ARTIFICIAL INTELLIGENCE
PROJECT SUMMARY REPORT

Sacco Domenico

REPORT INDEX

1 PROJECT SUMMARY

1.1 GOAL

The main goal of this project is to use a neural network to determine if a sound sample contains or not human voice and if so, trying to recognize the affinity with the voice of a particular speaker.

1.2 USED TOOLS

Datasets that were used are produced from .WAV files [1], using C++ functions to extract the Pulse Coded Modulation values. The sampling frequency is 44100Hz in all used samples and quantization is performed with 16 bits.

A C++ program was designed to build the dataset used by the neural network, its output file has the .train extension. In the first project version (non-real time sample analysis), also other .test files were produced.

The primary purpose of the .test files is to give an external validation, those .test files are not belonging to the training set and are coming from completely different WAV files.

Another C++ program is responsible for the training of the neural network and for the audio samples analysis.

Neural network is managed using the OpenCV module for C++. Also, Standard Vector Machine was implemented in some instances to test its suitability in replacement of the neural network.

In the latest stages of the project, also a library for MFCC features extraction [2], named "libmfcc.h" [3] was used.

2 NON-REAL TIME SPEAKER RECOGNITION

2.1 PROBLEM DESCRIPTION

Recognizing the speaker identity is not a simple task, many problems are occurring, both at theoretical and practical level:

- **Presence of noise**

Attempts to reduce noise were made using a lowpass filter of different orders, even managing to reduce the noise, filter application can bring significative degradation on voice samples, hence this solution, present in the first versions, was eventually discarded.

- **Moments of silence**

This problem is critical to achieve a training set of good quality. A thresholding mechanism on frequency domain has been applied, the value to be compared to the threshold has been calculated summing the average values of the Fourier specter segments (defined as frequency groups), not going beyond the threshold value implies discarding the sample.

- **Different intensities of the human voice samples**

This can be due for instance to the different speaker location during the recording

- **Sound distortion of the PC Microphone**

This last point implies the necessity to capture both the voice of the speaker to be recognized and of other speakers (to have the false training sample batch) through the PC microphone, since its distortion cannot be easily reproduced on voice samples coming from other sources (YouTube Videos for instance), creating and heterogeneous training set, both with microphone distorted and not distorted voice samples.

2.2 DATASET USED

A set of WAVE files was created through both registrations and downloading of YouTube videos, also environmental noises recordings were added to put the neural network in condition of distinguishing between human voice and those in the first place.

2.3 PROJECT IMPLEMENTATION

A program in C++ was written to create the required training and test set.

Applied procedure is the following:

- PCM (Pulse Coded Modulation) values are extracted from .WAV files
- Acquired PCM values are divided in windows (500ms each in the final version)
- Fast Fourier Transform is applied to each window
- Average values inside segments of the bandwidth are calculated
- Thresholding is applied: This step is critical to eliminate moments of silence during the speeches and other irrelevant samples.
- Samples with power above thresholding are stored into the .train file
- A binary value is applied to the sample to state its belonging or not to the particular speaker's voice the neural network will be trained to recognize. The binary value is determined from a tag applied to the file name. "£VOICE_TRUE" for the designed speaker's samples and "£VOICE_FALSE" for other samples.

At this point, a second C++ program using OpenCV was written

- Input and output values are acquired from the .train file
- Sample batch is split between training and test set
- Neural network is trained
- Neural network is finally tested

2.4 NEURAL NETWORK PROPERTIES

Used neural network is fully connected and is composed by three layers, the first of 50 neurons, the second of 5 neurons and the third of 1 neuron, which will define the belonging of the input sample to the designed speaker or not.

Training method used is the back propagation one, here follows other technical parameters:

- Learning Rate = 0.01
- Momentum = 0.8
- Term criteria: 1000000 iterations or 0.01 as maximum error

2.5 RESULTS DISCUSSION

Error rate of the neural network was 0.81% both in training and testing sets, aside from testifying a good choice of the training parameters, this also certifies the absence of overfitting.

In order to verify a certain degree of neural network validity, from three different independent registrations, other .test files were built, the building procedure of those files is basically identical to the one for the .train file, except that those files' samples are fed to the neural network only to be evaluated.

Neural network has been trained to answer within a range of [-1;1], with negative values for samples with distant properties from the designed speaker's voice and positive ones for the most similar ones.

Three. train files have been created:

- A file containing microphone registration of a stranger's voice
- A file containing the designed speaker voice
- An audio sample not acquired by the computer's microphone and containing a stranger's voice

It was decided to include a stranger's voice acquired with microphone to show that the recognition of the speaker identity is not related to the microphone distortion.

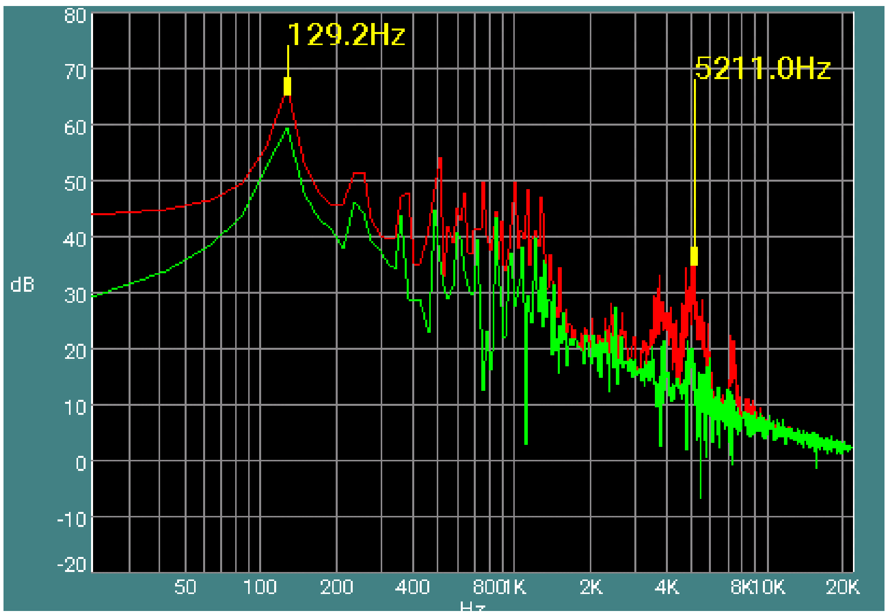
The neural network, despite the distortion, was able to tell the difference between the designed and not designed speaker.

Average intensity of the samples has been calculated, discarding useless samples applying the same thresholding mechanism used during the training.

The average answer of the neural network is the following one:

- Stranger's voice registered with microphone, external sample average answer: -0.529333
- Stranger's voice not registered with microphone, external sample average answer: -0.42845
- Designed speaker's voice registered with microphone, external sample average answer: 0.210817

Here follows the representation on Fourier domain, with logarithmic scale, of a human voice sample, is noticeable the dominance of the lower frequencies:



Human voice Fourier specter, "A" vowel, in red recording maximum values, in green current values

3 A SIMPLE REAL TIME SOUND RECOGNITION APPLICATION: RECOGNIZING A WHISTLE

With the project infrastructure already prepared to create and process the training and test files, it was decided to try bringing the problem solution one step further, trying to recognize in real time the whistle, a very simple sound feature.

3.1 PROBLEM DESCRIPTION

Real time recognition brings new and critical problems:

- Managing the sound Buffer
- Conceiving algorithms able to calculate the required result before another sample requires to be processed

Those two problematics were faced creating a new C++ program that, with a neural network, would interpret in real time if microphone input is a whistle or just voice/environmental sounds.

Neural network was previously trained and saved with another C++ program.

3.2 DATASET USED

A training batch was built, simply whistling at the microphone, thresholding mechanism was applied to eliminate useless samples as moments of silence.

To deal with the real-time buffer, the module WaveIn for microphone sound acquisition [4] was used.

Windows size in the final version is set to 125ms (eighth of a second).

3.3 PROBLEM SOLUTION

A more accurate sampling in Fourier domain has been applied, with 351 samples acquired. Low frequency values (0 – 5500Hz) were acquired from Fourier specter regularly at distance of 2 (one was taken, the following discarded and so on).

High frequencies did not contain useful informations to recognize a human whistle, therefore were used just to discriminate environmental noises.

The upper band (5500 Hz – 11000Hz) was taken with a sampling pattern deriving from a geometrical progression with 1.05 reason, to increase progressively the frequency of the values acquired in a faster way, covering this portion of the Fourier specter with less frequency values.

With the designed program, there is the chance to recognize a human whistle in real time, to show an example of application, a simple macro activating Cortana voice assistant [5], was created and it actives every time a whistle is recognized. Other macros can be created to execute any other command.

3.4 NEURAL NETWORK PROPERTIES

Due to the higher number of inputs, in this case 4 layers were used, the neural network structure is now 351-175-35-1.

Training method used is the back propagation one as the non-real time speaker recognition, same as the other technical parameters:

- Learning Rate = 0.01
- Momentum = 0.8
- Term criteria: 1000000 iterations or 0.01 as maximum error

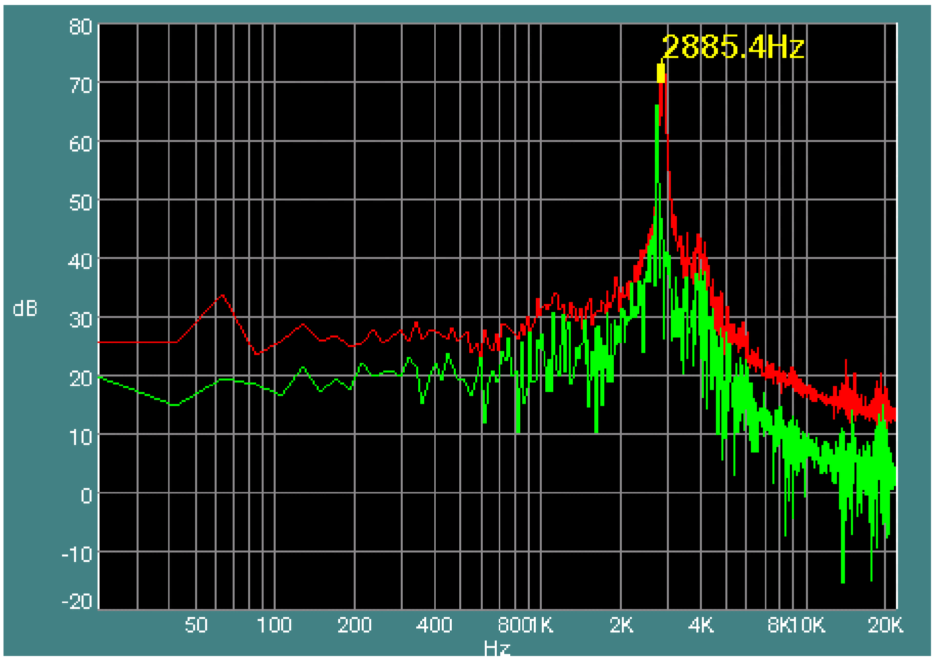
3.5 ADVANTAGES IN USING A NEURAL NETWORK

Human whistle is not very simple to model with a mathematical function, his behavior in the specter is quite similar to a Dirac distribution, although is more spreaded and varies of intensity through time:

Creating a solution of the whistle recognition problem without the help of a neural network can be very difficult, especially for all the parameters that must be included in an eventual model, including:

- Bandwidth
- Intensity (also in relation to the other specter components)
- Intensity variation thorough time
- Duration

With the usage of a neural network, all those problems were avoided, the problem was solved by extracting the features from acquired samples and training the network.



Human whistle, in red the recording maximum values, in green the current ones

4 REAL TIME SPEAKER RECOGNITION

4.2 PROPOSED SOLUTION

To solve this problem, the problem itself was divided in two subproblems:

- Distinguishing human voice from environmental noise
- Distinguishing designed speaker voice from human voice

Two kinds of labels were applied to each WAVE sample before being processed.

Those discriminating if it is human voice or noise:

- £HUMAN_TRUE
- £HUMAN_FALSE

Those discriminating whether it is or not the designed speaker's voice:

- £VOICE_TRUE
- £VOICE_FALSE

The first subproblem was solved acquiring the samples and generating the Fourier transform with the same criteria applied to the whistling recognition (higher number of frequency samples and different sampling politics for lower and upper parts of the specter), all audio files were processed. And a neural network was trained, with the same properties of the whistling recognizer.

The first choice on how to solve the first subproblem was using a Standard Vector Classifier to have as output a value of 0 in case of environmental noise and a value of 1 in case of human voice. The performance of the Standard Vector Classifier was compared with a neural network having the same structure of the whistler recognizer, the latter gave better performances, being able to discriminate environmental noises (like knocking on a table) from human voice better than the SVM does.

The second subproblem was solved processing only the audio files having human voice and extracting the MFCC features (in the number of 20), feeding them into a neural network. The neural network has three layers (20-3-1) and the same training characteristics of the other neural networks.

4.3 FINAL MODEL STRUCTURE

The final structure of the recognizer is a hierarchy of two neural networks, each solving its subproblem.

During the sample acquiring, if the first network detects only background noise, the second neural network is not even activated.

If the first network detects human voice, the second neural network is activated to determine its belonging to the designed speaker or not by evaluating the extracted MFCC features.

5 PROJECT CONCLUSIONS

Project version of non-real time speaker recognition brought good results both in internal and external texting (low error rate in both training and testing set, plus the correct recognition of the external samples).

The whistle recognizer brought very good practical results in a real time application, managing to distinguish from other impulsive background noises (knocking on the table, clicking with the mouse or typing on a keyboard for instance).

The last application, the real time voice recognizer, could be tested only with few non-designed speaker voice samples, therefore its effectiveness is limited. Although, the first of the two modules, having the goal of distinguishing from background noise and human voice, remains quite efficient in its performance, even managing to distinguish noises mainly belonging to the human voice frequency range.

6 BIBLIOGRAPHY AND SITOGRAPHY

[1] MCGill engineering, Peter Kabal, Audio File Format Specifications,

http://www-mmsp.ece.mcgill.ca/Documents/AudioFormats/WAVE/WAVE.html

[2] DOCBOX, Bob Sturm L. , Marcela Morvidone and Daudet, Laurent,

http://sciencedocbox.com/Geography/65628836-Musical-instrument-identification-using-multiscale-mel-frequency-cepstral-coefficients-sturm-bob-l-morvidone-marcela-daudet-laurent.html

[3] GitHub, jsawruk, LIBMFCC

https://github.com/jsawruk/libmfcc/blob/master/libmfcc.c

[4] Windows Dev Center, waveInOpen function,

https://msdn.microsoft.com/en-us/library/windows/desktop/dd743847(v=vs.85).aspx

[5] WINDOWSCENTRAL, Comprehensive list of Cortana voice commands on Windows 10

https://www.windowscentral.com/comprehensive-list-cortana-voice-commands-windows-10-0