Expert Systems With Applications 168 (2021) 114416

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

FuzzyGCP: A deep learning architecture for automatic spoken language identification from speech signals

Avishek Garain [a], Pawan Kumar Singh [b,*], Ram Sarkar [a]

[a] Department of Computer Science and Engineering, Jadavpur University, Kolkata 700032, West Bengal, India
[b] Department of Information Technology, Jadavpur University, Kolkata 700106, West Bengal, India

ARTICLE INFO

Keywords:
Spoken language identification
Speech signal
Deep learning
GAN
DNN
MLP
Ensemble learning
Choquet integral
Spectrogram

ABSTRACT

In this modern era, language has no geographic boundary. Therefore, for developing an automated system for search engines using audio, tele-medicine, emergency service via phone etc., the first and foremost requirement is to identify the language. The fundamental difficulty of automatic speech recognition is that the speech signals vary significantly due to different speakers, speech variation, language variation, age and sex wise voice modulation variation, contents and acoustic conditions and so on. In this paper, we have proposed a deep learning based ensemble architecture, called FuzzyGCP, for spoken language identification from speech signals. This architecture combines the classification principles of a Deep Dumb Multi Layer Perceptron (DDMLP), Deep Convolutional Neural Network (DCNN) and Semi-supervised Generative Adversarial Network (SSGAN) to increase the precision to maximum and finally applies Ensemble learning using Choquet integral to predict the final output, i.e., the language class. We have evaluated our model on four standard benchmark datasets comprising of two Indic language datasets and two foreign language datasets. Irrespective of the languages, the F1-score of the proposed language identification model is as high as 98% in MaSS dataset and worst performance is that of 67% on the VoxForge dataset which is much better compared to maximum of 44% by state-of-the-art models on multi-class classification. The link to the source code of our model is available here.

1. Introduction

Automatic spoken language identification (SLID) refers to the process of identification of the spoken language through any computing devices using speech signals. Such a system can act as a support towards speech recognition purposes in multilingual countries. This can be accomplished by determining the language of the spoken segments with the help of language recognizers. However, the intermixing of various languages, having common origin, poses huge challenge for precise classification of languages in an automated manner when a multilingual dataset is taken into consideration. Thus to find out a feasible solution to such mixing of data and many other challenges, SLID has attracted many researchers around the world to work on this topic. If we consider the country India only, we can see that 23 languages are recognized for official use (Languages of India (2017)). The interdependency existing among the Indic languages poses a real challenge for automatic classification of languages from the voice signals. Besides, this field has wide spectrum of applications and shall continue gaining it in near future. Ranging from usage in speech based evolving search engines to segregate results based on automatic location identification from the spoken languages to usage in the telecommunication industries for delivering proper customer care services, where spoken languages play a key role, the applications of SLID are truly widespread. In the aviation industry, where pilots need to know a standard language for communication purposes, an efficient automatic spoken language recognition system can eliminate this dependency by transferring the control to proper language translation systems based on language of the queries by the pilots. Doctors from all over the world can communicate with each other freely while making use of tele-medicine, if their spoken languages can be correctly identified and put forward properly to translation systems. The rising trend of globalization and the increasing popularity of the Internet have amplified the need for the competent SLID systems. An important application arises in call centers across the world dealing with speakers of different languages. With the huge volume of vocabularies, for indexing such speech data archives or searching from the same, which contain multiple languages, SLID systems are gaining more and more importance in recent times. Therefore, a comprehensive system for SLID is a pressing need to address the above-mentioned needs.

In this paper we have presented an ensemble based architecture which leverages the functionalities of deep learning models that include

* Corresponding author.
E-mail addresses: avishekgarain@gmail.com (A. Garain), pawansingh.ju@gmail.com (P.K. Singh), ramjucse@gmail.com (R. Sarkar).

https://doi.org/10.1016/j.eswa.2020.114416
Received 3 June 2020; Received in revised form 18 October 2020; Accepted 28 November 2020
Available online 8 December 2020

Deep Dumb Multi Layer Perceptron (DDMLP), Deep Convolutional Neural Network (DCNN) and Semi-supervised Generative Adversarial Network (SSGAN) in solving the problem of SLID mainly for Indic languages as well as some popular foreign languages. We have discussed the underlying challenges, probable discriminatory features as well as provided a detailed analysis for the same.

The remaining paper has been organized as follows. Section 2 provides a brief explanation of some previous works and their performances. Section 3 describes the datasets on which the proposed framework has been evaluated. The methodology that has been followed in designing our architecture is described in Section 4. This is followed by the results and concluding remarks in Sections 5 and 6 respectively.

2. Literature survey

Previously research works have been carried out in this domain by mainly making use of feature based approaches like MFCC (Mel-frequency cepstral coefficients) (Logan et al., 2000), LPC (Linear Predictive Coding) (O'Shaughnessy, 1988), Gaussian Mixture Model (GMM), PLP 9 (Hermansky, 1990), PHCC (Perceptual Harmonic Cepstral Coefficients) (Gu & Rose, 2001), Mel Scale Cepstral Analysis (Imai, 1983), Power Spectral Analysis (Stoica et al., 2005), LFCC (Linear Frequency Cepstral Coefficient) (Zhou et al., 2011), RASTA (Relative Spectral Analysis Technique) (Hermansky & Morgan, 1994) and Shifted-delta features (Wang et al., 2012).

Li et al. (2013) have given an introductory note on the fundamentals of the theory and the solutions, from both computational and phonological aspects of spoken language recognition. They have also given a detailed and comprehensive review of current trends and future research directions using the language recognition evaluation (LRE) formulated by the National Institute of Standards and Technology (NIST).

Albadr et al. (2019) in their study have employed the extreme learning machine (ELM) as the learning model for the task of SLID using some standard features. In addition, the authors have proposed an optimized Genetic Algorithm (OGA) with three different selection criteria namely K-tournament, roulette wheel and random for selecting the most appropriate initial weights and biases of the input hidden layer of the ELM to minimize the classification error. The proposed OGA–ELM with three different selection criteria has produced the highest accuracies of 99.50%, 100% and 99.38%, respectively.

In the paper by Zhang Jian et al. (2017), the authors have used F-ratio analysis method for analyzing the importance of weightage that should be given to different SLID feature vectors. After this, a weighted phone log-likelihood ratio (WPLLR) feature has been used to weight those dimensions more heavily which are having high F-ratio values. The authors have tested on the NIST 2007 dataset. The results show the effectiveness of their feature, with relative improvements in terms of average cost and equal error rate compared with the phone log-likelihood ratio (PLLR) feature.

The authors Lee and Jang (2018) in their paper have presented an approach based on a perspective of linguistics, specifically that of syllable structure. Their approach contains a section for labeling common syllable structures. Then, the authors have made use of a long short-term memory (LSTM) network in order to transform the MFCC of an audio sample to its structure of syllable. They have applied their work on 10 different languages and have achieved an accuracy of 70.40%. Their results has outperformed most of the methods based on acoustic–phonetic and phonotactic features in terms of efficiency.

The authors Shukla et al. (2019) in their paper have focused on an approach which is implicit. This is due to the absence of data in transcriptive form. They have proposed a new model based on attention mechanism which makes use of log-Mel spectrogram images as input. For training and evaluating the models, they have considered six languages namely English, German, Spanish, French, Russian and Italian and obtained an accuracy of 95.4%.

After the evolution of deep learning and availability of computational resources specifically Graphical Processing Units (GPUs) at cheaper costs, the research has seen a paradigm shift.

In the paper by Miao et al. (2019), the authors have aimed to improve traditional DNN (Delay Neural Network) x-vector language identification (LID) performance by employing Convolutional and Long Short Term Memory-Recurrent (CLSTM) Neural Networks by harnessing their advantage to strengthen feature extraction and capture longer temporal dependencies. The authors have introduced a frequency attention mechanism to give different weights to different frequency bands to generate weighted means and standard deviations. They have shown that CLSTM can significantly outperform a traditional DNN x-vector implementation and the proposed frequency attention method has outperformed time attention, particularly when the number of frequency bands matches the feature size.

The authors Madhu et al. (2017) in their work have proposed a framework by using language dependent prosodic information and phonotactic features. It consists of a Phonetic Engine which serves the purpose of front end for the SLID system and converts the speech sample fed as input into a sequence of phonetic symbols. Thereafter, syllable boundaries are recognized and phones within a syllable boundary are divided into groups. Then rules which are phonotactic in nature are applied to get syllables. Numeric representation of successive pairs of syllables is done to get phonotactic feature vectors. Vectors for features which are prosodic in nature are obtained by concatenating feature vectors of three successive syllables. Then these features are fed to a multilayer feed forward neural network based classifier in the language identification process. The data on which the classifier is trained consist of speech samples with a total of two hours duration from each of the seven languages. The language classes that are targeted include Hindi, Bangla, Telugu, Assamese, Punjabi, Manipuri and Urdu.

The letter by Wang et al. (2013) presents a study of application of phoneme posterior features for spoken language recognition. In their work, they have estimated phoneme posterior features from Multi Layer Perceptron (MLP) based phoneme recognizer, and further processed them through transformations. These transformations include taking logarithm, Principal Component Analysis (PCA), and appending shifted delta coefficients. The authors have reported that the resulting shifted-delta MLP (SDMLP) features have shown similar distribution as conventional shifted-delta cepstral (SDC) features, and SDMLP features are more robust compared to the SDC features.

Ferrer et al. (2014) in their work have proposed a new approach for SLID based on the estimated posteriors for a set of senones which represent the phonetic space of one or more languages. For speech recognition systems, these senones usually are the Hidden Markov Model (HMM) states of the acoustic model, which can be predicted by a neural network, if a Delay Neural Network/HMM hybrid approach for acoustic modeling is applied. Then they have derived a feature vector for every sample using these probabilities. Their proposed system is reported to give over 40% relative gain compared to state-of-the-art language identification systems at sample duration ranging from 3 to 120 s.

In the paper by Miao et al. (2018), the authors have exploited the latent abilities of conditional Generative Adversarial Network (cGAN) to firstly combine them with DNN based i-vector approach. Then they have tried to improve the language identification model using cGAN architecture. First, they have extracted the deep bottleneck features (DBF) which are phoneme dependent. Then they have combined them with output posteriors of a pre-trained DNN. After that they have used them to extract i-vectors in the normal way. They have classified these i-vectors using cGAN. Results show that cGAN architecture can significantly outperform DBF, DNNs and i-vector methods where 49-dimensional i-vectors are used, but not where 600-dimensional i-vectors are used. In the work by Snyder et al. (2018a), the authors

have applied the concept of x-vectors for recognition of spoken language. Their framework consists of a DNN that maps sequences of speech features to fixed-dimensional embedding, called x-vectors. Long term language characteristics have been captured in the network by a temporal pooling layer that aggregates information across time. Once the x-vectors are extracted, they make use of the same classification methodology as developed for i-vectors.

Dehak et al. (2011) in their paper have presented a new SLID system based on the total variability approach. They have employed various techniques to extract the most salient features in the lower dimensional i-vector space. Additional performance gain has been observed when the system has been combined with other acoustic systems.

2.1. Research Gap & Motivation

In recent times, many approaches which are based on GMM have been developed for SLID purposes. Among all such methods, the method of modeling of i-vectors happens to be one of the best methods and results in significant improvement in performance over the others (Dehak et al., 2011). In i-vector model, features based on acoustics are firstly transformed into higher dimensional vectors. Then a mapping of these vectors is done into a low-dimensional subspace. Each speech sample is denoted by a vector of fixed length called the i-vector. After the i-vectors are completed extracting, standard techniques like Gaussian back-end and Logistic regression are applied to the i-vectors of the test samples. However, the performance of this method heavily depends on the choice of hyper-parameters and suffers from drastic decrease in performance if used for separate classes of languages other than classes used for training. In recent times, the DNN based models have been used predominantly for acoustic modeling in the field of speech recognition as a replacement of GMM. In SLID, investigation on several strategies using DNNs has been done so far, and the most successful approaches that have paved out their way are those frameworks which are built using hybrid techniques (Jog et al., 2018). These are the frameworks where DNNs are trained to differentiate between senones and are combined with conventional language identification models. However, recognition results seem to fail miserably when it comes to identification of Indic languages which have various commonalities among them. Also use of same model for both foreign and Indic languages may give unsatisfactory results (Anjana & Poorna, 2018).

Though a significant amount of work have been performed by the researchers, however, to the best of our knowledge application of a common model on the datasets with significant diversity has not been explored much. This is because same model may not give desirable accuracy across the datasets, hence consistent performance over the varied datasets is required to prove the robustness and versatility of a model. Evaluating a common model on bi-lingual, tri-lingual and multi-lingual scenarios are not investigated as such till now. However, this aspect of the SLID research becomes more pertinent for Indic languages owing to their common roots of development, which may not be valid for many foreign languages. This fact is evident from our experimental outcomes. We have evaluated our model in such scenarios and results have been analyzed, and it can be said that the results are quite satisfactory keeping in mind the complexity of problem under consideration. The use of Generative Adversarial Networks (GANs) and ensemble mechanisms in this domain so far has been limited and yet to be retrospected and worked upon.

2.2. Contributions

In the light of the above-mentioned facts, we have proposed a new SLID model. The highlights of this work are as follows:

1. We have used two types of features — one type being the numeric values, while the other being images obtained from their corresponding spectrograms.
2. We have used a conventional DDMLP architecture as a classifier for numeric features.
3. The image based features are used to train architectures like DCNN and SSGAN. Usage of SSGAN in this context has not yet been explored much, and may be first-of-its-kind.
4. Finally, to obtain results beyond the reach of each of the models if used separately, we have formed a heterogeneous ensemble, called FuzzyGCP, by combining the results of the aforementioned architectures using a fuzzy integral measure.
5. The datasets we have considered are itself diverse enough to prove the efficiency and robustness of our model. The results are quite impressive considering the multi-lingual classification approach and the domain.
6. We have shown a detailed analysis of bi-lingual, tri-lingual and multi-lingual classification capabilities of our model for the Indic datasets, and only multi-lingual classification capability for the foreign datasets.

3. Dataset used

There are more than 7000 languages spoken throughout the world. We have come across various spoken language datasets, however, we have selected the datasets which consist of speech signal of some popular languages. They are enlisted in the Annexure section (Table 16).

The datasets that we have selected for evaluating the performance of our model consist of both foreign and Indic languages. The datasets are diverse in terms of speakers, gender and ethnicity. Also, we have intentionally selected languages which tend to be similar in terms of semantics and phono-tactic features and show inter-dependency among themselves. This was done in order to put our model through intense training and confusions, to improve its precision and generalize it properly. The class balance for these datasets is approximately perfect preventing any kind of class-based biased training thus giving great recall metrics.

3.1. IIIT Hyderabad dataset

The IIIT Hyderabad Indic speech databases (Prahallad et al., 2012) consist of data in textual and speech format for the languages namely Bangla, Hindi, Tamil, Kannada, Telugu, Malayalam, and Marathi. The creators of these datasets selected these languages based on the fact that the total number of articles found in Wikipedia written using each of the said languages is more than 10,000. The languages considered here have different dialects. To maintain the originality, they decided recording of the speech to be done in the dialect in which the native speakers were comfortable with. The dataset consists of 7000 audio samples approximately equally divided among the 7 language classes.

3.2. IIT Madras dataset

This dataset is result of a project on developing text-to-speech (TTS) synthesis systems for Indian languages (Baby et al., 2016) as well as enhancing quality of synthesis. We have applied our model on 6000 audio samples, 1000 of each language class. The comprising languages are English, Marathi, Tamil, Bangla, Telugu and Hindi.

3.3. VoxForge dataset

VoxForge (Voxforge.org) is a project which was set up to collect transcribed speech for use in Open Source Speech Recognition Engines (“SRE”s) such as Julius, ISIP, HTK and Sphinx. This dataset is huge and diverse both in terms of variety and size. Here, we have considered 2000 audio samples for each language category, thus avoiding class imbalance of any kind. The selection of samples is such that total recording time is approximately same for all the language classes. The languages considered are French, German, Italian, Portuguese and Spanish.

3.4. MaSS dataset

MaSS (Multilingual corpus of Sentence-aligned Spoken utterances) dataset (Boito et al., 2020) is an extension to the CMU Wilderness Multilingual Speech dataset (Black, 2019). They have prepared this dataset by considering multilingual links between speech segments of different languages. It consists of a voluminous and clean dataset where we find 8130 parallel spoken utterances of 8 languages with 56 language pairs. The language categories are Basque, English, Finnish, French, Hungarian, Romanian, Russian and Spanish. The quality of the final corpus is attested by means of human evaluation performed on a corpus subset (8 language pairs with 100 utterances).

4. Methodology

Every audio signal considered here is sampled to 5 s duration with a sampling rate of 44.1 kHz to maintain the uniformity in feature extraction. Lesser duration like 1 s or 2 s would have led to increased localization of feature learning, thereby reducing the generalization in learning over the whole time series information of the signal. For making the audio signals machine readable, we have made use of the Librosa library (McFee et al., 2015). For implementing our architecture we have made use of libraries like Tensorflow (Abadi et al., 2015) and Keras (Chollet et al., 2015).

4.1. Feature extraction

In this section at first we will discuss the challenges related to inter-dependencies among the different Indic languages owing to the common origin and others, which make the feature extraction process extremely difficult. Then we will discuss different feature extraction processes applied here. The Indic languages namely Marathi, Hindi and Bangla belong to an Indo-Aryan language family. Marathi has its grammar and syntax derived from Pali and Prakrit. It uses the retroflex nasal sound /ɳ/ most frequently. Much of the vocabulary of the language Hindi have been derived from Sanskrit. In Hindi, distinction of length into long and short vowels has been neutralized. It is a language which is syllable-timed, meaning words are not distinguished based on stress alone. Default stress in Hindi is given on the last syllable. Each content word except the final one has rising contour. Bangla is also derived from Magadhi Prakrit and Pali, and it is also a bound stress language. In this language, voiced stops have shorter closure duration than voiceless stops and breathy voiced stops have the shortest closure duration (Berkson, 2013).

The Indic languages namely Tamil, Telugu, Malayalam and Kannada belong to the Dravidian language family, where Telugu belongs to the south central group and Malayalam, Kannada and Tamil belong to the southern group. Tamil neither consists aspirated nor voiced stop like other Indian languages and also aspirated consonant is absent in this language (Keane, 2004). Telugu is influenced by Sanskrit and Prakrit. It shows the vowel harmony phenomenon, which is not characteristic of any other Dravidian language. In this phenomenon, quality of a vowel in a syllable is decided by vowels of the preceding (Bhaskararao, 2011). Malayalam is thought to be a branch of classical Tamil but has a large contribution from Sanskrit vocabulary (Caldwell, 1875). Kannada is influenced by Prakrit, Sanskrit and Pali languages (mustgo.com).

Here, 8 set of features (see Fig. 1) are extracted from the audio signals to obtain useful information for the rest of the working pipeline. They are as follows:

1. **MFCCs:** MFCCs are the coefficients which are derived from a type of cepstral representation of the audio clip
2. **Spectral bandwidth:** Wavelength interval where a radiated spectral quantity is not less than half its maximum value
3. **Spectral contrast:** Mean of the level difference between peaks and valleys in the spectrum
4. **Spectral roll-off:** Frequency below which 85% of the distribution magnitude is concentrated
5. **Spectral flatness:** Determined by dividing geometric mean of the power spectrum by arithmetic mean of the power spectrum
6. **Spectral Centroid:** Indicates the location of the center of mass of the spectrum
7. **Polynomial features:** Coefficients of fitting an nth-order polynomial to the columns of a spectrogram
8. **Tonnetz:** Tonal centroid

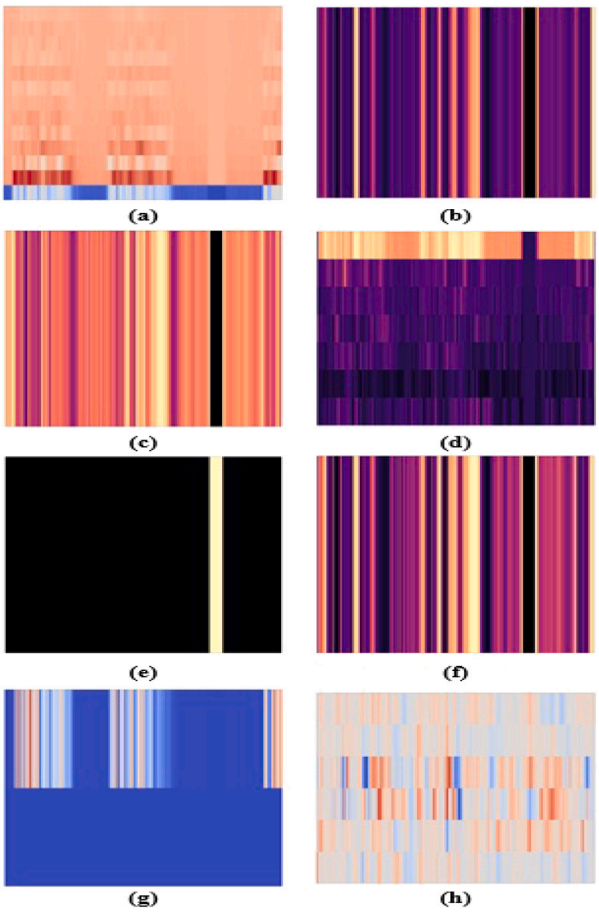

Fig. 1. Illustration of output images for a sample audio signal representing: (a) MFCCs, (b) Spectral bandwidth, (c) Spectral contrast, (d) Spectral roll-off, (e) Spectral flatness, (f) Spectral centroid, (g) Polynomial features and (h) Tonnetz.

The features mentioned above are first scaled using StandardScaler function of Scikitlearn library (Pedregosa et al., 2011) and then fed to the DDMLP classifier by averaging principle, the description of which is given as follows.

Let us consider a set of features $\mathbb{S} = \{S_1, S_2, \ldots, S_N\}$, where $S_i = \{s_1, s2, \ldots, s_M\}$. So any element located at ith row and jth column of the NxM dimensional feature array $\mathbb{S}$ can be denoted by $\mathbb{S}_{ij}$, where $1 \le i \le N$ and $1 \le j \le M$.

Let us denote the set of end features to be fed to the network by $\mathbb{F}$. Then any feature element $\mathbb{F}_j$ of the set of end features $\mathbb{F}$ is given by,

$$\mathbb{F}_\rfloor = \frac{\sum_{i=1}^{N} \mathbb{S}_{ij}}{N} \tag{1}$$

where $1 \le j \le M$.

This makes $\mathbb{F}$ to be a 1xM dimensional feature vector which is later processed and fed to the DDMLP classifier for the classification purpose. For the DCNN and SSGAN networks, however, the spectrogram based features are fed as these networks require image data to work upon. The individual spectrogram is first converted to grayscale image. These images are then concatenated together to form a single image as shown

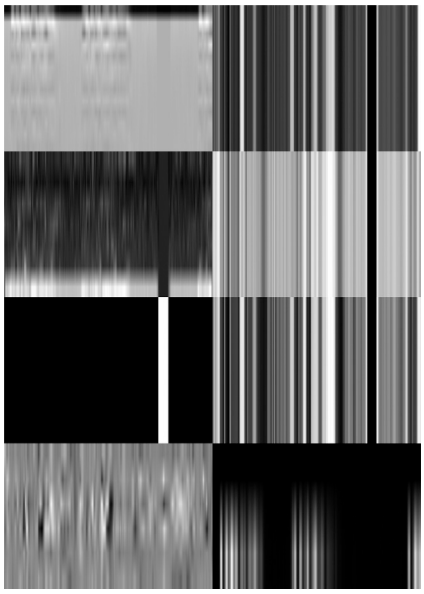

Fig. 2. Sample image representing feature after concatenation of individual spectrogram based features.

in Fig. 2. Such images are fed to the DCNN and SSGAN networks for the classification purpose.

4.2. Architecture

The overall architecture starting from processing of audio sample to identification of the language class is shown briefly in Fig. 3.

4.2.1. Deep dumb multi layer perceptron

Overview

MLP is a special class of neural network belonging to the class of feed-forward artificial neural network (ANN). A basic MLP unit consists of minimum three layers of nodes: input layer, hidden layer, and output layer (Haykin, 1994). Of these, input nodes use linear activation, and all the other nodes are the neurons that use an activation function which is nonlinear. Generally, it applies a technique belonging to the class of supervised learning called back propagation for the purpose of learning. The multiple layers that it contains and their property of non-linear activation aid to the classification of data that are not separable by using linear techniques. An MLP is called Deep Dumb if it consists of many hidden layers just stacked one after the other in a sequential manner.

Implementation of the architecture

We have used a DDMLP network (see Fig. 4) with 14 hidden layers along with an input layer and an output layer with Softmax activation. Extracted averaged out features from the audio clips are scaled and then fed as input to this model and output is softmax probability assigned to each language class. The output of this network is used in the later stage to form an ensemble learning.

4.2.2. Deep convolutional neural network

Convolutional layer

Primary building block of CNN (see Fig. 5) is the convolutional layer. The layer has a collection of parameters which mainly consists of a set of filters (or kernels) which are learnable. The kernels are simply small receptive fields, but can be extended through the full depth of the input volume. While moving forward, every filter is convolved throughout the width and height of the input volume. In the process,

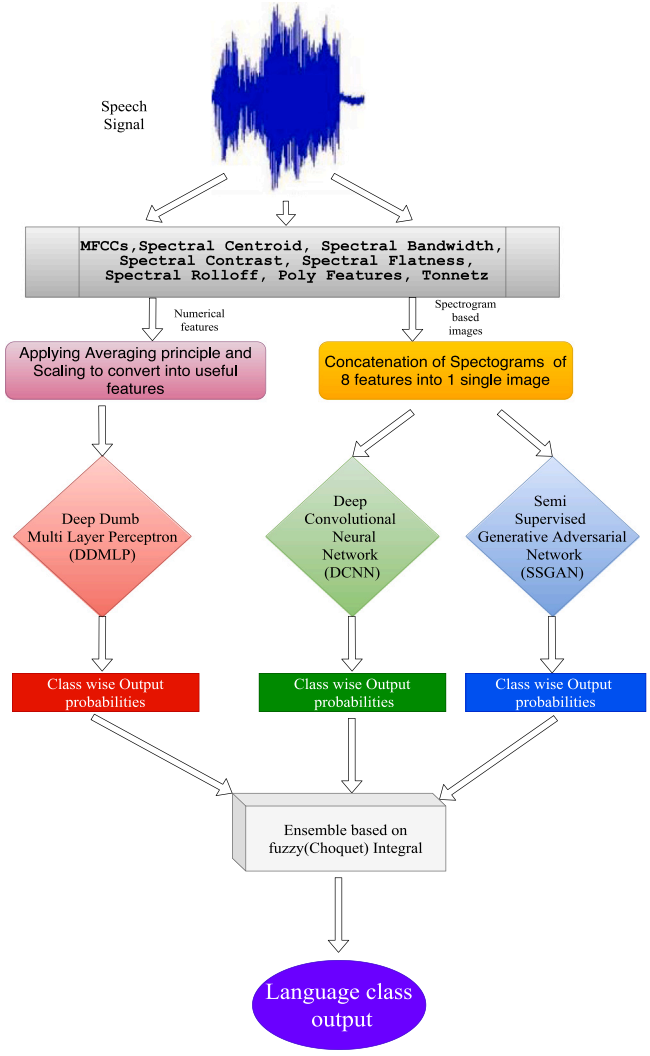

Fig. 3. Architecture of the proposed FuzzyGCP used for Spoken Language Identification.

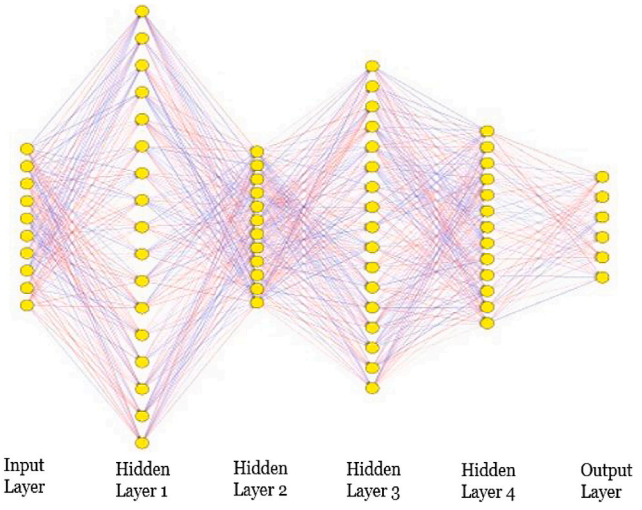

Fig. 4. A simple Deep Dumb Multi Layer Perceptron — DDMLP network.

the dot product is computed between the entries of the filter and the input. Hence, a 2-dimensional activation map of that filter is produced.

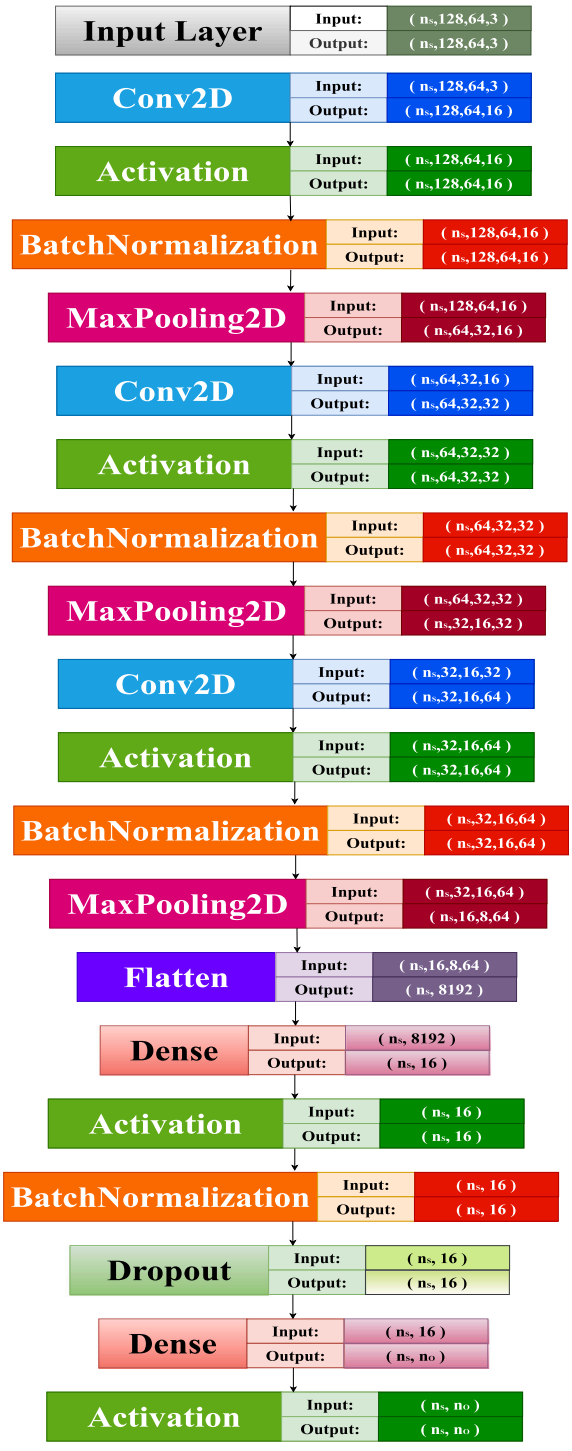

Fig. 5. Deep CNN model used in the present work.

As a direct result of which, the network learns to recognize the filters that activate whenever it detects some specific type of feature at some position in the input space.

For all the filters the activation maps are then stacked along the depth dimension. This generates the final output volume of the convolution layer. Therefore, each and every entry in this output volume can also be interpreted as an output of a neuron that considers a small region in the input and shares parameters with neurons in the same activation map.

Max Pooling layer

It is a method which is used to down-sample images using non-linear methods. It works by partitioning the input image into a collection of discrete and non-overlapping rectangular elements and, for each such rectangular region, outputs the maximum.

Intuitively, the exact location of a feature is less important than its rough location relative to other features. Thus a max-pooling gives rough estimation of edges. This is the idea behind the use of pooling in convolutional neural networks. It is common to periodically insert a pooling layer between successive convolutional layers in a CNN architecture (Ciresan et al., 2011).

The feature images obtained after concatenation of the spectrograms are fed to the model as input. The outputs thus obtained are later used to form the ensemble.

4.2.3. Semi supervised generative adversarial network

The semi-supervised GAN or SSGAN model is an extension of the GAN architecture that involves the simultaneous training of an unsupervised discriminator, supervised discriminator and a generator. Let us consider a standard classifier for classifying a data point x into one of the N possible classes, that is, labels of the data. This model accepts x as input and gives as output a N-dimensional vector of logits $\{l_1, \ldots, l_N\}$, that can thereafter be turned into class probabilities by applying the softmax function: $P_{model}(y = j|x) = \frac{exp(l_j)}{\sum_{n=1}^{N} exp(l_n)}$ In supervised learning, training of such a model is then done by minimizing the cross-entropy value between the observed outcome and the model predictive distribution $P_{model}(y|x)$. To make the classifier adapt to swift changes in quality of data samples, we can take a semi-supervised approach to generate fake data and mimic the possible noises and varieties that may be present in real data.

To apply semi-supervised learning approach with any standard classifier we can do so by simply adding samples to our dataset which are generated from the generator G of the GAN. These samples are then labeled with a new "generated" class $y = N + 1$, and correspondingly increase the dimension of our classifier output from N to $N + 1$ that is number of output classes increases by 1 unit. We may then make use of $P_{model}(y = N + 1|x)$ to supply the probability that x is fake, corresponding to $1 - D(x)$ in the original GAN framework. It can also learn from unlabeled data, as long as model knows that it corresponds to one of the N classes of real data by maximizing $log\ P_{model}(y \in \{1, \ldots, N\}|x)$. The brief working of this model is shown in Fig. 6.

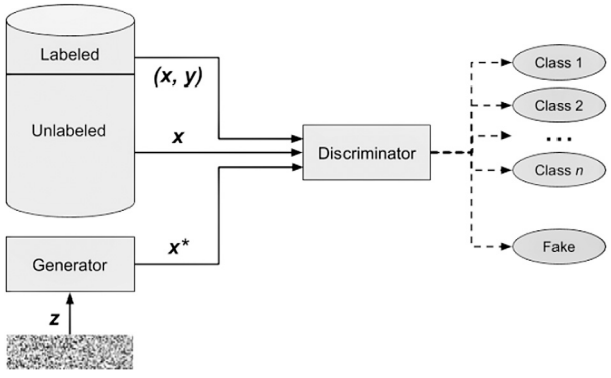

Fig. 6. Working procedure of multi-class SSGAN architecture.

Following the work by Salimans et al. (2016), we have implemented it in our own way. The loss of this multi-language classification framework can be decomposed into the supervised loss:

$$Loss_{supervised} = -\mathbb{E}_{x,y\sim P_{data}(x,y)} log\, P_{model}(y|x, y < N + 1) \quad (2)$$

The unsupervised loss is given by:

$$Loss_{unsupervised} = -\{\mathbb{E}_{x\sim P_{data}(x)} log[1 - P_{model}(y = N + 1|x)] + \mathbb{E}_{x\sim G} log[P_{model}(y = N + 1|x)]\} \quad (3)$$

The GAN loss of a discriminator:

$$Loss_{GAN} = -\{\mathbb{E}_{x,y\sim P_{data}(x,y)}[log P_{model}(y|x)] + \mathbb{E}_{x\sim G}[log\ P_{model}(y = N + 1|x)]\} \quad (4)$$

The best solution for minimizing both $Loss_{supervised}$ and $Loss_{unsupervised}$ is to have $exp[l_j(x)]$ = c(x) p(y=j,x)∀ j< N+1 and $exp[l_{N+1}(x)] = c(x)P_G(x)$ for some arbitrary function for scaling, $c(x)$. This can easily be deduced from the given values of $Loss_{supervised}$ and $Loss_{unsupervised}$. The unsupervised loss is thus consistent with the supervised loss as mentioned by Sutskever et al. (2015). Hence, we can get a clearer estimate of this optimal solution from the data by minimizing these two loss functions clubbed together. In practice, $Loss_{unsupervised}$ will only help if it is not trivial enough to minimize for our classifier, and thus we need to train G for approximation of the data distribution. One way to do this is by training G to minimize the game-value of the GAN model, using the discriminator, D, defined by our supervised classifier. This approach helps in introducing an interaction between G and the supervised classifier. Empirically we find that optimizing G using feature matching, Sutskever et al. (2015) GAN works very well for semi-supervised learning. On the other hand, training G using GAN by preventing learning from isolation, does not work at all.

Lastly, it is noted that the classifier with $N + 1$ outputs is over-parameterized, the reason being inclusion of an extra fake output class. On subtracting a general function $g(x)$ from each output logit, i.e. setting $l_j(x) \leftarrow l_j(x) - g(x)\forall j$, does not have any effect on the output of the softmax. This makes way for the hypothesis that we may equivalently fix $l_{N+1}(x) = 0\forall x$. In such a case $Loss_{supervised}$ becomes the standard supervised loss function of our original classifier with N classes. Also, our discriminator D is given by $D(x) = \frac{T(x)}{T(x)+1}$, where $T(x) = \sum_{n=1}^{N} exp[l_n(x)]$. The overall SSGAN architecture is shown in the Annexure section (Figure 11). The output of this model is later used for ensemble learning.

4.2.4. Ensemble learning

Usually, results obtained from a classifier may not be such precise or lack certainty. For architectures like GANs and CNNs, if trainable parameters are few enough, that is, features vectors have lesser dimensions they give great results. But they show degradation in precision if there is a steep increase in feature dimensions as pointed out by Miao et al. (2018). Similarly, architectures like MLP show increase in performance if higher dimensional features are fed to it for training. So application of fuzzy measures is useful for merging different classifiers to finally give one prediction result. It has been validated that popular fuzzy integral methods such as Sugeno and Choquet have great applications and been applied in a wide range of domains ranging from economics, mathematics to machine learning and pattern recognition (Wang et al., 2015).

Although both of these fuzzy integral methods are popular, Choquet fuzzy integral has been more widely applied than Sugeno integrals (Krishnan et al., 2015). A Choquet integral can be defined as an aggregation function that simultaneously keeps into consideration the importance of a classifier as well as its interaction with other classifiers in terms of output prediction. The definition of Choquet integral and fuzzy measures according to Murofushi and Sugeno (1989) are as follows. Let us assume X to be a set of various classifiers and the power set of X be denoted by P(X).

Definition 1. The fuzzy measure of X is a set function $z : P(X) \rightarrow [0, 1]$. This function satisfies the following conditions:

1. The boundary of z is : $z(\phi) = 0, z(X) = 1$
2. For each $A, B \in P(X)$ and $A \subset B$ then $z(A) \leq z(B)$ where z(k) is the grade of subjective importance of the classifier set k. The fuzzy singleton measures for each classifier are $z(x_i) = z^i$ and are commonly referred as densities. Not only must the value of each singleton be calculated, but also the value of function z for any combination of classifiers. The Sugeno λ-measure and fuzzy densities are used to calculate the fuzzy measure of any combination of classifiers. The λ-measure can be calculated by the following formula:

$$\lambda + 1 = \prod_{i=1}^{n}(1 + \lambda z^i), \lambda > -1 \quad (5)$$

Definition 2. z is the fuzzy measure of $X = \{x_1, x_2, \ldots, x_n\}$. The following equation shows the Choquet integral function of $f : X \rightarrow R$ and its relation with z:

$$C_z(f) = \sum_{i=1}^{n} f_i[z(A_i) - z(A_{i-1})] \quad (6)$$

The prediction result of classifier x_i, is denoted by f_i, and $[z(A_i) - z(A_{i-1})]$ depicts the relative importance of the classifier x_i. The fuzzy integral of f with respect to z is the result of integration.

Implementation of the architecture

The outputs from all the aforesaid architectures act as input for this fuzzy ensemble model. The class wise confidence probabilities and the obtained confusion matrices are used for getting the values of various parameters involved in calculating the ensemble model.

As mentioned in Siami et al. (2019), suppose in a sample data space S, data are divided into two classes by a classifier (E). A classifier index is specified by (i=1, …,P); j is the class index (j=1, …,M); and k is the instance index (k=1, …,N). For kth sample, the prediction result by the ith classifier is $[g_{i1}(k), g_{i2}(k), \ldots, g_{iM}(k)]$ where $g_{ij}(k)$ is the probability result of the ith classifier, which shows the probability of kth data belonging to class j. $[g_{1j}(k), g_{2j}(k), \ldots, g_{Pj}(k)]^T$ where $g_{ij}(k)$ is defined as $g_j(s_k)$ which can be interpreted as:

$g_j : S \rightarrow [0, 1], g_j(s_k) = [g_{1j}(k), g_{2j}(k), \ldots, g_{Pj}(k)]^T$ for sample s_k, we obtain a value for $g_j(s_k)$ as degree of support provided by each classifier with respect to the jth class for sample s_k. In addition to $g_j(s_k)$, the Choquet fuzzy integral operates on the fuzzy measures (z). This includes fuzzy densities as well as the fuzzy measure of any possible combination of classifiers. By calculating the Choquet integral of $g_j(s_k)$, z, we can provide the degree of support given by the ensemble model with respect to the jth class for sample s_k. The output class c_j for the sample s_k is the class with the largest integral value $C_z(f)$.

5. Results and analysis

In this section we have provided a detailed analysis of our findings from our experiments using the aforementioned architecture.

5.1. Evaluation metrics

For analyzing the performance of our model on various datasets, we have considered three standard performance metrics namely Precise, Recall and F1-score values with their corresponding class support division.

Precision is defined as:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (7)$$

Recall is defined as:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (8)$$

Here, TP (True Positive) = Number of audio files correctly classified into corresponding language classes

FP (False Positive) = Number of audio files classified to be belonging to a language class which they do not belong to

FN (False Negative) = Number of audio files classified to be not belonging to a language class which they actually belong to

F1-score is defined as:

$$F1 - score = 2 \times \frac{\text{Precision} \times \text{Recall}}{(Precision + \text{Recall})} \tag{9}$$

Support for a language class is defined as the number of audio files that lies in that language class.

5.2. Effect of sampling duration on classification accuracy

As it can be seen from Fig. 7 that there is a direct proportionality relationship between the length of sampling duration and language identification accuracy. Longer duration tends to give better results and this can be justified from the fact that longer sampling duration gives more information, i.e., features to be learned by the model for proper representation of the data which eventually helps classifiers to distinguish different classes of data.

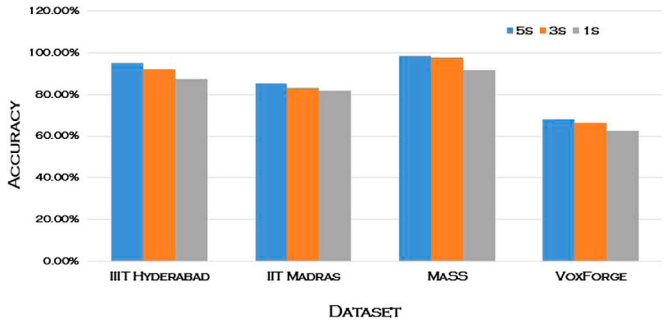

Fig. 7. SLID accuracy attained for various datasets with respect to their sampling duration.

5.3. Overall results

All the datasets are divided into train, validation and test sets in the ratio of 70:20:10. The corresponding identification accuracies of our model on the test sets are shown in Table 1.

We came across many works where both training and testing are done on whole dataset. This may lead to impressive results on the given dataset, but when tested on new data, it may lead to drastic degradation in performance. However, our results are evaluated on a test dataset which are then fed as a completely new data to our classification model for the testing purposes.

Identification accuracies attained by different state-of-the-art feature vectors like Parallel Phone Recognition and Language Modeling (PPRLM) (Zissman & Singer, 1994), i-vector (Snyder et al., 2015), x-vector (Snyder et al., 2018b) over four datasets are shown in Table 1.

Although these feature vectors find their use mainly for speaker recognition purposes, but considering each speaker as a language class, we have tried to give a fair comparison for the strategies.

For carrying out comparative analysis, we have used the 'Kaldi' framework (Povey et al., 2011) for computing i-vectors and x-vectors and the project by Srivastava et al. (2017) for computing results for PPRLM (codes). Fortunately, Kaldi has support for pretrained x-vectors (here) and Probabilistic Linear Discriminant Analysis (PLDA) backend. All scripts have been readily made available by the developers of the Kaldi project, the commands for which have been included in the Annexure. The same goes for other sets of features. The number of feature units in input layer is same as number of features extracted for the Time Delay Neural Network (TDNN) used for x-vector extraction. The whole procedure for x-vector feature generation is shown in Fig. 8. The X-Y-Z written in the blocks represent the number of filters, filter window size and dilation number at the corresponding layers respectively. After

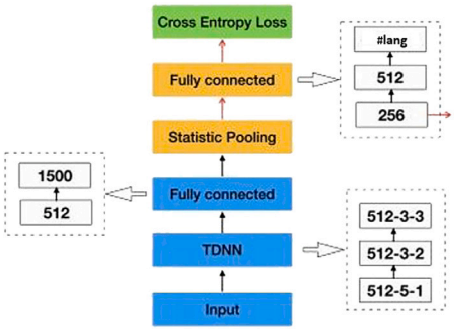

Fig. 8. Schematic diagram showing the baseline x-vector system.

Table 1
Comparison of test accuracies of our proposed FuzzyGCP model with different state-of-the-art feature vectors and language models on four SLID datasets.

Dataset	PPRLM	i-vector	x-vector	FuzzyGCP model
IIIT Hyderabad	81.32%	79.87%	84.47%	95.00%
IIT Madras	72.33%	74.55%	74.65%	81.51%
VoxForge	57.12%	56.53%	59.67%	68.00%
MaSS	86.21%	84.73%	88.57%	98.75%

extraction of the embeddings, PLDA is used as backend scoring by the standard Kaldi x-vector project.

For the i-vector calculation, given a speech utterance of the language, the channel dependent GMM-supervector (Campbell et al., 2006) m_l can be written as:

$$m_l = m + Tw_l \tag{10}$$

where, m is the GMM-supervector of the universal background model (UBM) (Reynolds et al., 2000) which is both language and channel-independent, T is a low-rank total variability matrix, and the posterior mean of w_s is a low-dimension vector called i-vector.

From the speech samples, 13-dimensional MFCCs and various other spectral features along with energy and their delta and acceleration coefficients, forming 38-dimensional acoustic features are extracted. The GMM consists of 64 Gaussian mixtures thus, resulting in a feature vector of 2432(D) dimensions. Given an utterance with D-dimensional acoustic vector sequence $X = \{x_1, \ldots, x_T\}$ belonging to language l, we can write

$$T = [t_1, \ldots, t_R] \tag{11}$$

where, t_r is an $MD \times 1$ column vector.

It can be observed that the x-vector based feature outperforms the i-vector based feature. The probable reason being that the x-vectors exploit the large increase in the amount of in-domain data better than the i-vectors, thus proving better compared to i-vectors (acoustic). However, all these state-of-the-art feature vectors show poorer results comparatively. The main reasons for this are these feature vectors are used in scenarios where the number of channels are very low, and enough resources are not available to perform an in-depth analysis of hyperparameters for comparison as it would be out of scope of the present work.

The distribution of sounds, shown in Table 2, helps in discriminating Indo-Aryan and Dravidian languages from others through their specific acoustic, phonetic and prosody characteristics (Aarti & Kopparapu, 2018).

5.3.1. IIIT hyderabad

It can be seen that the performance of our model in classifying Tamil is not so good as compared to other languages. It has been observed that there is presence of several keywords like 'Malayalam',

Table 2
Sound distribution found in different Indic languages.

Language	Vowels	Diphthongs	Liquids	Glides	Consonants	Nasals	Stops	Fricatives	Affric
Kannada	13	2	3	2	34	5	16	5	4
Malayalam	10	5	4	2	37	6	16	4	3
Hindi	10	2	4	2	37	5	20	4	4
Tamil	10	2	3	2	18	5	4	4	2
Marathi	12	2	3	2	45	3	16	3	7
Bengla	14	15	3	3	34	4	20	2	4
Telugu	11	2	3	2	36	3	15	5	4

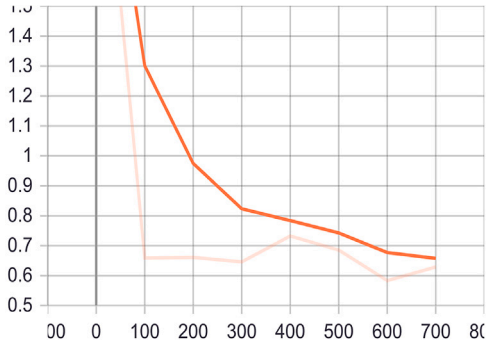

Fig. 9. Fake image loss.

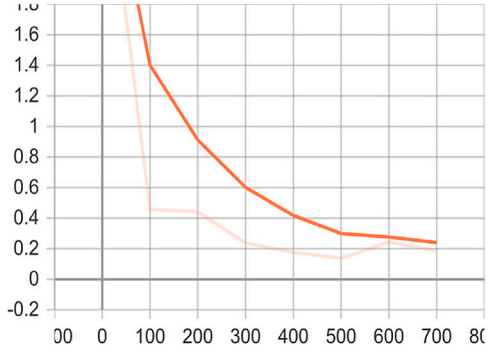

Fig. 10. Real image loss.

'Hindi', 'India' etc. which are also found in the Malayalam and Tamil audio clips. Presence of these keywords adds confusion thus resulting in mis-classification.

Additionally, in the Marathi audio clip, there are words like "Engreji" meaning English, which is also present in frequent basis in the Bangla language class. This, in turn, leads to comparatively poor identification results as well. The class-wise distribution of different evaluation metrics of multi-lingual approach is shown in Table 3. Our model performs best for the Hindi and Malayalam language classes of this dataset.

The worst performances are given for the Tamil and Marathi language classes. Comparison of performance of our FuzzyGCP model with the state-of-the-art models (Anjana & Poorna, 2018) is shown in Table 4.

SSGAN training metrics on the IIIT Hyderabad dataset are shown in Figs. 9–12. Fig. 9 shows the iterative training loss of the fake image generator, while Fig. 10 shows the iterative training loss of the real image discriminator. Fig. 11 shows the iterative training loss of the supervised classifier, and Fig. 12 shows the iterative training accuracy of the SSGAN's supervised classifier.

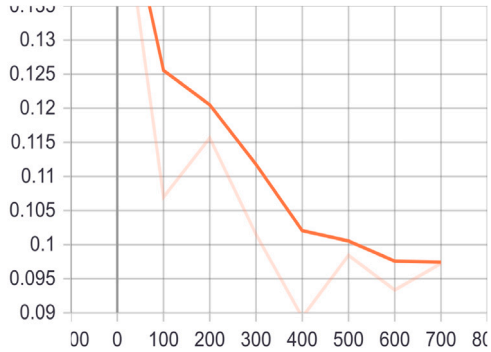

Fig. 11. Classification loss.

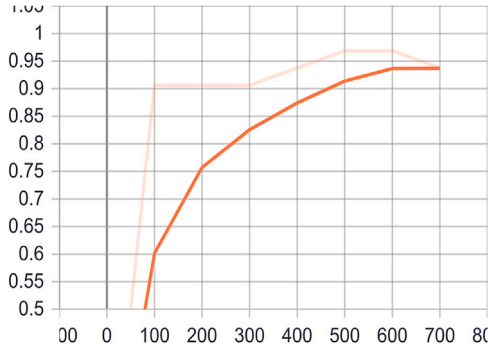

Fig. 12. SSGAN accuracy.

Table 3
Language-wise performance of our FuzzyGCP model on IIIT Hyderabad dataset.

Class	Precision	Recall	F1-score	Support
Bangla	0.95	0.95	0.95	104
Kannada	0.96	0.94	0.95	101
Marathi	0.92	0.90	0.91	94
Tamil	0.91	0.89	0.90	92
Malayalam	0.97	0.96	0.97	109
Telugu	0.96	0.96	0.96	96
Hindi	0.98	0.95	0.97	104
Macro avg	**0.95**	**0.95**	**0.95**	**700**

Table 4
Comparison of performance of our proposed model with other state-of-the-art models on IIIT Hyderabad dataset.

Classifier	Features used	Accuracy (%)	Precision (%)	Recall (%)
SVM	MFCC	78	24	53.3
	Formants	86	62.47	60
	Both	84	52.17	46.66
LDA	MFCC	80.02	31.51	57.14
	Formants	93.06	80.57	75.71
	Both	93.88	84.24	78.57
Our model	Extracted features	**95.00**	**95.00**	**95.00**

5.3.2. IIT madras

The class-wise distribution of different evaluation metrics of multi-lingual approach on IIT Madras dataset is shown in Table 5. Our model performs best for the Marathi and Telugu language classes. However, it is to be noted from Table 5 that the worst performances are obtained by our model for Hindi language followed by Bangla and Tamil languages.

Table 5
Class-wise performance measures of our proposed FuzzyGCP model on IIT Madras dataset.

Class	Precision	Recall	F1-score	Support
English	0.76	0.80	0.78	100
Marathi	0.84	0.95	0.89	100
Tamil	0.79	0.73	0.76	100
Bangla	0.72	0.79	0.75	100
Telugu	0.90	0.72	0.80	100
Hindi	0.75	0.71	0.73	100
Macro avg	**0.79**	**0.78**	**0.79**	**600**

Table 7
Performance of our FuzzyGCP model for individual language classes on VoxForge dataset.

Class	Precision	Recall	F1-score	Support
French	0.64	0.67	0.65	188
German	0.68	0.59	0.63	191
Italian	0.66	0.75	0.70	210
Portuguese	0.70	0.83	0.76	202
Spanish	0.67	0.56	0.61	209
Macro avg	**0.67**	**0.68**	**0.67**	**1000**

The class balance that is division of support values for each class leads to proper similarity in Recall values. It is already mentioned earlier that most of the Indic languages are originated from the early Brahmi script. Out of all these languages, Hindi is semantically the most similar language to Sanskrit, followed by Bangla. Thus, we can find a transitive relation here.

Let us represent the language classes Hindi, Sanskrit and Bangla by p, q, r respectively and $\mathcal{Z}$ denote the set of Indic languages.

A relation R on the set $\mathcal{Z}$ is a transitive relation if, for all $p, q, r \in \mathcal{Z}$, if p R q and q R r, then p R r. If thought logically:

$$\forall p, q, r \in \mathcal{Z} : (pRq \wedge qRr) \Rightarrow pRr \tag{12}$$

where p R q is the infix notation for $(p, q) \in R$.

Thus, we see that p and r are related to each other. This similarity in grammar and other linguistic features lead to more mis-classification between these language classes compared to others.

Performance comparison of our proposed model with other methods by Jog et al. (2018) on 6 different languages of IIT Madras dataset with common languages being Hindi, Marathi and Telugu is shown in Table 6. As we can see that the performance of the works by Jog et al. (2018) and Sarkar et al. (2013) is much better compared to our model. The probable significant reasons for the same may be, firstly, a difference in the set of languages used for experimentation. Perhaps those language classes show better distinguishing features. Secondly, they have trained and tested on the whole dataset which is bound to give better results as the classifier has got the sufficient samples to learn efficiently.

In Table 6, the language groups consist of following languages:

- 1 - Kannada, Malayalam, Marathi, Telugu, Hindi, and Manipuri
- 2 - English, Marathi, Tamil, Bangla, Telugu, and Hindi

5.3.3. VoxForge

The audio samples found in this database are very noisy. Jargon (merriam webmaster) is a type of linguistic shortcut, which helps in quicker and clearer communication, if everyone listening understands the terminology. However, if the listeners have different definitions of the terminologies, then jargon becomes noisy. In the VoxForge dataset, adequate number of jargons are found. Additionally, the whole dataset is vast consisting of samples with variable record durations and sampling rates. So, extracting the best quality data was difficult for us from the computational perspective. However, our model performs the best for the Portuguese and Italian language classes of this dataset and worst for the Spanish and German language classes respectively as seen in Table 7. As we can see from Table 8, the state-of-the-art models fail considerably when it comes to multi-lingual classification on this dataset.

5.3.4. MaSS

Hungary and Romania are neighboring countries. So, it is obvious that both the countries have linguistic and cultural influences over each other. Over the years there have been extension of deeper influence pertaining to the exchange of even basic characteristics of a language such as morphology and grammar. This influence is the prime reason for the relatively poor classification for the Hungarian and Romanian language classes compared to others. However, the best performance of our model has been found in case of this dataset. As seen in Table 9, our model performs best for the Finnish, Spanish and French language classes of this dataset. We could not show a comparison with other models, as no spoken language identification work has been done on this dataset so far.

5.4. Bi-lingual results

Bi-lingual results are shown for Indic datasets due to the presence of dependency between different languages. It can be seen that bi-lingual classification results are much higher compared to multi-lingual approach. This is pertaining to the fact that binary classification is much more robust to noise and dependency in features compared to the scenario where the classifier needs to distinguish among more language classes.

Table 6
Comparison of performance of our FuzzyGCP model with some existing works on IIT Madras dataset.

Method	Features	Language group	Test Accuracy/ Error Rates
Jothilakshmi et al. (2012)	MFCC	1	80.56%
Singh et al. (2013)	Prosodic features	1	EER is 7.46%
Aarti and Kopparapu (2017)	Delta MFCC, Double Delta MFCC	1	Lesser than 45% for different windows
Madhu et al. (2017)	Phonotactic, Prosodic features	1	72% and 68%
Sarkar et al. (2013)	MFCC	1	95.21%
Verma and Khanna (2013)	MFCC	1	81%
Jog et al. (2018)	Cochleagram Based Texture Descriptors	1	95.36%
Proposed method	**Extracted features**	2	**81.51%**

Table 8
Performance comparison of FuzzyGCP model with other state-of-the-art models on VoxForge dataset.

Model	Precision(%)	Recall(%)	F1-score(%)
kNN	44.33	36.04	40
SVM	41.44	44.67	44
Extratrees	37.23	34.5	35.81
Our model	**67**	**68**	**67**

Table 9
Language-wise performance measures attained by our FuzzyGCP model on MaSS Dataset.

Class	Precision	Recall	F1-score	Support
Basque	0.93	1.00	0.96	25
Russian	0.94	1.00	0.97	15
French	1.00	0.97	0.98	30
Romanian	1.00	0.88	0.93	24
Hungarian	0.96	0.92	0.94	29
Spanish	0.97	1.00	0.98	28
English	0.94	1.00	0.97	29
Finnish	1.00	1.00	1.00	28
Macro avg	**0.98**	**0.98**	**0.98**	**208**

Table 10
Precision values of class-wise binary classification on IIIT Hyderabad dataset.

Classes	Bangla	Kannada	Marathi	Tamil	Malayalam	Telugu	Hindi
Bangla	X	0.98	0.95	0.94	0.97	0.98	0.94
Kannada	0.98	X	0.98	0.92	0.97	1.00	0.99
Marathi	0.95	0.98	X	0.98	0.99	1.00	1.00
Tamil	0.94	0.92	0.98	X	0.97	0.92	0.96
Malayalam	0.97	0.97	0.99	0.97	X	1.00	0.98
Telugu	0.99	1.00	1.00	0.92	1.00	X	0.99
Hindi	0.94	0.99	1.00	0.96	0.98	0.99	X

Table 11
Recall values of class-wise binary classification on IIIT Hyderabad dataset.

Classes	Bangla	Kannada	Marathi	Tamil	Malayalam	Telugu	Hindi
Bangla	X	0.97	0.95	0.94	0.97	0.99	0.94
Kannada	0.97	X	0.98	0.92	0.97	1.00	0.98
Marathi	0.95	0.98	X	0.97	0.98	1.00	0.99
Tamil	0.94	0.92	0.97	X	0.97	0.92	0.96
Malayalam	0.97	0.97	0.98	0.97	X	1.00	0.97
Telugu	0.99	1.00	1.00	0.92	1.00	X	0.99
Hindi	0.94	0.98	0.99	0.96	0.97	0.99	X

5.4.1. Bi-lingual classification on IIIT Hyderabad dataset

The precision, recall and F1-score values for the bi-lingual binary classification achieved by our proposed model are shown in Tables 10, 11 and 12 respectively. The achievement of having F1-score values of 100% for the language pairs (Kannada, Telugu), (Marathi, Telugu) and (Malayalam, Telugu) with Telugu being the common language, shows that Telugu has significant features that clearly distinguish it from all these languages. However, it can be seen that the language pair (Malayalam, Telugu) has a lower F1-score value, showing comparatively more similarity. As discussed earlier, both the Malayalam and Tamil languages have some terms common between them. So, this pair attains a F1-score of 0.97 which is much less than the collective average. Similarly, the results for (Bangla, Hindi) language pair is also found to be less (0.94) as compared to global average.

Table 12
F1-scores of class-wise binary classification on IIIT Hyderabad dataset.

Classes	Bangla	Kannada	Marathi	Tamil	Malayalam	Telugu	Hindi
Bangla	X	0.97	0.95	0.93	0.97	0.99	0.93
Kannada	0.97	X	0.98	0.92	0.97	1.00	0.98
Marathi	0.95	0.98	X	0.98	0.99	1.00	0.99
Tamil	0.93	0.92	0.98	X	0.97	0.91	0.96
Malayalam	0.97	0.97	0.99	0.97	X	1.00	0.97
Telugu	0.99	1.00	1.00	0.91	1.00	X	0.99
Hindi	0.93	0.98	0.99	0.96	0.97	0.99	X

Table 13
Precision values of class-wise binary classification on IIT Madras dataset.

Classes	English	Marathi	Tamil	Bangla	Telugu	Hindi
English	X	0.93	0.75	0.61	0.73	0.72
Marathi	0.93	X	0.95	0.87	1.00	1.00
Tamil	0.75	0.95	X	0.89	0.89	0.98
Bangla	0.61	0.87	0.89	X	0.90	0.59
Telugu	0.73	1.00	0.89	0.90	X	0.92
Hindi	0.72	1.00	0.98	0.59	0.92	X

Table 14
Recall values of class-wise binary classification on IIT Madras dataset.

Classes	English	Marathi	Tamil	Bangla	Telugu	Hindi
English	X	0.92	0.74	0.60	0.72	0.71
Marathi	0.92	X	0.95	0.84	1.00	1.00
Tamil	0.74	0.95	X	0.88	0.88	0.97
Bangla	0.60	0.84	0.88	X	0.90	0.58
Telugu	0.72	1.00	0.88	0.90	X	0.90
Hindi	0.71	1.00	0.97	0.58	0.90	X

Table 15
F1-scores of class-wise binary classification on IIT Madras dataset.

Classes	English	Marathi	Tamil	Bangla	Telugu	Hindi
English	X	0.92	0.74	0.60	0.72	0.71
Marathi	0.92	X	0.95	0.84	1.00	1.00
Tamil	0.74	0.95	X	0.88	0.88	0.97
Bangla	0.60	0.84	0.88	X	0.89	0.58
Telegu	0.72	1.00	0.88	0.89	X	0.90
Hindi	0.71	1.00	0.97	0.58	0.90	X

5.4.2. Bi-lingual classification on IIT Madras dataset

The precision, recall and F1-score values for the bi-lingual binary classification achieved by our proposed model on IIT Madras dataset are shown in Tables 13, 14 and 15 respectively. As it can be seen that the binary classification of Hindi and Bangla shows the worst performance (58%), thus proving our earlier analysis to be true. Just as seen before, the proposed model achieves 100% F1-score for the (Marathi, Telugu) language pair proving them to be the most distinguishable among the Indic languages considered here. Also, the (Marathi, Hindi) language pair gives F1-score of 100%. On the other hand, the highest mis-classification is found for the (English, Bangla) and (English, Hindi) language pairs. Probably we fail to extract distinguishable features for the same.

5.5. Tri-lingual results

We have also shown the classification results on tri-lingual scenarios of Indic datasets as we can see that in the neighboring states many people are used to speak multiple languages.

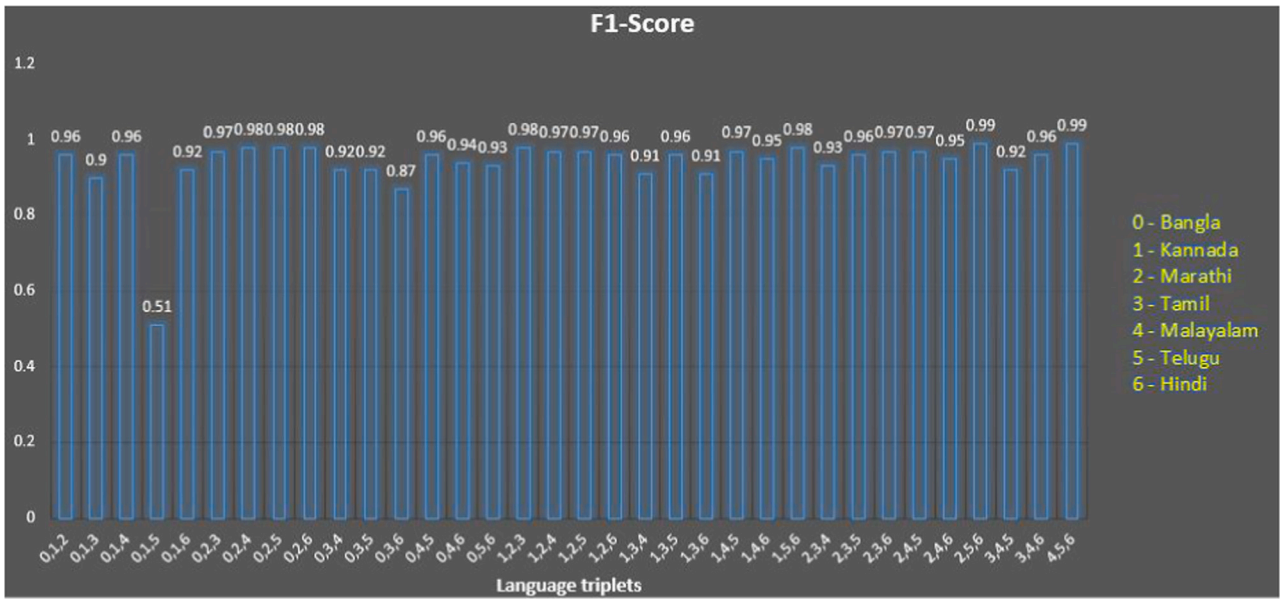

Fig. 13. Tri-lingual classification F1-scores for various language class triplets in IIIT Hyderabad dataset.

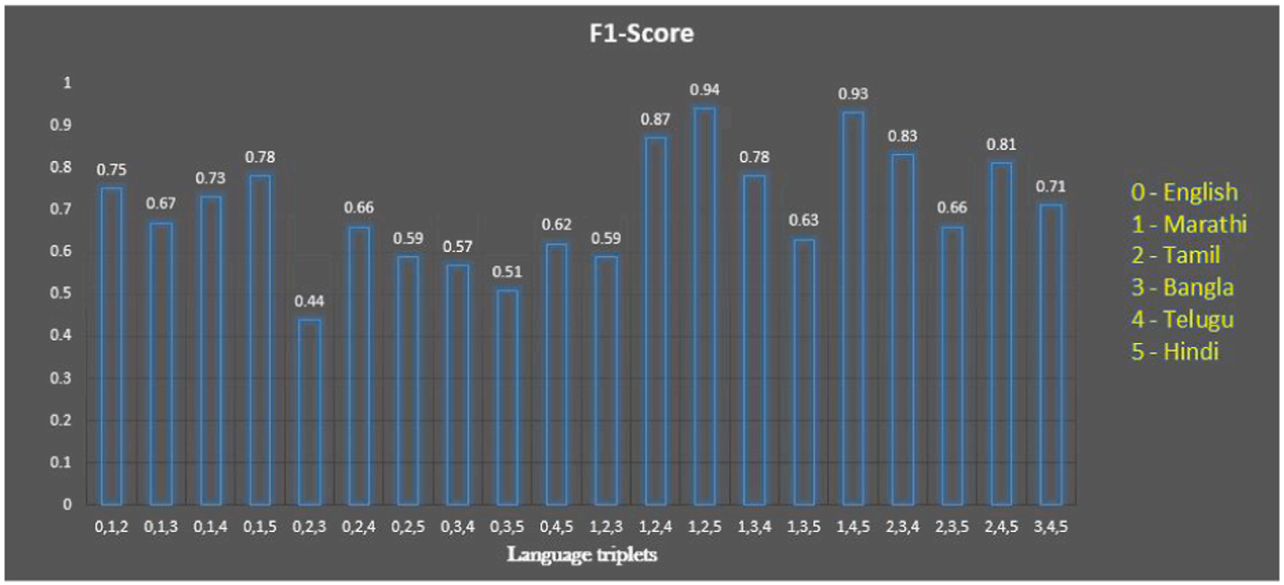

Fig. 14. Tri-lingual classification F1-scores for various language class triplets in IIT Madras dataset.

5.5.1. Tri-lingual classification on IIIT Hyderabad dataset

The tri-lingual classification results on IIIT Hyderabad dataset are shown in Fig. 13. The number triplets in the X-axis correspond to the equivalent language codes as shown in the legend. For example, the triplet (0,1,2) corresponds to the languages Bangla, Kannada and Marathi respectively. All the triplets where (Bangla, Hindi) and (Tamil, Malayalam) language pairs are present together have shown lower F1-scores compared to others. The reasons for the same have been explained before and the results here prove the correctness of the analysis. The triplet (Bangla, Kannada, Telugu) shows an abnormal dip in F1-score. The model is unable to detect any Bangla language in presence of the other two languages and gave 0% recall value for the same. It is a known fact that distinguishing Kannada language from Telugu language is very difficult. The situation might have been such that our model gives too much weightage to learning of features that could help it to classify both Kannada and Telugu language classes.

5.5.2. Tri-lingual classification on IIT Madras dataset

The tri-lingual classification results on IIT Madras dataset are shown in Fig. 14. The number triplets in the X-axis correspond to the equivalent language codes as shown in the legend. For example, the triplet (0,1,2) corresponds to the languages English, Marathi and Tamil respectively. All the triplets where (Bangla, Hindi) and (Tamil, Bangla) language pairs are present together have shown lower F1-scores compared to others, proving the correctness of our analysis discussed above. It can be seen that the tri-lingual classification F1-scores are much better in comparison to the multi-lingual approach. The reason surely is more relative differences in characteristic features which is an immediate result of decreased diversity in language classes.

6. Conclusion

In this paper, we have proposed a method for identifying the spoken languages from speech signals. In doing so, we have designed

a deep learning based ensemble architecture which we have named as FuzzyGCP. Use of SSGAN along with forming of an ensemble architecture is a new approach in this domain. Mapping the audio classification problem to image classification problem by making use of spectrograms is one of the key aspects of this architecture. Heterogeneous ensemble consisting of a conventional DDMLP as a classifier using numeric features along with DCNN and SSGAN as classifiers for image based features, proved to be quite a useful approach as observed from the results reported in Section 5. The diversity in datasets which we have considered consisting both of Indic and foreign languages prove the robustness and versatility of FuzzyGCP. A multi-lingual classification approach is always a challenging task compared to its bi-lingual and tri-lingual counterparts in the domain of SLID and it has been quite successfully accomplished here. However, challenges like the inter dependency of Indic languages among themselves, presence of common set of words in the languages and demographic influence on the languages need to be addressed with better understanding.

As for future scope, improvement may be done by using some feature selection algorithms, and by using lesser computational intensive architectures. Usage of sequential models like Gated Recurrent Units (GRUs), LSTMs etc. can be checked out to form an ensemble. Also in-depth analysis of the x-vector and i-vector based models with proper tuning of hyperparameters on the datasets explored in this work can be considered. Besides, there exists many other speech corpus which can be used for evaluation. The immediate benefit of a proper multi-lingual SLID system is that it can be developed based on the output of this model for other purposes like speaker profile generation, automatic translation switching frameworks, ease of understanding in tele-medicine purposes etc.

CRediT authorship contribution statement

Avishek Garain: Software, Methodology, Formal analysis, Writing - original draft. **Pawan Kumar Singh:** Conceptualization, Validation, Resources, Data curation, Writing - review & editing. **Ram Sarkar:** Investigation, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

We would like to thank the CMATER research laboratory of the Computer Science and Engineering Department, Jadavpur University, India for providing us the infrastructural support.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.eswa.2020.114416.

References

Aarti, B., & Kopparapu, S. K. (2017). Spoken Indian language classification using artificial neural network—An experimental study. In *2017 4th international conference on signal processing and integrated networks* (pp. 424–430). IEEE.

Aarti, B., & Kopparapu, S. K. (2018). Spoken Indian language identification: a review of features and databases. *Sādhanā*, *43*(4), 53.

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., ... Zheng, X. (2015). Tensorflow: Large-scale machine learning on heterogeneous systems.

Albadr, M. A. A., Tiun, S., Ayob, M., & AL-Dhief, F. T. (2019). Spoken language identification based on optimised genetic algorithm–extreme learning machine approach. *International Journal of Speech Technology*, *22*(3), 711–727.

Anjana, J., & Poorna, S. (2018). Language identification from speech features using SVM and LDA. In *2018 International conference on wireless communications, signal processing and networking* (pp. 1–4). IEEE.

Baby, A., Thomas, A., L, N., & Consortium, T. (2016). *Resources for Indian languages*.

Berkson, K. H. (2013). *Phonation types in Marathi: An acoustic investigation* (Ph.D. thesis), University of Kansas.

Bhaskararao, P. (2011). Salient phonetic features of Indian languages in speech technology. *Sadhana*, *36*(5), 587–599.

Black, A. W. (2019). CMU wilderness multilingual speech dataset. In *2019 IEEE international conference on acoustics, speech and signal processing* (pp. 5971–5975).

Boito, M. Z., Havard, W. N., Garnerin, M., Ferrand, ÉricLe., & Besacier, L. (2020). MaSS: A large and clean multilingual corpus of sentence-aligned spoken utterances extracted from the bible. In *Language resources and evaluation conference*.

Caldwell, R. (1875). *A comparative grammar of the Dravidian or South-Indian family of languages*. Trübner.

Campbell, W. M., Sturim, D. E., Reynolds, D. A., & Solomonoff, A. (2006). SVM based speaker verification using a GMM supervector kernel and NAP variability compensation. In *2006 IEEE international conference on acoustics speech and signal processing proceedings (vol. 1)* (p. I). IEEE.

Chollet, F. (2015). Keras. https://keras.io.

Ciresan, D. C., Meier, U., Masci, J., Gambardella, L. M., & Schmidhuber, J. (2011). Flexible, high performance convolutional neural networks for image classification. In *Twenty-second international joint conference on artificial intelligence*.

Dehak, N., Torres-Carrasquillo, P. A., Reynolds, D., & Dehak, R. (2011). Language recognition via i-vectors and dimensionality reduction. In *Twelfth annual conference of the international speech communication association*.

Ferrer, L., Lei, Y., McLaren, M., & Scheffer, N. (2014). Spoken language recognition based on senone posteriors. In *Fifteenth annual conference of the international speech communication association*.

Gu, L., & Rose, K. (2001). Perceptual harmonic cepstral coefficients for speech recognition in noisy environment. In *2001 IEEE international conference on acoustics, speech, and signal processing. proceedings (Cat. No. 01CH37221) (vol. 1)* (pp. 125–128). IEEE.

Haykin, S. (1994). *Neural networks: a comprehensive foundation*. Prentice Hall PTR.

Hermansky, H. (1990). Perceptual linear predictive (PLP) analysis of speech. *The Journal of the Acoustical Society of America*, *87*(4), 1738–1752.

Hermansky, H., & Morgan, N. (1994). RASTA processing of speech. *IEEE Transactions on Speech and Audio Processing*, *2*(4), 578–589.

Imai, S. (1983). Cepstral analysis synthesis on the mel frequency scale. In *IEEE international conference on acoustics, speech, and signal processing (vol. 8)* (pp. 93–96). IEEE.

Jog, A. H., Jugade, O. A., Kadegaonkar, A. S., & Birajdar, G. K. (2018). Indian language identification using cochleagram based texture descriptors and ANN classifier. In *2018 15th IEEE India council international conference* (pp. 1–6). IEEE.

Jothilakshmi, S., Ramalingam, V., & Palanivel, S. (2012). A hierarchical language identification system for Indian languages. *Digital Signal Processing*, *22*(3), 544–553.

Keane, E. (2004). Tamil. *Journal of the International Phonetic Association*, *34*(1), 111–116. http://dx.doi.org/10.1017/S0025100304001549.

Krishnan, A. R., Kasim, M. M., & Bakar, E. M. N. E. A. (2015). A short survey on the usage of choquet integral and its associated fuzzy measure in multiple attribute analysis. *Procedia Computer Science*, *59*, 427–434. http://dx.doi.org/10.1016/j.procs.2015.07.560, International Conference on Computer Science and Computational Intelligence (ICCSCI 2015). http://www.sciencedirect.com/science/article/pii/S187705091502089X.

Lee, R.-H. A., & Jang, J.-S. R. (2018). A syllable structure approach to spoken language recognition. In T. Dutoit, C. Martín-Vide, & G. Pironkov (Eds.), *Statistical language and speech processing* (pp. 56–66). Cham: Springer International Publishing.

Li, H., Ma, B., & Lee, K. A. (2013). Spoken language recognition: From fundamentals to practice. *Proceedings of the IEEE*, *101*(5), 1136–1159.

Logan, B. (2000). Mel frequency cepstral coefficients for music modeling. In *Ismir (vol. 270)* (pp. 1–11).

Madhu, C., George, A., & Mary, L. (2017). Automatic language identification for seven Indian languages using higher level features. In *2017 IEEE international conference on signal processing, informatics, communication and energy systems* (pp. 1–6). IEEE.

McFee, B., McVicar, M., Raffel, C., Liang, D., Nieto, O., Moore, J., Ellis, D., Repetto, D., Viktorin, P., ao Felipe Santos, J., & Holovaty, A. (2015). *librosa: v0.4.0*. Zenodo, http://dx.doi.org/10.5281/zenodo.18369.

merriam webmaster (0000). Jargon. Merriam-Webster. https://www.merriam-webster.com/dictionary/jargon.

Miao, X., McLoughlin, I., & Yan, Y. (2019). A new time-frequency attention mechanism for TDNN and CNN-LSTM-TDNN, with application to language identification. In *INTERSPEECH* (pp. 4080–4084).

Miao, X., McLoughlin, I., Yao, S., & Yan, Y. (2018). Improved conditional generative adversarial net classification for spoken language recognition. in *2018 IEEE spoken language technology workshop* (pp. 98–104).

Murofushi, T., & Sugeno, M. (1989). An interpretation of fuzzy measures and the Choquet integral as an integral with respect to a fuzzy measure. *Fuzzy Sets and Systems*, *29*(2), 201–227. http://dx.doi.org/10.1016/0165-0114(89)90194-2.

mustgo. com (0000). Kannada language - structure, writing and alphabet - MustGo. MustGo.com. URL https://www.mustgo.com/worldlanguages/kannada/.

O'Shaughnessy, D. (1988). Linear predictive coding. *IEEE Potentials*, *7*(1), 29–32.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., & Dubourg, V. (2011). Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, *12*, 2825–2830.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., & Vesely, K. (2011). The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, IEEE Catalog No.: CFP11SRW-USB.

Prahallad, K., Elluru, N. K., Keri, V., Rajendran, S., & Black, A. W. (2012). The IIIT-H indic speech databases. In *INTERSPEECH*.

Reynolds, D. A., Quatieri, T. F., & Dunn, R. B. (2000). Speaker verification using adapted Gaussian mixture models. *Digital Signal Processing*, *10*(1–3), 19–41.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training GANs. In *Proceedings of the 30th international conference on neural information processing systems* (pp. 2234–2242). Red Hook, NY, USA: Curran Associates Inc.

Sarkar, S., Rao, K. S., Nandi, D., & Kumar, S. S. (2013). Multilingual speaker recognition on Indian languages. In *2013 Annual IEEE India conference* (pp. 1–5). IEEE.

Shukla, S., & Mittal, G. (2019). Spoken language identification using convnets. In *European conference on ambient intelligence* (pp. 252–265). Springer.

Siami, M., Naderpour, M., & Lu, J. (2019). A Choquet fuzzy integral vertical bagging classifier for mobile telematics data analysis. in *2019 IEEE international conference on fuzzy systems* (pp. 1–6).

Singh, O. P., Haris, B., Sinha, R., Chettri, B., & Pradhan, A. (2013). Sparse representation based language identification using prosodic features for Indian languages. In *2013 Annual IEEE India conference* (pp. 1–5). IEEE.

Snyder, D., Garcia-Romero, D., McCree, A., Sell, G., Povey, D., & Khudanpur, S. (2018). Spoken language recognition using X-vectors. In *Odyssey* (pp. 105–111).

Snyder, D., Garcia-Romero, D., & Povey, D. (2015). Time delay deep neural network-based universal background models for speaker recognition. In *2015 IEEE workshop on automatic speech recognition and understanding* (pp. 92–97). IEEE.

Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., & Khudanpur, S. (2018). X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE international conference on acoustics, speech and signal processing* (pp. 5329–5333). IEEE.

Srivastava, B. L., Vydana, H., Vuppala, A. K., & Shrivastava, M. (2017). Significance of neural phonotactic models for large-scale spoken language identification. In *2017 International joint conference on neural networks* (pp. 2144–2151).

Stoica, P., & Moses, R. L. (2005). *Spectral analysis of signals*. Pearson Prentice Hall Upper Saddle River, NJ.

Sutskever, I., Józefowicz, R., Gregor, K., Rezende, D. J., Lillicrap, T. P., & Vinyals, O. (2015). Towards principled unsupervised learning. arXiv:1511.06440, CoRR abs/1511.06440.

Verma, V. K., & Khanna, N. (2013). Indian language identification using k-means clustering and support vector machine (SVM). In *2013 Students conference on engineering and systems* (pp. 1–5). IEEE.

Voxforge. org (2014). Free speech... Recognition (linux, windows and mac) - voxforge.org. http://www.voxforge.org/, (Accessed 25 June 2014).

Wang, H., Leung, C.-C., Lee, T., Ma, B., & Li, H. (2012). Shifted-delta MLP features for spoken language recognition. *IEEE Signal Processing Letters*, *20*(1), 15–18.

Wang, H., Leung, C., Lee, T., Ma, B., & Li, H. (2013). Shifted-delta MLP features for spoken language recognition. *IEEE Signal Processing Letters*, *20*(1), 15–18.

Wang, Q., Zheng, C., Yu, H., & Deng, D. (2015). Integration of heterogeneous classifiers based on choquet fuzzy integral. in *2015 7th international conference on intelligent human-machine systems and cybernetics* (vol. 1) (pp. 543–547).

Zhang Jian, B. X., Ruohua, Z., & Yonghong, Y. (2017). Weighted phone log-likelihood ratio feature for spoken language recognition. *Journal of Tsinghua University(Science and Technology)*, *57*(10), 1038. http://dx.doi.org/10.16511/j.cnki.qhdxxb.2017.25.042, http://jst.tsinghuajournals.com/EN/abstract/article_151999.shtml.

Zhou, X., Garcia-Romero, D., Duraiswami, R., Espy-Wilson, C., & Shamma, S. (2011). Linear versus mel frequency cepstral coefficients for speaker recognition. In *2011 IEEE workshop on automatic speech recognition & understanding* (pp. 559–564). IEEE.

Zissman, M. A., & Singer, E. (1994). Automatic language identification of telephone speech messages using phoneme recognition and N-gram modeling. In *Proceedings of ICASSP '94. IEEE international conference on acoustics, speech and signal processing* (vol. i) (pp. I/305–I/308).