Observing Pianist Accuracy and Form with Computer Vision

Jangwon Lee[1] Bardia Doosti[1] Yupeng Gu[1] David Cartledge[2] David J. Crandall[1]
Christopher Raphael[1]
[1] School of Informatics, Computing, and Engineering, Indiana University Bloomington
[2] Jacobs School of Music, Indiana University Bloomington
{leejang,bdoosti,yupgu,djcran,docartle,craphael}@indiana.edu

Abstract

We present a first step towards developing an interactive piano tutoring system that can observe a student playing the piano and give feedback about hand movements and musical accuracy. In particular, we have two primary aims: (1) to determine which notes on a piano are being played at any moment in time, (2) to identify which finger is pressing each note. We introduce a novel two-stream convolutional neural network that takes video and audio inputs together for detecting pressed notes and fingerings. We formulate our two problems in terms of multi-task learning and extend a state-of-the-art object detection model to incorporate both audio and visual features. We also introduce a technique for identifying fingerings if pressed piano keys are already known. We evaluate our techniques on a new dataset of multiple people playing several pieces of different difficulties on an ordinary piano.

1. Introduction

Learning to play a musical instrument is a common lifelong goal for many people. Unfortunately, it can also be out of reach: traditional music pedagogy involves regular, one-on-one interaction with a skilled teacher, which can be expensive or impossible for those who live with a limited budget or in rural areas. While online learning platforms such as Coursera [1] deliver high-quality courses, they have proven most effective for subjects like introductory computer science and mathematics, which are traditionally taught in lectures that can be readily captured on video and delivered to a large number of students.

Effective automated or online music education, in contrast, requires interactive systems that can observe a student's performances and give feedback on how to improve. While systems for music tutoring have been studied for some time [4, 6, 14, 14, 23], most of these require special electronic instruments that can record the notes that a student plays, for example through MIDI (Musical Instrument Digital Interface). Not only do these electronic instruments require an up-front investment, but they are also limited in the type of feedback they can provide: learning to play the piano, for example, requires not just hitting the right notes, but also using proper technique including posture and fingering. Learning improper technique may prevent a student from advancing properly and may cause injury, and bad habits can be very difficult to un-learn [10].

To make music instruction more affordable for more people, we want to develop automated systems that can observe a student playing any piano — perhaps a second-hand acoustic piano, for example, or one available in a local church or community center — and give feedback on both technique and musical accuracy, using only common computer hardware such as a laptop. As a starting point, in this paper we try to estimate, based on both video and audio data: (1) *which piano keys* the student is pressing at any moment in time, and (2) *which fingers* they are using to press those keys. A music tutoring system could then collect these observations over time to reconstruct the sequence of notes they play, including both pitches and durations, and how their fingering compared to those recommended by course materials.

The first of these problems — keypress detection — could be easily collected by the MIDI interface of a digital piano, but we want to handle acoustic instruments as well. This could also be addressed through analyzing audio with Automatic Music Transcription (AMT) [5,7,9,33], but much of this work considers only monophonic instruments, since recognizing multiple notes sounding simultaneously (as is common with piano) is a challenge. The second of these problems is even more difficult. We could require pianists to wear gloves with joint sensors, but these are expensive and would restrict natural hand motion. Depth cameras could help detect hand pose, but are also additional hardware that would need to be purchased by the student.

In this paper, we explore the idea of using audio and video data collected from an ordinary consumer laptop to observe both the notes a pianist plays and the fingerings they

978-1-7281-1975-5/19/$31.00 ©2019 IEEE
DOI 10.1109/WACV.2019.00165

use to play them. We consider two approaches in particular. We first introduce a novel two-stream Convolutional Neural Network that takes video and audio inputs together for detecting pressed notes and fingerings. We formulate these two problems as object detection with multi-task learning rather than standard image classification, because it reduces the search space for detecting pressed notes and identifying fingers. In particular, we extend the Single Shot MultiBox Detector (SSD) [20] to consider both audio signals and image frames to resolve ambiguities caused by finger or key occlusions, and design the model to focus on a single octave and hand to reduce the search space. Second, we apply an existing deep pose detector [28] to the fingering detection problem, assuming that note presses have been accurately identified. These two techniques offer complimentary strengths and weaknesses: the first is trained end to end, based on raw video and audio data, while the latter uses a hand-designed pipeline, but benefits from the additional data used to train the pose detector. We report experiments measuring recognition accuracy on a dataset of several pieces of varying difficulty played by multiple pianists, and demonstrate that our approaches are able to detect pressed piano keys and the piano player's fingerings with an accuracy higher than baselines.

2. Related Work

2.1. Intelligent Musical Instrument Tutoring

There is a growing body of literature that applies artificial intelligence technology to teaching musical instruments, including guitar [4], piano [6], and violin [34]. The purpose of these systems is to help students learn to play an instrument by guiding them through series of lessons, and then testing the student's comprehension by evaluating how well they can play new pieces of music. Much of the current literature on intelligent music tutoring pays particular attention to audio processing for analyzing the user's performance [8, 23], although many of these systems require specifically designed instruments and controllers [8]. Recent developments in Automatic Music Transcription (AMT) open the possibility that notes played by an acoustic instrument could be detected based on audio [5, 33], but we believe that an effective tutoring system must also be able to observe and give feedback on technique, such as fingering and hand positioning.

2.2. Computer Vision in Music Analysis

Computer vision can play an important role in providing proper feedback about a student's technique in playing an instrument. It also can help resolve ambiguities in the audio signals caused by complex interacting harmonics of polyphonic instruments like the piano. Akbari *et al.* created a four-stage image processing pipeline based on Hough transforms [15] for piano keypress detection [2]. Takegawa *et al.* attached color markers to the pianist's fingernails, and then applied a simple color-based technique with some musical rules for analyzing the pianist's finger movements [32]. Johnson *et al.* used a depth camera with Histograms of Oriented Gradients (HOG) features for detecting pianist hand posture [16]. However, there have been few attempts at integrating computer vision with audio signals to complement the limitations of each feature. Although a few studies have investigated multimodal fusion for music analysis [24, 35], their approaches are difficult to generalize to other musical instruments due to hardware requirements [35] or application-specific system design [24].

2.3. Deep Learning in Music Analysis

Deep learning has emerged as a powerful tool for many AI applications, for everything from object detection [20, 25] to learning motor control policies for robotic applications [17]. It also has become popular in Music Information Retrieval (MIR) research, and many researchers have applied deep learning for various applications such as automatic music transcriptions of drum [33], piano [12, 27], and chords [36], as well as for music recommendation [19]. Li *et al*. [18] used a CNN followed by Long Short-Term Memory (LSTM) units to "convert" audio into animations of how a simulated musician might play that music on their instrument. Shlizerman *et al*. [26] also produced body posture for piano and violin with LSTM units. Most deep learning approaches in the field of music analysis, however, have only focused on audio signals, and only a few deal with multimodal fusion for music analysis [22].

3. Approach

Our objective is to detect, at any moment in time, which piano keys are pressed and to identify which finger is pressing each of these keys. We consider these two problems in sequence.

3.1. Detecting Piano Keypresses

We could formulate the pressed key detection problem as image classification, with the task of assigning (to each video frame) a label indicating which notes are pressed among the 88 piano keys. We could then use a state-of-the-art image classification network (e.g. [37]), and train on a dataset of people playing piano with labeled ground truth. Such an approach could eventually achieve reasonable performance, but would likely require a very large amount of training data to see all reasonable combinations of keypress events. Moreover, such a formulation would not exploit other major sources of evidence like audio signals and hand movements, both of which can provide information about which notes are being played.

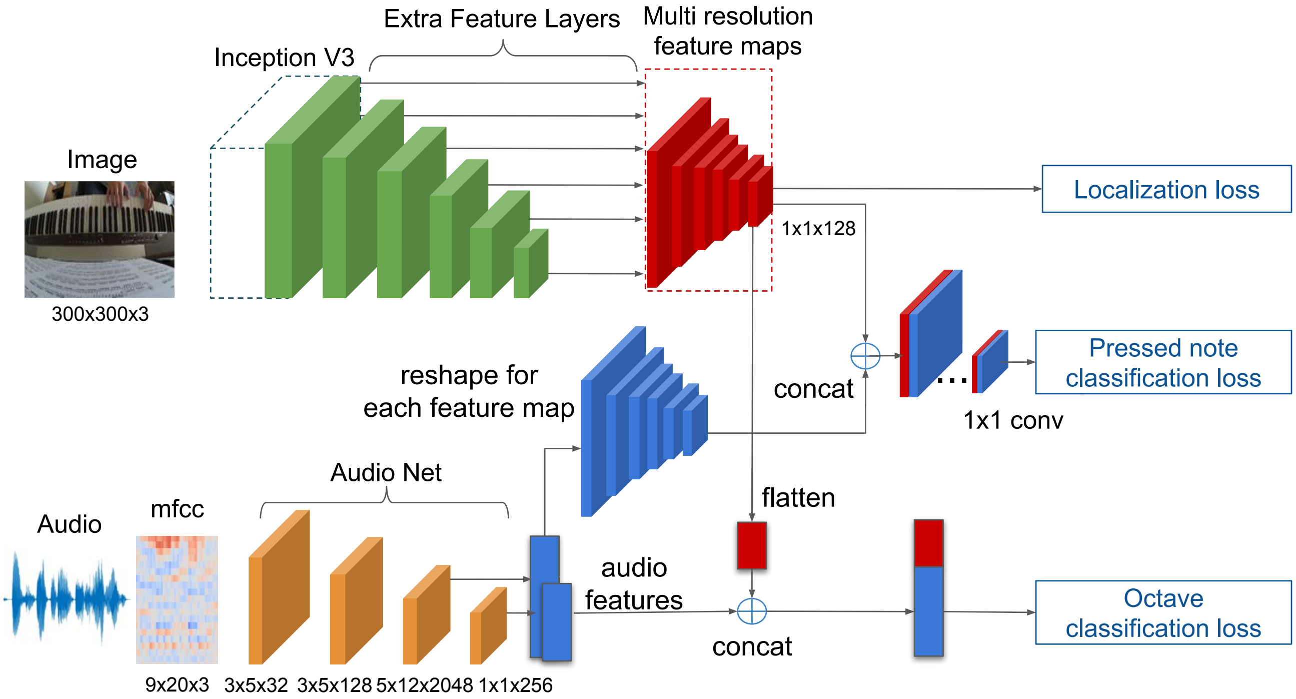


Figure 1. Outline of our two-stream architecture. The top row is the original SSD model with the different base network to handle visual stream input. We simply replace the VGG16 [29] with the Inception V3 [31] for getting more elaborate feature maps. The bottom row is a four-layer CNN to handle audio stream. We employ MFCC for audio feature extraction, and take a late fusion approach to integrate the audio and visual feature vectors. Since the audio features do not have the same spatial information as the visual features, we concatenate them along the depth axis for each multi-resolution feature map after reshaping the audio features, and do not use the audio features to compute localization loss. Our model is designed to focus on the piano key movements in single octaves, thus reducing the label space from 88 keys to 12 keys.

3.1.1 Architecture

Instead of simply applying a standard image classification model, we thus formulate the problem as multi-task learning with audio-visual data fusion. Our model focuses on the movements of piano keys in a single octave (which contains 12 notes, 7 white and 5 black) and uses audio signals corresponding to the current image frame to boost the performance of the classifier. Some important principles behind our approach are the following: (1) Each complete octave on the piano looks identical, differing only in its location with respect to the piano as a whole; (2) audio signals help resolve visual ambiguity caused by finger or key occlusions; and (3) visual features help resolve aural ambiguities caused by the interaction of complex harmonics.

Figure 1 shows the overall architecture of our model for analyzing pianist accuracy and form. We extend the state-of-the-art convolutional object detection network (SSD [20]) for multi-task learning by adding an audio stream. We define three tasks to identify key presses: (1) localization to delimit octave segments of the piano, (2) pressed piano note classification to identify played keys within a single octave, and (3) octave classification to identify which octaves are played at any given moment. Our model takes two inputs: an image frame of a person playing the piano (taken from above the keyboard), and a feature map representing the audio signal corresponding to the image frame.

The audio feature map is constructed from 20-dimensional Mel-Frequency Cepstral Coefficients (MFCCs) features [21] for 100 millisecond segments of video (which correspond to 6 consecutive frames of video recorded at 60 frames per second). We obtain 9 temporal feature sets with 100 ms for the window size, and then compute the first and second order derivatives of the MFCC features to construct three channels analogous to those of an RGB image. Each constructed audio feature map thus has dimensionality $9 \times 20 \times 3$. The videos are recorded at a resolution of 1920×1080 from a camera directly above the piano keyboard, but we resize the original image frames to 300×300 in preprocessing.

To integrate both visual and audio features, we take a late fusion approach which concatenates two feature vectors immediately before the final score functions. We extract audio features from the 3rd and 4th layers of the audio net, and then concatenate audio features along the depth axis of the multi-resolution image feature maps by reshaping them to have the same size and dimensionality. We do this because the proposed model separately predicts the confidences for each default box in SSD, and the audio features should be the same for all image subregions (since audio is related to the entire image, not just a subset). Once audio-visual data are concatenated, we employ 1×1 convolution to incorporate all features into the final decision.

We also employ audio-visual data fusion for octave classification. Since octave classification is not related to the size of bounding boxes, we only use the last map from each data stream's multi-resolution feature maps to predict one octave category at a time. We do not use audio features for localization within single octaves, as these are more difficult to reliably associate with a given octave.

3.1.2 Training

We extend the original objective function in SSD for handling multi-task learning. The extended objective function consists of three loss functions: (1) localization loss (L_{loc}), (2) pressed piano note classification loss (L_{key}), and (3) octave classification loss (L_{oct}). The overall objective function is a weighted sum of these losses:

$$L(x, y, c_{key}, c_{oct}, l, g) = \frac{1}{N}(L_{key}(x, c_{key}) + \alpha L_{loc}(x, l, g)) + \beta L_{oct}(y, c_{oct}) \tag{1}$$

where N is the number of matched bounding boxes, x is a binary indicator (0 or 1) for matching the default box to the ground truth box of the ground truth pressed piano note classification label of category p within a single octave, y is a binary indicator for matching the input image frame with the ground truth octave classification label of category q, c_{key} and c_{oct} indicate confidence scores of pressed piano note classification in single octave and octave classification respectively, and l and g represent the locations of the predicted box and the ground truth box. We now define these three losses in detail.

The keypress classification loss is a sigmoid function (instead of the softmax of the original SSD) for multi-class, multi-label classification,

$$L_{key}(x, c_{key}) = -\sum_{i \in Pos}^{N} x_{ij}^{p} \log(\hat{c}_i^p) - \sum_{i \in Neg} \log(\hat{c}_i^0)$$
$$\text{where } \hat{c}_i^p = \frac{1}{1 + \exp(-c_i^p)}. \tag{2}$$

Here, i and j represent the box number (i-th and j-th) of the default box and the ground truth box respectively.

For the octave loss, we likewise use a sigmoid function with cross-entropy loss,

$$L_{oct}(y, c_{oct}) = -y^q \log(\hat{c}^q) - (1 - y^q)\log(1 - \hat{c}^q)$$
$$\text{where } \hat{c}^q = \frac{1}{1 + \exp(-c^q)}. \tag{3}$$

For the localization loss, we use the original localization loss, which is a Smooth L1 function, to regress location parameters of the predicted bounding boxes,

$$L_{loc}(x, l, g) = \sum_{i \in Pos}^{N} \sum_{m \in \{cx, cy, w, h\}} x_{ij}^{k} \text{smooth}_{L1}(l_i^m - \hat{g}_i^m)$$
$$\hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx})/d_i^w \qquad \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy})/d_i^h$$
$$\hat{g}_j^w = \log\left(\frac{g_j^w}{d_h^w}\right) \qquad \hat{g}_j^h = \log\left(\frac{g_j^h}{d_i^h}\right), \tag{4}$$

(cx, cy) indicates the offsets for the center of the default bounding box d, and w and h represent its width and height.

We set the weight terms α and β to 1 by cross validation.

3.2. Fingering Identification

3.2.1 Architecture

To identify which finger is pressing each note on the keyboard, we frame the problem as object detection and employ the same architecture as above, except without octave classification or the audio stream (since audio signals provide no information about fingering). This problem is more challenging because the fingers move rapidly and are comparatively small objects to detect. Furthermore, the appearance of a given finger may be subject to variation caused by hand posture changes and occlusion.

We propose to use the output of the first network—the "key-pressed" information—to reduce the search space for detecting fingers. We assume that the input videos are recorded from a similar camera angle, and then use the key pressed information to crop the input image frames based on a rough locations of the pressed key on the piano. For example, we can remove the very left and right sides of the input image if our network estimates that middle C is being played. In this paper, we crop out about 30% of the original input image as a preprocessing phase, and then feed the resulting cropped images to the network to identify fingering.

3.2.2 Training

One problem with the object detection formulation is that it requires more expensive annotations because the network needs bounding boxes during training. In order to

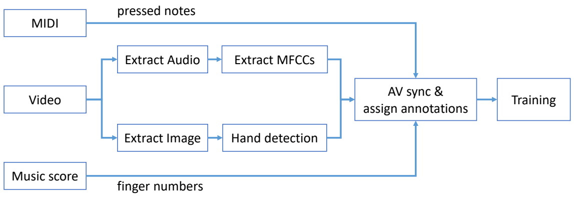


Figure 2. The pipeline to create our dataset of Hanon Exercises.

reduce this annotation cost, we first train our network on the publicly-available dataset of Bambach *et al.* [3], which contains hundreds of hand instances with pixel-level ground truth annotations in a variety of environments (albeit none including pianos). We then apply that trained model on our piano data to produce our own dataset for finger identification. We assume that hands are located nearby in adjacent image frames, and thus use the bounding box from the previous frame when the network trained on the EgoHands dataset fails to detect hands in any given frame. We then manually labeled each bounding box with the finger number(s) that are currently pressing keys, and train the proposed network on this dataset.

3.2.3 Using Key Pressed Information

The above technique can be trained end-to-end from video given ground truth finger labels, but this annotated data can still be costly to collect. We thus also explored an alternative technique. If we assume that we know exactly which keys are being pressed on the piano, either through MIDI or a computer vision technique such as that described above, we can then localize the coordinates of each key on the keyboard and each finger tip of the hand, and then estimate which fingers are pressing which keys by calculating the nearest fingertip (in image coordinate space) to each pressed key. In more detail, first we obtain the coordinates of each key based on a Hough transform of Sobel edges [30], using the approximate width of each key (which can be estimated based on the width of the keyboard). To infer the positions of the finger tips, we use Simon *et al.*'s hand keypoint detector [28] which estimates the positions of 21 joints of the hand. We then finally select the nearest finger to each pressed key as our estimate of which finger is pressing it.

4. Experimental Results

We conducted two sets of experiments to evaluate the proposed architecture and to compare to various baselines. In the first set of experiments, we focus on testing the accuracy of our model for pressed piano notes detection. In the second set of experiments, we evaluate the accuracy of our approach for identifying fingers used to press notes.

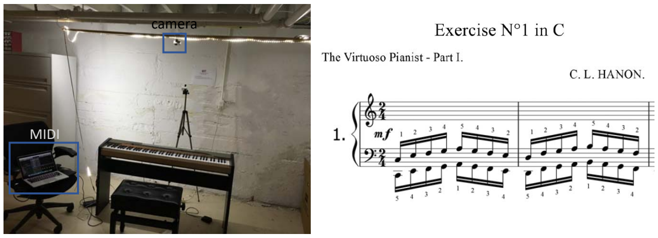


Figure 3. Our piano room with an experimental setup and a sample Hanon exercise. We recorded MIDI files while a person was playing the piano and then aligned them with music scores for annotating our dataset.

4.1. Datasets

We created new datasets of people playing the piano for training and testing our techniques. Figure 2 shows the pipeline that we used for generating our dataset given three different input files recorded while people played the piano: videos, MIDI files, and music scores. First, we extracted image frames and audio from the input video, and then applied the pre-trained hand detector on image frames to obtain bounding boxes of fingers. For the audio stream, we extracted MFCC features with 100 ms windows, and converted these into multi-channel images based on the first and second order derivatives, as described above. We extracted keypress information from MIDI synchronized with the video to create keypress ground truth labels. Finally, we manually annotated finger numbers.

We collected two datasets, one consisting of piano exercises and the other consisting of real pieces. For the former, we used several Hanon Exercises [11], which have a long history as technique-building exercises. Hanon exercises do not present music of great artistic interest, but they cover a wide range of the keyboard and systematically uses the entire hand with frequently repeating patterns, yielding naturally balanced and diverse data. Hanon is beneficial for finger identification ground truth collection for a similar reason, since the exercises all have finger numbers denoted in the score and are designed to exercise all five fingers evenly. Figure 3 shows our experimental setup, as well as the first few bars of Hanon Exercise number 1.

In particular, we collected two types of data with Hanon. **One Hand Hanon** contains a total of 10 videos of a person playing Hanon exercises 1 through 5 with one hand, and each video clip ranges from 50 to 120 seconds. The pianist played each exercise twice, once with the left hand and once with the right. In total, we collected 35,332 frames with ground-truth annotations. We split this dataset into five sets according to the exercise number, trained our model on exercises 1 to 3 (23,555 frames), and used the remaining exercises 4 and 5 (11,777 frames) for evaluation. **Two Hand Hanon** contains a total of 5 videos of a person play-

Method	Accuracy
Using a Single Sensory Input:	
Video Only (Inception V3 [31])	56.43%
Audio Only (Audio Net)	41.10%
Video and Audio Data Fusion:	
Two-stream w/o Multi-Task (Inception V3 + Audio Net)	75.05%
Multi-Task Learning to focus on a Single Octave:	
Video Only w/ Multi-Task (Inception V3 + Focusing a Single Octave)	82.37%
Two-stream w/ Multi-Task (Ours, Inception V3 + Audio Net + Focusing a Single Octave)	**85.69%**

Table 1. Pressed key detection accuracy on One Hand Hanon.

Method	Accuracy
Using a Single Sensory Input:	
Video Only (Inception V3 [31])	46.33%
Audio Only (Audio Net)	39.63%
Video and Audio Data Fusion:	
Two-stream w/o Multi-Task (Inception V3 + Audio Net)	65.33%
Multi-Task Learning to focus on a Single Octave:	
Video Only w/ Multi-Task (Inception V3 + Focusing a Single Octave)	65.82%
Two-stream w/ Multi-Task (Ours, Inception V3 + Audio Net + Focusing a Single Octave)	**75.37%**

Table 2. Pressed key detection accuracy on Two Hands Hanon.

ing the same Hanon exercises 1 through 5 with both hands, and each video clip ranges from 50 to 240 seconds. In total, we collected 51,596 frames with ground-truth annotations. Similar to the One Hand dataset, we split this dataset into five sets with regard to the exercise number, and trained our model on exercises 2 to 4 (36,115 frames) and evaluated on exercises 1 and 5 (15,481 frames). Note that this is a multi-label dataset for octave classification since the Hanon Exercises have both hands playing at the same time, but in different octaves.

Our second, more difficult dataset consists of six real pieces often played by new pianists, at six different levels of difficulty: Minuet by Alexander Reinagle, Minuet by Johann Sebastian Bach, Russian Polka by Michael Glinka, Melodie by Robert Schumann (Album für die Jugend Op.68 No.1), and Robert Volkmann Op.27 No.9. We had five pianists record the pieces, including two professionals, two with medium skill, and one beginner. We collected 65 minutes of video also at 60 fps, and annotated them frame by frame for both notes and fingering using a combination of MIDI and manual labeling.

4.2. Evaluation

4.2.1 Pressed Key Detection

We first evaluated the accuracy of the proposed architecture for pressed key detection. We compared our proposed multi-task video-audio fusion model with four different baselines. **(i) Video Only** uses video frames as input; it thus formulates the problem as a standard image classification problem using Inception V3 [31]. **(ii) Audio Only** uses just audio signals without image frames, using our Audio Net which is a four layer CNN described in Figure 1. **(iii) Two-stream w/o Multi-Task** uses audio-visual data fusion without our multi-task formulation which is designed to focus on the key movements in a single octave. This baseline uses Inception V3 and the Audio Net to handle each sensory input separately, and then takes a late-fusion approach for integrating both inputs. **(iv) Video Only w/ Multi-Task** uses our multi-task formulation, but only uses video for detecting the pressed keys.

We first measured classification accuracy, which is a percentage of correctly classified images among all image frames of the test set. Since our approach and the Video Only w/ Multi-Task baseline can produce more than one output at a time with different bounding boxes, we picked the single predicted box that had the highest confidence score in each image, and then assigned its predicted class (pressed note) to the image for both approaches. In addition, we only accepted the image as a true positive when the image was correctly classified in terms of both octave and note within the octave. We trained our model using RMSProp [13] for 50k iterations with learning rate 10^{-4}, 0.9 momentum, 0.9 decay, and batch size 32.

Table 1 shows the pressed key detection accuracy on the One Hand Hanon dataset. We observe that our two-stream approach with multi-task learning outperforms all baselines in terms of accuracy. Our model yielded 85.69% pressed key detection accuracy, and the experimental results confirm that our multi-task formulation and additional audio stream are able to boost the performance of the classifier.

We next measured classification accuracy on the Two Hands Hanon dataset. We used the same baselines, training strategies, and hyperparameters for this experiment. However, we replaced the final softmax function of all baseline approaches with the sigmoid function since the Two Hands dataset requires multi-label classification (because it contains two labels for each image). We picked the top two predicted boxes based on the confidence scores, then assigned each image their predicted classes. Table 2 shows the pressed notes detection results on this Two Hands Hanon dataset. Once more, the results confirm that our approaches outperform the performance of the baselines in terms of classification accuracy. This suggests that our multi-task formulation with audio-visual data fusion helped resolve ambiguities that can be caused when using a single stream only.

4.2.2 Fingering Identification

Our second set of experiments evaluated the accuracy of identifying fingers used to press piano keys. For these experiments, we trained our object detector on the One Hand Hanon for detecting fingers, with and without a preprocessing stage to crop the input image frame based on key pressed information. We then measured used finger detec-

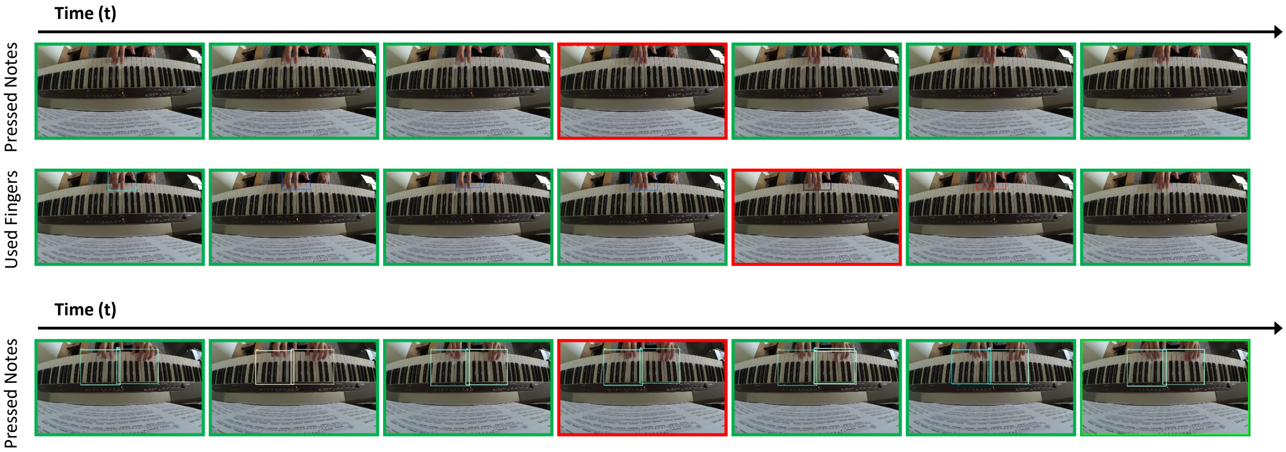


Figure 4. Examples of our pressed piano note detection and fingering identification results. **Top:** Pressed notes detection on our One Hand Hanon dataset. **Middle:** Fingering identification on our One Hand Hanon dataset. **Bottom:** Pressed notes detection on our Two Hands Hanon dataset. The small bounding boxes in the bottom images show a single octave of piano keys. The green border images indicate the correct predictions, whereas the red border images show the failure cases. The frames were captured every 100 milliseconds.

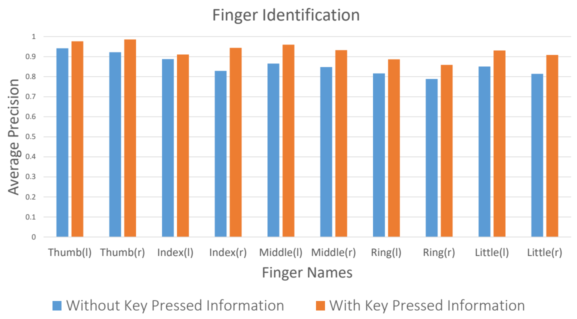


Figure 5. The accuracy of the proposed approach to identify fingers used to press piano notes. We evaluated the performance of two approaches (with key pressed information vs. without key pressed information) in terms of average precision. The x-axis shows finger names and (l) and (r) indicate the left and right hand respectively.

tion accuracy in terms of average precision.

Figure 5 presents finger detection accuracy of the two approaches, showing that pressed note information is beneficial. The network achieved better accuracy for all fingers in terms of average precision when it used key pressed information. The model with the pre-processing step yielded 0.929 for mean average precison (mAP), significantly more accurate than the model without key pressed information (0.856 in mAP). Figure 4 shows some qualitative results for both note detection and finger identication in Hanon exercise #5. Most false detections arise during the transition from one key to another.

Finally, we tried using our model trained on Hanon to detect fingering on a completely different style of music, Chaconne by Yiruma, to see how it responds to new finger postures. The results of our fingering identification are shown in Figure 6. The performance decreases, of course, due to several challenges including black keys (whereas the Hanon Exercises are only on white keys) and some large chords which require very different hand postures from those found in Hanon.

4.2.3 Fingering Identification with Hand Pose

Our third set of experiments evaluated the accuracy of identifying fingers with a stand-alone hand pose estimator [28], as opposed to a network trained end-to-end, under the assumption that we know exactly which keys are being played at any moment in time.

We first applied this technique to our Hanon dataset, and it could detect the correct fingerings in 99% and 96% of the frames in one hand and two hands Hanon, respectively. We then applied this technique to our second dataset of six real piano pieces of different difficulties played by multiple players, and it detected the correct fingerings in more than 90% of the frames in each video. Table 3 presents detailed results in terms of precision, recall, and F_1 score, showing that the system has the lowest accuracy in the fourth finger (ring finger). The confusion matrix in the Figure 7 shows that most confusion for the system happens between adjacent fingers, and especially between the third and fourth fingers. Figure 8(a) presents a sample failure case, where the third and the fourth finger are very close to each other. Crossovers also can confuse the system, for example when

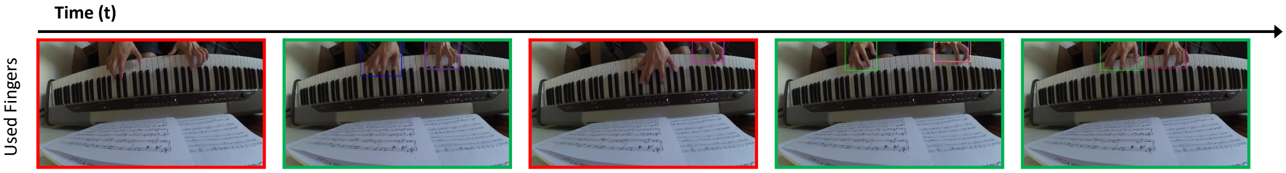


Figure 6. The accuracy of our fingering identification on a completely different style of music. In this case, our network often fails to identify fingers used to press keys since the training dataset is played only on white keys and does not contain large chords.

Finger	Precision	Recall	F_1 Score
Thumb	0.939	0.943	0.941
Index	0.958	0.928	0.943
Middle	0.964	0.843	0.899
Ring	0.708	0.850	0.773
Little	0.916	0.941	0.929

Table 3. Precision, Recall and F_1 score of finger identification with hand pose.

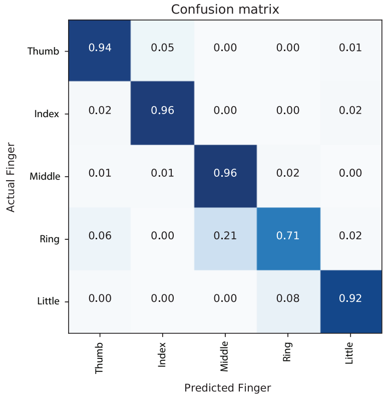
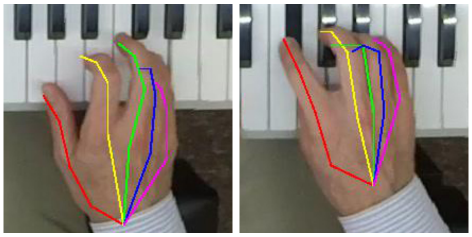


Figure 7. Confusion matrix of finger detection.



(a) (b)

Figure 8. Two failure cases in detecting the fingering. (a) Confusion between middle and ring finger because of vicinity of fingers. (b) Confusion between thumb and index finger due to a crossover and resulting occlusion.

the index finger crosses over (and thus occludes) the thumb in order to play a note. Figure 8(b) shows an example of this.

5. Conclusion

In this paper, we proposed a novel two-stream convolutional neural network to determine which notes on a piano are being played at any moment in time, and to identify the fingers used to press those notes. We formulated this problem as multi-task learning with audio-visual fusion, and characterized the accuracy of various variants of the technique.

The methods used for this study may be applied to other musical instruments for building an interactive musical instrument tutoring system. Our current approach does not utilize temporal information, which may help resolve ambiguities and remove other errors. Moreover, we trained our network separately on different datasets for each task (pressed piano key detection and fingering identification); building a unified model for both tasks and utilizing temporal information to improve perception accuracy would be a natural progression of this study as future work. Furthermore, research is also needed to provide interactive real-time feedback to the student.

6. Acknowledgements

This work was supported by the Indiana University Office of the Vice Provost for Research under the Faculty Research Support Program, as well as by the National Science Foundation (CAREER IIS-1253549). We would like to thank Mark De Zwaan for assisting with data collection and for helpful discussions.

References

[1] http://www.coursera.com/.

[2] M. Akbari and H. Cheng. Real-time piano music transcription based on computer vision. *IEEE Transactions on Multimedia*, 2015.

[3] S. Bambach, S. Lee, D. J. Crandall, and C. Yu. Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[4] M. Barthet, A. Anglade, G. Fazekas, S. Kolozali, and R. Macrae. Music recommendation for music learning: Hotttabs, a multimedia guitar tutor. In *The ACM Conference Series on Recommender Systems (RecSys) Workshop on Music Recommendation and Discovery*, 2011.

[5] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri. Automatic music transcription: challenges and future directions. *Journal of Intelligent Information Systems*, 2013.

[6] R. B. Dannenberg, M. Sanchez, A. Joseph, P. Capell, R. Joseph, and R. Saul. A computer-based multi-media tutor for beginning piano students. *Journal of New Music Research*, 1990.

[7] A. Dessein, A. Cont, and G. Lemaitre. Real-time polyphonic music transcription with non-negative matrix factorization and beta-divergence. In *International Society for Music Information Retrieval Conference (ISMIR)*, 2010.

[8] C. Dittmar, E. Cano, J. Abeßer, and S. Grollmisch. Music Information Retrieval Meets Music Education. In *Multimodal Music Processing*, 2012.

[9] C. Dittmar and D. Gärtner. Real-time transcription and separation of drum recordings based on nmf decomposition. In *DAFx*, 2014.

[10] W. Gieseking and K. Leimer. Piano technique, 1972.

[11] C. L. Hanon. *The virtuoso pianist: in sixty exercises for the piano*, volume 1. G. Schirmer, 1911.

[12] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck. Onsets and frames: Dual-objective piano transcription. *arXiv preprint arXiv:1710.11153*, 2017.

[13] G. Hinton, N. Srivastava, and K. Swersky. Overview of mini-batch gradient descent. *Neural Networks for Machine Learning - Lecture 6a, University of Toronto*, 2012.

[14] S. Holland. Artificial intelligence in music education: A critical review. In *Readings in Music and Artificial Intelligence*, 2000.

[15] P. V. Hough. Method and means for recognizing complex patterns, Dec. 18 1962. US Patent 3,069,654.

[16] D. Johnson, I. Dufour, D. Damian, and G. Tzanetakis. Detecting pianist hand posture mistakes for virtual piano tutoring. In *International Computer Music Conference (ICMC)*, 2016.

[17] S. Levine, C. Finn, T. Darrell, and P. Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 2016.

[18] B. Li, A. Maezawa, and Z. Duan. Skeleton plays piano: online generation of pianist body movements from midi performance.

[19] D. Liang, M. Zhan, and D. P. Ellis. Content-aware collaborative music recommendation using pre-trained neural networks. In *International Society for Music Information Retrieval Conference (ISMIR)*, 2015.

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. Berg. SSD: Single shot multibox detector. In *European Conference on Computer Vision (ECCV)*, 2016.

[21] P. Mermelstein. Distance measures for speech recognition, psychological and instrumental. *Pattern Recognition and Artificial Intelligence*, 1976.

[22] S. Oramas, O. Nieto, F. Barbieri, and X. Serra. Multi-label music genre classification from audio, text, and images using deep features. *arXiv preprint arXiv:1707.04916*, 2017.

[23] G. Percival, Y. Wang, and G. Tzanetakis. Effective use of multimedia for computer-assisted musical instrument tutoring. In *ACM International workshop on Educational multimedia and multimedia education*, 2007.

[24] A. Perez-Carrillo, J.-L. Arcos, and M. Wanderley. Estimation of guitar fingering and plucking controls based on multimodal analysis of motion, audio and musical score. In *International Symposium on Computer Music Multidisciplinary Research*. Springer, 2015.

[25] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018.

[26] E. Shlizerman, L. Dery, H. Schoen, and I. Kemelmacher-Shlizerman. Audio to body dynamics. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[27] S. Sigtia, E. Benetos, and S. Dixon. An end-to-end neural network for polyphonic piano music transcription. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 2016.

[28] T. Simon, H. Joo, I. Matthews, and Y. Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*, 2015.

[30] I. Sobel and G. Feldman. A 3x3 isotropic gradient operator for image processing. *a talk at the Stanford Artificial Project in*, 1968.

[31] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[32] Y. Takegawa, T. Terada, and S. Nishio. Design and implementation of a real-time fingering detection system for piano performance. In *International Computer Music Conference (ICMC)*, 2006.

[33] R. Vogl, M. Dorfer, G. Widmer, and P. Knees. Drum transcription via joint beat and drum modeling using convolutional recurrent neural networks. In *International Society for Music Information Retrieval Conference (ISMIR)*, 2018.

[34] J. Yin, Y. Wang, and D. Hsu. Digital violin tutor: an integrated system for beginning violin learners. In *ACM International Conference on Multimedia*, 2005.

[35] B. Zhang and Y. Wang. Automatic music transcription using audio-visual fusion for violin practice in home environment. 2009.

[36] X. Zhou and A. Lerch. Chord detection using deep learning. In *International Society for Music Information Retrieval Conference (ISMIR)*, 2015.

[37] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le. Learning transferable architectures for scalable image recognition. *arXiv preprint arXiv:1707.07012*, 2017.