CATEGORICAL CLASSIFICATION USING **C.N.N** FOR **F.S.D.D**

ZERGUINE SOHAIB[1]

18/01/2021

CONTENTS

ABSTRACT

As an exam project, I've chosen to make a deep learning model that can classify the spelling sounds of digits, the work is based on an open source dataset called **F**ree-**S**poken-**D**igits-**D**ataset, whose I've regularly contributed, it consits of 7 speakers (3500 sound recording, 350 for each digit). This open dataset is accessible on `https://github.com` via the following link: `https://github.com/Jakobovski/free-spoken-digit-dataset`. The Method of this work is to transform the (.wav) mono 8 KHz sounds into 128×128 16-Gray-scale pictures called systemically Spectrograms. Therefore making the **C**onvolutional-**N**eural-**N**etwork deep learning model on the Spectrograms.

[1] *Department of Physics, Aix–Marseille University, Marseille, France*

1 INTRODUCTION

Deep Learning is a modern way of coding that is getting involved into industries and people lives tremendously, Based on imitating neural properties in living systems, this type of coding that we can define as algorithms which allow fitting of learning Hyperparameter for huge sets of data, in order to make a classification that depends on the training of the Hyperparameter. When testing a sound, it should be transformed into spectrogram the next section of the article explains why, and then it will go throw the model where it will get weights from the hyperparameter set $\{\mathcal{P}_i\}$, hence the weights he got will be compared to the categorical weights for every category $\{\mathcal{P}_{i_\alpha}\}$, where $\alpha = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ and $\sum_{\alpha=0}^{9}(\{\mathcal{P}_{i_\alpha}\}) = \{\mathcal{P}_i\}$. While the picture is passed through the Model it will collect different parameters from the whole Hyperparameter, each belongs to different category, but the algorithm decision is built on the category that contributes more to its weight.
The dataset that I will work on is somehow premitif, as it doesn't contain abundantly data which is gathered from(6+1) contributers as I'm the last, with (3000+500) spelling sound for every digit, hereafter coding a model that can classify sounds with as much precision as possible, sound files aren't easy to deal with, due to their supperposition of frequencies what makes finding patterns in a raw sound file a non trivial job, so I tried to adopt a strategy of transforming a sound into a soundprint (each pronounciation has a morphism), encoded in a picture of 128*128 size and with 16 nuances of gray, this visualization of sounds is technically called Spectrogram.

2 CONCEPT

2.1 Explaining Sounds in physics

Sound is one of the most observed phenomenon in nature, it is a result of vibration of a particle or a bunch of particles with mass distribution $\rho_{(\vec{r},t)}$ within a medium $\mathcal{M}$, this system while vibrating create an adiabatic pressure wave P (the power amount of the vibrations is negligeable compared to that is needed to be resisted as heat in the medium at short distances). This wave satisfies the general wave equation, assuming that the direction of propagation is only on x_{axis} the equation is given as:

$$\frac{\partial^2 P}{\partial x^2} - \frac{1}{C_s^2}\frac{\partial^2 P}{\partial t^2} = -\mathcal{D} = 0$$

Such as $\mathcal{D}$ is a dissipation coefficient that equals zero in Acoustic limit (adiabatic approximation) and the speed of propagation of sound C_s is therefore defined as:

$$C_s = \sqrt{\frac{C_p \times P}{C_v \times |\rho|}} = \sqrt{\gamma \times \frac{P}{|\rho|}}$$

γ is the adiabatic index defined as $\frac{C_p}{C_v}$ where: C_p and C_v are specific heats of the medium of propagation at constant pressure and volume respectively. in simplest case where the system vibrations are rigourously periodic, the solution of this equation is:

$$P(x, t) = Ae^{-\imath kx}$$

Otherwise it will be resolved in general case as a Fourrier sum:

$$P(x, t) = \sum_{n=0}^{n} B_n(t)\sin(k_n x + \omega_n t) \tag{1}$$

The Fourrier Coefficients $B_n(t)$ are time dependent, they play the role of the different amplitudes assigned to the supperposed waves of different specific wavevector k_n and frequency ω_n. If we assume that all this sound waves are transverse, then $(k_n x = 0)$. That means that a real sound wave is composed with the superposition of (n) different simple sinus waves each with its specific amplitude and frequency.

- Meanwhile when one is wanting to record some sound, a microphone is needed, it is composed of high number of Bandwith-Frequency-Filters, each one detect its specific frequency with its associate amplitude(power), These BFF's record the sounds synchronously. After that, they will be digitized and encoded in sound files format (wav, mp3...).

- The rate that this operation occurs periodicly is called simply the sound Rate, it is defined in **KHz**. The larger the rate the more precise the sounds are.
- If the microphone has two detectors then the stereo or "dual channel" recording is possible. it is impportant if one's wanting to know the direction of sound. Elsewise the sound is said mono.
- The measure of sound amplitude is done with respect to a reference sound pressure, it is dimensionless, Defined in Decibel as:

$$D_s = 10\mathrm{Log}_{10}\left[\frac{P^2}{P^2_{ref}}\right]$$

The figure below 1 shows my drawing of how a sound wave could been fragmented into simple superposed synchronous sinus waves, as they could have been recorded as well.

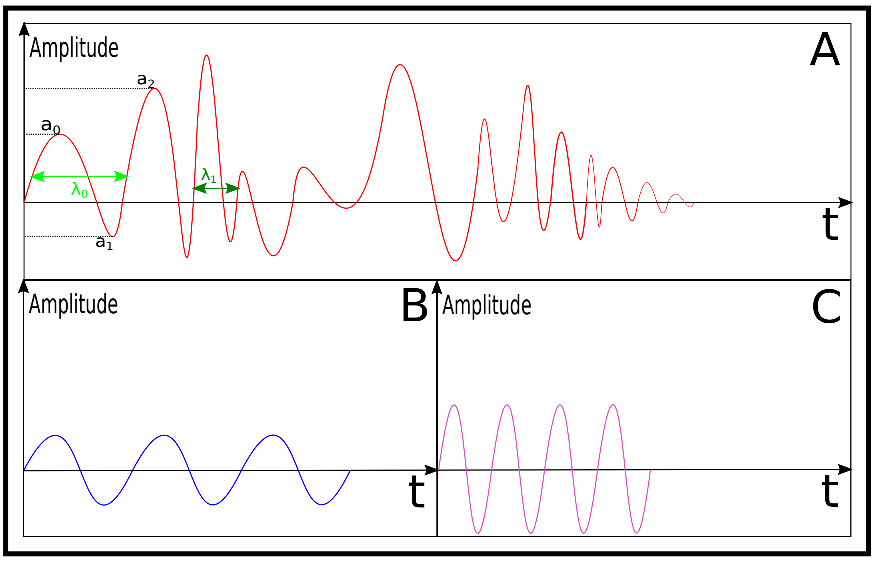
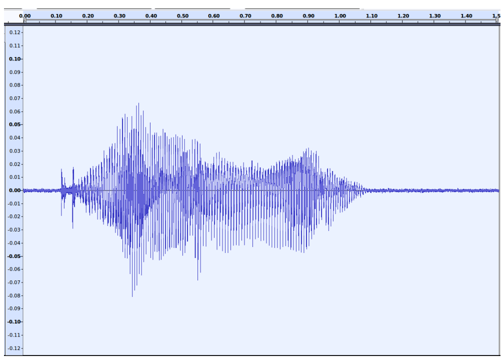


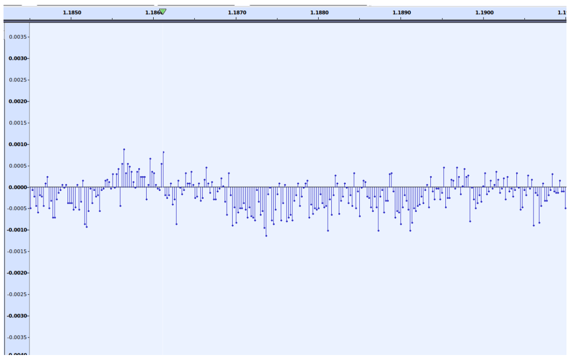
Figure 1: Demonstration of a simple Wave sound in nature **(A)** as a superposition of synchronous distributions of simple sinus waves 1 with different both amplitudes and frequencies **(B)** + **(C)**.

2.2 Spectrogram of a sound

When we record sounds we can easily obtain the waveform of our sounds, it measures the deviation of the amplitudes from the mean amplitude (noise of calm). These deviations are measured as segments perpendicular to the evolution axis, **t**(time axis). The sound wave is considered then as the envelop of these segments. The amount of time that separates segments equals $\frac{1}{\mathrm{Rate}=f_{max}}$ of the recording process. In my case all the audios are of a Rate with 8KHz, thus I can measure the vibrations 8000 times in one second. The following 2 I had taken a screenshot to my recording on Audacity®, which is a free software on Linux® Platform.



The waveform of my "Hello" recorded sound

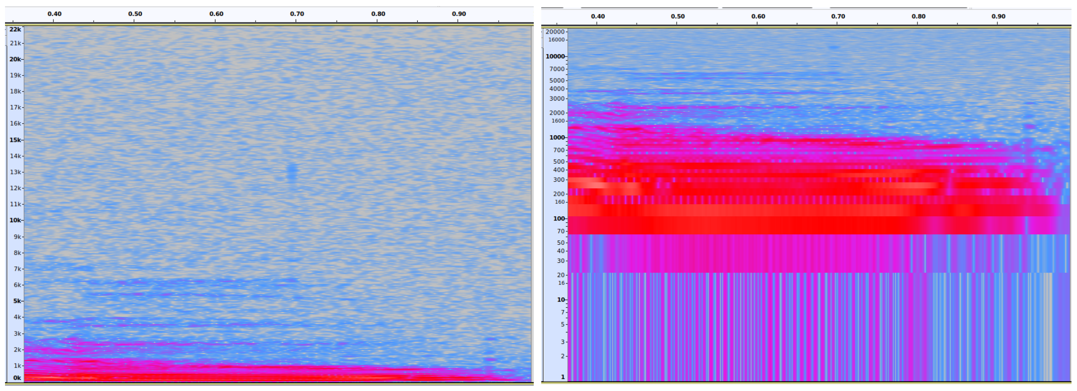


Sound encoded as segments distanced of the $\frac{1}{f_{max}}$

Figure 2: Visualization of Sound Representation

The working on an audio file format (that are designed essentially to be executed on the hardware) is head-breaking issue, for this cause, the sounds could been presented in a visual form, via another transform that maps a two dimensional waveform representation; with the issue of a non clear Fourrier components, into a very clear three dimensional representation, which encode at x_{axis} the time, at y_{axis} frequencies, and for z_{axis} amplitudes (that will be presented in a range of spectra) : $2\mathcal{D}_{rep} \longmapsto 3\mathcal{D}_{rep}$.

The results of such transformations are called Spectrograms, and depending on the type of sound that we need to analyze, we can represent them in linear frequencies scale or in a log frequencies scale. As presented in the following pictures 3:



RGB Colored Linear Spectrogram RGB Colored Logarithmic Spectrogram

Figure 3: Spectrograms of my spelling for "Hello" sound

3 FREE SOUND DIGIT DATASET

Is an ambitious open source project launched on October 14th 2016, by **Jacobovski** on Github platform, the project at its latest version 1.0.10, contains {3000 Spelling Sounds for 6 Speakers} + useful tools: Trimmer and Spectrogramer written both on Python. the Trimmer shorten and simplify the contribution to be a funny experience, and the spectrogrammer helps to dig and mine on the data. The Sounds that are included inside the dataset should satisfy these conditions:

- They should be clear and have been trimmed so that contain minimal silence at the edges of the file.
- They should been encoded in (wav) format, in mono channel, with a rate of 8KHz. So we can guarantee no overlap in the recorded waves (stereo case), and occupying small volume in memories.
- Every contribution should have been provided with Meta data that gives more about the contributers; (Country, Gender, Accent).

4 CONSTRUCTION OF MY DEEPLEARNING MODEL

In my project, I decided to transform the **F.S.S.D** into Spectrograms of 128×128 pixels with 16 Grey-scale amplitude representation. The justification of my choice are ordered by priority as the following:

1. The Spectrograms should have sufficient precision in every dimension while being sure to keep their storage size at a minimum. So they could been easily treated once loaded on the calculating machine.
2. Since the humain ear has a detecting rate of 100 Hz which is sufficient to distinguish this sounds, what pushes me to think of a windowing not so far from 100 scale for one second, I've chosen 128 scale for time x_{axis}.
3. The frequencies range will be scaled with same degrees for time $|(y_{axis})| = |(x_{axis})|$ in order to obtain a square picture. If we hear only the sounds between (20Hz $\sim$ 16KHz) than the frequencies scaling window is about 125 Hz. which is acceptable.

4. For the Amplitudes I decided to map them into 16 grey-scale, which is made a good differentiation between week and strong sounds, If my spelling was in the range of $d_{[D_s]} = [20, 70]$ decibels, then any rising or lowering of voice in human precision (3 decibels) is noticed with a different scale.

The following pictures 4 shows a Soundprint of the digit 7, in both my voice and george's voice:

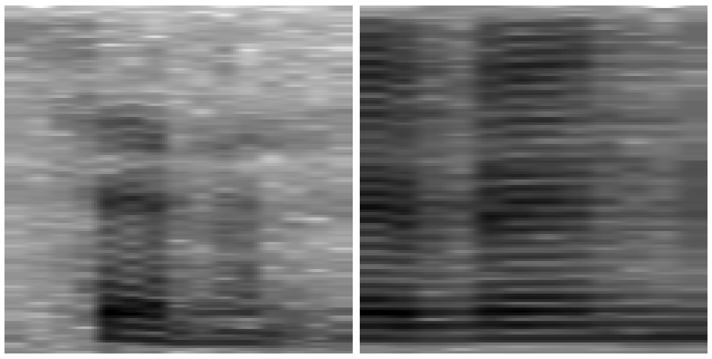


Figure 4: Soundprint of the digit "7", appears similar patterns, what gives a positive sign for the deeplearning work.

My Deep learning Classifying code is written in Python Language, Using Jupyter-Notebook environment, Holding on the Pionnering Deeplearning **A**pplication **P**rogramming **I**nterfaces:

1. **Tensorflow** as a provider of Mathematical functions and tools.
2. **Keras** as a back-end source of powerful and beautiful libraries for data processing. Which works by its own on-top of Tensorflow.
3. **Scipy** Which is a brilliant scientific package that is useful to deal with different data types, used to convert the (wav) sounds into (png) spectrograms.

4.1 Theory and the Raw Mapping

The Choices I've made are taken with care of not wasting any detail in the soundprints of the digit datasets$[(image_{height} = 128, image_{width} = 128), 16GrayScale]$, the result is therefore costful comparing to my ability of calculation, as I have an ordinary Ryzen 3500U Laptop, without a powerful GPU, that can perform the huge parrallel flows of data propagations and backpropagations. The Primary Model that I needed to adjust has the following properties, with a Hyperparmeter (H_{G128^2}) of (1156878) component. Which is really huge, it takes 6 minutes in average for every epoch. The fact that took much of time and makes the Model-tuning very long.

Layer	Output Shape	Parameters
Input $Conv2D_1$	((128,128),16)	160
$Conv2D_2$	((128,128),32)	12832
$Maxpooling2D_1$	((64,64),32)	0
$Dropout_1$	((64,64),32)	0
$Conv2D_3$	((64,64),64)	100416
$Maxpooling2D_2$	((21,21),64)	0
$Dropout_2$	((21,21),64)	0
$Conv2D_4$	((21,21),132)	1022340
$Maxpooling2D_3$	((4,4),132)	0
$Dropout_3$	((4,4),132)	0
Flatten	(2112)	0
Dense	(10)	21130

Table 1: The Raw Spectrograms Deeplearning Classifier Specs

Moreover, 3500 files of data aren't in fact sufficient to fine-tune (H_{G128^2}), in this situation the gain of every single detail of the digits soundprint allowed huge amounts of irrelevent noisy patterns, that need huge datasets of the order of 10^5.

4.2 Compactification of the $2\mathfrak{D} \longmapsto 3\mathfrak{D}$ Mapping

The need for compactification of the $2\mathfrak{D} \longmapsto 3\mathfrak{D}$ Mapping comes from the smallness of the datasets essentially, the nature of the sounds themselves (Just spelling digits) could be presented in compact patterns, throwing out patterns that are related to frequencies that doesn't contribute mostly to the spelling, the following figure 5 shows how the soundprint pattern is conserved in the compactified spectrograms:

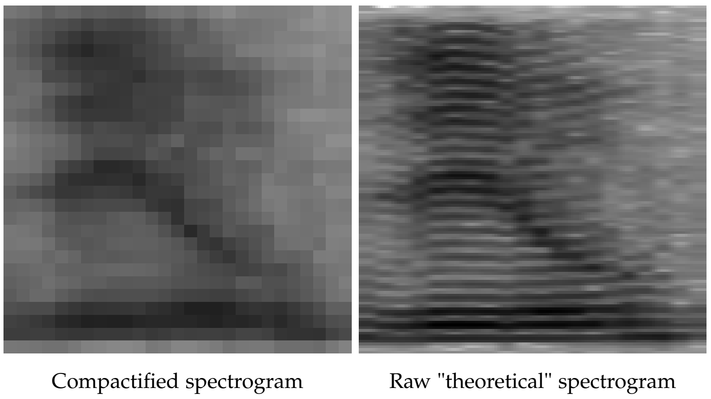


Figure 5: Same spectrogram of the digit "0", appears in the compactified $S_{george}(0)_{28\times 28}$ more noise free than $S_{george}(0)_{128\times 128}$ while preserving the morphism of the pattern once the datafile definition is reduced 21 times.

4.3 My Model

The classification will be performed then on the compactified spectrograms set either by making spectrograms with lower definition, or by resizing the spectrograms once injected in the datasets of the classifier. I have chosen the following **C.N.N** model which is summarised on the Table below:

Layer	Output Shape	Parameters
Input $Conv2D_1$	((28,28),16)	160
$Conv2D_2$	((28,28),32)	12832
$Maxpooling2D_1$	((14,14),32)	0
$Dropout_1$	((14,14),32)	0
$Conv2D_3$	((14,14),32)	9248
$Maxpooling2D_2$	((7,7),32)	0
$Dropout_2$	((7,7),32)	0
Flatten	(1568)	0
Dense	(10)	15690

Table 2: Compactified Model ($|H_{G28^2}| = 37930$)

Once the model is built, from the few first epochs, we can know if it is promising or disapointing, then the game turns into tuning game, such as choosing relevent filters, and good droprate at each dropout layer.

5 RESULTS

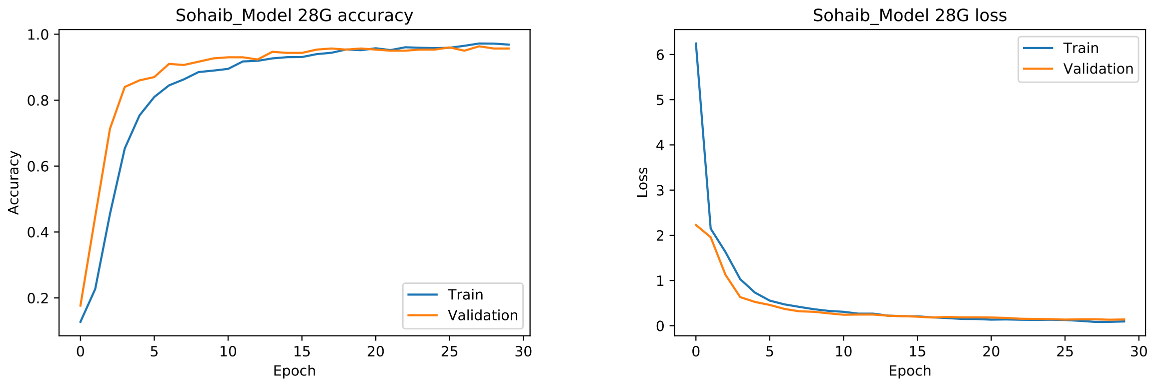
The quality of the Model could be judged according to three essential criteria:

1. **Accuracy :** This quantity is defined as the fitting performance of the model, the more it is higher, the more the model generelizes well.
2. **Entropy loss :** In Hyperparameter language, there is nothing called convergence of the values of the parameters, the fitting weights are subjects of change at every flow, until the general behaviour of the weights approaches to be invariant. there we can say that the statistical change in the weights is minor. Or equivalently the entropy loss of the Hyperparameter is minimal.

3. **Over Fitting Avoidance :** Over fitting begings when the fitting model, starts to interpolate the dataset, here the model will be suspecious for not generalizing efficiently.

5.1 Classifier fitting at F.S.D.D (v1.0.10)

It has reached an accuracy of 96.36% ; but suffering from overfitting, because of the similar accents of the 6 contributers (Half American, Half French).



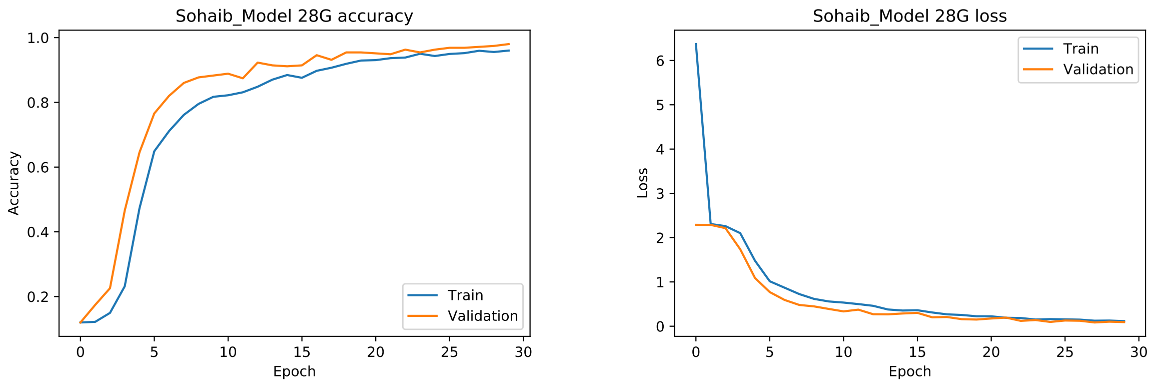
Compactified spectrogram

Raw "theoretical" spectrogram

Figure 6: Evolution of the accuracy and entropy loss for the F.S.D.D training based on v1.0.10

5.2 My Contribution included

Once My contribution is added, the accurecy raised to 98.00%, at the best try, without any overfitting, thanks to an introduction of new accent (Algerian).



Compactified spectrogram

Raw "theoretical" spectrogram

Figure 7: Evolution of the accuracy and entropy loss for the F.S.D.D training including my contribution data.

5.3 Testing Sounds from random people digit spelling

I have chosen to Predict my model on external sound speakers, so I went to Google Translator, where I can obtain its engine sounding with the American female accent Spelling, and after converting the audio sounds into (trimmed, mono, 8KHz rate, wav)format, My model predicted well (4/10) are fully correct (5/10) are coming correct for the second probability.
Similarily, the cause of lack of generalization, comes from the fact that all the contributers have **Male** gender, and are almost from the same range of age.
The dataset is performing Good, but it is not as mature as one can count on him to be accurate in general situations.
All the details Could been found in the Appendencies.

6 CONCLUSION

- The powerful model, is the one that satisfies the theoretical minimal requirements that I've estimated, but it is out of question for the moment, this project is going to be famous, if it can reach over 100 contributers (**50k**) from different regions from the world, both genders and from different ranges of ages.
- My approach of working on this topic, is to find always alternatives, as in real life each one will have its own problems, there is anything interesting in emitating successful models, whose one can find abundantly litterature.
- The accuracy achieved by the my CNN model, is relatively high, although it doesn't mean that it has an industrial quality performance, since the datasets are foetal.
- The open source community, gives free opportunities for people wanting to learn, as it made for the most of cases free Licence usage. Which facilitates the developping of our daily life. As for this example that I've chosen. I urge everyone who had learned from my experience on this workshop, to contribute to the dataset, on the link below: `https://github.com/Jakobovski/free-spoken-digit-dataset`.It takes from 40 minutes to several hours depending on your skills, but it will be a nice experience.

If you loved my work Please don't hesitate to follow me on Github®:
`https://github.com/ZSohaib`

BIBLIOGRAPHY

Richard Feynman, Lectures in Physics, Volume 1, 1969, Addison Publishing Company, Addison.
`https://github.com/Jakobovski/free-spoken-digit-dataset`
`https://keras.io/`
`https://soundoftext.com`

7 APPENDICES

The raw spectrograms model failure

The F.S.S.D v1.0.10 model

Main"trained after my contribution"