Spoken Language Identification using ConvNets

Sarthak[1], Shikhar Shukla[2], and Govind Mittal[3]

[1] Analytics Quotient, Bangalore, India
sarthak.sfc@gmail.com, sarthak.j@aqinsights.com
[2] Samsung R&D Institute India-Bangalore, Bangalore, India
shikhar.00778@gmail.com, shikhar.0077@samsung.com
[3] Birla Institute of Technology & Science, Pilani, Rajasthan, India
f2014530@pilani.bits-pilani.ac.in

Abstract. Language Identification (LI) is an important first step in several speech processing systems. With a growing number of voice-based assistants, speech LI has emerged as a widely researched field. To approach the problem of identifying languages, we can either adopt an implicit approach where only the speech for a language is present or an explicit one where text is available with its corresponding transcript. This paper focuses on an implicit approach due to the absence of transcriptive data. This paper benchmarks existing models and proposes a new attention based model for language identification which uses log-Mel spectrogram images as input. We also present the effectiveness of raw waveforms as features to neural network models for LI tasks. For training and evaluation of models, we classified six languages (English, French, German, Spanish, Russian and Italian) with an accuracy of 95.4% and four languages (English, French, German, Spanish) with an accuracy of 96.3% obtained from the VoxForge dataset. This approach can further be scaled to incorporate more languages.

Keywords: Language Identification · Raw Waveform · Convolutional Neural Networks · Machine Learning.

1 Introduction

Language Identification (LI) is a problem which involves classifying the language being spoken by a speaker. LI systems can be used in call centers to route international calls to an operator who is fluent in that identified language [12]. In speech-based assistants, LI acts as the first step which chooses the corresponding grammar from a list of available languages for its further semantic analysis [1]. It can also be used in multi-lingual voice-controlled information retrieval systems, for example, Apple Siri and Amazon Alexa.

Over the years, studies have utilized many prosodic and acoustic features to construct machine learning models for LI systems [18]. Every language is composed of *phonemes*, which are distinct unit of sounds in that language, such as *b* of black and *g* of green. Several prosodic and acoustic features are based on phonemes, which become the underlying features on whom the performance of the statistical model depends [20,5]. If two languages have many overlapping phonemes, then identifying them becomes a challenging task for a classifier. For example, the word *cat* in English, *kat* in Dutch, *katze* in German have different consonants but when used in a speech they all would sound quite similar.

Due to such drawbacks several studies have switched over to using Deep Neural Networks (DNNs) to harness their novel auto-extraction techniques [1,19]. This work follows an implicit approach for identifying six languages with overlapping phonemes on the VoxForge [23] dataset and achieves 95.4% overall accuracy.

In previous studies [1,17,19], authors use log-Mel spectrum of a raw audio as inputs to their models. One of our contributions is to enhance the performance of this approach by utilising recent techniques like Mixup augmentation of inputs and exploring the effectiveness of *Attention* mechanism in enhancing performance of neural network. As log-Mel spectrum needs to be computed for each raw audio input and processing time for generating log-Mel spectrum increases linearly with length of audio, this acts as a bottleneck for these models. Hence, we propose the use of raw audio waveforms as inputs to deep neural network which boosts performance by avoiding additional overhead of computing log-Mel spectrum for each audio. Our 1D-ConvNet architecture auto-extracts and classifies features from this raw audio input.

The structure of the work is as follows. In Section 2 we discuss about the previous related studies in this field. The model architecture for both the raw waveforms and log-Mel spectrogram images is discussed in Section 3 along with the a discussion on hyperparameter space exploration. In Section 4 we present the experimental results. Finally, in Section 5 we discuss the conclusions drawn from the experiment and future work.

2 Related Work

Extraction of language dependent features like prosody and phonemes was a popular approach to classify spoken languages [29,16,6]. Following their success in speaker verification systems, i-vectors have also been used as features in various classification networks. These approaches required significant domain knowledge [4,16]. Nowadays most of the attempts on spoken language identification rely on neural networks for meaningful feature extraction and classification [15,7].

Revay et al. [19] used the ResNet50 [9] architecture for classifying languages by generating the log-Mel spectra of each raw audio. The model uses a cyclic learning rate where learning rate increases and then decreases linearly. Maximum learning rate for a cycle is set by finding the optimal learning rate using *fastai* [11] library. The model classified six languages – English, French, Spanish, Russian, Italian and German – and achieving an accuracy of 89.0%.

Gazeau et al. [8] in his research showed how Neural Networks, Support Vector Machine and Hidden Markov Model (HMM) can be used to identify French, English, Spanish and German. Dataset was prepared using voice samples from Youtube News [27]and VoxForge [23] datasets. Hidden Markov models convert speech into a sequence of vectors, was used to capture temporal features in speech. HMMs trained on VoxForge [23] dataset performed best in comparison to other models proposed by him on same VoxForge dataset. They reported an accuracy of 70.0%.

Bartz et al. [1] proposed two different hybrid Convolutional Recurrent Neural Networks for language identification. They proposed a new architecture for extracting spatial features from log-Mel spectra of raw audio using CNNs and then using RNNs for capturing temporal features to identify the language. This model achieved an accuracy of 91.0% on Youtube News Dataset [27]. In their second architecture they used the Inception-v3 [22] architecture to extract spatial features which were then used as input for bi-directional LSTMs to predict the language accurately. This model achieved an accuracy of 96.0% on four languages which were English, German, French and Spanish. They also trained their CNN model (obtained after removing RNN from CRNN model) and the Inception-v3 on their dataset. However they were not able to achieve better results achieving and reported 90% and 95% accuracies, respectively.

Kumar et al. [12] used Mel-frequency cepstral coefficients (MFCC), Perceptual linear prediction coefficients (PLP), Bark Frequency Cepstral Coefficients (BFCC) and Revised Perceptual Linear Prediction Coefficients (RPLP) as features for language identification. BFCC and RPLP are hybrid features derived using MFCC and PLP. They used two different models based on Vector Quantization (VQ) with Dynamic Time Warping (DTW) and Gaussian Mixture Model (GMM) for classification. These classification models were trained with different features. The authors were able to show that these models worked better with hybrid features (BFCC and RPLP) as compared to conventional features (MFCC and PLP). GMM combined with RPLP features gave the most promising results and achieved an accuracy of 88.8% on ten languages. They designed their own dataset comprising of ten languages being Dutch, English, French, German, Italian, Russian, Spanish, Hindi, Telegu, and Bengali.

Montavon [17] generated Mel spectrogram as features for a time-delay neural network (TDNN). This network had two-dimensional convolutional layers for feature extraction. An elaborate analysis of how deep architectures outperform their shallow counterparts is presented in this reseacrch. The difficulties in classifying perceptually similar languages like German and English were also put forward in this work. It is mentioned that the proposed approach is less robust to new speakers present in the test dataset. This method was able to achieve an accuracy of 91.2% on dataset comprising of 3 languages – English, French and German.

In Table 1, we summarize the quantitative results of the above previous studies. It includes the model basis, feature description, languages classified and the used dataset along with accuracy obtained. The table also lists the overall results of our proposed models (at the top). The languages used by various authors along with their acronyms are English (En), Spanish (Es), French (Fr), German (De), Russian (Ru), Italian (It), Bengali (Ben), Hindi (Hi) and Telegu (Tel).

Table 1: Quantitative Review of Previous Studies along with our Results.

Year	Model basis	Features	Languages	Acc.	Remarks	Ref.
2019	1D ConvNet	Raw Audio	En, Fr, De, Es, Ru, It	93.7^1	Evaulation of our 1D ConvNet model with mixup for six languages.	self
2019	2D ConvNet	log-Mel	En, Fr, De, Es, Ru, It	95.4^1	Evaulation of our 2D ConvNet model with mixup for six languages.	self
2019	2D ConvNet-Bi-directional GRU-Attention	log-Mel	En, Fr, De, Es, Ru, It	95.0^1	Result after tuning the hyperparameters of our cnn-bi-directional GRU-attention model and applying mixup	self
2019	2D ConvNet	log-Mel	En, Fr, De, Es	96.3^1	Our evaluation of 2D ConvNet model for four languages.	self
2019	ResNet50	log-Mel	En, Fr, De, Es, Ru, It	89.0^1	Uses a pretrained ResNet50 architecture and cyclic learner to identify the language.	[19]
2018	SVM-HMM model	not defined	En, Fr, Es, De	70.0^1	HMMs were used to encode speech into sequences of vectors which were then fed into a neural network.	[8]
2017	Inceptionv3 CRNN	log-Mel	En, Fr, De, Es	96.0^2	Used Inception-v3 model followed by bi-directional LSTMs to extract convolutional and temporal features.	[1]
2017	CRNN	log-Mel	En, Fr, De, Es	91.0^2	A new architecture is used to extract spatial features by using CNNs and temporal features using RNNs.	[1]
2010	Gaussian Mixture Models	Perceptual Linear Prediction	Dut, En, Fr, De, It, Ru, Es, Ben, Hi and Tel	88.8^3	Used Gaussian mixture models coupled with RPLP features, which were prepared using MFCC and PLP.	[12]
2009	CNN-TDNN	log-Mel	En, Fr , De	91.2^1	Used a time delay neural network with SGD was used to identify language using log-Mel images as input.	[17]

Dataset: ^{1} - VoxForge [23]; ^{2} - Youtube News [27], ^{3} - Private

3 Proposed Method

3.1 Motivations

Several state-of-the-art results on various audio classification tasks have been obtained by using log-Mel spectrograms of raw audio, as features [25]. Convolutional Neural Networks have demonstrated an excellent performance gain in classification of these features [26,10] against other machine learning techniques. It has been shown that using *attention* layers with ConvNets further enhanced their performance [13]. This motivated us to develop a CNN-based architecture with *attention* since this approach hasn't been applied to the task of language identification before.

Recently, using raw audio waveform as features to neural networks has become a popular approach in audio classification [24,13]. Raw waveforms have several artifacts which are not effectively captured by various conventional feature extraction techniques like Mel Frequency Cepstral Coefficients (MFCC), Constant Q Transform (CQT), Fast Fourier Transform (FFT), etc.

Audio files are a sequence of spoken words, hence they have temporal features too.A CNN is better at capturing spatial features only and RNNs are better at capturing temporal features as demonstrated by Bartz et al. [1] using audio files. Therefore, we combined both of these to make a CRNN model.

We propose three types of models to tackle the problem with different approaches, discussed as follows.

3.2 Description of Features

As an average human's voice is around 300 Hz and according to Nyquist-Shannon sampling theorem all the useful frequencies (0-300 Hz) are preserved with sampling at 8 kHz, therefore, we sampled *raw audio* files from all six languages at 8 kHz

The average length of audio files in this dataset was about *10.4 seconds* and standard deviation was *2.3 seconds*. For our experiments, the audio length was set to 10 seconds. If the audio files were shorter than 10 second, then the data was repeated and concatenated. If audio files were longer, then the data was truncated.

3.3 Model Description

We applied the following design principles to all our models:

- *Every convolutional layer is always followed by an appropriate max pooling layer*. This helps in containing the explosion of parameters and keeps the model small and nimble.
- *Convolutional blocks* are defined as an individual block with multiple pairs of one convolutional layer and one max pooling layer. *Each convolutional block is preceded or succeded by a convolutional layer.*

- *Batch Normalization and Rectified linear unit activations were applied after each convolutional layer*. Batch Normalization helps speed up convergence during training of a neural network.
- Model *ends with a dense layer* which acts the final output layer.

3.4 Model Details: 1D ConvNet

As the sampling rate is 8 kHz and audio length is 10 s, hence the input is *raw audio* to the models with input size of (batch size, 1, 80000). In Table 2, we present a detailed layer-by-layer illustration of the model along with its hyperparameter.

Table 2: Architecture of the 1D-ConvNet model

Layer Name	**# filters / kernel / stride**	**output**	**# of parameters**
Conv1	(128, 3, 3)	(128, 26664)	384
(Convolutional Block 1) Conv1D MaxPool1D Conv1D MaxPool1D	(128, 3, 1) (3, 3) (128, 3, 1) (3, 3)	(128, 26658) (128, 8880) (128, 8880) (128, 2960)	49152 49,152
Conv1D MaxPool1D	(256, 3, 1) (3, 3)	(256, 2954) (256, 984)	98,304
(Convolutional Block 2) Conv1D MaxPool1D	(256, 3, 1) (3, 3)	(256, 978) (256, 326)	196,608
Conv1D MaxPool1D	(512, 3, 1) (106, 3)	(512, 320) (512, 1)	393,216
Dense Layer	(512, 6)	(6)	3,072

Hyperparameter Optimization: Tuning hyperparameters is a cumbersome process as the hyperparamter space expands exponentially with the number of parameters, therefore efficient exploration is needed for any feasible study. We used the *random search* algorithm supported by *Hyperopt* [2] library to randomly search for an optimal set of hyperparameters from a given parameter space. In Fig. 1, various hyperparameters we considered are plotted against the validation accuracy as violin plots. Our observations for each hyperparameter are summarized below:

Number of filters in first layer: We observe that having 128 filters gives better results as compared to other filter values of 32 and 64 in the first layer. A higher number of filters in the first layer of network is able to preserve most of the characteristics of input.

Kernel Size: We varied the receptive fields of convolutional layers by choosing the kernel size from among the set of {3, 5, 7, 9}. We observe that a kernel size of 9 gives better accuracy at the cost of increased computation time and larger number of parameters. A large kernel size is able to capture longer patterns in its input due to bigger receptive power which results in an improved accuracy.

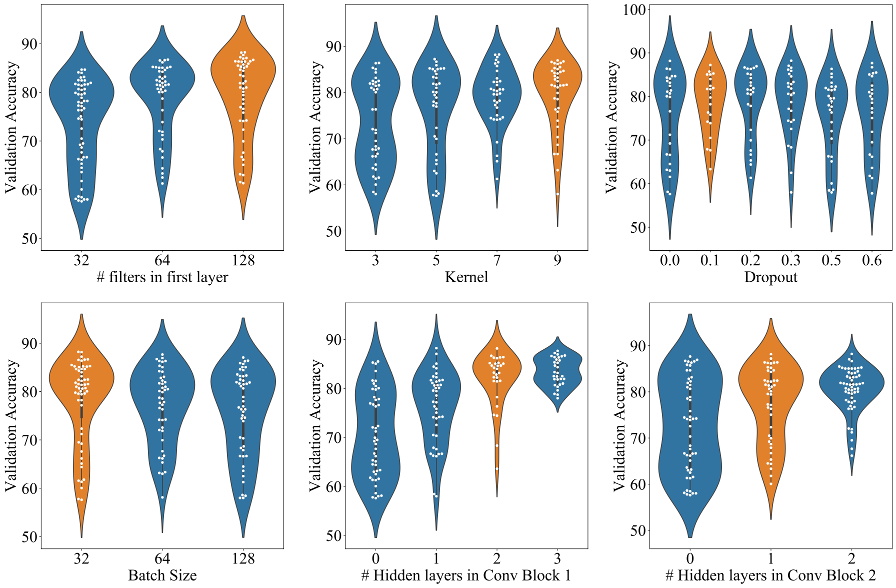


Fig. 1: Effect of hyperparameter variation of the hyperparameter on the classification accuracy for the case of 1D-ConvNet. Orange colored violin plots show the most favored choice of the hyperparameter and blue shows otherwise. One dot represents one sample.

Dropout: Dropout randomly turns-off (sets to 0) various individual nodes during training of the network. In a deep CNN it is important that nodes do not develop a co-dependency amongst each other during training in order to prevent overfitting on training data [21]. Dropout rate of 0.1 works well for our model. When using a higher dropout rate the network is not able to capture the patterns in training dataset.

Batch Size: We chose batch sizes from amongst the set {32, 64, 128}. There is more noise while calculating error in a smaller batch size as compared to a larger one. This tends to have a regularizing effect during training of the network and hence gives better results. Thus, batch size of 32 works best for the model.

Layers in Convolutional block 1 and 2: We varied the number of layers in both the convolutional blocks. If the number of layers is low, then the network does not have enough depth to capture patterns in the data whereas having large number of layers leads to overfitting on the data. In our network, two layers in the first block and one layer in the second block give optimal results.

3.5 Model Details: 2D ConvNet with Attention and bi-directional GRU

Log-Mel spectrogram is the most commonly used method for converting audio into the image domain. The audio data was again sampled at 8 kHz. The input to this model was the log-Mel spectra. We generated log-Mel spectrogram using the *LibROSA* [14] library. In Table 3, we present a detailed layer-by-layer illustration of the model along with its hyperparameter.

We took some specific design choices for this model, which are as follows:

- We added *residual connections* with each convolutional layer. Residual connections in a way makes the model selective of the contributing layers, determines the optimal number of layers required for training and solves the problem of vanishing gradients. Residual connections or skip connections skip training of those layers that do not contribute much in the overall outcome of model.

Table 3: Architecture of the 2D-ConvNet model

Layer Name	Output features	Number of filters / stride / padding	No. of parameters
(ConvBlock 1)			
Conv2D	(64, 128, 128)	(3, 3) / (1, 1) / (1, 1)	1,728
Conv2D	(64, 128, 128)	(3, 3) / (1, 1) / (1, 1)	36,864
AvgPool2D	(64, 64, 64)		
(ConvBlock 2)			
Conv2D	(128, 64, 64)	(3, 3) / (1, 1) / (1, 1)	73,728
Conv2D	(128, 64, 64)	(3, 3) / (1, 1) / (1, 1)	147,456
AvgPool2D	(128, 32, 32)		
(ConvBlock 3)			
Conv2D	(256, 32, 32)	(3, 3) / (1, 1) / (1, 1)	294,912
Conv2D	(256, 32, 32)	(3, 3) / (1, 1) / (1, 1)	589,824
AvgPool2D	(256, 16, 16)		
(ConvBlock 4)			
Conv2D	(512, 16, 16)	(3, 3) / (1, 1) / (1, 1)	1,179,648
Conv2D	(512, 16, 16)	(3, 3) / (1, 1) / (1, 1)	235,929
AvgPool2D	(512, 8, 8)		
Bi-directional GRU	(8, 1536)		1,769,472
Embedding Layer	(8, 768)		1,179,648
(Sequential Block)			
Dropout (0.2)	(256)		131,072
Linear			
Dropout (0.1)	(6)		1,536
Linear			

- We added *spatial attention* [3] networks to help the model in focusing on specific regions or areas in an image. Spatial attention aids learning irrespective of transformations, scaling and rotation done on the input images making the model more robust and helping it to achieve better results.
- We added *Channel Attention* networks so as to help the model to find interdependencies among color channels of log-Mel spectra. It adaptively assigns importance to each color channel in a deep convolutional multi-channel network. In our model we apply channel and spatial attention just before feeding

the input into bi-directional GRU. This helps the model to focus on selected regions and at the same time find patterns among channels to better determine the language.

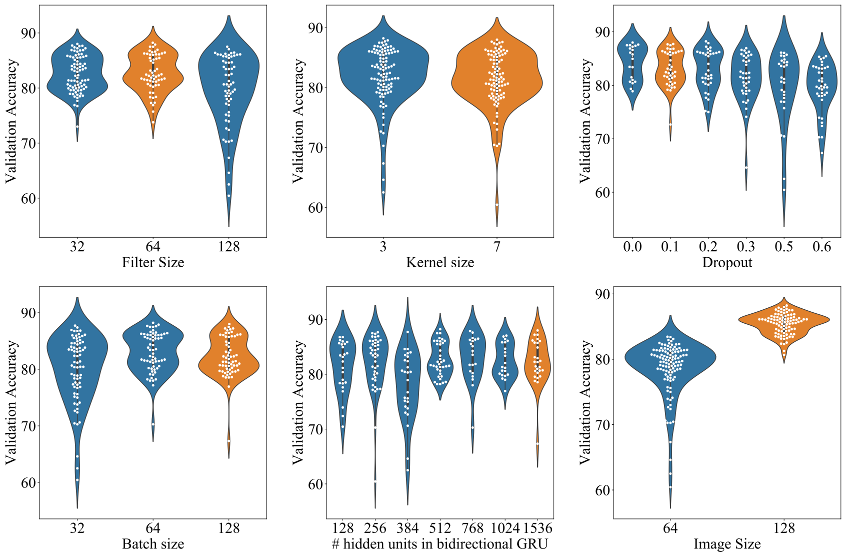


Fig. 2: Effect of hyperparameter variation of the six selected hyperparameter on the classification accuracy for the case of 2D-ConvNet. Orange colored violin plots show the most favored choice of the hyperparameter and blue shows otherwise. One dot represents one sample.

Hyperparameter Optimization: We used the *random search* algorithm supported by *Hyperopt* [2] library to randomly search for an optimal set of hyperparameters from a given parameter space. In Fig. 2 ,various hyperparameters we tuned are plotted against the validation accuracy. Our observations for each hyperparameter are summarized below:

Filter Size: 64 filters in the first layer of network can preserve most of the characteristics of input, but increasing it to 128 is inefficient as overfitting occurs.

Kernel Size: There is a trade-off between kernel size and capturing complex non-linear features. Using a small kernel size will require more layers to capture features whereas using a large kernel size will require less layers. Large kernels capture simple non-linear features whereas using a smaller kernel will help us capture more complex non-linear features. However, with more layers, backpropagation necessitates the need for a large memory. We experimented with large kernel size and gradually increased the layers in order to capture more complex features. The results are not conclusive and thus we chose kernel size of 7 against 3.

Dropout: Dropout rate of 0.1 works well for our data. When using a higher dropout rate the network is not able to capture the patterns in training dataset.

Batch Size: There is always a trade-off between batch size and getting accurate gradients. Using a large batch size helps the model to get more accurate gradients since the model tries to optimize gradients over a large set of images. We found that using a batch size of 128 helped the model to train faster and get better results than using a batch size less than 128.

Number of hidden units in bi-directional GRU: Varying the number of hidden units and layers in GRU helps the model to capture temporal features which can play a significant role in identifying the language correctly. The optimal number of hidden units and layers depends on the complexity of the dataset. Using less number of hidden units may capture less features whereas using large number of hidden units may be computationally expensive. In our case we found that using 1536 hidden units in a single bi-directional GRU layer leads to the best result.

Image Size: We experimented with log-Mel spectra images of sizes 64 × 64 and 128 × 128 pixels and found that our model worked best with images of size of 128 × 128 pixels.

We also evaluated our model on data with mixup augmentation [28]. It is a data augmentation technique that also acts as a regularization technique and prevents overfitting. Instead of directly taking images from the training dataset as input, mixup takes a linear combination of any two random images and feeds it as input. The following equations were used to prepared a mixed-up dataset:

$$\text{Input\_Image} = \alpha * I_1 + (1 - \alpha) * I_2, \tag{1}$$

and

$$\text{Input\_Label} = \alpha * L_1 + (1 - \alpha) * L_2, \tag{2}$$

where $\alpha \in [0, 1]$ is a random variable from a β-distribution, I_1.

3.6 Model details: 2D-ConvNet

This model is a similar model to 2D-ConvNet with Attention and bi-directional GRU described in section 3.5 except that it lacks skip connections, attention layers, bi-directional GRU and the embedding layer incorporated in the previous model.

3.7 Dataset

We classified six languages (English, French, German, Spanish, Russian and Italian) from the VoxForge [23] dataset. VoxForge is an open-source speech corpus which primarily consists of samples recorded and submitted by users using their own microphone. This results in significant variation of speech quality between samples making it more representative of real world scenarios.

Our dataset consists of 1,500 samples for each of six languages. Out of 1,500 samples for each language, 1,200 were randomly selected as training dataset for that language and rest 300 as validation dataset using k-fold cross-validation. To sum up, we trained our model on 7,200 samples and validated it on 1800 samples comprising six languages. The results are discussed in next section.

4 Results and Discussion

This paper discusses two end-to-end approaches which achieve state-of-the-art results in both the image as well as audio domain on the VoxForge dataset [23]. In Table 4, we present all the classification accuracies of the two models of the cases with and without mixup for six and four languages.

In the audio domain (using raw audio waveform as input), 1D-ConvNet achieved a mean accuracy of 93.7% with a standard deviation of 0.3% on running k-fold cross validation. In Fig 3 (a) we present the confusion matrix for the 1D-ConvNet model.

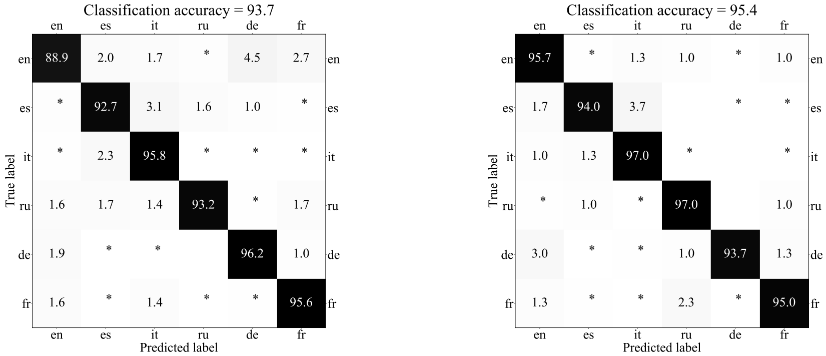
In the image domain (obtained by taking log-Mel spectra of raw audio), 2D-ConvNet with 2D attention (channel and spatial attention) and bi-directional GRU achieved a mean accuracy of 95.0% with a standard deviation of 1.2% for six languages. This model performed better when mixup regularization was applied. 2D-ConvNet achieved a mean accuracy of 95.4% with standard deviation of 0.6% on running k-fold cross validation for six languages when mixup was applied. In Fig 3 (b) we present the confusion matrix for the 2D-ConvNet model. 2D attention models focused on the important features extracted by convolutional layers and bi-directional GRU captured the temporal features.

Table 4: Results of the two models and all its variations

Languages	**Feature Desc.**	**Network**	**Mixup**	**Accuracy**
En, Es, Fr, De, Ru, It	Raw Waveform	1D ConvNet	No	93.7
	log-Mel Spectra	2D ConvNet	No	94.3
			Yes	95.4
		2D ConvNet with Attention and GRU	No	94.3
			Yes	95.0
En, Es, Fr, De	Raw Waveform	1D ConvNet	No	94.4
	log-Mel Spectra	2D ConvNet	No	96.0
			Yes	96.3
		2D ConvNet with Attention and GRU	No	94.7
			Yes	93.7

Misclassification Several of the spoken languages in Europe belong to the Indo-European family. Within this family, the languages are divided into three phyla which are Romance, Germanic and Slavic. Of the 6 languages that we selected Spanish (Es), French (Fr) and Italian (It) belong to the Romance phyla, English and German belong to Germanic phyla and Russian in Slavic phyla. Our model also confuses between languages belonging to the similar phyla which acts as an insanity check since languages in same phyla have many similar pronounced words such as *cat* in English becomes *Katze* in German and *Ciao* in Italian becomes *Chao* in Spanish.

Our model confuses between French (Fr) and Russian (Ru) while these languages belong to different phyla, many words from French were adopted into Russian such as automate (oot-oo-mate) in French becomes ABTOMaT (aff-taa-maat) in Russian which have similar pronunciation.



	en	es	it	ru	de	fr
en	88.9	2.0	1.7	*	4.5	2.7
es	*	92.7	3.1	1.6	1.0	*
it	*	2.3	95.8	*	*	*
ru	1.6	1.7	1.4	93.2	*	1.7
de	1.9	*	*		96.2	1.0
fr	1.6	*	1.4	*	*	95.6

(a)

	en	es	it	ru	de	fr
en	95.7	*	1.3	1.0	*	1.0
es	1.7	94.0	3.7		*	*
it	1.0	1.3	97.0	*		*
ru	*	1.0	*	97.0		1.0
de	3.0	*	*	1.0	93.7	1.3
fr	1.3	*	*	2.3	*	95.0

(b)

Fig. 3: Confusion matrix for classification of six languages with our (a) 1D-ConvNet and (b) 2D-ConvNet model. Asterisk (*) marks a value less than 0.1%.

Future Scope The performance of raw audio waveforms as input features to ConvNet can be further improved by applying silence removal in the audio. Also, there is scope for improvement by augmenting available data through various conventional techniques like pitch shifting, adding random noise and changing speed of audio. These help in making neural networks more robust to variations which might be present in real world scenarios. There can be further exploration of various feature extraction techniques like Constant-Q transform and Fast Fourier Transform and assessment of their impact on Language Identification.

There can be further improvements in neural network architectures like concatenating the high level features obtained from 1D-ConvNet and 2D-ConvNet, before performing classification. There can be experiments using deeper networks with skip connections and Inception modules. These are known to have positively impacted the performance of Convolutional Neural Networks.

5 Conclusion

There are two main contributions of this paper in the domain of spoken language identification. Firstly, we presented an extensive analysis of raw audio waveforms as input features to 1D-ConvNet. We experimented with various hyperparameters in our 1D-ConvNet and evaluated their effect on validation accuracy. This method is able to bypass the computational overhead of conventional approaches which depend on generation of spectrograms as a necessary pre-procesing step. We were able to achieve an accauracy of **93.7%** using this technique.

Next, we discussed the enhancement in performance of 2D-ConvNet using mixup augmentation, which is a recently developed technique to prevent overfitting on test data.This approach achieved an accuracy of **95.4%**. We also analysed how *attention* mechanism and recurrent layers impact the performance of networks. This approach achieved an accuracy of **95.0%**.

References

1. Bartz, C., Herold, T., Yang, H., Meinel, C.: Language identification using deep convolutional recurrent neural networks. In: International Conference on Neural Information Processing. pp. 880–889. Springer (2017)
2. Bergstra, J., Yamins, D., Cox, D.D.: Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures (2013)
3. Chen, L., Zhang, H., Xiao, J., Nie, L., Shao, J., Liu, W., Chua, T.S.: Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5659–5667 (2017)
4. Dehak, N., Torres-Carrasquillo, P.A., Reynolds, D., Dehak, R.: Language recognition via i-vectors and dimensionality reduction. In: Twelfth annual conference of the international speech communication association (2011)
5. Endah Safitri, N., Zahra, A., Adriani, M.: Spoken language identification with phonotactics methods on minangkabau, sundanese, and javanese languages. Procedia Computer Science **81**, 182–187 (12 2016). https://doi.org/10.1016/j.procs.2016.04.047
6. Ferrer, L., Scheffer, N., Shriberg, E.: A comparison of approaches for modeling prosodic features in speaker recognition. In: 2010 IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 4414–4417. IEEE (2010)
7. Ganapathy, S., Han, K., Thomas, S., Omar, M., Segbroeck, M.V., Narayanan, S.S.: Robust language identification using convolutional neural network features. In: Fifteenth annual conference of the international speech communication association (2014)
8. Gazeau, V., Varol, C.: Automatic spoken language recognition with neural networks. Int. J. Inf. Technol. Comput. Sci.(IJITCS) **10**(8), 11–17 (2018)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90, `https://doi.org/10.1109/CVPR.2016.90`
10. Hershey, S., Chaudhuri, S., Ellis, D.P., Gemmeke, J.F., Jansen, A., Moore, R.C., Plakal, M., Platt, D., Saurous, R.A., Seybold, B., et al.: Cnn architectures for large-scale audio classification. In: 2017 ieee international conference on acoustics, speech and signal processing (icassp). pp. 131–135. IEEE (2017)
11. Howard, J., et al.: fastai. `https://github.com/fastai/fastai` (2018)
12. Kumar, P., Biswas, A., Mishra, A.N., Chandra, M.: Spoken language identification using hybrid feature extraction methods. arXiv preprint arXiv:1003.5623 (2010)

13. Lee, J., Kim, T., Park, J., Nam, J.: Raw waveform-based audio classification using sample-level cnn architectures. arXiv preprint arXiv:1712.00866 (2017)
14. LibROSA: https://librosa.github.io/librosa/. `https://librosa.github.io/librosa/`, accessed on 16 Jul 2019
15. Lopez-Moreno, I., Gonzalez-Dominguez, J., Plchot, O., Martinez, D., Gonzalez-Rodriguez, J., Moreno, P.: Automatic language identification using deep neural networks. In: 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP). pp. 5337–5341. IEEE (2014)
16. Martinez, D., Plchot, O., Burget, L., Glembek, O., Matějka, P.: Language recognition in ivectors space. In: Twelfth Annual Conference of the International Speech Communication Association (2011)
17. Montavon, G.: Deep learning for spoken language identification. In: NIPS Workshop on deep learning for speech recognition and related applications. pp. 1–4 (2009)
18. Obuchi, Y., Sato, N.: Language identification using phonetic and prosodic hmms with feature normalization. In: Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005. vol. 1, pp. I–569. IEEE (2005)
19. Revay, S., Teschke, M.: Multiclass language identification using deep learning on spectral images of audio signals. arXiv preprint arXiv:1905.04348 (2019)
20. Rong Tong, Bin Ma, Donglai Zhu, Haizhou Li, Eng Siong Chng: Integrating acoustic, prosodic and phonotactic features for spoken language identification. In: 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings. vol. 1, pp. I–I (May 2006). https://doi.org/10.1109/ICASSP.2006.1659993
21. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research **15**(1), 1929–1958 (2014)
22. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016)
23. voxforge.org: Free speech recognition (linux, windows and mac) - voxforge.org. `http://www.voxforge.org/`, accessed on 16 Jul 2019
24. WEI, Q., LIU, Y., RUAN, X.: A report on audio tagging with deeper cnn, 1d-convnet and 2d-convnet
25. Xu, K., Zhu, B., Kong, Q., Mi, H., Ding, B., Wang, D., Wang, H.: General audio tagging with ensembling convolutional neural networks and statistical features. The Journal of the Acoustical Society of America **145**(6), EL521–EL527 (2019)
26. Xu, Y., Huang, Q., Wang, W., Foster, P., Sigtia, S., Jackson, P.J., Plumbley, M.D.: Unsupervised feature learning based on deep models for environmental audio tagging. IEEE/ACM Transactions on Audio, Speech, and Language Processing **25**(6), 1230–1241 (2017)
27. Youtube: Retrieved from www.youtube.com. `http://www.youtube.com`, accessed on 16 Jul 2019
28. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017)
29. Zissman, M.A.: Comparison of four approaches to automatic language identification of telephone speech. IEEE Transactions on speech and audio processing **4**(1), 31 (1996)