Convolutional Neural Networks and Ensemble Methods to Identify Musical Elements in Optical Music Recognition

Jenaro Augusto Barbosa[1], **Edimilson Batista dos Santos**[1]

[1]Federal University of Sao Joao del-Rei – UFSJ
Department of Computer Science – DCOMP
Sao Joao del-Rei, MG, Brazil

jenaro@ufsj.edu.br, edimilson.santos@ufsj.edu.br

Abstract. *Optical Music Recognition (OMR) is an important tool to recognize a scanned page of music sheet automatically, which has been applied to preserving music scores. In this paper, we present a comparative study among a Convolutional Neural Network (CNN) architecture, named CREATES, and Ensemble Learning methods, such as Random Forest and XGBoost, to classify musical symbols. The initial results show that CREATES is promising in this task and outperforms ensemble methods on the HOMUS dataset. However, CNN require more computing power.*

1. Introduction

A significant amount of musical works produced in the past are still available only as original manuscripts or as photocopies on date. In several historical cities, it is possible to find important music collections dated from the beginning of 18th century, which are in this state. Thus, some works, such as Copista system [de Paulo et al. 2015], have proposed to apply computer science techniques to musical scores recognition through a process named Optical Music Recognition (OMR).

Optical Music Recognition (OMR) is a Computer Science field applied to music that deals with problems like recognition of handwritten scores. The applications in OMR are similar to Optical Character Recognition (OCR) tools. However, it is not a straightforward extension from the OCR, since the problems to be faced are substantially different.

The OMR is needed for the preservation of musical works, which requires digitalization and should be transformed into a machine-readable format. An OMR program should thus be able to recognize the musical content and make semantic analysis of each musical symbol of a musical work. Generally, such a task is challenging because it requires the integration of techniques from some quite different areas, i.e., computer vision, artificial intelligence, machine learning, and music theory. In spite of existing applications that converts handwriting scores into editable scores, most of these applications a) do not work with manuscript scores [Bainbridge and Bell 2001], b) are very expensive and c) are not open source. All these reasons encourage building a brand-new tool on the OMR.

The development of an OMR system can be divided into some distinct parts, e.g.: the image acquisition, image preprocessing and digital image recovery, the recognition of musical symbols with computer vision and machine learning, the music notation reconstruction and the symbolic music output. In this paper, we focus on the step of recognition of musical symbols through machine learning techniques. Specifically, we propose to apply more robust classification algorithms to identify such symbols, such as Convolutional Neural Networks (CNN) [Dumoulin and Visin 2016] and Ensemble Learning methods, which combines the results of multiple classifiers to produce improved results. The initial idea is to compare these techniques, mainly exploring the characteristics of CNN that can obtain good results without the application of many data pre-processing methods.

In [Barbosa and Santos 2021], the authors presented a CNN architecture named **CREATES** - **C**onvolutional neu**R**al n**E**twork **A**pplied **T**o id**E**ntification mu**S**ical - which outperforms other state-of-the-art classifiers. However, the time required to train the CNN was much higher as well. We intend to verify if the ensembles methods can get results as good as CNN in a shorter time. For this, CREATES was applied to the HOMUS dataset, as well as two ensemble learning methods: Random Forest [Breiman 2001] and XGBoost [Bruce and Bruce 2017], which have been applied in different domains.

The remainder of this paper is organized as follows. Section 3 brings some related work. In Section 4, Convolutional Neural Networks (CNN) and Ensemble Learning methods, including XGBoost and Random Forest, are presented. Section 5 explains the proposed methodology. Besides, the experiments and analysis of results are presented. Finally, Section 6 brings the concluding remarks and points out some future work.

2. Background

2.1. Optical Music Recognition

Optical music recognition (OMR) is an area of research that has been under development for over 50 years to enable computers to read and recognize musical notation in documents, and is commonly associated with Optical Character Recognition (OCR). However, OMR differs from OCR by considering the semantic part and other issues inherent to musical notation [Calvo-Zaragoza et al. 2020].

The development of OMR tools seeks to guarantee access to music and information to the most diverse audiences. It is a challenge due to the nature of the documents and the high level of care that must be used for handling them. It is increasingly understood the need to build tools capable of recognizing and digitizing these documents in software that can be easily manipulated or consulted, in addition to formats that can be interpreted in their entirety by computers.

The OMR is an area of easy interest to attract, in addition to having a good motivation for its application: the conservation and propagation of this data. There are many challenges in the OMR area, as explained by Calvo-Zaragoza et. al [Calvo-Zaragoza et al. 2020], who highlights that this area is still difficult to access for new researchers, especially those without significant musical training. There are few introductory materials available and, in addition, the field has struggled to define itself and build a shared terminology.

2.2. Convolutional Neural Networks

Convolutional neural networks (CNN) emerge with a focus on image recognition, taking inspiration from an important characteristic of human beings: their ability to find patterns. CNN seeks to use the special features present in the images, where a logic of importance is applied to certain points of the image, considering the characteristic or relevance of that point. Thus, a high accuracy rate can be obtained, maintaining a balance in relation to the training time, since it is not necessary to pre-process the images for the model (or simpler processes are applied).

CNN has a set of initial layers and hidden layers in the center, where the function of extracting the features is delegated. Two specific layers stand out: convolution and pooling, which perform the feature extraction process. Finally a set of dense layers, which perform the learning process.

A convolution operation can be interpreted as a similarity measure between two signals. For that, in each convolution performed on the image, a kernel K of the same dimension is applied. The convolution operations are applied to all input components, thus creating convolutional layers. Pooling operations are matrix operations similar to convolutions, but are generally used to aggregate values, reducing the spatial variance and dimensionality of the inputs. In this way, there are no network parameters to be adjusted. Pooling can be defined based on the Maximum, Minimum, Average function, among others. These operations, both pooling and convolution, make up the core of CNN. A very complete material on the subject was presented by [Dumoulin and Visin 2016], focusing on convolutional arithmetic applied to CNN.

2.3. Ensemble Learning Methods

The concept of Ensemble Learning, also called cluster learning, is based on the idea of combining several simpler prediction models (weak learner), training them for the same task and producing, from these, a more complex clustered model (strong learner), which is the sum of its parts. Two algorithms have gained attention: Random Forest and XGBoost.

2.3.1. Random Forest

Random Forest [Breiman 2001] is a type of clustering classifier that gathers a set of random decision trees to define the results. A decision tree is a basic supervised algorithm that performs the construction of a tree to represent a domain. Random Forest can be used for classification and regression problems.

Random Forest creates small subsets from the data and selects subsets of attributes, creating mini trees randomly. The selection of attributes is random: it chooses one to be the root node and generates the child nodes and, finally, repeats this process several times, depending on the number of defined trees. The result is given from the best result obtained, checking the result of each tree.

2.3.2. XGBoost

XGBoost (eXtreme Gradient Boosting) is an implementation of stochastic gradient boosting. This implementation is computationally efficient with many options and is available as a package for the main data science software languages [Bruce and Bruce 2017]. The XGB library implements the gradient boosting decision tree algorithm. It was designed to be highly efficient, flexible and portable. Gradient boosting is an approach where new models are created that predict the residuals or errors of prior models and then added together to make the final prediction. It is called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models. XGB provides a parallel tree boosting that solves many data science problems in a fast and accurate way. This approach supports both regression and classification predictive modeling problems.

3. Related Work

Many musical works produced in the past are still currently available only as original manuscripts or as photocopies. The OMR is needed for the preservation of these works, which requires digitalization and should be transformed into a machine-readable format. Despite the many research activities on optical music recognition (OMR), the results for handwritten musical scores are far from ideal.

A lot of work on the OMR include, for instance: staff lines detection and removal [Rebelo and Cardoso 2013] [Dalitz et al. 2008] [Fujinaga 2004] [dos Santos Cardoso et al. 2009], music symbol segmentation [Rossant and Bloch 2006] [Fornés et al. 2005], a tool proposal to convert handwriting scores into a digital music representation [de Paulo et al. 2015].

For many works, the machine learning area offers several methods to classify the music symbols. In [Wen et al. 2015], the authors propose a new combined neural network classifier, which has the potential to achieve a better recognition accuracy. In [Rebelo et al. 2010], a comparative study of several recognition algorithms of music symbols is presented.

Deep learning models have showed promising results on OMR tasks. The authors in [van der Wel and Ullrich 2017] present a deep learning architecture called a Convolutional Sequence-to-Sequence model to both move towards an end-to-end trainable OMR pipeline, and apply a learning process that trains on full sentences of sheet music instead of individually labeled symbols. In [Huang et al. 2019], the authors note that music object detection is a fundamental part of the OMR pipeline. Thus, they proposed an end-to-end detection model based on a deep convolutional neural network and feature fusion. This model is able to directly process the entire image and then output the symbol categories and the pitch and duration of notes. Already in [Barbosa and Santos 2021], the authors proposed a CNN architecture named CREATES to classify musical symbols for OMR systems. The results of the experiments showed that CREATES outperforms other state-of-the-art classifiers. However, it needs longer time than others classifiers. Thus, in this paper, we propose to improve the CREATES architecture and carry out a comparison to ensembles methods, which are techniques that create multiple models and then combine them to produce improved results.

4. Methodology

CREATES CNN architecture was proposed in [Barbosa and Santos 2021] as part of the solution proposed for the problem of recognition of musical elements. Fig. 1 briefly presents the stages of this solution. The first two stages in Fig. 1, (A) and (B), are applied to both CREATES and other classifiers.

The stage (A) - raw state - is the first stage with images in the initial state from

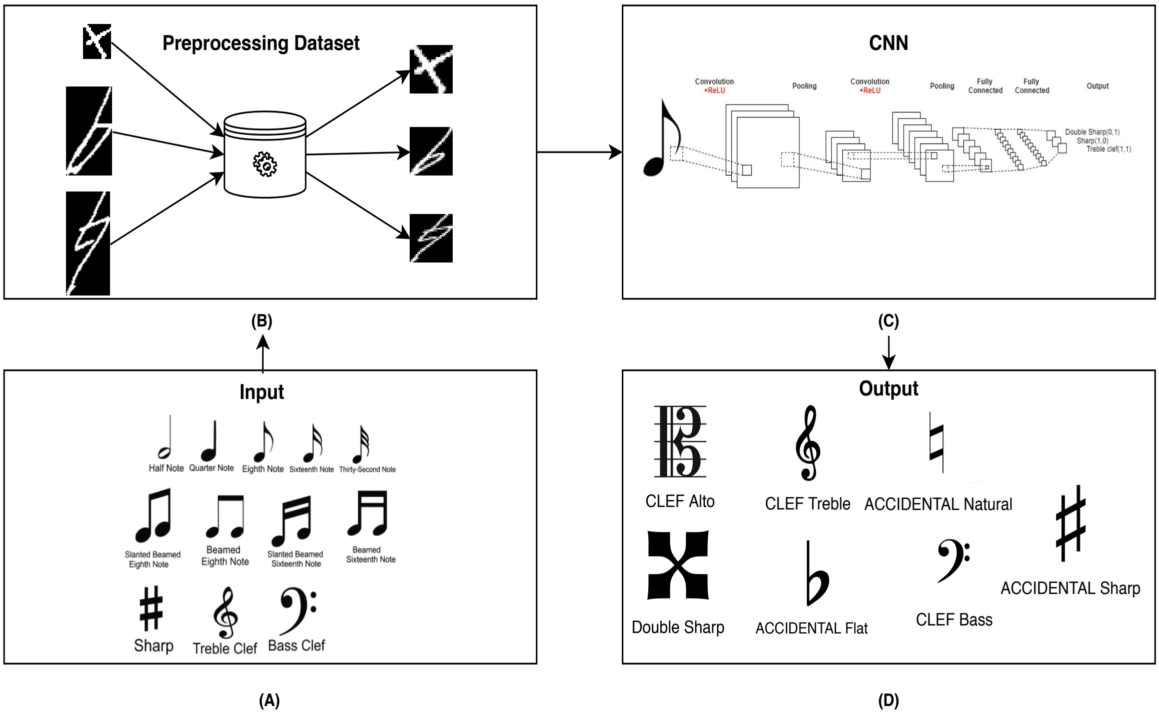

Figure 1. Flowchart of the solution proposed for the problem of recognition of musical elements.

the dataset HOMUS[1] (Handwritten Online Musical Symbols), which has around 15000 samples, of 32 types of musical symbols from 100 different musicians, for research on the recognition of online handwritten music notation. Each musician has their own writing style, as it occurs in handwritten text. Some symbols can be seen in Fig. 2. The images in the set have size of 91x91 pixels, in grayscale. There is little discrepancy between the number of elements for each class.

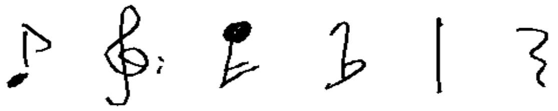

Figure 2. Examples of sample symbols from the HUMUS dataset.

In the second stage (B), a pre-processing in the images was performed to define a standard size (28x28). Here, the OpenCV library is used for this and also an interpolation process to resize. A numerical value was assigned to each musical symbol to label the classes (destination). Next, a random separation process is carried out to select and generate training and test datasets. Thus, the original dataset was divided in 70% for training and 30% for testing. These pre-processed datasets were used in the experiments with the classifiers created by CREATES and ensembles methods, in case Random Forest and XGBoost.

For the training set, a technique to increase the number of samples of each class

[1] HOMUS dataset available in `http://grfia.dlsi.ua.es/homus/`

was used. This technique is available in the *API Keras*[2] and was applied through the *ImageDataGenerator*. The training set was used as input for the algorithms to induce the classifiers, also applying the cross-validation strategy. For this, the *StratifiedKFold* tool from the *Scikit-learn* library in Python was used, setting the number of folds K = 5. The induced classifiers were then tested with the test dataset.

4.1. Modified CREATES Architecture

In the third stage of the solution presented in Fig.1 (C), a CNN architecture was developed from the open source library known as *TensorFlow* to classify a set of musical symbols.

The CNN original architecture was presented in [Barbosa and Santos 2021] and named CREATES. In this paper, this architecture was modified in search of better results for HOMUS dataset. The new architecture for CREATES is presented in Fig. 3. It can be divided in two parts: i) extraction of features (Fig. 3-(A)) and ii) set of dense layers (Fig. 3-(B)). In the step of feature extraction, the CNN architecture has five convolution layers, three pooling layers and four fill layers, the *ZeroPadding*. These fill layers can add rows and columns of zeros to the top, bottom, left, and right of an image. A transform layer is applied as a divider of the two parts of the architecture: the Flatten layer. This model does not have a regularization layer. The parameters, such as number of filters and kernel size, have been defined empirically. In all of these layers, the ReLu activation function is used, since it is popular in works of the area [LeCun et al. 2015].

The second part of the CREATES architecture (Fig. 3-(B)) is responsible for performing model learning and training. In this part, there are four dense layers and one regularization layer *(dropout(0,2))*. The regularization layer applies a 20% discard rate on the value of the previous layer's output to avoid overfitting. On the last layer, the activation function of type *softmax* is applied.

5. Experiments and Analysis of Results

In the experiments, we used CREATES in addition to the following ensemble methods: Random Forest and XGBoost. For the Random Forest and XGBoost algorithms, the implementation platform of *Google Colab*[3] was used, which is free for studies and research with artificial intelligence and data analysis, along with the *Python API*.

XGBoost was run from *XGBClassifier* [4], with the default setting. Only the parameters *objective*, for which *multi:softmax* is used, and the number of classes (set to 32) were changed. Random Forest was run from the library *RandomForestClassifier*[5], configured with default settings. The only variation is the *random state = 42*.

All algorithms were run with the same HOMUS dataset and evaluated during the training and testing process using the classifier evaluation metrics: accuracy, precision, recall and F1-score, which were run from the *Sklearn metrics Classification_report*[6].

[2] https://keras.io/api/preprocessing/image/

[3] Colab: `https://colab.research.google.com/notebooks/intro.ipynb`

[4] XGBClassifier Documentation `https://xgboost.readthedocs.io/en/latest/python/python_api.html\#xgboost.XGBClassifier`

[5] The Library RandomForestClassifier `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html`

[6] Function definitions API `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html`

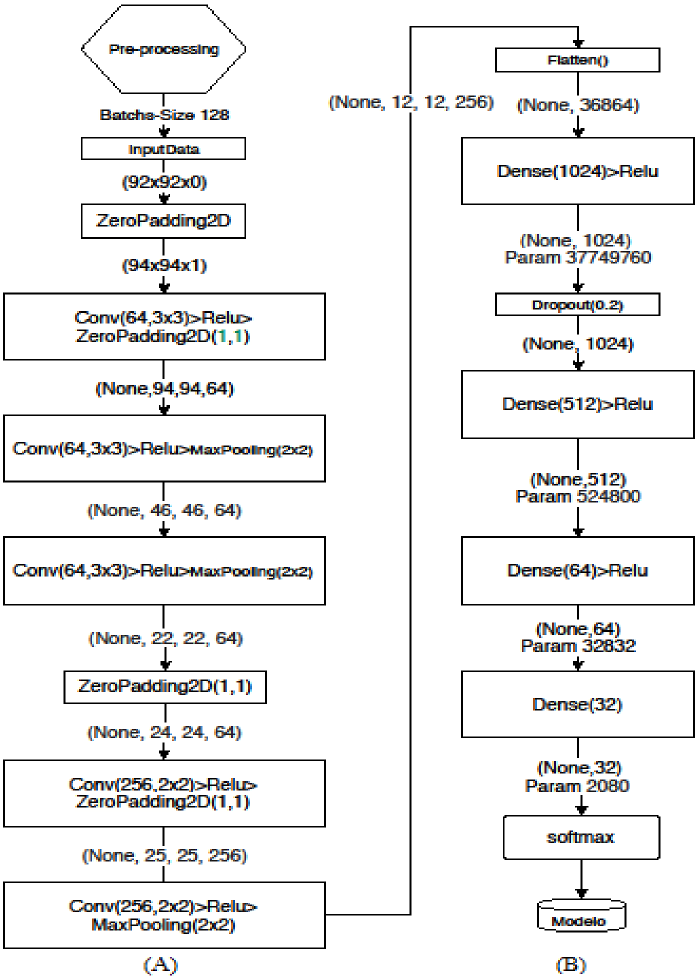

Figure 3. New CREATES CNN architecture.

Table 1 exhibts the results obtained by algorithms. Comparing CREATES to Random Forest and XGBoost algorithms for accuracy, it is possible to highlight the superiority of CREATES, with a variation of 20% and 21%, respectively. Between Random Forest and XGBoost, there was a small variation of 1% to 2%, which can be considered interesting for certain usage scenarios, with gain for Random Forest.

Table 1. Results obtained by CREATES, Random Forest and XGBoost algorithms in the HOMUS dataset.

Algorithms	**Accuracy**	**Precision**	**Recall**	**F1-score**
CREATES	0.85	0.87	0.87	0.87
RandomForest	0.65	0.67	0.66	0.65
XGBoost	0.64	0.63	0.64	0.63

It is important to note that accuracy is a metric that may not clearly represent the potential of the algorithms, as it does not reflect the correct result of unbalanced classes. In view of this, other metrics (Precision, Recall, F1-Score) were also used as comparison among the algorithms. Observing Table 1, it is possible to note that CREATES also got better results than both ensemble methods for these metrics, keeping the same proportion as for accuracy.

According to Table 1, it is possible to see that CREATES clearly differs from the others, demonstrating its ability to adequately deal with this type of problem, with metrics reaching a level of 87%. If the results of the algorithms in some specific classes are analyzed, such as the 12-8 Time class, it is possible to affirm that, for certain classes, the algorithms behave very closely. See the results obtained for this class in Table 2.

Table 2. Comparison among algorithms for 12-8-Time class.

Algorithms	Precision	Recall	F1
CREATES	0.97	0.97	0.97
RandomForest	0.77	0.97	0.86
XGBoost	0.89	0.95	0.92

The experiments carried out in this work bring results that demonstrate the possibility of using convolutional neural networks (CNN) to classify musical symbols. However, in the HOMUS dataset, CREATES still spent more time in the training process (around 5 hours) than the ensemble methods (using Google Colab, it was not possible to measure the training time of the ensemble methods reliably). In the prediction/classification process (using the test dataset), it takes around 2 seconds. In order, to obtain results in a reasonable time, CREATES was trained with the help of GPU, to the detriment of the other two ensemble methods that did not need this resource.

6. Conclusions

In this paper, it was proposed the investigation of algorithms and classification methods that can help in the recognition of musical elements for OMR applications. our main proposal was to compare Convolutional Neural Networks and ensemble methods, which can obtain better results than simple classifiers. The results obtained by CNN were considered satisfactory when compared to two ensemble learning methods, XGBoost and Random Forest. However, it needs more time and computational resources for training.

A specific architecture for the CNN called CREATES has been redefined. CREATES was evaluated on the HOMUS dataset, known in the literature for recognizing handwritten musical notation. When compared to XGBoost and Random Forest, CREATES CNN got better classification results. However, it needs more computational resources such as GPU usage. Despite this, it was not necessary to apply feature extraction methods, in pre-processing steps, which are normally required with the use of other classification methods. Thus, it is possible to conclude that CNN can bring advantages that contribute to the advancement of research in the area of OMR.

According to the literature, it is noticed that the area of OMR (Character Recognition) manuscripts is still new and full of obstacles. The results obtained in this paper indicate that there is much to be done in this area. Thus, it is intended to continue the development of this work, taking advantage of the experience and the results acquired and applying to the development of a system to convert handwriting scores into a digital music representation.

References

Bainbridge, D. and Bell, T. (2001). The challenge of optical music recognition. *Computers and the Humanities*, 35(2):95–121.

Barbosa, J. and Santos, E. (2021). Creates - convolutional neural network applied to identification of musical elements in omr. In *Anais do XVIII Simpósio Brasileiro de Computação Musical*, pages 221–224, Porto Alegre, RS, Brasil. SBC.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Bruce, P. and Bruce, A. (2017). *Practical Statistics for Data Scientists*. O'Reilly Media, Sebastopol, CA.

Calvo-Zaragoza, J., Jr, J. H., and Pacha, A. (2020). Understanding optical music recognition. *ACM Computing Surveys (CSUR)*, 53(4):1–35.

Dalitz, C., Droettboom, M., Pranzas, B., and Fujinaga, I. (2008). A comparative study of staff removal algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):753–766.

de Paulo, A. M., Schiavoni, F. L., de Matos Laia, M. A., and Madeira, D. L. A. (2015). Copista-sistema de omr para a recuperaç ao de acervo histórico musical. *XV SBCM-Computer Music: Beyond the frontiers of signal processing and computational models*.

dos Santos Cardoso, J., Capela, A., Rebelo, A., Guedes, C., and da Costa, J. P. (2009). Staff detection with stable paths. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):1134–1139.

Dumoulin, V. and Visin, F. (2016). A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*.

Fornés, A., Lladós, J., and Sánchez, G. (2005). Primitive segmentation in old handwritten music scores. In *International Workshop on Graphics Recognition*, pages 279–290. Springer.

Fujinaga, I. (2004). Staff detection and removal. In *Visual Perception of Music Notation: On-Line and Off Line Recognition*, pages 1–39. IGI Global.

Huang, Z., Jia, X., and Guo, Y. (2019). State-of-the-art model for music object recognition with deep learning. *Applied Sciences*, 9(13).

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

Rebelo, A., Capela, G., and Cardoso, J. S. (2010). Optical recognition of music symbols. *International Journal on Document Analysis and Recognition (IJDAR)*, 13(1):19–31.

Rebelo, A. and Cardoso, J. S. (2013). Staff line detection and removal in the grayscale domain. In *2013 12th International Conference on Document Analysis and Recognition*, pages 57–61. IEEE.

Rossant, F. and Bloch, I. (2006). Robust and adaptive omr system including fuzzy modeling, fusion of musical rules, and possible error detection. *EURASIP Journal on Advances in Signal Processing*, 2007(1):081541.

van der Wel, E. and Ullrich, K. (2017). Optical music recognition with convolutional sequence-to-sequence models. In Cunningham, S. J., Duan, Z., Hu, X., and Turnbull, D., editors, *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017*, pages 731–737.

Wen, C., Rebelo, A., Zhang, J., and Cardoso, J. (2015). A new optical music recognition system based on combined neural network. *Pattern Recognition Letters*, 58:1–7.