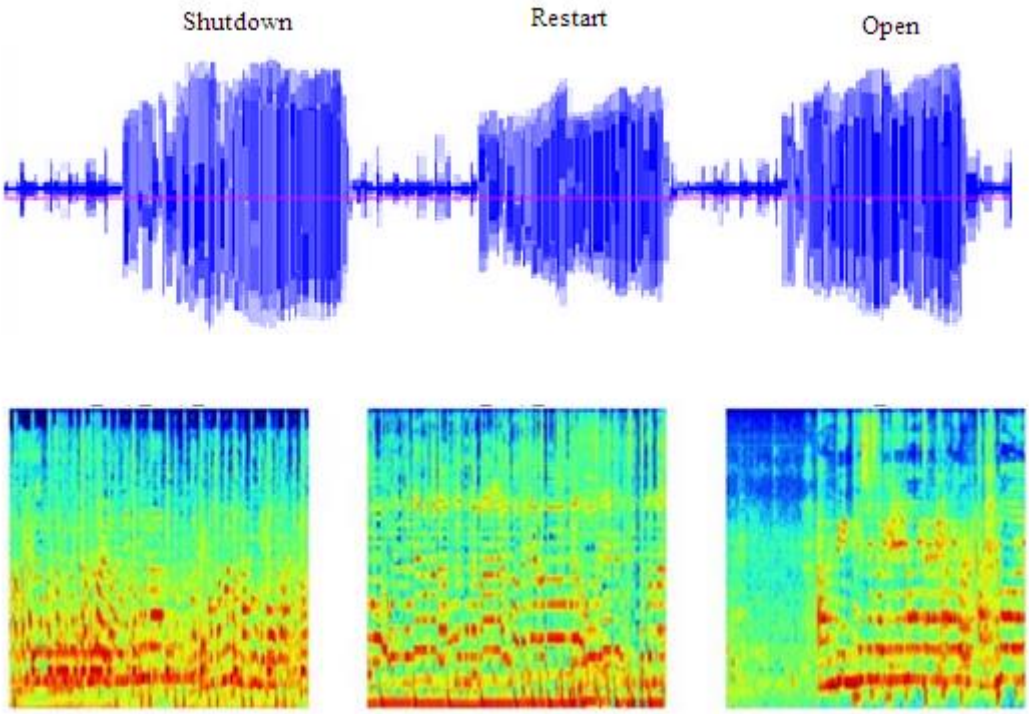

Figure2: Real Time Speech input for computer system and its spectrogram

4.PROPOSED METHODS AND ALGORITHMS

Some commonly used methods and algorithms in various fields:

I.Machine Learning:

- Supervised Learning: This is the most common type of machine learning where the algorithm is trained on a labeled dataset and then used to predict the label for new data.
- Unsupervised Learning: This is where the algorithm learns patterns and relationships in the data without any labeled data.
- Reinforcement Learning: This type of learning is used in scenarios where an agent learns how to take actions to maximize a reward signal in an environment.

- Deep Learning: This is a type of machine learning that uses neural networks with multiple layers to learn complex patterns in data.

II.Optimization:

- Gradient Descent: This is an iterative optimization algorithm that finds the minimum of a function by moving in the direction of steepest descent.
- Newton's Method: This is another optimization algorithm that finds the minimum of a function by iteratively updating the solution using the function's Hessian matrix.
- Genetic Algorithm: This is a metaheuristic optimization algorithm inspired by the process of natural selection that uses crossover and mutation to evolve a population of potential solutions to a problem.

III.Computer Vision:

- Convolutional Neural Networks (CNNs): These are a type of deep neural network that are specifically designed for image classification and object detection tasks.
- Haar Cascades: This is a machine learning-based approach for object detection that uses features extracted from an image to identify objects.
- Optical Flow: This is a computer vision technique that tracks the movement of objects in a video by analyzing the changes in pixel intensity over time.

IV.Natural Language Processing:

- Bag-of-Words: This is a simple and commonly used technique in NLP that represents a text document as a bag of its individual words, ignoring grammar and word order.
- Recurrent Neural Networks (RNNs): These are a type of neural network that are designed for sequential data, such as text or speech, by using feedback loops to incorporate past information into the model.
- Transformer Networks: These are a type of neural network architecture that uses self-attention mechanisms to process sequential data, such as text or speech, by attending to different parts of the input sequence.

The proposed method uses Hybrid Approach of SVM, CNN and Adam Optimization algorithm [5] such type of approach required high performance system so in our implementation we try to give attention on time complexity of algorithm with GPU system. The algorithm use the hybrid approach in such a way that if binary data base input applied for the training and testing then SVM system work well with higher accuracy and if general input speech dataset use then system going to refer for CNN and Adam Optimization algorithm. The GPU is the high processing unit on which training of sample achieved in less time and its support to improve our system for training and testing. Graphics processing unit (GPU) is belongs to NVDIA Tesla K40 C GPU.

A. *DATA REPRESENTATION BY USING TRAINING AND ITS EVALUATION*

Training and Validation of data is a crucial part of speech system based on this output system perform or giving the output to from system. Few of the concept of training and validation of data done based on labeling concept and it is given in detail as follow.

I. Training Data: Training data is a set of examples that are used to train a machine learning model. This data is labeled, which means that each example has a target or output value that the model is trained to predict. The more diverse and representative the training data is, the better the model will perform on new, unseen data.

II. Validation Data: Validation data is a set of examples that are used to evaluate the performance of a machine learning model during training. The model is trained on the training data, and then its performance is measured on the validation data. The performance metrics used to evaluate the model depend on the type of problem being solved, but typically include accuracy, precision, recall, and F1 score.

III. Cross-Validation: Cross-validation is a technique used to evaluate the performance of a machine learning model by splitting the data into multiple parts or folds. The model is trained on one part of the data and evaluated on the other part. This process is repeated multiple times, with each part of the data serving as the validation set. The results are then averaged to give an overall estimate of the model's performance.

IV. Overfitting and Underfitting: Overfitting occurs when a model is too complex and fits the training data too closely, resulting in poor performance on new, unseen data. Underfitting occurs when a model is too simple and cannot capture the complexity of the data, also resulting in poor performance. To prevent overfitting, regularization techniques such as L1 and L2 regularization can be used. To prevent underfitting, more complex models can be used or the training data can be augmented.

V. Hyperparameter Tuning: Hyperparameters are parameters that are set before training the model, such as the learning rate, batch size, and number of epochs. These hyperparameters can have a significant impact on the performance of the model, and their optimal values can be found through a process called hyperparameter tuning. This involves trying different values for the hyperparameters and evaluating the model's performance on the validation data.

The process of preparing for speech recognition by processing and analyzing input speech wave files can be referred to as "data preprocessing" or "data preparation". The creation of trained samples from this processed data can be described as "sample generation" or "model training". The act of using a GPU machine to speed up the training process can be called "GPU acceleration" or "parallel computing". Instead of referring to "extracting features of input speech signal", you can say "feature extraction". Finally, when discussing the smoothness of the input data distribution, you can use the phrase "data uniformity" or "data consistency".

The existing method and proposed hybrid method suggested about 11 keyword classes from existing methods and real time dataset of 10 different classes the system is check for offline as well for online phase.

The GPU machine that we are using for our current work has proven to be a game-changer in terms of speed and efficiency. With its tremendous processing power, we are now able to train our neural networks on large datasets in a fraction of the time it used to take. This has allowed us to process a much greater amount of data and achieve more accurate results.

To extract features from the input speech signal, we have taken a large number of training samples. These samples are then fed into our CNN for training. However, it is important to note that CNN training is most effective when the inputs to the neural network have a smooth distribution and are normalized. Therefore, we have taken extra care to ensure that our data is properly normalized and has a smooth distribution.

To further ensure the smoothness of our data distribution, we have plotted histograms of the values of our training samples. This has allowed us to visualize the distribution of our data and make any necessary adjustments to ensure that it is as smooth as possible.

Various data label along with exact count extracted from Google dataset for training are shown in table 1.

The given dataset applied only for CNN for training the same dataset applied for SVM system also if given dataset is suitable for binary classification then SVM system is use in our implementation.

Table 1: Dataset for Training applied to hybrid Method of CNN and SVM.

Sr. No	**Label**	**Count**
1.	Down	2359
2.	Go	2372
3.	Left	2353
4.	No	2375
5.	Off	2357
6.	On	2367
7.	Right	2367
8.	Stop	2380
9.	Unknown	8193
10.	Up	2375
11.	Yes	2377

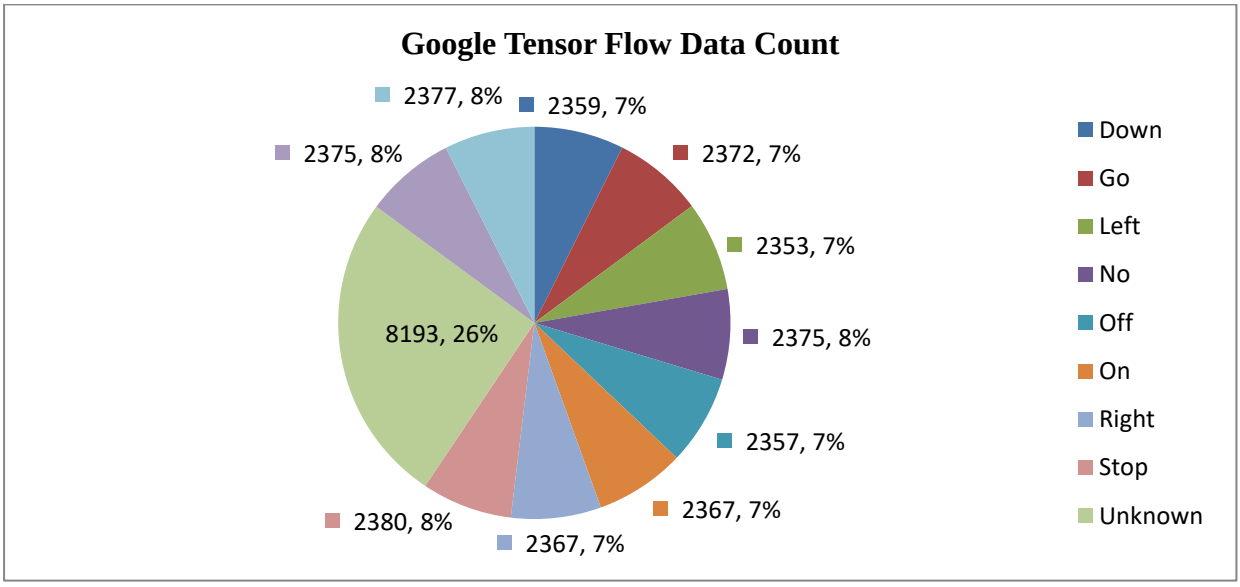

Figure3: Graphical Representation of Training classes with number of count.

Our proposed method uses the real time application for that use machine operating command and it's been consider as classifier for training and testing phase. These types of classifiers are applied for hybrid method of SVM and CNN. The lists of classifiers are considered as 10. Each different label is recorded and taken count of each is shown in Table-2.

Table 2: Different dataset for Training applied to hybrid Method of CNN and SVM.

Sr. no.	**Label**	**Count**
1.	On_	2200
2.	Off_	1820
3.	Login_	2002
4.	Logoff_	1560
5.	Open_	2030
6.	Close_	1756
7.	Save_	2456
8.	Shutdown_	1257
9.	Start_	2345
10.	Delete_	1264

The data label count with graphical representation is shown as follow.

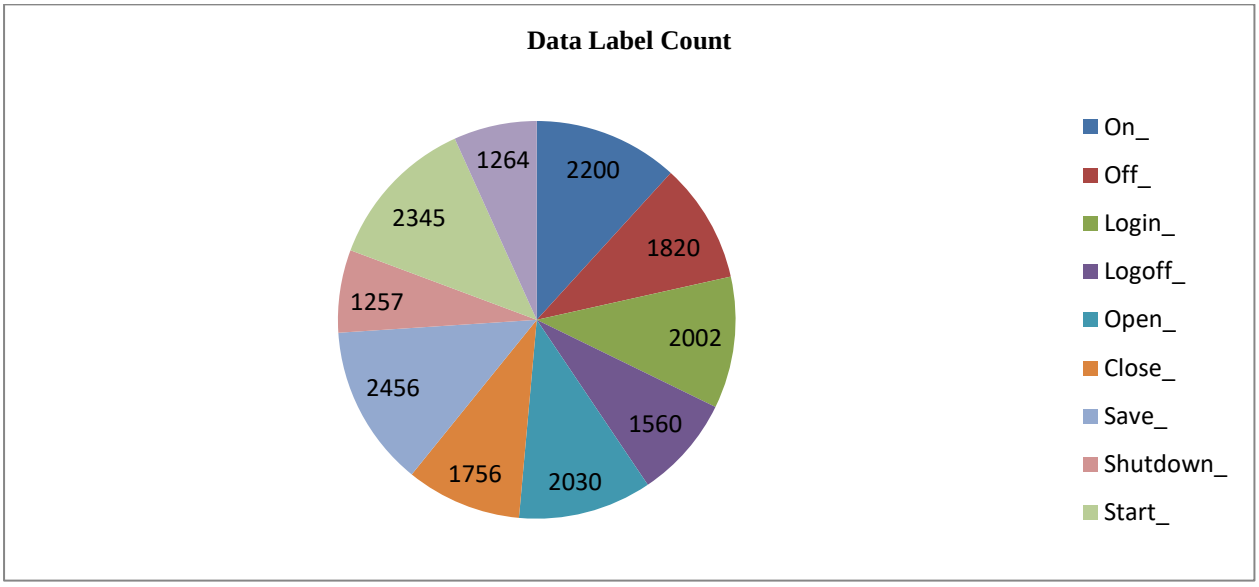

Figure4: Graphical Representation of Training classes with number of count.

Plot the distribution of the different class labels in the training and validation sets. The test set has a very similar distribution to the validation set.

During the training process, an augmented data store is generated to enable automatic data augmentation. This involves resizing spectrograms in the training dataset and randomly translating them up to 100 ms of 10 frames forwards or backwards in time. After the translation, the spectrograms are scaled along the time axis by 20 percent, which can enhance the efficacy of the training data and prevent over fitting of the network. Additionally, an augmented image data store is created during training to produce augmented images in real-time, which are then inputted into the network. No augmented spectrograms are saved in memory. So we are using an Adam optimization algorithm, which is used for training of CNN. Adam optimization is computationally efficient with less memory requirement than other optimization algorithms. This optimization is best suited for problems that are huge in terms of data along with parameters. It is also suitable for problems with very noisy/or sparse gradients. So Adam optimization gives more efficiency in training CNN. Hyper-parameters have intuitive interpretation and typically require little tuning. Training and validation of label with respective to its distribution is shown in fig.4. All Experiments are carried out on CPU or GPU (NVDIA Tesla K40 C GPU).

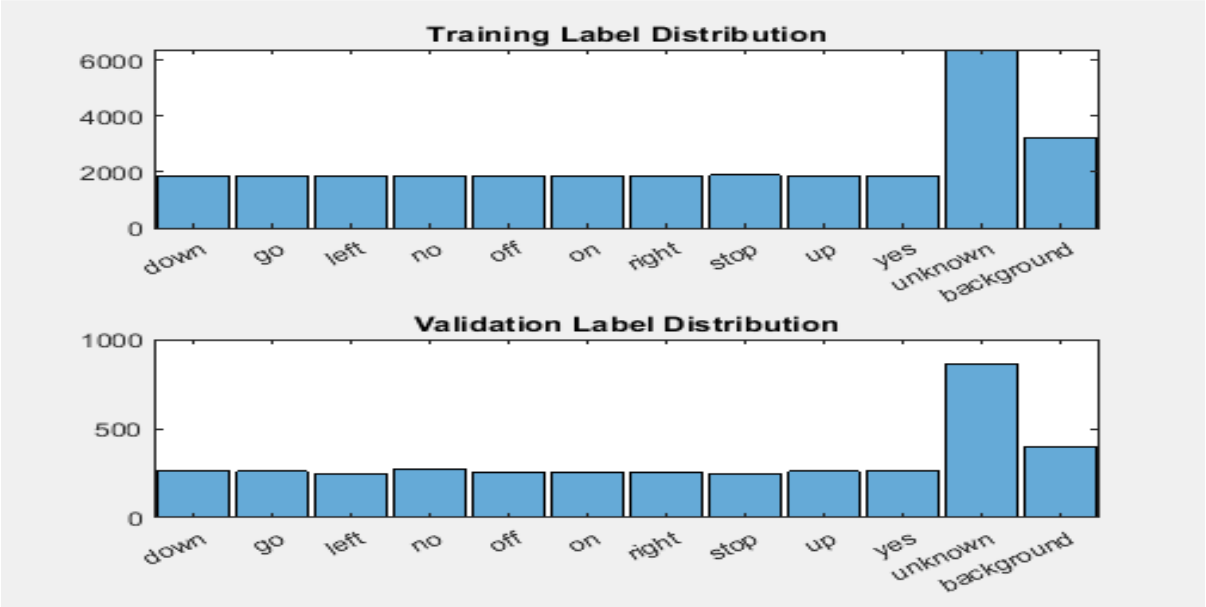

Figure: 4 Existing training and validation label of classes distribution in CNN System

B. NEURAL NETWORK ARCHITECTURE

Typical neural network architecture consists of several layers of interconnected nodes, or neurons, which are organized into input, hidden, and output layers. The input layer receives the raw data or features, the hidden layers process and transforms the data through a series of mathematical operations, and the output layer produces the final predictions or classifications.

Some common neural network architectures include:

1. Feed forward neural networks - also known as multilayer perceptrons (MLPs), these are the simplest type of neural network and consist of one input layer, one or more hidden layers, and one output layer.
2. Convolutional neural networks (CNNs) - these are designed for image and video processing and use convolutional layers to extract features from the input data.
3. Recurrent neural networks (RNNs) - these are designed for processing sequential data, such as speech and text, and use feedback loops to capture temporal dependencies.
4. Long short-term memory (LSTM) networks - a type of RNN that can retain information over long periods of time, making them well-suited for tasks that involve predicting sequences of events.
5. Autoencoders - a type of neural network used for unsupervised learning, where the goal is to reconstruct the input data from a compressed representation learned by the network.

6. Generative adversarial networks (GANs) - a type of neural network architecture that consists of two networks, a generator and a discriminator, which are trained together to generate new data that is similar to a training dataset.

To prevent over fitting and promote generalization, a dropout regularization technique can be applied to the last fully connected layer of the CNN architecture. The network comprises five convolutional layers with a limited number of filters to minimize the risk of memorizing specific features of the training data. Sequence of number of layers proposed in CNN are been given as follow [33].

a. Data Input Layer
b. Convolution 2d Layer
c. Batch Normalization Layer
d. Relu Layer
e. maxPooling2dLayer
f. convolution2dLayer
g. Batch Normalization Layer
h. Relu Layer
i. maxPooling2dLayer
j. convolution2dLayer
k. Batch Normalization Layer
l. ReluLayer
m. maxPooling2dLayer
n. convolution2dLayer
o. batch Normalization Layer
p. Relu Layer
q. convolution2dLayer
r. batch Normalization Layer
s. Relu Layer
t. maxPooling2dLayer
u. dropout Layer
v. fully Connected Layer
w. Softmax Layer
x. weighted Classification Layer

C. TESTING PHASE OF DATA USING NETWORK

This is the third phase of real time speech recognition system, which proves with better and very efficient results. In this step calculate the final accuracy of the network on the training set (without data augmentation) and validation set. The network is very accurate on this data set. However, the training, validation, and test data all have similar distributions that do not necessarily reflect real-world environments. This limitation particularly applies to the unknown category also, which contains utterances of only a small number of words with the counter of 8193.

We are calculating confusion matrix for evaluation of system and prediction of the unknown speech words. Plot the confusion matrix. Display the precision and recall for each class by using column and row summaries. Sort the classes of the confusion matrix. For classification linear discriminant analysis algorithm is used. Our system work for both recorded and real time data also. Detect spoken words using live audio from user. We tested our newly trained speech command detection network on streaming audio from microphone. Try saying one of the known commands. Then, try saying one of the unknown words. Specify the audio sampling rate and classification rate in Hz and create an audio device reader that can read audio from your microphone.

5.RESULTS

The result has been displayed over here for SVM system and CNN system separately as well in combination for training as well for testing. For our system we have tried huge dada set which is Google data set name as tenser-flow and created data set by human for real time application with replica of each has been mention in table-2. At firstly due to huge amount of data, we choose GPU to train our network. We have referred the training data from given table 1. Training, validation and prediction error of 11 key datasets of proposed system are shown in table 3. Based on the information presented in the table-1, it can be inferred that utilizing Google Tensor Flow data and training Convolutional Neural Networks (CNN) with Adam optimization on a Graphics Processing Unit (GPU) results in improved performance and reduced errors during both training and testing. Additionally, our proposed method has a significant advantage over previous research in terms of the time required for predicting test data in speech recognition tasks. Specifically, our method achieves a validation and testing time of only 4.51ms for large volumes of data, which is a substantial improvement. Therefore, we can conclude that the utilization of Google Tensor Flow data, training CNN with Adam optimization on GPU, and our proposed method for predicting test data can lead to significant improvements in speech recognition tasks.

Table 3: Training, Validation and Predictions.

Sr. No.	**Working Phase**	**Values**
1	Training error	15.660%
2	Validation error	16.640%
3	Prediction Time on CPU	3.0006 ms

To elaborate results of the proposed system we plotted the bar chart representation and it is been calculated for evaluation of system and prediction of the unknown speech words. The bar graph is shown for individual system as well for hybrid system of SVM and CNN. is shown in table 3. Improved result about 94.9% from the existing system in fraction but improves in the training by reducing training time. The current system plot confusion matrix by considering precision and recall factor and result gives as.

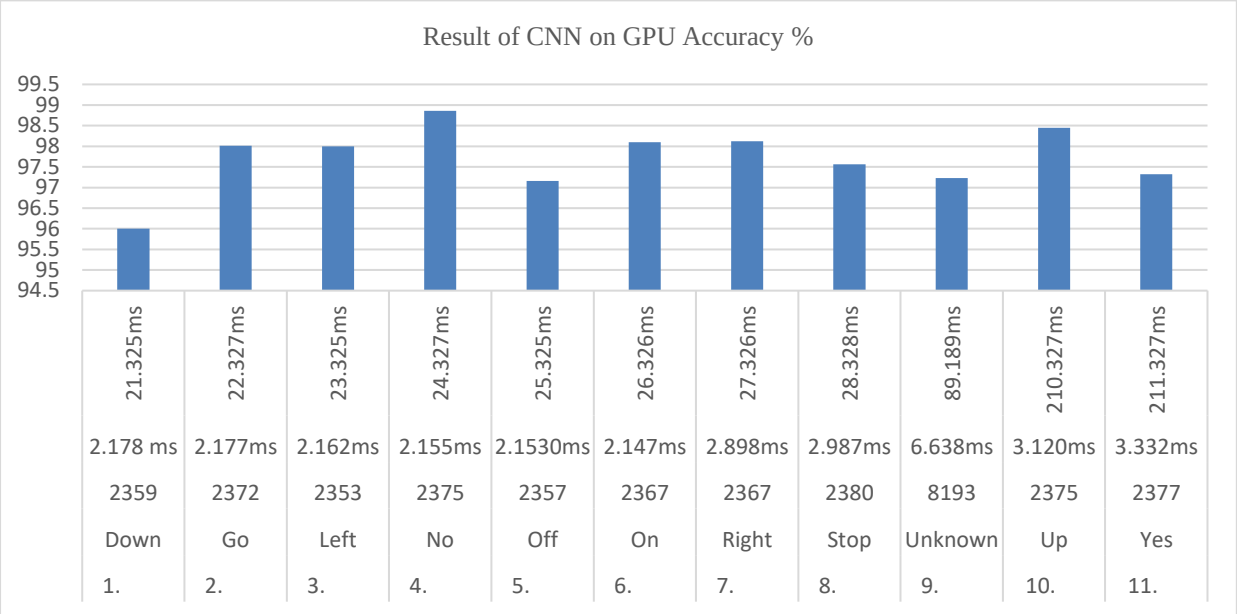

Figure: 5. Result of Tensor Flow using CNN and Adam Optimization techniques on GPU.

Here in this system the real time dataset is use. The number of counts is given in the table-2 are converted in proportion of 100 count and result has been calculated for the both systems separately for different dataset is shown in figure-6.

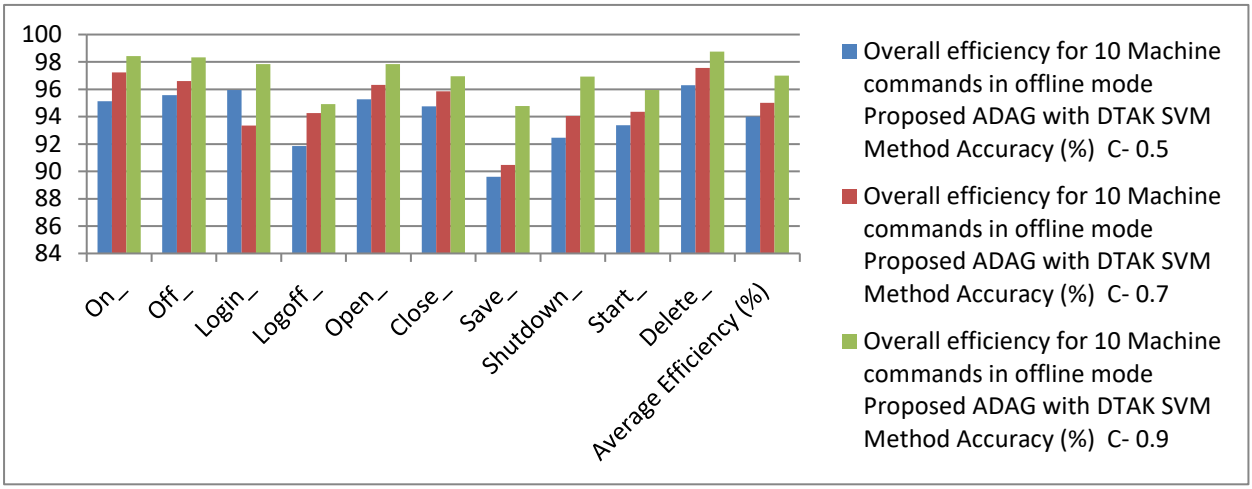

Figure: 6. Real time dataset applied for SVM system and overall efficiency of 10 classifiers on GPU.

Important benefits of using Adam optimization on other optimization algorithms using CNN and SVM system i.e hybrid system perform outstanding performance. The training part not matter most in term of time complexity but testing result gives prominent accuracy with the help of GPU. The resultant table-4 of hybrid system is given as follow. So, the applicability of GPU training and testing perform result is prominent than existing system.

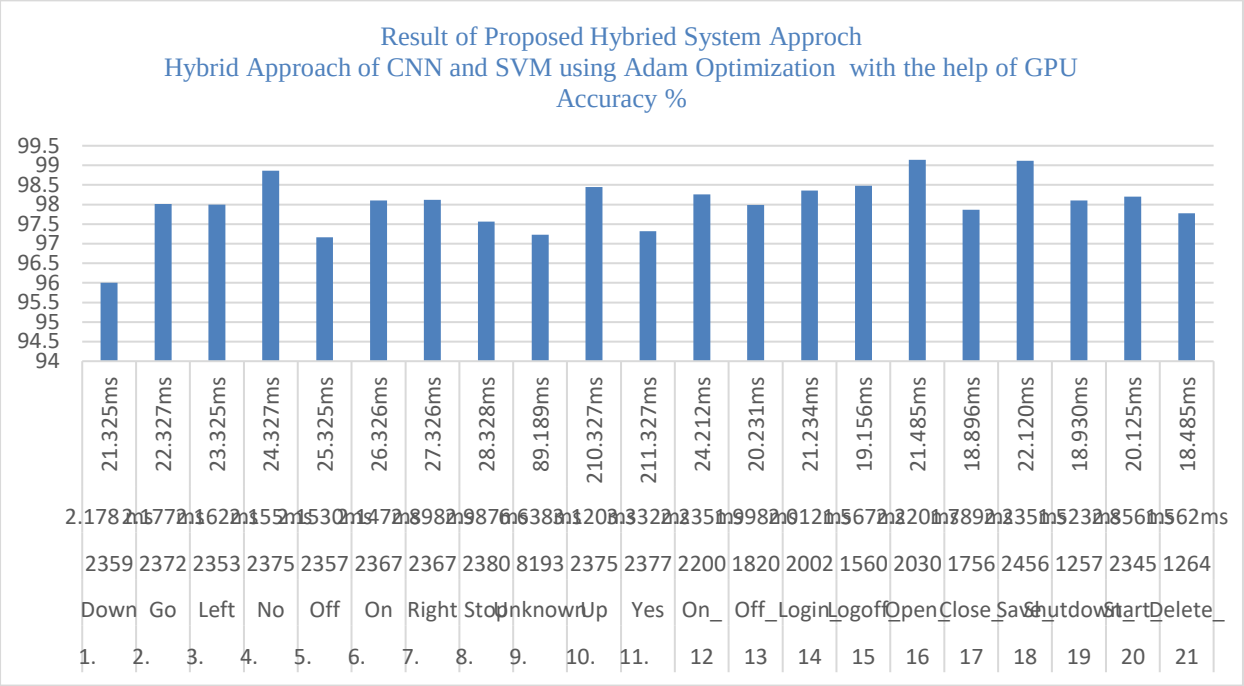

Figure: 7. Hybrid Approach of CNN and SVM using Adam Optimization with the help of GPU

The proposed system shows in figure-7 gives result for combination of both dataset i.e Google tensor flow and Real time dataset created by author. The SVM and CNN system working with help of Adam optimization on GPU shows better result than existing system.

6. CONCLUSION

The overall system implies that hybrid approach everywhere plays important role to to give better accuracy and efficiency. The proposed work improves results by all angles of Training, validation and Testing. The variation in training time complexity is slightly less than the existing system but testing accuracy overcome the previous system. Even we have referred huge Google Speech dataset; it contains 105,000 wave audio files and extracts log-Mel spectrogram for the same as well the author created data of real time application to interact with system which contain 18690 samples. In first phase Adam optimization algorithm is used for training of Convolutional Neural Network. Such training performed through high configures CPU or GPU (NVDIA Tesla K40 C GPU). The proposed system architecture uses SVM and CNN with five convolutional layers with few filters. Lastly, accuracy has been calculated for network on the training set which requires slight less time than existing method on both dataset. It takes only 4.5116 ms which outperforms all existing methods of speech recognition. Use of Adam optimization on other optimization algorithms and CNN with SVM makes computationally efficient. Hyper-parameters have intuitive interpretation and typically require little tuning. Due to all supporting algorithm CNN and SVM hybrid system plays important role to improve the performance testing result 98.005% better than existing system in fraction. By consideration of results, we conclude that our system is better for real time

applications for working on machine commands as well slandered data set by interfacing with hand held device with hands free communication.

RFERENCES

1. Simonyan K and Zisserman A 2014 Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
2. Tonguc¸ G and Ozkara B O 2020 Automatic recognition of student emotions from facial expressions during a lecture.", *Computers & Education.* 148: 103797.
3. Li, Xuejiao, and Zixuan Zhou , "Speech Command Recognition with Convolutional Neural Network." , *Semanticscholar* , 2017.
4. Sadeghi H and Raie A A 2019 Histogram distance metric learning for facial expression recognition.", *Journal of Visual Communication and Image Representation*. 62: 152–165.
5. T. Sainath, C. Parada, "Convolutional neural networks for small-footprint keyword spotting", *Proceedings Interspeech*, pp. 1478-1482, 2015.
6. Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization.", *arXiv preprint arXiv*:1412.6980 , 2014.
7. Gaikwad, Santosh K., Bharti W. Gawali, and Pravin Yannawar. "A review on speech recognition technique.", *International Journal of Computer Applications* 10.3 (2010): 16-24.
8. G. Hinton, Li Deng, Dong Yu, G.E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82 –97, nov.2012.
9. G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pretrained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. Audio Speech Lang. Processing*, vol. 20, no. 1, pp. 30–42, Jan. 2012.
10. Alex Graves, Abdel-rahman Mohamed and Geoffrey Hinton, "Speech Recognition With Deep Recurrent Neural Networks," *IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada*, Print ISSN: 1520-6149 Electronic ISSN: 2379-190X DOI: 10.1109/ICASSP.2013.6638947, May 2013.
11. Herve A Bourlard and Nelson Morgan. "Connectionist speech recognition: a hybrid approach" *Springer Science & Business Media*, volume 247, 2012.
12. McClelland, James L., and Jeffrey L. Elman. " The TRACE model of speech perception." *Cognitive psychology* 18.1 (1986): 1-86.
13. Rohlicek, J. Robin, et al. "Continuous hidden Markov modeling for speaker-independent word spotting." *International Conference on Acoustics, Speech, and Signal Processing*,. IEEE, 1989.
14. Rose, Richard C., and Douglas B. Paul. "A hidden Markov model based keyword recognition system." *International Conference on Acoustics, Speech, and Signal Processing. IEEE*, 1990.

15. Goodfellow I J, Erhan D, Carrier P L, Courville A, Mirza M, Hamner B, Cukierski W, Tang Y, Thaler D, Lee D H and Zhou Y 2013 "Challenges in representation learning: A report on three machine learning contests.", *International conference on neural information processing, Springer, Berlin, Heidel-berg.* 117-124
16. Silaghi, Marius-Calin, and Hervé Bourlard. "Iterative Posterior-Based Keyword Spotting Without Filler Models: Iterative Viterbi Decoding and One-Pass Approach." *Tech. Rep.* , 2000.
17. LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." *nature* 521.7553 (2015): 436.
18. Kumar P, Happy S L and Routray A 2016 A real-time robust facial expression recognition system using HOG features. In: 2016 International Conference on Computing, Analytics and Security Trends (CAST), IEEE. 289-293
19. David Grangier, Joseph Keshet, and Samy Bengio, "Discriminative keyword spotting," Automatic speech and speaker recognition." *Large margin and kernel methods,* pp. 175–194, 2009.
20. Tabibian, Shima, Ahmad Akbari, and Babak Nasersharif. "An evolutionary based discriminative system for keyword spotting." *International Symposium on Artificial Intelligence and Signal Processing (AISP)*. IEEE, 2011.
21. KP Li, JA Naylor, and ML Rossen, "A whole word recurrent neural network for keyword spotting," *in Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1992, vol. 2, pp. 81–84.
22. Santiago Fernández, Alex Graves, and Jürgen Schmidhuber, "An application of recurrent neural networks to discriminative keyword is spotting," *in Artificial Neural Networks–ICANN 2007*, pp. 220–229. Springer, 2007.
23. X. Sun, J. Park, K. Kang, and J. Hur, ''Novel hybrid CNN-SVM model for recognition of functional magnetic resonance images,'' in Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC), Oct. 2017, pp. 1001–1006, doi: 10.1109/SMC.2017.8122741.
24. Dong Yu , Li Deng, "Automatic Speech Recognition: A Deep Learning Approach", Springer Publishing Company, Incorporated, 2014.
25. Solovyev, Roman A., et al., "Deep learning approaches for understanding simple speech commands." *arXiv preprint arXiv:*1810.02364, 2018.
26. G. Chen, C. Parada, and G. Heigold, "Small-footprint Keyword Spotting using Deep Neural Networks," *in Proceedings ICASSP*, 2014.
27. Pan S J and Yang Q 2009 A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering.* 22(10): 1345–1359.
28. Das H.S. and Roy P. 2019 "A deep dive into deep learning techniques for solving spoken language identification prob-

lems.", Intelligent Speech Signal Processing. *Academic Press. 81-100.*

29. Hinton G.E., Srivastava N., Krizhevsky A, Sutskever I, Salakhutdinov R R 2012 Improving neural networks by preventing co-adaptation of feature detectors.", *arXiv preprint. arXiv:*1207.0580.
30. Wilpon, J. G., L. G. Miller, and P. Modi. "Improvements and applications for key word recognition using hidden Markov modeling techniques." *International Conference on Acoustics, Speech, and Signal Processing. IEEE,* 1991.
31. Ian, Yoshua Bengio, and Aaron Courville. "Deep learning." *MIT press*, 2016.
32. L. Toth, "Combining Time-and Frequency-Domain Convolution in Convolutional Neural Network-Based Phone Recognition," *in Proc. ICASSP*, 2014.
33. Bhosale, R. S.; Chaudhari, N. S. Accelerating speech recognition system by Adam optimization and CNN for real time system using GPU. *Int J Control Autom*, 2019, 12.4: 11-19.