Télecom SudParis
Cassiopée Project - Group 97

PRANTOS
An optimal music recognition system

Supervisor: Stephane Maag
Students: Luiz Fernando Ferro, Matheus Ramos Parracho, Gabriel Gomes Siqueira

June 2023

Contents

1. Introduction

The learning process of musicians involves a lot of commitment, and proficiency in reading music sheets is one of the most important competences for them. However, mastering this skill can be challenging and complex. PRANTOS is a system where the user can input a music sheet picture, which is then converted and returned as the corresponding audio. In addition, the system provides more features than a simple image conversion. It also allows the users to playback the converted audio at different speeds by adjusting the tempo, making it easier to learn complex compositions. This tool, designed on a simple and interactive platform, enables users to establish a better association of the aspects of the sheet and their corresponding sound, improving the studies and the ability of reading music sheets.

2. State of the Art

There are several studies and technologies in the area of music information retrieval that aim to address the problem of music sheet recognition and audio conversion. In the domain of optical character recognition (OCR), one technology is highlighted: Optimal Music Recognition (OMR).

2.1. Optimal Music Recognition

The Optimal Music Recognition (OMR) system uses complex image processing methods and machine learning techniques to computationally read music notation and extract the symbols from the music sheets. This problem was firstly approached in the 1960s, despite the limitations from that time. In 2012, this problem was separated into four different stages: preprocessing, music symbols recognition, musical notation reconstruction and final representation construction [1]. With the development of machine learning, all of these stages could be implemented by using deep learning techniques [2], which is the current stage of OMR solutions.

3. Methodology

The solution proposed with PRANTOS is an application that converts the music sheet into audio and allows the user to change the tempo of the output. To develop this project, the technical aspects were divided into three different segments that are described in this section: conversor model, web and mobile application development. Furthermore, section 3.4 describes the tasks assignments and management for the group members, giving a detailed description of the progress during the project development.

The general architecture of PRANTOS is synthesized in Figure 1. The user inputs the desired sheet image in the front-end interface (either in web or mobile application), which is then forwarded to the back-end server. The server firstly runs the OMR model that outputs the sheet representation in GUIDO format, a computer music notation format that is readable both for computer and humans [3]. After this process, the GUIDO file is then processed in a conversor that returns the corresponding content in MIDI format, which is used to retrieve the audio in WAV format. Once the server has the GUIDO and the audio files, it sends back to the front-end application, where both are displayed for the user. In the interface, it is possible to read the GUIDO notation and preview/download the generated audio.

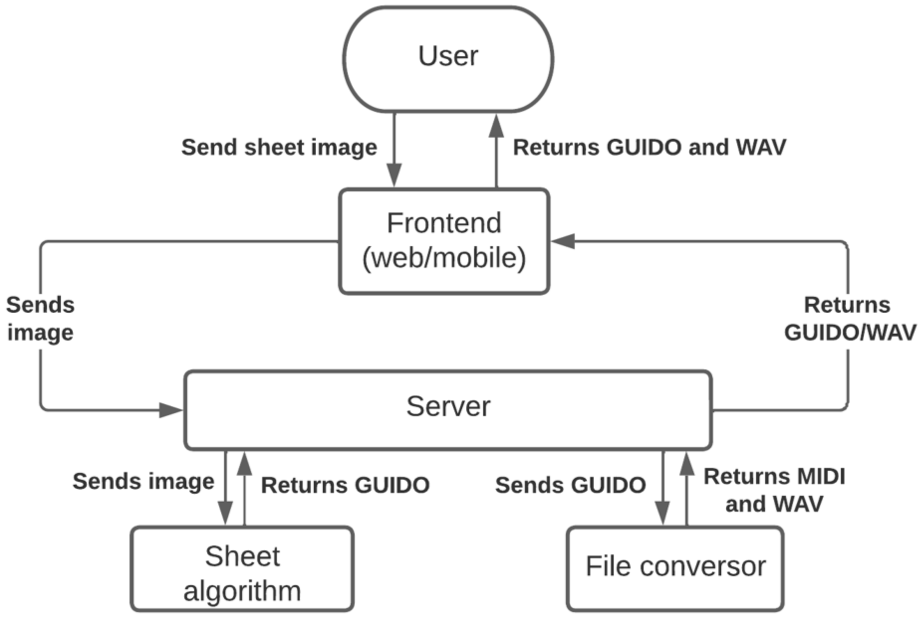

Figure 1. System architecture.

3.1. Conversor model

The conversor model is the most important technical aspect of PRANTOS and it is divided into two different blocks: the image conversion to GUIDO; the second block is the conversor that receives the GUIDO file and outputs a MIDI format, which is used to generate the audio output in .WAV format.

3.1.1. Sheet image conversion to GUIDO format

The optical music recognition algorithm is composed of five different steps. Firstly, a noise filtering and binarization process is applied in the image in order to achieve a better quality and accuracy in the conversion. Then, the image is cropped so each staff (set of five horizontal lines) can be interpreted at once. This cropping stage allows the system to input images with more than one staff, therefore, bigger compositions. The following step consists in identifying and removing the staff lines, leaving only the elements associated with the sheet (the position and length of notes, clef, accidentals, time signature, etc.). Furthermore, the fifth step consists in constructing new staff lines in the image with the goal to facilitate the recognition of the element's positions. The reason why the staffs are firstly removed to be then added again is because the reconstructed version has a bigger number of horizontal lines (7) than the original one (5), which makes it easier to recognize the position of the notes in the sheet. Finally, this image is processed to detect and recognize the symbols of the music sheet. The system identifies the clef, time signature, notes and uses the bold double bar lines in the end of the sheet to identify the end of the detection.

Once the system detects all the elements that are part of the sheet, they are processed and written in a machine-readable format called GUIDO. This type of format translates a note in the format "Note Pitch / Length", i.e. c1/4 would stand for a quarter C1. This way, the sheet image is transcripted to a format that can be processed and transformed into an audio.

3.1.2. GUIDO conversion to MIDI format

To achieve the conversion from GUIDO to the MIDI format, we used a conversor developed by the SALIERI Project, which is an academic project to study algorithmic aspects in music. The conversor was developed in the 1990s, and it was distributed as a Linux or Windows 32 bits Executable, an OS/2 Executable or the C/C++ files to compile manually.

The package has the executable "gmn2midi" that runs the conversor and uses the following syntax to run: "gmn2midi [options] <filename>.gmn". Also, it has a file named "gmn2midi.cfg" to adjust some settings to the MIDI file, such as the parameters of dynamic intensity or multiplication factors for the music.

We tested and observed that the Windows executable was compatible with the newest version of the OS. On the other hand, regarding the Linux OS, there were some difficulties when running a 32 bits binary on a new processor. And as the server is running on a Linux OS, the solution was to implement the conversion in the Wine: a compatibility layer that translates the Windows API calls into their corresponding Linux equivalents, allowing the Windows application to run directly on the Linux system. As soon as the model finishes the translation from the image to the GUIDO file, the server manages a call to the Wine API to run the Windows binary and then generate the MIDI file.

After obtaining the MIDI file, we also make a conversion to WAV based on the BPM the user sets in the browser. This conversion is based on a Python library that implements a software called "FluidSynth" to convert the MIDI notes into audio signals. Finally, the tempo is then applied based on the rate the user sends, which was achieved by using a Python library called "librosa" to compress or extend the WAV audio file. This way, the user receives the audio file.

3.2. Web interface development (Frontend)

The web application is the main environment to use PRANTOS. For the Frontend (Web interface), we developed an application in React, a Javascript Framwork, that enables the user to use the platform in a simple and interactive way. The web page contains clear instructions of how to use the system by indicating where to upload the image and how to choose the tempo marking (in bpm). Once the image is uploaded, a preview is displayed on the page. Then, once the user chooses the tempo runs the model, the web application returns the converted audio and also displays the GUIDO format. This way, the user can study the sheet by associating it with the GUIDO notation and the audio. If desired, the user is also capable of downloading these two files generated by the conversor. Figure 2 shows a screenshot of the web application.

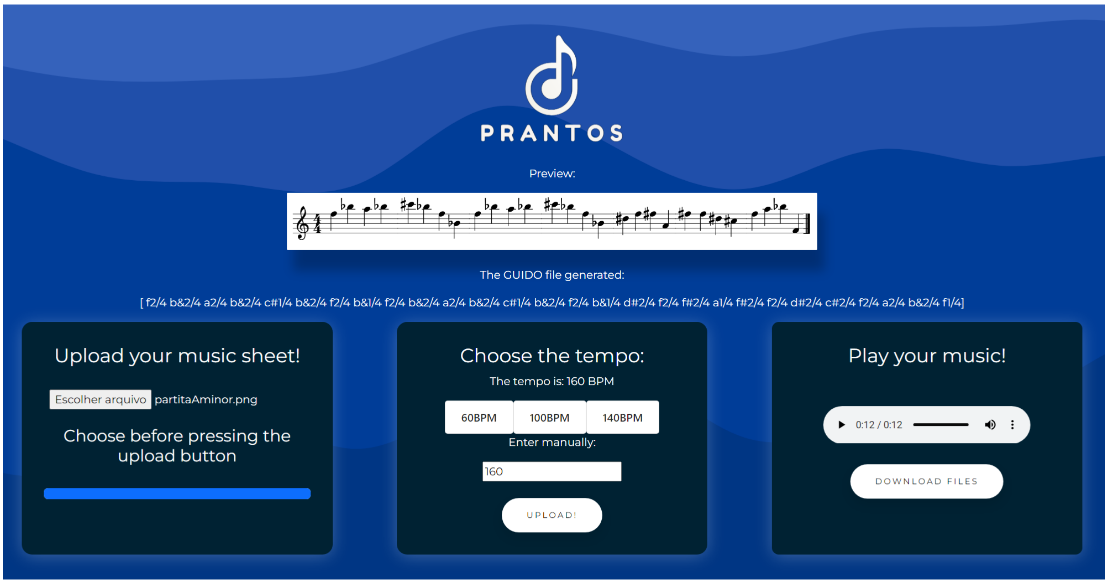

Figure 2. Web application after running the conversor.

The web application consists of two different pages that can be selected by a navigation bar. One of them corresponds to the application shown in Figure 2, while the other is an informative page explaining the purpose of PRANTOS.

3.3. Mobile application development

The mobile application is an alternative for using PRANTOS. The application was developed using React Native on the Expo Platform. Expo provides pre-made functionalities for React-Native Components and allows real-time debugging for Android, IOS as well as Web applications, Expo also allows the developer to easily deploy the application without having to create a *.apk* file or sending it to App or Google store.

As for the proper application, it offers functionalities such as two forms of uploading the image document: selecting photos from the user gallery or directly taking a photo of the music sheet. After uploading the image, the application calls the OMR (Optical Music Recognition) API that returns the audio file. This audio file is then displayed on the player screen that allows the user to listen to the generated music and stored if the user wants to listen to the generated audio file at another time. Figure 3 shows some pictures of the developed mobile interface.

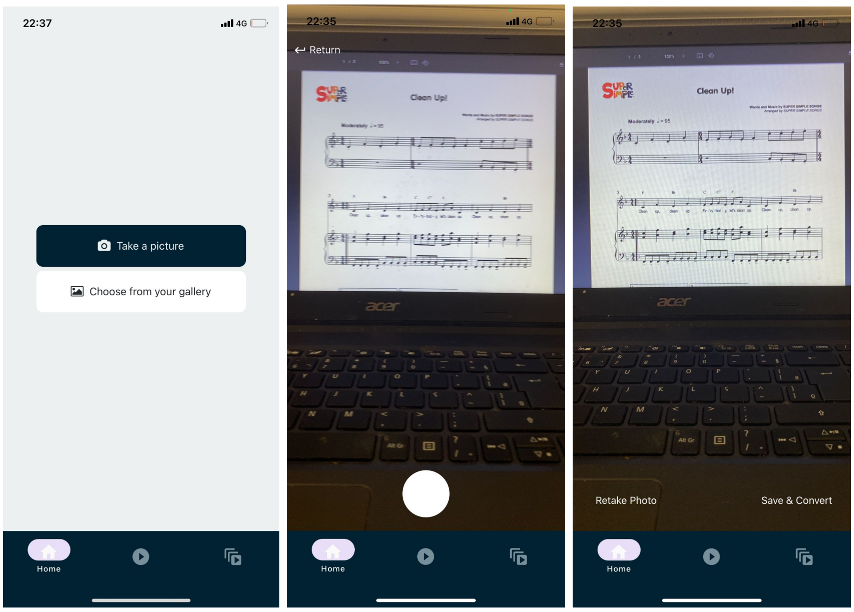

Figure 3. Mobile application interface.

3.4. Server (Backend)

The server was developed using Python (Anaconda) with a web framework called Flask. It allows us to run HTTP requests and manage them. With this, we were able to create the API and serve the Frontend. The Flask script is responsible for all the communication between the user and the OMR. All the errors are also managed by the server. To test the communication, we used Insomnia to make and edit the requests and manage all the protocol. The server is responsible for managing all the steps to convert the image and send the results to the user and it runs on a Linux machine.

3.5. Team management

To study and implement all the technical bricks used to build PRANTOS, the team was divided to work in each of the segments in parallel. The work assignments are defined in the Gantt chart shown in Figure 4.

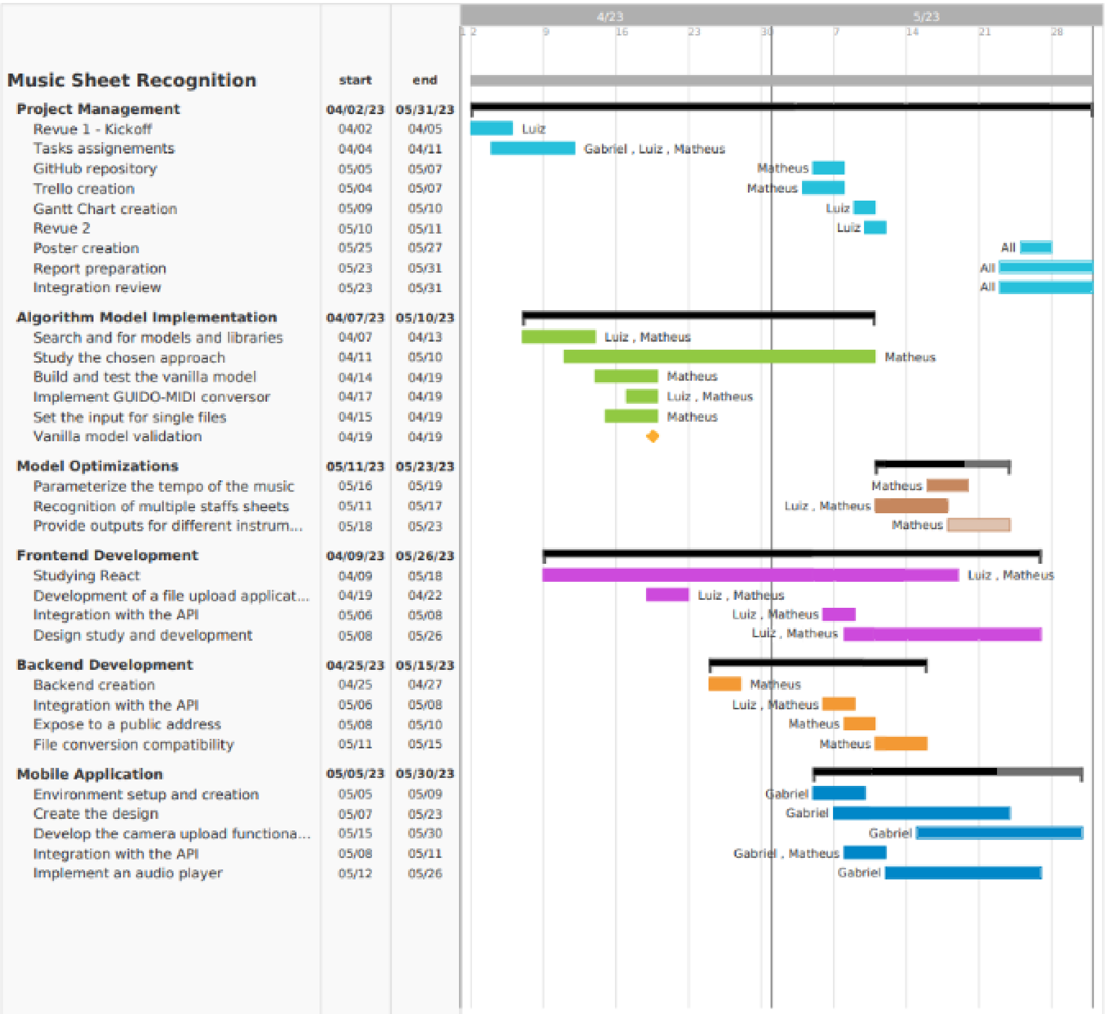

Figure 4. Gantt chart.

In the division proposed, two of the students focused on implementing the conversor model and developing the web application (front and back end), while the other developed the mobile application. The deadlines assigned for the tasks were based on a first assumption and were not strictly followed, despite being used as a guidance for the progress of our project. This division served as a model to organize our tasks in an appropriate time, in a way that eventual delays would not affect the schedule. Some of the tasks established above are considered as additional features of our project that we desired to develop, but it wouldn't affect the system performance in case of failure. For example, the task "Provide outputs for different instruments" could not be concluded, but it did not affected the development of the other parts. As some of the tasks required different levels of commitment, a more detailed discussion concerning the complexity and issues faced during the project is done in section 5.

In addition, we managed our project using the Trello platform (accessible via t.ly/PyJQ) and a GitHub repository (accessible via t.ly/VPKC), where it was possible to easily share and administrate all the pending tasks during the project development.

4. Results

With the development and integration of all the technical aspects discussed in the previous section, it was possible to create the PRANTOS platform. The user is capable of uploading a music sheet image, selecting the desired tempo for the audio output and running the conversion. The model returns the audio output for the given input parameters and it also provides the GUIDO format for the user as an additional way for the user to associate the audio with the music sheet.

For the evaluation of the audio obtained, we used different music sheets that explored the technical aspects and particularities of the notes, such as their lengths (half, quarter, eighth, etc.). Figure 5 shows the music sheet of a simple C major scale.

Figure 5. Major C scale sheet.

By using Figure 5 as input in the application, the system outputs the audio and the GUIDO format, which is transcripted below:

[c1/4 d1/4 e1/4 f1/4 g1/4 a1/4 b1/4 c2/4 d2/4 e2/4 f2/4 g2/4 a2/4 b2/4]

The sequence of the GUIDO format can be easily associated with Figure 5 as it represents a basic C major scale for two octaves, where the expected sequence of notes is CDEFGAB. The output specifies the note, the octave and the length of note (the quarter denoted by "/4") and it correctly translates the sheet notes into a computer readable format. Naturally, the GUIDO file is then processed and converted to the audio output.

As another example, Figure 6 uses the composition "Partita in A minor" by Johann Sebastian Bach.

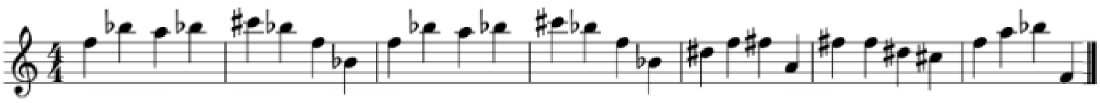

Figure 6. Partita in A minor, by Johann Sebastian Bach.

For this sheet, the application outputs the following GUIDO sequence:

[f2/4 b&2/4 a2/4 b&2/4 c#1/4 b&2/4 f2/4 b&1/4 f2/4 b&2/4 a2/4 b&2/4 c#1/4 b&2/4 f2/4 b&1/4 d#2/4 f2/4 f#2/4 a1/4 f#2/4 f2/4 d#2/4 c#2/4 f2/4 a2/4 b&2/4 f1/4]

When comparing the generated audio with the original music, it is noticeable that the system can convert the music sheet with a high accuracy. The only note that is incorrectly converted is *c#1/4*, when the expected value was *c#3/4*, two pitches higher. However, this variation does not represent necessarily a mistake, but a simple limitation of the system in dealing with a bigger range of octaves.

Other sheet examples were tested for the system and a satisfactory result was obtained for all of them. To evaluate and ensure the quality of the model, we tested the conversion for famous compositions (of which the corresponding audio is accessible) and also for randomly generated sheets on the internet [4].

5. Discussion

The development of PRANTOS required a deep study and comprehension of all the technical aspects presented in the methodology. This section will comment on the limitations of the project and the future searches that could improve the system.

As the major limitation of the project, the short amount of time to study and develop all the technical parts complicated the process of deploying and developing new features and improvements for PRANTOS. Despite having approximately two months to work on the project, it was possible to achieve a satisfactory OMR model with web and mobile applications.

Focusing on the system limitations, we can highlight the dependency of the model in detecting specific symbols, such as the clef and the double bold line that is used to detect the end of the sheet. As music sheets can be found with several representations, a different format that does not follow the model requirements may lead to a misinterpretation of the sheet and, therefore, an inaccurate conversion. This inflexibility of the inputs is unfortunately required to detect the elements of the sheet with accuracy and it could not be improved in time, but it is an interesting point for future searches.

Furthermore, the system is standardized to generate piano sound for the audio outputs. A desired feature of the project was to allow the user to choose the type of instrument the audio would be returned, but implementing this would require a deep and complex search on the GUIDO to MIDI conversion and also in the compatibility of the music sheets. Receiving a piano sheet as input and choosing the output of a guitar would demand to transpose the notes in octaves, which would imply a complex adjustment on the conversion model. Despite not being able to provide this feature, the output on piano sound still allows users to easily associate the sound with the sheet. It would be an interesting feature to improve PRANTOS, but it is not crucial for its purpose. Naturally, other similar improvements would be interesting such as playing multiple outputs with different instruments simultaneously, but were not the focus on the main goals of PRANTOS.

The web and mobile applications were developed in a simple way, but with a purpose of being interactive and friendly to the user. Future research in improving both applications would be interesting in the meaning of achieving a higher usability. The mobile application is a feature that demanded special attention to be successfully deployed, but it can still have its usability limited by the restrictions of the model in demanding good resolution and framed images. Therefore, improving the flexibility of the model in recognizing the symbols would also allow further improvements in both applications.

6. Conclusion

The PRANTOS project was successfully deployed as it accomplishes the purpose of providing an efficient tool that converts music sheet images in audio files (and GUIDO format as well). The interfaces developed for both web and mobile applications allow the user to have a friendly, easy and interactive experience with the system. Despite its limitations, it still provides a satisfactory solution to the problem of music sheet recognition. In conclusion, PRANTOS can increase the level of apprenticeship and short the path of mastering the skill of reading music sheets.

7. References

[1] Rebelo, A. et al. (2012). Optimal Music Recognition: state-of-the-art and open issues. *International Journal of Multimedia Information Retrieval 1:173-190.*

[2] Wel, E. & Ullrich, K. (2017). Optical Music Recognition with Convolutional Sequence-to-Sequence Models", *18th International Society for Music Information Retrieval Conference*. Suzhou, China.

[3] GUIDO Music Notation. *http://science.jkilian.de/salieri/GUIDO/index.html*

[4] Sheet Music Generator. *http://www.randomsheetmusic.com/*