THE UNIVERSITY OF DANANG

DANANG UNIVERSITY OF SCIENCE AND TECHNOLOGY

FACULTY OF INFORMATION TECHNOLOGY

GRADUATION PROJECT THESIS

INFORMATION TECHNOLOFY FACULTY

PROJECT TITLE:

ISOLATING SINGING VOICE FROM STEREO MUSIC WITH CONVOLUTIONAL NEURAL NETWORKS

Instructor: **PGS. TS. NGUYỄN TẤN KHÔI**

Student: **DƯƠNG HUỲNH SƠN**

Student ID: **102150242**

Class**:** **15TCLC1**

Da Nang, 10/2019

INTRODUCTION

1. <u>*Project Overview:*</u>

1.1. Context:

The field of Music Information Retrieval (MIR) [1] concerns itself, among other things, with the analysis of music in its many facets, such as melody, timbre or rhythm. Singing is used to produces musically relevant sounds by the human voice, and it is employed in most cultures for entertainment or self-expression. The singing voice becomes immediately the main focus of attention when we listen to musical pieces with a voice part. Now a days, in multimedia technology various audio editor software's are available. Mixture of singing voice and music accompaniment known as a song. Music recording are either monaural (single channel) or stereo (two channel) basis. Speech is an acoustic signal produced from a speech production system. Sound is a representation of an audio signal. 20 Hz to 20 kHz are the audio frequency range. The human auditory system has a better capability in separating sounds from different sources.

Speech separation is a very challenging task in signal processing. An Audio signal classification system detecting the audio type of a signal (speech, background noise and musical genres). A singing voice separation system has its applications in areas such as automatic lyrics recognition and alignment [2], singer identification [3], musical information retrieval [4], karaoke [5], musical genre classification [6], melody extraction [7], audio signal classification [8], etc.. As for commercial applications, it is evident that the karaoke industry [9], estimated to be worth billions of dollars globally, would directly benefit from such technology.

1.2. Purpose:

In this project, I focus on the purpose of isolating the singing voice from the source contains the mixture of vocals and musical instruments sound, showed by the diagram below.

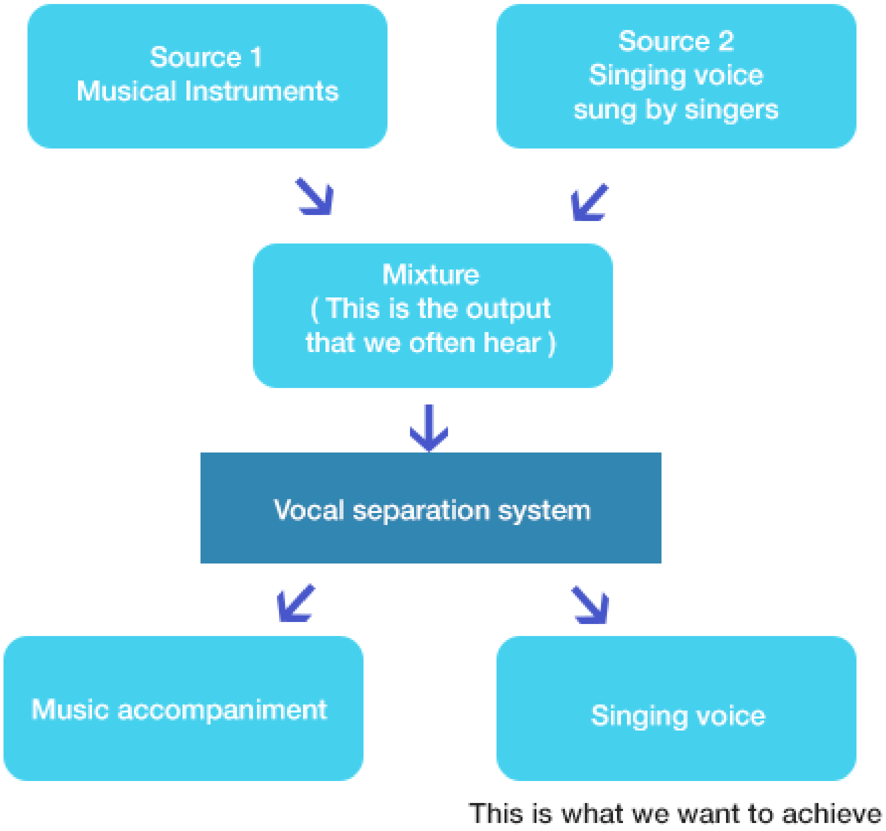


Figure 1 Diagram of Separation system.

As I mentioned on the Overview part, the system will isolate the singer's vocal sound from provided music mixture and return back the result of vocal-only file to users.

1.3. Project scope:

2. ***Structure of the thesis:***

***INTRODUCTION* -** This chapter gives information about the context and purpose of the project as well as giving the scope of the problems which will be focused on the thesis.

***Chapter 1: THEORIES AND TECHNOLOGIES* –** This chapter introduces about all knowledge theories and technologies used in this project.

***Chapter 2: ANALYSIS AND DESIGN* –** This chapter covers the main features, software requirement specifications and database design of the project.

Chapter 1: THEORIES AND TECHNOLOGY

First of all, we have to understand about sound and signal representation, and some methods that I am using in this work.

1.1. <u>Signal:</u>

1.1.1. Signal representation in modern computer:

The sound or signal data are saved in our devices as a waveform - a graph that shows a wave's change in displacement over time. A waveform's amplitude controls the wave's maximum displacement. Below is a sample of waveform visualization.

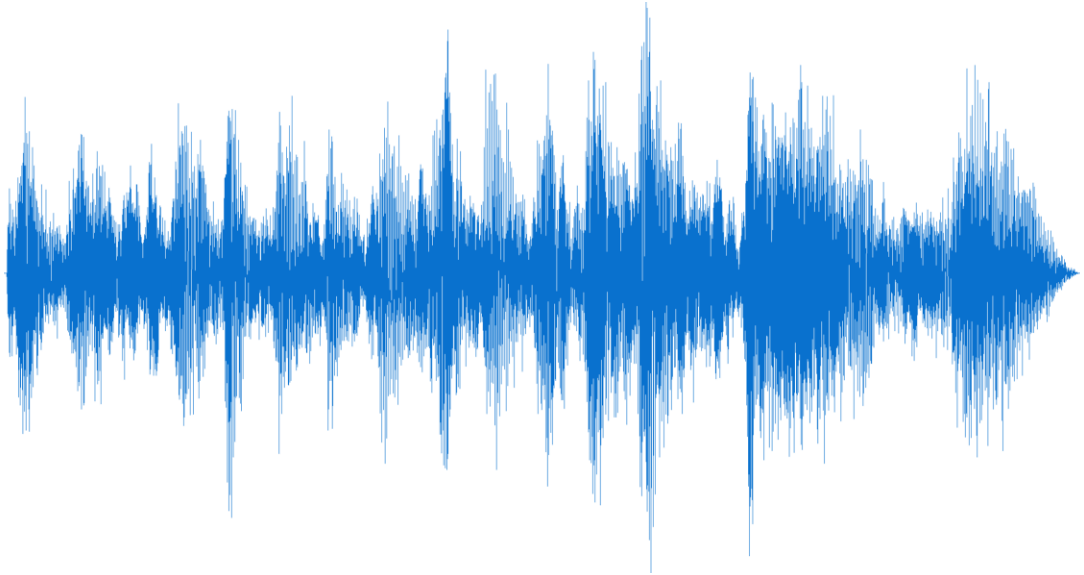


Figure 2: Waveform sample

Because the waveform or *time-domain signal* can just only provide us the access to the amplitude values of the signal over time. Which is not enough for analyzing work!

We have to transform these representations to another one to get access to more value and features from the source file.

1.1.2. *Short-time Fourier transform (STFT):*

The short-time Fourier transform (STFT) [10], is a Fourier-related transform [11] used to determine the sinusoidal frequency and phase content of local sections of a signal as it changes over time. It defines a particularly useful class of *time-frequency distributions*, which specify complex amplitude versus time and frequency for any signal.

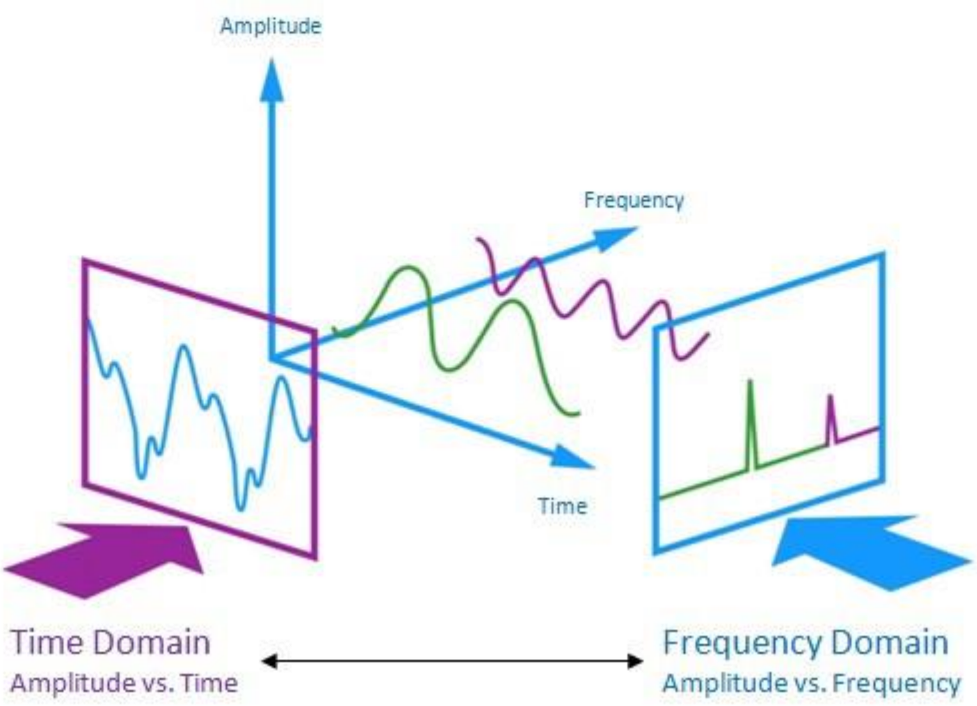


Figure 3: Fourier Transform (Picture from towardsdatascience.com)

The upper figure describes how our waveforms can be converted to Frequency Domain; therefore, we can get Amplitude vs. Frequency of Audio clips. If we have the window size for time domain small enough, we can get more information about Time. That is the idea of STFT. The below figure will show more about the representation of the sound after doing the STFT.

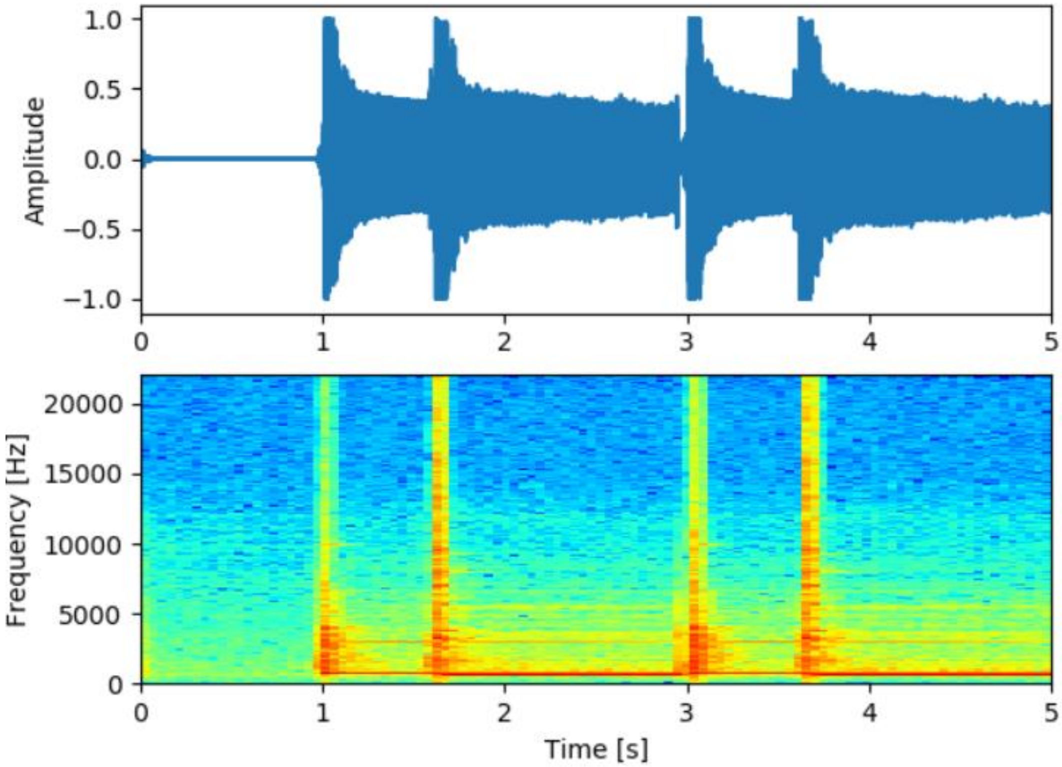


Figure 4: Waveform and STFT (picture from stackexchange.com)

The x-axis represents time, y-axis for Frequency and inside the picture, the warmer color, the higher amplitude of corresponding frequency is in the certain time. After doing STFT, we can have all information of Time, Frequency and Amplitude.

As we saw the output of STFT can be considered as a 1-channel image in which the warmer pixel is, the higher amplitude of Frequency in corresponding Time. This leads us to another approach to solve the problem, which shifts from signal processing into the problem of image to image translation [12].

1.2. Convolutional Neural Networks:

1.2.1. Deep Learning:

Deep learning [13] (also known as deep structured learning or hierarchical learning) is part of a broader family of machine learning methods based on artificial neural networks [14]. Learning can be supervised, semi-supervised or unsupervised.

Neural networks are a set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns. They interpret sensory data through a kind of machine perception, labeling or clustering raw input. The patterns they recognize are numerical, contained in vectors, into which all real-world data, be it images, sound, text or time series, must be translated.

Deep learning is the name we use for “stacked neural networks”; that is, networks composed of several layers. The layers are made of nodes. A node is just a place where computation happens, loosely patterned on a neuron in the human brain, which fires when it encounters sufficient stimuli. A node combines input from the data with a set of coefficients, or weights, that either amplify or dampen that input, thereby assigning significance to inputs with regard to the task the algorithm is trying to learn; e.g. which input is most helpful is classifying data without error? These input-weight products are summed and then the sum is passed through a node’s so-called activation function, to determine whether and to what extent that signal should progress further through the network to affect the ultimate outcome, say, an act of classification. If the signals pass through, the neuron has been “activated.”

Here’s a diagram of what one node might look like.

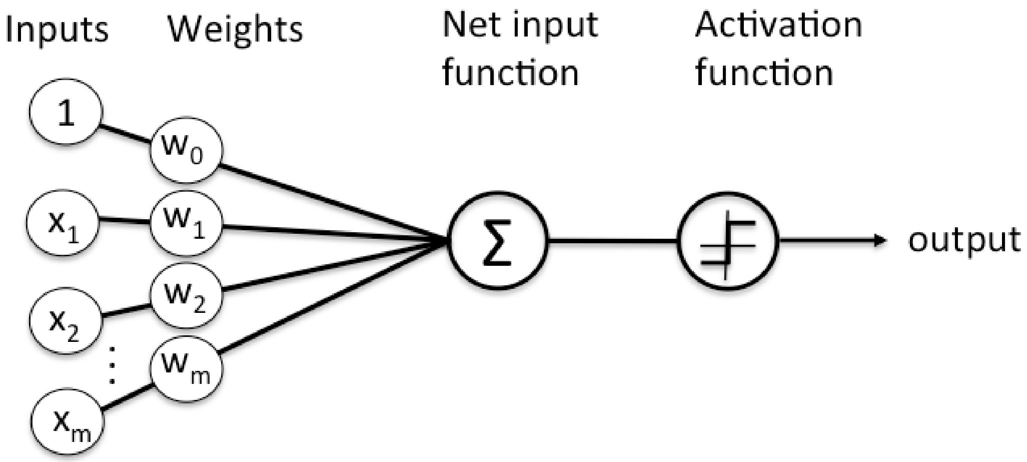


Figure 5: Perceptron node

A node layer is a row of those neuron-like switches that turn on or off as the input is fed through the net. Each layer's output is simultaneously the subsequent layer's input, starting from an initial input layer receiving your data.

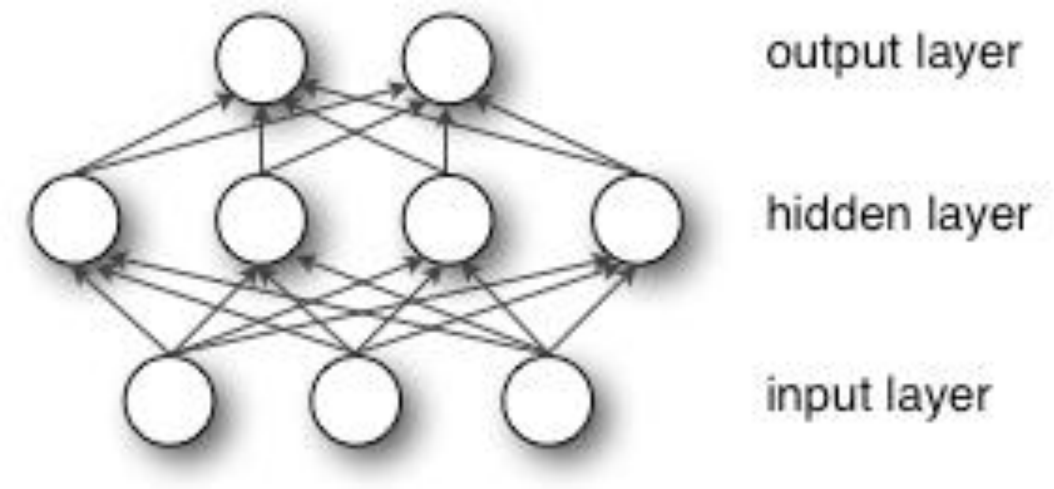


Figure 6: Multi Layers Perceptron

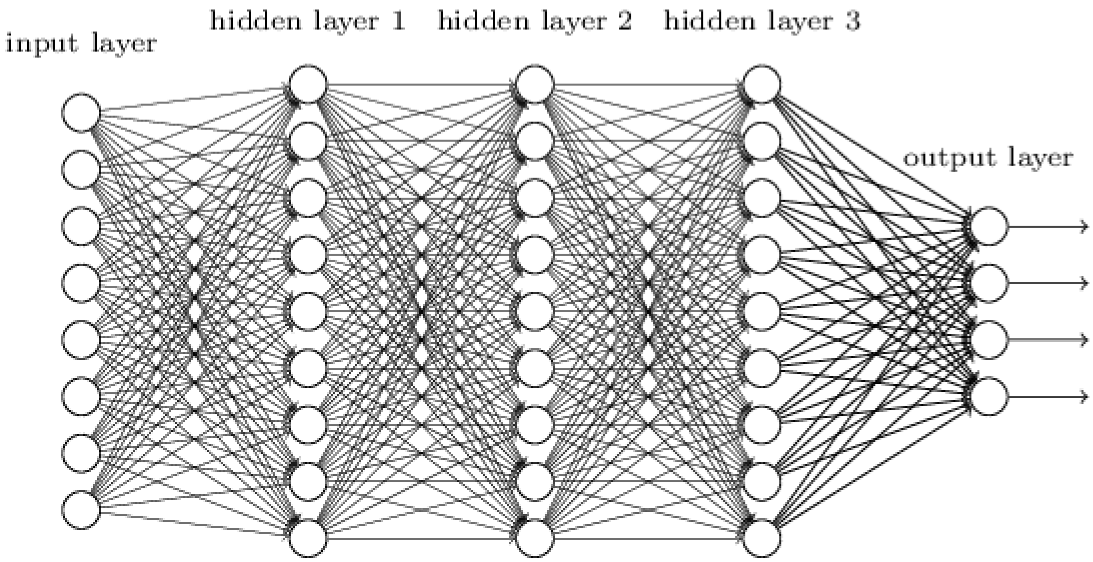


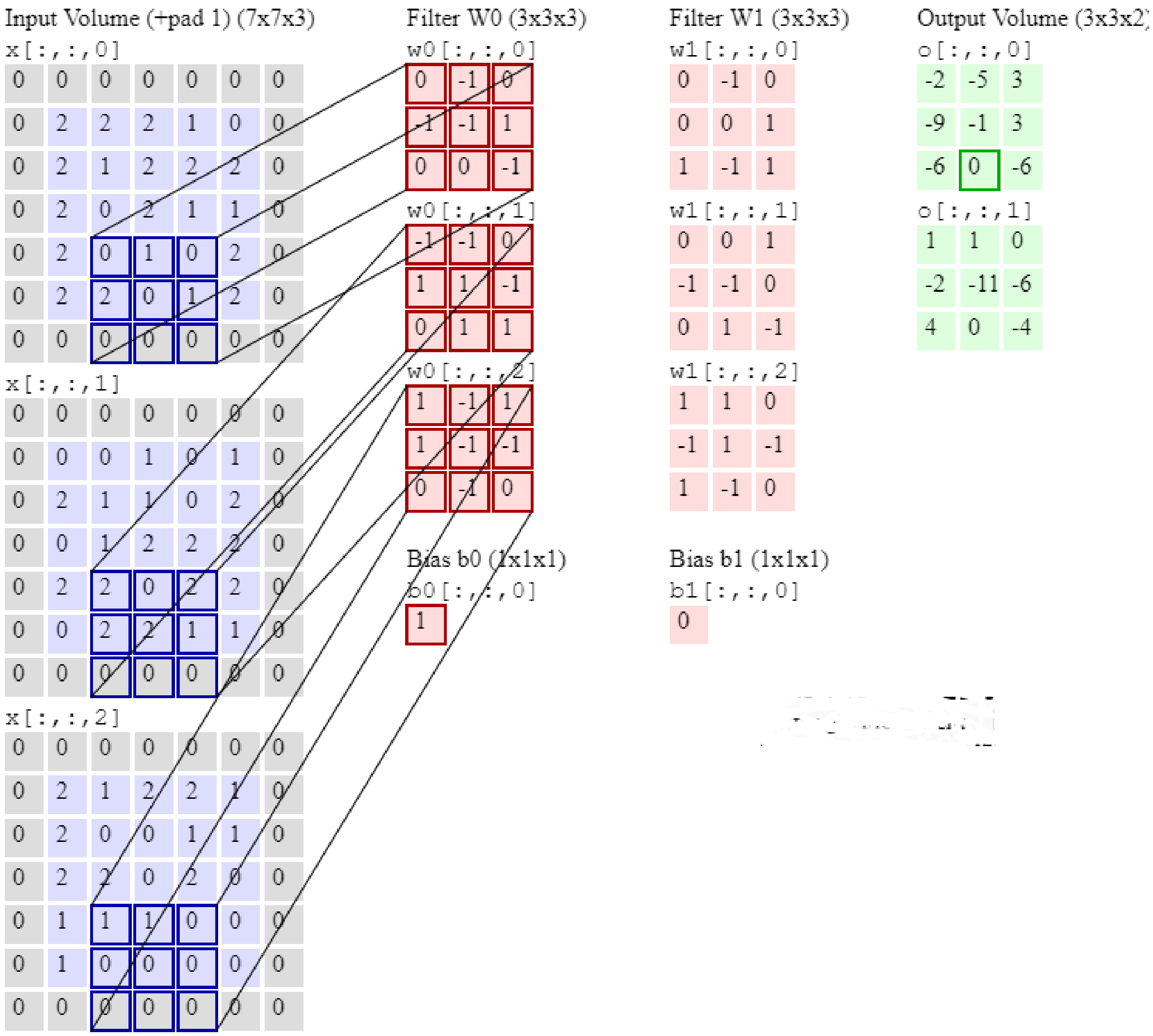
Figure 7: Deep Neural Networks(picture from neuralnetworksanddeeplearning.com)

1.2.2. Deep Convolutional Neural Networks:

Convolutional neural networks [15] are neural networks used primarily to classify images (i.e. name what they see), cluster images by similarity (photo search), and perform object recognition within scenes. For example, convolutional neural networks (ConvNets or CNNs) are used to identify faces, individuals, street signs, tumors and many other aspects of visual data.

Convolutional networks perceive images as volumes; i.e. three-dimensional objects, rather than flat canvases to be measured only by width and height. That’s because digital color images have a red-blue-green (RGB) encoding, mixing those three colors to produce the color spectrum humans perceive. A convolutional network ingests such images as three separate strata of color stacked one on top of the other.

So, a convolutional network receives a normal color image as a rectangular box whose width and height are measured by the number of pixels along those dimensions, and whose depth is three layers deep, one for each letter in RGB. Those depth layers are referred to as channels.



Input Volume (+pad 1) (7x7x3)

x[:,:,0]

0	0	0	0	0	0	0
0	2	2	2	1	0	0
0	2	1	2	2	2	0
0	2	0	2	1	1	0
0	2	0	1	0	2	0
0	2	2	0	1	2	0
0	0	0	0	0	0	0

x[:,:,1]

0	0	0	0	0	0	0
0	0	0	1	0	1	0
0	2	1	1	0	2	0
0	0	1	2	2	2	0
0	2	2	0	2	2	0
0	0	2	2	1	1	0
0	0	0	0	0	0	0

x[:,:,2]

0	0	0	0	0	0	0
0	2	1	2	2	1	0
0	2	0	0	1	1	0
0	2	2	0	2	0	0
0	1	1	1	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0

Filter W0 (3x3x3)

w0[:,:,0]

0	-1	0
-1	-1	1
0	0	-1

w0[:,:,1]

-1	-1	0
1	1	-1
0	1	1

w0[:,:,2]

1	-1	1
1	-1	-1
0	-1	0

Bias b0 (1x1x1)

b0[:,:,0]

1

Filter W1 (3x3x3)

w1[:,:,0]

0	-1	0
0	0	1
1	-1	1

w1[:,:,1]

0	0	1
-1	-1	0
0	1	-1

w1[:,:,2]

1	1	0
-1	1	-1
1	-1	0

Bias b1 (1x1x1)

b1[:,:,0]

0

Output Volume (3x3x2

o[:,:,0]

-2	-5	3
-9	-1	3
-6	0	-6

o[:,:,1]

1	1	0
-2	-11	-6
4	0	-4

Figure 8:Convolution in CNNs

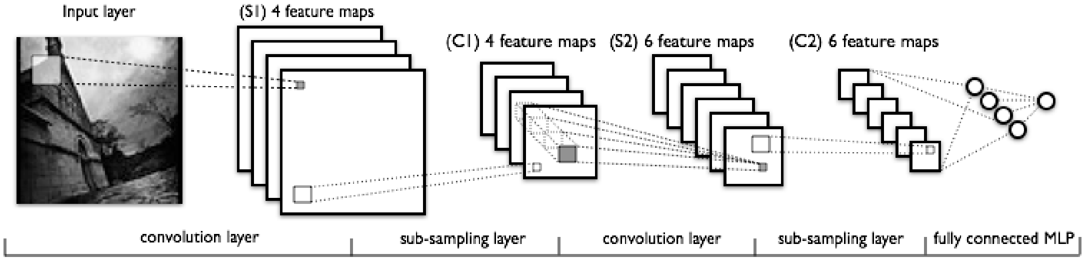


Figure 9: Convolutional Neural Networks

Here is a visualization of a convolutional Neural Networks.

1.3. Building and Training CNNs with Fastai library – Pytorch:

Fastai [16] is an open source library for training and testing neural networks with the simplification of fast and accurate training process using modern best practice. They build it on top of Pytorch [17], which built and supported by Facebook AI [18]. In addition, Fastai has many modern and great functions for us to implement our algorithms.

One of the greatest thing comes from Fastai is the one cycle learning [19], which shows that use of cyclic functions as a learning rate policy provides substantial improvements in performance for a range of architectures. In addition, the cyclic nature of these methods provides guidance as to times to drop the learning rate values (after 3 - 5 cycles) and when to stop the training. All of these factors reduce the guesswork in setting the learning rates and make these methods practical tools for everyone who trains neural networks.

1.4. Google Colab:

Training a deep learning networks is a very expensive process; therefore, we need a fast-enough GPU [20] to do so. But GPU is not quite cheap for a student like me to buy yet. Fortunately, Google has provided us a platform which gives us free Tesla K80 [21] GPU for training our Deep Neural Networks for totally free.

Google has done the coolest thing ever by providing a free cloud service based on Jupyter Notebooks that supports free GPU. Not only is this a great tool for improving your coding skills, but it also allows absolutely anyone to develop deep learning applications using popular libraries such as PyTorch, TensorFlow [22], Keras [23], and OpenCV [24]. You can create notebooks [25] in Colab, upload notebooks, store notebooks, share notebooks, mount your Google Drive and use whatever you’ve got stored in there, import most of your favorite directories, upload your personal Jupyter Notebooks, upload notebooks directly from GitHub [26], upload Kaggle [27] files, download your notebooks, and do just about everything else that you might want to be able to do.

1.5. Other technologies:

There are some minor technologies that I will not go into much details, but I will referent those when mentioning.

1.6. Conclusion:

By studying and learning about the above technologies, we successfully applied the concepts and their mechanism operating in this project to create genealogical tree system in mobile application.

Some of these technologies are not new, but they are widely using and a trend for software development industry. Therefore, understanding the concept is very important, help to apply properly technology for each project, in order to improve the efficiency and usability.

Chapter 2: ANALYSIS AND DESIGN

This chapter will go into detail the requirements, describing nonfunctional requirements, design constraints and other factors necessary to provide a complete and comprehensive description of the requirements for the application. This consists of a package containing Datasets building, Data pre-processing, CNNs model building, and model testing. Shows an overview of what functions the application can satisfy. In addition, it defines the architecture, modules, and data for a system to satisfy specified requirements.

System Design process is to provide sufficiently detailed data and information about the system and it is a system element to enable the implementation consistent with architectural entities as defined in models and views of the system architecture. It shows the components of the application, the structure of datasets, the deep neural networks model that make up the system.

*2.1. **Main features:***

The main feature this system has is convert from the data of raw audio file (.mp3, .wav) into the vocal-only audio file.

*2.2. **Datasets:***

Datasets is the folder of data which supports the training and testing process in Machine Learning field. In this work, datasets contain the sets of mixture audios and the corresponding vocal-only audio files.

There are a lot of datasets providers in the internet, in this work, I am using MUSDB18 [28] provided by Sigsep.

The **musdb18** is a dataset of 150 full lengths music tracks (~10h duration) of different genres along with their isolated *drums*, *bass*, *vocals* and *others* stems.

musdb18 contains two folders, a folder with a training set: "*train*", composed of 100 songs, and a folder with a test set: "*test*", composed of 50 songs. Supervised approaches should be trained on the training set and tested on both sets.

All signals are stereophonic and encoded at 44.1kHz.

The data from **musdb18** is composed of several different sources:

- 100 tracks are taken from the *DSD100* dataset, which is itself derived from The 'Mixing Secrets' Free Multitrack Download Library.

- 46 tracks are taken from the *MedleyDB* licensed under Creative Commons (BY-NC-SA 4.0).

- 2 tracks were kindly provided by Native Instruments originally part of their stems pack.

- 2 tracks are from the Canadian rock band *The Easton Ellises* as part of the heise stems remix competition, licensed under Creative Commons (BY-NC-SA 3.0).

2.3. <u>Data processing:</u>

Once having datasets downloaded, I use Librosa and Torch.stft to convert from waveform data to STFT data.

Chapter 3: References

[1] W. B. d. Haas and F. . Wiering, "Hooked on Music Information Retrieval," *Empirical Musicology Review,* vol. 5, no. 4, pp. 176-185, 2010.

[2] A. . Mesaros and T. . Virtanen, "Automatic recognition of lyrics in singing," *Eurasip Journal on Audio, Speech, and Music Processing,* vol. 2010, no. 1, p. 546047, 2010.

[3] J. . Shen, B. . Cui, J. . Shepherd and K.-L. . Tan, "Towards efficient automated singer identification in large music databases," , 2006. [Online]. Available: http://net.pku.edu.cn/~cuibin/papers/2006-sigir.pdf. [Accessed 30 10 2019].

[4] R. P. Smiraglia, "Musical Works and Information Retrieval," *Notes,* vol. 58, no. 4, pp. 747-764, 2002.

[5] P. . Arora, "Karaoke for social and cultural change," *Journal of Information, Communication and Ethics in Society,* vol. 4, no. 3, pp. 121-130, 2006.

[6] . . Bagci and . . Erzin, "Boosting Classifiers for Music Genre Classification," *Lecture Notes in Computer Science,* vol. , no. , pp. 575-584, 2006.

[7] R. P. Paiva, "An approach for melody extraction from polyphonic audio: Using perceptual principles and melodic smoothness," *Journal of the Acoustical Society of America*, vol. 122, no. 5, pp. 2962-2962, 2007.

[8] R. S. S. Kumari, D. . Sugumar and V. . Sadasivam, "Audio Signal Classification Based on Optimal Wavelet and Support Vector Machine," , 2007. [Online]. Available: http://yadda.icm.edu.pl/yadda/element/bwmeta1.element.ieee-000004426756. [Accessed 30 10 2019].

[9] B. . Mak and W. W. Chan, "In pursuit of operational improvement in the karaoke box business," *World leisure journal,* vol. 48, no. 2, pp. 48-53, 2006.

[10] N. A. Khan, M. N. Jafri and S. A. Qazi, "Improved resolution short time Fourier transform," , 2011. [Online]. Available: https://researchgate.net/profile/nabeel_khan3/publication/235677974_improved_resolution_short_time_fourier_transform/links/555068a008ae956a5d24bf47.pdf?disablecoverpage=true. [Accessed 30 10 2019].

[11] C. . Fan and S. . Wang, "A fast Fourier transform algorithm using Hadamard transform," , 1986. [Online]. Available: https://ieeexplore.ieee.org/iel6/8362/26344/01169074.pdf. [Accessed 30 10 2019].

[12] J. . Lin, Y. . Xia, T. . Qin, Z. . Chen and T.-Y. . Liu, "Conditional Image-to-Image Translation," , 2018. [Online]. Available: http://cvpr2018.thecvf.com/program/main_conference. [Accessed 30 10 2019].

[13] J. . Schmidhuber, "Deep Learning," *Scholarpedia,* vol. 10, no. 11, p. 1527–54, .

[14] A. J. Mansfield, "An introduction to neural networks," , 1990. [Online]. Available: http://www2.econ.iastate.edu/tesfatsi/neuralnetworks.cheungcannonnotes.pdf. [Accessed 30 10 2019].

[15] "Convolutional Neural Networks (LeNet) – DeepLearning 0.1 documentation," , . [Online]. Available: http://deeplearning.net/tutorial/lenet.html. [Accessed 30 10 2019].

[16] T. . Marianne, "#Deep Learning - fast.ai · Making neural nets uncool again," , 2018. [Online]. Available: https://thierry.marianne.io/2018/deep-learning---fastai-making-neural-nets-uncool-again. [Accessed 30 10 2019].

[17] C. E. Perez, "PyTorch, Dynamic Computational Graphs and Modular Deep Learning," , . [Online]. Available: https://medium.com/intuitionmachine/pytorch-dynamic-computational-graphs-and-modular-deep-learning-7e7f89f18d1. [Accessed 30 10 2019].

[18] "Facebook AI Research," , . [Online]. Available: https://research.facebook.com/ai. [Accessed 30 10 2019].

[19] L. N. Smith, "Cyclical Learning Rates for Training Neural Networks," , 2017. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7926641. [Accessed 30 10 2019].

[20] Y. J. Mo, J. . Kim, J.-K. . Kim, A. . Mohaisen and W. . Lee, "Performance of deep learning computation with TensorFlow software library in GPU-capable multi-core computing platforms," , 2017. [Online]. Available: https://ieeexplore.ieee.org/document/7993784. [Accessed 30 10 2019].

[21] "High Performance Computing - Supercomputing with Tesla GPUs," , . [Online]. Available: http://www.nvidia.com/object/tesla-supercomputing-solutions.html. [Accessed 30 10 2019].

[22] M. R. Karim, "TensorFlow: Powerful Predictive Analytics with TensorFlow: Predict valuable insights of your data with TensorFlow," , 2018. [Online]. Available: https://amazon.com/tensorflow-powerful-predictive-analytics-valuable-ebook/dp/b07bhrms9s. [Accessed 30 10 2019].

[23] "Why use Keras?," , . [Online]. Available: https://keras.io/why-use-keras/. [Accessed 30 10 2019].

[24] I. . Culjak, D. . Abram, T. . Pribanić, H. . Dzapo and M. . Cifrek, "A brief introduction to OpenCV," , 2012. [Online]. Available: https://ieeexplore.ieee.org/document/6240859. [Accessed 30 10 2019].

[25] T. d. Kok, "Combine Stata with Python using the Jupyter Notebook," , 2016. [Online]. Available: https://ideas.repec.org/p/boc/scon16/2.html. [Accessed 30 10 2019].

[26] "About · GitHub," , . [Online]. Available: https://github.com/about. [Accessed 30 10 2019].

[27] "Kaggle - Our Team," , . [Online]. Available: http://www.kaggle.com/pages/team. [Accessed 30 10 2019].

[28] Rafii; Zafar ; Liutkus; Antoine ; Fabian-Robert; Mimilakis; Stylianos Ioannis; Bittner; Rachel;, "The {MUSDB18} corpus for music separation," *MUSDB18*, vol. https://doi.org/10.5281/zenodo.1117372, 2017.