TOWARDS SCORE FOLLOWING IN SHEET MUSIC IMAGES

Matthias Dorfer **Andreas Arzt** **Gerhard Widmer**
Department of Computational Perception, Johannes Kepler University Linz, Austria
matthias.dorfer@jku.at

ABSTRACT

This paper addresses the matching of short music audio snippets to the corresponding pixel location in images of sheet music. A system is presented that simultaneously learns to read notes, listens to music and matches the currently played music to its corresponding notes in the sheet. It consists of an end-to-end multi-modal convolutional neural network that takes as input images of sheet music and spectrograms of the respective audio snippets. It learns to predict, for a given unseen audio snippet (covering approximately one bar of music), the corresponding position in the respective score line. Our results suggest that with the use of (deep) neural networks – which have proven to be powerful image processing models – working with sheet music becomes feasible and a promising future research direction.

1. INTRODUCTION

Precisely linking a performance to its respective sheet music – commonly referred to as audio-to-score alignment – is an important topic in MIR and the basis for many applications [20]. For instance, the combination of score and audio supports algorithms and tools that help musicologists in in-depth performance analysis (see e.g. [6]), allows for new ways to browse and listen to classical music (e.g. [9, 13]), and can generally be helpful in the creation of training data for tasks like beat tracking or chord recognition. When done on-line, the alignment task is known as score following, and enables a range of applications like the synchronization of visualisations to the live music during concerts (e.g. [1, 17]), and automatic accompaniment and interaction live on stage (e.g. [5, 18]).

So far all approaches to this task depend on a symbolic, computer-readable representation of the sheet music, such as MusicXML or MIDI (see e.g. [1, 5, 8, 12, 14–18]). This representation is created either manually (e.g. via the time-consuming process of (re-)setting the score in a music notation program), or automatically via optical music recognition software. Unfortunately automatic methods are still highly unreliable and thus of limited use, especially for more complex music like orchestral scores [20].

The central idea of this paper is to develop a method that links the audio and the image of the sheet music *directly*, by *learning* correspondences between these two modalities, and thus making the complicated step of creating an in-between representation obsolete. We aim for an algorithm that simultaneously learns to *read notes*, *listens* to music and *matches* the currently played music with the correct notes in the sheet music. We will tackle the problem in an end-to-end neural network fashion, meaning that the entire behaviour of the algorithm is learned purely from data and no further manual feature engineering is required.

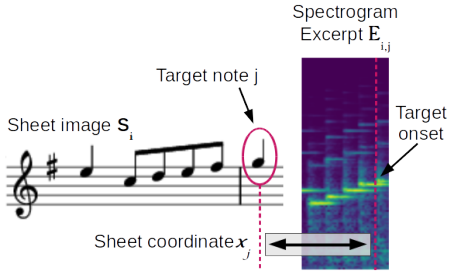
2. METHODS

This section describes the audio-to-sheet matching model and the input data required, and shows how the model is used at test time to predict the expected location of a new unseen audio snippets in the respective sheet image.

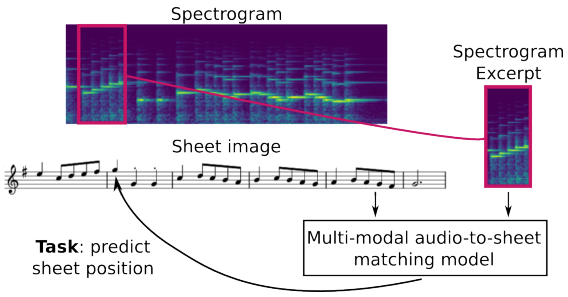
2.1 Data, Notation and Task Description

The model takes two different input modalities at the same time: images of scores, and short excerpts from spectrograms of audio renditions of the score (we will call these *query snippets* as the task is to predict the position in the score that corresponds to such an audio snippet). For this first proof-of-concept paper, we make a number of simplifying assumptions: for the time being, the system is fed only a *single staff line* at a time (not a full page of score). We restrict ourselves to *monophonic music*, and to the *piano*. To generate training examples, we produce a fixed-length query snippet for each note (onset) in the audio. The snippet covers the target note onset plus a few additional frames, at the end of the snippet, and a fixed-size context of 1.2 seconds into the past, to give some temporal context. The same procedure is followed when producing example queries for off-line testing.

A training/testing example is thus composed of two inputs: Input 1 is an image $\mathbf{S}_i$ (in our case of size 40×390 pixels) showing one staff of sheet music. Input 2 is an audio snippet – specifically, a spectrogram excerpt $\mathbf{E}_{i,j}$ (40 frames $\times$ 136 frequency bins) – cut from a recording of the piece, of fixed length (1.2 seconds). The rightmost onset in spectrogram excerpt $\mathbf{E}_{i,j}$ is interpreted as the target note j whose position we want to predict in staff image $\mathbf{S}_i$. For the music used in our experiments (Section 3) this context is a bit less than one bar. For each note j (represented by its corresponding spectrogram excerpt $\mathbf{E}_{i,j}$) we annotated its *ground truth* sheet location x_j in sheet image $\mathbf{S}_i$. Coor-



(a) Spectrogram-to-sheet correspondence. In this example the rightmost onset in spectrogram excerpt $\mathbf{E}_{i,j}$ corresponds to the rightmost note (target note j) in sheet image $\mathbf{S}_i$. For the present case the temporal context of about 1.2 seconds (into the past) covers five additional notes in the spectrogram. The staff image and spectrogram excerpt are exactly the multi-modal input presented to the proposed audio-to-sheet matching network. At train time the target pixel location x_j in the sheet image is available; at test time $\hat{x}_j$ has to be predicted by the model (see figure below).



(b) Schematic sketch of the audio-to-sheet matching task targeted in this work. Given a sheet image $\mathbf{S}_i$ and a short snippet of audio (spectrogram excerpt $\mathbf{E}_{i,j}$) the model has to predict the audio snippet's corresponding pixel location x_j in the image.

Figure 1: Input data and audio-to-sheet matching task.

dinate x_j is the distance of the note head (in pixels) from the left border of the image. As we work with single staffs of sheet music we only need the x-coordinate of the note at this point. Figure 1a relates all components involved.

Summary and Task Description: For training we present triples of (1) staff image $\mathbf{S}_i$, (2) spectrogram excerpt $\mathbf{E}_{i,j}$ and (3) ground truth pixel x-coordinate x_j to our audio-to-sheet matching model. At test time only the staff image and spectrogram excerpt are available and the task of the model is to predict the estimated pixel location $\hat{x}_j$ in the image. Figure 1b shows a sketch summarizing this task.

2.2 Audio-Sheet Matching as Bucket Classification

We now propose a multi-modal convolutional neural network architecture that learns to match unseen audio snippets (spectrogram excerpts) to their corresponding pixel location in the sheet image.

2.2.1 Network Structure

Figure 2 provides a general overview of the deep network and the proposed solution to the matching problem. As mentioned above, the model operates jointly on a staff image $\mathbf{S}_i$ and the audio (spectrogram) excerpt $\mathbf{E}_{i,j}$ related to a note j. The rightmost onset in the spectrogram excerpt is the one related to target note j. The multi-modal model consists of two specialized convolutional networks: one dealing with the sheet image and one dealing with the audio (spectrogram) input. In the subsequent layers we fuse the specialized sub-networks by concatenation of the latent image- and audio representations and additional processing by a sequence of dense layers. For a detailed description of the individual layers we refer to Table 1 in Section 3.4. The output layer of the network and the corresponding localization principle are explained in the following.

2.2.2 Audio-to-Sheet Bucket Classification

The objective for an unseen spectrogram excerpt and a corresponding staff of sheet music is to predict the excerpt's location x_j in the staff image. For this purpose we start with horizontally quantizing the sheet image into B non-overlapping buckets. This discretisation step is indicated as the short vertical lines in the staff image above the score in Figure 2. In a second step we create for each note j in the train set a target vector $\mathbf{t}_j = \{t_{j,b}\}$ where each vector element $t_{j,b}$ holds the probability that bucket b covers the current target note j. In particular, we use soft targets, meaning that the probability for one note is shared between the two buckets closest to the note's true pixel location x_j. We linearly interpolate the shared probabilities based on the two pixel distances (normalized to sum up to one) of the note's location x_j to the respective (closest) bucket centers. Bucket centers are denoted by c_b in the following where subscript b is the index of the respective bucket. Figure 3 shows an example sketch of the components described above. Based on the soft target vectors we design the output layer of our audio-to-sheet matching network as a B-way soft-max with activations defined as:

$$\phi(y_{j,b}) = \frac{e^{y_{j,b}}}{\sum_{k=1}^{B} e^{y_{j,k}}} \tag{1}$$

$\phi(y_{j,b})$ is the soft-max activation of the output neuron representing bucket b and hence also representing the region in the sheet image covered by this bucket. By applying the soft-max activation the network output gets normalized to range $(0, 1)$ and further sums up to 1.0 over all B output neurons. The network output can now also be interpreted as a vector of probabilities $\mathbf{p}_j = \{\phi(y_{j,b})\}$ and shares the same value range and properties as the soft target vectors.

In training, we optimize the network parameters Θ by minimizing the Categorical Cross Entropy (CCE) loss l_j between target vectors $\mathbf{t}_j$ and network output $\mathbf{p}_j$:

$$l_j(\Theta) = -\sum_{k=1}^{B} t_{j,k}\, log(p_{j,k}) \tag{2}$$

The CCE loss function becomes minimal when the network output $\mathbf{p}_j$ exactly matches the respective soft target vector $\mathbf{t}_j$. In Section 3.4 we provide further information on the exact optimization strategy used. [1]

[1] *For the sake of completeness*: In our initial experiments we started to predict the sheet location of audio snippets by minimizing the Mean-Squared-Error (MSE) between the predicted and the true pixel coordinate (MSE regression). However, we observed that training these networks is much harder and further performs worse than the bucket classification approach proposed in this paper.

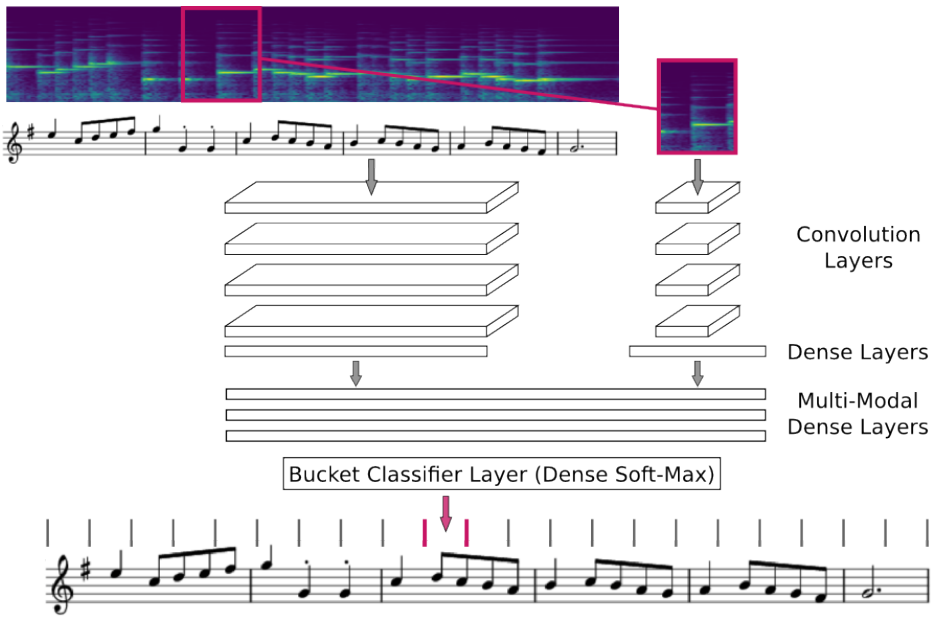


Figure 2: Overview of multi-modal convolutional neural network for audio-to-sheet matching. The network takes a staff image and a spectrogram excerpt as input. Two specialized convolutional network parts, one for the sheet image and one for the audio input, are merged into one multi-modality network. The output part of the network predicts the region in the sheet image – the classification bucket – to which the audio snippet corresponds.

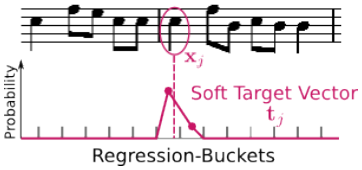


Figure 3: Part of a staff of sheet music along with soft target vector $\mathbf{t}_j$ for target note j surrounded with an ellipse. The two buckets closest to the note share the probability (indicated as dots) of containing the note. The short vertical lines highlight the bucket borders.

2.3 Sheet Location Prediction

Once the model is trained, we use it at test time to predict the expected location $\hat{x}_j$ of an audio snippet with target note j in a corresponding image of sheet music. The output of the network is a vector $\mathbf{p}_j = \{p_{j,b}\}$ holding the probabilities that the given test snippet j matches with bucket b in the sheet image. Having these probabilities we consider two different types of predictions: (1) We compute the center c_b^* of bucket $b^* = \text{argmax}_b\, p_{j,b}$ holding the highest overall matching probability. (2) For the second case we take, in addition to b^*, the two neighbouring buckets $b^* - 1$ and $b^* + 1$ into account and compute a (linearly) probability weighted position prediction in the sheet image as

$$\hat{x}_j = \sum_{k \in \{b^*-1, b^*, b^*+1\}} w_k c_k \qquad (3)$$

where weight vector $\mathbf{w}$ contains the probabilities $\{p_{j,b^*-1}, p_{j,b^*}, p_{j,b^*+1}\}$ normalized to sum up to one and c_k are the center coordinates of the respective buckets.

3. EXPERIMENTAL EVALUATION

This section evaluates our audio-to-sheet matching model on a publicly available dataset. We describe the experimental setup, including the data and evaluation measures, the particular network architecture as well as the optimization strategy, and provide quantitative results.

3.1 Experiment Description

The aim of this paper is to show that it is feasible to learn correspondences between audio (spectrograms) and images of sheet music in an *end-to-end* neural network fashion, meaning that an algorithm learns the entire task purely from data, so that no hand crafted feature engineering is required. We try to keep the experimental setup simple and consider one staff of sheet music per train/test sample (this is exactly the setup drafted in Figure 2). To be perfectly clear, the task at hand is the following: For a given audio snippet, find its x-coordinate pixel position in a corresponding staff of sheet music. We further restrict the audio to monophonic music containing half, quarter and eighth notes but allow variations such as dotted notes, notes tied across bar lines as well as accidental signs.

3.2 Data

For the evaluation of our approach we consider the Nottingham [2] data set which was used, e.g., for piano transcription in [4]. It is a collection of midi files already split into train, validation and test tracks. To be suitable for audio-to-sheet matching we prepare the data set (midi files) as follows:

[2] www-etud.iro.umontreal.ca/~boulanni/icml2012

Sheet-Image 40×390	Spectrogram 136×40
5×5 Conv(pad-2, stride-1-2)-64-BN-ReLu	3×3 Conv(pad-1)-64-BN-ReLu
3×3 Conv(pad-1)-64-BN-ReLu	3×3 Conv(pad-1)-64-BN-ReLu
2×2 Max-Pooling + Drop-Out(0.15)	2×2 Max-Pooling + Drop-Out(0.15)
3×3 Conv(pad-1)-128-BN-ReLu	3×3 Conv(pad-1)-96-BN-ReLu
3×3 Conv(pad-1)-128-BN-ReLu	2×2 Max-Pooling + Drop-Out(0.15)
2×2 Max-Pooling + Drop-Out(0.15)	3×3 Conv(pad-1)-96-BN-ReLu
	2×2 Max-Pooling + Drop-Out(0.15)
Dense-1024-BN-ReLu + Drop-Out(0.3)	Dense-1024-BN-ReLu + Drop-Out(0.3)
Concatenation-Layer-2048	
Dense-1024-BN-ReLu + Drop-Out(0.3)	
Dense-1024-BN-ReLu + Drop-Out(0.3)	
B-way Soft-Max Layer	

Table 1: Architecture of Multi-Modal Audio-to-Sheet Matching Model: BN: Batch Normalization, ReLu: Rectified Linear Activation Function, CCE: Categorical Cross Entropy, Mini-batch size: 100

1. We select the first track of the midi files (right hand, piano) and render it as sheet music using Lilypond. [3]
2. We annotate the sheet coordinate x_j of each note.
3. We synthesize the midi-tracks to *flac*-audio using Fluidsynth [4] and a *Steinway* piano sound font.
4. We extract the audio timestamps of all note onsets.

As a last preprocessing step we compute *log-spectrograms* of the synthesized flac files [3], with an audio sample rate of 22.05kHz, FFT window size of 2048 samples, and computation rate of 31.25 frames per second. For dimensionality reduction we apply a normalized 24-band logarithmic filterbank allowing only frequencies from 80Hz to 8kHz. This results in 136 frequency bins.

We already showed a spectrogram-to-sheet annotation example in Figure 1a. In our experiment we use spectrogram excerpts covering 1.2 seconds of audio (40 frames). This context is kept the same for training and testing. Again, annotations are aligned in a way so that the rightmost onset in a spectrogram excerpt corresponds to the pixel position of target note j in the sheet image. In addition, the spectrogram is shifted 5 frames to the right to also contain some information on the current target note's onset and pitch. We chose this annotation variant with the rightmost onset as it allows for an online application of our audio-to-sheet model (as would be required, e.g., in a score following task).

3.3 Evaluation Measures

To evaluate our approach we consider, for each test note j, the following ground truth and prediction data: (1) The true position x_j as well as the corresponding target bucket b_j (see Figure 3). (2) The estimated sheet location $\hat{x}_j$ and the most likely target bucket b^* predicted by the model. Given this data we compute two types of evaluation measures.

The first – the *top-k bucket hit rate* – quantifies the ratio of notes that are classified into the correct bucket allowing a tolerance of $k-1$ buckets. For example, the *top-1 bucket hit rate* counts only those notes where the predicted bucket b^* matches exactly the note's target bucket b_j. The *top-2 bucket hit rate* allows for a tolerance of one bucket and so on. The second measure – the *normalized pixel distance* – captures the actual distance of a predicted sheet location $\hat{x}_j$ to its corresponding true position x_j. To allow for an evaluation independent of the image resolution used in our experiments we normalize the pixel errors by dividing them by the width of the sheet image as $(\hat{x}_j - x_j)/width(\mathbf{S}_i)$. This results in distance errors living in range $(-1, 1)$.

We would like to emphasise that the quantitative evaluations based on the measures introduced above are performed only at time steps where a note onset is present. At those points in time an explicit correspondence between spectrogram (onset) and sheet image (note head) is established. However, in Section 4 we show that a time-continuous prediction is also feasible with our model and onset detection is not required at run time.

3.4 Model Architecture and Optimization

Table 1 gives details on the model architecture used for our experiments. As shown in Figure 2, the model is structured into two disjoint convolutional networks where one considers the sheet image and one the spectrogram (audio) input. The convolutional parts of our model are inspired by the VGG model built from sequences of small convolution kernels (e.g. 3×3) and max-pooling layers. The central part of the model consists of a concatenation layer bringing the image and spectrogram sub-networks together. After two dense layers with 1024 units each we add a B-way soft-max output layer. Each of the B soft-max output neurons corresponds to one of the disjoint buckets which in turn represent quantised sheet image positions. In our experiments we use a fixed number of 40 buckets selected as follows: We measure the minimum distance between two subsequent notes – in our sheet renderings – and select the number of buckets such that each bucket contains at most one note. It is of course possible that no note is present in a bucket – e.g., for the buckets covering the clef at the

[3] http://www.lilypond.org/
[4] http://www.fluidsynth.org/

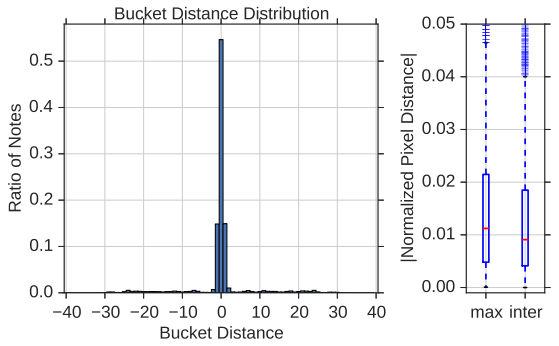


Figure 4: Summary of matching results on test set. *Left*: Histogram of bucket distances between predicted and true buckets. *Right*: Box-plots of absolute *normalized pixel distances* between predicted and true image position. The box-plot is shown for both location prediction methods described in Section 2.3 (maximum, interpolated).

	Train	**Valid**	**Test**
Top-1-Bucket-Hit-Rate	79.28%	51.63%	54.64%
Top-2-Bucket-Hit-Rate	94.52%	82.55%	84.36%
mean($\|NPD_{max}\|$)	0.0316	0.0684	0.0647
mean($\|NPD_{int}\|$)	0.0285	0.0670	0.0633
median($\|NPD_{max}\|$)	0.0067	0.0119	0.0112
median($\|NPD_{int}\|$)	0.0033	0.0098	0.0091
$\|NPD_{max}\| < w_b$	93.87%	76.31%	79.01%
$\|NPD_{int}\| < w_b$	94.21%	78.37%	81.18%

Table 2: *Top-k bucket hit rates* and *normalized pixel distances* (NPD) as described in Section 3.4 for train, validation and test set. We report mean and median of the absolute NPDs for both interpolated (int) and maximum (max) probability bucket prediction. The last two rows report the percentage of predictions not further away from the true pixel location than the width w_b of one bucket.

beginning of a staff. As activations function for the inner layers we use rectified linear units [10] and apply batch normalization [11] after each layer as it helps training and convergence.

Given this architecture and data we optimize the parameters of the model using mini-batch stochastic gradient descent with Nesterov style momentum. We set the batch size to 100 and fix the momentum at 0.9 for all epochs. The initial learn-rate is set to 0.1 and divided by 10 every 10 epochs. We additionally apply a weight decay of 0.0001 to all trainable parameters of the model.

3.5 Experimental Results

Figure 4 shows a histogram of the signed bucket distances between predicted and true buckets. The plot shows that more than 54% of all unseen test notes are matched exactly with the corresponding bucket. When we allow for a tolerance of ± 1 bucket our model is able to assign over 84% of the test notes correctly. We can further observe that the prediction errors are equally distributed in both directions – meaning too early and too late in terms of audio. The results are also reported in numbers in Table 2, as the top-k bucket hit rates for train, validation and test set.

The box plots in the right part of Figure 4 summarize the absolute *normalized pixel distances (NPD)* between predicted and true locations. We see that the probability-weighted position interpolation (Section 2.3) helps improve the localization performance of the model. Table 2 again puts the results in numbers, as means and medians of the absolute NPD values. Finally, Fig. 2 (bottom) reports the ratio of predictions with a pixel distance smaller than the width of a single bucket.

4. DISCUSSION AND REAL MUSIC

This section provides a representative prediction example of our model and uses it to discuss the proposed approach. In the second part we then show a first step towards matching *real* (though still very simple) music to its corresponding sheet. By *real music* we mean audio that is not just synthesized midi, but played by a human on a piano and recorded via microphone.

4.1 Prediction Example and Discussion

Figure 5 shows the image of one staff of sheet music along with the predicted as well as the ground truth pixel location for a snippet of audio. The network correctly matches the spectrogram with the corresponding pixel location in the sheet image. However, we observe a second peak in the bucket prediction probability vector. A closer look shows that this is entirely reasonable, as the music is quite repetitive and the current target situation actually appears twice in the score. The ability of predicting probabilities for multiple positions is a desirable and important property, as repetitive structures are immanent to music. The resulting prediction ambiguities can be addressed by exploiting the temporal relations between the notes in a piece by methods such as dynamic time warping or probabilistic models. In fact, we plan to combine the probabilistic output of our matching model with existing score following methods, as for example [2]. In Section 2 we mentioned that training a sheet location prediction with MSE-regression is difficult to optimize. Besides this technical drawback it would not be straightforward to predict a variable number of locations with an MSE-model, as the number of network outputs has to be fixed when designing the model.

In addition to the network inputs and prediction Fig. 5 also shows a *saliency map* [19] computed on the input sheet image with respect to the network output. [5] The saliency can be interpreted as the input regions to which most of the net's attention is drawn. In other words, it highlights the regions that contribute most to the current output produced by the model. A nice insight of this visualization is that the network actually focuses and recognizes the heads of the individual notes. In addition it also directs some attention to the style of stems, which is necessary to distinguish for example between quarter and eighth notes.

[5] The implementation is adopted from an example by Jan Schlüter in the recipes section of the deep learning framework *Lasagne* [7].

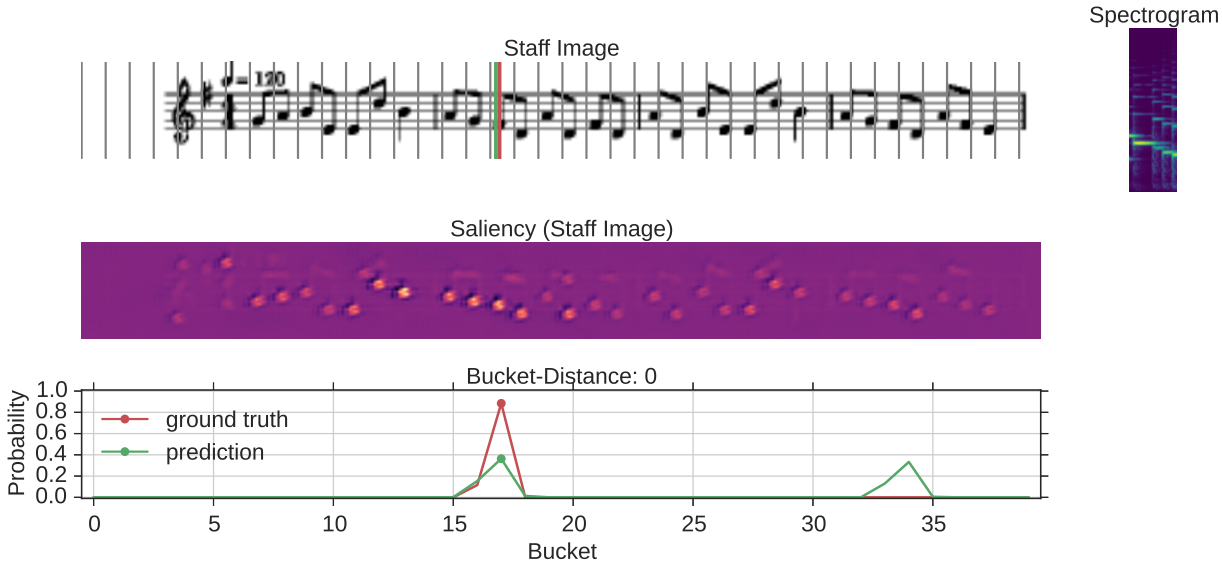


Figure 5: Example prediction of the proposed model. The top row shows the input staff image $\mathbf{S}_i$ along with the bucket borders as thin gray lines, and the given query audio (spectrogram) snippet $\mathbf{E}_{i,j}$. The plot in the middle visualizes the salience map (representing the attention of the neural network) computed on the input image. Note that the network's attention is actually drawn to the individual note heads. The bottom row compares the ground truth bucket probabilities with the probabilities predicted by the network. In addition, we also highlight the corresponding true and predicted pixel locations in the staff image in the top row.

The optimization on soft target vectors is also reflected in the predicted bucket probabilities. In particular the neighbours of the bucket with maximum activation are also active even though there is no explicit neighbourhood relation encoded in the soft-max output layer. This helps the interpolation of the true position in the image (see Fig. 4).

4.2 First Steps with Real Music

As a final point, we report on first attempts at working with "real" music. For this purpose one of the authors played the right hand part of a simple piece (Minuet in G Major by Johann Sebastian Bach, BWV Anhang 114) – which, of course, was not part of the training data – on a *Yamaha AvantGrand N2* hybrid piano and recorded it using a single microphone. In this application scenario we predict the corresponding sheet locations not only at times of onsets but for a continuous audio stream (subsequent spectrogram excerpts). This can be seen as a simple version of online score following in sheet music, without taking into account the temporal relations of the predictions. We offer the reader a video [6] that shows our model following the first three staff lines of this simple piece. [7] The ratio of predicted notes having a pixel-distance smaller than the bucket width (compare Section 3.5) is 71.72% for this real recording. This corresponds to a average normalized-pixel-distance of 0.0402.

[6] https://www.dropbox.com/s/0nz540i1178hjp3/Bach_Minuet_G_Major_net4b.mp4?dl=0

[7] Note: our model operates on single staffs of sheet music and requires a certain context of spectrogram frames for prediction (in our case 40 frames). For this reason it cannot provide a localization for the first couple of notes in the beginning of each staff at the current stage. In the video one can observe that prediction only starts when the spectrogram in the top right corner has grown to the desired size of 40 frames. We kept this behaviour for now as we see our work as a proof of concept. The issue can be easily addressed by concatenating the images of subsequent staffs in horizontal direction. In this way we will get a "continuous stream of sheet music" analogous to a spectrogram for audio.

5. CONCLUSION

In this paper we presented a multi-modal convolutional neural network which is able to match short snippets of audio with their corresponding position in the respective image of sheet music, without the need of any symbolic representation of the score. First evaluations on simple piano music suggest that this is a very promising new approach that deserves to be explored further.

As this is a proof of concept paper, naturally our method still has some severe limitations. So far our approach can only deal with monophonic music, notated on a single staff, and with performances that are roughly played in the same tempo as was set in our training examples.

In the future we will explore options to lift these limitations one by one, with the ultimate goal of making this approach applicable to virtually any kind of complex sheet music. In addition, we will try to combine this approach with a score following algorithm. Our vision here is to build a score following system that is capable of dealing with any kind of classical sheet music, out of the box, with no need for data preparation.

6. ACKNOWLEDGEMENTS

This work is supported by the Austrian Ministries BMVIT and BMWFW, and the Province of Upper Austria via the COMET Center SCCH, and by the European Research Council (ERC Grant Agreement 670035, project CON ESPRESSIONE). The Tesla K40 used for this research was donated by the NVIDIA corporation.

7. REFERENCES

[1] Andreas Arzt, Harald Frostel, Thassilo Gadermaier, Martin Gasser, Maarten Grachten, and Gerhard Widmer. Artificial intelligence in the concertgebouw. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, Buenos Aires, Argentina, 2015.

[2] Andreas Arzt, Gerhard Widmer, and Simon Dixon. Automatic page turning for musicians via real-time machine listening. In *Proc. of the European Conference on Artificial Intelligence (ECAI)*, Patras, Greece, 2008.

[3] Sebastian Böck, Filip Korzeniowski, Jan Schlüter, Florian Krebs, and Gerhard Widmer. madmom: a new Python Audio and Music Signal Processing Library. *arXiv:1605.07008*, 2016.

[4] Nicolas Boulanger-lewandowski, Yoshua Bengio, and Pascal Vincent. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 1159–1166, 2012.

[5] Arshia Cont. A coupled duration-focused architecture for realtime music to score alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):837–846, 2009.

[6] Nicholas Cook. Performance analysis and chopin's mazurkas. *Musicae Scientae*, 11(2):183–205, 2007.

[7] Sander Dieleman, Jan Schlüter, Colin Raffel, Eben Olson, Søren Kaae Sønderby, Daniel Nouri, Eric Battenberg, Aäron van den Oord, et al. Lasagne: First release., August 2015.

[8] Zhiyao Duan and Bryan Pardo. A state space model for on-line polyphonic audio-score alignment. In *Proc. of the IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, 2011.

[9] Jon W. Dunn, Donald Byrd, Mark Notess, Jenn Riley, and Ryan Scherle. Variations2: Retrieving and using music in an academic setting. *Communications of the ACM, Special Issue: Music information retrieval*, 49(8):53–48, 2006.

[10] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011.

[11] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[12] Özgür İzmirli and Gyanendra Sharma. Bridging printed music and audio through alignment using a mid-level score representation. In *Proceedings of the 13th International Society for Music Information Retrieval Conference*, Porto, Portugal, 2012.

[13] Mark S. Melenhorst, Ron van der Sterren, Andreas Arzt, Agustín Martorell, and Cynthia C. S. Liem. A tablet app to enrich the live and post-live experience of classical concerts. In *Proceedings of the 3rd International Workshop on Interactive Content Consumption (WSICC) at TVX 2015*, 06/2015 2015.

[14] Marius Miron, Julio José Carabias-Orti, and Jordi Janer. Audio-to-score alignment at note level for orchestral recordings. In *Proc. of the International Conference on Music Information Retrieval (ISMIR)*, Taipei, Taiwan, 2014.

[15] Meinard Müller, Frank Kurth, and Michael Clausen. Audio matching via chroma-based statistical features. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, London, Great Britain, 2005.

[16] Bernhard Niedermayer and Gerhard Widmer. A multipass algorithm for accurate audio-to-score alignment. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, Utrecht, The Netherlands, 2010.

[17] Matthew Prockup, David Grunberg, Alex Hrybyk, and Youngmoo E. Kim. Orchestral performance companion: Using real-time audio to score alignment. *IEEE Multimedia*, 20(2):52–60, 2013.

[18] Christopher Raphael. Music Plus One and machine learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2010.

[19] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv:1412.6806*, 2014.

[20] Verena Thomas, Christian Fremerey, Meinard Müller, and Michael Clausen. Linking Sheet Music and Audio - Challenges and New Approaches. In Meinard Müller, Masataka Goto, and Markus Schedl, editors, *Multimodal Music Processing*, volume 3 of *Dagstuhl Follow-Ups*, pages 1–22. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2012.