Available online www.jsaer.com

***Journal of Scientific and Engineering Research*, 2020, 7(9):135-137**

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

Dataset Generation Tool for Language Identification Systems that Use Deep Convolutional Recurrent Neural Networks

Varuzhan H. Baghdasaryan

Bachelor of Computer Systems and Informatics, National Polytechnic University of Armenia, Armenia.
www.varuzh2014@gmail.com

Abstract Datasets are an integral part of the field of machine learning. The purpose of the article is to design a tool that can generate a human-labelled dataset from the environment (both noisy and clear) for language identification (LID) systems that use deep convolutional recurrent neural networks (CRNN). The article describes the general principles and steps of data collection, processing, dataset generation, the problems that arise as a result of these stages and the ways to solve them.

Keywords Language identification system (LID), deep convolutional recurrent neural network (CRNN), dataset, noisy environment

Introduction

In our day's many languages identification systems use online video sharing platforms (such as YouTube) or pre-created human-labelled datasets to train a language identification system. However, it is also possible to generate a human-labelled dataset from conferences, meetings, forums or meeting environment with friends. But there are some problems. The biggest one is the problem of a noisy environment, which can be a barrier for generating a useful dataset in some cases.

Data collection from the environment is done through a microphone and the created recording is saved with the extension ".wav". WAV is a format that realizes a simple specification used across multiple different file formats. WAV files are simple to work with.

As mentioned in the LID system article [1], a hybrid Convolutional Recurrent Neural Network operates on spectrogram images of the provided audio snippets and as we know a spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time[2].

As a result of the dataset generation process, spectrograms are created, which later become input parameters for the neural network. Therefore, the best way to eliminate noise is to process the spectrograms correctly.

Materials and Methods

This is a brief description of how the system works: the system takes the recording received from a microphone and has a ".wav" extension. The next step is to make segments with 10 seconds duration from the recording. After that system generates spectrograms form segments and checks for bad images. The final results are spectrogram images dataset containing description files for training, validation and testing parts of the dataset. Description files have ".csv" extension and contain the links of the spectrograms in the local memory and their corresponding indexes (label of data name). The steps described are shown in Figure 1.

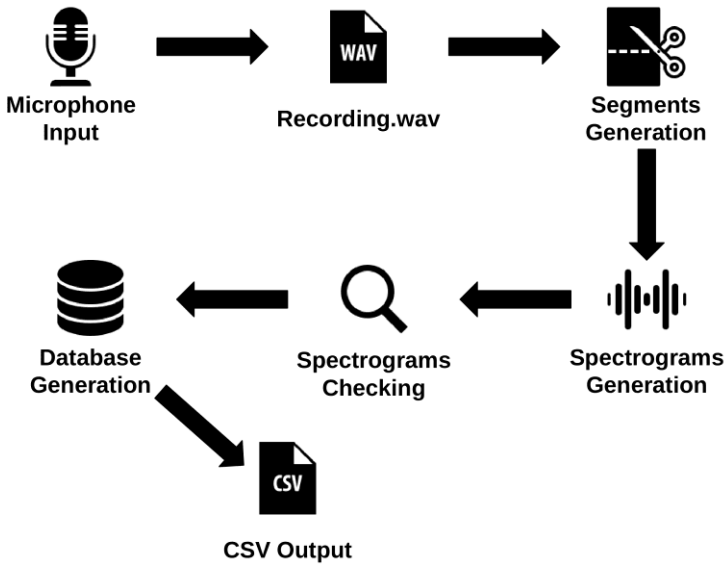

Figure 1: Structure of the system

Configurations for recording:

1. Sampling size is 16 bit integer format.
2. Channels number is 1.
3. Sampling rate is 44100 Hz.
4. The number of frames per buffer is 1024.

The process of splitting the input data after segmentation is done in the following stages:

1. Determine the minimum number of data segments (after this: smallest) in all data.
2. Test files (after this: test) are determined by the following formula: smallest * 0.05.
3. Training files (after this: train) are determined by the following formula: smallest * (0.8 - 0.05). 0.8 is the train set and validation set split.
4. Validation files are determined by the following formula: smallest - train - test.

The spectrogram generation process can be done in two ways:

1. Generation from a noisy environment.
2. Generation from a clear environment.

Configurations for generating spectrograms from noisy and/or clear environments:

1. Spectrogram image contains 50 pixel per second.
2. Spectrogram image is 129x500.
3. Channels number is 1.
4. Channel is mono.
5. Rate is 10k [3].

The creation of spectrograms is followed by the process of checking bad spectrograms. The steps are as follows:

1. The image becomes a vector of the sequence of pixel values.
2. The arithmetic mean of the pixel values is calculated.
3. The arithmetic mean of the pixel values calculated in the previous step is subtracted from each pixel value of the image.

The image is removed from the dataset if the number of non-zero elements in the vector obtained by subtraction is equal to zero.

Conclusion

Conclusion in this work, the designed system can be used as a tool for generating a human-labelled dataset for the CRNN neural network. It can filter noise and check bad spectrograms at the same time and generate a human-labelled dataset.

References

1. Christian Bartz, Tom Herold, Haojin Yang and Christoph Meinel, “Language Identification Using Deep Convolutional Recurrent Neural Networks”, 2017.
2. Muhammad Huzaifah, “Comparison of Time-Frequency Representations for Environmental Sound Classification using Convolutional Neural Networks”, 2017.
3. P.P. Vaidyanathan, “The origins of the sampling theorem”, IEEE Communications Magazine, Volume: 37, Issue: 4, 1999, pp. 106 - 108.