DeepScores and Deep Watershed Detection: current state and open issues

Ismail Elezi *
ZHAW Datalab
Winterthur, Switzerland
elez@zhaw.ch

Lukas Tuggener *
ZHAW Datalab
Winterthur, Switzerland
tugg@zhaw.ch

Marcello Pelillo
Ca' Foscari University
Venice, Italy
pelillo@unive.it

Thilo Stadelmann
ZHAW Datalab
Winterthur, Switzerland
stdm@zhaw.ch

***Abstract*— This paper gives an overview of our current Optical Music Recognition (OMR) research. We recently released the OMR dataset *DeepScores* as well as the object detection method *Deep Watershed Detector*. We are currently taking some additional steps to improve both of them. Here we summarize current and future efforts, aimed at improving usefulness on real-world task and tackling extreme class imbalance.**

I. INTRODUCTION

The accurate localization and classification of musical symbols is a key component in every functioning Optical Music Recognition (OMR) system [1]. In pursuit of our goal of advancing the state of the art in optical music symbol detection we have created the large *DeepScores* [2] dataset of synthetic music scores together with ground truth to enable the training of very deep neural networks. Additionally, we created a custom object detection method called Deep Watershed Detection [3], that is designed to work particularly well on optical music notation data. Both these contributions currently carry some drawbacks and flaws that hamper performance and usability. In this paper, we give an overview of our current as well as planned efforts to alleviate these issues.

II. UPDATES TO THE *DeepScores* DATASET

A. Shortcomings of the initial release

At its initial release, *DeepScores* had two main weaknesses: first, it was fully geared towards our application in conjunction with Audiveris; many common symbols that where not interesting in that context have been omitted, which severely limited the usability of *DeepScores* in other contexts. Second, *DeepScores* consist only of synthetically rendered music sheets, since labelling hundreds of thousands of music sheets by hand is prohibitively expensive. However, the common use case for OMR is scans or even photos of music sheets. This discrepancy can lead to severe performance drops between model training and actual use.

B. Enhanced character set

In an effort to make *DeepScores* more universally usable we created a new version—called *DeepScores-extended*—containing annotations for a far greater number of symbols. According to our knowledge and discussions with other members of the community, no crucial symbols are missing from the *DeepScores-extended* annotations. The full list of supported symbols is available online[1].

C. Richer musical information

While the interest of the authors lies in the detection of musical symbols, this task is not the full problem of OMR. The reconstruction of semantically valid music from detected symbols is at least as challenging as the detection. To enable research focused on reconstructing higher-level information, we have added additional information to the *DeepScores* annotations. Every labeled object now has an *onset* tag that tells the start beat of the the given object. All noteheads additionally have their relative position on the staff as well as their duration in their annotation (see Figure 1).

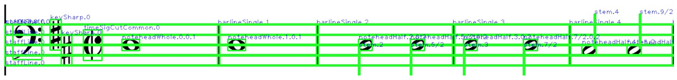

Fig. 1. Small piece of music notation with DeepScores-extended annotations overlayed. The naming is either classname.onset or classname.onset.relativecoordinate.duration, depending on availability.

D. Planned improvements

A drawback of the *DeepScores* dataset is that it is synthetic. We are currently working on a much smaller dataset, meant for transfer-learning, that consists of pages originally taken from *DeepScores* that are printed and then digitized again. Then, through a global centering and orientation alignment of the scan, the original annotations are made valid again for the scanned version. We use different printers, scanners, cell-phone cameras, and paper qualities to make the noise introduced by this process resemble the real world use case as much as possible. Naively training a Deep Watershed Detector on this new dataset, we observed that the detector was unable to find anything on the testing set despite that the loss function converged. This led us to believe that severe overfitting is going on, and we were able to get promising results by simply adding l2-regularization and performing more careful training (see Figure 2 for a qualitative result of the detector on the new dataset).

III. FURTHER RESEARCH ON DEEP WATERSHED DETECTION

A. Augmenting inputs

DeepScores, unlike many academic datasets, is extremely unbalanced. In fact, the most common class (notehead black)

*Equal contribution

[1] *tuggeluk.github.io/deepscores_syms_list*