Full-page music symbols recognition: state-of-the-art deep models comparison for handwritten and printed music scores

Ali Yesilkanat, Yann Soullard, Bertrand Coüasnon, Nathalie Girard

Full-page music symbols recognition: state-of-the-art deep models comparison for handwritten and printed music scores

Ali Yesilkanat[1], Yann Soullard[1,2], Bertrand Coüasnon[1], Nathalie Girard[1]
[1]Univ. Rennes, CNRS, IRISA
[2]LETG, UMR 6554, Univ. Rennes 2
[ali.yesilkanat, yann.soullard, bertrand.couasnon, nathalie.girard]@irisa.fr

Abstract

The localization and classification of musical symbols on scanned or digital music scores pose significant challenges in Optical Music Recognition, such as similar musical symbol categories and a large number of overlapping tiny musical symbols within high-resolution music scores. Recently, deep learning-based techniques show promising results in addressing these challenges by leveraging object detection models. However, unclear directions in training and evaluation approaches, such as inconsistency between usage of full-page or cropped images, handling image scores at full-page level in high-resolution, reporting results on only specific object categories, missing comprehensive analysis with recent state-of-the-art object detection methods, cause a lack of benchmarking and analyzing the impact of proposed methods in music object recognition. To address these issues, we perform intensive analysis with recent object detection models, exploring effective ways of handling high-resolution images on existing benchmarks. Our goal is to narrow the gap between object detection models designed for common objects and relatively small images compared to music scores, and the unique challenges of music score recognition in terms of object size and resolution. We achieve state-of-the-art results across mAP and Weighted mAP on two challenging datasets, namely DeepScoresV2 and the MUSCIMA++ datasets, by demonstrating the effectiveness of this approach in both printed and handwritten music scores.

1. Introduction

Optical Music Recognition (OMR) is a field of research that focuses on developing automated systems for recognizing and interpreting music scores from scanned or digital images. The localization and classification of musical symbols, known as music object recognition, represents a crucial and challenging component within the OMR pipeline. Deep learning-based techniques gather significant attention in this area, with initial successes achieved with CNNs [1, 2, 3].

In early deep learning-based approaches on music score recognition, hardware limitations necessitate the resizing of images to lower resolutions, resulting in information loss and reduced detection accuracy for small objects [4]. To address this, a common approach involves performing detection on overlapping cropped regions extracted from the original images, followed by fusion using methods like Non-Maximum Suppression (NMS) to eliminate duplicate detections in overlapping regions [5, 6].

However, using cropped images for musical object recognition presents several challenges, *e.g.,* some objects are partially cropped and lost. Also, during merging, it is challenging to establish an appropriate intersection-over-union (IoU) threshold for NMS, causing some objects may be erroneously duplicated and not effectively eliminated. While the issues related to cropping-based music object detection have been mentioned in the literature, no comprehensive analysis of these challenges has been conducted [5, 7].

In recent years, significant progress has been made in page-level music score recognition through the utilization of advanced object detector backbones that offer large-scale feature extraction capabilities, facilitated by increased computational resources [7, 8, 9]. However, some recent works have chosen to focus on subsets of classes, either due to the importance of certain classes for OMR [10, 11] or to address challenges related to small classes [7], leading to ambiguity in benchmarking and evaluation. Additionally, evaluation based on test sets created from cropped scores without considering merging further complicates comparisons [5, 7]. As a result, it becomes challenging to compare proposed methods without complete reproduction.

In this paper, we contribute to the field of music object recognition in the following ways: i) We demonstrate that full-page training outperforms cropping-based training in music object recognition at the full-page level evaluation,

providing a comprehensive analysis of both approaches. ii) We present a comprehensive analysis of state-of-the-art object detection models and backbones for music object detection in both handwritten and printed music. iii) We show the effect of resolution, cropping, and architecture on the performance of music object detection models. iv) We set a new benchmark on both printed and handwritten music recognition tasks by introducing FocalNet [12] backbones to the music score recognition task by utilizing Cascade R-CNN [13] detector, achieving state-of-the-art results on the DeepScoresV2 [8] and MUSCIMA++ [14] datasets.

Our research contributes to music object recognition and has broader implications for tasks detecting small objects, *e.g., aerial imagery [15]*, aiming to offer valuable insights and guidance to the computer vision community for addressing similar challenges across diverse domains.

2. Related Work

In recent years, music object detection has undergone a significant transformation driven by notable advancements in computer vision and deep learning fields. These advancements are facilitated by the availability of large annotated datasets, namely MUSCIMA++ [14] and DeepScores [8, 16], as well as increased computational power. As a result, Optical Music Recognition (OMR) research has made significant progress in music score recognition and interpretation.

Using Convolutional Neural Networks (CNNs) for image analysis leads to significant advancements in various domains, including OMR. This increases the development of diverse object detection algorithms, which are also suitable for the challenges of music object recognition. These algorithms are broadly categorized into two main categories: one-stage detection and two-stage detection approaches. One-stage detection models, including YOLO [17], SSD [18], and RetinaNet [19], directly generate category probabilities and coordinate positions of objects, resulting in faster detection speeds. On the other hand, two-stage detection algorithms, *e.g., Fast R-CNN [20], Faster R-CNN [21], and R-FCN [22]*, offer higher detection accuracy, although at a slower speed.

In the field of music object detection, researchers employ different object detection models on various datasets to detect scores. Zaragoza et al. [1] propose the first CNN-based staff detector and removal algorithm and, utilize CNNs for binarization and detection of a small symbol set from historical documents [2]. Hajič Jr and Pecina [3] employ Faster R-CNN for detecting noteheads on handwritten music scores. Pacha et al. [5] propose handwritten music object detection utilizing Faster R-CNN, R-FCN, and SSD on MUSCIMA++ dataset [14] by cropping scores. Zhang et al. [7] introduce staff-line removal and a modified YOLO V4 architecture for page-level handwritten music object recognition, but their reported results are based on 20 symbols from the MUSCIMA++ dataset. They also conduct custom training and testing on cropped segments from music scores, making direct comparisons impossible.

Pacha et al. [4] proposed a baseline method for music score detection on full-page level utilizing various object detection models, including Faster R-CNN, U-Net, and RetinaNet, and evaluated their performance on different datasets, including DeepScores and MUSCIMA++. Huang et al. [10] proposed a one-stage object detection network for OMR tasks using a dataset constructed from MuseScore dataset incorporating a feature fusion mechanism within the YOLO architecture. Hajič Jr et al. [23] employ a method that involves segmenting the input score image into a binary image using a semantic segmentation model. This binary image was then processed using a connected component detector. Tuggener et al. [9] introduced the Deep Watershed Detector, leveraging ResNets to predict dense energy maps and directly process the entire image without cropping each staff. While their method showed good performance on small symbols, challenges such as inaccurate bounding boxes and the detection of rare classes are encountered. Tuggener et al. [8] utilize HRNets in order to benefit from high-resolution representations as a backbone for Faster R-CNN on DeepScoresV2. Ru [11] utilize YOLO V4 to detect noteheads for chord detection on DeepScoresV2.

In conclusion, the integration of CNN-based object detectors provides significant advances in music object recognition, benefiting from annotated datasets and enhanced computing power. However, unresolved questions persist regarding training detectors using high-resolution full-page dense music scores, cropping versus full-page training choices, and the need for standardized benchmarks due to varied study approaches, which demand further investigation.

3. Method

Our goal is to locate and classify music symbols on both full-page high-resolution handwritten and printed scores. We conduct a thorough analysis of detectors and backbones for musical object detection, comparing their performance in detecting objects within large images. The objective is to uncover the strengths and limitations of these approaches, focusing on accurately detecting musical objects.

3.1. Object Detectors

Our work incorporates a diverse set of object detectors, including both CNN-based models, *e.g., Faster R-CNN and Cascade R-CNN*, and a transformer-based model, *e.g., DINO*. These detectors allow us to explore and analyze the strengths and capabilities of different architectures for music object detection.

Faster R-CNN [21] is a widely-used framework for detecting objects based on deep learning and consists of two components: a region proposal network (RPN) and a Fast R-CNN detector. The RPN is a fully convolutional network trained to generate region proposals for objects, while the Fast R-CNN network classifies the object. Both the RPN and the Fast R-CNN benefit from shared features trained alternatively for either task.

Cascade R-CNN [13] is an object detection framework that enhances detection accuracy by using a cascade structure with multiple stages. It builds upon Faster R-CNN and refines object proposals using a resampling procedure and multiple specialized regressors optimized on the resampled distributions of the different stages, improving the quality of object detection.

DINO [24] is an end-to-end transformer-based object detector. Extension of DETR [25], The proposed DINO model improves the training efficiency and the detection performance by using contrastive denoising training, look forward twice, and mixed query selection strategies. This increases success in detecting small objects with great accuracy.

3.2. Object Detection Backbones

We leverage state-of-the-art object detector backbones for feature extraction to enhance the accuracy and robustness of our music object detection framework. Specifically, we employ HRNet and Inception ResNet V2, focusing on feature extraction from high-resolution images, by following the work of Tuggener et al. [8], and Pacha et al. [4], respectively. We also propose to utilize the FocalNet and Swin Transformer in music score recognition for the first time. They demonstrate remarkable performance in COCO benchmark [26] when used as backbones [12, 24, 27].

Inception - ResNet V2 [28] is an influential backbone architecture widely used in object detection tasks. Combining the strengths of Inception and ResNet models, it utilizes parallel and residual connections to enhance feature representation and facilitate effective learning.

HRNet [29] has proven to be a highly effective backbone model when detecting small objects in large images. Its unique architecture maintains high-resolution representations throughout the network, enabling precise localization and improved feature extraction by leveraging multi-scale information and preserving fine-grained details.

Swin Transformer [27] is a versatile vision transformer with a hierarchical feature representation based on patch merging as the network deepens. A swin Transformer block is based on shifted windows where self-attention is computed within non-overlapping local windows while maintaining cross-window connections.

FocalNet [12] emerges as a powerful alternative to self-attention mechanisms from Transformers in computer vision. It shows superior performance in various computer vision tasks, including image classification, object detection, and segmentation, by employing a focal modulation mechanism instead of traditional self-attention methods while maintaining similar computational costs.

4. Experimental Setup

This section provides a comprehensive overview of the datasets used in our study, detailing their annotation setups and the implementation specifics of our experiments. Finally, we explain the evaluation metrics employed to measure the performance of our proposed methods.

4.1. Datasets

In our evaluation of the object detection models, we considered both the DeepScoresV2 dataset, comprising printed musical scores, and the MUSCIMA++ dataset, consisting of handwritten musical scores. By evaluating the best combination of models and components identified during the ablation study on the DeepScoresV2 dataset, we provide comprehensive results on the performance of the selected model on both printed and handwritten musical scores, enriching our understanding of the generalization capabilities across different music notation styles.

DeepScoresV2 [8] is a large artificial dataset for Common Western Modern Notation (CWMN), comprising 300,000 images with detailed annotations for symbol classification, image segmentation, and object detection tasks. A collection of MusicXML files sourced from MuseScore [30], the dataset is rendered into images using five unique fonts for visual diversity. The latest version, DeepScoresV2, includes complete annotations, covering essential symbols, *i.e., stems, beams, barlines, ledger lines, slurs*. Additionally, a *denser* version has been released, which includes 1,714 diverse and challenging images with annotations compatible with the MUSCIMA++ dataset.

In this study, we use the dense edition of the DeepScoresV2 dataset along with its MUSCIMA++ annotation set, which encompasses a diverse range of 72 categories. In order to prioritize the core objectives of OMR systems research and focus on the critical aspects of music object detection, we intentionally exclude 8 categories from our investigation: *beam, dynamicCrescendoHairpin, dynamicDiminuendoHairpin, slur, staff, stem, tremoloMark, tuple*. These categories, which can be efficiently identified using grammatical rules, are effectively addressed by existing tools in the OMR field [31]. Additionally, *accidentalDoubleFlat, numeral, graceNoteAcciaccatura* are not considered due to their absence in the test set.

We observe a disparity in the presence of *flag128s* between the images and the MUSCIMA++ annotation set, potentially causing confusion during flag detection. To

address this, we introduce two novel classes, flag128Up and flag128Down, by incorporating them into the MUSCIMA++ annotation set and mirroring their definitions from the DeepScores annotation set. This results in a unified annotation set with 63 categories, named Collabscore$_{63}$, used in our ablation study. However, to ensure comparability within the scientific community, we also present results using the DeepScores annotation set in 136 categories, using the best-performing architecture identified from the study.

The dataset encompasses distinct train and test splits, consisting of 1,362 and 352 images, respectively. In our ablation study, we construct a dedicated validation set by randomly removing 176 images from the training set, representing half the size of the test set.

MUSCIMA++ [14] dataset is comprised of 140 images that showcase handwritten music notation. It boasts a Music Notation Graph that features annotations, *i.e., bounding boxes, class labels, and image masks for all primitives*. This graph is also able to display the syntactic connections among primitives through directed edges.

MUSCIMA++ is an extension of the CVC-MUSCIMA [32] dataset, which holds 1,000 images from 20 musical compositions that are copied by 50 different musicians. Furthermore, we strictly follow the guidelines proposed in [14] for dataset partitioning. We use V1 and V2 annotation versions depending on their suitability for comparisons with state-of-the-art, containing 105 and 115 classes, respectively.

4.2. Implementation Details

In alignment with the DeepScoresV2 baseline [8], we use Faster R-CNN and HRNet model with the same configuration. The anchor generator in Cascade R-CNN follows the same specifications as Faster R-CNN. Images are resized to 3000x2000 pixels denoted as resolution Res$_{avg}$, which matches the average resolution of the dataset.

During the full-page training process on DeepScoresV2, Faster R-CNN and Cascade R-CNN detectors are trained using on-the-fly random cropped images of size 1000x500 pixels unless an alternative is explicitly specified. DINO training is performed without any cropping strategy, as we observe that DINO does not perform well when random cropping is used. To be able to fit the GPU memory during DINO trainings, images are resized to 0.75 times the original resolution 2250x1500, denoted as Res$_{small}$. It is important to mention that random cropping is only applied during training, while the entire music score is fed as input during inference without any cropping, which enables us to detect all musical symbols at the page level. Remarkably, we intentionally exclude any other form of augmentation or transformation during training to present the raw performance of the architectures without any additional enhancements.

Table 1: Full-page level evaluation of Faster R-CNN - Inception ResNet V2 architecture trained on MUSCIMA++ V1 with 105 classes using cropped images and full-page scores. Full-page training outperforms cropping-based training in full-page evaluation, in other words, score level evaluation. †: our reproduction

	Test On Cropped Scores				Test on Full-Page Scores			
	AP0.5		mAP		AP0.5		mAP	
Strategy	Mean	W. Mean	Mean	W. Mean	Mean	W. Mean	Mean	W. Mean
Cropping [5]	0.816	0.942	-	-	-	-	-	-
Cropping [5]†	0.803	0.944	0.576	0.668	0.736	0.928	0.540	0.661
Full-Page	-	-	-	-	**0.849**	**0.953**	**0.642**	**0.724**

In our full-page experiments on MUSCIMA++, the images are resized to 3500x2000 pixels, which is the average resolution of the dataset, and trained on random cropped images of size 1000x500 pixels, similar to DeepScoresV2 experiments. In our cropping experiments on MUSCIMA++, we follow the approach used by Pacha et al. [5]. We create the same cropped regions and annotations as they do and use the Faster R-CNN model with Inception-ResNet-V2 backbone, which they found to have the best mAP. We resize the input images to 580x350 pixels, which is the average resolution of the cropped images. To combine our detections to evaluate on full-page level, we use NMS with an IoU threshold of 0.8.

We conducted parallel training using four GPUs, specifically utilizing Nvidia A100 GPUs for all DINO architectures while employing Nvidia V100 GPUs for other tasks. All the DINO trainings employ a batch size of 1, while Faster R-CNN and Cascade R-CNN with HRNet and Inception ResNet V2 backbones use a batch size of 16. Cascade R-CNN with FocalNet and Swin Transformer backbones employ a batch size of 8. HRNet backbone is configured with V2p W32 configuration, while for FocalNet backbone base and tiny configurations are used and denoted as FocalNet$_B$ and FocalNet$_T$. In our experiments, we utilize the *SwinL* configuration for the Swin Transformer, while for DINO, we adopt the configuration that incorporates 5-scale feature maps. All backbones are pretrained on the Imagenet-1K [33], except Swin Transformer, which is pretrained on the Imagenet-22K [33].

The AdamW [34] optimizer with a learning rate of 10^{-4} is employed for training. We observe the weighted mean Average Precision (mAP) on the validation set and multiply the learning rate by 10^{-1} if there is no improvement for five consecutive epochs. The minimum achievable learning rate is set to 10^{-6}, and training is stopped if the weighted mAP of the validation set does not increase for the past eight epochs for stability.

4.3. Evaluation Metrics

In evaluating the object detection models, we use the Average Precision (AP) metric and follow both Pascal VOC [35] and COCO [26] evaluation protocols. The AP considers precision and recall to provide an overall measure of accuracy. To calculate mAP, as described in the COCO protocol, we use 10 predefined IoU thresholds ranging from 0.50 to 0.95 with an increment of 0.05, then take their average. Additionally, we calculate AP at an IoU threshold of 0.5, denoted as AP0.5, a widely accepted standard for assessing object localization in various music score recognition studies and used in Pascal VOC protocol. Finally, we calculate these metrics class-wise and report their mean and weighted mean values.

5. Results

In this section, we present the comprehensive results of our music object recognition experiments, focusing on both cropped and full-page images from the MUSCIMA++ and full-page images from DeepScoresV2 datasets.

5.1. Full-Page vs. Cropping-Based Training

We reproduce the results with Faster R-CNN - Inception ResNet V2 on cropped images, following the approach by [5] with the version of containing the staff lines. Following the work of [5], we used version V1 annotations of the MUSCIMA++, having 105 classes on the training set. Table 1 presents the full-page evaluation on MUSCIMA++ Test Set.

After verifying similar results on the cropped test set, we combined the detections and evaluated them against the original ground truths from the full-page test set. For the full-page case, we employed the same architecture for training on the entire page. The only configuration difference between full-page and cropping model architectures lies in the training resolution and random cropping during training, as detailed in Section 4.2.

The results demonstrate that even if the detector performs well on individual cropped images, the performance will degrade after merging. We observe that, after generating the cropped images, 3 classes are automatically lost on the training set. More than that, on the test set, 5 more classes also disappear. The reason for this is that the cropped regions are extracted by centering the staff lines. This means that objects that are far away from the staff lines are lost, *e.g., arpeggio_"wobble"*. Additionally, objects that are likely to be wider, *i.e., hairpin-cresc., hairpin-decr.*, are cut in half and therefore not included. To ensure a fair comparison with the full-page model, we set the APs to 0 for these classes, as the overall pipeline lacks the ability to learn to detect them.

To show the issues with lost or cut objects are not the only concerns, we additionally compute APs on the full-page using only the objects that appear in the cropped test set annotations after cropping by removing 5 mentioned classes above from the test set. This evaluation yields mean AP0.5 of 0.802, weighted mean AP0.5 of 0.937, mean mAP of 0.582, and weighted mean mAP of 0.667. Despite the improvement observed in these results compared to the model with cropped images in Table 1, it becomes apparent that the model trained at a full-page level consistently outperforms the model trained using cropped images during evaluation on full-page scenario. This highlights the benefit of having large contexts *i.e., receptive fields,* for music object detection.

Table 2: Comparison of Faster R-CNN and Cascade R-CNN detectors with Inception ResNet V2, HRNet, Swin Transformer (SwinL) and FocalNet$_B$ detectors on the DeepScoresV2 Test Set with Collabscore$_{63}$ annotation set.

	AP0.5		MAP	
ARCHITECTURE	MEAN	W. MEAN	MEAN	W. MEAN
Faster R-CNN - Inception R. V2	0.990	**0.994**	0.878	0.898
Faster R-CNN - HRNet	0.994	**0.994**	0.881	0.911
Cascade R-CNN - HRNet	0.993	0.992	0.920	0.939
Cascade R-CNN - SwinL	0.992	0.990	0.919	0.926
Cascade R-CNN - FocalNet$_B$	**0.996**	0.992	**0.929**	**0.940**
DINO - FocalNet$_B$	0.923	0.967	0.849	0.904

Table 3: Comparison of DINO on different resolutions and random cropping area on the DeepScoresV2 Test Set with Collabscore$_{63}$ annotation set. The results demonstrate that applying random cropping during training significantly decreases the detection success of DINO.

				AP0.5		MAP	
DETECTOR	BACKBONE	RESOLUTION	CROPPING	MEAN	W. MEAN	MEAN	W. MEAN
DINO	FocalNet$_B$	Res$_{Small}$	✗	0.923	0.967	0.849	0.904
DINO	FocalNet$_T$	Res$_{Small}$	✗	**0.924**	**0.968**	0.847	0.902
DINO$_S$	FocalNet$_T$	Res$_{Small}$	✗	0.918	0.967	0.840	0.901
DINO$_S$	FocalNet$_T$	Res$_{Avg}$	✗	0.914	**0.968**	**0.865**	**0.924**
DINO$_S$	FocalNet$_T$	Res$_{Avg}$	✓	0.642	0.736	0.468	0.478

5.2. Full-Page analysis

We explore music object detection at a full-page level.

5.2.1 Printed Music Object Detection

Table 2 provides a comprehensive analysis comparing the performance of Faster R-CNN, Cascade R-CNN, and DINO with HRNet and FocalNet$_B$ Backbones on DeepScoresV2 dataset with Collabscore$_{63}$ annotation set. Notably, Cascade R-CNN emerges as the top-performing detector, while FocalNet$_B$ stands out as the superior backbone. Cascade R-CNN benefits of multiple stages compared to Faster R-CNN. Combining Cascade R-CNN and FocalNet$_B$ yields the best overall results across various metrics.

Table 4: Performance of different cropping configurations on Cascade R-CNN - FocalNet$_B$ architecture on the DeepScoresV2 Test Set with Collabscore$_{63}$ annotation set.

	AP0.5		mAP	
Training Cropping Size	Mean	W. Mean	Mean	W. Mean
1000x500 pixels	**0.996**	0.992	0.929	0.940
1000x1000 pixels	0.989	0.989	**0.934**	**0.949**
1500x1000 pixels	0.992	**0.994**	0.916	0.938
2000x1000 pixels	0.990	0.990	0.920	0.940

Table 5: Comparison of the Impact of Targeted vs. Comprehensive Class Training using Cascade R-CNN - FocalNet$_B$ architecture on DeepScoresV2 on 3000x2000 pixels input resolution and 1000x500 random cropping during training.

		AP0.5		mAP	
Train Ann. Set	Test Ann. Set	Mean	W. Mean	Mean	W. Mean
Collabscore$_{63}$	Collabscore$_{63}$	**0.996**	**0.992**	**0.929**	**0.940**
DS$_{136}$	Collabscore$_{63}$	0.835	0.670	0.773	0.625

In Table 3, we present the results obtained from the DINO architecture and smaller DINO architecture, denoted as DINO$_S$, whose embedding and hidden dimensions are divided by half, and also illustrate the negative effects of employing random cropping during DINO training, highlighting a contrast with the performance of Cascade R-CNN and Faster R-CNN presented in Table 2. Our observations reveal that in our task, in contrast to common object detection benchmarks, characterized by detecting small and dense objects in high-resolution images, the sensitivity of positional encoding to spatial disruptions and scale variations is increased due to training on random cropped images and testing on full-page images strategy. Hence, we use the full score for training without random cropping, requiring significantly higher GPU memory. As a result, this prevents us from using the best backbone we found according to Table 2, FocalNet$_B$ with Res$_{avg}$.

Furthermore, our experiments reveal that the success rate decreases as the image resolution is reduced. This observation also suggests that if we can accommodate the GPU memory requirements of the original DINO detector with FocalNet$_B$ backbone by applying Res$_{avg}$ to the input, we may achieve significantly better results with DINO. To ensure a fair comparison and show the effect of the resolution, we utilize a smaller FocalNet$_T$ backbone when evaluating the DINO detector trained by Res$_{small}$ and Res$_{avg}$. Still, it can perform lower in our task, which can be related to insufficient data for a transformer to achieve convergence [36]. Until now, employing DINO as the detector is less appropriate than Cascade R-CNN on DeepScoresV2.

Table 4 presents the impact of various cropping resolutions on the performance. The evaluation was conducted using the Cascade R-CNN - FocalNet$_B$ architecture, which exhibited the highest Weighted mAP as shown in Table 2. In addition to the 1000x500 pixels cropping area mentioned in Table 2, we also incorporated 1000x1000 pixels, 1500x1000 pixels, and 2000x1000 pixels cropped regions in our analysis. Interestingly, our findings reveal that increasing the area of the random cropping does not yield a linear improvement in performance in the Cascade R-CNN detector. We also examine APs for individual symbols between 1000x500 pixels and 1000x1000 pixels, noting a general increase but no substantial improvement for any specific object.

Table 6: Evaluation on DeepScoresV2 with DeepScores Annotation Set (136 Classes) demonstrates Cascade R-CNN - FocalNet$_B$ outperforming other listed architectures at 3000x2000 pixels resolution with 1000x500 random cropping, achieving state-of-the-art results. Increasing input resolution and using optimal random cropping on training further improves performance. †: our reproduction. ⋆: 5500x4000 pixels input and 1000x1000 random cropping. DWD: Deep Watershed Detector.

	AP0.5		mAP	
Architecture	Mean	W. Mean	Mean	W. Mean
DWD - ResNet101 [9]	0.503	0.422	0.203	0.422
Faster R-CNN - Inception R. V2 [4]†	0.939	0.724	0.827	0.641
Faster R-CNN - HRNet [9]	0.799	0.676	0.700	0.608
Faster R-CNN - HRNet [9]†	0.946	0.726	0.828	0.651
Cascade R-CNN - FocalNet$_B$	0.977	0.725	0.902	0.679
Cascade R-CNN - FocalNet$_B$ ⋆	**0.981**	**0.729**	**0.940**	**0.700**

In Table 5, we highlight the advantages of training the music object detection pipeline with only the symbols we consider necessary to recognize, as opposed to training with all the symbols from the labeling. This narrowed training approach yields higher performance, demonstrating the importance of targeted symbol selection.

In order to establish comparability within the scientific community, we provide our findings on the DeepScores Annotation Set, which comprises 136 classes, in Table 6. Pacha et al. [4] propose utilizing Faster R-CNN - Inception ResNetV2 architecture for DeepScoresV1 as full-page detector; since we are working on DeepScoresV2, we employ the same architecture to evaluate and report results, allowing for direct comparisons with their approach. The achieved reproduction of Faster R-CNN - HRNet configuration surpasses the reported results [9], possibly due to our training meta parameters, detailed in Section 4.2. Once more, among the various architectures, the configuration of Cascade R-CNN - FocalNet$_B$ proves to be the most effective choice within this annotation set by obtaining state-of-the-art results on DeepScoresV2. Moreover, by employing the highest resolution along with optimal random cropping during training leads to improved outcomes.

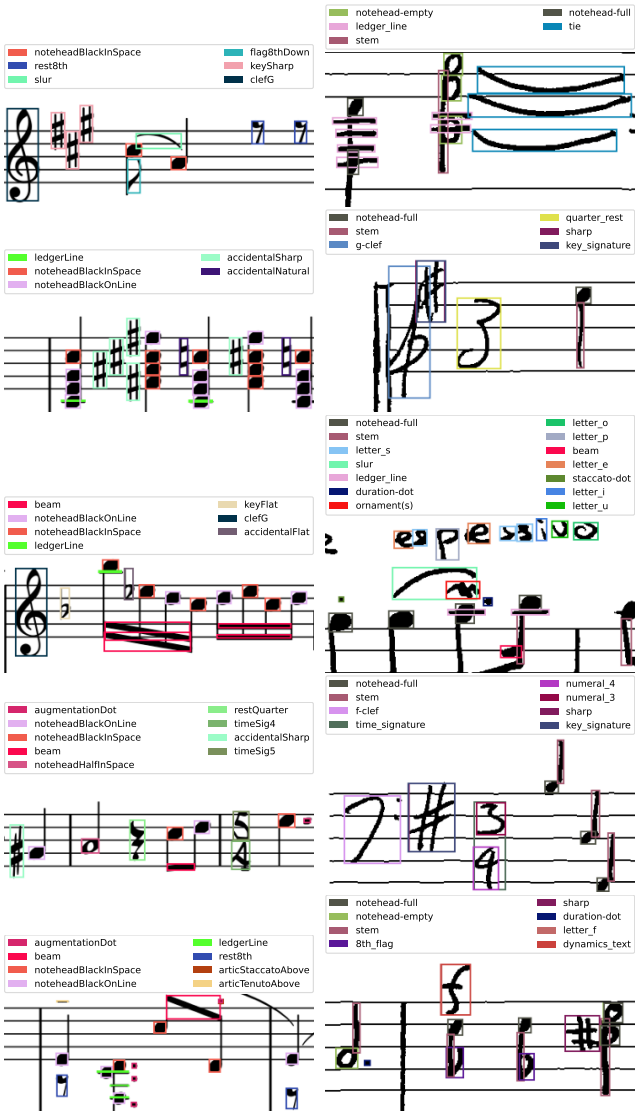
Table 7: Evaluation on MUSCIMA++ V2 Test Set demonstrates Cascade R-CNN - FocalNet$_B$ outperforming Faster R-CNN - Inception ResNet V2 at 3500x2000 pixels resolution with 1000x500 random cropping size, achieving state-of-the-art results. Increasing resolution and using optimal random cropping on training further improves performance. †: our reproduction. ⋆: 3500x2500 pixels input and 1000x1000 random crop.

	AP0.5		MAP	
ARCHITECTURE	MEAN	W. MEAN	MEAN	W. MEAN
Faster R-CNN - Inception R. V2 [4]†	0.849	0.953	0.642	0.724
Cascade R-CNN - FocalNet$_B$	**0.882**	**0.965**	0.787	0.789
Cascade R-CNN - FocalNet$_B$ ⋆	**0.882**	0.964	**0.790**	**0.793**

5.2.2 Handwritten Music Object Detection

We evaluate the Cascade R-CNN and FocalNet$_B$ combination on a handwritten music dataset. Table 7 showcases the results obtained using the architecture with existing approaches on the MUSCIMA++ dataset on V2 annotation set. To facilitate a fair comparison with these approaches, we utilize the MUSCIMA++ V2 annotation set. Pacha et al. [4] trained the Faster R-CNN with Inception ResNet V2 architecture at a lower resolution and obtain 0.039 mAP and 0.079 weighted mAP. In contrast, we employ the same architecture with the average resolution of the dataset, achieving higher results. Furthermore, the study by Shatri and Fazekas [37] reproduces this architecture on MUSCIMA++, further confirming the correctness of our chosen settings. These results solidify the prowess of the Cascade R-CNN - FocalNet architecture in handwritten music score recognition, as it demonstrates state-of-the-art performance across all annotated symbols within the MUSCIMA++ V2 dataset. Similar to the observations in Table 6, using the maximum resolution with an optimal random cropping strategy during training increases the detection performance.

Figure 1 shows sample detections from the test sets of both the DeepScoresV2 and MUSCIMA++ datasets. In the DeepScoresV2, objects like stems often pose challenges due to their narrow width, approaching a single pixel, making them difficult to detect accurately. Moreover, given our optimization of the detector for the Collabscore$_{63}$ annotation set, which also excludes stems, this outcome is expected. In contrast, MUSCIMA++ presents an opposite scenario. Despite stems having greater width due to handwritten nature, the inclusion of handwritten symbols introduces higher variability, leading to occasional detection failures.



(a) DeepScoresV2 (b) MUSCIMA++

Figure 1: Illustrating exemplary detection results achieved by the Cascade R-CNN - FocalNet$_B$ architectures on the DeepScoresV2 and MUSCIMA++ datasets. The showcased images are selectively cropped to ensure optimal readability of the detected elements.

6. Conclusion

In conclusion, our study provides a comprehensive analysis of three object detection models and four backbones for music object recognition on printed scores. We highlight the importance of page-level training and testing compared to cropping-based approaches. We also observe that the detection performance is improved by reducing and grouping music object categories. Nevertheless, specific objects, *e.g.*, *stems*, still pose a challenge, even with the most successful detectors. This emphasizes the need for future work integrating syntactic musical score recognition with object detectors to achieve a complete and functional optical music recognition system.

Our findings also reveal the superiority of the Cascade R-CNN model with the FocalNet$_B$ backbone, achieving state-

of-the-art results across multiple evaluation metrics for both printed and handwritten music scores on the DeepScoresV2 and MUSCIMA++ datasets. Despite promising results in general object detection, we show that Transformers, *e.g.*, DINO, does not perform as well as expected in detecting small objects in high-resolution images, such as in music score recognition.

This work not only contributes insights into music object recognition but also suggests promising directions for future research, including improving transformer-based detectors for small object detection, extending our approach to other music recognition datasets, and enhancing model robustness.

Acknowledgments

This work was performed using HPC resources from GENCI-IDRIS (Grant 2023-AD011012867R1) and funded by the French National Research Agency (ANR), under Grant ANR CollabScore ANR-20-CE27-0014.

References

[1] J. C. Zaragoza, A. Pertusa, J. Oncina, Staff-line detection and removal using a convolutional neural network, Mach. Vis. Appl. (2017). doi:10.1007/s00138-017-0844-4.

[2] J. C. Zaragoza, G. Vigliensoni, I. Fujinaga, A machine learning framework for the categorization of elements in images of musical documents, in: Proc. TENOR, 2017, pp. 17–23.

[3] J. Hajič Jr, P. Pecina, Detecting noteheads in handwritten scores with convnets and bounding box regression, arXiv preprint arXiv:1708.01806 (2017).

[4] A. Pacha, J. Hajič, J. Calvo-Zaragoza, A baseline for general music object detection with deep learning, Applied Sciences (2018). doi:10.3390/app8091488.

[5] A. Pacha, K.-Y. Choi, B. Coüasnon, Y. Ricquebourg, R. Zanibbi, H. Eidenberger, Handwritten music object detection: Open issues and baseline results, in: Proc. DAS, 2018, pp. 163–168.

[6] J. Calvo-Zaragoza, D. Rizo, End-to-end neural optical music recognition of monophonic scores, Applied Sciences (2018). doi:10.3390/app8040606.

[7] Y. Zhang, Z. Huang, Y. Zhang, K. Ren, A detector for page-level handwritten music object recognition based on deep learning, Neural Comput. Appl. (2023). doi:10.1007/s00521-023-08216-6.

[8] L. Tuggener, Y. P. Satyawan, A. Pacha, J. Schmidhuber, T. Stadelmann, The DeepScoresV2 dataset and benchmark for music object detection, in: Proc. ICPR, 2021, pp. 9188–9195.

[9] L. Tuggener, I. Elezi, J. Schmidhuber, T. Stadelmann, Deep watershed detector for music object recognition, in: Proc. ISMIR, 2018, pp. 271–278.

[10] Z. Huang, X. Jia, Y. Guo, State-of-the-art model for music object recognition with deep learning, Applied Sciences (2019). doi:10.3390/app9132645.

[11] Y. Ru, Computer assisted chord detection using deep learning and yolov4 neural network model, JPCS (2021). doi:10.1088/1742-6596/2083/4/042017.

[12] J. Yang, C. Li, X. Dai, J. Gao, Focal modulation networks, in: Proc. NeurIPS, 2022, pp. 4203–4217.

[13] Z. Cai, N. Vasconcelos, Cascade R-CNN: Delving into high quality object detection, in: Proc. CVPR, 2018, pp. 6154–6162.

[14] J. Hajič, P. Pecina, The MUSCIMA++ dataset for handwritten optical music recognition, in: Proc. ICDAR, 2017, pp. 39–46.

[15] G. Chen, H. Wang, K. Chen, Z. Li, Z. Song, Y. Liu, W. Chen, A. Knoll, A survey of the four pillars for small object detection: Multiscale representation, contextual information, super-resolution, and region proposal, TSMCS (2022). doi:10.1109/TSMC.2020.3005231.

[16] L. Tuggener, I. Elezi, J. Schmidhuber, M. Pelillo, T. Stadelmann, Deepscores-a dataset for segmentation, detection and classification of tiny objects, in: Proc. ICPR, 2018, pp. 3704–3709.

[17] J. Redmon, A. Farhadi, YOLO9000: Better, faster, stronger, in: Proc. CVPR, 2017, pp. 7263–7271.

[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, SSD: single shot multibox detector, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Proc. ECCV, 2016, pp. 21–37.

[19] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proc. ICCV, 2017, pp. 2980–2988.

[20] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proc. CVPR, 2014, pp. 580–587.

[21] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, IEEE TPAMI (2017) 1137–1149. doi:10.1109/TPAMI.2016.2577031.

[22] J. Dai, Y. Li, K. He, J. Sun, R-FCN: Object detection via region-based fully convolutional networks, in: Proc. NIPS, 2016, pp. 379–387.

[23] J. Hajič Jr, M. Dorfer, G. Widmer, P. Pecina, Towards full-pipeline handwritten OMR with musical symbol detection by U-Nets., in: Proc. ISMIR, 2018, pp. 225–232.

[24] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, H.-Y. Shum, DINO: Detr with improved denoising anchor boxes for end-to-end object detection, in: Proc. ICLR, 2023, pp. 7329–7338.

[25] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: Proc. ECCV, 2020, pp. 213–229.

[26] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: Common objects in context, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Proc. ECCV, 2014, pp. 740–755.

[27] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: Proc. ICCV, 2021, pp. 10012–10022.

[28] C. Szegedy, S. Ioffe, V. Vanhoucke, A. A. Alemi, Inception-v4, Inception-ResNet and the impact of residual connections on learning, in: Proc. AAAI, 2017, p. 4278–4284.

[29] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, B. Xiao, Deep high-resolution representation learning for visual recognition, TPAMI (2021). doi:10.1109/TPAMI.2020.2983686.

[30] MuseScore, Free music composition and notation software — musescore, `https://musescore.org`, 2023. (Accessed on 06/23/2023).

[31] B. Couasnon, Using a grammar for a reliable full score recognition system, in: Proc. ICMC, 1995, p. 187–194.

[32] A. Fornés, A. Dutta, A. Gordo, J. Lladós, CVC-MUSCIMA: a ground truth of handwritten music score images for writer identification and staff removal, IJDAR (2012). doi:10.1007/s10032-011-0168-2.

[33] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: Proc. CVPR, 2009, pp. 248–255.

[34] L. Ilya, H. Frank, et al., Decoupled weight decay regularization, in: Proc. ICLR, 2019.

[35] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge: A retrospective, IJCV (2015). doi:10.1007/s11263-014-0733-5.

[36] Y. Bai, J. Mei, A. L. Yuille, C. Xie, Are transformers more robust than CNNs?, in: Proc. NeurIPS, 2021, pp. 26831–26843.

[37] E. Shatri, G. Fazekas, DoReMi: First glance at a universal OMR dataset, arXiv preprint arXiv:2107.07786 (2021).