Language Identification Using CRNN

Chaitanya Patil, Siddharth Kothari, Vishal Singh
School of Informatics, Computing and Engineering, Indiana University

ABSTRACT

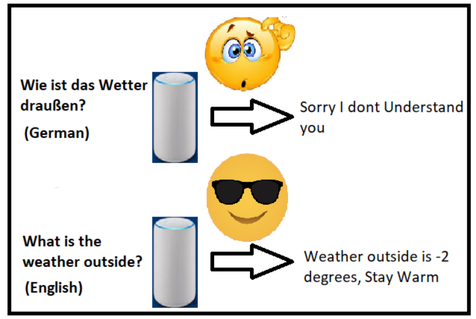
Language Identification (LID) systems are used to classify the spoken language from a given audio sample and are typically the first step for many spoken language processing tasks, such as Automatic Speech Recognition (ASR) systems. Without automatic language detection, speech utterances cannot be parsed 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.

REFERENCES

[1] I. Lopez-Moreno, J. Gonzalez-Dominguez, O. Plchot, D. Martinez, J. Gonzalez - Rodriguez, P. Moreno, "Automatic language identification using deep neural networks", *Proc. ICASSP*, pp. 5374-5378, 2014.
[2] F. Richardson, D. Reynolds, N. Dehak, "Deep neural network approaches to speaker and language recognition", *IEEE SIGNAL PROCESSING LETTERS, VOL. 22, NO. 10,* OCTOBER 2015
[3] Y. Zhao, X. Jin, and X. Hu. Recurrent convolutional neural network for speech processing. In IEEE International Conference on Acoustics, Speech and Signal Processing, 2017.

MOTIVATION

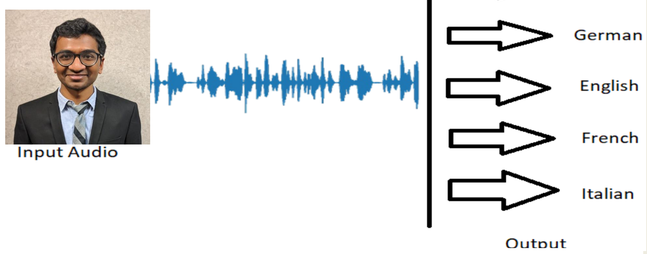
- Language Identification is the first step of ASR. All the voice assistants mostly have English as default language and it requires manual setting to change the language so that it can respond better.



- How would it be if the voice assistant responds to voice commands irrespective of language?

DATASETS

- We used 2500 audio files for each of 5 languages Catalan, German, English, French and Italian collected from https://voice.mozilla.org/en/datasets
- Dataset include speech data from people with different demographic background such as age, sex, and accent that can help train the accuracy of language identification of recognition engines.

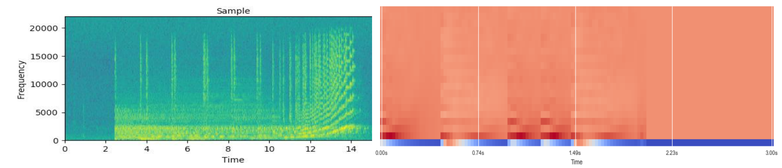


- The dataset contains 35 hours of data (each file 10/5 seconds). We have converted mp3 files into wave format which are later converted into spectrogram arrays padded(for 10 sec data) with zeros so that the dimensions of all the training samples are equal.

FEATURE EXTRACTION

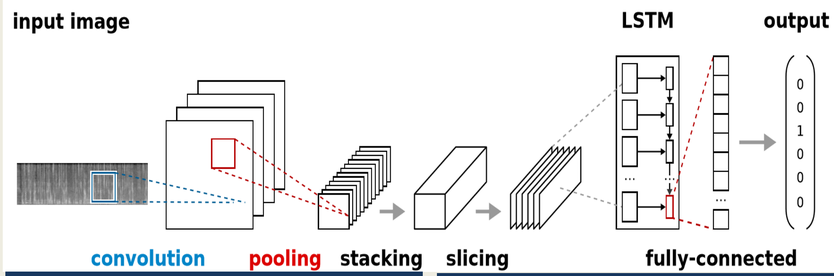
We tried 2 approaches to convert the audio file into features which are used for classification:

- Mel Frequency Cepstrum Coefficient: A feature widely used in automatic speech and speaker recognition. Number of MFCC which were taken for the for training the NN model are 40. Returned size of input: [972 x 40]
- Short-term Fourier transform: Number of FFT considered were 256. Returned size of input: [247 x 129] (5 second audio file)



ARCHITECTURE

Sound events often occur in unstructured environments where they exhibit wide variations in their frequency content and temporal structure.
Convolutional neural networks (CNN) are able to extract higher level features that are invariant to local spectral and temporal variations.
Recurrent neural networks (RNNs) are powerful in learning the longer term temporal context in the audio signals.
CNNs and RNNs as classifiers have recently shown improved performances over established methods in various sound recognition tasks. We combine these two approaches in a Convolutional Recurrent Neural Network (CRNN) and apply it for Language Identification task



CNN

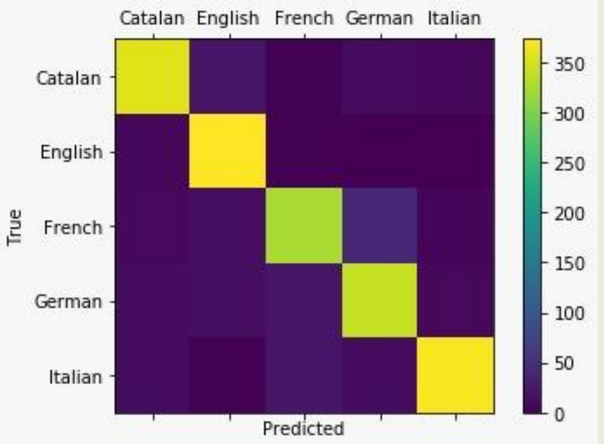
The first part is a convolutional feature extractor that takes a spectrogram image representation of the audio file as input . This feature extractor convolves the input image in several steps and produces a feature map which is fed to the LSTM network.

LSTM

The output of the above convolutional layer f × c is sliced along the time axis into t time steps. Each time step represents the extracted frequency features, used as input to the LSTM

RESULTS

	MFCC Features	STFT Features
Tuned CRNN	91%	71%
Mobile Net	74%	45%



DISCUSSION

MFCC Features (Dimension: 972 x 40)

- Obtained a cross-validation accuracy of ~91% using MFCC features, this could be due to the fact that our dataset has varied data.
- Obtained a CV accuracy of ~75% using MFCC features on training using the noise data

STFT Features (Dimension: 247 x 129)

- Obtained a CV accuracy of ~71%, this could be due to the fact that time stamps selected for training is very less (247)
- Using MobileNet and InceptionV3 NN did not produce any significantly better results

FUTURE WORK

1. Training the model on larger data using more features extracted using both MFCC and STFT
2. Apply deeper convolutions on the dataset with more features
3. Expand the model to incorporate more languages