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

Lukas Tuggener
ZHAW Datalab & USI
Winterthur & Lugano, Switzerland
tugg@zhaw.ch, lukas.tuggener@usi.ch

Ismail Elezi
Ca' Foscari University & ZHAW Datalab
Venice, Italy & Winterthur, Switzerland
ismail.elezi@unive.it, xisa@zhaw.ch

Jürgen Schmidhuber
IDSIA & USI
Manno & Lugano, Switzerland
juergen@idsia.ch

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

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

Abstract

We present the DeepScores dataset with the goal of advancing the state-of-the-art in small objects recognition, and 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 millions of 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, beyond the scope of optical music recognition (OMR) research. We present a detailed statistical analysis of the dataset, comparing it with other computer vision datasets like Caltech101/256, PASCAL VOC, SUN, SVHN, ImageNet, MS-COCO, smaller computer vision datasets, as well as with other OMR datasets. Finally, we provide baseline performances for object classification and give pointers to future research based on this dataset.*

1. 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 availability of large image datasets [6] and increased computational power provided by GPUs [29, 30, 5].

Optical music recognition (OMR) [31] is a classical and challenging area of computer vision that aims at converting scans of written music to machine-readable form, much like optical character recognition (OCR) [27] does it for printed text. To the best of our knowledge, there are no OMR systems yet that fully leverage 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 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 [8, 6, 42, 7, 28, 23], 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 [33], SSD [25] and YOLO [32], demonstrate very poor off-the-shelf performances when applied to environments with large input images containing multiple small objects (see Section 4).

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 [11, 22]: 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

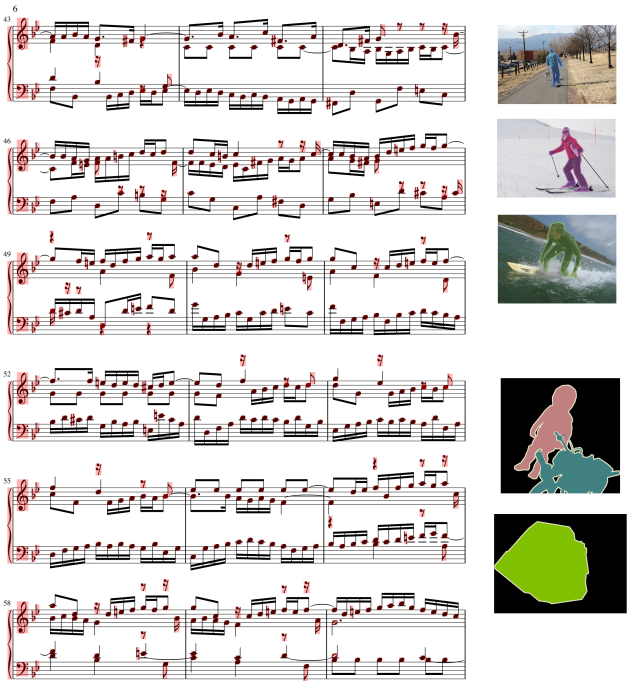
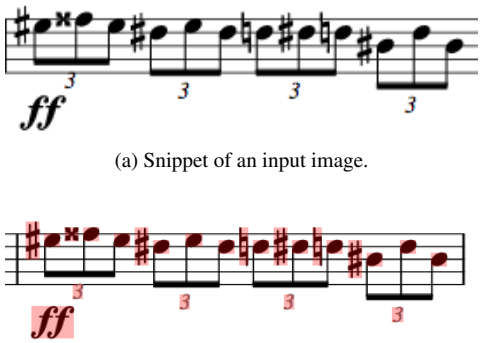
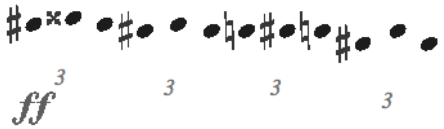


Figure 1: A typical image and ground truth from the *DeepScores* dataset (left), next to examples from the MS-COCO (3 images, top right) and PASCAL VOC (2 images, bottom right) datasets. Even tough the music page is rendered at a much higher resolution, the objects are still smaller; the size ratio between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

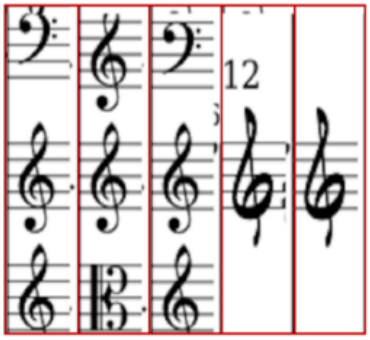


(a) Snippet of an input image.

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



(c) Color-based pixel level labels (the differences can be hard to see, 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.

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

hard and does not imply a natural ordering of the symbols as for example in the SVHN dataset [28].

In this paper, we present the *DeepScores* dataset with the following contributions: a) a curated and publicly available[1] 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 2) and a quantitative and qualitative analysis of *DeepScores* (see Section 3); d) computation of an object classification baseline (see Section 4) together with an outlook on how to facilitate next generation computer vision research using *DeepScores* (see Section 5).

2. *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 in 118 classes. The aim of the dataset is to facilitate general research on small object recognition, with direct applicability to the recognition of musical symbols. We provide the dataset with three different kinds of ground truths (in the order of progressively increasing task complexity): object classification, semantic segmentation, and object detection.

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 [10, 20], leading to superhuman performance [4, 5] and subsequent ImageNet object classification breakthroughs [19]. Shortly afterwards, similar systems achieved human-level accuracy also on ImageNet

[1] URL hidden to protect double blind review

[35, 36, 37, 15]. Generally speaking, the ImageNet dataset [6] 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 4), 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 pretrained on ImageNet, and alter it for the task of image segmentation or image detection [26, 33]. We expect that the same technique can be used when it comes to 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 [3, 26]. The task arguably is 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 [7] and MS-COCO [23].

In *DeepScores*, we provide ground truth for each pixel in all the images, having roughly 10^{12} labeled pixels in the dataset. In the next section, we compare these figures with existing datasets.

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 in the image. State-of-the-art algorithms are pipeline convolutional models, typically having combined cost functions for detection and classification [33, 25, 32]. The task can be combined with segmentation, which means that the algorithm is required to provide masks (instead of bounding boxes) for each of the objects in the image [14]. It differs from mere segmentation in the fact that the result shows which pixels together form an object. 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 so far, and highly challenging: the above-mentioned algorithms did not work well on our dataset in preliminary comprehensive experiments. We attribute this to the fact that most of the models used for object detection are fitted to datasets which have few but large objects. On the contrary, our dataset contains a lot of very small objects, which means that new models might need to be created in order to deal with it.

2.1. 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, in addition of having the highest resolution. In particular, images of *DeepScores* 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 1 contains quantitative comparisons of *DeepScores* with other datasets, while the following paragraphs bring in also qualitative aspects.

SVHN, the street view house numbers dataset [28], 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, with 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 millions of 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.

Other datasets

A number of other datasets have been released during the years, which have helped the progress of the field, and some of them have been used for different competitions. **MNIST** [21] is the first "large" dataset in the fields of machine learning and computer vision. It has tens of thousands of 28x28 pixels grayscale images, each containing a handwritten digit. The dataset is a solved classification problem and during the last decade has been used mostly for prototyping new models. Nowadays, this is changing, with more challenging datasets like **CIFAR-10/CIFAR-100** [18] be-

Dataset	#classes	#images	#objects	#pixels
MNIST	10	70k	70k	55m
CIFAR-10	10	60k	60k	61m
CIFAR-100	100	60k	60k	61m
Caltech-101	101	9k	9k	700m
Caltech-256	256	31k	31k	2b
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
CASIA online	**7356**	5090	1.35	nn
CASIA offline	**7356**	5090	1.35m	nn
GTSRB	43	50k	50k	nn
DeepScores	118	300k	**80m**	**1.5t**

Table 1: 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 used the SUN 2012 object detection specifications for SUN, and the statistics of ILSVRC 2014 [34] detection task for ImageNet.

ing preferred. Similar to MNIST, those datasets contain an object per image (32x32 color pixels), which do not make them ideal for more challenging problems like detection and segmentation.

Caltech-101/Caltech-256 [12] are more interesting datasets considering that both the resolution and the number of images are larger. Still, the images contain only a single object, making them only useful for the process of image classification. **SUN** [42] is a scene understanding dataset, containing over 100k images, each labeled with a single class.

The online and offline Chinese handwriting databases, **CASIA-OLHWDB** and **CASIA-HWDB** [24], were produced by $1'020$ writers using a digital pen on paper, such that both online and offline data were obtained. The samples include both isolated characters and handwritten texts (continuous scripts). Both datasets have millions of samples, separated into $7'356$ classes, making them far more interesting and challenging than digit datasets.

The German traffic sign recognition benchmark **(GTSRB)** is a multi-category classification competition held at IJCNN 2011 [16]. The corresponding dataset comprises a comprehensive collection of more than $50'000$ lifelike traffic sign images, reflecting the strong variations in visual appearance of signs due to distance, illumination, weather conditions, partial occlusions, and rotations. The dataset has 43 classes with unbalanced class frequencies.

2.2. 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 4).

Handwritten scores

The Handwritten Online Musical Symbols dataset **HOMS** [1] 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 [9] contains handwritten music images, which have been specially designed for writer identification and staff removal tasks. The database contains $1'000$ music sheets written by 50 different musicians with characteristic handwriting styles.

MUSICMA++ [13] 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** [2] 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.

Print quality scores

The **MuseScore Monophonic MusicXML Dataset** [40] is one of the largest OMR dataset to date, consisting of $7'000$ 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 dataset, 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 the 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 the modern deep learning algorithms.

[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 2: Statistical measures for the occurrences of symbols per musical sheet and per class (rounded).

While convolutional neural networks have been used before for OMR [41], *DeepScores* for the first time enables the training of very large and deep models.

3. The *DeepScores* dataset

3.1. Quantitative properties

DeepScores contains around $300'000$ pages of digitally rendered music scores and has ground truth for 118 different symbol classes. The number of labeled music symbol instances is roughly 80 million (4-5 orders of magnitudes 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 2 gives the mean, standard deviation, median, maximum and minimum number of symbols per page in the second column.

Another interesting aspect of *DeepScores* is the class distribution (see Figure 4). Obviously, some classes contain more symbols than other classes (see also Table 2, column 3). It can be seen that the average number of elements per class is $600k$ but the standard deviation is $4m$, illustrating that the distribution of symbols per class is very skewed.

Figure 3 visualizes the symbol classes together with their occurrence probability. The most common class is `noteheadBlack`, which provides slightly more than half of the symbols in the dataset. The top 10 classes are responsible for 86% of the musical symbols found.

3.2. 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 $5'000$ labeled image patches per class. This means we do not provide each of the 80m symbols as a single patch for classification purposes, but constrain the dataset for this simpler task to a random subset of reasonable size (see Section 4). The patches have a size of 45 x 170 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, there is an accompanying XML file for each image in *DeepScores*. The XML file has an `object` node for each symbol instance present on the page, which contains class and bounding box coordinates.

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 118 for ease of use in the softmax layer of potential models.

3.3. Dataset construction

DeepScores is constructed by synthesizing from a large collection of written music in a digital format: crowdsourced MusicXML files publicly available from MuseScore[3] and used by permission. The rendering of MuscXML with accompanying ground truth for the three flavors of granularity is done by a custom software using the SVG back-end of the open-source music engraving software LilyPond. The rendered SVG files not only contain all the musical symbols, but also additional tags that allow for identifying what musical symbol each SVG path belongs to.

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 5). Python scripts finally extract the three types of ground truth from this basis of SVG data and save the images as PNG using the CairoSVG library.

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 (`noteheadBlack`) 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 for everyone to define composite symbols in an application-dependent way.

4. Anticipated use and impact

4.1. Unique challenges

One of the key challenges this dataset poses upon modeling approaches is the sheer amount of objects on a single image. Two other properties of music notation impose challenges: First, there is a big variability in object size as can be seen for example in Figure 6. Second, music notation has the special feature that context matters: two objects having

[3] https://musescore.com

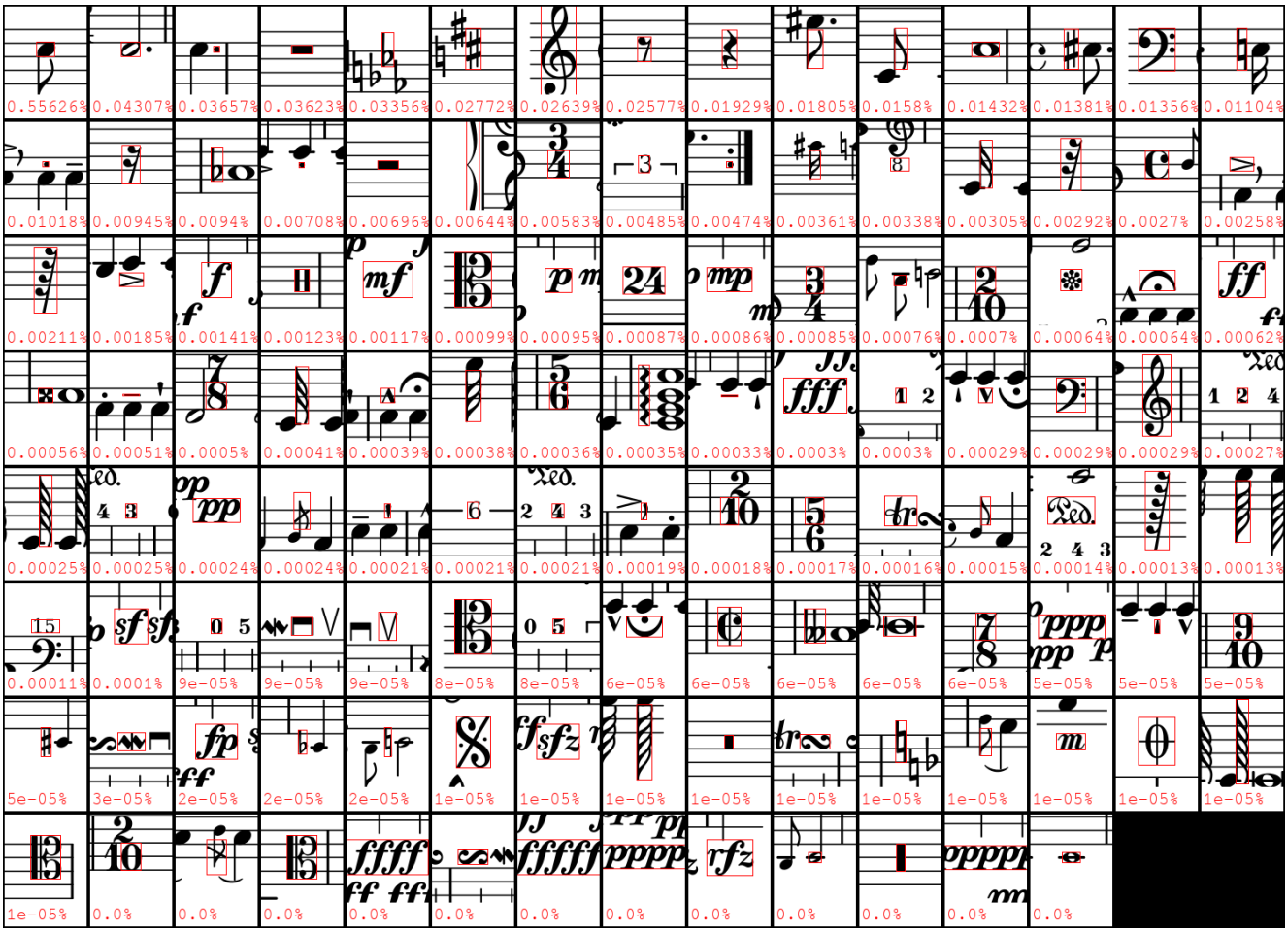


Figure 3: Examples for each of the 118 classes present in *DeepScores*, ordered by their frequency of occurrence. Even though the resolution is reduced for this plot, some of the larger symbols like `brace` (row 2, column 6) or `gClef` (row 1, column 7) are only shown partially to keep a fixed size for each patch. The symbols' full resolution in the dataset is such that the inter-line distance between two staff lines amounts to 20 pixels.

identical appearance can belong to a different class depending on the local surroundings (see Figure 7). To our knowledge there is no other freely available large scale dataset that shares this trait.

Moreover, datasets like ImageNet are close to being perfectly balanced, with the number of images/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.

4.2. Towards next-generation computer vision

Classifying the musical symbols in *DeepScores* should nevertheless not be a problem: all these symbols have very clear black and white borders, their shape has limited variability and they are rendered at very high resolution (see Figure 3). Due to these reasons we assumed that classification on *DeepScores* should be a relatively easy task, given CNNs usually deal well with these kinds of objects. To support this assumption, we fitted a simple residual-CNN [15] with 25 convolutional layers and about 8 million trainable parameters. Using the Adam optimizer [17] with the hyper-parameters proposed by the authors, we reached an accuracy of over 0.98 in just ten epochs. This shows that classification will indeed not be a big issue and CNNs are able to deal with labels that not only depend on an object but also its surroundings.

Detection, however, is more challenging: we evaluated SSD's and YOLO's fitness for the detection task on *DeepScores* and applied Faster R-CNN - with very little success. We conjecture that one of the main problems is that these

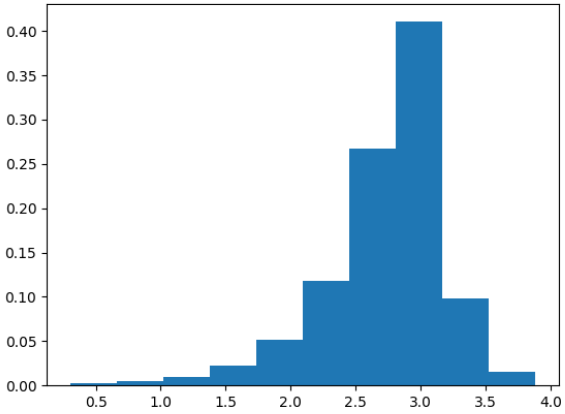


Figure 4: Histogram for the distribution of symbols over all images (logarithmic scale on abscissa, ordinate weighted to give unit area). The majority of images contain from 100 to 1000 objects.

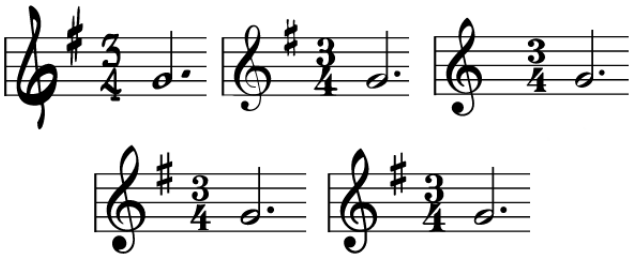


Figure 5: The same patch, rendered using five different fonts.

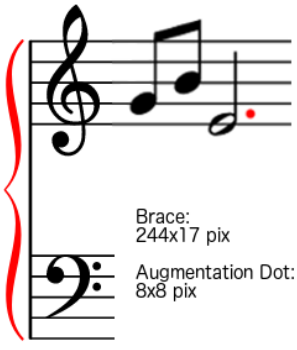


Figure 6: The possible size difference of objects in music notation, illustrated by `brace` and `augmentationDot`.

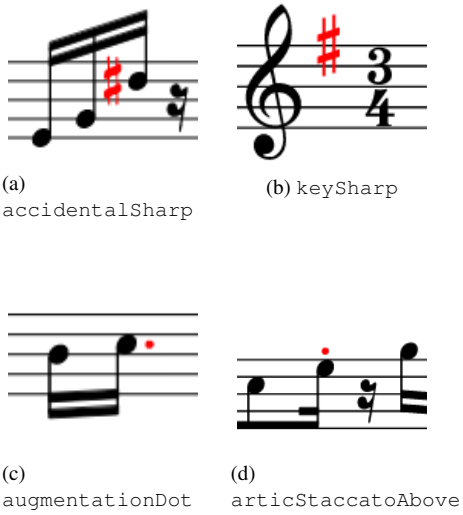


Figure 7: Examples for the importance of context for classifying musical symbols: in both rows, the class of otherwise similar looking objects changes depending on the surrounding objects.

region proposal-based systems seem to become computationally overwhelmed for this type of data, due to the sheer number of proposals necessary to find the many small objects.

Both observations - easy classification but challenging detection - lie at the heart of what we think makes *DeepScores* very useful: it offers the challenging scenario of many tiny objects that cannot be approached using current datasets (see Section 2). 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 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 objects and with images containing very small objects in them. This might be automated driving and other robotics use cases, medical applications with full-resolution imaging techniques as data sources, 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 [26, 33] 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. *DeepScores* will be of value

specifically in the area of OMR, but more generally to allow the development of algorithms that focus on the fine-grained structure of smaller objects while simultaneously being able to scale to many objects of that nature.

5. Conclusions

We have presented the conception and creation of *DeepScores* - the largest publicly and freely available dataset for computer vision applications in terms of image size and 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 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.

A weakness of the *DeepScores* dataset is that all the data is 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 [39, 38]. Future work *on* the dataset will include developing and publishing scripts to perturb the data in order to make it look more like real (scanned) written music, and evaluation of the transfer performance of models trained on *DeepScores*.

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 Herv Bitteur of Audiveris, the permission to use MuseScore data, and the collaboration with ScorePad GmbH.

References

[1] J. Calvo-Zaragoza and J. Oncina. Recognition of pen-based music notation: The homus dataset. *International Conference on Pattern Recognition, pp 3038-3043*, 2014.

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

[3] D. C. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. Neural networks for segmenting neuronal structures in EM stacks. In *ISBI Segmentation Challenge Competition: Abstracts*, 2012.

[4] D. C. Ciresan, U. Meier, J. Masci, and J. Schmidhuber. A committee of neural networks for traffic sign classification. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1918–1921, 2011.

[5] D. C. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition CVPR 2012*, June 2012. Long preprint arXiv:1202.2745v1 [cs.CV], Feb 2012.

[6] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. *In Computer Vision and Pattern Recognition, IEEE Conference on (pp. 248-255). IEEE.*, 2009.

[7] 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, 88(2), 303-338.*, 2010.

[8] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. *In Computer Vision and Pattern Recognition, IEEE Conference on.*, 2004.

[9] 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, Volume 15, Issue 3, pp 243-251*, 2012.

[10] K. Fukushima. Neocognitron: A self-organizing neural network for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, 1980.

[11] 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.

[12] G. Gregory, H. Alex, and P. Pietro. Caltech-256 object category dataset. *Technical Report - California Institute of Technology*, 2007.

[13] J. Hajic and P. Pecina. In search of a dataset for handwritten optical music recognition: Introducing muscima++. *arXiv:1703.04824*, 2017.

[14] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask r-cnn. *In Proceedings of the IEEE internation conference of computer vision*, 2017.

[15] 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 (pp. 770-778).*, 2016.

[16] J. S. C. I. J. Stallkamp, M. Schlipsing. The german traffic sign recognition benchmark: a multi-class classification competition. *Proc. 11th ICDAR*, 2011.

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

[18] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. 2009.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *In Advances in neural information processing systems (pp. 1097-1105).*, 2012.

[20] 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. *In Advances in neural information processing systems*, pages 396–404, 1990.

[21] Y. LeCun, C. Cortes, and C. J. Burges. The mnist database of handwritten digits, 1998.

[22] C. Lee and S. Osindero. Recursive recurrent nets with attention modeling for OCR in the wild. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2231–2239, 2016.

[23] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, and C. L. Zitnick. Microsoft coco: Common objects in context. *In European conference on computer vision (pp. 740-755). Springer, Cham.*, 2014.

[24] C.-L. Liu, F. Yin, D.-H. Wang, and Q.-F. Wang. Casia online and offline chinese handwriting databases. In *Proc. 11th ICDAR*, 2011.

[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. *In European conference on computer vision (pp. 21-37). Springer, Cham.*, 2016.

[26] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3431-3440).*, 2015.

[27] S. Mori, H. Nishida, and H. Yamada. *Optical character recognition*. John Wiley & Sons, Inc., 1999.

[28] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. *In NIPS workshop on deep learning and unsupervised feature learning (Vol. 2011, No. 2, p. 5)*, 2011.

[29] K.-S. Oh and K. Jung. GPU implementation of neural networks. *Pattern Recognition*, 37(6):1311–1314, 2004.

[30] R. Raina, A. Madhavan, and A. Y. Ng. Large-scale deep unsupervised learning using graphics processors. *In Proceedings of the 26th annual international conference on machine learning (pp. 873-880)*, 2009.

[31] A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. R. S. Marcal, C. Guedes, and J. S. Cardoso. Optical music recognition: state-of-the-art and open issues. *International Journal of Multimedia Information Retrieval*, 1(3):173–190, Oct 2012.

[32] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 779-788).*, 2016.

[33] 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 (pp. 91-99).*, 2014.

[34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[35] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556.* ., 2014.

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

[37] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).*, 2015.

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

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

(CVPR), 2011 IEEE Conference on (pp. 1521-1528). IEEE., 2011.

[40] E. van der Wel and K. Ullrich. Optical music recognition with convolutional sequence-to-sequence models. *arXiv:1707.04877*, 2017.

[41] E. van der Wel and K. Ullrich. Optical music recognition with convolutional sequence to-sequence models. *CoRR*, abs/1707.04877, 2017.

[42] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. *In Computer vision and pattern recognition (CVPR), 2010 IEEE conference on (pp. 3485-3492). IEEE.*, 2010.