Kaggle Speech Recognition Challenge

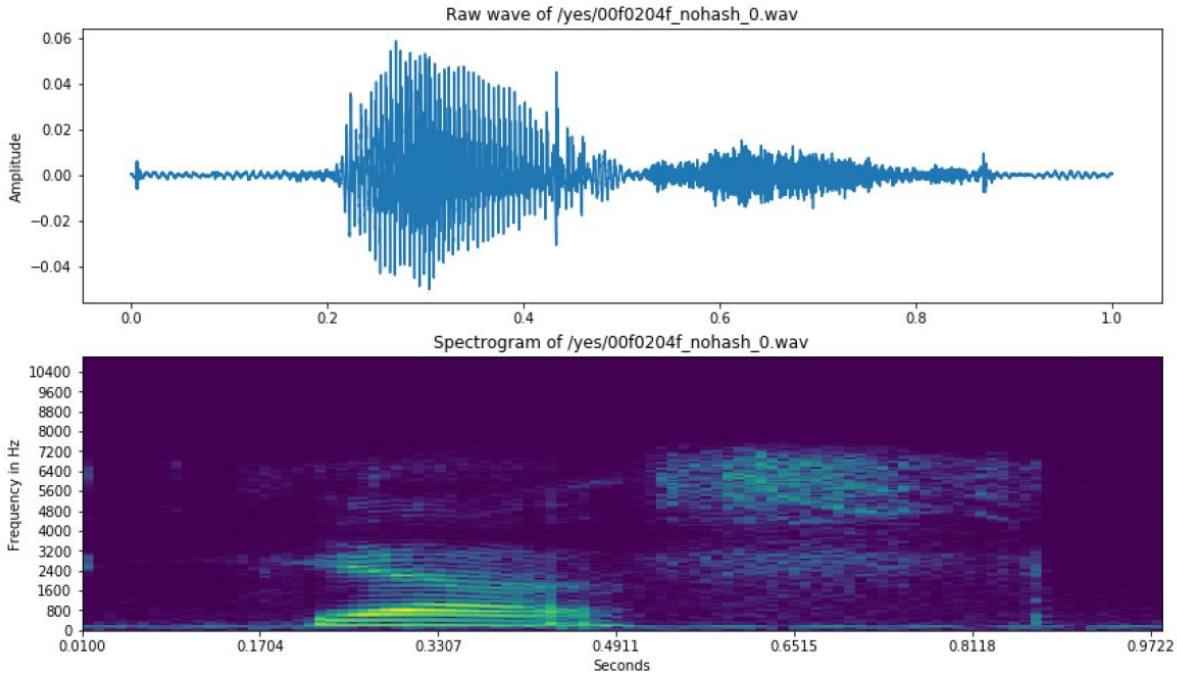
The task is to recognize a one-second audio clip, where the clip contains one of a small number of words, like “yes”, “no”, “stop”, “go”, “left”, and “right”.

In general, speech recognition is a difficult problem, but it’s much easier when the vocabulary is limited to a handful of words. We don’t need to use complicated language models to detect phonemes, and then string the phonemes into words, like Kaldi does for speech recognition.

The dataset consists of about 64000 audio files which have already been split into training / validation / testing sets. You are then asked to make predictions on about 150000 audio files for which the labels are unknown.

Earlier versions of this kernel could read .7z archive files directly.
But, recently Kaggle is not able to read .7z archives directly after their config updates for the Notebook Environment. So, we need to extract the 7z archive into a directory and work on it. Later, we need to delete it especially, if no. of files > 500. Read the instruction given below in three steps.
Step 1: Install Python packages in an internet-enabled notebook
Step 2: Unpack your .7z file
Step 3: Then after you are finished working with the images you can delete them so that your commit will succeed (max number of files in working directory for a commit = 500)

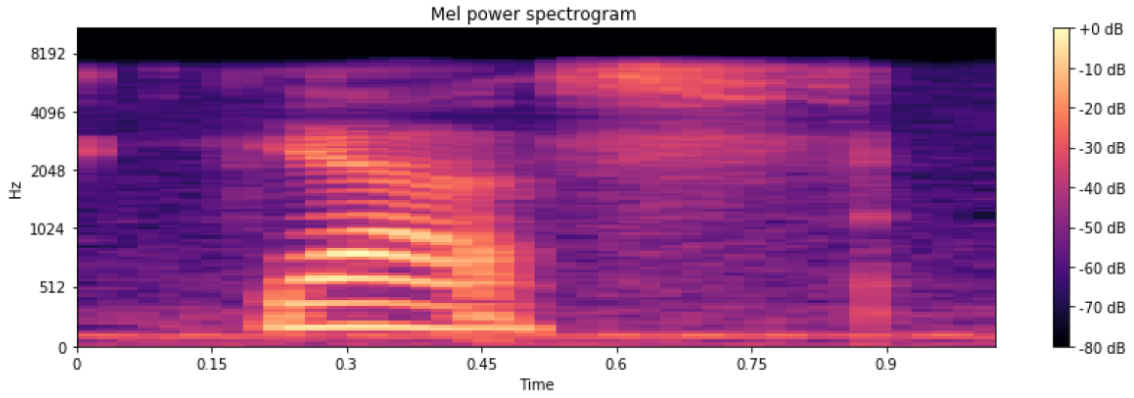
The first step that it does is convert the audio file into a spectrogram, which is an image representation of sound. This is easily done using LibRosa.



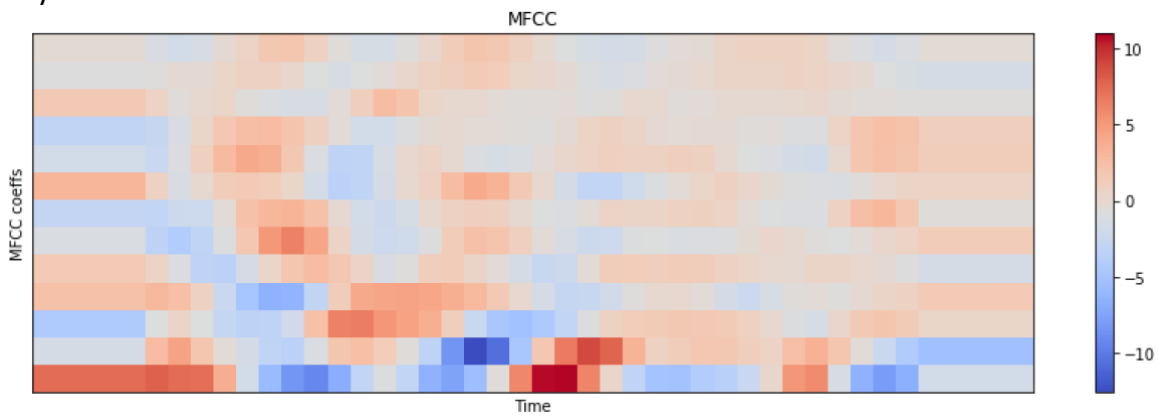
Now we've converted the problem to an image classification problem, which is well studied. To an untrained human observer, all the spectrograms may look the same, but neural networks can learn things that humans can't.

One usual problem with deep learning models is that they are usually "black-box" in the sense that it is very difficult to explain why the model reaches a certain decision. Attention is a powerful tool to make deep neural network models explainable: the picture below demonstrates that the transition from phoneme /a/ to phoneme /i/ is the most relevant part of the audio that the model used to decide (correctly) that the word is "right". Please refer to our paper for confusion matrix and more attention plots.

If you want to get to know some details about MFCC take a look at this great tutorial. *https://github.com/librosa/librosa/blob/master/examples/LibROSA%20demo.ipynb.*
You can see, that it is well prepared to imitate human hearing properties.
You can calculate Mel power spectrogram and MFCC using for example librosa python package



In classical, but still state-of-the-art systems, MFCC or similar features are taken as the input to the system instead of spectrograms.
However, in end-to-end (often neural-network based) systems, the most common input features are probably raw spectrograms, or mel power spectrograms.
For example, MFCC decorrelates features, but NNs deal with correlated features well. Also, if you'll understand mel filters, you may consider their usage sensible.
It is your decision which to choose!

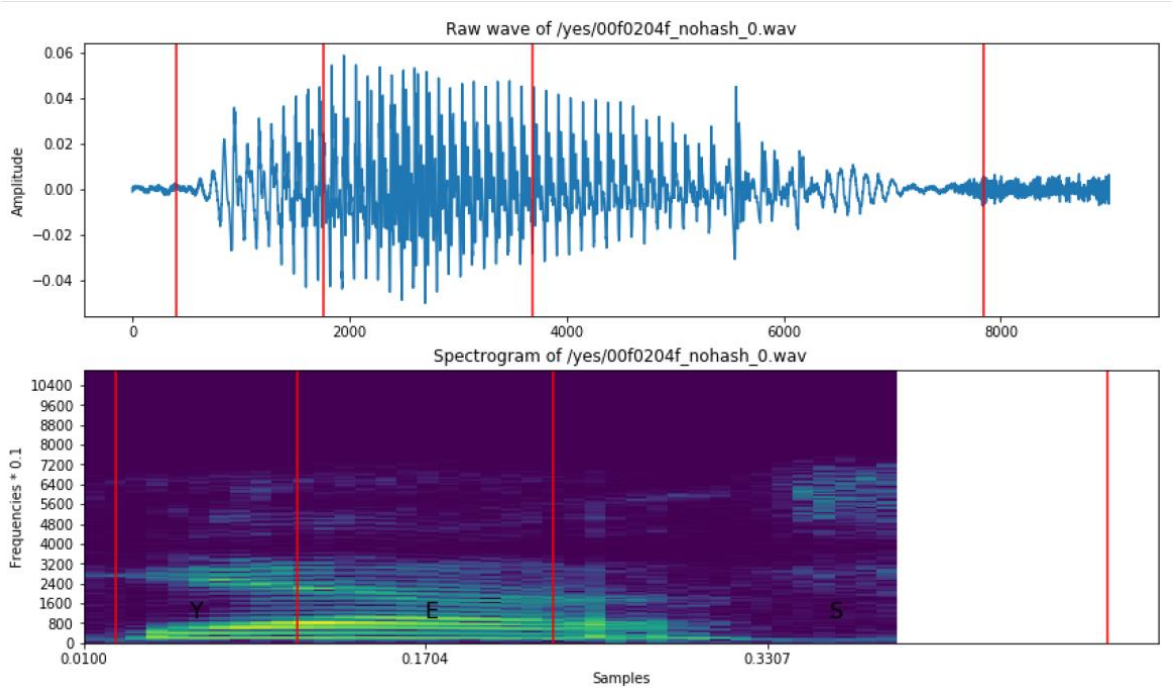


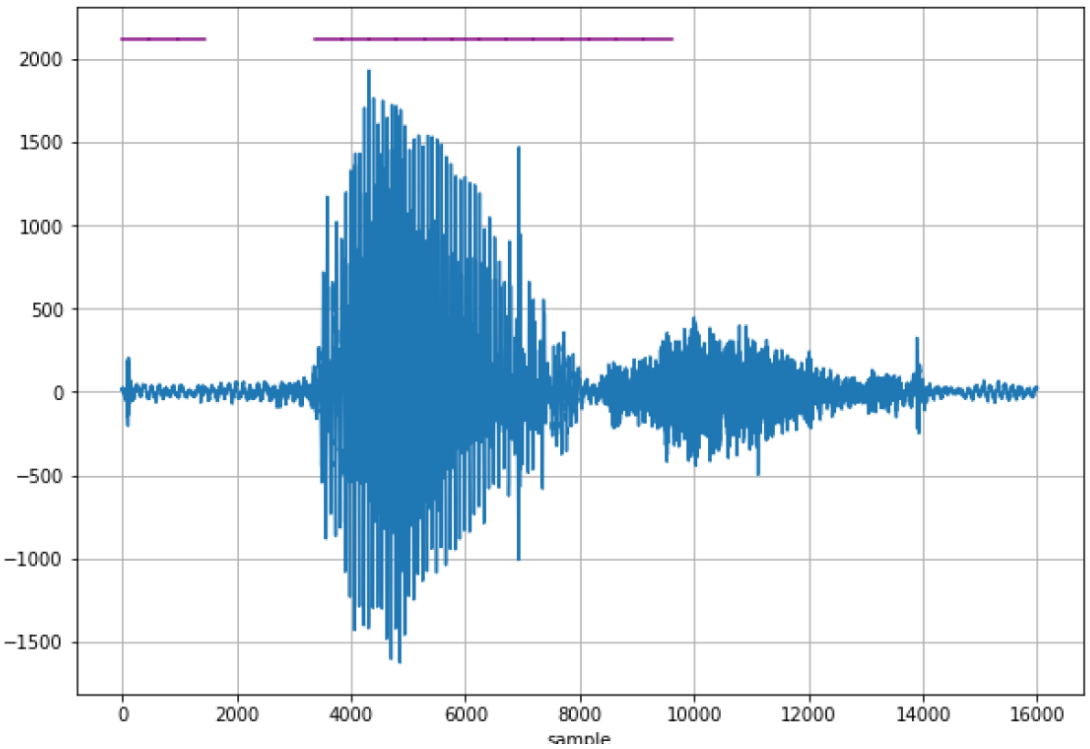
We can agree that the entire word can be heard. It is impossible to cut all the files manually and do this basing on the simple plot. But you can use for example webrtcvad package to have a good VAD.

Feature's extraction steps

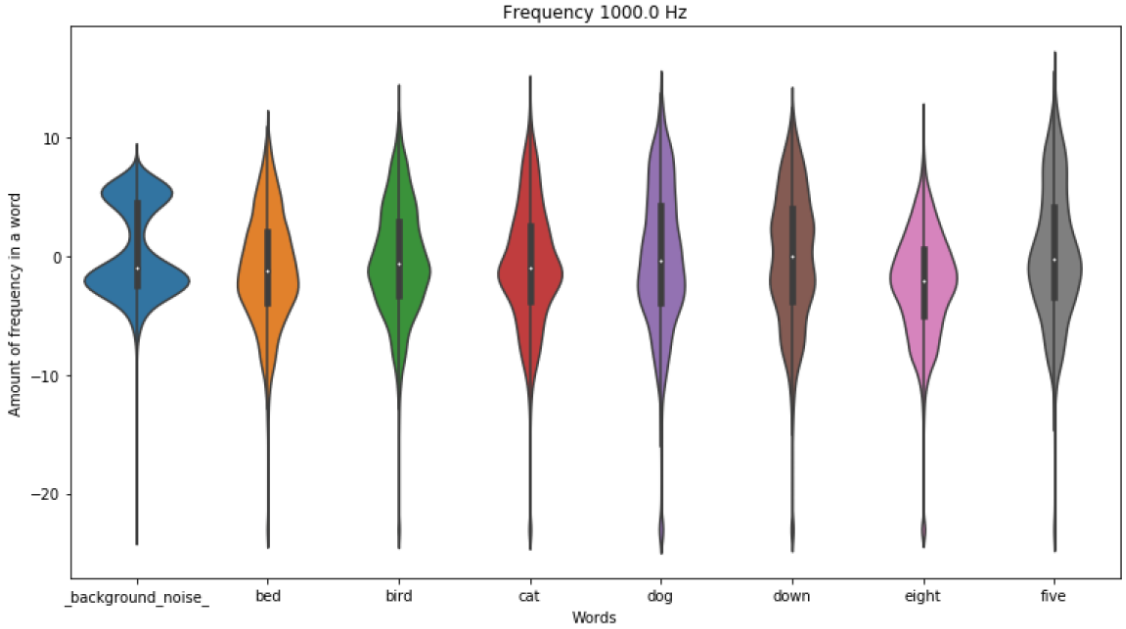
A generalized feature extraction algorithm for an audio data sample is like:

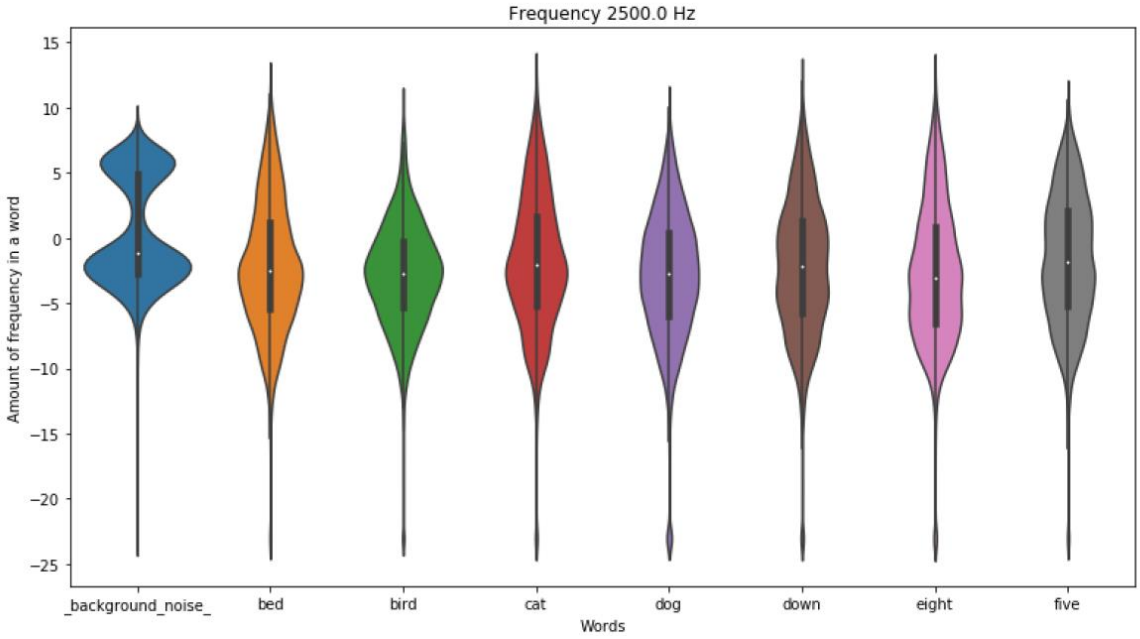
1. Resampling
2. VAD
3. Maybe padding with 0 to make signals be equal length
4. Log spectrogram (or MFCC, or PLP)
5. Feature's normalization with mean and std
6. Stacking of a given number of frames to get temporal information

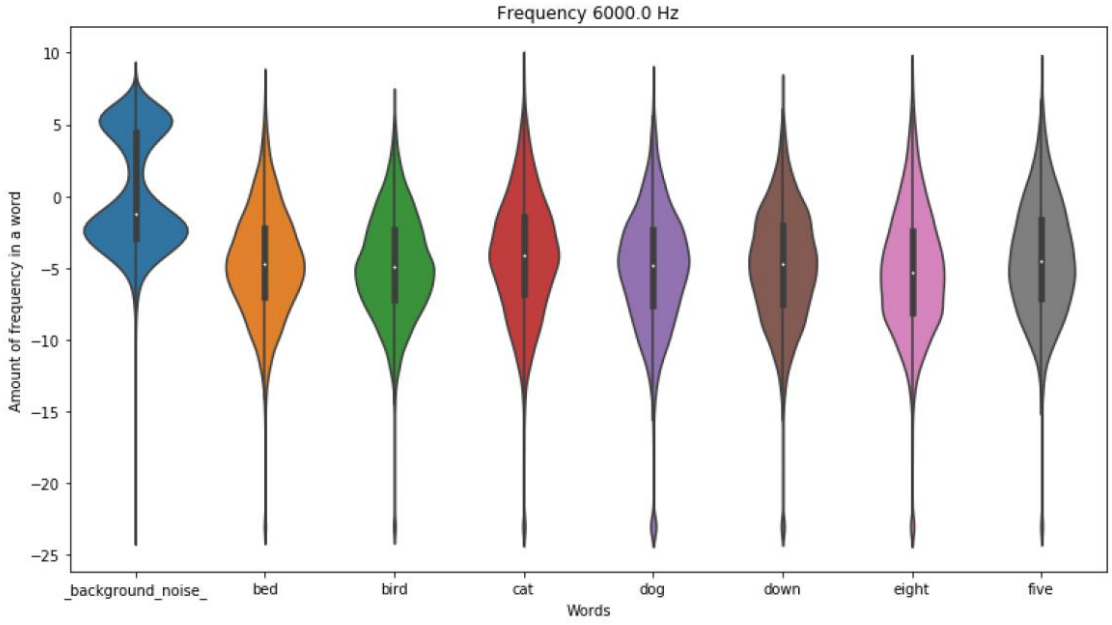




Visualizing some frequency components:







References:

- https://www.tensorflow.org/versions/master/tutorials/audio_recognition
- https://www.kaggle.com/davids1992/speech-visualization-and-investigation
- How to do Speech Recognition with Deep Learning
- A quick hack to align single-word audio recordings
- Speech Processing for Machine Learning: Filter banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What's In-Between
- Speech Recognition: You down with CTC?
- Convolutional Neural Networks for Small-footprint Keyword Spotting
- small-footprint keyword spotting using deep neural networks
- Deep Speech: Scaling up end-to-end speech recognition
- Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks
- Towards End-to-End Speech Recognition with Recurrent Neural Networks