# **SUMMARY**

**Topic title:** Extract singing voice from stereo music using convolutional neural networks.

**Student name:** Duong Huynh Son

**Student ID:** 102150242

**Class:** 15TCLC1

Nowadays, along with the rapid growth of current music industry and karaoke industry, not only artists need singing voice audio for making their own product but the music lovers also need the instruments audio for their karaoke entertainment.

The aim of this project was to create a system which helps the users to extract singing voice from their music as input. They can use this system as multiple purposes such as making their own music mixture product, or helping users to take the singing voice for analysing work in the field of music information retrival or creating their karaoke audio file for the entertainment purposes.

I have done sufficiently from the conception to design of this work, as well as the analysis and implementation of the project. I used Python and its libraries to build the final application, with the help of Google Colaboration Platform supporting cloud GPU for training and testing process.

|  |  |
| --- | --- |
| DA NANG UNIVERSITY  **UNIVERSITY OF SICIENCE AND TECHNOLOGY**  FALCUTY OF INFORMATION TECHNOLOGY | **THE SOCIALIST REPUBLIC OF VIETNAM**  Independence - Freedom - Happiness |

**GRADUATION PROJECT REQUIREMENTS**

Student Name: Dương Huỳnh Sơn Student ID : 102150242

Class: 15TCLC1 Major: Information Technology

*Topic title:*

Extract singing voice from stereo music using Convolutional Neural Networks

1. *Project topic :* ☐*has signed intellectual property agreement for final result*
2. *Initial figure and data:*

Data from MUSDB18 dataset of 150 full length music tracks (~10 hours duration) of different genres along with their isolated *drums, bass, vocal* and *others* stems.

*Content of the explanations and calculations:*

The content of the explanations contains 5 parts:

* 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 analyzing works and Neural Networks design for this problem.
* Chapter 3: Implementation and Result Evaluation: This chapter shows an implementation of this project, including pictures and a brief explanation for each main function.
* Conclusion: The concluding section of the project simultaneously emphasizes the problem solved, as well as presenting issues still unresolved and provides recommendations and suggestions.
* References: Presentation about detail of referenced information used in this thesis.

1. Name of instructor: Assoc. Prof. Nguyen Tan Khoi.
2. *Date of assignment : 15/08./2019.*
3. *Date of completion : 15/12./2019.*

|  |  |
| --- | --- |
|  | *Đà Nẵng, 18 / 12 / 2019* |
| **Head of Division**…………………. | **Instructor** |

# **PREFACE**

I would first like to thank my instructor, Assoc. Prof. Dr. Nguyen Tan Khoi for his continuous support, supervision, motivation, and guidance throughout the tenure of my project in spite of his hectic schedule. He remained a driving spirit in my project and his experience gave me the understanding in handling research projects as well as helping me to clarify the abstruse concepts, requiring knowledge and perception, handling critical situations and in understanding the objective of my work.

I also want to thank my family and friends, who gave me the strength and confidence during my time of learning and during the implementation phase of this project. They have given a lot of love and encouragement for me which helped pass over the difficulties and fatigues.

Without their generous help, my senior year would not have been successful.

Sincerely,

# **ASSURRANCE**

I guarantee:

**1.** The contents of this senior project are performed by myselves following the guidance of instructor Assoc. Prof. Dr. Nguyen Tan Khoi.

**2.** All references used in this senior project thesis, are quoted with the author’s name, project name, time and location to publish clearly and faithfully.

**3.** All invalid copies, educated statute violation or cheating will be borne the full responsibility by myself.

Student Performed

Dương Huỳnh Sơn

# **TABLE OF CONTENT**

[SUMMARY i](#_Toc27526778)

[PREFACE iii](#_Toc27526779)

[ASSURRANCE iv](#_Toc27526780)

[TABLE OF CONTENT v](#_Toc27526781)

[LIST OF TABLES, PICTURES vii](#_Toc27526782)

[LIST OF SYMBOL, ACRONYM viii](#_Toc27526783)

[INTRODUCTION 9](#_Toc27526784)

[**1.** **Reason for doing thesis** 9](#_Toc27526785)

[**2.** **Scope and Objective** 9](#_Toc27526786)

[**3.** **Methods** 9](#_Toc27526787)

[1. Chapter 1: THEORIES AND TECHNOLOGIES 11](#_Toc27526788)

[**1.1.** **Signal** 11](#_Toc27526789)

[**1.1.1.** **Signal representation in modern computer** 11](#_Toc27526790)

[**1.1.2.** **Short-time Fourier Transform (STFT):** 12](#_Toc27526791)

[**1.2.** **Neural Networks** 13](#_Toc27526792)

[**1.2.1.** **Simple Perceptron** 13](#_Toc27526793)

[**1.2.2.** **Neural Networks** 14](#_Toc27526794)

[**1.3.** **Convolutional Neural Networks** 15](#_Toc27526795)

[**1.4.** **Building and Training CNNs with Fastai library – Pytorch:** 15](#_Toc27526796)

[**1.5.** **Google Colaboration:** 16](#_Toc27526797)

[**1.6.** **Other technologies** 16](#_Toc27526798)

[**1.7.** **Conclusion** 17](#_Toc27526799)

[2. Chapter 2: ANALYSIS AND DESIGN 18](#_Toc27526800)

[**2.1.** **Main features** 18](#_Toc27526801)

[**2.1.1.** **Extract audio data from Youtube video** 18](#_Toc27526802)

[**2.1.2.** **Play, pause and stop playing music** 18](#_Toc27526803)

[**2.1.3.** **Extract input audio to vocal-only and karaoke audio file** 19](#_Toc27526804)

[**2.2.** **Datasets** 19](#_Toc27526805)

[**2.3.** **Extract Spectrogram from audio file** 20](#_Toc27526806)

[**2.4.** **Network architecture** 21](#_Toc27526807)

[**2.4.1.** **U-Net architecture** 21](#_Toc27526808)

[**2.4.2.** **Skip connections** 22](#_Toc27526809)

[**2.4.3.** **Network architecture for singing voice extraction problem** 22](#_Toc27526810)

[**2.4.4.** **Reconstruction signal from Spectrogram** 23](#_Toc27526811)

[3. CHAPTER 3: IMPLEMENTATION AND RESULT EVALUATION 25](#_Toc27526812)

[**3.1.** **Implementation** 25](#_Toc27526813)

[**3.1.1.** **Train model** 25](#_Toc27526814)

[**3.1.2.** **Model Evaluation:** 25](#_Toc27526815)

[**3.1.3.** **General Diagram** 26](#_Toc27526816)

[**3.1.4.** **Setting environment and install application** 27](#_Toc27526817)

[**3.1.5.** **Using the application** 27](#_Toc27526818)

[**3.2.** **Result Evaluation** 29](#_Toc27526819)

[**3.2.1.** **Evaluation data** 29](#_Toc27526820)

[**3.2.2.** **Result Evaluation** 29](#_Toc27526821)

[CONCLUSION 33](#_Toc27526822)

[**1.** **Result** 33](#_Toc27526823)

[**2.** **Future work** 33](#_Toc27526824)

[REFERENCES 35](#_Toc27526825)

# **LIST OF TABLES, PICTURES**

TABLE 1.1 {size 13}.........................................................................................................

TABLE 1.2 ……................................................................................................................

TABLE 1.3 ……................................................................................................................

…….……..........................................................................................................................

PICTURE 1.1 ...................................................................................................................

PICTURE 1.2 ....................................................................................................................

PICTURE 1.3 ....................................................................................................................

…….……..........................................................................................................................

***Ghi chú:***

* Mỗi table, hình vẽ/ sơ đồ phải được đánh số và có tên;
* Đánh số table và đánh số hình vẽ/ sơ đồ riêng. Quy luật đánh số như sau:
  + Chữ số thứ nhất chỉ tên chương;
  + Chữ số thứ hai chỉ thứ tự table biểu, sơ đồ, hình,…trong mỗi chương.

# **LIST OF SYMBOL, ACRONYM**

SYMBOL:

…….……..........................................................................................................................

…….……..........................................................................................................................

…….……..........................................................................................................................

…….……..........................................................................................................................

…….……..........................................................................................................................

…….……..........................................................................................................................

ACRONYM:

…….……..........................................................................................................................

…….……..........................................................................................................................

…….……..........................................................................................................................

…….……..........................................................................................................................

*Ghi chú:*

* Ký hiệu: mỗi mục ký hiệu gồm ký hiệu và phần tên gọi, diễn giải ký hiệu.
* Cụm từ viết viết tắt là các chữ cái và các ký hiệu thay chữ được viết liền nhau, để thay cho một cụm từ có nghĩa, thường được lặp nhiều lần trong đồ án.

# **INTRODUCTION**

* + - 1. **Reason for doing thesis**

There are a lot of music songs in the market but are not published the vocal and karaoke version separately, which cannot meet the singing and entertainment demand of many artists and music lovers. With the singing voice only audio, users and artists use that to create their own new mixture. In addition, it can be used in extracting information from singing voice to lyrics text and also be used in searching and storing data, as well as tagging contents based on these singing voice-alone audios. With the karaoke audios, music lovers can sing on those audios without noises from the original singer’s singing voice, from there, many cover versions have been released, which are also, sometimes, even much more popular than the original one.

* + - 1. **Scope and Objective**

In this project, I am researching in applying Convolutional Neural Networks trained on MUSDB18 datasets provided by SigSep with the good result. With this method, all the songs data are converted to Spectrogram based on Short-time Fourier Transform. This is the most popular way to feed an audio file into a Neural Networks.

The main objective is researching to create an application for end-user to input their Youtube video link of song, then provide users 2 file vocal-only and non-vocal music audio.

* + - 1. **Methods**

*Observation method:* Observe the need of artists and music lovers who need the vocal-only and non-vocal audio files for their interest and works.

*Theories analyzing method:* Research about related papers, works all around the world to find out which current method is applied to solve related problems such as signal processing methods, data analyzing, data transformations, etc. From those researches to summarize information, chose methods to apply into this problem. I did many researches and jump to the method of using Convolutional Neural Networks with U-Net architecture to extract desired data from Spectrogram of audio input data.

*Experimental method:* Use the current applications related to problem to conclude the necessary functions for my application in this thesis.

*Modeling method:* This is the last method of this thesis. After doing and getting information from above methods, I modeled the project by plotting diagrams and figures to show the big picture of the thesis and develop the final usable application.

# **1. Chapter 1: THEORIES AND TECHNOLOGIES**

This system applies many technologies related to sound, signal and deep learning that combine to create a system can provide exactly what users need – vocal audio file and karaoke file. To learn more about the technologies, I would like to present the concept of some key technologies that I used in this project.

Start with sound and signal representation, and some methods that I am using in this work.

## **Signal**

### **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.

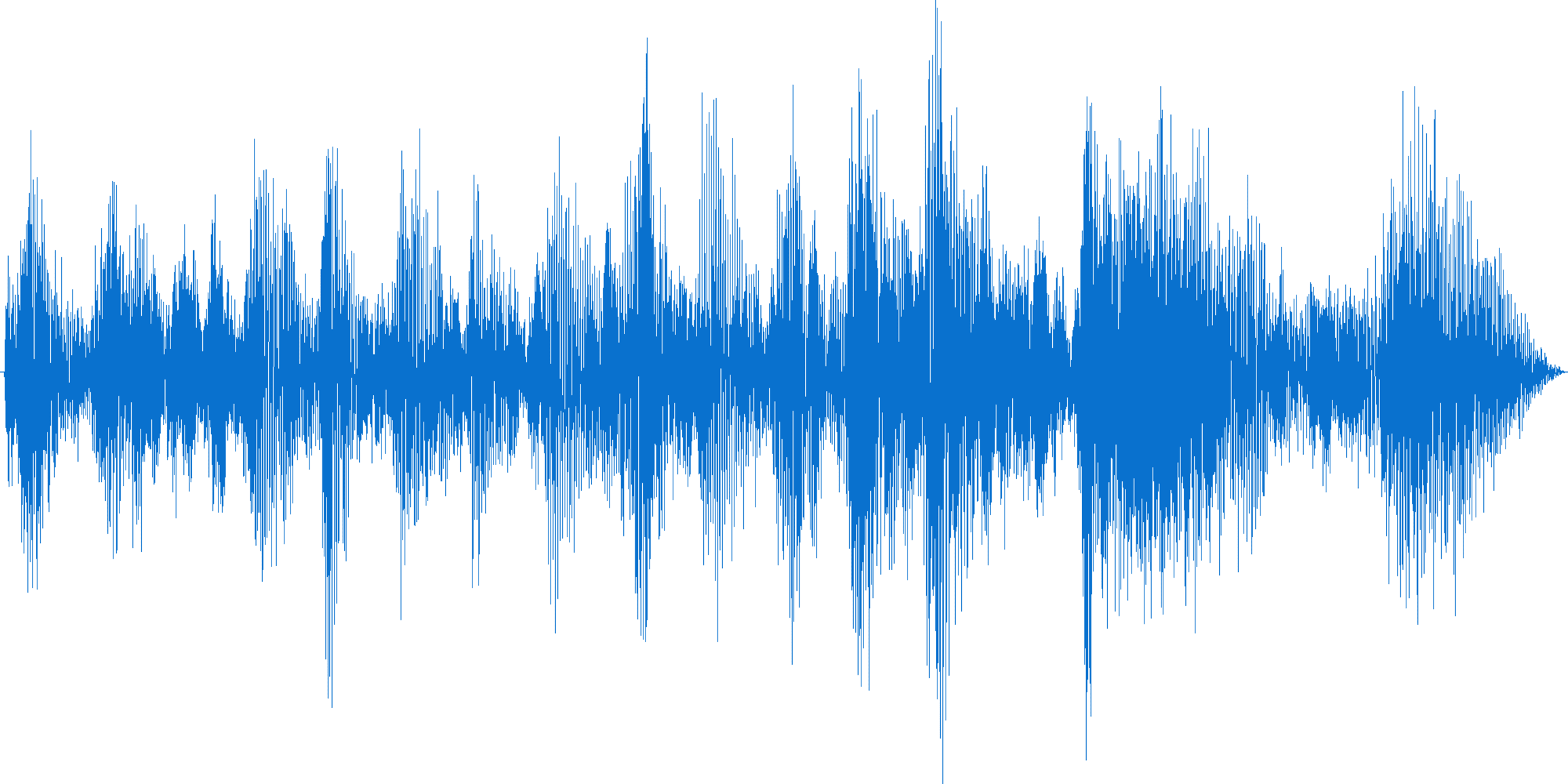


Figure.1.1: Waveform sample

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!

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

* + 1. **Short-time Fourier Transform (STFT):**

The Short-time Fourier transform (STFT), is a Fourier-related transformation used to determine the frequency and phase information of small section of signal as it changes over time.

A close up of text on a white background

Description automatically generated

Figure 1.2: Fourier Transform (credit: towarddatascience.com)

The above figure describes how the waveforms can be converted to Frequency Domain; therefore, I can get Amplitude and Frequency of audio clips. If we have the window size for the time domain small enough, we can get more information about Time. That is the main idea of STFT.

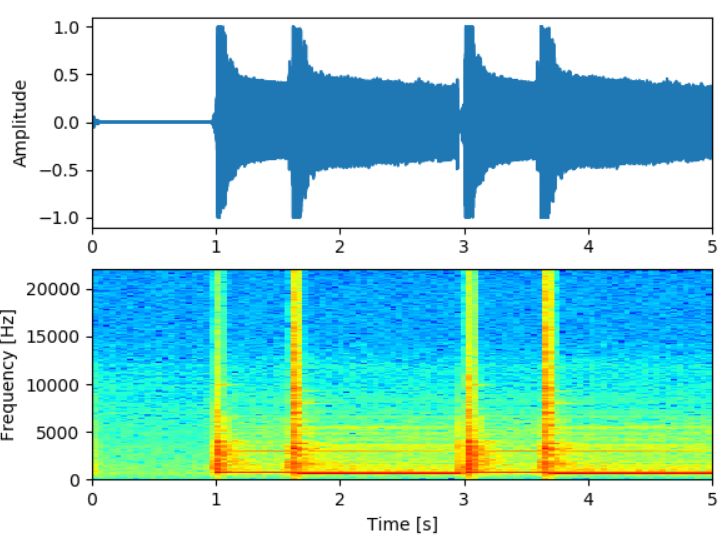


Figure 1.3: Waveform and STFT Spectrogram (credit: stackexchange.com)

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. It leads is to another approach to solve the problem, which shifts from signal processing into the problem into the problem of image to image translation.

## **Neural Networks**

### **Simple Perceptron**

Perceptron is the simplest unit of Neural Networks, developed in 1960 by Frank Rosenblatt, inspired by previous researches of Warren McCulloch and Walter Pitts.

A perceptron receives some binary data and provide a single binary output.

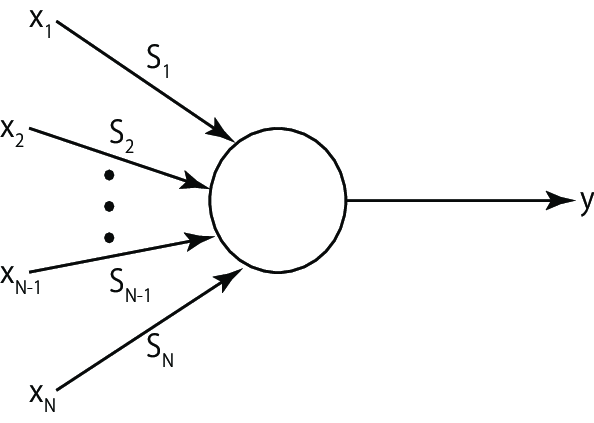


Figure 1.4: Simple Perceptron

In the above figure, perceptron receive N input X1, X2, …, XN with N parameters S1, S2 , … , SN  to produce 1 single output y.

* + 1. **Neural Networks**

Neural networks are a set of algorithms, 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. In this problem, the input of Neural Networks is a 2D matrix and output is a corresponding 2D matrix representing Spectrogram of audio clips.

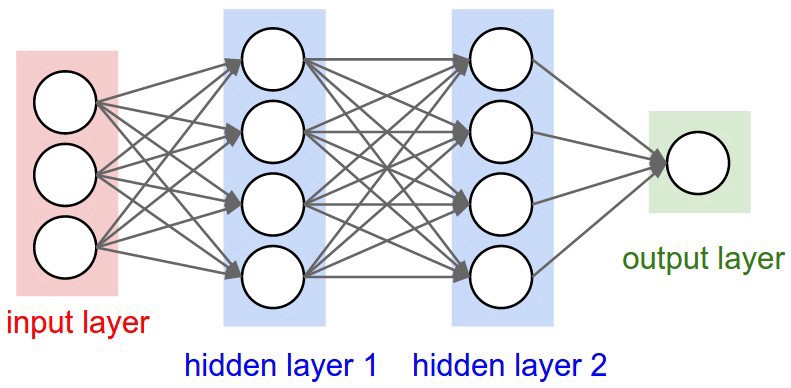


Figure 1.5: Simple Neural Networks (credit: medium.com)

The above Neural Network contains 3 layers:

* Input layer: The very left layer of the Network, represent the input of the Network.
* Output layer: The very right of the Network, represent the output of the Network.
* Hidden layer: In between Input layer and Output layer.

In each Neural Network, there can be several hidden layers but only 1 Input layer and 1 Output layer. In each layer, number of node and perceptron can be different depending on the complexity of problem. All the perceptron can be linked together into Fully Connected Network, then we can calculate the size of the Network based on the number of layers and perceptron. In the above Network contains:

* 4 layers.
* 12 nodes.
* 3 \* 4 + 4 \* 4 + 4 \* 1 = 32 parameters (weights).
  1. **Convolutional Neural Networks**

Convolutional neural networks are neural networks used primarily to classify images (i.e. labeling images), 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.

* 1. **Building and Training CNNs with Fastai library – Pytorch:**

FASTAI 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, which built and supported by Facebook AI. In addition, FASTAI has many modern and great functions for us to implement our algorithms.

One of the greatest things comes from Fastai is the one cycle learning, 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. **Google Colaboration:**

Training a deep learning networks is a very expensive process; therefore, we need a fast-enough GPU 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 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, Keras, and OpenCV. You can create notebooks 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, upload Kaggle files, download your notebooks, and do just about everything else that you might want to be able to do.

* 1. **Other technologies**

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

* 1. **Conclusion**

By studying and learning about the above technologies, we successfully applied the concepts and their mechanism operating in this project to create an application which can extract singing voice and karaoke audio file from music audio file.

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.

1. **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.

* 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.

* + 1. **Extract audio data from Youtube video**

In the main application, when the main UI shows up, users can enter Youtube video link, the application will download the video and extract audio data from that video.

This feature does help users in finding and getting audio file easier. Instead of manually find the audio file and feed into application, this system can help users do that part, so it can be very easy to use since Youtube is the largest video social media in the Internet nowadays.

* + 1. **Play, pause and stop playing music**

After downloading audio files from internet, users can play, pause and stop playing that clip of audio. This feature helps users determine the right song and audio after getting it from internet.

This feature also helps users play the output files which are vocal-only file and karaoke audio file.

* + 1. **Extract input audio to vocal-only and karaoke audio file**

This is the main feature and the goal of this thesis which I want to achieve. The function takes audio files as input and provide 2 audio files (vocal and karaoke) using Convolutional Neural Networks.

* 1. **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 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).

All files from the musdb18 dataset are encoded in the Native Instruments stems format (.mp4). It is a multitrack format composed of 5 stereo streams, each one encoded in AAC @256kbps. These signals correspond to:

* The mixture
* The drums
* The bass
* The rest of the accompaniment
* The vocals.

For each file, the mixture corresponding to the sum of all the signals.

Below is the data table of genre:

*Table 2.1: Dataset Musdb18 structuring on genres*

|  |  |
| --- | --- |
| Genre | Number of songs |
| Country | 3 |
| Electronic | 8 |
| Heavy Metal | 12 |
| Jazz | 3 |
| Pop | 11 |
| Pop/Rock | 72 |
| Rap | 8 |
| Reggae | 2 |
| Rock | 17 |
| Other | 14 |
| Total | **150** |

* 1. **Extract Spectrogram from audio file**

In general, the method we are using to store audio file is store it as the waveform format, as the figure below.

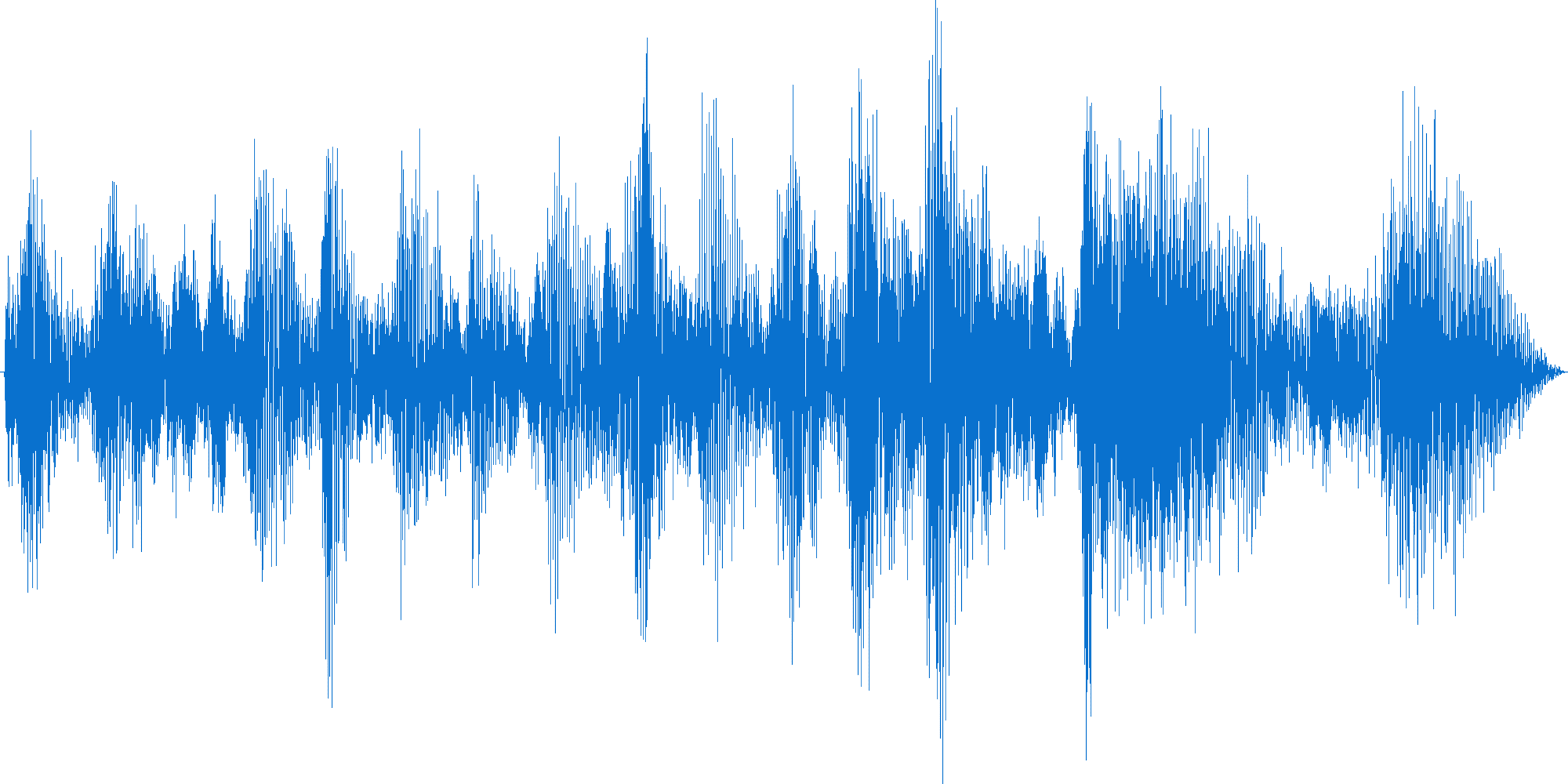
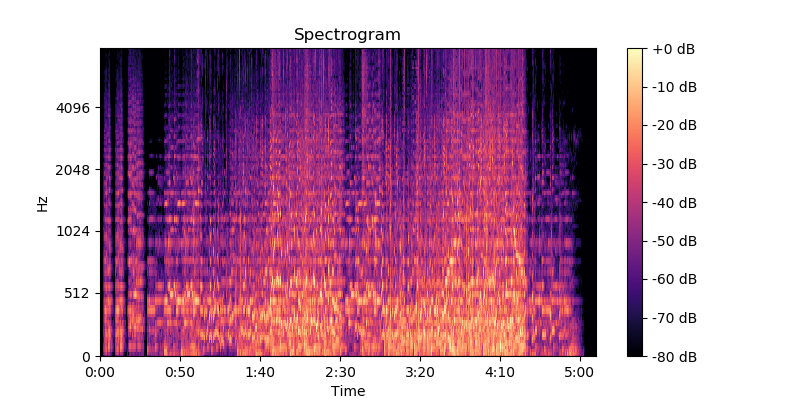


Figure 2.1: Waveform representation of audio in computer

As the above figure, we can only witness the information of time and amplitude of the waveform, it is very hard for us to determine the frequency of data signal through time, can lead to the uncertain in analyzing and extracting information.

So, Spectrogram is the more precise way to representing audio data. As it contains information about amplitude of frequency in corresponding time.



*Figure 2.2: Spectrogram representation of audio*

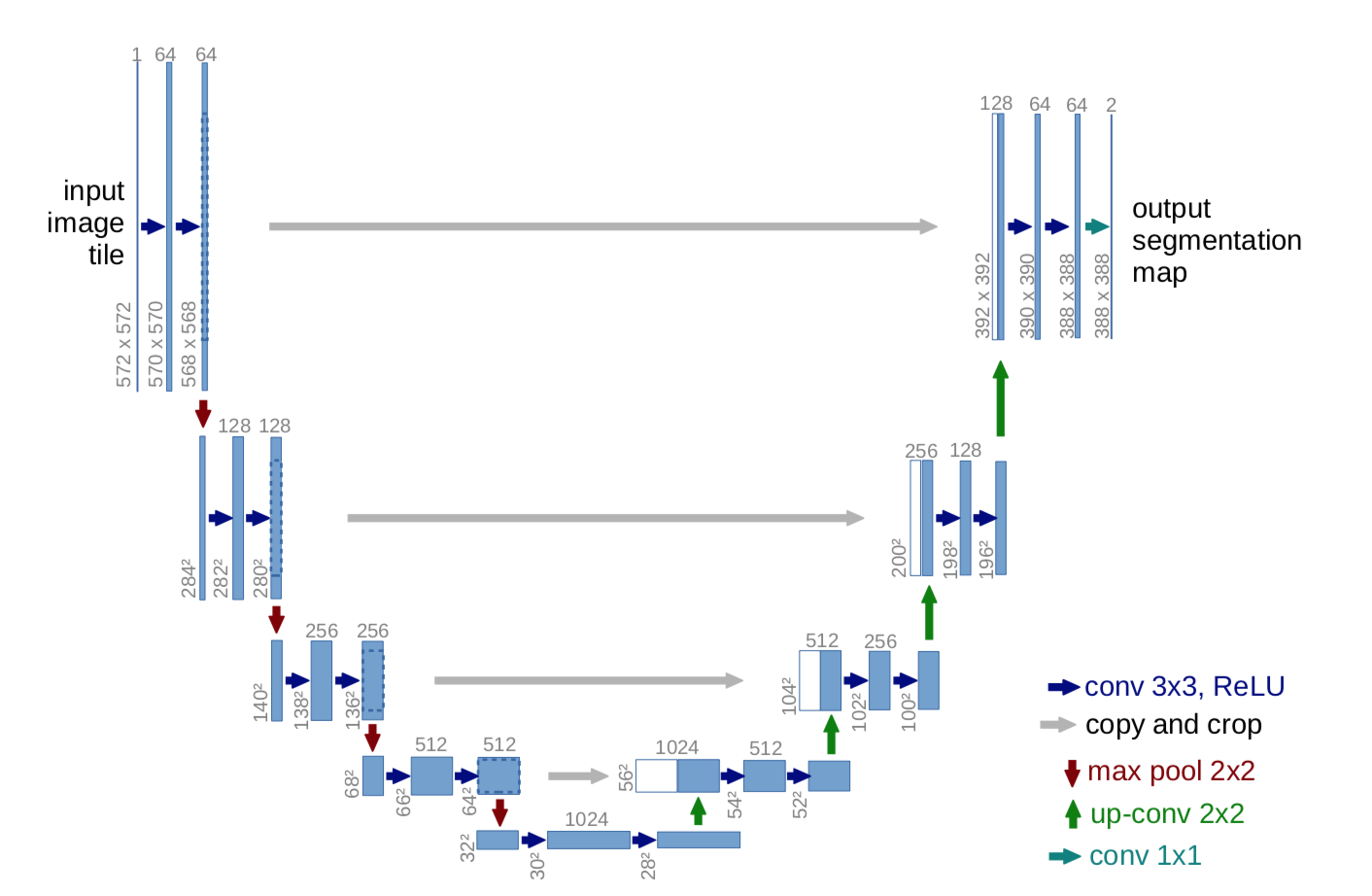
* 1. **Network architecture**

The main method I will be using is apply Convolutional Neural Networks to analyze and solve the problem, with the input of audio spectrogram based on Short-time Fourier Transform method, output is the corresponding spectrogram of vocal-only audio.

The goal of Network is predicting the output Spectrogram of vocal-only audio file when we feed the input song’s spectrogram, the size of input and output spectrogram is the same.

* + 1. **U-Net architecture**

U-Net architecture is firstly used in extracting image of biologic cell, with the advantage of increasing precision of cell segmentation in medical image. The network produce output of 2 value: 0 – non-cell, 1 – cell.



*Figure 2.3: U-Net architecture in cell segmentation problem*

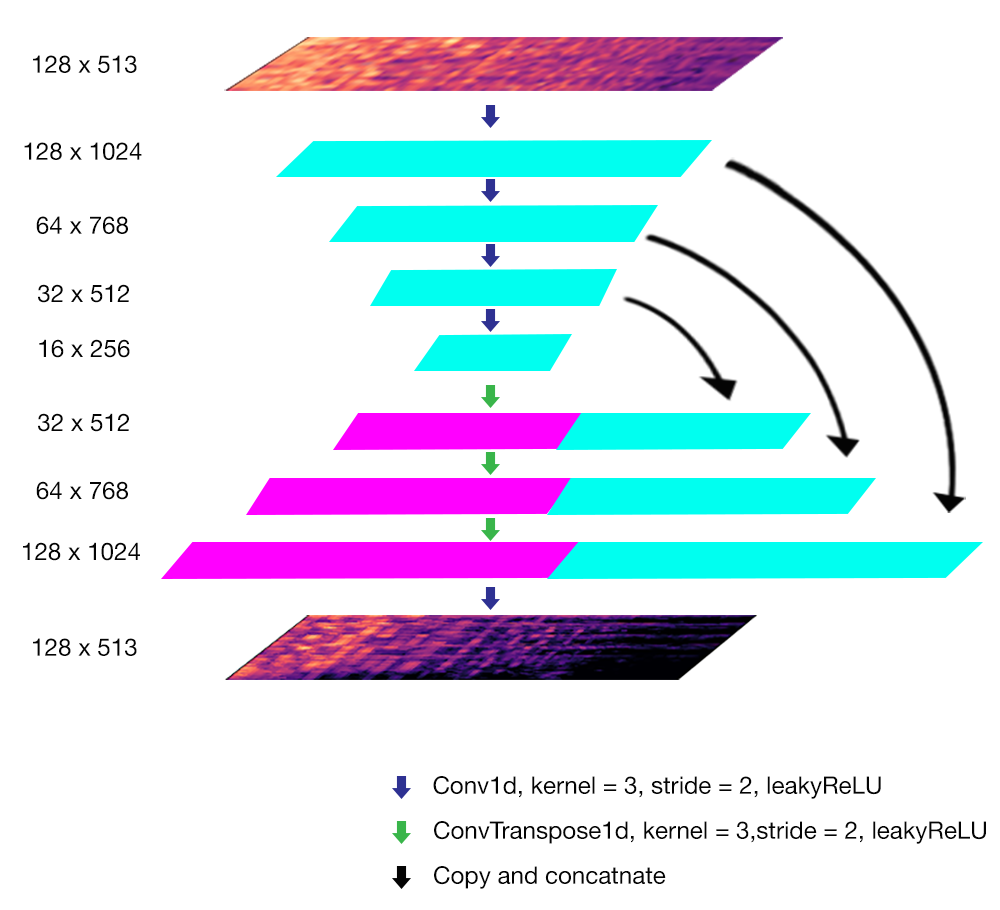
U-Net architecture is built based on encoder-decoder model. Encoder is the part which contains many convolution layers to reduce input matrix dimension while keep the useful information for output calculation. Decoder part takes responsibility for taking output from encoder part and reproduce the information based on those data.

* + 1. **Skip connections**

If we assume the processed spectrograms in the previous section as an image, we can see instead of produce output of 2 value 0 and 1, we need to predict the real value of each pixel in the spectrogram image which range from 0 to 1. In the segmentation problem, some of the false output values make no huge effect to the overall result, but in this problem, 1 wrong pixel can lead to the error of frequency in the output audio. Therefore, I need to build and experiment a Network with the high precision. In the U-Net architecture, there are skip connections between layers and layers, information of a layer in Encoder will be straightly transfer and concatenate to the later corresponding layer, which have the same size. Thank to those skip connections, information can be kept and help the network produce the more precise output.

* + 1. **Network architecture for singing voice extraction problem**

In this project, I use 4 convolution layers, size of *m \* 3* with *m \* n* is the size of encoder’s input, followed by 3 ConvTranspose layer, and simultaneously copy and concatenate information directly from the corresponding layers in Encoder part to these layers. Finally, I use a Convolution layer to produce the final output. The network architecture can be visualized as figure.



*Figure 2.4: Network architecture for singing voice extraction problem*

As we can see, I am using the 1D Convolution layers instead of 2D Convolution layers because of the better performance when processing. 2D Convolution in this Network is not performing well with the long processing time, therefore, 1D Convolution with above architecture is the better choice for me.

* + 1. **Reconstruction signal from Spectrogram**

Neural Network uses magnitude of the frequency as an input for processing data, provide us output of the same size magnitude of frequency matrix. Phase information of the output value is taken from the phase of input value. From those information of magnitude and phase of frequency through time, I use the *Inverse Fourier Transform* to reconstruct the signal back to time domain according to this formula:

With:

* *y:* the desired signal in time domain
* *:* Inverse Short-time Fourier Transform
* *s:* Magnitude value (Neural network’s output)
* *i:* Imagine unit in complex number (*i =* )
* *angle:* Phase information taken from input STFT
* *x:* input signal in time domain.

From that, we can conduct a formula for the entire Spectrogram matrix to reconstruct signal:

With:

* *X:* vector of input signal in time domain
* *Y:* desired vector of output signal in time domain
* *S:* magnitude matrix converted from *X* using *STFT.*

After getting the signal data of singing voice, we can extract karaoke signal according to the following formula:

With:

* *X:* vector of song music data (karaoke + vocal)

1. **CHAPTER 3: IMPLEMENTATION AND RESULT EVALUATION**
   1. **Implementation**
      1. **Train model**

I trained the model on Musdb18 dataset with the MSE loss function defined as:

With:

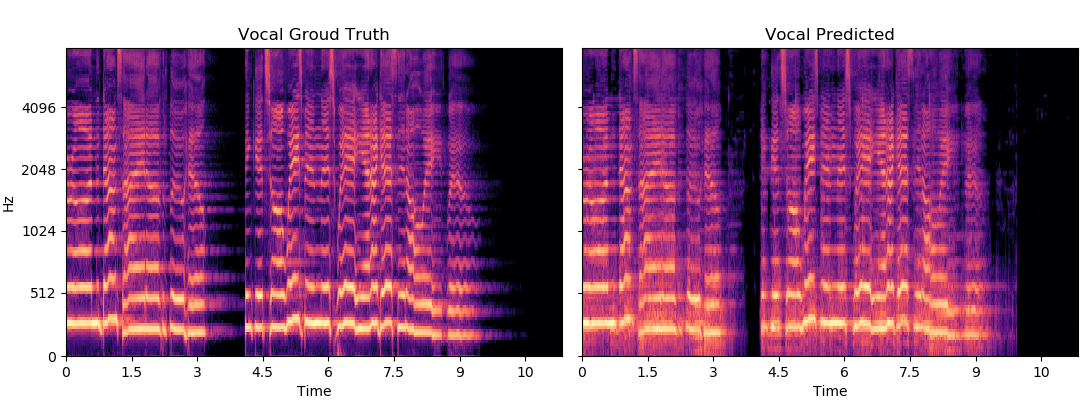
* : Neural Network’s output with parameters
* *Y:* Ground Truth
* *X:* Input matrix

After 60 epoch training with Cycle Learning method, we shifted from MSE loss to MAE loss to reduce the effect of noises to the output. MAE loss is defined as below:

With:

* : Neural Network’s output with parameters
* *Y:* Ground Truth
* *X:* Input matrix
  + 1. **Model Evaluation:**

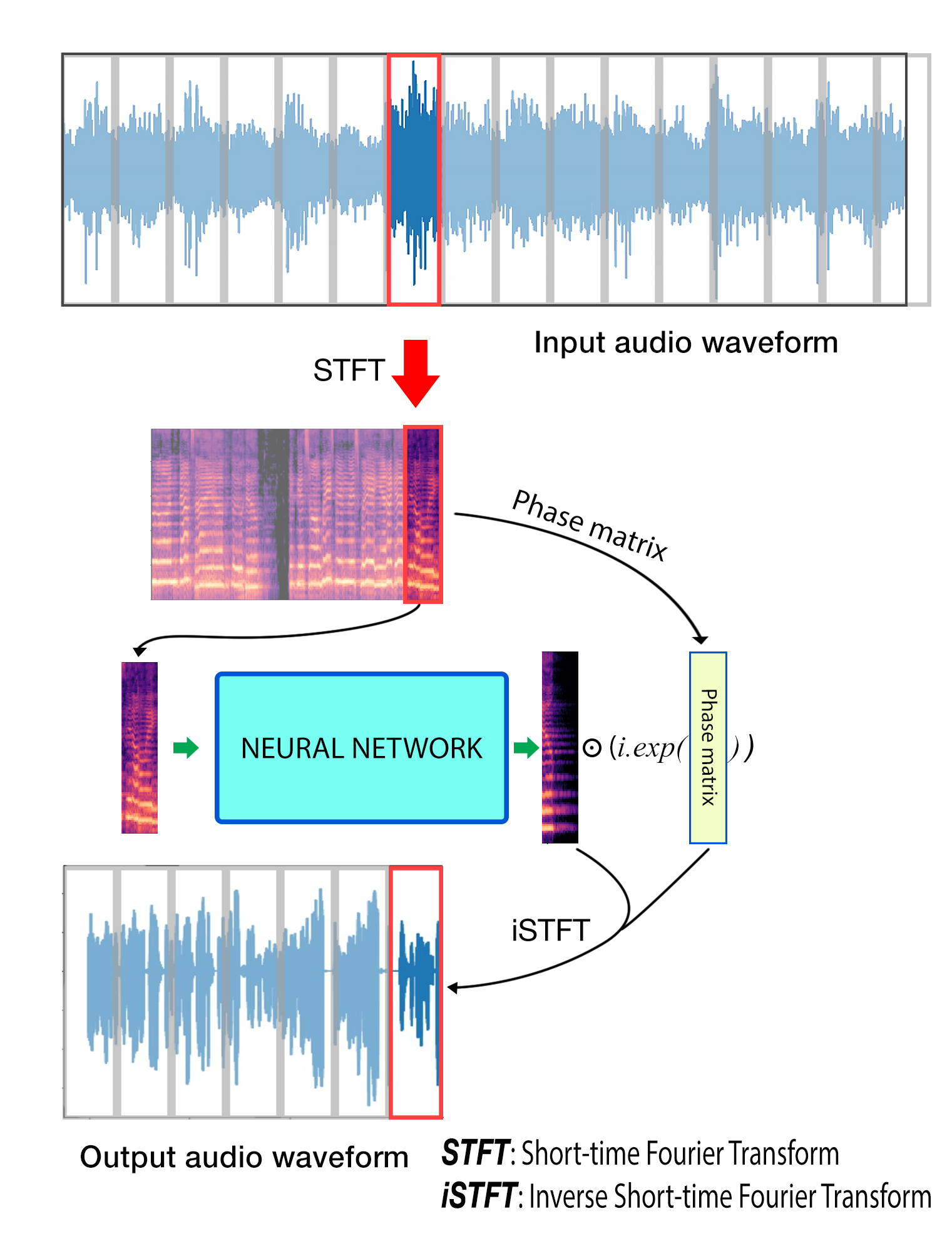
After training 100 epoch, our MSE metric has value ~ 0.006, MAE value ~0.024.



*Figure 3.1: Ground Truth and Predicted Spectrogram comparison.*

* + 1. **General Diagram**

From audio files with *\*.mp3, \*.wav,* the system will process and produce 2 audio files of vocal-only and karaoke audio file according to the diagram below.



*Figure 3.2: General Diagram*

* + 1. **Setting environment and install application**

To avoid conflict with other environments in the system, we use Anaconda to manage virtual Python environment. In this project, we use Windows 10 for running application.

* Install Anaconda for Windows

We can download and install Anaconda for windows at <https://www.anaconda.com/distribution/#download-section>

* Set up virtual environment in Anaconda

After downloading and installing Anaconda, we can create environment for running application in Anaconda Prompt:

conda create --name audio python=3.7

activate audio

conda install -c pytorch pytorch

conda install -c conda-forge ffmpeg

* Download application and install libraries

git clone <https://github.com/sonduong305/Vocal-separator.git>

cd Vocal-separator

pip install requirements.txt

* Run the application

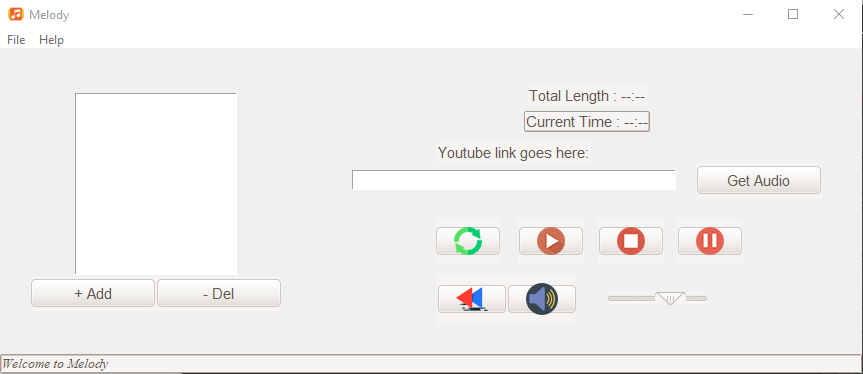
python main.py

* + 1. **Using the application**

After installing and setting up the environment, we can run our application with command

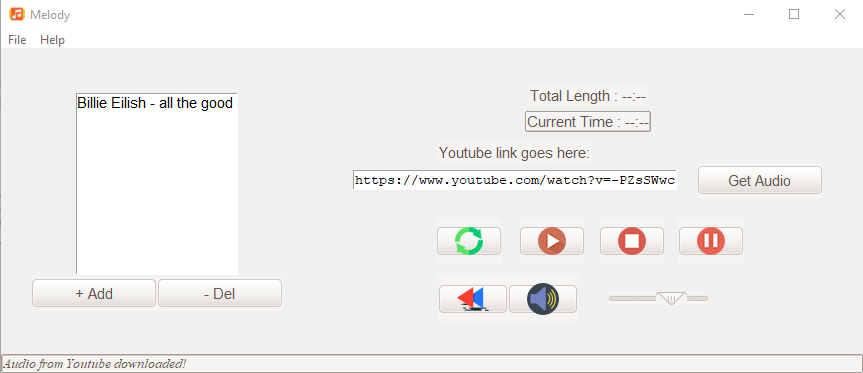
python main.py

Below are some image of application:



*Figure 3.3: Main UI*

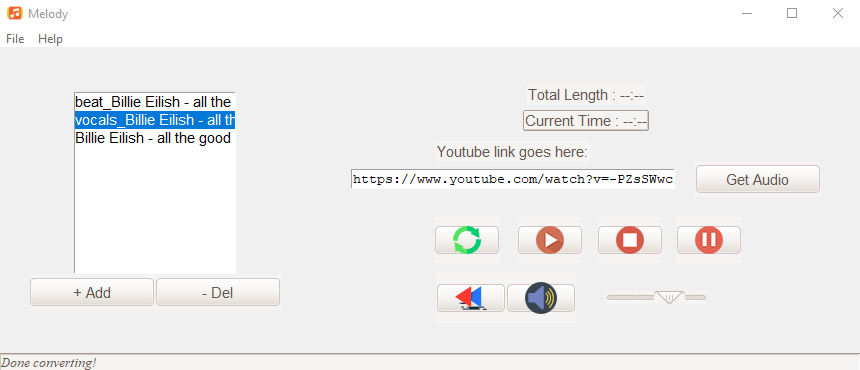
After pasting Youtube video link to textbox and hit “Get Audio”, Youtube audio file will be downloaded to local machine and ready to convert and extract.



*Figure 3.4: Download audio from Youtube and list to playlist*

Now we can select the audio file from the list, hit  button, then the application will start extracting karaoke and vocal audio file from selected audio.

After getting vocal and karaoke file, we can play those audios by selecting it and hit **play** button.



*Figure 3.5: Extract vocal and karaoke file from audio*

* 1. **Result Evaluation**

The result is evaluated in Musdb18 test data.

* + 1. **Evaluation data**

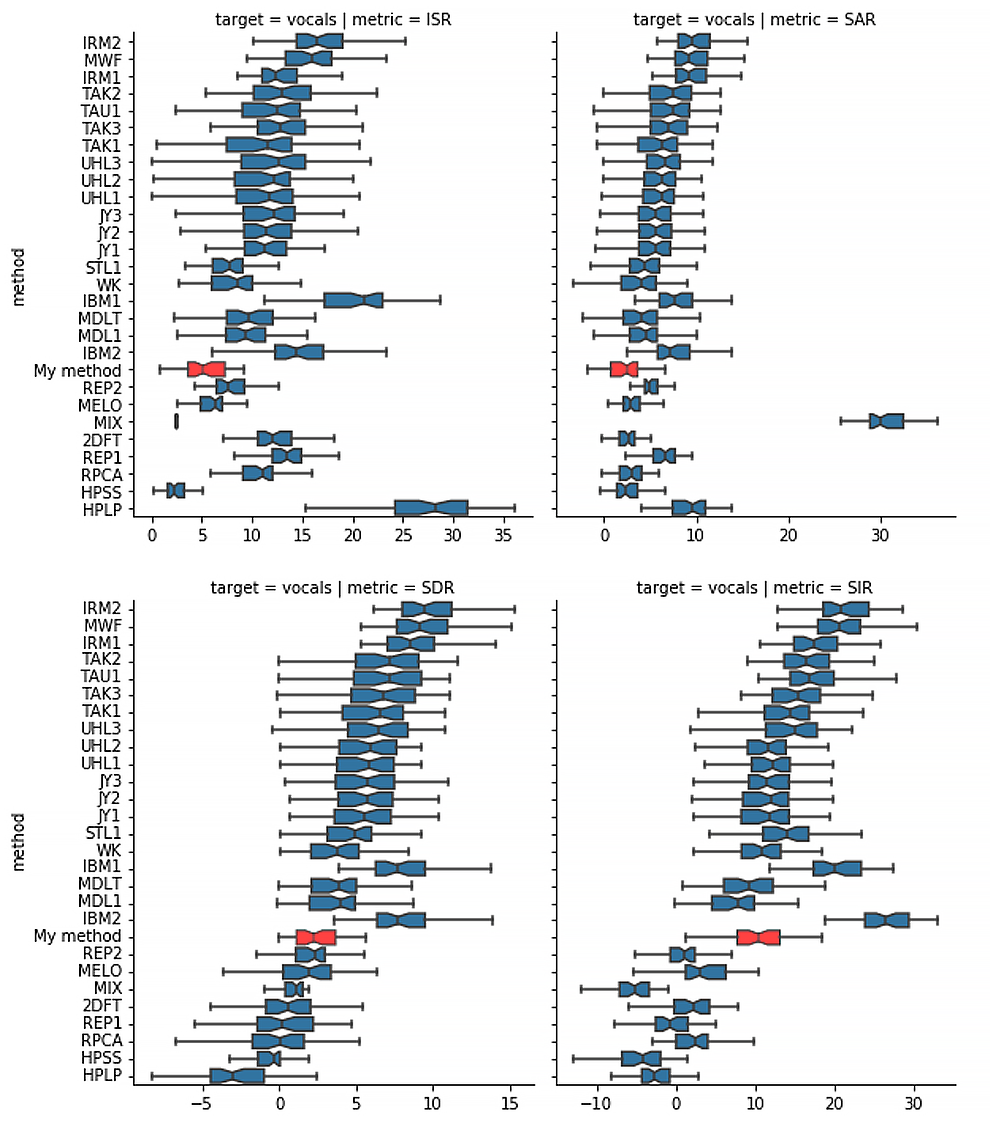
Evaluation data has the total length of 3.5 hour long, divided up by genres:

*Table 3.1: Evaluation data in genres*

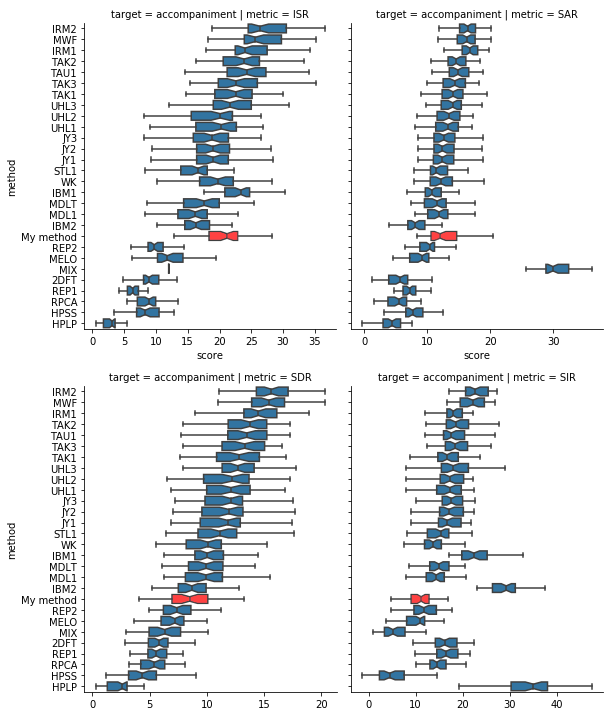
|  |  |  |
| --- | --- | --- |
| Genre | Number of songs | Percentage |
| Electronic | 4 | 8% |
| Heavy Metal | 4 | 8% |
| Pop/Rock | 36 | 72% |
| Rap | 3 | 6% |
| Reggae | 2 | 4% |
| Rock | 1 | 2% |
| Total | **50** | **100%** |

* + 1. **Result Evaluation**

Below are boxplot diagrams of the evaluation on Musdb18 test data, compare with current methods in the Campaign.



*Figure 3.3: Evaluation score on vocal*



*Figure 3.4: Evaluation score in Accompaniment (karaoke)*

Some of the above methods has been trained on many different datasets, the size of datasets is also big, therefore, their result is better than mine.

When hearing the output as audio file with the input of high quality English songs, the application can produce a good and usable result. There are still noises in some cases because of the small datasets cannot cover all the instruments and some singers have a very unique vocal, make our application confusing when doing prediction.

About processing time, application has the average time of processing and extracting of 40s for 180s input music on CPU Intel® Xeon® Processor E3-1231v3. With GPU accelerated, model can perform even better with 20s for 180s input music on GPU GeForce GTX 1050Ti.

# **CONCLUSION**

**Result**

After doing this project, I have some certain knowledge and achievements by research, experiment and build application.

About Theories

* Have basic knowledge about Neural Networks, Deep Learning, Machine Learning.
* Have knowledge about sound processing, signal processing, signal representations.
* Research about Convolutional Neural Networks and its application in Computer Vision and Signal Processing Area.

About Application

* Ability to work with modern Deep Learning Frameworks such as Pytorch, FASTAI, ..
* Created a usable application for extracting singing voice and karaoke from music audio with the user-friendly User Interface.

Nevertheless, this project still has many problems that need to solve:

* Need more data enough to cover the major instrumental sound, so the model can learn and extract more precisely.
* UI can be improved to have more functions and features for users to use.
* There are still some songs in different languages which makes the application get struggle when predicting due to the structure of sound in different languages with different voices.

**Future work**

With some disadvantage we have already mentioned in evaluation part, the

future work of this project is as follow:

* Collect more data for the model to train and predict more precisely. Data should cover enough instrumental sounds, singing voice from different languages and difference singer to get more general prediction from the network.
* Improve the User Interface to help users have more options for the extracted audio.
* As we see, the predicting speed is fast enough for real-time prediction, so in the future, there will be a feature in which users can listen to the output while processing input, get better User Experience.
* Using singing voice to convert from speech to text and render to the original video to make karaoke video.

# **REFERENCES**

# 