SIGHT TO SOUND: AN END-TO-END APPROACH FOR VISUAL PIANO TRANSCRIPTION

A. Sophia Koepke[†], *Olivia Wiles*[†], *Yael Moses*[‡], *Andrew Zisserman*[†]

[†]VGG, Department of Engineering Science, University of Oxford
[‡]The Interdisciplinary Center, Herzliya

ABSTRACT

Automatic music transcription has primarily focused on transcribing audio to a symbolic music representation (e.g. MIDI or sheet music). However, audio-only approaches often struggle with polyphonic instruments and background noise. In contrast, visual information (e.g. a video of an instrument being played) does not have such ambiguities. In this work, we address the problem of transcribing piano music from visual data alone. We propose an end-to-end deep learning framework that learns to automatically predict note onset events given a video of a person playing the piano. From this, we are able to transcribe the played music in the form of MIDI data. We find that our approach is surprisingly effective in a variety of complex situations, particularly those in which music transcription from audio alone is impossible. We also show that combining audio and video data can improve the transcription obtained from each modality alone.

Index Terms— visual music transcription, automatic music transcription, music information retrieval, deep learning

1. INTRODUCTION

Automatic music transcription (AMT) describes the process of automatically transcribing raw data – typically audio information – into a symbolic music representation (e.g. music notation or MIDI data). Such technology can be used to transcribe music when improvising or deliberately composing, making it easily reproducible. However, AMT from audio alone is challenging in multiple situations, such as in the presence of multiple notes or instruments or when there is background noise. While digital instruments automatically transcribe music using sensors rather than audio (e.g. a digital piano uses keypress sensors to write MIDI data), acoustic instruments are typically not equipped with such sensors.

In this paper, we propose an end-to-end deep learning approach that uses only visual information for transcribing piano music while ignoring audio cues, i.e. visual music transcription (VMT). We obtain pseudo ground-truth data to train our framework using an audio-based method.

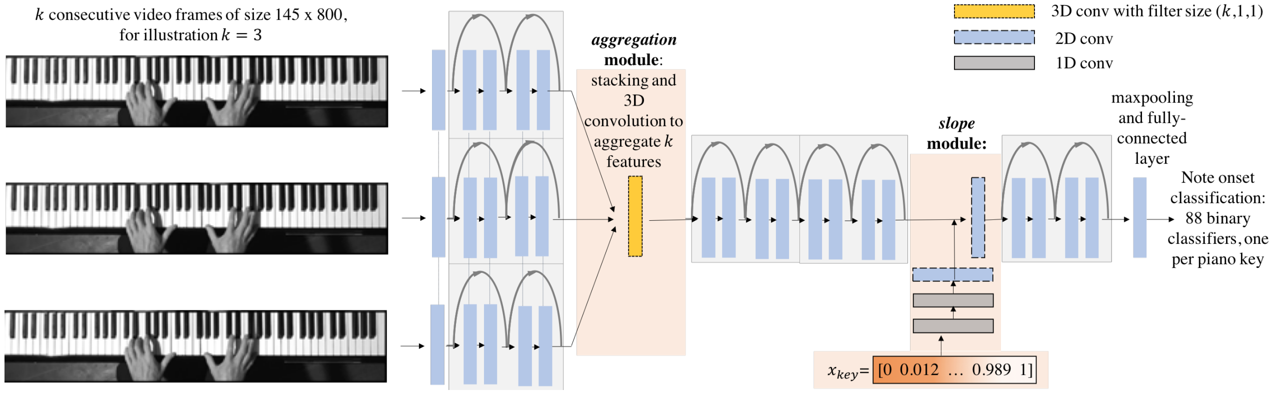
Using visual cues alone for music transcription is possible because simply watching a pianist play reveals information about the notes being produced. For example, the positioning of the hand and keys reveals information about the keys being pressed. Furthermore, the motion between frames provides localisation information about the onset of notes. Thus, it is reasonable to expect that musical audio information can be extracted from purely visual data. Using video removes the ambiguities that arise from relying on audio alone when multiple notes sound simultaneously. However, this is a challenging task since the fingers may move without pressing a key and keypresses can be occluded by the hands.

Given a video of a pianist playing, we automatically predict the pitch onset events (i.e. which and when keys are being pressed) in each video frame. We can then stitch onset events together to extract a music transcription for an entire video.

This enables a number of possible applications. An obvious use case is to transcribe silent piano videos. Also, in a similar manner to lip reading in the speech domain improving speech recognition, note onset estimation from visual information can improve audio music transcription compared to using audio alone.

2. RELATED WORK

Music transcription from visual information. Most previous methods for transcribing piano music from visual data alone are designed for constrained settings; they rely on detecting pressed keys and do not make use of temporal information in terms of hand and finger motion. [1, 2] use RGB images and require difference images between the background and current video frame to detect hands and piano keys. This is difficult to obtain when the illumination changes across the video or when shadows appear. [2] add an illumination correction step in their pipeline, but the authors report limitations for drastic light changes or vibrations of the camera or piano. [3] adds depth information, which enables velocity prediction. [4] also use depth cameras to identify key presses for a piano tutoring system. [5] predicts per-frame key presses; however, their set-up is quite constrained and can only predict a single key press per frame. Our method on the other hand, does not require depth information or background images and can deal with illumination, shadow changes, and vibration of the camera or piano. A few works have tackled VMT for other instruments (e.g., guitar [6], violin [7, 8], and clarinet [9, 10]). [11, 12] fuse visual and audio information



k consecutive video frames of size 145 x 800, for illustration $k = 3$
aggregation **module**: stacking and 3D convolution to aggregate k features
3D conv with filter size (k,1,1)
2D conv
1D conv
slope **module:**
maxpooling and fully-connected layer
Note onset classification: 88 binary classifiers, one per piano key
x_{key}= [0 0.012 … 0.989 1]

Fig. 1: An overview of our network architecture. Note onset prediction from k video frames of piano playing. Our models use $k = 5$. The network architecture is based on the ResNet18 architecture. The activations from k consecutive input frames are aggregated using a 3D convolution (*aggregation* module). x_{key} is a vector that is passed into the *slope module* which encourages the network to preserve spatial information at later stages.

together for guitar and piano transcription respectively.

Our work is most similar to [13] and [12], both of which use learning-based approaches to VMT. [13] presents a multi-step pipeline that requires significant preprocessing: given a processed crop of a single key, their Convolutional Neural Networks (CNNs) predict whether it has been pressed. [13] relies on key presses that are clearly visible from video frame differences. This is not the case when there is video jitter, instrument vibrations, low-resolution video data, or video recorded from directly above the keyboard. We cannot compare our method on [13]'s data as their provided MIDI and video information is not aligned. However, we evaluate our data on more challenging and varied music pieces and settings. [12] present a deep learning approach that uses both audio and visual information to detect key presses. They only demonstrate their method on high-quality videos (recorded at 60 fps) and simple pieces (e.g., piano exercises that have at most one note per hand at the same time). We compare our visual only performance to [12]'s on their data in the experiments.

Music transcription from audio information. [14] provides a detailed review of AMT methods, including those for piano. We use [15]'s *Onsets and Frames* framework to obtain pseudo ground-truth to train our networks.

3. NETWORK ARCHITECTURE

In this section we introduce a spatio-temporal model architecture (see Fig. 1) for performing VMT. The model is tasked to predict note onsets (a note onset is the start of a note – for piano playing this coincides with the pressing of a piano key). It uses temporal information by way of the *aggregation* module and maintains spatial information through the *slope* module.

Overview. Our model is based on the ResNet18 architecture [16]. Given 5 consecutive grayscale video frames, the model is tasked to predict all note onsets occurring within $\pm 32ms$ around the middle video frame. For a video frame rate of 25 fps, 5 frames cover $0.2s$. The 5 input frames are each passed through the first ResNet18 block (with shared weights).

Aggregation module. This module allows the model to make use of temporal information (e.g. the motion of the hand between frames) to determine whether a note is being pressed down (an onset). The output features of size $64 \times 73 \times 400$ ($d \times h \times w$) from the first ResNet18 block are aggregated by stacking them and passing them through a $5 \times 1 \times 1$ 3D convolution, resulting in a channel-wise temporal weighted average of the activations corresponding to the input frames.

Slope module. Classification CNNs are designed to be invariant to spatial positioning. However, in our case spatial localisation is essential, as the location of the hand within the image gives a large amount of information as to the octave and thereby the actual note being played. To allow the network to preserve spatial information, we also pass as an input a slope vector $x_{key} \in [0, 1]^{88}$, which contains 88 linearly spaced valued between 0 and 1 to represent the relative position of a key on the keyboard. This constant slope vector x_{key} is passed through two 1D convolutional layers, with filter size 3 and padding of 1, expanded to size $64 \times 10 \times 50$ before being concatenated to the output of the third ResNet18 block of size $256 \times 10 \times 50$. The concatenated activations are then passed through another convolutional layer with filter size (3×3) and padding of 1 resulting in features of size $256 \times 10 \times 50$ before being passed through the rest of the ResNet18 model.

Loss function. The outputs of the final fully-connected layer for our model are 88 probabilities, one for each of the MIDI notes that the piano covers. The models are trained by minimising a binary cross-entropy loss function for each note.

4. DATASETS AND TRAINING

4.1. Datasets

We curated two new datasets of piano playing (PianoYT and MIDI test set) for training our model and to test its generalisation capabilities. We also test our models on the Two Hands Hanon test set from [12].

PianoYT: This dataset contains over 20 hours of piano playing videos uploaded to YouTube. All videos are recorded from a top view. We split the data into 209 training/validation videos and 19 test videos. 172 of the training videos and all test videos were recorded by Paul Barton[1]. We obtain pseudo MIDI ground-truth from the audio in the video using the *Onsets and Frames* framework [15].

MIDI test set: In order to evaluate how robust our method is, we also test on 8 recorded videos of an amateur pianist that does not appear in the training set. For this, we recorded data with actual MIDI ground truth using a phone camera. The MIDI was recorded with a digital piano and then aligned with the audio of the recorded video. It consists of a variety of piano pieces (e.g. BWV 778, Schumann Op.15 No.1, Hanon exercises 1 and 5).

Two Hands Hanon test set: The third evaluation dataset is the *Two Hands Hanon test set* in [12] (i.e. Hanon exercises 1 and 5). This dataset contains less challenging pieces with fewer chords and the notes are within a smaller range than those in the MIDI test set.

4.2. Training details

The models are trained on the PianoYT training set using pseudo ground-truth MIDI. Video frames were extracted at their native frame rate. We performed a visual registration procedure resulting in a resized crop of 145 × 800 pixels such that the keyboard is fully visible and roughly in the same location within the crops.

Because of the relative sparsity of onset events, we reweigh training examples in each batch (i.e. class balancing) such that the weight of onset events is equal to that of non-onset events. The models are trained in PyTorch [17] using the Adam optimizer [18] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and batchsize of 24. The initial learning rate is set to 0.001 and training was stopped when the validation loss plateaued. The classification threshold was set to 0.4 using the validation set for all models. For data augmentation, we resize the crops to 150×805 and randomly crop 145×800 pixel regions. In addition to spatial jitter, we jitter the brightness and add Gaussian noise with a factor of 1% of the mean value of the image to 40% of the training images.

5. EXPERIMENTS

We evaluate our model in multiple settings. First, we demonstrate that we can indeed extract onsets from visual information alone (section 5.1). We then demonstrate that this is useful in the case of corrupted audio (section 5.2) and that it can be used to produce MIDI and thereby the audio corresponding to the entire video (section 5.3).

Metrics: We report precision, recall, accuracy, and F_1 scores for the onset estimation on different test sets. For details about the calculation of these metrics, see [19]. For the PianoYT and the MIDI test set, we report note-level metrics. For the Two Hands Hanon test set, we (similar to the authors of [12]) report frame-level metrics after finetuning to not just predict onsets, but also sustained notes.

5.1. Visual pitch onset estimation

We test how well our model (*ResNet* + *aggregation* + *slope*) can extract onsets from visual information alone. We also perform a model ablation study to demonstrate the utility of the aggregation and slope modules by comparing to two baselines: (i) *ResNet* is a ResNet18 model that takes as input 5 frames; the network's first layer is modified to have 5 channels. This model has neither the temporal aggregation nor the slope modules. (ii) *ResNet* + *aggregation* is a ResNet18 model with temporal aggregation after the first ResNet block but without the slope module.

Results: For the test set of the PianoYT dataset, the estimated MIDI prediction is compared to the pseudo-ground truth. In addition to that, we test our models on videos with actual MIDI ground truth (MIDI test set). Finally, in order to compare to [12], we report results on their Two Hands Hanon test set. Results are given in Table 1.

We see that our custom additions to the ResNet18 backbone architecture (e.g. temporal aggregation and slope module) improve our model's performance. Furthermore, there is a large difference in results when testing on the MIDI test set as opposed to [12]'s Two Hands Hanon test set. The MIDI test set contains more difficult music pieces than [12]'s Two Hands Hanon test set, as more chords are played and a much wider range of notes is covered. In order to bridge the domain gap between our training data (PianoYT dataset) and [12]'s Two Hands Hanon test set (after removing radial distortion of the images), we finetuned our models on their training set. We obtain a frame-level note accuracy (pressed key accuracy) of 87.43%, outperforming their best model that uses audio and visual information. Our results demonstrate the generalisability of our model both to unseen pianists and to more challenging pieces as compared to other methods.

5.2. Audio-visual pitch onset estimation for noisy audio

In Table 2, we demonstrate that using audio and visual information together can be useful especially when the audio is

[1] https://www.youtube.com/user/PaulBartonPiano

Model	Prec	Rec	Acc	F_1-score
PianoYT test set				
ResNet	61.40	67.59	50.29	63.72
ResNet+aggregation	**63.86**	67.87	52.20	65.26
ResNet+aggregation+slope	62.23	**73.00**	**53.33**	**66.63**
MIDI test set				
ResNet	65.26	42.82	36.86	49.94
ResNet+aggregation	72.83	52.44	44.97	59.57
ResNet+aggregation+slope	**74.76**	**73.08**	**59.59**	**72.91**
Two Hands Hanon test set from [12]				
ResNet †	92.36	86.25	80.27	88.53
ResNet+aggregation †	92.55	**93.69**	86.96	92.76
ResNet+aggregation+slope †	**93.07**	93.40	**87.43**	**93.00**
[12]‡ 2-stream w/ Multi-Task	-	-	75.37	-

Table 1: Precision, recall, accuracy and F_1-score for pitch onset estimation on the PianoYT test set, the MIDI test set and [12]'s Two Hands Hanon test set for our model (ResNet + aggregation + slope) and two baselines (ResNet, ResNet + aggregation). † fine-tuned on the training set from [12] (after removing radial distortion). ‡ Pressed key accuracy taken from [12] for their best performing model that takes both, audio and visual information as input.

corrupted or mixed with other sounds or noise.

The performance of the Onsets and Frames framework [15] decreases drastically when the audio is noisy. Mixing the PianoYT test set with other piano audio with a signal-to-noise ratio of 1 results in an F_1 score of 67.52% compared to the pseudo-ground truth obtained for the clean audio. In order to see how we can improve the pitch onset estimation using audio and visual information together, we train a 3-layer perceptron that combines the visual and audio information. It takes the concatenatation of [15]'s 512-dimensional final feature vector from the noisy PianoYT audio with our final feature vector (ResNet+aggregation+slope) as input. As can be seen in Table 2, this results in a significantly improved F_1 score of 81.82%, outperforming the audio-based and our visual based method which achieved an F_1 score of 66.73% on this data. This demonstrates that using our model to leverage visual information is clearly beneficial for obtaining note onset information.

5.3. Producing MIDI

We combine the outputs that our model produces for every 5-frame window to re-create the audio of a full video. For a given video, we pass all outputs from our model through a Gaussian filter ($\sigma = 5$) to add temporal smoothing and threshold the smoothed signal resulting in a binary signal for every note which can be saved as MIDI data. Fig. 2 shows spectrograms for one of the test videos in our MIDI test set computed from the generated and the ground-truth audio (synthesized from MIDI) respectively.

In this example, we notice that the generated audio is able to capture the rough structure of the piece and correctly predicts most of the notes. Yet, there remains room for improvement as the timing of the note onsets sometimes is slightly off. Our model was trained to only predict note onset events. However, it tends to predict multiple onset events (e.g. not just at the very beginning of a note) as it is not trained to learn the notion of a note ending.

Model	Prec	Rec	Acc	F_1-score
Noisy audio PianoYT test set, SNR $= 1$				
Audio to note onsets [15]	63.10	80.12	58.93	67.52
Audio-visual MLP (ours + [15])	**87.32**	**78.20**	**71.97**	**81.82**

Table 2: Precision, recall, accuracy and F_1-score for pitch onset estimation for the noisy PianoYT test set where the clean audio is mixed with other piano audio. The performance of the audio to onset estimation suffers with added noise. Audio and visual features are ingested by the MLP which combines visual features from our model with audio results from [15], and results in a significant improvement.

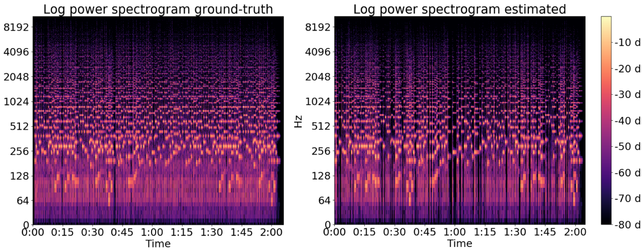


Fig. 2: Spectrogram comparison for generated audio from our MIDI prediction (right) and ground truth (left) for a test video. Our model's MIDI prediction captures the structure of the music piece and predicts most of the ground truth notes correctly. Note that our model gives sparser predictions than the ground truth (e.g. darker vertical lines appear around 1:00 min in the spectrogram on the right).

6. DISCUSSION

We proposed an end-to-end deep-learning framework to tackle the problem of transcribing piano music from visual data alone. Our system predicts note onset events given a top-view video of a person playing the piano. We demonstrated this on different test sets which vary in difficulty. Here, we focussed on piano data but our method could be extended to any other instrument with a spatial layout similar to the piano (e.g. organ, harpsichord, marimba, harp, etc.). We trained our frameworks with pseudo ground-truth data but it would be interesting to extend our framework by training on actual ground truth data with accurate note offset information.

Furthermore, work should be done to allow for different viewpoints and to better exploit the temporal information between distant frames.

Acknowledgements
This work is supported by the EPSRC programme grant Seebibyte EP/M013774/1: Visual Search for the Era of Big Data. We thank Ruth Fong for help with smoothing the output.

7. REFERENCES

[1] Potcharapol Suteparuk, "Detection of piano keys pressed in video," *Dept. of Comput. Sci., Stanford Univ., Stanford, CA, USA, Tech. Rep*, 2014.

[2] Mohammad Akbari and Howard Cheng, "Clavision: visual automatic piano music transcription.," in *NIME*, 2015.

[3] Albert Nisbet and Richard Green, "Capture of dynamic piano performance with depth vision," https://albertnis.com/resources/2017-05-10-piano-vision/Nisbet_Green_Capture_of_Dynamic_Piano%20Performance_with_Depth_Vision.pdf.

[4] Seungmin Rho, Jae-In Hwang, and Junho Kim, "Automatic piano tutoring system using consumer-level depth camera," in *International Conference on Consumer Electronics (ICCE)*. IEEE, 2014.

[5] Souvik Sinha Deb and Ajit Rajwade, "An image analysis approach for transcription of music played on keyboard-like instruments," in *Proceedings of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing*. ACM, 2016.

[6] Shir Goldstein and Yael Moses, "Guitar music transcription from silent video," in *BMVC*, 2018.

[7] Bingjun Zhang, Jia Zhu, Ye Wang, and Wee Kheng Leow, "Visual analysis of fingering for pedagogical violin transcription," in *Proceedings of the 15th ACM international conference on Multimedia*. ACM, 2007.

[8] A. Sophia Koepke, Olivia Wiles, and Andrew Zisserman, "Visual pitch estimation," in *Sound and Music Computing Conference*, 2019.

[9] Alessio Bazzica, J.C. van Gemert, Cynthia C.S. Liem, and Alan Hanjalic, "Vision-based detection of acoustic timed events: a case study on clarinet note onsets," in *Proc. of the First Int. Workshop on Deep Learning and Music*, 2017.

[10] Pablo Zinemanas, Pablo Arias, Gloria Haro, and Emilia Gomez, "Visual music transcription of clarinet video recordings trained with audio-based labelled data," in *ICCV Workshop*, 2017.

[11] Christian Dittmar, Andreas Männchen, and Jakob Abeber, "Real-time guitar string detection for music education software," in *14th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, 2013.

[12] Jangwon Lee, Bardia Doosti, Yupeng Gu, David Cartledge, David Crandall, and Christopher Raphael, "Observing pianist accuracy and form with computer vision," in *Proc. WACV*, 2019.

[13] Mohammad Akbari, Jie Liang, and Howard Cheng, "A real-time system for online learning-based visual transcription of piano music," *Multimedia Tools and Applications*, vol. 77, no. 19, 2018.

[14] Emmanouil Benetos, Simon Dixon, Zhiyao Duan, and Sebastian Ewert, "Automatic music transcription: An overview," *IEEE Signal Processing Magazine*, vol. 36, no. 1, 2018.

[15] Curtis Hawthorne, Erich Elsen, Jialin Song, Adam Roberts, Ian Simon, Colin Raffel, Jesse Engel, Sageev Oore, and Douglas Eck, "Onsets and frames: Dual-objective piano transcription," in *Proc. ISMIR*, 2018.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proc. ICCV*, 2016.

[17] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, "Automatic differentiation in PyTorch," in *NeurIPS Autodiff Workshop*, 2017.

[18] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[19] Mert Bay, Andreas F. Ehmann, and J. Stephen Downie, "Evaluation of multiple-f0 estimation and tracking systems.," in *Proc. ISMIR*, 2009.