DeepScores – A Dataset for Segmentation, Detection and Classification of Tiny Objects

Lukas Tuggener
ZHAW Datalab & USI
tugg@zhaw.ch

Ismail Elezi
University of Venice & ZHAW
ismail.elezi@unive.it

Jürgen Schmidhuber
IDSIA & USI
juergen@idsia.ch

Marcello Pelillo
University of Venice
pelillo@unive.it

Thilo Stadelmann
ZHAW Datalab
stdm@zhaw.ch

***Abstract*—We present the DeepScores dataset with the goal of advancing the state-of-the-art in small object recognition by placing the question of object recognition in the context of scene understanding. DeepScores contains high quality images of musical scores, partitioned into $300,000$ sheets of written music that contain symbols of different shapes and sizes. With close to a hundred million small objects, this makes our dataset not only unique, but also the largest public dataset. DeepScores comes with ground truth for object classification, detection and semantic segmentation. DeepScores thus poses a relevant challenge for computer vision in general, and optical music recognition (OMR) research in particular. We present a detailed statistical analysis of the dataset, comparing it with other computer vision datasets like PASCAL VOC, SUN, SVHN, ImageNet, MS-COCO, as well as with other OMR datasets. Finally, we provide baseline performances for object classification, intuition for the inherent difficulty that DeepScores poses to state-of-the-art object detectors like YOLO or R-CNN, and give pointers to future research based on this dataset.**

I. Introduction

Increased availability of data and computational power has often been followed by progress in computer vision and machine learning. The recent rise of deep learning in computer vision for instance has been promoted by the availability of large image datasets [15] and increased computational power provided by GPUs [6], [10], [14].

Optical music recognition (OMR) [5] is a classical and challenging area of document recognition and computer vision that aims at converting scans of written music to machine-readable form, much like optical character recognition (OCR) [2] does for printed text. While results on simplified tasks show promising results [4], [32], there is yet no OMR solution that leverages the power of deep learning. We conjecture that this is caused in part by the lack of publicly available datasets of written music, big enough to train deep neural networks. The *DeepScores* dataset has been collected with OMR in mind, but as well addresses important aspects of next generation computer vision research that pertain to the size and number of objects per image.

Although there is already a number of clean, large datasets available to the computer vision community [15]–[19], those datasets are similar to each other in the sense that for each image there are a few large objects of interest. Object detection approaches that have shown state-of-the-art performance under these circumstances, such as Faster R-CNN [20], SSD [21] and YOLO [22], demonstrate very poor off-the-shelf performances when applied to environments with large input images containing multiple small objects (see Section IV).

Sheets of written music, on the other hand, usually have dozens to hundreds of small salient objects. The class distribution of musical symbols is strongly skewed and the symbols have a large variability in size. Additionally, the OMR problem is very different from modern OCR [3], [23]: while in classical OCR, the text is basically a 1D signal (symbols to be recognized are organized in lines of fixed height, in which they extend from left to right or vice versa), musical notation can additionally be stacked arbitrarily also on the vertical axis, thus becoming a 2D signal. This superposition property would exponentially increase the number of symbols to be recognized, if approached the usual way (which is intractable from a computational as well as from a classification point of view). It also makes segmentation very hard and does not imply a natural ordering of the symbols as for example in the SVHN dataset [18].

In this paper, we present the *DeepScores* dataset with the following contributions: a) a curated dataset of a collection of hundreds of thousands of musical scores, containing tens of millions of objects to construct a high quality dataset of written music; b) available ground truth for the tasks of object detection, semantic segmentation, and classification; c) comprehensive comparisons with other computer vision datasets (see Section II) and a quantitative and qualitative analysis of *DeepScores* (see Section III); d) computation of an object classification baseline and a qualitative assessment of current off-the-shelf detection methods along with reasoning why detection needs new approaches on *DeepScores* (see Section IV); e) proposals on how to facilitate next generation computer vision research using *DeepScores* (see Section V). The data, a recommended evaluation scheme and accompanying TensorFlow [35] code are freely available[1].

II. *DeepScores* in the context of other datasets

DeepScores is a high quality dataset consisting of pages of written music, rendered at 400 dots per inch (dpi). It has $300,000$ full pages as images, containing tens of millions of objects, separated into 123 classes (cp. Figure 1). The aim of the dataset is to facilitate general research on small

[1] https://tuggeluk.github.io/deepscores/

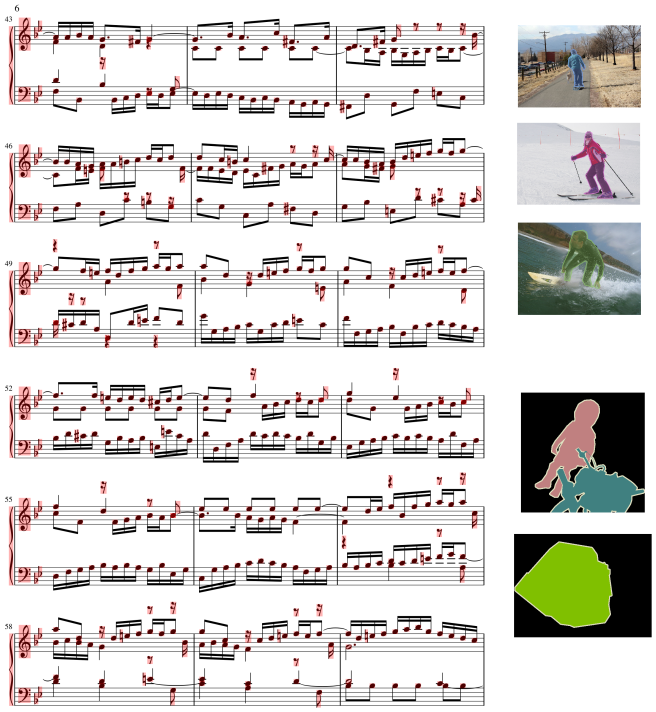
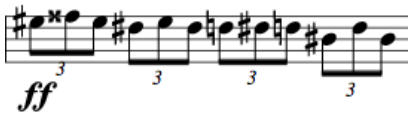
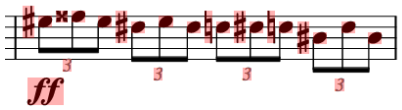


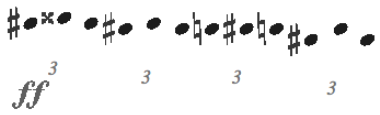
Fig. 1: A typical image and ground truth from the *DeepScores* dataset (left), next to examples from the MS-COCO (3 images, top right) and PASCAL VOC (2, bottom right) datasets. Even though the music page is rendered at a much higher resolution, the objects are still smaller; the size ratio between the images is truthful despite all images being downscaled.



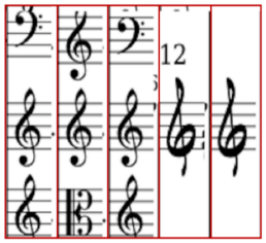
(a) Snippet of an input image.



(b) Boundig boxes rendered over single objects from snippet 2a for object detection.



(c) Color-based pixel level labels (the differences are hard to recognize visually, but there is a distinct color per symbol class) for semantic segmentation.



(d) Patches centered around specific symbols (in this case: `gClef`) for object classification.

Fig. 2: Examples for the different flavors of ground truth available in *DeepScores*.

object recognition, with direct applicability to the recognition of musical symbols. We provide the dataset with ground truth for the following tasks: object classification, semantic segmentation, and object detection (cp. Figure 2).

Object classification in the context of computer vision is the procedure of labeling an image with a single label. Its recent history is closely linked to the success of deep convolutional learning models [8], [9], leading to superhuman performance [6] and subsequent ImageNet object classification breakthroughs [24]. Shortly afterwards, similar systems achieved human-level accuracy also on ImageNet [11], [25], [26]. Generally speaking, the ImageNet dataset [15] was a key ingredient to the success of image classification algorithms.

In *DeepScores*, we provide data for the classification task even though classifying musical symbols in isolation is not a challenging problem compared to classifying ImageNet images. But providing the dataset for classification, in addition to a neural network implementation that achieves high accuracy (see Section IV), might help to address the other two tasks. In fact, the first step in many computer vision models is to use a deep convolutional neural network pre-trained on ImageNet, and alter it for the task of image segmentation or image detection [20], [27]. We expect that the same technique can be used when it comes to OMR and detecting very small objects.

Semantic segmentation is the task of labeling each pixel of the image with one of the possible classes. State-of-the-art models are typically based on fully convolutional architectures [13], [27]. The task is arguably a significantly more difficult problem than image classification, with the recent success being largely attributed to the release of high quality datasets like PASCAL VOC [16] and MS-COCO [19].

In *DeepScores*, we provide ground truth for each pixel in all the images, having roughly 10^{12} labeled pixels in the dataset.

Object detection is the by far most interesting and challenging task: to classify all the objects in the image, and at the same time to find their precise position. State-of-the-art algorithms are pipeline convolutional models, typically having combined cost functions for detection and classification [20]–[22]. Similar to the case of semantic segmentation above, the PASCAL VOC and especially MS-COCO datasets have played an important part on the recent success of object detection algorithms.

In *DeepScores*, we provide bounding boxes and labels for each of the musical symbols in the dataset. With around 80 million objects, this makes our dataset the largest one released

Dataset	#classes	#images	#objects	#pixels
SUN	**397**	17k	17k	6b
PASCAL VOC	21	10k	30k	2.5b
MS COCO	91	330k	3.5m	100b
ImageNet	200	**500k**	600k	125b
SVHN	10	200k	630k	4b
DeepScores	123	300k	**80m**	**1.5t**

TABLE I: Information about the number of classes, images and objects for some of the most common used datasets in computer vision. The number of pixels is estimated due to most datasets not having fixed image sizes. We use the SUN 2012 object detection specifications for SUN, and the statistics of the ILSVRC 2014 [1] detection task for ImageNet.

so far, and highly challenging. More on the challenges of *DeepScores* is provided in section IV.

A. Comparisons with computer vision datasets

Compared with some of the most used datasets in the field of computer vision, *DeepScores* has by far the largest number of objects, as well as the highest resolution. In particular, images of *DeepScores* typically have a resolution of $1,894$ x $2,668$ pixels, which is at least four times higher than the resolutions of datasets we compare with. Table I contains quantitative comparisons of *DeepScores* with other datasets, while the following paragraphs bring in also qualitative aspects.

SVHN, the street view house numbers dataset [18], contains $600,000$ labeled digits cropped from street view images. Compared to *DeepScores*, the number of objects in SVHN is two orders of magnitude lower, and the number of objects per image is two to three orders of magnitude lower.

ImageNet contains a large number of images and (as a competition) different tracks (classification, detection and segmentation) that together have proven to be a solid foundation for many computer vision projects. However, the objects in ImageNet are quite large, while the number of objects per image is very small. Unlike ImageNet, *DeepScores* tries to address this issue by going to the other extreme, providing a very large number of very small objects on images having significantly higher resolution than all the other mentioned datasets.

PASCAL VOC is a dataset which has been assembled mostly for the tasks of detection and segmentation. Compared to ImageNet, the dataset has slightly more objects per image, but the number of images is comparatively small: our dataset is one order of magnitude bigger in the number of images, and three orders of magnitude bigger in the number of objects.

MS-COCO is a large upgrade over PASCAL VOC on both the number of images and number of objects per image. With more than 300k images containing more than 3 million objects, the dataset is very useful for various tasks in computer vision. However, like ImageNet, the number of objects per image is still more than one order of magnitude lower than in our dataset, while the objects are relatively large.

B. Comparisons with OMR datasets

A number of OMR datasets have been released in the past with a specific focus on the computer music community. *DeepScores* will be of use both for general computer vision as well as to the OMR community (compare Section IV).

a) Handwritten scores:

The Handwritten Online Musical Symbols dataset **HOMUS** [28] is a reference corpus with around $15,000$ samples for research on the recognition of online handwritten music notation. For each sample, the individual strokes that the musician wrote on a Samsung tablet using a stylus were recorded and can be used in online and offline scenarios.

The **CVC-MUSCIMA** database [29] contains handwritten music images, which have been specifically designed for writer identification and staff removal tasks. The database contains $1,000$ music sheets written by 50 different musicians with characteristic handwriting styles.

MUSCIMA++ [30] is a dataset of handwritten music for musical symbol detection that is based on the MUSCIMA dataset. It contains $91,255$ written symbols, consisting of both notation primitives and higher-level notation objects, such as key signatures or time signatures. There are $23,352$ notes in the dataset, of which $21,356$ have a full notehead, $1,648$ have an empty notehead, and 348 are grace notes.

The **Capitan Collection** [31] is a corpus collected via an electronic pen while tracing isolated music symbols from early manuscripts. The dataset contains information on both the sequence followed by the pen (capitan stroke) as well as the patch of the source under the tracing itself (capitan score). In total, the dataset contains $10,230$ samples unevenly spread over 30 classes.

b) Print quality scores:

The **MuseScore Monophonic MusicXML Dataset** [4] is one of the largest OMR dataset to date, consisting of $17,000$ monophonic scores. While the dataset has high quality images, it doesn't resemble real-world musical scores which are not monophonic and thus have many lines per image.

Further OMR datasets of printed scores are reviewed by the **OMR-Datasets** project[2]. *DeepScores* is by far larger than any of these or the above-mentioned datasets, containing more images and musical symbols than all the other datasets combined. In addition, *DeepScores* contains only real-world scores (i.e., symbols in context as they appear in real written music), while most other datasets are either synthetic or reduced (containing only symbols in isolation or just a line per image). The sheer scale of *DeepScores* makes it highly usable for modern deep learning algorithms. While convolutional neural networks have been used before for OMR [4], *DeepScores* for the first time enables the training of very large and deep models.

III. The *DeepScores* dataset

A. Quantitative properties

DeepScores contains around $300,000$ pages of digitally rendered music scores (see Sections III-C and IV-A for a

[2]See https://apacha.github.io/OMR-Datasets/.

Statistic	Symbols per sheet	Symbols per class
Mean	243	650k
Std. dev.	203	4m
Maximum	7′664	44m
Minimum	4	18
Median	212	20k

TABLE II: Statistical measures for the occurrence of symbols per musical sheet and per class (rounded).

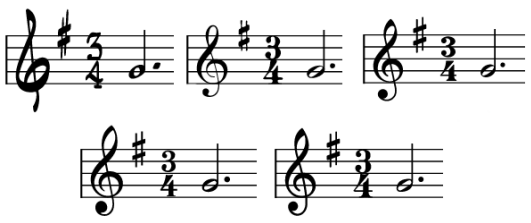


Fig. 3: The same patch of a musical sheet, rendered using five different fonts.

justification of synthetic data) and has ground truth for 123 different symbol classes. The number of labeled music symbol instances is roughly 80 million (4-5 orders of magnitude higher than in the other music datasets; when speaking of symbols, we mean labeled musical symbols that are to be recognized as objects in the task at hand). The number of symbols on one page can vary from as low as 4 to as high as $7,664$ symbols. On average, a sheet (i.e., an image) contains around 243 symbols. Table II gives the mean, standard deviation, median, maximum and minimum number of symbols per page in the "symbols per sheet" column.

Another interesting aspect of *DeepScores* is the class distribution. Obviously, some classes contain more symbols than other classes (see also Table II, column 3). It can be seen that the average number of elements per class is 650k but the standard deviation is 4m, illustrating that the distribution of symbols per class is very unbalanced.

B. Flavors of ground truth

In order for *DeepScores* to be useful for as many applications as possible, we offer ground truth for three different tasks. For object classification, there are up to $3,000$ labeled image patches per class, i.e. we do not provide each of the 80m symbols as a single patch for classification purposes. Instead, we constrain the dataset for this simpler task to a random subset of reasonable size (see Section IV). The patches have size 220×120 and contain the full original context of the symbol (i.e., they are cropped out of real world musical scores). Each patch is centered around the symbol's bounding box (see Figure 2d).

For object detection, an accompanying XML file for each image in *DeepScores* holds an `object` node for each symbol instance present on the page. It contains its class and bounding box coordinates, visualized in Figure 2b.

For semantic segmentation, there is an accompanying PNG file for each image. This PNG has identical size as the initial image, but each pixel has been recolored to represent the symbol class it is part of. As in Figure 2c, the background is white, with the published images using grayscale colors from 0 to 123 for ease of use in the softmax layer of potential models.

C. Dataset construction

DeepScores is constructed by synthesizing sheet music from a large collection of written music in a digital format: crowd-sourced MusicXML files publicly available from MuseScore[3] and used by permission. The rendering of MusicXML and the generation of accompanying ground truth is one of the main contributions of this work. Going from online MusicXML archives to a curated dataset is non-trivial due to extensive musical know-how and non-available custom software components necessary to create examples containing symbols and *corresponding* object locations. It involves a) code to be injected in the LilyPond[4] SVG backend such that the printed SVG paths contain additional meta data for each individual symbol; b) software that maps each found path to one of the predefined object classes and renders colored PNG files correctly (i.e., crisp edges, exact localization etc) as well as XML descriptions; and c) software that constructs the final ground truth out of generated meta data. All steps have been aligned with musicians to guarantee fitness for the OMR task.

To achieve a realistic variety in the data even though all images are digitally rendered and therefore have perfect image quality, five different music fonts have been used for rendering (see Figure 3). The challenge of the dataset however is not in the variety of the presentation of the different symbol instances, as is the case with traditional object detection datasets (see Section IV-A).

A key feature of a dataset is the definition of the classes to be included. Due to their compositional nature, there are many ways to define classes of music symbols: is it for example a "c" note with duration 8 (`cNote8th`) or is it a black notehead (`noteheadBlack`) and a flag (`flag8thUp` or `flag8thDown`)? Adding to this complexity, there is a huge number of special and thus infrequent symbols in music notation. The selected set is the result of many discussions with music experts and contains the most important symbols. We decided to use atomic symbol parts as classes which makes it possible to define composite symbols in an application-dependent way. As a result of these discussions we also decided to focus on fixed-shape symbols and have left out stems, barlines, staff and ledger lines.

IV. Experiments and impact

A. Unique challenges

One of the key challenges *DeepScores* poses upon modeling approaches is the sheer amount of objects on a single image.

[3] See https://musescore.com.

[4] See http://lilypond.org/.

There are two additional properties of music notation imposing challenges. First, there is a big variability in object size ranging from less than hundred to many thousands of pixels in area. Second, context matters in music notation: two objects having the same appearance can belong to a different class depending on the local surroundings (see Figure 4). To our knowledge there is no other freely available large scale dataset that shares this trait.

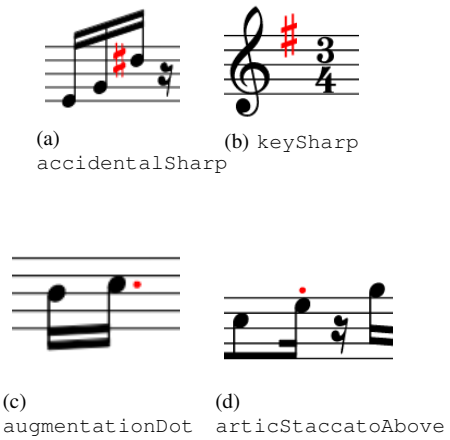


Fig. 4: Examples of the importance of context for classifying musical symbols: in both rows, the class of otherwise similar looking objects changes depending on the surrounding objects.

Moreover, datasets like ImageNet are close to being perfectly balanced, with the number of images and objects per class being a constant. This clearly isn't the case with the *DeepScores* dataset, where the most common class contains more than half of the symbols in the dataset, and the top 10% of classes contain more than 85% of the symbols in the entire dataset. This extremely skewed distribution resembles many real-world cases for example in anomaly detection and industrial quality control.

B. Analysis of off-the-shelf deep learning models

Merely classifying the musical symbols in *DeepScores* is expected to be relatively simple: all symbols have very clear black and white borders, their shape has limited variability and they are rendered at a very high resolution. We fitted a simple residual CNN [26] with 25 convolutional layers and about 8 million trainable parameters. Using the Adam optimizer with the hyperparameters proposed by the authors [12], we reached a macro average accuracy of over 0.98 in about ten epochs. This shows that CNNs are able to deal with labels that not only depend on an object, but also its surroundings.

Detection, however, is more challenging. This is due to the sheer amount of small objects present on each image, which stands in stark contrast to the low number of prominent objects present in natural images. Since the leading detection systems SSD, YOLO and Faster R-CNN have been developed with natural images in mind, *DeepScores* is a tough challenge for them. Evaluating the input on a fixed $S \times S$ grid makes YOLO inherently unfit to deal with clusters of small symbols: *"YOLO imposes strong spatial constraints on bounding box predictions since each grid cell only predicts two boxes and can only have one class. This spatial constraint limits the number of nearby objects that our model can predict. Our model struggles with small objects that appear in groups, such as flocks of birds"* [22]. SSD uses features from six of the top layers, but it still struggles with small objects as visible in Figure 4 of the original publication [21].

Therefore, we ran experiments only with Faster-RCNN, using smaller anchors as in the published configuration to adapt to our task. The system was unable to find any symbols at all. It is unclear whether fine tuning the region proposal network and anchor setup will lead to a good performance on *DeepScores*. Instead, a novel detection method currently under development and based on fully convolutional neural networks [27] shows promising preliminary results on *DeepScores*, validating our intuition that the dataset is well-suited for the development of new methods focused on many tiny objects.

C. Expected impact

Both observations—easy classification but challenging detection—lie at the heart of what we think makes *DeepScores* very useful: on the one hand, it offers the challenging scenario of many tiny objects that cannot be approached using current datasets. On the other hand, *DeepScores* is probably the easiest scenario of that kind, because classifying single musical objects is relatively easy and the dataset contains a vast amount of training data. *DeepScores* thus is a prime candidate to develop next generation document recognition and computer vision methods that scale to many tiny objects on large images: many real-world problems deal with high-resolution images, with images containing hundreds of objects and with images containing very small objects in them. This might be OMR itself, automated driving and other robotics use cases, medical applications with full-resolution imaging techniques as data sources, vision-based industrial quality control, or surveillance tasks e.g. in sports arenas and other public places.

Finally, *DeepScores* will be a valuable source for pre-training models: transfer learning has been one of the most important ingredients in the advancement of computer vision. The first step in many computer vision models [20], [27] is to use a deep convolutional neural network pre-trained on ImageNet, and alter it for the task of image segmentation or object detection, or use it on considerably smaller, task-dependent final training sets.

V. Conclusions

We have presented the conception and creation of *DeepScores* - one of the largest publicly and freely available datasets for OMR and computer vision applications in general in terms of image size and number of contained objects. Compared to other well-known datasets, *DeepScores* has large images (more than four times larger than the average) containing many (one to two orders of magnitude more) very small (down to a few pixels, but varying by several orders of magnitude) objects that

change their class belonging depending on the visual context. The dataset is made up of sheets of written music, synthesized from the largest public corpus of MusicXML. It comprises ground truth for the tasks of object classification, semantic segmentation and object detection.

We have argued that the unique properties of *DeepScores* make the dataset suitable for use in the development of general next generation computer vision methods that are able to work on large images with tiny objects. This ability is crucial for real-world applications like robotics, automated driving, medical image analysis, industrial quality control or surveillance, besides OMR. We have motivated that object classification is relatively easy on *DeepScores*, making it therefore the potentially cheapest way to work on a challenging detection task. We thus expect impact on general object detection algorithms.

One weakness of the *DeepScores* dataset is that all the data is currently digitally rendered. Linear models (or piecewise linear models like neural networks) have been shown to not generalize well when the distribution of the real-world data is far from the distribution of the dataset the model has been trained on [33], [34]. Our experiments show that networks trained on *DeepScores* do generalize to high quality scans, but processing lower quality images remains a challenge. To address this issue, we currently construct training data that consist of flatbed-scans and photos of low-res prints. Many colleagues mentioned that ground truth for non-fixed shape symbols (e.g. slurs, beams) is of high importance to them, therefore are we working *on* an updated version of *DeepScores* that carries this information.

Future work *with* the dataset will – besides the general impact predicted above – directly impact OMR: the full potential of deep neural networks is still to be realized on musical scores.

ACKNOWLEDGEMENTS

This work is financially supported by CTI grant 17963.1 PFES-ES "DeepScore". The authors are grateful for the support of Hervé Bitteur of Audiveris, the permission to use MuseScore data, and the collaboration with ScorePad AG.

REFERENCES

[1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, A. C. Berg and L. Fe-Fei, *ImageNet Large Scale Visual Recognition Challenge*, International Journal of Computer Vision, 2015.

[2] M. Shunji, N. Hirobumi, Y. Su and J. Hiromitsu, *Optical character recognition*, John Wiley & Sons, Inc., 1999.

[3] C. Y. Lee and S. Osindero, *Recursive Recurrent Nets with Attention Modeling for OCR in the Wild*, Computer Vision and Pattern Recognition, 2016.

[4] E. van der Wel and K. Ullrich, *Optical Music Recognition with Convolutional Sequence to-Sequence Models*, http://arxiv.org/abs/1707.04877, 2017.

[5] A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. R. S. Marcal, C. Guedes and J. S. Cardaso, *Optical music recognition: state-of-the-art and open issues*, International Journal of Multimedia Information Retrieval, 2012.

[6] D. C. Ciresan, U. Meier and J. Schmidhuber, *Multi-Column Deep Neural Networks for Image Classification*, Computer Vision and Pattern Recognition, 2012.

[7] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella and J. Schmidhuber, *High-Performance Neural Networks for Visual Object Classification*, arXiv:1102.0183v1 [cs.AI], 2011.

[8] K. Fukushima, *Neocognitron: A self-organizing neural network for a mechanism of pattern recognition unaffected by shift in position*, Biological Cybernetics, 1980.

[9] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard and L. D. Jackel, *Handwritten digit recognition with a back-propagation network.*, NIPS, 1990.

[10] K-S. Oh and K. Jung, *GPU implementation of neural networks*, Pattern Recognition, 2004.

[11] R. K. Srivastava, K. Greff and J. Schmidhuber, *Highway networks*, arXiv preprint arXiv:1505.00387, 2015.

[12] D. Kingma and J. Ba, *Adam: A method for stochastic optimization*, International Conference on Learning Representations, 2015.

[13] D. C. Ciresan, A. Giusti, L. M. Gambardella and J. Schmidhuber, *Neural Networks for Segmenting Neuronal Structures in EM Stacks*, ISBI Segmentation Challenge Competition: Abstracts, 2012.

[14] R. Raina, A. Madhavan and A. Y. Ng, *Large-scale deep unsupervised learning using graphics processors*, ICML, 2009.

[15] J. Deng, W. Dong, R. Socher, J. L. Li and L. Fei-Fei, *Imagenet: A large-scale hierarchical image database.*, Computer Vision and Pattern Recognition, 2009.

[16] M. Everingham, L. Van Gool, C. K. Williams, J. Winn and A. Zisserman, *The pascal visual object classes (voc) challenge.*, International journal of computer vision, 2010.

[17] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva and A. Torralba, *Sun database: Large-scale scene recognition from abbey to zoo.*, Computer Vision and Pattern Recognition, 2010.

[18] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu and A. Y. Ng, *Reading digits in natural images with unsupervised feature learning.*, NIPS workshop, 2011.

[19] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan and C. L. Zitnick, *Microsoft coco: Common objects in context.*, European Conference in Computer Vision, 2014.

[20] S. Ren, K. He, R. Girshick and J. Sun, *Faster R-CNN: Towards real-time object detection with region proposal networks.*, NIPS, 2014.

[21] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu and A. C. Berg, *Ssd: Single shot multibox detector.*, European Conference in Computer Vision, 2016.

[22] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, *You only look once: Unified, real-time object detection.*, ICCV, 2016.

[23] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud and V. Shet, *Multi-digit number recognition from street view imagery using deep convolutional neural networks.*, arXiv preprint arXiv:1312.6082., 2013.

[24] A. Krizhevsky, I. Sutskever and G. E. Hinton, *Imagenet classification with deep convolutional neural networks.*, NIPS, 2012.

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, V. Vanhoucke and A. Rabinovich, *Going deeper with convolutions.*, Computer Vision and Pattern Recognition, 2015.

[26] K. He, X. Zhang, S. Ren and J. Sun, *Deep residual learning for image recognition.*, Computer Vision and Pattern Recognition, 2016.

[27] J. Long, E. Shelhamer and T. Darrell, *Fully convolutional networks for semantic segmentation.*, Computer Vision and Pattern Recognition, 2015.

[28] J. Calvo-Zaragoza and J. Oncina, *Recognition of Pen-Based Music Notation: The HOMUS Dataset.*, ICPR, 2014.

[29] A. Fornes, A. Dutta, A. Gordo and J. Llados, *A Ground-truth of Handwritten Music Score Images for Writer Identification and Staff Removal.*, International Journal on Document Analysis and Recognition, 2012.

[30] J. Hajic and P. Pecina, *The MUSCIMA++ Dataset for Handwritten Optical Music Recognition.*, ICDAR 2017

[31] J. Calvo-Zaragoza, D. Rizo and J. M. Inesta, *Two (note) heads are better than one: pen-based multimodal interaction with music scores.*, International Society of Music Information Retrieval conference, 2016.

[32] Jorge Calvo-Zaragoza,Jose J. Valero-Mas and Antonio Pertusa, *End-to-End Optical Music Recognition Using Neural Networks*, ISMRIR 2017

[33] A. Torralba and A. Efros, *Unbiased look at dataset bias*, Computer Vision and Pattern Recognition, 2011

[34] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, R. Erhan, I. Goodfellow and R. Fergus, *Intriguing properties of neural networks.*, arXiv:1312.6199., 2013.

[35] Martin Abadi et al., *Tensorflow: Large-scale machine learning on heterogeneous distributed systems.*, arXiv preprint arXiv:1603.04467, 2016.