Towards a Dataset for Estimation of Keyboard Fingerings

Filip Danielsson[1][0009−0007−9188−3943] and Pavel Pecina[2][0000−0002−1855−5931]

[1] Czech Technical University in Prague, daniefi1@fit.cvut.cz
[2] Charles University, pecina@ufal.mff.cuni.cz

Abstract. In this paper, we describe a system that combines keyboard detection, key segmentation, piano transcription, and hand-tracking into a pipeline for general piano performance videos recorded from an overhead perspective. Keyboard localization was performed by a fine-tuned YOLOv8 model on manually annotated frame samples, followed by a procedure for segmenting and labeling the keys. Keyboard fingerings were predicted using a distance metric between the hands and key boundaries, achieving an 82% accuracy. Utilizing the knowledge of hand positions demonstrated up to a 6.6% improvement in piano transcription F1 score.

Keywords: piano fingering, dataset, piano transcription, hand tracking

1 Introduction

Learning the piano and hand movements associated with it was a difficult multivariate task. Learners of all levels could benefit from seeking clarity on their approach from a teacher or an automatic system. Choosing appropriate keyboard fingerings was one technical aspect that could be enhanced with guidance, especially for beginners.

Videos of piano performances contain valuable information about keyboard fingerings, movement, and hand shape that could be utilized in learning systems. It has been shown that human fingering agreement was 71.4% on a subset of the PIG dataset[3]. Using just labeled notes for training fails to take the more complex reasons behind the finger choices into account and will likely never surpass the human agreement

A pipeline for extracting information about the notes, keys, and hands from readily available videos on a large scale would allow for quantitative research on pianist mechanics. The accuracy of state-of-the-art piano transcription and hand-tracking models has reached a point where such features could be produced automatically from generally available video recordings. Previous works on videos are from a single source and require ground truth MIDI files hence limiting the use cases.

As of now, no system has been found that was capable of automatically extracting and processing general overhead piano videos with varying settings and performers. One previous work was identified that has worked with extracting

data from online piano videos [2]. The authors created a dataset of piano fingerings from videos with a similar format and attempted to model these fingerings using the dataset. The work relied on video creators providing ground truth MIDI files with the videos. The amount of available videos with corresponding MIDI files was much smaller and manual work was needed to download and pair each MIDI file with the video. Finding videos with ground truth MIDI files in good lighting conditions for specific pieces might not be possible. Additionally, the authors had to synchronize the MIDI files with the videos by using a heuristic and maximizing fingering confidence scores. The system also relies on having the full 88 white key range visible, avoiding the need to estimate key labels from hand positions in a zoomed video [2].

In this work, note onsets were estimated using a transcription system from the audio [1]. MIDI estimates are based on the audio directly so synchronization will not be needed. The downside of automatic transcription, lowered transcription accuracy, was in part remedied by using the additional video input. Additionally, a more robust processing pipeline was developed, allowing for a fully automatic system for detecting and extracting relevant sections on a much broader range of videos.

2 Implementation

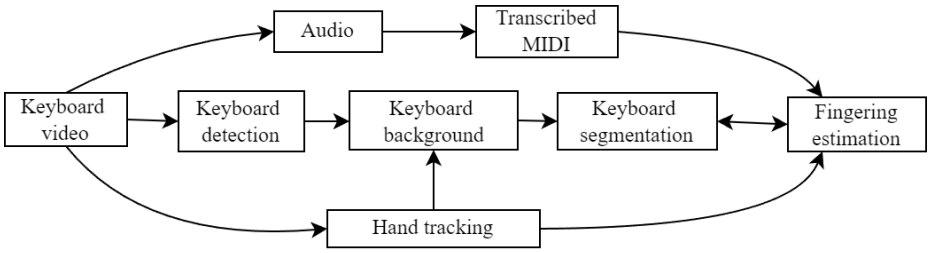

Fig. 1. A diagram of the pipeline

The data processing pipeline was designed to produce a keyboard fingering estimation from videos with an overhead keyboard as seen in Fig. 1. Initially, the audio component was extracted from the input video and analyzed for note events using automatic piano transcription [1]. The frames of the video are used for extracting hand landmarks [6] and finding sections that contain a keyboard [4]. If such sections are present, the background keyboard frame without the hands was estimated. The background frame was used to segment and label the individual keys. The fingering was estimated from the hand landmarks and key locations. The key labels can additionally be octave-corrected by minimizing finger distances.

2.1 Video Collection

The target videos this work was going to extract and process contain footage of someone playing the piano filmed from an overhead perspective. The videos can contain different types of pianos, and the camera can be zoomed into different parts. Five YouTube channels were chosen, each containing a significant number of keyboard performances. 100 of the latest videos were downloaded from each channel to create a 500-video development dataset. When possible, videos are acquired at a 30FPS frame rate and 480p resolution; 480p was enough to distinguish individual key bounds at low zoom levels visually.

The type of videos contained in this dataset varies vastly, about half of the videos contain some overhead keyboard frames. Out of those, some are performances, and others are tutorial-style videos with commentary in between playing. Some common variations of this format display cropped musical scores or a falling piano roll above the keyboard. Some videos are vlogs and might not contain any piano music at all, and others contain music filmed from a different perspective or for other instruments.

2.2 Keyboard Detection and Sectioning

Manually identifying where a keyboard was located was time-consuming and would not be possible for large sets of data. Detecting the keyboard was needed both for validating the format of sections in the video and finding the locations of each key. The detection model should work on a variety of sources, identify sections with a keyboard filmed from an overhead perspective, and provide the bounding box of the keyboard in each section.

To train and test the keyboard detection possibilities, two frames were extracted at random points in each video, creating a set of 990 sample frames, 556 of which contain a keyboard from an overhead angle and 434 do not. The 556 images containing a keyboard were manually annotated with bounding boxes. The frames were resized to fit into a 640x640 pixel boundary and divided into a 33-67% test-train split. The YOLOv8 small model [5] was fine-tuned on the training data with default parameters. It achieved a 100% detection accuracy and 0.995 mAP score with a 0.5 IOU threshold on the test set; no hyper-parameters were modified. Both the dataset and the model are released with this work.

2.3 Background Extraction

The keyboard in detected frames was usually obstructed by hands in some way. The background was estimated as the pixel-wise mean over the sections containing a keyboard without the hands. Hand-obstructed locations are removed from the calculation utilizing hand landmark data. A sum of frames was accumulated with bounds around hand landmarks excluded. By default, the background of the video contains the keyboard scene with the most frames.

2.4 Hand Tracking

Tracking the hands accurately was crucial for estimating interactions with the piano. Most of the videos present are well-lit and should work reasonably well with an off-the-shelf solution. Mediapipe hands [6] provides an efficient hand detection and pose estimation system with 3D landmark output. Some errors arise

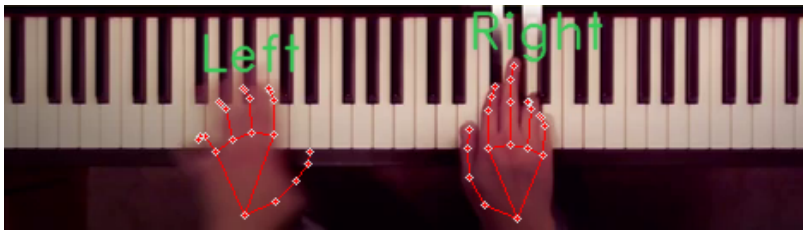

Fig. 2. Example of hand tracking on a pianist

when using this approach. The handedness prediction was sometimes wrong, causing a few frames of two hands with the same handedness. This was resolved by re-assigning left and right-handedness according to the average horizontal position similar to [2]. Additionally, fast striding movements can cause motion blur, making the algorithm ineffective for a few frames. Gaps smaller than 15 frames are therefore filled in using linear interpolation.

2.5 Keyboard Segmentation

The key segmentation algorithm receives a background estimate of a video or section and the bounding box of the keyboard. A multi-step process was employed to separate and label each key in the keyboard area. This involves converting to the HSL colorspace to distinguish black and white keys by lightness, creating an approximate mask for black keys, estimating key boundaries, matching edges to define horizontal bounds, and labeling keys based on typical piano key groupings. The vertical boundaries of white keys are derived proportionally from the black keys, and the note C4 was identified based on its position relative to the keyboard width, with octave predictions refined by considering hand locations.

Constants and thresholds used in the described process were found by experimentation on the 500 video dataset. Due to the repeating nature of the keys, it might not be possible to pinpoint the key labels from the background visuals alone. Key label estimates are hence refined using the piano transcription and hand positions. Labels will be shifted up or down by an octave (12 notes) to minimize the sum of minimum key-to-finger distances.

3 Experiments

3.1 Keyboard Fingering Estimation

Knowing the positions of the hands and keys allows for estimating the fingering used for playing each note. Experimenting with estimation approaches required

an evaluation video with corresponding ground truth MIDI key presses to isolate any errors created by the piano transcription model. A 2.5-minute long video recording was captured on a digital piano together with the MIDI output containing 1227 notes. The ground truth MIDI was aligned to the video by maximizing the F1 score to the piano transcription estimate. Ground truth fingerings were subsequently manually labeled by us for each of the 1227 notes. Labels were in the format {LR}_{12345} indicating left-right handedness and the finger number. Consistent with other works [3, 2], each note was only assigned a single fingering corresponding to the finger that caused the onset of the note. This does not capture rare cases where the fingering changes during a single note duration.

Datapoints were extracted using the pipeline and a heuristic was tested against the evaluation video. Fingering will be assigned by measuring the finger-key distance between the played key and landmarks on the hands. Onsents are taken from the transcribed midi and distances are calculated between each key corresponding to the onset and the fingertip landmark. If the fingertip F was outside the bounding box the distance was calculated as the shortest distance from F to the closest point P on a key bounding box plus the distance from P to the center of the key C. If the fingertip landmark F was inside the bounding box the distance was just the Euclidean distance to the center of the key.

The finger-key distance was computed between the fingertips on both hands in the frames that contain any note onsets. The finger with the minimal finger-key distance was the fingering label of the pressed note. Comparing the estimates with the ground truth labels of the 1227 notes results in an 82.23% accuracy.

As seen in table 1. the accuracy was also compared to several other distance metrics such as *bounds_0* where the fingertips inside the bounding box are simply assigned the distance of 0. For approximately 10% of the notes two or more fingertips have a distance of 0, in these cases, the fingertips with the lowest index are prioritized. Metrics *center* and *bottom center* measure the distance to the center of the key and the center of the bottom edge of the key.

Distance metric	**Accuracy (%)**
center	39.12
bottom center	51.51
bounds_0	82.07
bounds	82.23

Table 1. Fingering estimation accuracy for different distance metrics

3.2 Transcription correction

Piano transcription false positives commonly show up in the harmonics of a note [1], i.e. the same notes in different octaves. This type of error could be identified with hand positions as the hands are usually too far away. A distance parameter was added to correct the produced transcription. Notes distanced

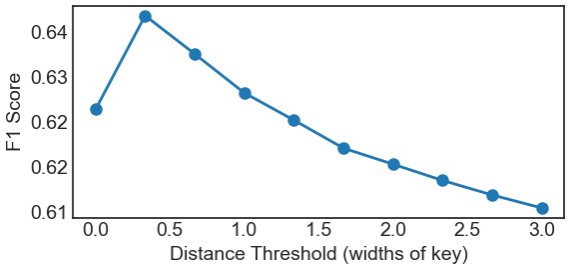

Fig. 3. Transcription accuracy vs hand distance threshold

above a certain threshold from any of the hand landmarks would be deemed erroneous and removed. The threshold distance should be measured in real units, such as the average white key width to enable usage across videos.

Removal of false positives was unlikely to change piano transcription metrics such as those for note onset/offset or velocity evaluation. The transcription correction was therefore evaluated frame-wise. The frame-wise metrics are usually produced for a time between frames corresponding to the hop length used in the preprocessing pipeline. In this case, regressing note onsets/offsets were quantized into two different frame periods 10ms and 33.3ms. 30 FPS (corresponding to a 33.3ms frame period) was the most prevalent frame rate in the channel videos and the evaluation video, in most cases this will be the maximum resolution at which hand landmarks can be obtained accurately. 10ms frames are used in the evaluation of [1], in this case, landmarks from the previous video frame are used for the correction.

	33ms frames			**10ms frames**		
	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)
No correction	47.95	72.26	57.64	47.78	71.90	57.41
Best correction	58.55	71.11	64.22	58.72	71.06	64.30

Table 2. Transcription performance on 33ms and 10ms frames

A max distance threshold of approximately 0.34 key widths performs the best on the evaluation video, reaching a top F1 score of 64.2%. This was an increase of 6.6% over the 57.6% without corrections. As seen in table 2., the improvement comes from a 10%+ increase in precision; recall was slightly lowered as some true positives can be deleted in the process. Performance on 10ms frames achieves a similar max F1 of 64.3%. Figure 3. displays the improvement of the frame-wise F1 score with 33ms frames across different values of the distance threshold.

4 Conclusion

In conclusion, this work addressed the task of automatically extracting and analyzing piano performance videos filmed from an overhead perspective. Leveraging recent advancements in computer vision and machine learning, particularly in piano transcription and hand tracking. An automatic processing pipeline was created to extract relevant data points, including key and keyboard locations. The usefulness of the data collection approach has been demonstrated by estimating fingerings with an accuracy of up to 82% using a heuristic approach. Furthermore, transcription correction techniques improve frame-wise F1 scores by 6.6%, indicating the potential of refining transcription accuracy with video data. Despite these advancements, several limitations exist, including the need for fine-tuning a hand landmark detection model for piano-specific scenarios, improving testing data, and key segmentation reliability. Source codes for the project are publicly available at https://github.com/uel/piano-video. In summary, this work represents a first step towards automating the analysis of piano performances using overhead videos, with promising avenues for further research.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Kong, Q., Li, B., Song, X., Wan, Y., Wang, Y.: High-resolution piano transcription with pedals by regressing onset and offset times. IEEE/ACM Trans. Audio, Speech and Lang. Proc. **29**, 3707–3717 (10 2021). https://doi.org/10.1109/TASLP.2021.3121991, https://doi.org/10.1109/TASLP.2021.3121991
2. Moryossef, A., Elazar, Y., Goldberg, Y.: At your fingertips: Extracting piano fingering instructions from videos (2023)
3. Nakamura, E., Saito, Y., Yoshii, K.: Statistical learning and estimation of piano fingering. CoRR **abs/1904.10237** (2019), http://arxiv.org/abs/1904.10237
4. Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A.: You only look once: Unified, real-time object detection. CoRR **abs/1506.02640** (2015), http://arxiv.org/abs/1506.02640
5. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. CoRR **abs/1612.08242** (2016), http://arxiv.org/abs/1612.08242
6. Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C., Grundmann, M.: Mediapipe hands: On-device real-time hand tracking. CoRR **abs/2006.10214** (2020), https://arxiv.org/abs/2006.10214