Recognizing Printed Melodies using Object Detection and a Convolution Neural Network

Onur Kocahan and Friedrich Hartmann

March 17, 2021

1 Introduction

The goal of Optical Music Recognition (OMR) is to reproduce a sequence of document-written musical notation by a machine. Since the language of musical notation contains hundreds of symbols, this problems becomes really complex.
Our aim is to to treat only a fragment of the musical notation language, namely the fragment of American and Europe folk music. It has a huge restriction on the alphabet, since it does not contain chords, for example, and is more appropriate of our use case. Because we want to apply OMR to an application that photocopies a tune of a song-book and transforms it later to an audio file. This can be further used to learn how to sing this tune.
Available OMR Datasets such as DeepScores [Tuggener, 2021] contains around 30 000 000 sheets of written music with close to a hundred millions of small objects. This would really blow up our project. That's why we generate our own Dataset.
We further follow the general framework to OMR that contains according to Rabela et al. [2012] the following steps.

1. image pre-processing
2. recognition of the musical symbols;
3. reconstruction of the musical information in order to build a logical description of musical notation; and
4. construction of a musical notation model to be represented as a symbolic description of the musical sheet.

The most challenging task in this framework is point 2. An common approach to that is to use Neural Networks. A baseline for that is given by Pacha et al. [2018]. The authors applied Faster R-CNN, RetinaNet and U-Net on various datasets. The performance on DeepScores, which is not hand-written and therefore easier be trained, is not promising as the mean average precisions of the intersection over union ratios are 19.6%, 9.8%, and 24.8%, respectively. An other approach by Metaj and Magnolfi [2021] is more promising, but has really good results around 95% only for a selection where the bounding boxes have at least an intersection over union ratio of 0.50. That's why we invent a new approach on the object detection that is not based on a neural net.
We further apply a Convolutional Nueronal Network to classify the outputs of our objects detection mechanism which are then be used to generate an audio file.
Implementation details can be read up in our GitHub project `https://github.com/lutacluny/Sheet-Music-Recognition`.

2 Musical Notation Background

This section is about giving the reader a short summary of musical notation.
There exists various styles of writing music, but we focus only on the modern staff notation which is characterized by a staff line.
The position of the note head determines its name. It can be on the line, between two lines and above or

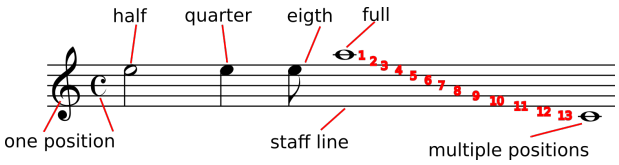

Figure 1: Musical notation simplified

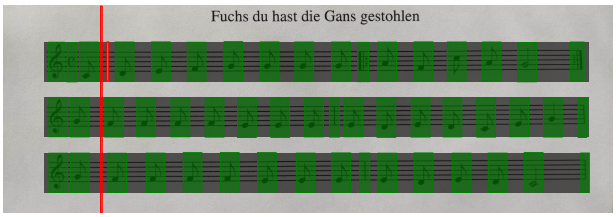

Figure 2: Abstracting a tune

under the staff using ledger lines. Therefore, limiting the ledger lines to one upper and one lower, a line can hold 13 different notes.
The value of a note is specified by its shape and defines its duration which is given relatively to the beat. That means that the note value *quarter* has the length of a quarter beat.
A visualization of the explanations above is given in Figure 1. Other musical symbols do not depend on its vertical position at the line.

3 Training Data Sampling

We build a highly flexible framework to generate a database that contains image files which can be used to train a Neuronal Net. It is build of musical symbols extracted as vector graphics by the help of the tool abc2mps S. [2021].
As one generated image file per musical symbol is not sufficient for training, we introduced several parameters for data augmentation. These are output dimension, scaling, rotation, horizontal shift, vertical shift and the respective number of outputs in the ranges defined by these parameters. It is for example possible to get 5 output files with respect to the scaling in the range 80% to 120%.
Our database supports at the moment up to 107 different musical symbols. Therefore, the number of training examples is $107 \cdot N$, where N is the number of augmented files per symbol.

4 Object Detection

In this section we describe our approach on the object detection. As the input is a single image file, the general task is to split this image into several images, such that each of the resulting images contains one and only one musical symbol which can recognized by a machine. The first step is to transform the image from RBG into black and white.
Our idea is further based on the observation that tunes of American and European folk music, written in the modern staff notation, contain recurrent patterns which is illustrated by Figure 2.
As it can be observed, each line, colored black, has the same height and width. Furthermore, the distance between two lines remains the same. That gives the possibility to identify first the most upper line and then calculate the position of the other lines. This can be achieved by representing the image as a matrix and identifying a column, that contains only staff lines. Because such a column has a unique pattern on the distances between the lines, which can be easily calculated on the matrix. The red colored line illustrates this fact in Figure 2.
After segmentation of the lines we split each line into symbols or groups of symbols, until each of the resulting segments contains one and only one symbol. To accomplish this, we take advantage of the fact that a column of the respective matrix representation associated with a line, has a significantly bigger amount of black pixels. The reference value of a column without a note is calculated identifying a column that matches the orange line in Figure 2. This is performed the same way as for the red marked column.

The success of this procedures depends on photo quality. It is mandatory that the staff lines are parallel to the edges of the photocopy. But in practice, this not really an issue, since most cameras have a grid implemented.
Furthermore, there are two hyper-parameters that

percentage of symbol contained in the image	< 50	< 90	< 100	100
	1	11	65	487

Figure 3: For example, it can be observed that on 54 images 10% of the symbol is not visible anymore.

needs to be chosen appropriated. Like we explained before, what matters is the amount of black pixels per column. So, there needs to be a thresh hold that separates columns containing musical symbols from those containing only staff lines. This is given as ϵ_{black} in per cent. The second one ,δ, is the separate width, given as a fraction of the images width. Only columns marked as containing a note, for which its distance exceeds the separate width, are split.
It turned out that $\epsilon_{black} = 60\%$ and $\delta = 1/120$ in the first run and $\epsilon_{black} = 140\%$ and $\delta = 1/10$ in the second shows desirable results. Our test images are split into 543 different images. It holds further that each new image contains one and only one musical symbol. The quality is shown in Figure 3.

It has to be finally admitted that choosing the right values for ϵ_{black} and δ is challenging. Finding a promising strategy on that, which performs good apart form our test suite, is not part of this project.

5 Image Classification

foo

6 Post Processing

The final step is to transform the output labels into an audio file. This is performed by first converting the output labels into a format that can be processed by PySynth g4brielvs [2021] which is the tool that generates the audio file.

References

g4brielvs. PySynth. `https://tomita.readthedocs.io/en/latest/`, March 2021.

Stiven Metaj and Federico Magnolfi. MNR_MUSCIMA_Notes_Recognition. `https://github.com/StivenMetaj/MNR_MUSCIMA_Notes_Recognition`, March 2021.

Alexander Pacha, Jan jr. Hajič, and Jorge Calvo-Zaragoza. A baseline for general music object detection with deep learning. *Applied Sciences*, 2018.

Ana Rabela, Ichiro Fujinaga, Filipe Paszkiewicz, Andre R. S. Marcal, Carlos Guedes, and Jaime S. Cardoso. Optical music recognition: state-of-the-art and open issues. *International Journal of Multimedia Information Retrieval*, 2012.

Lee S. abs2mps. `https://github.com/leesavide/abcm2ps/issues`, March 2021.

Lukas Tuggener. DeepScores. `https://tuggeluk.github.io/deepscores/`, March 2021.