Language Identification using Deep Learning

Chaitanya Patil, Siddharth Kothari, Vishal Singh (chpatil, sidkotha, singhvis)

Abstract
Language Identification(LID) systems are used to classify the spoken language from a given audio sample and are typically the first step for many spoken language processing tasks, such as Automatic Speech Recognition (ASR) systems. Without automatic language detection, speech utterances cannot be parsed correctly and grammar rules cannot be applied, causing subsequent speech recognition steps to fail.We have implemented CRNN architecture used in the previous research with a few modifications of our own to explore the architecture through various experiments. We checked the performance with 3 different spectrogram representation techniques. Robustness of model was also tested with 10 different noises and 2 different loudness levels. [Christian Bartz and Meinel(2017)]

Motivation

Language need to be selected explicitly before for intelligent assistants like Siri, Alexa or the Google Assistant to work properly. With globalization, there is constant increase of multi lingual household which can led to a inferior user experience. Language Identification (LID) provides a option to mitigate this problem by recognizing the spoken language which can then be used communicate or execute the voice commands in different languages.

1. Introduction

The advancement of DNN and higher performance achieved through it has led to it's application to various problems across different fields and one such area is Automatic Speech Recognition (ASR) and specifically Language Identification (LID). Problems using audio signals can be solved by creating an image version of them i.e. by converting it in the time-frequency domain [Fred Richardson(2015)]. This allows machine to learn features on it's own saving both the time and resources used in traditional systems where domain-specific expert knowledge was used to extract hand-crafted features from the audio samples. In this project, we use a hybrid network architecture with Recurrent Neural Network (RNN) stacked over Convolutional Neural Network (CNN).

2. Related Work

In last few years, neural network is used to extract features. CNN and LSTM are most popular especially for LID as they produce a higher accuracy as compared to traditional approaches and are also simpler in design.

2.1 CRNN Network

An architecture used by
[Christian Bartz and Meinel(2017)]
consist of two parts. For the first part, audio is represented in spectrogram image to be used as an input to a convolutional feature extractor. They used this feature extractor to convolve, in multiple steps, the input image into a feature map with a height of one. The feature map is sliced along the x-axis and each slice is used as a time step for the subsequent BLSTM network. The design of the convolutional feature extractor is based on the well known VGG architecture. For better feature extraction, they used deeper convolution through Inception and Inceptionv3.

They performed the language identification on EU Speech Repository data, which involved speech recorded by one person in the native tongue and recording quality was high. The data was tested on Youtube News and noise where they got reasonably high accuracy by using Inception and Incpetionv3

Our architecture is similar to the mentioned work but the entire process varies in the following way: (1) We used MFCC, STFT and Mel Frequency to reduce the number of features of the input spectrogram image for faster configuration (2) We used different dataset which includes data from different accents and demographics and lower quality recording which made it similar to real world scenario.

3. Proposed System

In our work, we use spectrogram representation techniques to convert the audio clip into time-frequency domain. The techniques also help in reducing the dimensions and generate an average features for the audio. This audio is passed through CNN which is used to capture spatial information, and then to RNN to capture information through a sequence of time steps for identifying the languages. In this section, we present the datasets we used for training the network, the audio representation used for training our models, and the structure of our proposed network in detail.

3.1 Dataset

We have used Common Voice's multi-language dataset, which is the largest publicly available voice dataset of its kind [data(2019)]. It has voice data for around 20 languages. The data is publicly

collected and has different demographic and accents(Table 1). We have used Catalan, German, English, French and Italian Languages. French, Italian, and Catalan are all Romance language and German, and English are both Germanic language. We choose two sets of similar languages to see how well model differentiate between languages from same family and language of different family. We took 2500 mp3 audio files for each language and converted into wav format. These 2500 files varies from 5 second to 10 seconds. We have used 10 different kind of noises (birds, casino, cicadas, computer keyboard, eating chips, frogs, jungle, machine gun, motorcycles, and oceans). We have created two versions of noise one with loudness decreased by 10dB and one with 30dB. We have combined audio file with these two noise files.

3.2 Audio Representation

We convert our audio data to spectrogram representations for training our models. For experimentation, we have used MFCC, STFT and Mel Scale to discretize the spectograms and evaluated the performance of the network.

3.2.1 STFT

The short-time Fourier transform (STFT) (Figure 1), is a Fourier-related transform used to determine the sinusoidal frequency and phase content of local sections of a signal as it changes over time.In practice, the procedure for computing STFTs is to divide a longer time signal into shorter segments of equal length and then compute the Fourier transform separately on each shorter segment. This reveals the Fourier spectrum on each shorter segment.

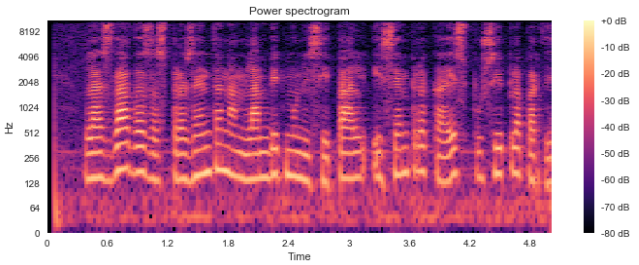

Figure 1. Spectrogram in STFT

3.2.2 MFCC

Mel-frequency cepstral coefficients (MFCCs) (Figure 2) are coefficients that collectively make up an MFC. The mel frequency cepstral coefficients of a signal are a small set of features (usually about 10-20) which concisely describe the overall shape of a spectral envelope. In MIR, it is often used to describe timbre. We have taken 150 mfcc for each audio file with sampling rate of 22050.

3.2.3 Mel Scale

The Mel(Figure 3) , named by Stevens, Volkmann, and Newman in 1937, is a perception criterion for pitches which is claimed by audience to be heard from equal distances. The

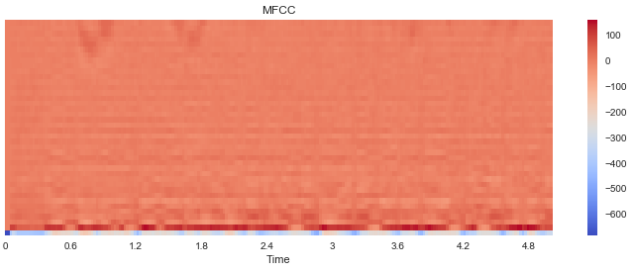

Figure 2. Spectrogram in MFCC

name Mel comes from melody to indicate the fact that scale is based on pitch comparisons.

The formula for converting Hertz to Mel is as follows:

$$Mel = 1127 \times \log_e(1 + \frac{f}{700}) \quad (1)$$

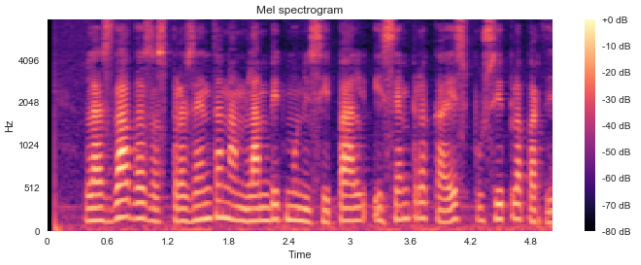

Figure 3. Spectrogram in Mel-Frequency scale

4. Architecture

For our network architecture, we followed the overall structure of the network proposed by [Christian Bartz and Meinel(2017)] in their similar work on language identification. This network architecture consists of two parts. The first part is a convolutional feature extractor that takes a spectrogam image representation of the audio file's input. This feature extractor convolves the input image in several steps and produces a feature map with very few features. Basic architecture is as given in Figure 4

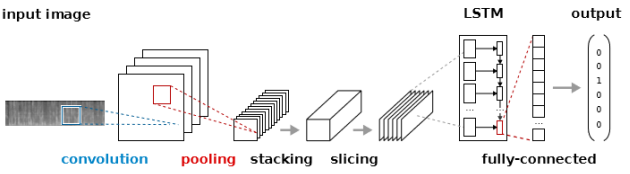

Figure 4. CRNN Architecture

This feature map is sliced along the y-axis and each slice is used as a time step for the subsequent LSTM network. The architecture of [Christian Bartz and Meinel(2017)] has BLSTM, we are using LSTM after convolution network.

5. Experiments and Results

	Accent(Top 2)	Age (Top 2)	Gender
English	21% United States English, 8% England English	18% 19-29, 14% 30-39	41% M, 10% F
German	71% Deutschland Deutsch, 3% Schweizerdeutsch	31% 19-29, 17% 40-49	76% M, 5% F
French	74% Français de France, 2% Français de Belgique	25% 40-49, 23% 19-29	72% M, 7% F
Catalan	69% central, 7% valencià	28% 40-49, 26% 50-59	44% F, 38% M
Italian	-	37% 19-29, 13% 30-39	67% M, 7% F

Table 1. Language Data Distribution

5.1 Experiments

We extracted 3 types of features which have been mentioned in Section 3.2. The audio clips which we used were in the range of 5-10 seconds. To make the data of uniform length we padded the shorter data with zeros to convert each input of uniform length

We addressed the following question in our experiments-

1. Can we increase the classification accuracy with the help of network introduced in section 4.
2. How robust the model is to noise data?
3. How do different features perform with our proposed architecture?

5.1.1 MFCC Features

After padding the data with zeros to make the input of uniform length we extracted 40 mfccs from the audio. We used the *librosa.feature.mfcc* function in python to extract the features. 40 mfccs were chosen to not make the size of data too huge. The dimension of each input file is 972x40 (timexmfcc). We divided the 2500 audio file for each language into train:2000 and test:500 and trained the CRNN.

5.1.2 STFT Features

To avoid padding, we used 5 sec audio clips to extracting STFT features so the dimensionality of the input is reduced without affecting quality . We used the *librosa.stft* function in python to extract the features. We set *n_stft=256*. The dimension of each input file is 247x129. We followed the same distribution for train and test as in section *5.1.1*

5.1.3 Mel-Scale Features

For extracting Mel-Scale features we used 5 sec audio clips for similar reasons as STFT features. We used the function *librosa.features.melspectrogram* in python to extract the features. The dimension of each input file is 242x128. We followed the same distribution for train and test as in section *5.1.1*

Since we obtained the best accuracy using the MFCC features, we proceeded to test the robustness of our proposed network on noisy data. Noise was added to the data by combining original audio with noise audio clips. Two types of noisy data was produced-

- Original Audio + Noise Data, where loudness of noise data was reduced by 10 db
- Original Audio + Noise Data, where loudness of noise data was reduced by 30 db

We tested the accuracy by running the 30db and 10db noisy data on the original network. Then we proceeded to train the network using the two noisy data and check the accuracy of the other noisy data on the model trained using the first noisy data. We had the following hypotheses:

- The 30db noisy data would perform better than the 10db noisy data on the original network
- The model trained using 30db noisy data would have a better accuracy in classifying language than the 10db noisy data
- The 30db noisy data would perform better on the model trained using 10db noisy data then 10db noisy data would on the model trained using 30db noisy data

We also trained networks by replacing the CNN part of our network by MobileNet [Andrew G. Howard(2017)] and Inceptionv3 [Tsang(2018)] architectures followed by LSTM and checked accuracy with the original data.

5.2 Evaluation Criteria

We checked the accuracy by the checking the percentage of number of labels correctly classified. The *loss function* used by us is *categorical crossentropy* which penalizes if the label predicted by the network does not match the actual label.

5.3 Results

The test accuracy obtained for different dataset using the 3 types features are as follows-

Features	Original Audio	Louder Noise(-10db)	Softer Noise(-30db)
MFCC	0.914	0.839	0.875
STFT	0.776	-	-
Mel	0.692	-	-

The confusion matrix obtained for the 5 labels when testing for data with no noise is as shown in Figure 5. We can observe that 'English' is predicted well while 'French' and 'German' are predicted as each other in many case. Similarly Catalan is also not predicted well. This could be attributed to the fact the the audio files for English were less noisy and had speech in the entirety of their audio file while it was not the case with the other languages. Similar results can be observed when testing with data with the two types of noise.

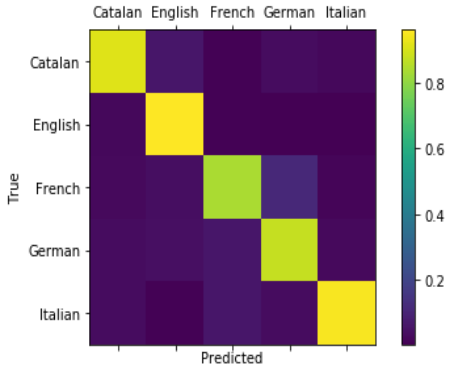

Figure 5. MFCC Confusion Matrix - Data w/o Noise

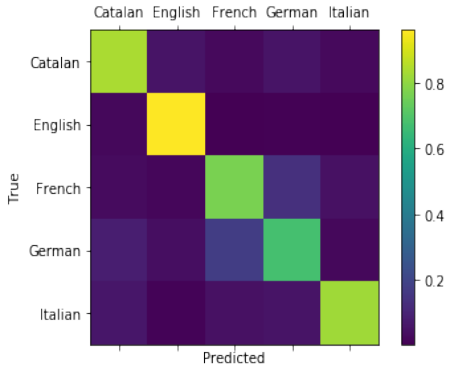

Figure 6. MFCC Confusion Matrix - Data w Noise(-10db)

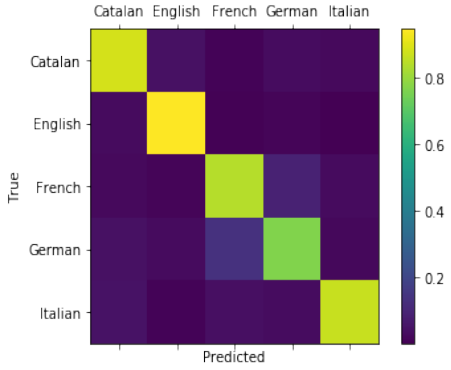

Figure 7. MFCC Confusion Matrix - Data w Noise(-30db)

When testing for audio with different levels of noise, we obtained the following results:

- When we tested for sound with louder noise against the model trained with softer noise, we obtained an accuracy of 82.24%
- When we tested for sound with softer noise against the model trained with louder noise, we obtained an accuracy of 87.91%

6. Summary and Conclusions

In this project we have proposed a language identification system which aims at identifying the language used in an audio signal by converting the problem in image domain. Deriving from the original paper [Christian Bartz and Meinel(2017)] we used our CRNN network to classify language from audio after extracting different different kinds of features. The idea is to test the architecture for which [Christian Bartz and Meinel(2017)] were getting high accuracy on data with different accents, demographic of speaker and low quality recordings. We also used different techniques to convert audio to spectrogram image. Even with such a varied data, we obtained an accuracy of 91% which is similar to the accuracy they got for better quality YouTube News data.

References

[Andrew G. Howard(2017)] Bo Chen Dmitry Kalenichenko Weijun Wang Tobias Weyand Marco Andreetto Hartwig Adam Andrew G. Howard, Menglong Zhu. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017. URL https://arxiv.org/abs/1704.04861.

[Christian Bartz and Meinel(2017)] Haojin Yang Christian Bartz, Tom Herold and Christoph Meinel. Language identication using deep convolutional recurrent neural network, 2017. URL https://arxiv.org/abs/1708.04811.

[data(2019)] data. Dataset source, 2019. URL https://voice.mozilla.org/en/datasets.

[Fred Richardson(2015)] Najim Dehak Fred Richardson, Douglas Reynolds. Deep neural network approaches to speaker and language recognition. 2015. URL https://ieeexplore.ieee.org/document/7080838/authors#authors.

[Tsang(2018)] Sik-Ho Tsang. Review: Inception-v3—1st runner up (image classification) in ilsvrc 2015, 2018. URL https://medium.com/@sh.tsang/review-inception-v3-1st-runner-up-image-classif