Comparison of the Use of the DEMUCS Neural Network On Different Platforms for the Separation of Sources Of Musical Origin

Raul Pérez Alarcón
Software Engineering Department
Universidad Nacional Mayor de San MarcosUNMSM
Lima, Peru
mailto:raul.perez4@unmsm.edu.pe

Luis Marcelo Pacheco Alvaro
Software Engineering Department
Universidad Nacional Mayor de San MarcosUNMSM
Lima, Peru
luis.pacheco8@unmsm.edu.pe

Ciro Rodriguez
Software Engineering Department
Universidad Nacional Mayor de San MarcosUNMSM
Lima, Peru
crodriguezro@unmsm.edu.pe

Favio Guevara Puente
Department of Electronic Engineering and Informatics
Universidad Nacional Federico Villarreal UNFV
Lima, Peru
2018008787@unfv.edu.pe

Ivan Petrlik
Systems Engineering Department
Universidad Nacional Federico Villarreal UNFV
Lima, Peru
ipetrlik@unfv.edu.pe

Yuri Pomachagua
Systems Engineering Department
Universidad Nacional Federico Villarreal UNFV
Lima, Peru
ypomachaguab@unfv.edu.pe

***Summary* — This paper makes a comparison between 3 systems deployed on different platforms (Web, Desktop, Mobile) which implement the DEMUCS neural network, responsible for separating sources of musical origin. The objective of this work is to determine on which platform the neural network can be executed more quickly for the use of the average user and from this to propose an optimal architecture for standard development. For this purpose, we selected 12 songs to be separated in the systems of the 3 platforms mentioned and we measured the time it takes for each system to execute the required separation and thus choose the best platform as a starting point. The results and conclusions of the work support the reason for choosing the platform, from which the development architecture was proposed.**

Keywords: Neural Network, Audio, Upmixing, Machine Learning, Deep Learning, DEMUCS

I. Introduction

The separation and investigation of sound sources or resources of musical origin have been a subject of research in recent years [1]. The development, improvement or invention of computational models is encouraged even in international competitions such as The Music Demixing (MDX) (Defossez, 2021) [2]. However, these advances are usually applied in fields such as telecommunications, pedagogy and professional development in the musical field Being able to separate and be able to individually hear the musical instruments [3] and voice coming from an original song (separation of sources of musical origin) has different purposes, be it for activities such as sheet music transcription [4](Miron, Janer, et al. 2017) , making musical arrangements, learning the section of an instrument in a song, among other uses [5] (Kadandaleet al.2020).

In recent years, various artificial intelligence models have been developed for this purpose, such as Demucs [2] (Défossez , Usunier , et al. 2021) , Spleeter [9] [10](Hennequin, Khlif, et al. 2020), DOA (Muñoz, Drossos, et al. 2020) among others.

Now these models, so that they can be brought to daily use, need to be implemented in a software system, which can be deployed on platforms such as web, desktop or mobile at the developer's choice. (Dua , Yadav , et al 2020).

However, not all systems that implement these neural networks run at the same speed, as they are conditioned to the type of platform for which the system has been developed. (Cucumber, Bender, 2018)

The main objective of this research is to find in which system the neural network can be executed in the most efficient way possible, with the shortest waiting time.

From there, to give an architecture proposal that allows the development of a system that executes the neural network as efficiently as possible. For this, we will take the DEMUCS neural network as a base and we will test it in software that has already been implemented and is freely accessible.

II. Materials and Methods

For the present work we will use three systems deployed on three corresponding platforms: Desktop Computer, Mobile Phone and Web Application. We will choose 12 songs to separate using the DEMUCS neural network algorithm that has been implemented in the 3 platforms described above. The programs that contain the DEMUCS audio separation neural network are:

- Desktop: “ Demucs UI”
- Mobile : “ Moises.Ai ”
- Web Application: “ MelodyMl ”

To run the “ Demucs UI” system we use a Ryzen 5 2600 desktop PC with 16 Gb RAM as we can see in the Figure 1. To run the “ Moises.Ai ” system we use an LG k9 cell phone. And for MelodyMl tests a stable 100 mbps broadband connection .

Accepted manuscript. Published in IEEE CICN 2022, pp. 185-188. https://doi.org/10.1109/CICN56167.2022.10008289

© 2022 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.



Fig. 1. PC with 16 Gb RAM. Source: self made

A. *Interface A*

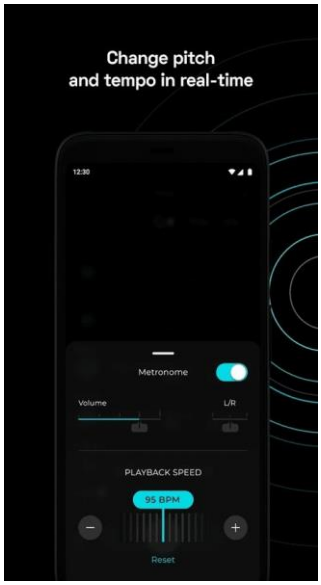


Fig. 2. Registration of the mobile application. Source: self made

We can see the Fig. 2. of the registration of the mobile application, highlighting its simplicity.

As we can see in Fig. 3., the main interface of login or entry to the application would have a standard format of most applications since users are more accustomed to it and will feel comfortable and not safe when using the application.

B. *Interface B*

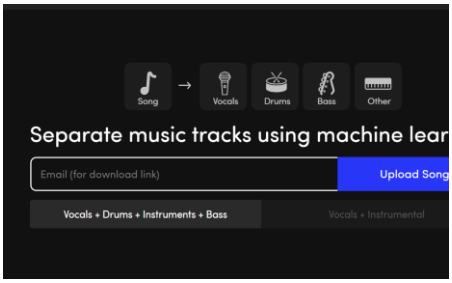


Fig. 3. Web version of the tool.

The second interface shows the web version of the tool, where the simplicity can be seen, however it has slowness problems.

C. *Interface C*

The C interface shows the free access developed program, we select the available hardware.

- Song ID =1 : Amaranth – Nightwish - 3:51
- Song ID =2 : Just Awake – FALILV - 3:38
- Song ID =3 : Ringing Bloom – Roselia - 5:34
- Song ID =4 : Inferno – Mrs. Green Apple - 3:32
- Song ID =5 : I Was Made for Lovin ' You – Kiss - 7:54
- Song ID =6 : Bad – Michael Jackson - 4:16
- Song ID =7 : Livi'n on a Prayer – Bon Jovi - 4:10
- Song ID =8 : In Between – The cure - 3:12
- Song ID =9 : More than Words – Extreme - 5:37
- Song ID =10: Roxane – The Police - 3:11
- Song ID =11: Boys Don't Cry – The Cure - 2:41
- Song ID=12: Heart of Glass – Blondie- 4:10

The results of the separation in the different platforms were analyzed for these 12 songs.

TABLE I. MEASUREMENT OF TIME FOR MOISES.AI

ID	Duration	Moises.Ai (Mobile App)
1	3 min 51 sec	12 min 45 sec
2	3 min 38 sec	10 min 10 sec
3	5 min 34 sec	12min 58 sec
4	3 min 32 sec	10min 29 sec
5	7 min 54 sec	16 min 54 sec
6	4 min 16 sec	12 min 50 sec
7	4 min 10 sec	12 min 46 sec
8	3 min 12 sec	10min 21 sec
9	5 min 37 sec	13 min 10 sec
10	3 min 11 sec	10min 21 sec
11	2 min 41 sec	9min 2 sec
12	4 min 10 sec	12 min 49 sec

TABLE II. TIME MEASUREMENT FOR VOCAL REMOVER

ID	Duration	Vocal Remover (Web App)
1	3 min 51 sec	14 min 28 sec
2	3 min 38 sec	15 min 53 sec
3	5 min 34 sec	15 min 47 sec
4	3 min 32 sec	15 min 30 sec
5	7 min 54 sec	18 min 33 sec
6	4 min 16 sec	13 min 50 sec

ID	Duration	Vocal Remover (Web App)
7	4 min 10 sec	13 min 46 sec
8	3 min 12 sec	14 min 30 sec
9	5 min 37 sec	14 min 21 sec
10	3 min 11 sec	13 min 53sec
11	2 min 41 sec	11 min 30 sec
12	4 min 10 sec	13 min 49 sec

TABLE III. TIME MEASUREMENT FOR DEMUCS UI

ID	Duration	Demucs UI System (Desktop System)
1	3 min 51 sec	8 min 30 sec
2	3 min 38 sec	7 min 26 sec
3	5 min 34 sec	8 min 02 sec
4	3 min 32 sec	7 min 49 sec
5	7 min 54 sec	12 min 03 sec
6	4 min 16 sec	9 min 43 sec
7	4 min 10 sec	8 min 47 sec
8	3 min 12 sec	6 min 56 sec
9	5 min 37 sec	9 min 43 sec
10	3 min 11 sec	6 min 49 sec
11	2 min 41 sec	6 min 28 sec
12	4 min 10 sec	8 min 23 sec

TABLE IV. TIME MEASUREMENT FOR DEMUCS UI

ID	Duration	Moises.Ai (Mobile App)	Vocal Remover (Web App)	DEMUCS UI System (Desktop System)
1	3 min 51 sec	12 min 45 sec	14 min 28 sec	8 min 30 sec
2	3 min 38 sec	10 min 10 sec	15 min 53 sec	7 min 26 sec
3	5 min 34 sec	12min 58 sec	15 min 47 sec	8 min 02 sec
4	3 min 32 sec	10min 29 sec	15 min 30 sec	7 min 49 sec
5	7 min 54 sec	16 min 54 sec	18 min 33 sec	12 min 03 sec
6	4 min 16 sec	12 min 50 sec	13 min 50 sec	9 min 43 sec
7	4 min 10 sec	12 min 46 sec	13 min 46 sec	8 min 47 sec
8	3 min 12 sec	10min 21 sec	14 min 30 sec	6 min 56 sec
9	5 min 37 sec	13 min 10 sec	14 min 21 sec	9 min 43 sec
10	3 min 11 sec	10min 21 sec	13 min 53sec	6 min 49 sec
11	2 min 41 sec	9min 2 sec	11 min 30 sec	6 min 28 sec
12	4 min 10 sec	12 min 49 sec	13 min 49 sec	8 min 23 sec

III. RESULTS AND PROPOSAL

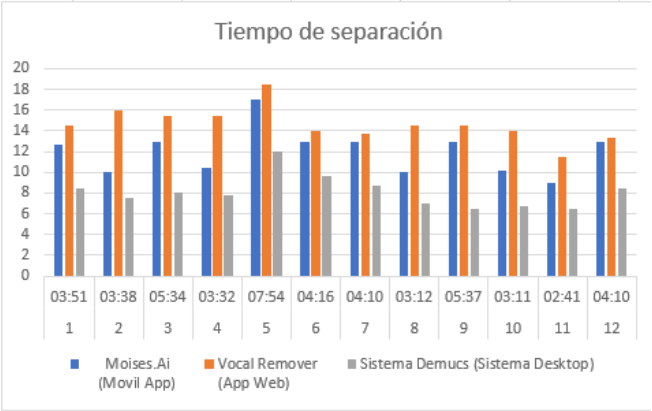


Fig. 4. Comparative bar graph. Source: self made

TABLE V. COMPARATIVE TABLE WITH AVERAGE

ID	Duration	Moises.Ai (Mobile App)	Vocal Remover (Web App)	Demucs UI System (Desktop System)
1	03:51	12.7	14.	8.5
2	03:38	10.06	16	7.5
3	05:34	13	15.45	8
4	03:32	10.5	15.5	7.75
5	07:54	17	18.5	12
6	04:16	13	14	9.7
7	04:10	13	13.75	8.74
8	03:12	10.08	14.15	7
9	05:37	13	14.5	6.48
10	03:11	10.12	14	6.7
11	02:41	9	11.5	6.5
12	04:10	13	13.3	8.5
Average		12.0383	14.6250	8.1142

It is evident that the different platforms that help us to break down or separate different musical rhythms do not have the same performance, so when carrying out the tests and subsequent comparisons we were able to rescue many data that made us realize the differences in the platforms:

The environment where the tests were best performed were the desktop ones as shown in tables I,II,II,IV andV, which makes us think that in general, the tool has many more advantages if it is run from a personal computer.

We found that when performing the aforementioned tests, the desktop tool is 80.24 % better than the web option, and a hefty 48.36 % better than the mobile version. For the design of an architecture that can support our solution, it is proposed to use this desktop version tool, since the execution times are lower and they are much more compatible in the company's equipment.

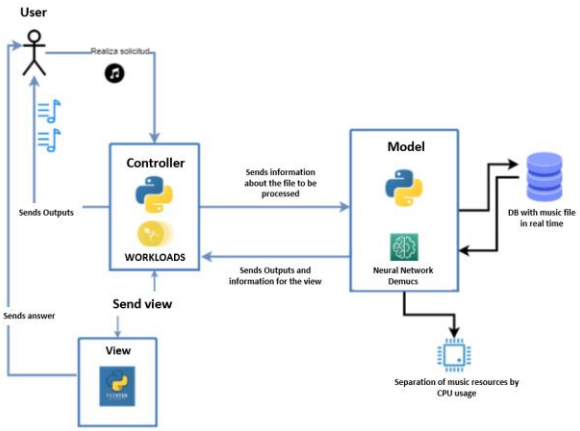


Fig. 5. MVC architecture model for the proposed Desktop system.

IV. CONCLUSIONS

After comparing and measuring the times of 12 different songs that have a high rhythmic content to be able to execute it in the tools, we propose that the desktop version be used for the design of the architecture destined for this purpose since it provides the best playing time execution, better ease of deployment and great compatibility. We can be self-sufficient in the execution of controls and data processing, since the result does not have any obstacles or delays, unlike the web version, you have to wait a certain number of minutes for the information to arrive by mail, while the mobile version takes more time because it is limited to the hardware of Smartphones, which is usually less than that of desktop computers.

It is taken as the best option for the implementation of the desktop environment architecture, due to the ease of deployment and execution in local environments.

REFERENCES

[1] Dua, Yadav, et al.. An Improved RNN-LSTM based Novel Approach for Sheet MusicGeneration. From Computer Science, 2020.

[2] Défossez, Usunier, et al. Music Source Separation in the Waveform Domain. arXiv preprint arXiv:1911.13254, 2021.

[3] Munoz, Drossos, et al. Multichannel Singing Voice Separation by DeepNeural Network Informed DOA Constrained CMNMF. 2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP), 2020.

[4] Miron, Janer,2017. Monaural score-informed source separation for classical music using convolutional neural networks. 18th International Society for Music Information Retrieval Conference, 2017.

[5] Kadandale, Montesinos, et al. Multi-channel U-Net for Music Source Separation 2020 IEEE 22nd International Workshop on Multimedia Signal, 2020.

[6] Cucumber, Bender. Separation of music sources using convolutional neural networks with multiple decoders. Conference on Audio, Acoustics and Sound. UNTREF, 2018.

[7] Défossez, Usunier, et al. Music Source Separation in the Waveform Domain. arXiv preprint arXiv:1911.13254, 2021.

[8] Alexandre Defossez. Hybrid Spectrogram and Waveform Source Separation. arXiv preprint arXiv:2111.03600, 2021.

[9] Hennequin, Khlif, et al. Spleeter: a fast and efficient music source separation tool with pre-trained models. Journal of Open Source Software, 2020, vol. 5, not 50, 2020.

[10] Rodriguez, C.; Angeles, D.; Chafloque, R.; Kaseng, F. and B. Pandey, "Deep Learning Audio Spectrograms Processing to the Early COVID-19 Detection," 2020 12th International Conference on Computational Intelligence and Communication Networks (CICN), Bhimtal, India, 2020, pp. 429-434, doi: 10.1109/CICN49253.2020.9242583.