CNN-Based Accidental Detection in Dense Printed Piano Scores

Kwon-Young Choi, Bertrand Coüasnon, Yann Ricquebourg
Univ Rennes, CNRS, IRISA, F-35000
Rennes, France
{*kwon-young.choi, bertrand.couasnon, yann.ricquebourg*}*@irisa.fr*

Richard Zanibbi
Rochester Institute of Technology
Rochester, NY, USA
rlaz@cs.rit.edu

***Abstract*—The recognition of mid-18th to mid-20th century piano scores presents segmentation challenges caused by touching and broken symbols produced by imprinting techniques and time degradation. We present a new notehead accidental dataset containing 2955 images from dense and damaged piano scores. We address this detection problem with very small training samples using a simple Spatial Transformer (ST)-based Convolutional Neural Network detector improved through bootstrapping and contextual information, and more powerful deep learning detectors (Faster R-CNN, R-FCN, and SSD) with transfer-learning on the COCO dataset. We trained all our detectors using 5 fold cross-validation and obtain 98.73% mean Average Precision (mAP) for an Intersection over Union (IoU) threshold of 0.75 with our best detector. Our ST-based detector obtains a slightly lower mAP of 94.81%, but runs 40 times faster, and uses 18 times less memory.**

***Keywords*-Optical Music Recognition; Deep Learning; Symbol Detection; Data Augmentation**

I. Introduction

The recognition of mid-18th to mid-20th century dense and damaged piano scores presents unique segmentation problems of touching and broken music symbols as shown in Figure 2 due to their imprinting techniques and time degradation. Segmentation and classification of music symbols is an early task of the pipeline and should be highly precise and reliable because a segmentation or classification error could propagate and ruin the latest stages like music notation reconstruction. The segmentation of music symbols is the most challenging task because of the lack of previous work/datasets to test and compare approaches. With the introduction of recent deep learning architectures for object detection, we can now apply an end-to-end approach to segmentation and classification of music symbols. However, deep learning architectures generally need a lot of annotated training, which we do not have for old printed scores.

In this work, we consider the problem of detecting a single accidental symbol with the a priori knowledge of the position of the associated note head (Figure 1). We address this task using a new Spatial Transformer-based detector that is both small and fast, and compare this with three state-of-the-art object detectors . Our objective is to train these detectors with a limited number of annotated samples and to design methods general enough to be easily adapted for other symbols.

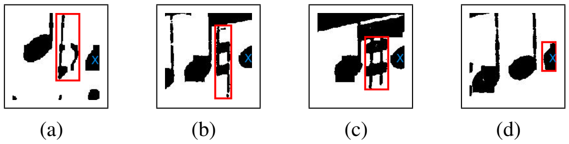


Figure 1. Task definition: detector should predict the position (red box) and class of an accidental (flat, natural, sharp or no accidental (rejection)) using raw image pixel and possibly the centroid of the note head (blue cross).

In summary, we describe the semi-automatic generation of ground-truth data for our task in Section III-A. We then present two ways of tackling the problem of training deep detectors with limited data. In Section III-B, we use a relatively simple detector with fast inference time but low out-of-the-box accuracy, and improve its results using bootstrapping techniques and contextual information. In Section III-C, we use three different state-of-the-art object detectors to produce highly precise accidental detectors using transfer learning. Finally, we evaluate and discuss the various tradeoffs of speed vs accuracy for different detectors in Section IV and Section IV-C, and conclude in Section V.

II. Related Work

In this Section we review the field of OMR with a focus on symbol detection and classification. We also present the grammar-based DMOS system for generating document recognition systems - we used this system to semi-automatically generate the dataset published with this work. Finally, we review the Spatial Transformer network and state-of-the-art detectors for object detection.

A. Optical Music Recognition

OMR studies by [1] or [2] typically present the OMR workflow as multiple consecutive stages: image preprocessing, staff detection with possible removal, music symbol segmentation/classification and finally music notation reconstruction. However, many works reorganize, merge or remove some of these stages.

Preprocessing and Staff Line Detection: Existing work in OMR tends to use common document pre-processing operations. Binarization is used to isolate connected components from the background, and often score pages are skew-corrected and have noise removal applied. Next, staff line detection has been performed using combinations of filters, pixel projection profiles, run-length analysis, contour-line tracking and graph path search. The height between two staff lines is an important feature in OMR, and this *interline distance* is estimated for later use. Recent work like [3] has used Convolutional Neural Networks (CNNs) to do pixel-wise classification to locate staff lines.

In this work, we use a graph-based Kalman filtering method [4] to detect and remove staves from the original image and is able to process broken or curved lines accurately. The symbol detection methods that we present are generally robust to remaining staff removal artifacts.

Symbol Detection: Music scores are constructed using a lot of relatively simple shapes like lines and blobs in a complex bi-dimensional structure. This fact has pushed OMR systems to use simple extraction algorithm like graphical primitive detection or connected components, and then use complex adhoc rules to merge or over-segment primitives [1]. The classification of music symbols can be done using a variety of techniques like simple filters, template matching or classifiers like HMM, neural network, K-NN and SVM as presented in [5].

More recently, convolutional-based neural network detectors [6] that merge the segmentation and classification steps have been applied to a variety of dataset like the newly annotated handwritten dataset of modern music, the MUSCIMA++ dataset [7] or on mensural music scores by [8]. Fully convolutional neural networks have also been used by [9] and [10] which allows for pixel wise segmentation of music symbols.

Reconstruction: Finally, the last step of the recognition process is to reconstruct the music notation and validate the structure produced. [2] shows that because of the strong structure and graphical rules of music notation, it makes sense to model this organization using a grammar. Most of these methods are used at the end of the OMR pipeline, to check the validity of the recognized music structure. One exception is the DMOS method described in Section II-B, where the grammar drives both the recognition and validation of the structure produced.

We believe that low-level symbol segmentation problems caused by the density, noise or pre-processing of a music score (see Figure 2) should not be part of the grammar for an OMR system, as it is too complex to be modeled explicitly. We wish to devise a method that delegates the segmentation task to a statistical model, in our case a Convolutional Neural Network (CNN) designed to do both segmentation and classification. We developed our method with the goal that it could be applied to any kind of structured document.

B. The DMOS Syntactical Method

The DMOS syntactical method was introduced by [11], and is a general off-line method for recognizing structured documents. The first version of the system included a grammar for musical scores. DMOS uses attributed two-dimensional grammars to define the symbolic and graphical representation of documents, producing constituent parse trees. The contextual information produced by the grammar can also be used to restrict the search space of our detector, as explained in Section III.

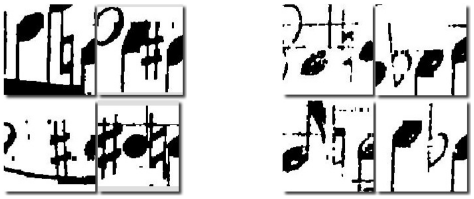

(a) Touching Symbols (b) Broken Symbols

Figure 2. Hard segmentation problems on accidentals

The hierarchical graphical structure produced, for example a simple music note as illustrated in Figure 3, is described by a set of rules that can search through the use of backtracking and check the coherence of different note elements. This ability to pinpoint inconsistencies can be used to efficiently produce semi-annotated data by reducing the amount of manual verification. Although the grammar is tailored to deal with complex polyphonic orchestral scores, segmentation had to be addressed using dedicated rules, which are difficult to produce and maintain. This detection of music symbol is the task we are proposing to resolve using Convolutional Neural Network-based detectors.

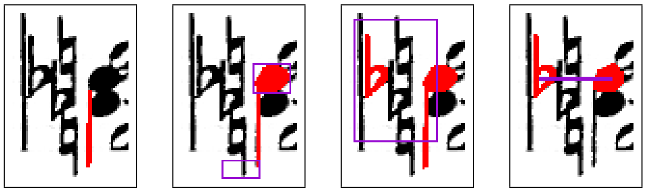

(a) Stem (b) Note head (c) Accidental (d) Alignement

Figure 3. Grammar workflow. Recognized elements are red. Violet squares are zones where recognizable elements are searched for. The construction of a musical note starts (a) find a potential stem, (b) two possible locations (top-right and bottom-left) are searched for a note head, (c) a potential accidental is searched at the left of the note head, (d) an alignment check is done between the note head and accidental.

C. Convolutional Based Object Detector

In this work, we use two different approaches to detect a single accidental using a small labeled training sample: 1) a novel Spatial Transformer-based network, and 2) state-of-the-art general object detectors (Faster R-CNN, R-FCN, and SSD) using transfer learning.

Spatial Transformer: First, we use a simple convolutional neural network (CNN) architecture based on Spatial Transformer (ST) networks proposed by [12]. The ST network is composed of two stacked CNNs: a localization network and a classification network. The localization network has the task to output a 2D affine transformation for a given input image. The Spatial Transformer Layer applies this transformation to the input image, that will be then fed to the classification network. In the original work of [12], the ST was intended as an attention model and not as a localization model. However, in our context of single symbol detection, we view this affine transformation as an opportunity to build a very simple music symbol detector.

Faster R-CNN: The Faster R-CNN [13] is one of the pioneer object detection architectures in Deep Learning and is now widely used in very diverse tasks. The detection process happens in two steps. First, in a Region Proposal Network (RPN) stage, a feature extractor (VGG-16 or resnet 101) is used to process input images. Then, at some intermediate layer of the feature extractor, anchor boxes are used in a sliding window manner to predict class agnostic box proposals. This RPN is trained using a multi-loss function taking into account both localization and objectness score produced by the RPN. Secondly, some of these proposals (usually 300) are cropped from the feature layer used to predict them, and the rest of the feature extractor is processed. Unlike the RPN stage, the second stage outputs class-specific bounding boxes refinement for each of the proposals. Finally, a similar multi-task loss is used to optimize the second stage detection.

R-FCN: The R-FCN detector proposed by [14] is an adaptation of the Faster R-CNN architecture designed for even faster detection. While the Faster R-CNN avoids a lot of computation by sharing a single network for both RPN and full detection stages, it still needs to process each region proposal until the end of the feature extractor. That is why the R-FCN architecture proposes to extract region proposals only at the last layer of the feature extractor and therefore reduces the amount of computation for each proposal. They also propose a position-sensitive cropping mechanism using position-sensitive score maps in order to retain the localization information for each proposed region. R-FCN is much faster than the Faster R-CNN, while maintaining comparable accuracy.

SSD: The third object detector we propose to use is the Single Shot Detector (SSD) [15]. Unlike the Faster R-CNN and R-FCN that use two stage predictions, the SSD architecture predicts directly class and bounding boxes of objects from a single pass of the feature extractor. This model is typically significantly faster than two stage detectors like Faster R-CNN and R-FCN.

III. Methodology

In Section III-A, we propose a new dataset for detecting accidental symbols. Using this dataset, we propose two main approaches of producing a CNN based detector with enough precision to be used in an OMR system: a contextual approach in Section III-B and an end-to-end approach in Section III-C. An overview of our different detectors is presented in Table II along with the main experimental conditions such as transfer learning from the COCO dataset, detection of a single object or multiple objects, use of the note head centroid as an input feature and use of bootstrapping to augment training data.

Table I
Dataset produced by the XXX pipeline driving a simple connected component based segmentation, a simple music symbol classifier and a manual check

No accidental (Reject)	Natural	Sharp	Flat	Total
968	968	777	242	2955

A. Dataset Construction

OMR surveys declare that the recognition of contemporary printed music scores is in practice already done by state-of-the-art OMR system. We believe that this statement does not concern more ancient printed music of the romantic and classical period, from around 1750 to 1950 period. These music scores are typically produced using engraving techniques that present very different graphical characteristics and segmentation problems as opposed to scores produced using computer software. That is why we propose a new small dataset for single accidental detection in dense and noisy piano scores[1].

We used three different scores from the composers Friedrich Kuhlau, Felix Mendelssohn, and Richard Wagner edited in the 19th century. The constitution of this dataset was semi-automated by using DMOS to analyze the layout of the score. Using the produced structure, we were able to extract potential locations of accidental by looking to the left of note heads. Connected components in the location were then classified using a simple CNN based classifier trained on isolated printed music symbols. Finally, we manually verified every potential accidental symbol, obtaining 2955 examples containing three accidental classes and a reject class (for when a note has no accidental), see Table I. In Figure 1, we show how we position our detection window, with the target notehead on the right side, using four times the size of the space between stafflines as estimated by DMOS (the *interline* distance) as the window side length.

Given the omnipresence of the target note head, we require that the detector localize the note head if there is no accidental associated with it. This allows us to define rejection using a concrete symbol detection goal, rather than trying to detect missing accidentals in background noise.

Bootstrapping Strategies: Having a small initial number of training examples, we use a translation-based data augmentation method, randomly moving the window framing the accidental (or note head for rejection). We propose four different variations shown in Figure 5. The Figure shows how we set boundaries on possible positions for randomly located windows. The *unconstrained* model in Figure 5a requires the accidental, or note head for rejection, to always be entirely in the image. We avoid introducing the

[1] https://www-intuidoc.irisa.fr/en/choi_accidentals/

Table II
Architecture and Data Usage for Accidental Detectors. The table shows whether detectors use transfer learning, can detect multiple objects, make use of the associated notehead location, or use bootstrapped samples. Version labels (v1, v2, . . .) differentiate different experimental conditions for the same detector and are reused in Table III.

Detector	Transfer Learn.	Mult. Objs	Note Head	Bootstrap
CNN + ST v1				
CNN + ST v2			✔	
CNN + ST v3				✔
CNN + ST v4			✔	✔
Faster R-CNN v1	✔	✔		
Faster R-CNN v2	✔	✔	✔	
Faster R-CNN v3	✔	✔		✔
R-FCN	✔	✔		
SSD	✔	✔		

vertical displacements not present in the original data using the *vertical* model in Figure 5b, where the window must be vertically centered around the centroid of the note head. We still allow a small range of 10 pixels of vertical variation.

The note head being a strong visual cue linked to the accidental, we propose a third generation model called *note_head* (see Figure 5c) where at least half of the current note head should always be inside the sampling window. Finally, we combined the *vertical* and *note_head* model constraints (the *vertical_note_head* model in Figure 5d). For each of these bootstrapping strategies, we augment the dataset in different quantities: 25k, 50k, 100k, 200k, 400k.

We also use this data augmentation opportunity to balance our dataset and over-sample less frequent classes like the flat and natural. Our previous use of the centroid position of the current note head can now be used to distinguish between multiple accidentals, and help the network to pick the right accidental to localize anywhere in the image.

B. Modified Spatial Transformer Detector

We modify a Spatial Transformer Network to construct an accidental detector as shown in Figure 4. We use only four parameters for affine transformations instead of the original six in [12] by zeroing out the two shearing parameters to produce axis-aligned bounding boxes. The main modification of the ST architecture is the forwarding of the affine transformation produced by the initial localization network to the new multi-task network that produces both classification and a localization correction for a symbol. This localization correction is added to the initial localization to produce the final detection. Localization and classification is learned jointly using a weighted multi-task loss 1 composed of a mean squared loss for the localization L_{reg} and categorical cross-entropy for classification L_{cls}. A weighting coefficient λ is used to normalize the localization loss with the classification loss.

$$L(t, t_{corr}, p) = L_{cls}(p, p^*) + \lambda \cdot L_{reg}(t + t_{corr}, t^*) \quad (1)$$

Here, p and p^* are respectively predicted class and ground-truth class. t, t_{corr} and t^* are respectively the initial transformation produced by the localization network, the transformation correction produced by the multi-task network and the ground-truth transformation.

Use Of Contextual Information: To this localization and multi-task network, we propose an improvement in order to use more contextual knowledge available during a typical OMR workflow. Knowing that the position of the note head is strongly correlated to the position of the accidental, we provide this coordinate as an input feature using two additional neurons in the first feed-forward layer of the localization network and multi-task network. Similarly, the affine transformation produced by the localization network is forwarded to the first feed-forward layer for the multi-task network to provide more contextual information.

We chose the origin to be the upper left corner of the window and normalize the coordinates using the size of the window, that is the bottom right corner has a coordinate of (1, 1). In the original training samples, every note head centroid will have the same coordinate (1, 0.5), because the window is positioned relative to the note head.

C. Multiple ROIs Object Detector Approach

Instead of specializing our detection model to the data we want to process, we show in this section the use of more complex and general CNN based detectors like the Faster R-CNN, R-FCN and SSD. These detectors typically extract multiple Region Of Interests (ROIs) from the input image and make either two step detection (Faster R-CNN and R-FCN) or single step detection (SSD). This strategy has the advantage of dramatically improving detection precision at the expense of adding a lot of computation, as discussed in Section IV-C. We use the official implementation of [16] that can be used to train and compare Faster R-CNN, R-FCN and SSD models. For the Faster R-CNN and R-FCN models, input images are typically scaled to 600 pixels on the shorter edge while keeping a maximum width or height of 1024 pixels. The SSD models only takes fixed size input images of 300x300 pixels. We propose to reuse our cropping strategy presented in III-A for the input of the network. This strategy produces relatively small images of around 130x130 pixels. However, we note that the up-sampling to the normal input size of the different object detector models does not deteriorate the image like a down-sampling strategy and it also allows us to use umodified Resnet 101 and MobileNet v1 feature extractors, without having to change ratio and scales of anchor boxes.

Faster R-CNN: Our objective is to build the most precise music symbol detector possible in order to minimize errors early in the OMR pipeline. That is why we use the Resnet 101 feature extractor that produces excellent accuracy while providing pre-trained weights on the Common Objects

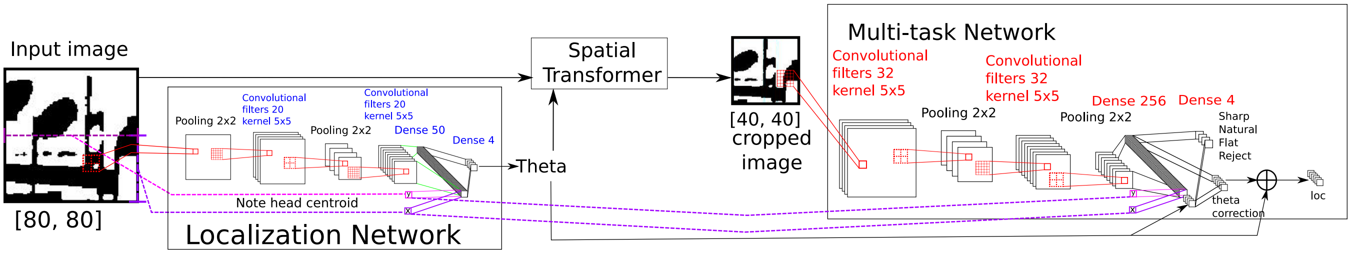
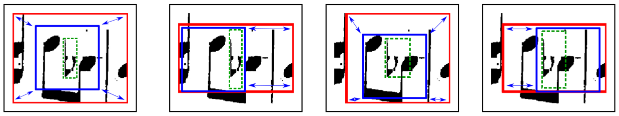


Figure 4. Accidental detection using a Spatial Transformer (ST). The localization network takes an 80x80 image as input, and produces an axis-aligned bounding box represented by an affine transformation in 4 outputs nodes. The sub-image in the bounding box, resized to 40x40 pixels, is then classified by the Multi-task network into four classes III-A. The Multi-task network refines the output of the Localization network by producing offset for theta. We train the whole architecture end-to-end using a weighted multi-task loss composed of a classification loss and localization loss.



(a) *unconstrained* (b) *vertical* (c) *notehead* (d) *v_notehead*

Figure 5. Four randomized sample bootstrapping techniques. The red square shows possible areas where the blue square, which is the sampling window, can be positioned. The green zone is always inside the blue sampling zone.

in Context (COCO) dataset for both Faster R-CNN and R-FCN models. The COCO dataset is a large object detection dataset containing around 200K images with 1.5 million object instances and is currently one of the major dataset used to train and evaluate object detection models. The ability of using pre-trained weights is essential because our dataset of 2955 examples is far too small to train these complex architectures from scratch. By using transfer learning, we can reduce over-fitting and benefit from the start of powerful feature extractors learned on the COCO dataset. It is therefore a way for us to reduce the amount of training data needed to produce an accurate detector.

Using the Faster R-CNN, we also propose to combine this complex object detector architecture with our previous propositions of bootstrapping and concatenation of contextual information (section III-B). We experiment using the best performing bootstrapping method, which is *vertical* as shown in Figure V, in order to augment the number of training samples. In combination with the bootstrapping method, which could lead to a confusion for the object detector of the correct accidental to localize, we concatenate the (x, y) coordinate information of the center of the note head to the first fully-connected layer after the crop and resize operation of the selected ROIs. Originally, the coordinates of the center of the note head are relative to the top left corner of the original image and scaled relatively. Because of the crop operation of ROIs, we duplicate the note head centroid position of one dataset example for each ROI and translate and rescale the coordinate relatively to the cropped area. The Faster R-CNN, R-FCN and SSD, are trained using the classical multi-loss function, combining a classification loss L_{cls} (Softmax) and localization loss L_{reg} (Smooth L1):

$$L(a_i, I) = L_{cls}(p_i, p_i^*) + \lambda \cdot [a_i > 0] \cdot L_{reg}(t_i, t_i^*) \qquad (2)$$

For each anchor a_i of image I, we search for the best matching predicted box t_i. If such a box exist, a_i is assigned to be *positive* and enable the localization loss L_{reg}. p_i and p_i^* are respectively the predicted class and the ground-truth class, t_i^* is the ground-truth bounding box associated with the anchor a_i. Here, $\lambda = 2$ meaning that the localization loss has twice as much weight as the classification loss. All other parameters are left to their default values.

R-FCN: R-FCN is similar to the Faster R-CNN except for in how the ROIs are computed and extracted from the feature extractor. This led to a significant speed-up as shown by [16] and our own results in section IV. The loss function is the same as the Faster R-CNN, see equation 2.

SSD: By doing detection in single step fashion, the Single Shot Detector is able to produce multiple detection with much faster speed than the Faster R-CNN and the R-FCN. We also use a different, more lightweight, feature extractor known as MobileNet v1. We resize all input images at 300x300 pixels as the model does not accept variable size input. No bootstrapping and contextual information was used for the R-FCN and SSD detectors. We use the same multi-task loss function as the Faster R-CNN, but use a Weighted sigmoid function for the classification loss. All parameters are left by default and use $\alpha = 1$ meaning that both classification and localization loss has the same weight.

D. Training Protocol

Contextual Bootstrapping Approach: The training of our Spatial Transformer architecture, shown in Figure 4, used in our contextual bootstrapping approach is done in a single end-to-end approach using a multi-task loss function composed of a categorical cross-entropy loss for classification and mean-squared error loss for localization. The normalization of the two losses is done by multiplying the localization. After a quick grid search for this parameter in {1,5,10,15,20}, the best results were obtained using a value of 20. The network is trained using the Adam backpropagation algorithm with a learning rate of 0.0001 and a batch size of 50.

Multiple ROIs Object Detector Approach: For training the Faster R-CNN, R-FCN and SSD models, we mainly reuse the recommended parameters by [16]. We chose to use pre-trained weights on the COCO dataset as mentioned before in section III-C for all feature extractors used: Resnet 101 and MobileNet v1. We train the Faster R-CNN and R-FCN with SGD configured with a learning rate of respectively 0.0001 and 0.0003. For the SSD, we use RMSProp with a learning rate of 0.004 and a batch size of 24.

Cross-Validation: Our dataset consists only of 2955 original images with very imbalanced classes. We do a 5 fold cross-validation in order to test our different approaches to produce reliable results. We implement this strategy by splitting the original dataset of 2955 images into 5 folds of ∼593 examples. We iterate 5 times and each time we choose a different fold to be the testing fold and use the remaining 4 folds for training. In the context of bootstrapping as seen in section III-B, we make sure that the data augmentation only operates on the training folds and happens only after the cross-validation splitting is done. That way, there are no possibilities that different bootstrapped images coming from the same original image are present in both training and test set. Using this cross-validation method, we therefore propose both the mean and standard deviation for every results presented in the next section.

IV. Results

Using the cross-validation protocol described in the previous section, we evaluate our detectors using the mean Average Precision (mAP) metric proposed by the PASCAL VOC Challenge in [17]. This metric allows us to jointly evaluate classification and localization accuracy and compare the impact of bootstrapping of our ST-based detector in Table V. We also compare with state-of-the-art detectors in Table III. However, we make two small modifications to this metric. The mAP metric uses an IoU threshold in order to decide if a detection is a True Positive (TP) or a False Positive (FP). It is common to use 0.5 as the IoU threshold for object detection. In our context of precise music symbol detection, we propose to add a second threshold of 0.75, which is much more strict and more representative of the level of precision we want to obtain. Also, rejection (i.e., the absence of any detection target) is not considered in the original mAP metric. That is why we propose to ignore the localization if the model correctly predict the input image to be a rejection (no accidental). Although, we define our rejection task to localize the note head, our objective is to give the network something stable to localize in order to simplify the rejection.

A. Detector Comparison

Using the mAP metric, Table III shows the performance of our different approaches. We can see that results are very good with an mAP of ∼99% with an IoU threshold of 0.5 except for our ST detector which only performs at ∼97%. Using an IoU threshold of 0.75 more clearly distinguish the detectors and place first the R-FCN with a mAP of 98.7%, then Faster R-CNN followed by SSD and finally the ST based detector. These results show the superiority of using multiple ROIs generated from different part of the images instead of a single ROI from the whole image.

Table III
Results comparing the best Spatial Transformer (ST) based detector, Faster R-CNN, R-FCN and SSD. Results shown are mAP (in %) with an IoU threshold of either 0.5 or 0.75. See Table II for an overview of each detectors.

Detectors	mAP with IoU >0.5		mAP with IoU >0.75	
	$\mu(\%)$	$\sigma(\%)$	$\mu(\%)$	$\sigma(\%)$
ST v4	97.25	1.68	94.81	2.99
Faster R-CNN v1	98.73	0.94	98.34	0.73
Faster R-CNN v2	98.85	0.67	98.65	0.59
Faster R-CNN v3	86.91	3.79	84.80	3.86
R-FCN	**99.17**	**0.30**	**98.73**	**0.40**
SSD	98.93	0.67	97.81	0.92

Table IV
Speed and memory consumption of the ST based detector, Faster R-CNN, R-FCN and SSD. Measures were taken on a Nvidia GPU K80.

Detectors	ST	SSD	R-FCN	Faster R-CNN
Speed (ms/image)	2	14	80	180
Memory (Mb)	260	300	4800	4800

However, more complex detectors come with additional overhead, as shown in Table IV. The Faster R-CNN is about 90 times slower than the ST based detector, and takes about 18 times more memory. Given that this detector will be intensively used by the OMR system (more than thousand of calls by page of music score), using a Faster R-CNN will provoke a significant slow-down of the recognition process.

B. Impact Of Bootstrapping And Contextual Information

In the case of the ST based detector, we found that the augmentation of training data almost always leads to better localization as shown in Table V. Another interesting observation is that different sampling strategies led to different results. The unconditional inclusion of the note head in the sampled image does not lead into an improvement, which seems to correlate with the property of translation invariance found in classical CNN architecture based on convolution and pooling operation. We also found that reducing the vertical displacement of the sampled images relatively to the vertical position of the note head lead to better results than allowing an unconstrained positional sampling. This seems to confirm our starting hypothesis that introducing variation in a very stable characteristic of our data does not help the ST based detector to converge.

In case of the use of the Faster R-CNN, we found that bootstrapping techniques actually hurt the precision of the detection. To better analyze this result, we divided the dataset into four categories: *single accidental* where only one accidental is visible in the image, *multi accidental* where multiple accidentals are visible, *reject without accidental* where no accidental are visible and finally *reject with accidental* where an accidental is visible in the image but should not be detected as it is not associated with the correct note head. We found that when using bootstrapping the results for both *multi accidental* and *reject with accidental* decreased significantly: 11% decreased for mAP with IoU > 0.5 and 17% decreased for mAP with IoU > 0.75. Our conclusion is that the architecture of the Faster R-CNN, designed for multi-object detection with strong translation invariance using the anchor boxes mechanism, is not suitable to be used in combination with bootstrapping for our particular task of contextual detection.

Again, for the ST based detector, we found that the use of contextual information like the centroid position of the current note head always helps the detector improve the detection results. In contrast, this add of information for the Faster R-CNN did not change anything to the results.

Finally, for the ST based detector, combining bootstrapping techniques and contextual information lead to an improvement of 9.3% mAP for an IoU threshold of 0.5 and 30.8% mAP for an IoU threshold of 0.75 (line 1 and 6 comparison in Table V). In the contrary, for more complex detectors like the Faster R-CNN, the use of contextual information or bootstrapping techniques did not improve the already very good results.

C. Discussion

After seeing the results of our experiments, we show the clear superiority of the Faster R-CNN and R-FCN for the task of detecting an accidental. However, we also propose less powerful models like the SSD and ST based detector for having more efficient and faster inference time for less accuracy. Also, if we consider using semi-supervised or unsupervised architecture in order to resolve the detection of music symbols, the Spatial Transformer should be simpler to integrate as it was designed as an attention model and integrates in any kind of neural network architectures.

In regards of the full OMR task, we only show here how to resolve a small subset of the pipeline. Future works will be oriented towards two main points: extend detection to other symbols, further reduce the number needed of training samples and propose a new corpus of dense and complex orchestral printed scores.

We plan to extend the detection of music symbols in a bottom up fashion, first adding multiple note heads detection and then gracefully integrate accidentals, followed by articulation marks, ornaments…

We also have done some preliminary experiments by further reducing the quantity of training samples to 1/5th of what is used in this work and the Faster R-CNN, R-FCN and SSD models show only a small degradation of the results. More investigation is needed in order to characterize the relation between the accuracy of the detectors relative to the number of training samples and the size of the window in which we want to detect a symbol. Our focus on reducing the number of needed training samples is based on the observation of the fact that manually annotating data is very slow and costly. Moreover, it has to be done again each time we work in a new corpus/type of documents.

Even though the situation in OMR recently improved by the introduction of the MUSCIMA++ dataset which contains localization, class and relationship information of music symbols in handwritten scores, the dataset is still extremely homogeneous because the corpus was originally designed for the staff lines removal task. We feel that the OMR community is neglecting printed scores because of recent software printed music scores with very good impression quality. However, there is still a huge quantity of music scores printed or engraved from the 18th to the early 20th century. These scores present a lot of challenges because of their printing techniques, time degradation, bad scanning qualities and complexity of the classic or romantic music style. While the dataset used in this work is available here [2], we will propose to the OMR community a new corpus of printed scores with the optic of automatically generating ground-truth data instead of manually annotating scores.

V. Conclusion

A renewed interest is shown toward OMR in the computer vision and pattern recognition research communities, because many interesting challenges remain to be overcome. In this work, we concentrated on designing a method that produces an accurate segmentation and classification of the accidental associated with a note head, without a priori rules concerning segmentation problems. We propose four different detectors: a Spatial Transformer based detector, SSD, R-FCN and Faster R-CNN. We show the tradeoff in speed over accuracy in different detectors with the best detector having 98.73% mAP for an IoU threshold of 0.75. The fastest ST based detector shows very bad out of the box performance. However, by using contextual information like the position of the note head and bootstrapping techniques, we improve significantly the accuracy of the detection by 9.3% mAP for an IoU threshold of 0.5 and 30.8% mAP for an IoU threshold of 0.75. Much more work is still needed in order to implement the whole OMR pipeline. In our future work, we are planning to further extend our detection models to other type of symbols like note heads, further reduce training data and propose a new corpus of printed music scores to the OMR community in order to research automatic ground-truth data generation.

References

[1] A. Fornés and G. Sánchez, “Analysis and Recognition of Music Scores,” in *Handbook of Document Image Processing and*

[2] https://www-intuidoc.irisa.fr/en/choi_accidentals/

Table V

Mean Average Precision results (in %) with IoU >0.5 or >0.75 (5-fold cross-valid., 2,955 samples) for the Spatial Transformer based detector, Figure 4. We explore the use of: *nh* which means the note head centroid is provided to the input of the network, different data *quantity*, different *bootstrapping methods* and different localization *loss weight*.

Spatial Transformer-based Detector Conditions				mAP with IoU >0.5		mAP with IoU >0.75	
nh	quantity	bootstrapping method	loss weight	μ(%)	σ(%)	μ(%)	σ(%)
✓	400k	vertical	20	**97.3**	1.7	**94.8**	3.0
✓	200k	vertical	20	96.0	2.4	92.0	2.8
✓	100k	vertical	20	94.6	3.5	86.1	6.3
✓	25k	vertical	20	92.9	3.1	68.0	1.5
✓	original		20	90.0	4.0	62.5	4.3
	original		20	88.0	3.6	64.0	3.7
	400k	vertical	20	94.3	3.5	86.2	5.9
✓	400k	vertical_note_head	20	96.0	**1.6**	92.2	**1.1**
✓	400k	unconstrained	20	94.4	2.1	92.4	2.3
✓	400k	note_head	20	95.7	1.9	88.4	2.5
✓	400k	vertical	10	96.6	1.9	93.8	2.2
✓	400k	vertical	1	95.6	1.9	89.7	3.5

Recognition, D. Doermann and K. Tombre, Eds. Springer London, 2014, pp. 749–774.

[2] 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*, vol. 1, no. 3, pp. 173–190, Mar. 2012.

[3] J. Calvo-Zaragoza, G. Vigliensoni, and I. Fujinaga, "Staff-Line Detection on Grayscale Images with Pixel Classification," in *SpringerLink*. Springer, Cham, Jun. 2017, pp. 279–286.

[4] V. P. d'Andecy, J. Camillerapp, and I. Leplumey, "Kalman filtering for segment detection: Application to music scores analysis," in *Proceedings of 12th International Conference on Pattern Recognition*, vol. 1, Oct. 1994, pp. 301–305 vol.1.

[5] A. Rebelo, G. Capela, and J. S. Cardoso, "Optical recognition of music symbols," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 13, no. 1, pp. 19–31, Nov. 2009.

[6] A. Pacha, K.-Y. Choi, B. Coüasnon, Y. Ricquebourg, R. Zanibbi, and H. Eidenberger, "Handwritten Music Object Detection: Open Issues and Baseline Results," in *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*. IEEE, Apr. 2018, pp. 163–168.

[7] J. Hajič and P. Pecina, "The MUSCIMA++ Dataset for Handwritten Optical Music Recognition," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, Nov. 2017, pp. 39–46.

[8] A. Pacha and J. Calvo-Zaragoza, "Optical Music Recognition in Mensural Notation with Region-Based Convolutional Neural Networks," in *ISMIR*, 2018.

[9] J. Hajic, M. Dorfer, G. Widmer, and P. Pecina, "Towards Full-Pipeline Handwritten OMR with Musical Symbol Detection by U-Nets," in *ISMIR*, 2018.

[10] L. Tuggener, I. Elezi, J. Schmidhuber, and T. Stadelmann, "Deep Watershed Detector for Music Object Recognition," in *19th International Society for Music Information Retrieval Conference, Paris, 23. - 27. September 2018*. Society for Music Information Retrieval, 2018.

[11] B. Coüasnon, "DMOS : A generic document recognition method, application to an automatic generator of musical scores, mathematical formulae and table structures recognition systems," in *Sixth International Conference on Document Analysis and Recognition, 2001. Proceedings*, 2001, pp. 215–220.

[12] M. Jaderberg, K. Simonyan, A. Zisserman, and k. kavukcuoglu, "Spatial Transformer Networks," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 2017–2025.

[13] 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 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 91–99.

[14] j. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 379–387.

[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," in *14th European Conference on Computer Vision*, vol. 9905, Amsterdam, 2016, pp. 21–37.

[16] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7310–7311.

[17] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010.