UNIVERSITATEA "ALEXANDRU-IOAN CUZA" DIN IAȘI

FACULTATEA DE INFORMATICĂ

LUCRARE DE LICENȚĂ

Music Recognition Using Convolutional Neural Networks

propusă de

Andrei Vavilov

Sesiunea: Iulie, 2021

Coordonator științific

Conf. Dr. Vitcu Anca

UNIVERSITATEA ”ALEXANDRU-IOAN CUZA” DIN IAȘI

FACULTATEA DE INFORMATICĂ

Music Recognition Using Convolutional Neural Networks

Andrei Vavilov

Sesiunea: Iulie, 2021

Coordonator științific

Conf. Dr. Vitcu Anca

Avizat,

Îndrumător lucrare de licență,

Conf. Dr. Vitcu Anca.

Data: .... Semnătura: ....

Declarație privind originalitatea conținutului lucrării de licență

Subsemnatul **Vavilov Andrei** domiciliat în **România, jud. Iași, orș. Târgu Frumos, Strada Tudor Vladimirescu, nr. 59**, născut la data de **05 iulie 1999**, identificat prin CNP **1990705225638**, absolvent al Facultății de informatică, **Facultatea de informatică** specializarea **informatică**, promoția 2021, declar pe propria răspundere cunoscând consecințele falsului în declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale nr. 1/2011 art. 143 al. 4 și 5 referitoare la plagiat, că lucrarea de licență cu titlul **Music Recognition Using Convolutional Neural Networks** elaborată sub îndrumarea doamnei **Conf. Dr. Vitcu Anca**, pe care urmează să o susțin în fața comisiei este originală, îmi aparține și îmi asum conținutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licență să fie verificată prin orice modalitate legală pentru confirmarea originalității, consimțind inclusiv la introducerea conținutului ei într-o bază de date în acest scop.

Am luat la cunoștință despre faptul că este interzisă comercializarea de lucrări științifice în vederea facilitării falsificării de către cumpărător a calității de autor al unei lucrări de licență, de diplomă sau de disertație și în acest sens, declar pe proprie răspundere că lucrarea de față nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Data: .... Semnătura: ....

Declarație de consimțământ

Prin prezenta declar că sunt de acord ca lucrarea de licență cu titlul **Music Recognition Using Convolutional Neural Networks**, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test, etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de informatică.

De asemenea, sunt de acord ca Facultatea de informatică de la Universitatea ”Alexandru-Ioan Cuza” din Iași, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Absolvent **Andrei Vavilov**

Data: .... Semnătura: ....

Contents

Motivation

In the age of big data and immense computational power, artificial intelligence has come to be the new standard in the computer science field. Various types of data can be understood, learnt, predicted and even produced by a well-tuned neuronal network, making the principles of machine learning a must for a scientist nowadays.

Applications of neuronal networks can be found in any discipline: from medicine to physics, social sciences and languages. The purpose of this thesis is to depict how artificial intelligence can find it's place and purpose in a previously profoundly human field: art.

Introduction

The current thesis is an attempt to materialize the intersection of artificial intelligence and arts, especially, music.

The application consists of two major parts:

- First, the machine learning component, represented by a neuronal network framework, implemented from scratch (i.e. without the explicit support of existent frameworks or libraries). The module presents the necessary functionalities for constructing a neuronal network: layers, activation functions, metrics, optimizers, models and support for data generation and preprocessing.

- Second, the visual support, compound of two illustrative animations, created in Autodesk Maya 2019, and a Graphical User Interface, in which a hypothetical user can interact with the application by feeding it an YouTube link of a song. As a result, the application will decide whether the input song is played on a piano or not (prediction Piano/Other). Depending on the decision of the neuronal network, the corresponding Maya animation will be played.

0.1 Context

The first analogy between the way computers can process information and the way the human brain works (as we know, at least, to this day), has been made by Warren McCullough and Walter Pitts in 1944, who later became the founding members of what is sometimes called the first cognitive science department [5]. The primary idea is elegant in it's simplicity: the neuron, the basis of the human cognitive apparatus, can be modelled in machine learning as an unit in a network.

Since then, numerous studies and breakthroughs have been made, as well as various frameworks and tools which make implementing a neuronal network accessible without possessing the full mathematical background needed prior.

0.2 Purpose and modus operandi

One of the purposes of this thesis is to implement the functionalities and the logic behind a neuronal network ab initio, as well as creating and fine-tuning a model. In order to complete this task, we consulted multiple sources(e.g. [6],[10],[8]) which presented the theoretical and mathematical aspects of constructing the aforementioned classifier. The implementation was constructed with the support of various tools from the Python programming language (e.g. Numpy, Librosa, Tenserflow Keras, Kapre), which will be extensively discussed in the following chapters.

As discussed before, the second component of the thesis regards the visual part of the application. For obtaining a interactive use of the project, we used Autodesk Maya 2019 (for creating the animations) and Python tkinkter for creating a simplistic GUI.

The logical flow of the pipeline is as follows: the user feeds an YouTube link of a song to the GUI. The model (previously trained and saved) computes a prediction regarding the category under which the input falls (Piano/Other). Given this result, the corresponding animation is played.

0.3 Personal contributions

Numerous attempts of creating a medium between artificial intelligence and other disciplines have been made since the rise of this field. Arts, especially music, is no exception.

The particularity of the current thesis is the approach we had in completing the task: implementing from scratch the neural network framework, and, implicitly, understanding the mathematical and theoretical subtleties of it, as well as creating the visual aid which aims to touch on (although briefly) 3D animations.

0.4 Structure

The structure of the present thesis follows the major constituent parts described before and is as follows:

1. Chapter One
   - Similar applications : in which other akin projects are mentioned;
2. Chapter Two
   - Technologies used : in which technologies needed for creating the application and their purpose and functionalities are discussed;
3. Chapter Three
   - Architecture and implementation : in which the actual implementation is discussed explicitly. This section contains technical and theoretical aspects of the thesis and showcases relevant code;
4. Chapter Four
   - Use cases: in which the functionalities of the project are presented.

0.5 Acknowledgements

I would like to express my special thanks of gratitude to the academical staff of the Faculty of Computer Science Iași for the opportunity to do this project.

Obviously, the present thesis would not be possible without the help and guidance of the professor that directed it, PhD. Anca Vitcu.

Chapter 1

Similar applications

Neural Network Models as well as Machine Learning algorithms provide various versatile and adaptive methods for non-linear problem solving. Because of these reasons , there are numerous applications which perform the task of musical/audio classification using the aforementioned techniques. We will discuss in the following sections about three of those applications.

1.1 Spektrum

Spektrum is a multi-platform music genre classificator and music recommandation system developed in the context of the course "Application Challenges for Machine Learning on the example of IBM Power AI", by the team consisting of Marte Vinje, Moritz Klimmek, Thomas Salzer, Aaron Hümmecke, Lukas Vorwerk [9] . The application consists of two parts:

- Music Genre Classification: Performed by a Convolutional Neural Network Model which analyzes the MEL spectogram of a given input song.
- Music Recomandation: Generating suggestions by making use of a combination collaborative filtering and content based filtering.

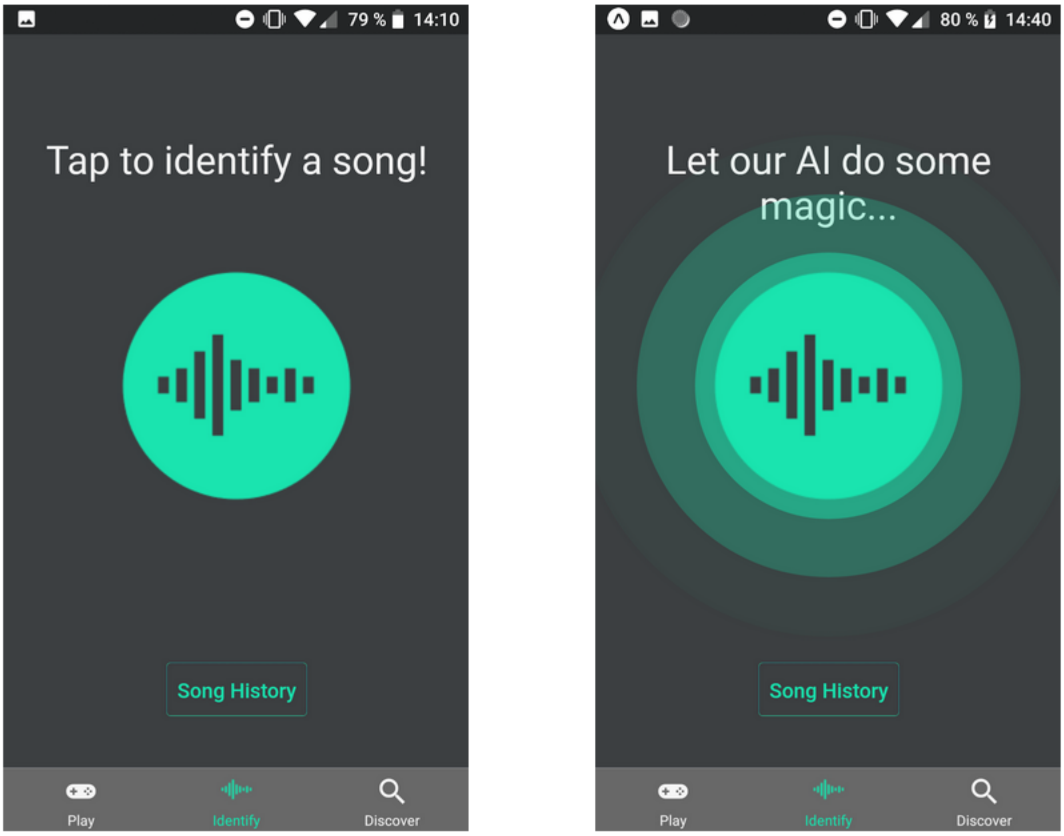

Figure 1.1: The interface of the Spektrum application

1.2 MuseNet

MuseNet is a Deep Neural Network Framework developed by OpenAI that can generate 4-minute snippets of original musical composition with up to 10 different instruments, combining various styles ranging from Mozart to The Beatles. MuseNet creates music by determining the patterns over a given style as input, generating the respective sequence of notes and chords. It also computes a relational graph between the its various current styles, in order to incorporate as many related musical features as possible.

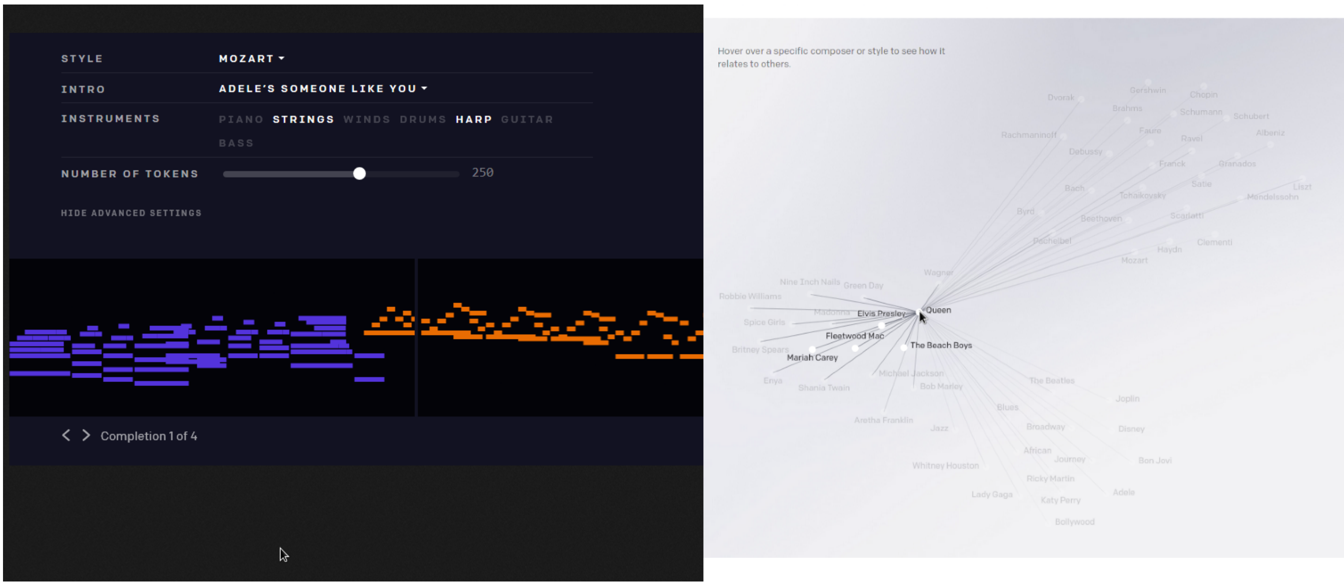

Figure 1.2: An illustration of the MuseNet's functionalities

1.3 Magenta

Magenta is an open source project (started by a group of engineers from the Google Brain team), based on deep learning and reinforcement learning used for generating and creating art. Based on Google's Tensorflow as well and Magenta.js, the produce original songs, images, drawings and material textures.

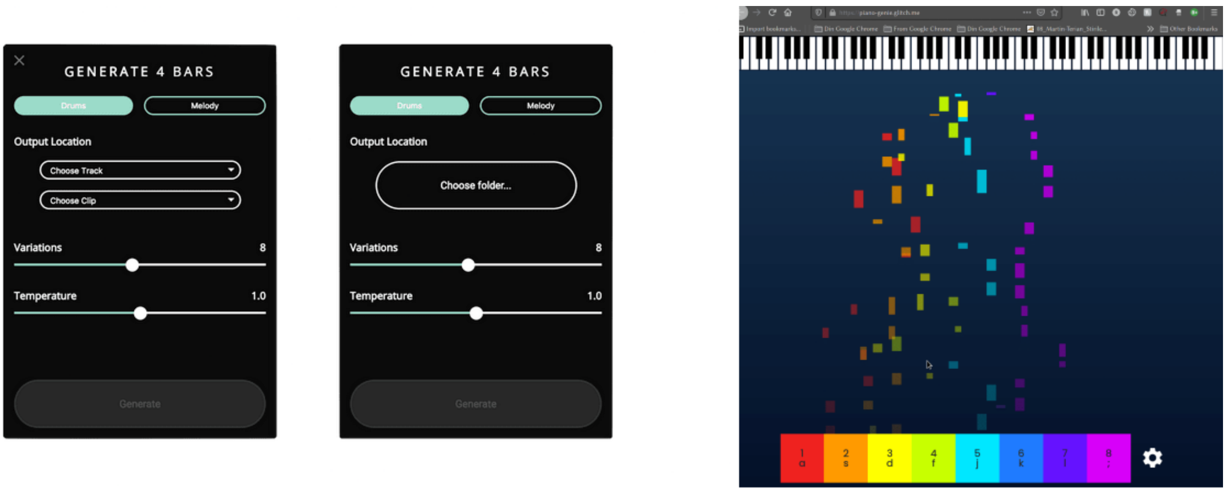

Figure 1.3: Applications based on Magenta

Chapter 2

Technologies used

Python provides many modules for numeric calculus, data processing and manipulation as well as tools for user interaction. In the following chapter we will go through the modules and libraries used for achieving the goal of the application, instrument classification.

2.1 Scipy

SciPy is a Python-based ecosystem of open-source software for mathematics, science, and engineering. The SciPy library contains a collection of numerical algorithms and domain-specific toolboxes, including signal processing, optimization, statistics, and much more.[3] The aforemetioned library's submodule *Input and output* provides a variety of ways of reading and writing data in numerous file formats. For the data processing part of the application, we had choosen the *scipy.io.wavfile.read* function because it facilitates the data manipulation process by being compatible with the numpy module, mentioned below (the output of the *scipy.io.wavfile.read* function is a tuple consisting of the sample rate(int) and the audio data(as a *ndarray*).

2.2 NumPy

Numpy is a Python library (part of the SciPy ecosystem) that provides the needed tools in order to perform the mathematical operations behind the Neural Network Framework. The core of the module is the *ndarray* object, which encapsulates an optimized n-dimensional array. Thus, we use *ndarray* objects for computing and storing the various operations performed throughout the application (e.g. the dot product, the data reshaping). The *ndarray* array is represented as a matrix, whose dimensions are referred as *shape*. Some of the advantages brought by the numpy module are:

- Forward and backward operation compatibility for the various objects contained withing the application
- Fast execution time caused by the optimization of the library [2] (by using precompiled C code)
- Support for large number/big data processing.

2.3 TensorFlow

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.[4] From the vast amount of methods and submodules contained within TensorFlow, we used Keras, an utilitary library built on top of the TensorFlow API. In order for the Neural Network Framework to be able to handle large amounts of data, we inherited in our custom *DataGenerator class* the *tensorflow.keras.utils.Sequence* class. The *Keras DataGenerator class* along with the custom Neural Network Framework *DataGenerator* provide among other functionalities the *getitem* method, which yields a batch of data at a given index, thus overcoming the big data handling problem.

2.4 Kapre

Kapre (Keras Audio Preprocessors) is an audio data preprocessing library built on top of *Keras*, specialized on audio signal handling.[1] Kapre facilitates sound transformation by efficiently performing various operations (e.g. *Short Time Fourier Transformation*, *Magnitude*) in a GPU optimized consistent manner . For the input data modeling we had chosen the *get mel spectogram layer* function from the module *kapre.composed*. It applies sequentially *STFT, Magnitude and mel filterbank* transformations over the given input data, returning a mel spectogram with an user specified sample rate and output shape.

2.5 Librosa

Librosa is a python package for music and audio analysis. It provides the building blocks necessary to create music information retrieval systems. [7] Out of the Librosa API we have choosen the following tools :

- *resample*: A function that resamples an audio file from its original rate to an user given sample rate.
- *to mono*: A function which converts a multi channel signal to mono signaling by averaging the sample values.

2.6 youtube-dl

Youtube-dl is an open source tool for downloading videos and songs from the YouTube platform. In the context of the final application, the youtube-dl module provides the end user with the possibilty of classifying a song just by inserting a link to a YouTube video.

2.7 Tkinkter

Tkinkter is the standard Python GUI Toolkit, a simple yet versatile module was used for the minimalistic Graphic User Interface of the final application.

Chapter 3

Architecture and implementation

Amet venenatis urna cursus eget. Quam vulputate dignissim suspendisse in est ante. Proin nibh nisl condimentum id. Egestas maecenas pharetra convallis posuere morbi. Risus viverra adipiscing at in. Vulputate eu scelerisque felis imperdiet. Cras adipiscing enim eu turpis egestas pretium aenean pharetra. In aliquam sem fringilla ut morbi tincidunt augue. Montes nascetur ridiculus mus mauris. Viverra accumsan in nisl nisi scelerisque eu ultrices vitae. In nibh mauris cursus mattis molestie a iaculis. Interdum consectetur libero id faucibus nisl tincidunt eget. Gravida in fermentum et sollicitudin ac orci. Suscipit adipiscing bibendum est ultricies. Etiam non quam lacus suspendisse. Leo urna molestie at elementum eu facilisis sed odio morbi. Egestas congue quisque egestas diam in arcu cursus. Amet consectetur adipiscing elit ut aliquam purus.

3.1 Titlul secțiunii 1

Eros donec ac odio tempor. Facilisi morbi tempus iaculis urna id volutpat. Faucibus in ornare quam viverra orci sagittis eu. Amet tellus cras adipiscing enim eu turpis egestas. Integer feugiat scelerisque varius morbi. Platea dictumst vestibulum rhoncus est pellentesque elit ullamcorper dignissim. Bibendum arcu vitae elementum curabitur. Eu nisl nunc mi ipsum faucibus. Id aliquet lectus proin nibh nisl condimentum id venenatis a. Cras adipiscing enim eu turpis egestas pretium. Quisque non tellus orci ac auctor augue mauris augue. Malesuada pellentesque elit eget gravida cum. Ut lectus arcu bibendum at. Massa id neque aliquam vestibulum morbi blandit. Posuere ac ut consequat semper viverra nam. Viverra adipiscing at in tellus integer feugiat scelerisque varius morbi. Morbi enim nunc faucibus a pellentesque sit amet porttitor eget. Eu feugiat pretium nibh ipsum consequat nisl vel. Nisl purus in mollis nunc sed.

3.2 Titlul secțiunii 2

Elementum sagittis vitae et leo duis ut diam quam nulla. Purus sit amet volutpat consequat mauris nunc. Tincidunt augue interdum velit euismod in pellentesque massa. Nunc sed augue lacus viverra vitae congue. Porttitor leo a diam sollicitudin. Faucibus pulvinar elementum integer enim. Adipiscing bibendum est ultricies integer quis auctor elit. Blandit aliquam etiam erat velit scelerisque in. A iaculis at erat pellentesque adipiscing commodo elit at. Erat nam at lectus urna duis. Consequat ac felis donec et. Fermentum posuere urna nec tincidunt praesent semper feugiat nibh sed. Proin gravida hendrerit lectus a. Pretium viverra suspendisse potenti nullam ac tortor vitae purus. Arcu cursus euismod quis viverra nibh cras pulvinar mattis. Gravida arcu ac tortor dignissim convallis aenean. Quam nulla porttitor massa id neque aliquam vestibulum morbi. Sed viverra ipsum nunc aliquet. Quis enim lobortis scelerisque fermentum dui faucibus in.

Chapter 4

Use cases

Amet venenatis urna cursus eget. Quam vulputate dignissim suspendisse in est ante. Proin nibh nisl condimentum id. Egestas maecenas pharetra convallis posuere morbi. Risus viverra adipiscing at in. Vulputate eu scelerisque felis imperdiet. Cras adipiscing enim eu turpis egestas pretium aenean pharetra. In aliquam sem fringilla ut morbi tincidunt augue. Montes nascetur ridiculus mus mauris. Viverra accumsan in nisl nisi scelerisque eu ultrices vitae. In nibh mauris cursus mattis molestie a iaculis. Interdum consectetur libero id faucibus nisl tincidunt eget. Gravida in fermentum et sollicitudin ac orci. Suscipit adipiscing bibendum est ultricies. Etiam non quam lacus suspendisse. Leo urna molestie at elementum eu facilisis sed odio morbi. Egestas congue quisque egestas diam in arcu cursus. Amet consectetur adipiscing elit ut aliquam purus.

4.1 Titlul secțiunii 1

Eros donec ac odio tempor. Facilisi morbi tempus iaculis urna id volutpat. Faucibus in ornare quam viverra orci sagittis eu. Amet tellus cras adipiscing enim eu turpis egestas. Integer feugiat scelerisque varius morbi. Platea dictumst vestibulum rhoncus est pellentesque elit ullamcorper dignissim. Bibendum arcu vitae elementum curabitur. Eu nisl nunc mi ipsum faucibus. Id aliquet lectus proin nibh nisl condimentum id venenatis a. Cras adipiscing enim eu turpis egestas pretium. Quisque non tellus orci ac auctor augue mauris augue. Malesuada pellentesque elit eget gravida cum. Ut lectus arcu bibendum at. Massa id neque aliquam vestibulum morbi blandit. Posuere ac ut consequat semper viverra nam. Viverra adipiscing at in tellus integer feugiat scelerisque varius morbi. Morbi enim nunc faucibus a pellentesque sit amet porttitor eget. Eu feugiat pretium nibh ipsum consequat nisl vel. Nisl purus in mollis nunc sed.

Concluzii

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Nunc mattis enim ut tellus elementum sagittis vitae et. Placerat in egestas erat imperdiet sed euismod. Urna id volutpat lacus laoreet non curabitur gravida. Blandit turpis cursus in hac habitasse platea. Eget nunc lobortis mattis aliquam faucibus. Est pellentesque elit ullamcorper dignissim cras tincidunt lobortis feugiat. Viverra maecenas accumsan lacus vel facilisis volutpat est. Non odio euismod lacinia at quis risus sed vulputate odio. Consequat ac felis donec et odio pellentesque diam volutpat commodo. Etiam sit amet nisl purus in. Tortor condimentum lacinia quis vel eros donec. Phasellus egestas tellus rutrum tellus pellentesque eu tincidunt. Aliquam id diam maecenas ultricies mi eget mauris pharetra. Enim eu turpis egestas pretium.

Bibliography

[1] Keunwoo Choi, Deokjin Joo, and Juho Kim. "Kapre: On-GPU Audio Preprocessing Layers for a Quick Implementation of Deep Neural Network Models with Keras". In: *Machine Learning for Music Discovery Workshop at 34th International Conference on Machine Learning*. ICML. 2017.

[2] The SciPy community. "Numpy Official Documentation". In: (2008-2021).

[3] The SciPy community. "SciPy Official Documentation". In: (2008-2021).

[4] The Tensorflow community. "Tensorflow Official Documentation". In: ().

[5] Larry Hardesty. "Explained: Neural networks". In: (2017).

[6] Harrison Kinsley and Daniel Kukieła. *Neural Networks from Scratch in Python*. 2020.

[7] Brian McFee et al. "librosa: Audio and music signal analysis in python". In: *Proceedings of the 14th python in science conference*. Vol. 8. 2015.

[8] Narayan Srinivasan. "Building an Audio Classifier using Deep Neural Networks". In: (2017).

[9] Marte Vinje et al. "Music Genre Detection – A Complete System". In: (2020).

[10] Mingqing Yun and Jing Bi. "DEEP LEARNING FOR MUSICAL INSTRUMENT RECOGNITION". In: (2017).