2017 16th IEEE International Conference on Machine Learning and Applications

Towards Self-Learning Optical Music Recognition

Alexander Pacha, Horst Eidenberger
Interactive Media Systems, TU Wien, Vienna, Austria
alexander.pacha@tuwien.ac.at, horst.eidenberger@tuwien.ac.at

***Abstract*—Optical Music Recognition (OMR) is a branch of artificial intelligence that aims at automatically recognizing and understanding the content of music scores in images. Several approaches and systems have been proposed that try to solve this problem by using expert knowledge and specialized algorithms that tend to fail at generalization to a broader set of scores, imperfect image scans or data of different formatting. In this paper we propose a new approach to solve OMR by investigating how humans read music scores and by imitating that behavior with machine learning. To demonstrate the power of this approach, we conduct two experiments that teach a machine to distinguish entire music sheets from arbitrary content through frame-by-frame classification and distinguishing between 32 classes of handwritten music symbols which can be a basis for object detection. Both tasks can be performed at high rates of confidence (>98%) which is comparable to the performance of humans on the same task.**

I. Introduction

Music plays a central role in our cultural heritage with written music scores being an essential way of communicating the composer's intention to musicians that perform a piece of music. The music notation encodes the information into a graphical form that follows certain syntactic and semantic rules to encode pitch, rhythm, tempo, and articulation. Optical Music Recognition (OMR) tries to recognize and understand the notation and the contents of an image for a machine to be able to comprehend the music. Given a system that is able to translate an image into a machine-readable format, the applications are manifold, including preservation and digitization of hand-written manuscripts, supporting music education or accompanying musicians that practice their performance.

Although considerable research has been conducted and many systems have been developed [1] that reportedly perform well on the specific set of music scores for which they have been designed for, the robustness and extensibility of these systems is limited due to the underlying architecture and used algorithms that discard information and propagate errors from one step to the next, e.g. an error in the binarization which is often the first step of an OMR system might cause the symbol detection to detect notes where there are none. Many algorithms have been proposed to improve individual steps of this linear process, but to the best of our knowledge, there exists no system that is capable of automatically recognizing a large set of real-world data with satisfactory precision, good usability, and reasonably low editing costs [2] of errors that were introduced during the process. Many people could benefit from digitizing a large body of music scores that is accessible and searchable [3]. As a result, there are ongoing projects to do so including SIMSSA[1] and OpenScore[2]. To support such projects, we propose a new approach: rather than designing features and defining rules by hand, the system should learn to extract features and appropriate rules by itself (given a certain amount of supervision). Ideally, such a system is capable of transcribing music scores as accurately as humans.

II. Related work

OMR has been a subject of interest at least since 1966 [4], and received substantial attention by Bainbridge and Bell [5] who established a general framework for OMR that has been adopted by many researchers [1]. Since then, many researchers suggested entire OMR systems [6], [7] or proposed specialized algorithms for solving or improving sub-tasks such as binarization [8] or staff-line detection and removal [9], [10]. However, most of them use ad-hoc solutions based on expert knowledge that follow widely used practices that work best on datasets fulfilling certain prerequisites, e.g. detecting staff-lines with horizontal projections requires the scores to have straight staff-lines. Unfortunately, these systems tend to experience difficulties when confronted with images that deviate from the expected input format for which they were designed (e.g. if the staff-lines are curved due to the bonding of a textbook). Adding another preprocessing step or improving an algorithm can help to overcome one or the other limitation, but might not help a system to gain robustness beyond a certain level.

In the last few years, machine learning - and especially Deep Learning with Convolutional Neural Networks (CNNs) - received a lot of attention with results that surpass human-level performance on computer vision tasks such as image classification [11]. Wen et al. proposed a machine learning approach for symbol segmentation and symbol classification [12] in combination with a pre-defined ruleset. Calvo-Zaragoza et al. [13] classify music scores at pixel-level with CNNs into foreground, background, and staff-lines. Gallego et al. [14] use auto-encoders to remove staff lines and finally Pinheiro Pereira et al. [15] classify handwritten

[1]http://simssa.ca/, last visited on Oct. 4, 2017
[2]http://openscore.cc, last visited on Oct. 4, 2017

0-7695-6321-X/17/31.00 ©2017 IEEE
DOI 10.1109/ICMLA.2017.00-60

music symbols from the HOMUS database [16] into 32 different categories with a precision of over 96%. Together, they provide strong evidence, that machine learning can successfully be applied to develop new types of OMR systems that are robust and extensible to a wide range of scores.

III. How humans read scores

We believe that an OMR system should be able to read and comprehend music scores with all their facets as well as humans. To the best of our knowledge, there exists no system that would come close to human performance [1]. As far as it is understood today, humans process visual scenes in a hierarchical way at three levels [17, p. 557]:

1) Low-level, where contrast, orientation, color, and movement are processed, primarily in the retina and ganglion cells [17, p. 600]
2) Intermediate-level, where the layout of the scene is processed by parsing the visual image into contours and surfaces of objects, segregating them from the background, involving the primary visual cortex [17, p. 619].
3) High-level, where actual object identification is performed, by matching surfaces and contours to known shapes from our memory (or more precisely to their neuronal representation) which happens primarily in the Inferior Temporal Cortex [17, p. 622]

By processing visual information in this hierarchical way, humans become very good at arriving at scene descriptions, grasping the gist of a scene. But reading music scores includes not only the visual perception of objects, but also relating objects to each other and to the context, a process where, unfortunately, today little is known about how humans perform this task, apart from certain brain regions that have been identified to be involved in this process [18], [17, p. 1353]. Note that for relating elements to each other and interpreting them correctly, it appears that humans use all information available. For music scores, this includes the staff-lines as the reference system, knowledge about the type of music, the notational system and also prior knowledge such as the probabilities of continuations within idioms [18] to resolve ambiguities if the available information is incomplete or doubtful. The expectancy can even replace a stimulus, making up for misprints as shown in the Goldovsky experiment [18] indicating that reading involves both top-down (or conceptually-driven) and bottom-up (or data-driven) processes.

Learning from the way humans read scores, binarizing the image as a first step or removing staff-lines seems to be counterproductive as it discards potentially relevant information. In summary, we conclude that OMR systems could benefit from operating directly on the input image (which is possible with Deep Learning), providing feedback loops from later steps to refine earlier steps and consider information that might not have been used so far.

IV. How machines read scores

David Marr proposed a computational framework of vision that has three levels and to us appears very useful when discussing vision problems [19]:

- Computational theory, which specifies how a vision task can be solved in principle
- Algorithmic level, that gives precise details on how the theory can be implemented. In other words: What is the input and output and how to obtain the output given the input?
- Hardware for realizing the algorithm in a physical system (which is not necessarily computer hardware, but in our case it is)

Given this framework, we think that the computational theory of how humans or machines can read scores is correct and sound: detecting systems, staves and staff-lines and using them as structural guidance is a solid foundation; segmenting elements into smaller parts and constructing a relational mapping leads to a symbolic representation; finally, this symbolic representation can be interpreted in its context, according to syntactic and semantic rules that correspond to a particular notational language.

The algorithmic level, however, seems to be much harder to solve, possibly because the inherent complexity of the problem is often underestimated. Many proposed approaches can be seen as concept-driven because they use prior knowledge of the specific object, in this case, music sheets. We believe that a data-driven, Deep Learning approach is a viable alternative that should be investigated further. Therefore, we propose the following five questions as a model for bottom-up music processing that are specifically formulated to facilitate the development of such an Optical Music Recognition algorithm.

Can a machine mimic human behavior in ...

Q-I distinguishing between music scores and arbitrary content?

Q-II understanding the structure of music scores (staves, systems) and distinguish basic music symbols from each other and from the background?

Q-III detecting and locating music symbols (notes, rests, ornaments, accidentals, bar-lines, articulations, ...) in the scores?

Q-IV understanding the relation of objects to each other in music scores (the relation between a note and the staff-lines, an accidental to the left of a note which relates to that note, etc.)?

Q-V fully understanding the syntax and semantics of music scores (inferring the actual note from relative position, shape and preceding symbols such as key signatures or accidentals)?

These five questions define our research program for the data-driven investigation of the OMR problem using deep networks. In our opinion, each question can be solved using an appropriate model and sufficient data. Note that the questions are of increasing complexity with Q-V representing a complete system that is capable of reading scores and fully understanding their content like humans. Q-I and Q-II can be implemented by using CNNs that operate directly on the raw input data. A promising approach for Q-III is to extend a classifier into an object detector by using region proposal networks [20]. As for questions Q-IV and Q-V, Recurrent Neural Networks (RNN) seem to be a good fit [21], as they can learn relationships in sequential data and already achieved remarkable results in Optical Character Recognition [22], a task that is comparable to OMR but in many regards simpler [5].

V. Experimentation

To evaluate whether a data-driven approach is suitable for improving the state-of-the-art in OMR, two experiments were conducted that try to answer Q-I and partially Q-II. The first, to recognize music scores in an image and classify that image into one of the two categories: 'scores' or 'other'. The second, to classify isolated handwritten music symbols into 32 different classes, reproducing [15] in greater depth and improving their results significantly. For both experiments, a Convolutional Neural Network was trained using the popular Deep Learning frameworks Keras[3] and Tensorflow[4]. The resulting models can then be used for inference on almost any machine including mobile devices (see Figure 1) to classify images from the live camera-feed and display a frame-by-frame classification.

A. Datasets

The dataset used for training, validation, and testing in the first experiment contains over 5500 images of which 2000 images contain scores and 3500 images contain something else (see Table I). The largest portion was obtained by using two publicly available datasets: the MUSCIMA database, which contains 1000 handwritten music scores [23] and the training database of the Pascal VOC Challenge 2006 which contains over 2600 images [24] that were considered part of the ground-truth for the category 'other'. Additionally, we created a new dataset containing 2000 imperfect but realistic images, by taking 1000 images depicting music scores and 1000 images of text documents and other objects with a smartphone camera. Preliminary testing showed that text documents were likely to be confused with scores, especially if they contain tables. Hence, a large portion of the additional images contains such documents in order to enable the network to learn the distinction. The complexity of the scores ranges from simple childrens' tunes to modern orchestral scores, taken in various lighting conditions and from different angles.

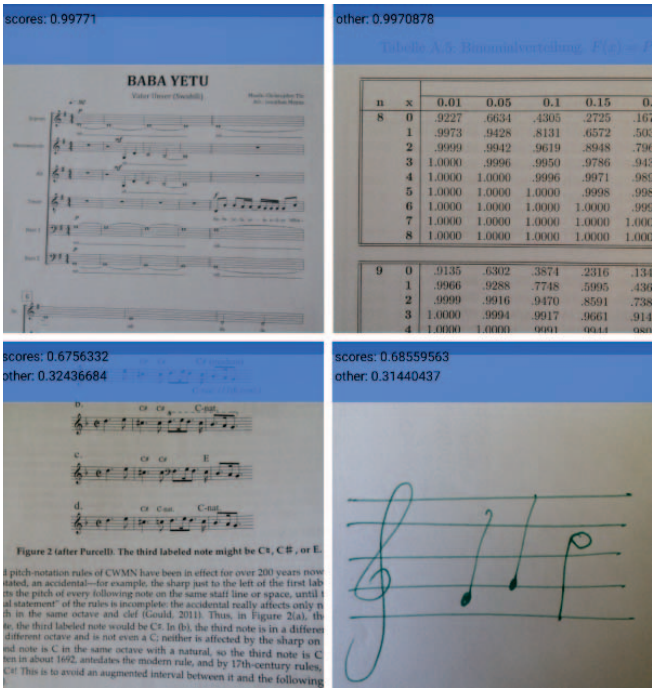


Figure 1: Screenshots of the Android application, classifying a sheet of music scores (left top) and a table with data (right top) with a certainty of 99%. When presented with images that contain scores and text (left bottom) or unusual forms (right bottom), certainty drops to approximately 70% but the system still classifies the image correctly.

The dataset for the second experiment is the Handwritten Online Musical Symbols (HOMUS) dataset [16] that contains 15200 samples of hand-written musical symbols, written by 100 different musicians[5].

B. Architecture and Training

For both experiments, various network architectures were evaluated, including a VGG-like architecture [25] and residual networks [26].

The first experiment attempts to answer Q-I and uses color-images that are non-uniformly resized to 128x128 pixels for the first trial and 256x256 pixels for the second. For the second experiment that is targeted towards Q-II, black and white images are generated from the textual representation of strokes by connecting the points of each stroke. Since individual symbols vary drastically in size, while CNNs expect a fixed-size image as input, the following two approaches were evaluated:

[3]http://keras.io/, last visited on Oct. 4, 2017

[4]http://www.tensorflow.org/, last visited on Oct. 4, 2017

[5]Note that the original dataset contained a few mistakes and artifacts that were reported to the authors and corrected before the training see https://github.com/apacha/Homus for details, last visited on Oct. 4, 2017

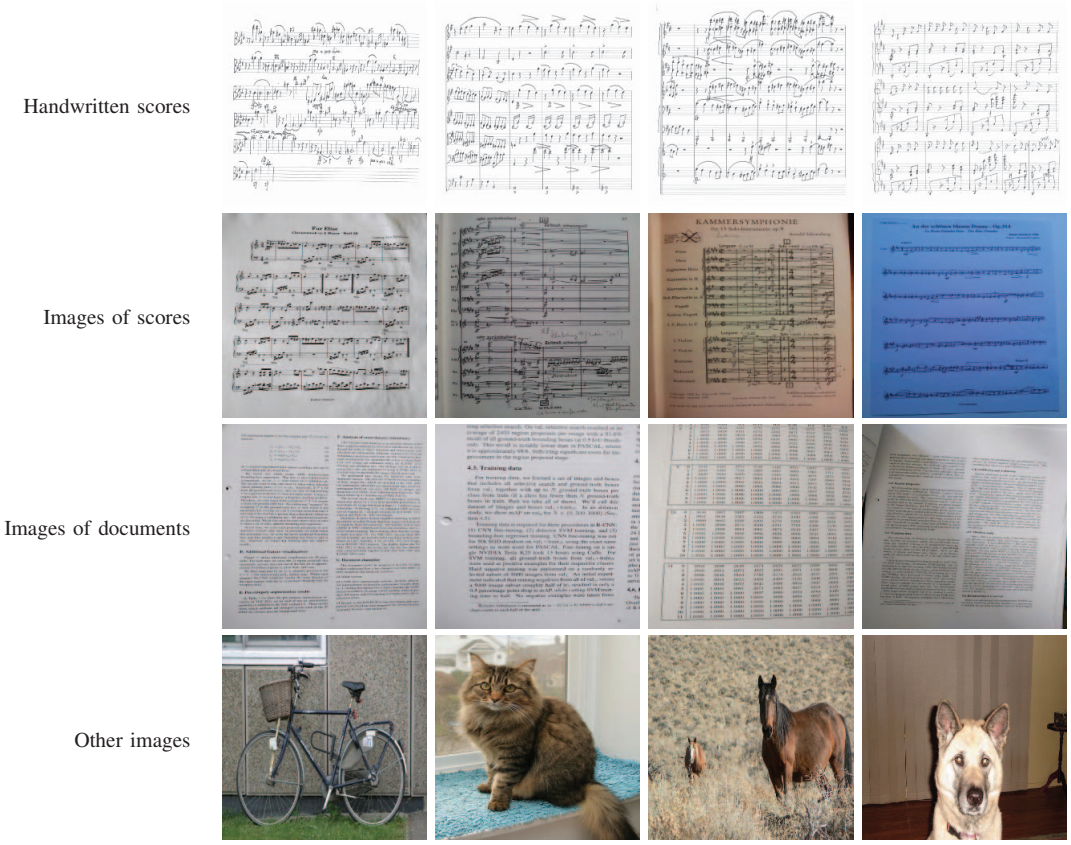


Table I: Sample images of the various categories, as they were shown to the classifier during training (non-uniformly resized). The upper two rows form the class 'scores' and the lower two rows the category 'other'.

1) Drawing the symbols in the center of a large enough canvas that fits most of them (e.g. 192x96 pixels, with only 23 out of 15200 symbols exceeding this size)
2) Drawing each symbol in a canvas that exactly fits its size and rescaling all symbols non-uniformly to a fixed size, e.g. 96x96 pixels

These particular sizes were empirically selected because they yielded the best results while allowing multiple down-scaling operations by a factor of two without interpolation.

Batch-normalization, early-stopping, weight-decay and dynamic learning-rate-reduction are used as regularization strategies to improve training speed and overall performance. Random-rotation by 10° and random-zoom of 20% are used as data-augmentation strategies to simulate the images being taken from slightly different points-of-view which leads to results that are robust to minor variations.

C. Evaluation

To evaluate each experiment, the respective dataset was split into three parts of which 80% are used as training data, 10% are used for validation during the training and for hyperparameter optimization and the final 10% are used for evaluating the performance of the trained model on previously unseen data.

To obtain a baseline, a subset of the images was also shown to a number of people that were asked to perform the same classification task in a desktop application on a computer screen. The application did not allow for zooming and the users classified the images using the keyboard but were allowed to go back and revise their decisions without any time constraints.

1) First Experiment: Typical training took 30 epochs before early stopping the training to prevent overfitting. The trained model classified 98.5% of the images in the test set correctly on the 128x128 pixels condition and 100% on the 256x256 condition, meaning that this task appears almost trivial to the machine.

The more than 500 images from the test set were also shown to three different users, who were asked to manually classify them either as 'something that displays music scores' or 'something else'. The images were down-scaled to the same 128x128 pixels that correspond to approximately 3.5cm on a desktop screen. In total, they classified over 1500 images with an average precision of 96.49%. The main

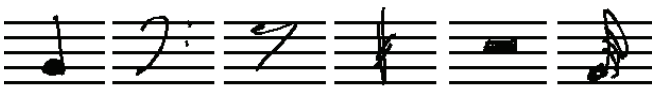


Figure 2: Superimposed staff-lines over isolated symbols to create meaningful context. Five parallel lines are drawn with an equal spacing of 14 pixels between each line [16]. From left to right: Quarter-Note, F-Clef, Eighth-Rest, Sharp, Whole-Half-Rest, Sixty-Four-Note

source of error was due to the very small images. Partially repeating the process with images of size 256x256 pixels, which corresponds to approximately 7cm on a desktop screen, showed that humans can perform this task without exceptional errors.

2) Second Experiment: The second experiment contains a wide range of conditions whose effects were investigated: image-size, stroke-thickness, superimposing staff-lines (see Figure 2) and of course the hyperparameters for the training of a deep neural network, including the network architecture, the used optimizer, and minibatch-size. A total of over 150 different hyperparameter-combinations were tested and documented. The following hyperparameters have empirically shown to work very well for this task:

- Monitoring the accuracy on the validation set after each epoch and reducing the learning-rate by a factor of 0.5 if it does not improve for 8 epochs. Similarly, the entire training was stopped if no improvement was observed for 20 epochs.
- Adam, Adadelta and Stochastic gradient descent (SGD) were evaluated as optimizers with Adadelta performing slightly better than Adam and much better than SGD.
- Evaluated minibatch-sizes included 16, 32 and 64 but the impact is rather small and in our opinion can be neglected.

The obtained results reach up to 98.02% accuracy on a test-set of 1520 images which is a significant improvement, compared to previously reported results of 97.26% [27] and 96.01% [15]. For images with undistorted symbols drawn on a fixed canvas (Section V-B, approach 1) a Res-Net architecture with 25 convolutional layers and about five million parameters performed best. Similar results were obtained with a VGG architecture for non-uniformly resized symbols (Section V-B, approach 2) that consists of 13 convolutional layers and about 8 million parameters.

The results of the best run, broken down by symbol class, are given in Table II and show that the network struggled most with notes and rests that are only discriminable by the number of flags, such as Thirty-Two- and Sixty-Four-Notes.

Five users were asked to perform the same task on a random sample of the dataset. In total, they classified 1520 images with an average precision of 95% and experiencing most difficulties in Quarter-Rests and Sixteenth-Rests that both have manifestations that deviate from their printed counterparts dramatically or are simply ambiguous (see Figure 3).

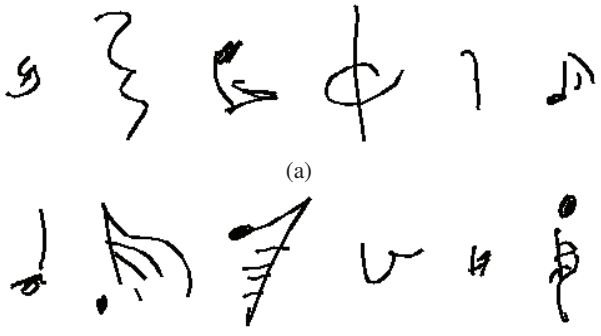
Table II: The recall and precision per class for the best trained residual network in comparison to human performance on the same task.

	Residual Network		Human test subjects	
Class name	**Recall**	**Precision**	**Recall**	**Precision**
12-8-Time	1.00	1.00	1.00	0.97
2-2-Time	1.00	1.00	0.95	1.00
2-4-Time	0.97	0.95	1.00	0.98
3-4-Time	0.95	1.00	1.00	0.97
3-8-Time	1.00	1.00	1.00	1.00
4-4-Time	1.00	0.98	0.97	1.00
6-8-Time	1.00	1.00	1.00	1.00
9-8-Time	1.00	1.00	1.00	1.00
Barline	1.00	0.98	0.97	0.92
C-Clef	1.00	1.00	1.00	0.91
Common-Time	1.00	1.00	0.97	1.00
Cut-Time	0.95	1.00	0.98	0.98
Dot	0.97	1.00	1.00	1.00
Double-Sharp	1.00	1.00	0.97	1.00
Eighth-Note	0.99	0.95	0.92	0.98
Eighth-Rest	1.00	1.00	0.98	0.86
F-Clef	1.00	1.00	0.97	0.92
Flat	0.97	1.00	0.95	0.95
G-Clef	1.00	0.95	0.98	0.98
Half-Note	1.00	1.00	0.97	0.94
Natural	0.95	1.00	0.74	1.00
Quarter-Note	1.00	1.00	0.93	0.95
Quarter-Rest	0.95	0.95	0.89	0.82
Sharp	1.00	1.00	1.00	0.97
Sixteenth-Note	0.94	0.95	0.90	0.92
Sixteenth-Rest	0.97	0.97	0.76	0.81
Sixty-Four-Note	0.96	0.95	0.94	0.94
Sixty-Four-Rest	0.97	0.97	0.83	0.97
Thirty-Two-Note	0.91	0.95	0.99	0.91
Thirty-Two-Rest	0.97	0.95	0.91	0.89
Whole-Half-Rest	1.00	0.98	1.00	1.00
Whole-Note	1.00	0.98	1.00	0.98

Another very interesting detail was observed: When superimposing staff-lines as depicted in Figure 2, test-accuracy remains at high rates of up to 97.03%, indicating that the network can learn to ignore them almost entirely, thus providing evidence that staff-line removal might be omitted in future systems, as discussed in Section III.

VI. Conclusion

Given the results presented in Section V-C we conclude that Q-I can be answered with yes, showing that humans and machines can achieve similar results on the given dataset. Detecting music scores and distinguishing them from arbitrary content is a relatively easy problem compared to the entire challenge of Optical Music Recognition but what experiment 1 shows, is that machines can learn something as abstract as the concept of 'what music scores look like' by just providing enough data and using a Deep Learning approach. As for Q-II, we showed that a CNN can be trained to distinguish handwritten music symbols from each other at high rates of confidence, even with staff-lines being present.



(a)

(b)

Figure 3: Examples of symbols from the test set that were misclassified by the machine (a) and by humans (b). Their intended classes from left top to right bottom: Sixteenth-Rest, 2-4-Time, Sixteenth-Note, Cut-Time, Quarter-Rest, Sixteenth-Note, Quarter-Note, Sixty-Four-Note, Sixty-Four-Rest, Quarter-Rest, Natural, and Sixty-Four-Note.

When combining these results with the work from [28] and [13] we conclude that Q-II can also be answered with yes.

VII. Future work

To promote collaboration and reproducibility, all datasets, the entire source-code and the raw data from both experiments have been released on Github at https://github.com/apacha/MusicScoreClassifier and https://github.com/apacha/MusicSymbolClassifier under a liberal MIT-license. We are confident, that by following the described path, an OMR system can be created that is capable of not only classifying entire images but also recognizing the structure of the document, reliably detecting objects in the image and even understanding the relation of elements to each other without formulating explicit rules by only training appropriate models on a comprehensive dataset.

References

[1] A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. R. Marcal, C. Guedes, and J. S. Cardoso, "Optical music recognition: state-of-the-art and open issues," *International Journal of Multimedia Information Retrieval*, vol. 1, no. 3, pp. 173–190, 2012.

[2] P. Bellini, I. Bruno, and P. Nesi, "Assessing optical music recognition tools," *Computer Music Journal*, vol. 31, no. 1, pp. 68–93, 2007.

[3] A. Laplante and I. Fujinaga, "Digitizing musical scores: Challenges and opportunities for libraries," in *Proceedings of the 3rd International workshop on Digital Libraries for Musicology*. ACM, 2016.

[4] A. Rebelo, G. Capela, and J. S. Cardoso, "Optical recognition of music symbols," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 13, no. 1, pp. 19–31, 2010.

[5] D. Bainbridge and T. Bell, "A music notation construction engine for optical music recognition," *Software: Practice and Experience*, vol. 33, no. 2, pp. 173–200, 2003.

[6] L. Pugin, J. Hockman, J. A. Burgoyne, and I. Fujinaga, "Gamera versus Aruspix – two optical music recognition approaches," in *ISMIR 2008–Session 3C–OMR, Alignment and Annotation*, 2008.

[7] Y.-S. Chen, F.-S. Chen, and C.-H. Teng, "An optical music recognition system for skew or inverted musical scores," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 27, no. 07, 2013.

[8] Q. N. Vo, S. H. Kim, H. J. Yang, and G. Lee, "An MRF model for binarization of music scores with complex background," *Pattern Recognition Letters*, vol. 69, pp. 88 – 95, 2016.

[9] C. Dalitz, M. Droettboom, B. Pranzas, and I. Fujinaga, "A comparative study of staff removal algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 5, May 2008.

[10] J. dos Santos Cardoso, A. Capela, A. Rebelo, C. Guedes, and J. P. da Costa, "Staff detection with stable paths," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 6, June 2009.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[12] C. Wen, A. Rebelo, J. Zhang, and J. Cardoso, "A new optical music recognition system based on combined neural network," *Pattern Recognition Letters*, vol. 58, pp. 1 – 7, 2015.

[13] J. Calvo-Zaragoza, G. Vigliensoni, and I. Fujinaga, "Document analysis for music scores via machine learning," in *Proceedings of the 3rd International workshop on Digital Libraries for Musicology*. ACM, 2016, pp. 37–40.

[14] A.-J. Gallego and J. Calvo-Zaragoza, "Staff-line removal with selectional auto-encoders," *Expert Systems with Applications*, vol. 89, 2017.

[15] R. M. Pinheiro Pereira, C. E. Matos, G. Braz Junior, J. a. D. de Almeida, and A. C. de Paiva, "A deep approach for handwritten musical symbols recognition," in *Proceedings of the 22Nd Brazilian Symposium on Multimedia and the Web*, ser. Webmedia '16. New York, NY, USA: ACM, 2016, pp. 191–194.

[16] J. Calvo-Zaragoza and J. Oncina, "Recognition of pen-based music notation: The HOMUS dataset," in *2014 22nd International Conference on Pattern Recognition*, Aug 2014, pp. 3038–3043.

[17] E. R. Kandel, J. H. Schwartz, T. M. Jessell, S. A. Siegelbaum, and A. J. Hudspeth, *Principles of neural science*. McGraw-hill New York, 2012, vol. 5.

[18] J. Sloboda, *Exploring the musical mind*. Oxford University Press, 2005.

[19] J. P. Frisby and J. V. Stone, *Seeing, Second Edition: The Computational Approach to Biological Vision*, 2nd ed. The MIT Press, 2010.

[20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015.

[21] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 99, 2016.

[22] C.-Y. Lee and S. Osindero, "Recursive recurrent nets with attention modeling for OCR in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[23] A. Fornés, A. Dutta, A. Gordo, and J. Lladós, "CVC-MUSCIMA: a ground truth of handwritten music score images for writer identification and staff removal," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 15, no. 3, pp. 243–251, 2012.

[24] M. Everingham, A. Zisserman, C. K. I. Williams, and L. Van Gool, "The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results," http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf, 2006.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[27] J. Calvo-Zaragoza, A.-J. Gallego, and A. Pertusa, "Recognition of handwritten music symbols with convolutional neural codes," *Proceedings of the 14th IAPR International Conference on Document Analysis and Recognition*, 2017.

[28] J. Calvo-Zaragoza, A. Pertusa, and J. Oncina, "Staff-line detection and removal using a convolutional neural network," *Machine Vision and Applications*, pp. 1–10, 2017.