RESEARCH ARTICLE

Real World Music Object Recognition

Lukas Tuggener*†, Raphael Emberger*†, Adhiraj Ghosh*†, Pascal Sager*†, Yvan Putra Satyawan‡, Javier Montoya‡, Simon Goldschagg§, Florian Seibold§, Urs Gut¶, Philipp Ackermann||, Jürgen Schmidhuber**††, and Thilo Stadelmann† ‡‡

Abstract

We present solutions to two of the most pressing issues in contemporary optical music recognition (OMR). We improve recognition accuracy on low-quality, real-world (i.e. containing ageing, lighting, or dirt artefacts among others) input data and provide confidence-rated model outputs to enable efficient human post-processing. Specifically, we present (i) a sophisticated input augmentation scheme that can reduce the gap between sanitised benchmarks and realistic tasks through a combination of synthetic data and noisy perturbations of real-world documents; (ii) an adversarial discriminative domain adaptation method that can be employed to improve the performance of OMR systems on low-quality data; (iii) a combination of model ensembles and prediction fusion, which generates trustworthy confidence ratings for each prediction. We evaluate our contributions on a newly created test set consisting of manually annotated pages of varying real-world quality, sourced from International Music Score Library Project (*IMSLP*) / the Petrucci Music Library. With the presented data augmentation scheme, we achieve a doubling in detection performance from 36.0% to 73.3% on noisy real-world data compared to state-of-the-art training. This result is then combined with robust confidence ratings paving the way for OMR to be deployed in the real world. Additionally, we show the merits of unsupervised adversarial domain adaptation for OMR raising the 36.0% baseline to 48.9%.

All our code and data are freely available at: `https://github.com/raember/s2anet/tree/TISMIR_publication`.

Keywords: Optical Music Recognition, Deep Learning, Data Augmentation, Adversarial Training, Model Ensembles, Open Data

1. Introduction

Optical music recognition (OMR) (Rebelo et al., 2012; Calvo-Zaragoza et al., 2020) is a classical and challenging area of document analysis, that aims to convert images of written music to machine-readable, encoded form. A crucial component of any OMR pipeline is a music object recognition (MOR) system. In recent years MOR systems have reached greatly increased performance thanks to the adoption of deep learning (Pacha et al., 2018b; Tuggener et al., 2018b) and the availability of large datasets (Hajič and Pecina, 2017; Tuggener et al., 2018a, 2021).

Despite these advancements, we have identified two major roadblocks that hold current MOR systems back from reaching their full potential in a practical, real-world setting. Even though deep neural networks have been consistently revolutionising different computer vision tasks like classification, object detection, segmentation, image retrieval, and many more, they often fail to replicate the benchmark performances and results on new domains. This issue is attributed to *cross-domain mismatch*, some of the problems surrounding this issue are highlighted below.

Firstly, the currently available (MOR) training datasets are either synthetically generated or scans of very high quality, which are visually very close to synthetic imagery. This causes the resulting

*These authors contributed equally
†Centre for Artificial Intelligence, ZHAW School of Engineering, Zurich University of Applied Sciences, Winterthur, Switzerland
‡Work done while with the ZHAW School of Engineering
§ScorePad AG, Erlenbach, Switzerland
¶Work done while studying at the ZHAW School of Engineering
||Inst. of appl. Inform. Technology, ZHAW School of Engineering
**The Swiss AI Lab IDSIA, Lugano, Switzerland
††AI Initiative, KAUST, Thuwal, Saudi Arabia
‡‡Fellow, ECLT European Centre for Living Technology, Venice, Italy

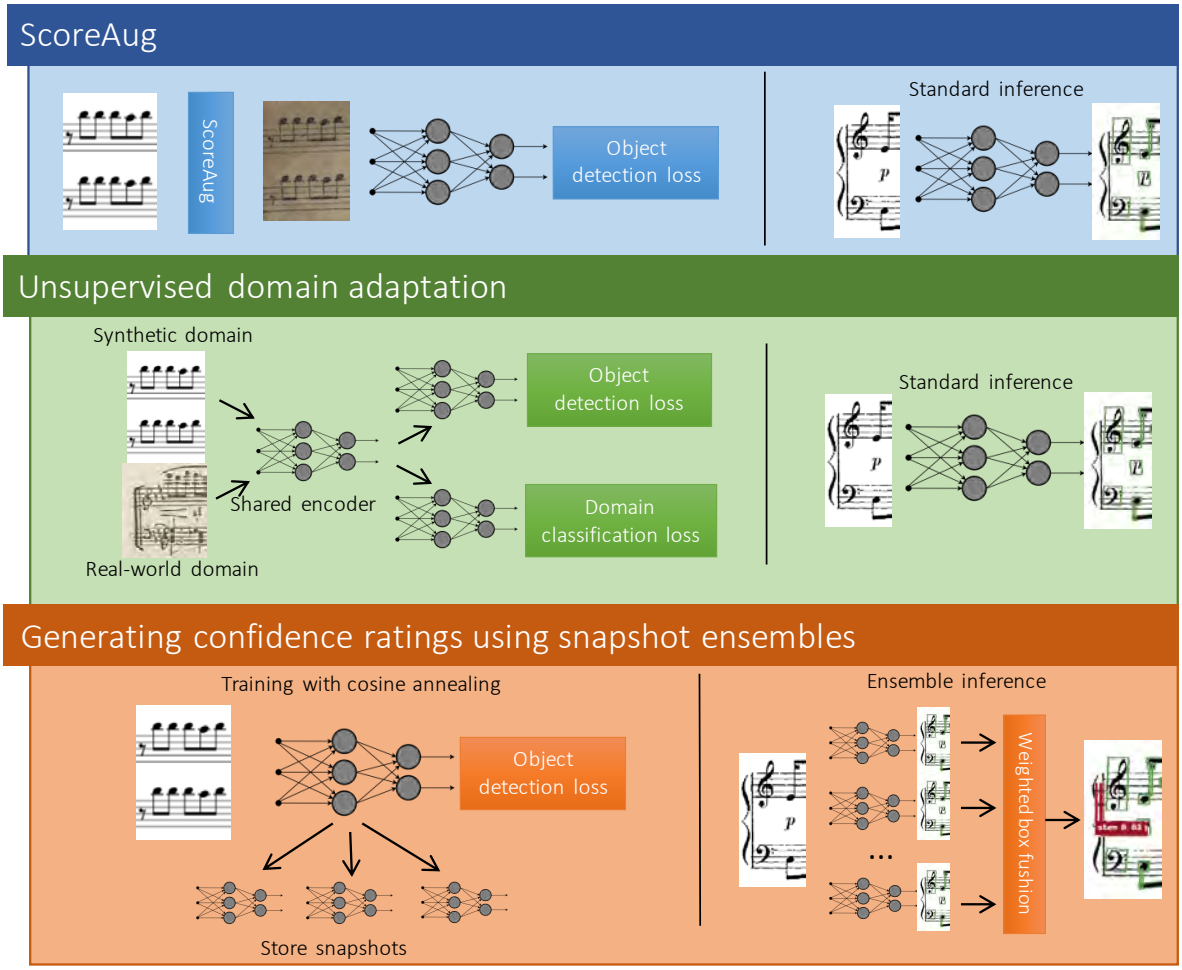

Figure 1: Graphical overview of the methods we contribute. *ScoreAug* (Section 5.1) in the top row, unsupervised domain adaptation (Section 5.2) in the center and snapshot-ensemble-based confidence ratings (Section 5.3) at the bottom.

detectors to perform very well on clean samples (Elezi et al., 2018; Pacha and Calvo-Zaragoza, 2018), but they struggle significantly when confronted with sub-optimal data quality as it is common in real-world applications (later referred to as real-world data). This can be scans of old or degraded pages, or for example, smartphone pictures under non-ideal conditions.

Secondly, deep neural networks are notoriously overconfident in their predictions – especially if an input lies outside the previously observed training data (Nguyen et al., 2015). This has great implications for the practical usability of MOR because it forces quality control, which is typically performed manually by humans, to check every detection with high diligence. This is particularly cumbersome for sheet music, where clusters of many tightly packed symbols are very common.

We attempt to resolve the issue of creating effective MOR systems for real-world musical sheet images by casting it as a *domain shift* problem. In order to bridge this domain gap between synthetic and realistic images, we propose two approaches: *ScoreAug* (Section 5.1), which creates augmentations for diversity in feature distributions during training and unsupervised domain adaptation (UDA) (Section 5.2): Training on one data distribution enables the model to also perform well on different target distributions.

ScoreAug uses real-world, scanned blank pages with natural signs of degradation and combines them with the synthetic input from our initial dataset. By mending those two together, realistic-looking samples can be created on-the-fly. The result of that operation is that the model generalises more to real-world data, thereby bridging the domain shift.

Unsupervised domain adaptation addresses the cross-domain mismatch issue by manipulating the target domain samples (in our experiments the real-world images). In our research, we select a *domain-adversarial loss* in order to enforce target (real-world data) embeddings to be similar or close to the source (synthetic data) embeddings in a latent feature space. We bridge the gap between domains without the need for the generative modelling

capabilities of adversarial models but with the help of a binary domain discriminator.

We address the problem of overconfident predictions by using an ensemble method. Good ensembles result when the predictions of the ensemble members are both accurate and have independent errors (Wen et al., 2020). Thus, the prediction confidence can be estimated over the predictions of several members and hence be better quantified.

We use SnapshotEnsemble (Huang et al., 2017), a method that creates ensemble members at no additional training cost. At inference time, each ensemble member makes a prediction independent of the other members. These predictions are fused into an average prediction with higher accuracy and more reliable confidence ratings and thus facilitate subsequent quality control.

In summary, the contributions of this paper are as follows (c.f. Figure 1):

- A MOR test dataset (*RealScores*, see Section 4) that contains 14 pages of real-world sheet music. The annotations for this data were newly created by hand and follow the class definitions and data structure of *DeepScoresV2* (Tuggener et al., 2021);
- *ScoreAug* (see Section 5.1): A sophisticated data augmentation scheme and training schedule using a combination of synthetic data (*DeepScoresV2*) and perturbations sourced from a diverse array of real-world documents (*IMSLP*), which are combined using randomised heuristics;
- An adversarial discriminative method for implementing unsupervised domain adaptation (see Section 5.2) in MOR for finding indiscriminate representations for the distributions of synthetic (*DeepScoresV2*) and real-world (*IMSLP*) features and bridging the gap between these domains in a latent feature space;
- Trustworthy confidence ratings (Section 5.3) for symbol level detections based on a prediction fusion algorithm that utilises confidence scores of ensemble outputs to calculate average predictions and confidence ratings.

The rest of this paper is organised as follows: Section 2 gives a thorough introduction to making OMR more robust in practice by surveying related work. In Section 3 we introduce our baseline model on which all of our solutions are built. Section 4 presents the *RealScores* dataset, a newly sourced and annotated small test set for real-world MOR. In Section 5 we present our proposed methods. Section 6 contains descriptions and results of all our experiments. Lastly, in Section 7 we draw conclusions and discuss possible future work.

2. Survey of Related Work

Music Object Detection Traditionally, OMR systems consisted of a cascade of components such as staff-line removal (Fujinaga, 2004; Dalitz et al., 2008), symbol segmentation (Bellini et al., 2001) and symbol classification (Toyama et al., 2006), which were built using classical computer vision methods. With the advent of increased computing power and the availability of large-scale datasets (Hajič and Pecina, 2017; Tuggener et al., 2021), deep-learning-based approaches (Schmidhuber, 2015) started to take over. Deep learning methods resulted in greatly increased performances in the above-mentioned tasks (Gallego and Calvo-Zaragoza, 2017). More recent works apply convolutional neural networks directly to the raw input data, making multi-step designs obsolete (Pacha et al., 2018a; Hajic Jr et al., 2018; Tuggener et al., 2018b). There are efforts to solve the whole OMR problem in one single step, as it is state of the art in related fields such as text (Chowdhury and Vig, 2018) or speech (Chiu et al., 2018) recognition. However, due to the high complexity of music notation, all existing solutions focus on a simplified problem such as mensural notation (Pugin, 2006) or monophonic scores, both typeset (van der Wel and Ullrich, 2017; Calvo-Zaragoza and Rizo, 2018b) and handwritten (Baró et al., 2018).

Input Data Augmentation Input data augmentation has a rich history in deep learning. However, it is mostly used to improve performance on a single domain. Typically, data augmentation consists of scaling, translations, and rotations Ciregan et al. (2012); Sato et al. (2015). On larger natural datasets such as ImageNet more sophisticated transforms like random cropping, image flipping, and colour normalization have become commonplace Krizhevsky et al. (2017). Generative adversarial networks have been employed to generate additional realistic training data Zhu et al. (2017). Recently, automatic search of optimal augmentation strategies on a per dataset basis has become the standard Cubuk et al. (2019).

There has already been some effort to address the domain gap between existing datasets and real-world data using data augmentation in the context of MOR. Datasets that have been altered to mimic realistic data have been created, either by applying a sequence of graphics filters (Calvo-Zaragoza and Rizo, 2018a) or by printing and scanning the data (Elezi et al., 2018).

To the best of our knowledge, this is the first work to present an input augmentation technique, that combines algorithmic distortions with real-world perturbations for MOR.

Domain Adaptation for Object Detection UDA is an unsupervised learning approach (Simmler et al., 2021) to transfer knowledge obtained from a source

domain with labelled data to a target domain with unlabelled data. One of the fundamental approaches in UDA was proposed by Tzeng et al. (2017), creating a generalised framework for adversarial adaptation in image classification.

Recently, UDA methods for tasks outside classification, such as object detection, have attracted increasing attention, which is the primary focus of our OMR models. Chen et al. (2018) was one of the pioneering works on this task. The authors observed image and instance level shifts and proposed segregated components to alleviate the domain discrepancy.

Adversarial approaches for discriminative UDA have recently reflected strong results in object detection (Zhu et al., 2019; Lehner et al., 2022; Li et al., 2022). The primary goal of most of the adversarial approaches addressed above is adversarial feature alignment between the source and target domain.

In the context of MOR, Mateiu et al. (2019) employed a domain adversarial neural network (Ganin and Lempitsky, 2015) to enable the classification of individual handwritten symbols in old music manuscripts. Castellanos et al. (2021) use UDA to improve document analysis (splitting of the input in a layered version containing different information, e.g. staffs, notes or background) on historical music sheets.

To the best of our knowledge, this is the first work to employ and systematically evaluate UDA techniques for a full-fledged MOR system.

Confidence Ratings Most state-of-the-art approaches to estimate predictive uncertainty rely on ensembles (Gustafsson et al., 2020; Wen et al., 2020; von Oswald et al., 2021; Wenzel et al., 2020; Durasov et al., 2021; Xia et al., 2021; Huang et al., 2017). Bayesian deep learning approaches like MC-dropout have interesting properties but fail to deliver in practice due to computational or technical constraints (Dürr et al., 2020).

Since we train our models for 1000+ epochs and the input images are large (i.e. require a lot of memory), we focus on approaches known as "economic ensembles", such as HypernetEnsembles von Oswald et al. (2021) or Masksembles Durasov et al. (2021). For these methods, the computational and memory costs do not increase linearly with the number of ensemble members and thus scale well with large deep learning models.

SnapshotEnsemble was proposed by Huang et al. (2017) and tries to achieve the seemingly paradoxical goal of producing an ensemble at no additional training cost. Their method leverages work on cyclic learning rate schedules (Smith, 2017). They lower the learning rate at a very fast pace, thus encouraging the model to converge quickly to its first local minimum.

***DeepScoresV2* dataset**	
Model	AP (overlap = 0.50)
Baseline model	89.3%
DWD	50.3%
Faster R-CNN	79.9%

Table 1: The AP at 0.5 overlap for our baseline model and two state-of-the-art models (DWD, Faster R-CNN Tuggener et al. (2021)) on *DeepScoresV2*.

Then the optimisation is continued with a higher learning rate to dislodge the model from this local minimum again. This procedure is repeated multiple times. At each local minimum, the model is saved (i.e. a snapshot is taken). Ensembling the snapshots result in consistently lower error rates than single models. In this work, we exclusively employ SnapshotEnsembles due to their minimal compute requirements.

To the best of our knowledge, this is the first work to employ uncertainty measures in the context of OMR and systematically evaluate their merits.

3. Baseline Model

All our experiments are based on the *S^2A-Net* architecture (Han et al., 2021), which allows for oriented detections unlike earlier methods (Tuggener et al., 2018b; Pacha et al., 2018b). The *S^2A-Net* is an anchor-based object detector that uses a single-shot alignment network to generate accurately oriented object detections. Its novel feature alignment and oriented detection modules are fed using a ResNet-based backbone (He et al., 2016) and feature-pyramid networks (Lin et al., 2017).

We achieve good results on the "oriented mode" of *DeepScoresV2* (Tuggener et al., 2021) when training *S^2A-Net*s by scaling the data with a factor of 0.5 and then using random crops of 1000 by 1000 pixels. We are able to conserve GPU memory whilst keeping high precision by just using a singular anchor ratio of 1.0 and a singular anchor scale of 4. We train our models with SGD using a learning rate of $\alpha = 2.5 \cdot 10^{-3}$ and a momentum of 0.9. Table 1 contains the average precision (AP) of our baseline model against two state-of-the-art models, which illustrates the competitive performance of our new *S^2A-Net* based approach. The complete training details can be found in the published code.

4. The *RealScores* Data

So far, no real-world test data is available to benchmark models on. Such data is crucial to observe how well our models will perform when facing a domain gap. To create a benchmark dataset for real-world OMR, we sourced digitised music scores from the International Music Score Library Project

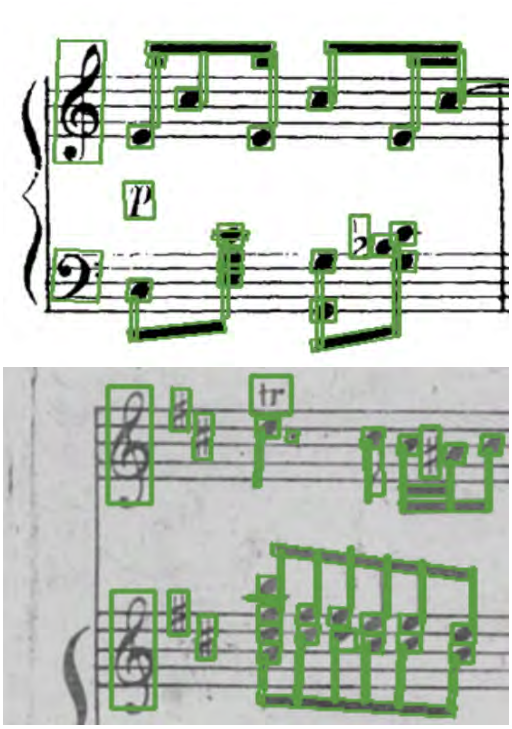

Figure 2: Example snippets from two *RealScores* pages with ground truth annotations overlayed.

(*IMSLP*) / Petrucci Music Library[1]. Of the downloaded music scores, only those with specific characteristics were considered for the new test set: The sheets had to be scans or photographs of music scores and be visibly non-synthetic, meaning that they come with scanning artefacts, discolourations, stains, folds, be angled, and have other imperfections. Music scores that were handwritten, of very bad quality, or using non-standard notation were considered out of scope for this work. The selected samples had to be annotated by hand using ScorePad's current OMR pipeline (Stadelmann et al., 2018). The resulting test set consists of 12 music sheets with a total of 12553 annotations that we name *RealScores*. The annotations are stored in the same format that *DeepScoresV2* (Tuggener et al., 2021) introduced. Due to its limited size, only 61 of the original 136 classes are present. Excerpts from two samples with their corresponding annotations are visible in Figure 2.

In a second step, we sourced a number of "blank" pages from the aforementioned Petrucci Music Library. This was possible because many uploaded music scores would be sourced from completely scanned books, including the front and back covers. Such scans sometimes contain blank pages without any written music, but all the perturbations that naturally occur on sheets of paper. This is a valuable source of real-world noise that can be overlayed with synthetic data. We manually looked through the sourced data for suitable blanks, then converted them into pictures and normalised their size to fit with the synthetic data of *DeepScoresV2*. A total of 51 such blank pages were selected – 30 of which have a significant portion of the sheet border visible, and 21 do not. Figure 3 shows six of those pages.

[1] Petrucci Music Library (*IMSLP*): https://imslp.org/wiki/Main_Page

Figure 3: Example blank pages.

5. Methods

In this section, we present our proposed methods to address domain shift and overconfidence. In Section 5.1 we propose a powerful data augmentation method, in Section 5.2 we present an alternative solution based on UDA, and in Section 5.3 we describe our scheme to produce confidence ratings.

5.1 Input Data Augmentation

We propose a sophisticated data augmentation scheme to address the domain gap that we call *ScoreAug*. With *ScoreAug*, input samples first can be blurred, get salt-and-pepper-like noise, get irregular edges in the border area, be rotated by a small angle, or become augmented with other irregularities not found in a synthetic dataset like *DeepScoresV2*. Additionally, we go one step beyond these algorithmic perturbations

	Blanks	Scores
Salt and Pepper Noise	-	P_{snp}
No additional Augmentations	-	P_{aug}
Horizontal Flip	50%	-
Vertical Flip	50%	-
Crop and Resize	20%	-
Randomise Brightness	50%	-
Higher Contrast	-	20%
Small Angle Rotation	60%	60%
Additional Brightness	-	40%
Gaussian Blur	-	P_{blur}

Table 2: Probabilities of augmentations as part of *ScoreAug* that can be applied to either the blanks, synthetic scores, or both at the same time. Note that P_{aug} decides how likely any other augmentations (after the salt and pepper noise) will be applied, in order to not only feed *ScoreAug*mented samples to the model. Our final model uses $P_{\text{snp}} = 0\%$, $P_{\text{aug}} = 30\%$, $P_{\text{blur}} = 10\%$.

and complement them by overlaying them on our blank pages from the *RealScores* dataset. Using this combination of augmentation techniques, we aim to bring synthetic data close enough to the real-world domain to train models that generalise to real-world inputs. For a given synthetic input image, we select one out of our set of the 51 blank pages. To increase variability, the blank page and the synthetic data undergo a variety of further augmentations, as shown in Table 2.

To ensure alignment with the transformed image data, the ground-truth bounding boxes also undergo the same transformations. Upon completing these augmentations, the foreground and background are merged by preserving the darker pixel at each position. This means that darker pixels overpower the lighter shades, preserving the dark symbols from the augmented synthetic dataset (the foreground) and replacing the pixels of its white background with the darker pixels from the augmented blank pages (the background). This yields optically similar results to real-world scanned music scores as seen in Figure 4, which can be adapted with hyperparameters (P_{snp}, P_{aug}, P_{blur}) to adjust to one's needs.

5.2 Unsupervised Adversarial Domain Adaptation

The most common approach to overcome a domain shift is supervised domain adaptation, where densely annotated images are required in the target domain (annotations generally involve instance-level bounding boxes for object detection). Such a solution would require the collection and annotation of a full-scale training dataset consisting of data from the target domain. This approach, therefore, would be tedious and lack the ability to scale, especially for detecting tiny objects in images that are cumbersome to annotate, such as notes in sheet music. Unsupervised

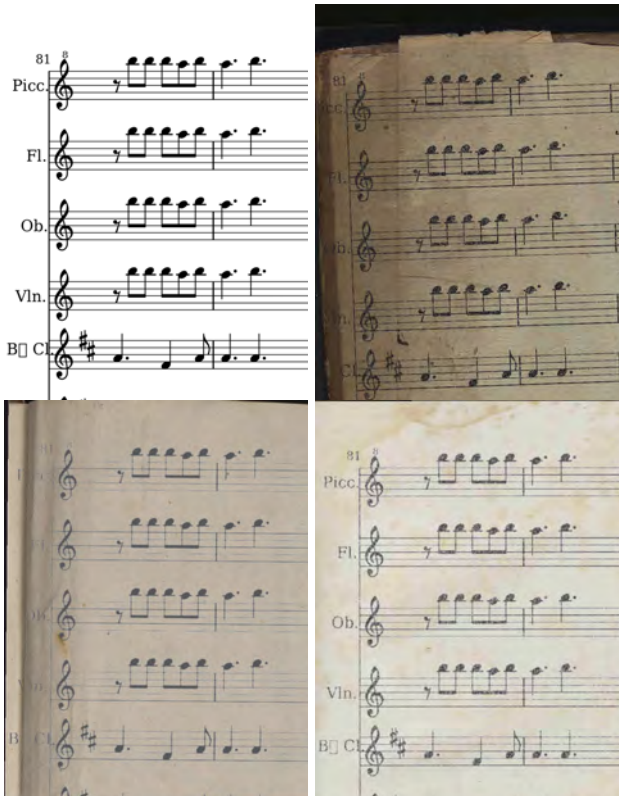

Figure 4: *ScoreAug* examples (top right, bottom row) derived from the same synthetic sample (top left).

domain adaptation (UDA), on the other hand, reduces the expense of annotation by only requiring annotations in the source domain.

Adversarial domain adaptation, which strives to minimise the domain dependency of an object detector via a domain-adversarial loss function utilising a discriminator, is a popular approach for UDA. As highlighted in Tzeng et al. (2017), adversarial domain adaptation is similar to generative adversarial learning, where a generator and discriminator are pitted against each other. For UDA, this concept is used to train a neural network to be unable to differentiate between two domains (in our case synthetic and real-world sheet music images) and ultimately show similar performance on source and target domain samples.

Here, the source domain is the *DeepScoresV2* dataset. For our target domain data, we source non-annotated real-world images from *IMSLP*. Our system consists of a baseline *S²A-Net* (comprising a backbone network $f^{\theta}_{backbone}$ and an object detector $f^{\theta'}_{detect}$), a gradient reversal layer and a small domain classifier neural network $f^{\theta''}_{domain}$. The network weights are denoted by θ, θ' and θ'' respectively. The system is trained using two independent losses, the domain confusion loss L_{domain} and the object detection loss L_{detect}. The following paragraph gives an overview of each component. See Figure 5 for a graphical overview.

The baseline *S²A-Net* is configured as described in

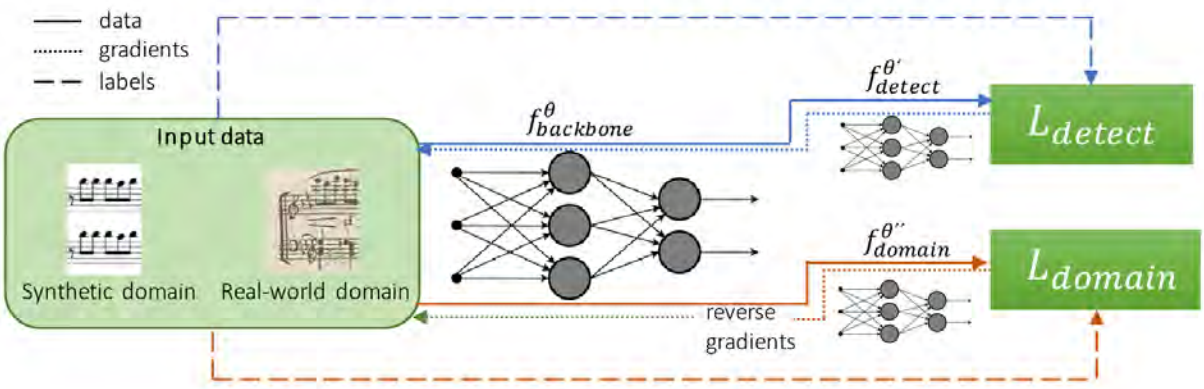

Figure 5: Overview of our UDA system, with data, gradient, and label flow of step (I) shown in orange, of step (II) in green and of step (III) in blue.

Section 3. Here we start with networks that have been fully trained on *DeepScoresV2* to ensure that the network filters are tuned to sheet music. The gradient reversal layer Ganin et al. (2016) can be viewed as a virtual layer in the network that is only active on the backward pass inverting all gradients passing through it. This causes the layers coming after this layer to *maximise* the training loss (in our case $f^{\theta}_{backbone}$ maximizing L_{domain}, causing the backbone embeddings to carry as little information about the domain as possible). $f^{\theta''}_{domain}$ has the job of classifying if an embedding generated by $f^{\theta}_{backbone}$ stems from a data point of the source domain or the target domain. L_{domain} is a binary classification target based on the input domain, as used in GANs Goodfellow et al. (2020). Finally, L_{detect} is the base *S²A-Net* loss to train the whole object detector.

Training the whole system requires the following steps: (I) training $f^{\theta''}_{domain}$ based on L_{domain} (θ'' is getting updated, θ is frozen); (II) use the gradients generated by L_{domain} and propagate them through $f^{\theta''}_{domain}$, applying the gradient reversal layer and propagating the resulting gradients through $f^{\theta}_{backbone}$ tuning θ to *maximise* L_{domain}; and (III) use labelled samples from the source domain and do a regular *S²A-Net* training step (training $f^{\theta}_{backbone}$ and $f^{\theta'}_{detect}$ based on L_{detect}). Steps (I) and (II) are pitted against each other in an adversarial game, with the goal of "deleting" information that allows the discriminator to differentiate between domains based on the output of the backbone, making the system “domain blind”. Step (III) is necessary since the backbone changes and the object detection head needs to adapt to the embeddings accordingly.

Preliminary experiments show that for models without pretraining on the *DeepScoresV2* dataset the resulting UDA do not perform at all. We conjecture that as we are dealing with unlabelled target domain data, it is crucial to learn good representations initially, otherwise the embeddings produced by the backbone would be too noisy and the domain classifier is unable to learn anything.

While implementing adversarial discriminative domain adaptation, we methodologically distinguish our work from Tzeng et al. (2017) in the following ways:

- We do not use separate networks for source and target domains for efficient sharing of weights.
- We do not fix the weights of our object detection module to allow the object detection module to adapt to the changes in the backbone.
- We do not adopt the asymmetric objective mappings of the feature extractor (in our case the output from *S²A-Net*).

5.3 Confidence Ratings

A music sheet often contains hundreds of musical symbols. Even if OMR software works very reliably, the probability of some misclassifications is high due to the high number of symbols. To identify such misclassifications, it is helpful to analyse the prediction confidence of the model. However, deep learning networks for classification are over-confident because their Softmax layer, which assigns decimal probabilities to each class, tends to push the probabilities either close to 0 or close to 1. Therefore, the model outputs cannot directly be used as a useful measure of confidence Dürr et al. (2020).

We mitigate this issue by using SnapshotEnsemble (Huang et al., 2017) (i.e. multiple predictions) to quantify the predictive uncertainty of our model. This method generates several snapshots (i.e. ensemble members) during training. During inference, each snapshot creates independent predictions of bounding boxes. We use the Weighted Box Fusion (WBF) (Solovyev et al., 2021) algorithm to fuse the bounding boxes. This method constructs the averaged bounding boxes with a corresponding confidence score by utilising the position and confidence scores of all proposed boxes. This overall score can be used as a measurement of the predictive uncertainty.

***DeepScoresV2* dataset**	
Model	AP (overlap = 0.25)
Baseline	87.6%
ScoreAug	86.0%
ScoreAug + *Finalise*	83.3%

***RealScores* dataset**	
Model	AP (overlap = 0.25)
Baseline	36.0%
ScoreAug	56.5%
ScoreAug + *Finalise*	73.7%

Table 3: The AP for the baseline model and models with *ScoreAug* and *Finalise* data augmentation on the *DeepScoresV2* and the *RealScores* datasets.

6. Experiments and Results

6.1 Input Data Augmentation

Experimental Setup To measure the impact of *ScoreAug*, we trained one baseline model with *ScoreAug* and another without it – each for 2000 epochs. Both models were trained on half-resolution, cropped samples to allow for larger batch sizes and faster convergence. During training, we used a learning rate of $\alpha = 2.5 \cdot 10^{-3}$ throughout and used linear warmup with a ratio of $\frac{1}{3}$ for the first 500 epochs. We observed that the models lack global awareness (e.g. predicting noteheads at the corner of the page), therefore we trained some models an additional 200 epochs on full pages, we denote this step as *Finalise*. During evaluation, we make sure to only consider the results of classes that have at least one positive prediction per model. We evaluate our models using average precision (AP) at an overlap of 25%. We use this unusually low overlap threshold due to the very small object sizes common in MOR, which cause detections that are very usable in practice to often be below the 50% mark.

Results Thanks to *ScoreAug* and *Finalise* we observe an absolute increase in AP of roughly 40% compared to models trained for the same number of epochs and without using both (see Table 3). We observe that on the source Dataset *DeepScoresV2* the performance slightly degrades from 87.6% to 83.3%. However, this is to be expected since we move from a model specifically trained on and for synthetic data to one that can handle a much wider variety of data.

6.2 Unsupervised Adversarial Domain Adapdation

Experimental Setup Pretraining the *S*^{2}*A-Net* for UDA showcased impressive results, allowing us to train for relatively few epochs. In our experiments, the pretrained checkpoint had been trained for 250 epochs on the *DeepScoresV2* dataset. We train our UDA pipeline for 30 further epochs. For the domain discriminator, the source domain label is set to 1 and the target domain label is set to 0. The input feature size is 128, based on the output from *S*^{2}*A-Net*, and the hidden feature size is 256. Batch normalisation Ioffe and Szegedy (2015) is applied to calculate the mean and standard deviation per dimension over the mini-batches. We use an Adam optimiser Kingma and Ba (2021) with an initial learning rate of 0.01 and constant epoch-driven decay for both targets. We train with a batch size of 4 for both source and target data loaders, which is the maximum batch size our GPUs allowed while keeping the number of samples balanced between domains. In our experiments, for a fair comparison, we follow the same configuration as the baseline model, in terms of *S*^{2}*A-Net* initialisation and *DeepScoresV2* data loader structures. We limit data augmentations on the *RealScores* data to geometric transformations such as scaling by a factor of 0.5 and random cropping of 1000 by 1000 pixels, both of which are similar to the *DeepScoresV2* data loader samples.

***DeepScoresV2* dataset**	
Model	AP (overlap = 0.25)
Baseline	87.6%
UDA	72.4%

***RealScores* dataset**	
Model	AP (overlap = 0.25)
Baseline	36.0%
UDA	48.9%

Table 4: The AP for the baseline model and a model with UDA on *DeepScoresV2* and the *RealScores* dataset.

Results Table 4 shows the average precisions for UDA. For the target domain *RealScores* we observe a gain of 12.9% from 36.0% to 48.9%. UDA results in the largest source domain performance loss from 87.6% down to 72.4%. This gain is not quite as impressive as for *ScoreAug*, but we believe it shows the merits of this fully unsupervised approach for MOR. Additionally, the UDA models have been trained only on low-resolution samples to overcome current GPU constraints. It is likely that results would improve in the future with higher resolution images which have generally aided object detection models dealing with tiny objects, such as in MOR.

6.3 Produce Confidence Ratings

Experimental Setup We train different ensemble versions as well as a model not utilising ensembles on the *DeepScoresV2* dataset. We train each model for 1000 epochs and with *ScoreAug*. For the model not utilising ensembles, we use a constant learning rate of $\alpha = 2.5 \cdot 10^{-3}$. For the SnapshotEnsemble models, we start with the same learning rate and decrease it over 500 epochs to $1 \cdot 10^{-5}$ using one single cosine annealing cycle. This rather long cycle is a pretraining of the model before the actual ensemble members are generated. To obtain the ensembles, we train the

***DeepScoresV2* dataset**	
Model	AP (overlap = 0.25)
ScoreAug	82.1%
ScoreAug ensemble (10 cycles)	85.6%
ScoreAug ensemble (20 cycles)	87.3%
ScoreAug ensemble (30 cycles)	83.4%

***RealScores* dataset**	
Model	AP (overlap = 0.25)
ScoreAug	37.9%
ScoreAug ensemble (10 cycles)	44.6%
ScoreAug ensemble (20 cycles)	46.7%
ScoreAug ensemble (30 cycles)	47.0%

Table 5: The AP for the model not utilizing ensembles and ensemble models with different cosine annealing cycle lengths on the *DeepScoresV2* and the *RealScores* dataset.

model for 500 additional epochs with shorter cosine annealing cycles over 10, 20, and 30 epochs with learning rates in the range of $1 \cdot 10^{-5} \leq \alpha \leq 7.5 \cdot 10^{-3}$.

After training, the AP for a given overlap of 0.25 is calculated on the test set. In addition, the overlap between snapshots is calculated by using the output from one model as ground truth and the output from a second model as the prediction. We build a set of 10 ensemble members iteratively. We start with an empty set of snapshots. First, the snapshot with the highest AP is added. Afterwards, we add the snapshot which has (i) an overall AP which is max. 5% worse than the AP of our best model; and (ii) has the smallest average overlap with the models which are already added to our set of ensemble members. We repeat this procedure until our set of ensemble members contains 10 snapshots.

The final predictions are the fused boxes generated by the WBF algorithm. The fusion threshold of WBF was set to 0.3 meaning that boxes with the same label and an intersection over union (IoU) of ≥ 0.3 are fused into one box. Since WBF can be used to quantify predictive uncertainty, we use this score to remove predictions with a confidence score below 10% on the *RealScores* dataset. We have found that this improves the prediction quality and reduces false positive rates in particular. In contrast, the predictions on the *DeepScoresV2* dataset are of high quality and no bounding boxes have to be removed based on their confidence score.

Results We have found that ensembles yield better results than a single model. Table 5 shows the AP of the ensemble approaches as well as the AP of the model not utilising ensembles. It can be observed that ensembles improve the AP by up to 5.2 p.p. on the *DeepScoresV2* dataset and up to 9.1 p.p. on the *RealScores* dataset compared to the model without ensembles. Of the three Cosine Annealing cycle lengths validated, the ensemble with a cycle length of 20 worked best on the *DeepScoresV2* dataset while the ensemble with a cycle length of 30 achieves the highest AP on the *RealScores* dataset. Compared to the results reported in Table 3, the ensemble approaches achieved a lower AP. Since the model's prediction accuracy increases continuously with more training epochs, we suspect that this is due to the fact that the ensembles are trained for only 1000 epochs, while the models in Table 3 are trained for 2000 epochs. However, it is likely that the ensemble would achieve similar or slightly better results, since ensembles typically improve results Dietterich (2000).

Having a high precision and thus a low false-positive rate is particularly important for OMR since it is easier for human annotators to find and label missing annotations than to identify wrong predictions. The confidence ratings can be used to reduce the number of false positive predictions and to increase the precision. We have found that removing predictions with a confidence score below 10% increases precision from 87.8% to 97.2% on the *DeepScoresV2* dataset, and from 35.7% to 41.9% on the *RealScores* dataset respectively. Thus, retaining only predictions with a confidence score bigger than a predefined threshold allows to increase precision at the expense of recall.

Additionally, we assess the confidence ratings visually. Figure 6 shows result excerpts from model outputs with the predictions coloured according to their confidence score. These visualisations can provide useful insights for creating annotations. In accordance with the previous findings, we have observed that analysing the predictions with low confidence is particularly helpful as wrong predictions usually have low confidence.

As in Section 6.1 (c.f. Table 3), we examine the effect of using *ScoreAug* in combination with *Finalise* (i.e. train on full pages) for the ensemble approach. We perform *Finalise* for 50 epochs on each ensemble member obtained. The results with and without *Finalise* are shown in Table 6. The effectiveness of using *Finalise* can be observed particularly clearly on the *RealScores* dataset. When training the ensemble approach for 1000 epochs with *ScoreAug* but without *Finalise*, we achieve an AP of 46.7%. If *ScoreAug* is combined with 50 additional *Finalise* epochs per ensemble member, the AP further improves to 63.6%. *Finalise* thus improves the results not only for single models but also for ensembles. On the source dataset, this once again leads to a small loss in performance from 87.3% to 81.5%. However, *Finalise* in combination with SnapshotEnsembles has the disadvantage that after creating the ensemble members, each member must be fine-tuned separately. This increases the duration of the fine-tuning linearly with the number

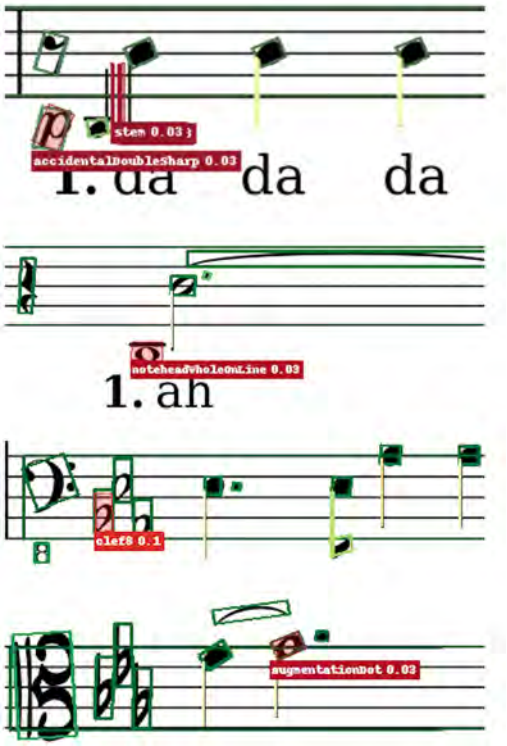

Figure 6: Four cropped visualisation samples of predictions made by an ensemble. The colour of the bounding box indicates the model's confidence (green means high confidence, and red means low confidence). For symbols with a confidence score below 30%, we plot not only the coloured bounding box but also the assigned label as well as the confidence score.

of ensemble members.

A combination of UDA with *ScoreAug* and snapshot ensembles is currently not indicated by the individual results: performance on *RealScores* exceeds 73% using *ScoreAug* + *Finalise* (see Table 3) and reaches beyond 63% when adding confidence ratings via ensembling (see Table 6), but only achieves ca. 49% using UDA (over a baseline of 36%, see Table 4). We do not expect a strong performance boost from a combination, especially since integration is technically uncertain due to the fact that UDA relies on fully pretrained networks and the fragile interplay between steps (I) to (III) that require specific learning rates (see Section 5.2).

7. Conclusions and Future Work

We presented multiple successful avenues towards improving the practical usability of OMR systems. Together, they improve the speed of professional-grade music digitalization on medium-quality scores by more than a factor of 3 over a strong baseline Tuggener et al. (2018b) for high-quality scores. Specifically, 11 minutes per page using the baseline could be reduced to 3.5 minutes on average within the digitalization pipeline of ScorePad AG, which consists of our MOR solution mated to a proprietary backend that combines all the information and features a human in the loop correction step. A fully manual transcription by professional musicians would take ca. 40 minutes.

***DeepScoresV2* dataset**	
Ensemble (cycle length = 20)	AP (overlap = 0.25)
ScoreAug	87.3%
ScoreAug & *Finalise*	81.5%

***RealScores* dataset**	
Ensemble (cycle length = 20)	AP (overlap = 0.25)
ScoreAug	46.7%
ScoreAug & *Finalise*	63.6%

Table 6: The AP for the ensemble trained with a cosine annealing cycle length of 20. The model is trained once with *ScoreAug* only and once with *ScoreAug* in combination with 50 subsequent *Finalise* cycles.

To bridge the domain gap between synthetic datasets and real-world data, algorithmic input augmentation paired with noise sourced from aged real-world documents proved especially fruitful, increasing average detection precision by nearly 50% on the *RealScores* data. In conjunction with *Finalise*, the model performed twice as well as the model trained on synthetic data only.

Unsupervised adversarial domain adaptation showed some promise, outperforming the baseline by 36%. We believe this could be further improved by a UDA method working at very high resolution, to prevent the destruction of fine-grained information in the small patterns of music notation. Both domain adaptation methods had a marginally adverse effect on the performance of the model on synthetic data. In a practical setting, this can be alleviated by employing a data quality classifier and using multiple expert models for high and low-quality data

Using ensembles in combination with weighted box fusion has improved the AP by up to 9.1pp. Besides the better results, ensembles allow us to calculate reliable confidence ratings. These confidence ratings can be used to identify misclassifications, and thus to simplify the manual post-processing of the predictions.

The current models cannot deal with hand-written music scores which could be addressed in the future. Another drawback is the heavy reliance on exact interline scaling, we observed a steep performance drop-off when the interline space is outside of the 8 to 12 pixel range. SnapshotEnsembles creates ensembles without additional training costs by storing snapshots during a single training cycle. The resulting snapshots are fine-tuned separately by using *Finalise* to achieve better performance on real-world data. Training would be more efficient if *Finalise* could be incorporated into the ensemble-generating training cycle and does not have to be done for each ensemble member separately.

References

Baró, A., Riba, P., and Fornés, A. (2018). A starting point for handwritten music recognition. In *1st International Workshop on Reading Music Systems*.

Bellini, P., Bruno, I., and Nesi, P. (2001). Optical music sheet segmentation. In *Proceedings First International Conference on WEB Delivering of Music. WEDELMUSIC 2001*, pages 183–190.

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

Calvo-Zaragoza, J. and Rizo, D. (2018a). Camera-primus: Neural end-to-end optical music recognition on realistic monophonic scores. In *19th International Society for Music Information Retrieval Conference, (ISMIR)*, pages 248–255.

Calvo-Zaragoza, J. and Rizo, D. (2018b). End-to-end neural optical music recognition of monophonic scores. *Applied Sciences*, 8(4):606.

Castellanos, F. J., Gallego, A.-J., and Calvo-Zaragoza, J. (2021). Unsupervised domain adaptation for document analysis of music score images. In *22nd International Society for Music Information Retrieval Conference, (ISMIR)*, pages 81–87.

Chen, Y., Li, W., Sakaridis, C., Dai, D., and Van Gool, L. (2018). Domain adaptive faster r-cnn for object detection in the wild. In *IEEE conference on computer vision and pattern recognition (CVPR)*, pages 3339–3348.

Chiu, C.-C., Sainath, T. N., Wu, Y., Prabhavalkar, R., Nguyen, P., Chen, Z., Kannan, A., Weiss, R. J., Rao, K., Gonina, K., Jaitly, N., Li, B., Chorowski, J., and Bacchiani, M. (2018). State-of-the-art speech recognition with sequence-to-sequence models. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4774–4778.

Chowdhury, A. and Vig, L. (2018). An efficient end-to-end neural model for handwritten text recognition. In *British Machine Vision Conference 2018, (BMVC)*, page 202.

Ciregan, D., Meier, U., and Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. In *2012 IEEE conference on computer vision and pattern recognition (CVPR)*, pages 3642–3649. IEEE.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. (2019). Autoaugment: Learning augmentation strategies from data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 113–123.

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

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15.

Durasov, N., Bagautdinov, T., Baque, P., and Fua, P. (2021). Masksembles for uncertainty estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13539–13548.

Dürr, O., Sick, B., and Murina, E. (2020). *Probabilistic Deep Learning: With Python, Keras and TensorFlow Probability*. Manning Publications.

Elezi, I., Tuggener, L., Pelillo, M., and Stadelmann, T. (2018). Deepscores and deep watershed detection: current state and open issues. In *1st International Workshop on Reading Music Systems*.

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

Gallego, A.-J. and Calvo-Zaragoza, J. (2017). Staff-line removal with selectional auto-encoders. *Expert Systems with Applications*, 89:138–148.

Ganin, Y. and Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In *international conference on machine learning (ICML)*, pages 1180–1189.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11):139–144.

Gustafsson, F. K., Danelljan, M., and Schon, T. B. (2020). Evaluating scalable bayesian deep learning methods for robust computer vision. In *IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 318–319.

Hajič, J. and Pecina, P. (2017). The muscima++ dataset for handwritten optical music recognition. In *14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pages 39–46.

Hajic Jr, J., Dorfer, M., Widmer, G., and Pecina, P. (2018). Towards full-pipeline handwritten omr with musical symbol detection by u-nets. In *19th International Society for Music Information Retrieval Conference, (ISMIR)*, pages 225–232.

Han, J., Ding, J., Li, J., and Xia, G.-S. (2021). Align deep features for oriented object detection. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–11.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition (CVPR)*, pages 770–778.

Huang, G., Li, Y., Pleiss, G., Liu, Z., Hopcroft, J. E., and Weinberger, K. Q. (2017). Snapshot ensembles:

Train 1, get M for free. In *5th International Conference on Learning Representations, (ICLR)*.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning (ICML)*, pages 448–456.

Kingma, D. P. and Ba, J. (2021). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90.

Lehner, A., Gasperini, S., Marcos-Ramiro, A., Schmidt, M., Mahani, M.-A. N., Navab, N., Busam, B., and Tombari, F. (2022). 3d-vfield: Adversarial augmentation of point clouds for domain generalization in 3d object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17295–17304.

Li, Y.-J., Dai, X., Ma, C.-Y., Liu, Y.-C., Chen, K., Wu, B., He, Z., Kitani, K., and Vajda, P. (2022). Cross-domain adaptive teacher for object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7581–7590.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). Feature pyramid networks for object detection. In *IEEE conference on computer vision and pattern recognition (CVPR)*, pages 2117–2125.

Mateiu, T. N., Gallego, A.-J., and Calvo-Zaragoza, J. (2019). Domain adaptation for handwritten symbol recognition: A case of study in old music manuscripts. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 135–146.

Nguyen, A., Yosinski, J., and Clune, J. (2015). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *IEEE conference on computer vision and pattern recognition (CVPR)*, pages 427–436.

Pacha, A. and Calvo-Zaragoza, J. (2018). Optical music recognition in mensural notation with region-based convolutional neural networks. In *19th International Society for Music Information Retrieval Conference, (ISMIR)*, pages 240–247.

Pacha, A., Choi, K.-Y., Coüasnon, B., Ricquebourg, Y., Zanibbi, R., and Eidenberger, H. (2018a). Handwritten music object detection: Open issues and baseline results. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 163–168. IEEE.

Pacha, A., Hajič, J., and Calvo-Zaragoza, J. (2018b). A baseline for general music object detection with deep learning. *Applied Sciences*, 8(9):1488.

Pugin, L. (2006). Optical music recognitoin of early typographic prints using hidden markov models. In *Proceedings of the 7th International Society for Music Information Retrieval Conference (ISMIR)*, pages 53–56.

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

Sato, I., Nishimura, H., and Yokoi, K. (2015). Apac: Augmented pattern classification with neural networks. *arXiv preprint arXiv:1505.03229*.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61:85–117.

Simmler, N., Sager, P., Andermatt, P., Chavarriaga, R., Schilling, F.-P., Rosenthal, M., and Stadelmann, T. (2021). A survey of un-, weakly-, and semi-supervised learning methods for noisy, missing and partial labels in industrial vision applications. In *2021 8th Swiss Conference on Data Science (SDS)*, pages 26–31.

Smith, L. N. (2017). Cyclical learning rates for training neural networks. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pages 464–472.

Solovyev, R., Wang, W., and Gabruseva, T. (2021). Weighted boxes fusion: Ensembling boxes from different object detection models. *Image and Vision Computing*, pages 1–6.

Stadelmann, T., Amirian, M., Arabaci, I., Arnold, M., Duivesteijn, G. F., Elezi, I., Geiger, M., Lörwald, S., Meier, B. B., Rombach, K., et al. (2018). Deep learning in the wild. In *IAPR Workshop on Artificial Neural Networks in Pattern Recognition*, pages 17–38.

Toyama, F., Shoji, K., and Miyamichi, J. (2006). Symbol recognition of printed piano scores with touching symbols. In *18th International Conference on Pattern Recognition (ICPR)*, volume 2, pages 480–483.

Tuggener, L., Elezi, I., Schmidhuber, J., Pelillo, M., and Stadelmann, T. (2018a). Deepscores-a dataset for segmentation, detection and classification of tiny objects. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 3704–3709.

Tuggener, L., Elezi, I., Schmidhuber, J., and Stadelmann, T. (2018b). Deep watershed detector for music object recognition. In *19th International Society for Music Information Retrieval Conference (ISMIR)*, pages 271–278.

Tuggener, L., Satyawan, Y. P., Pacha, A., Schmidhuber, J., and Stadelmann, T. (2021). The deepscoresv2 dataset and benchmark for music object detection. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 9188–9195.

Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. (2017). Adversarial discriminative domain

adaptation. In *IEEE conference on computer vision and pattern recognition (CVPR)*, pages 7167–7176.

van der Wel, E. and Ullrich, K. (2017). Optical music recognition with convolutional sequence-to-sequence models. In *18th International Society for Music Information Retrieval Conference, (ISMIR)*, pages 731–737.

von Oswald, J., Kobayashi, S., Sacramento, J., Meulemans, A., Henning, C., and Grewe, B. F. (2021). Neural networks with late-phase weights. In *9th International Conference on Learning Representations, (ICLR)*.

Wen, Y., Tran, D., and Ba, J. (2020). Batchensemble: an alternative approach to efficient ensemble and lifelong learning. In *8th International Conference on Learning Representations, (ICLR)*.

Wenzel, F., Snoek, J., Tran, D., and Jenatton, R. (2020). Hyperparameter ensembles for robustness and uncertainty quantification. In *34th International Conference on Neural Information Processing Systems (NeurIPS)*, pages 6514–6527.

Xia, Y., Zhang, J., Jiang, T., Gong, Z., Yao, W., and Feng, L. (2021). Hatchensemble: an efficient and practical uncertainty quantification method for deep neural networks. *Complex & Intelligent Systems*, 7:2855–2869.

Zhu, X., Liu, Y., Qin, Z., and Li, J. (2017). Data augmentation in emotion classification using generative adversarial networks. *arXiv preprint arXiv:1711.00648*.

Zhu, X., Pang, J., Yang, C., Shi, J., and Lin, D. (2019). Adapting object detectors via selective cross-domain alignment. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 687–696.