Multi-Pitch Estimation: Data Augmentation Techniques

David Dai, Manasi Ganti, Edward Qin, Andrey Risushkin, Edward Zhang
University of Washington
{kun02, mganti, edwardcq, risuka, ezhang8}@uw.edu

1 Introduction

For music listeners, apps such as Shazam already exist to quickly match real-world (often noisy) sound to a known, predefined set of music. However, there lacks an equivalent app to provide real-time transcription from detected pitches to sheet music, enabling musicians to perform music without publicly available scores such as improvisation. Additionally, such a tool can open up the world of music composition to a much wider audience, as aspiring composers can simply perform their compositions to obtain their scores without ever typesetting their compositions. We aim to provide such a tool with the ability to transcribe performances of any combination of instrumentation into scores in real-time.

This report represents our first step towards our final goal, as we focus on the capability to recognize the pitches of any combination of notes performed. We first start with a single piano, recognizing few-note chords. We describe our method for improving performance on real-world, noisy, and even cross-instrument data, to show how we can extend our method to support multiple instruments beyond piano.

Our project therefore needed to provide identification of the most likely pitches at any moment in an audio file, and preferably produce this in close to real time (within five minutes of the audio being recorded). We chose to develop a model that outputs a stream of pitch predictions, that a downstream application can apply smoothing and filtering to for the final transcription.

2 Problem Statement

Our problem of multi-pitch estimation can be formalized in terms of multi-label classification. Our goal is to identify which pitches are present in a single note.

The model input is a single 88×1 column of a CQT spectrogram, representing a 50ms slice of audio.

The model output is a multi-hot encoded representation of the predicted notes. Specifically, we have 89 total classes (88 classes representing each of 88 notes on a full-size standard piano and 1 class representing “silence”, or no sound). The model outputs a binary vector of size 89, where a 0 or 1 at index i indicates the absence or presence of the corresponding note, respectively.

In all experiments, our model performs standard multilabel classification and is not told beforehand how many pitches are to be expected. The forward pass of the model produces a scores vector of size 89, where a sigmoid function is applied to transform vector values between 0 and 1, and any pitch with a surpassing the score threshold of ≥ 0.5 is considered “chosen”.

In order to discuss success criteria, we utilize the following metrics. Note that we define overall precision and recall as the average precision and recall over all labels.

- Correctness: An “all or nothing” approach. If our model prediction is P and the ground truth label is L, then the prediction is correct if and only if $P_i = L_i \; \forall i$.
- Accuracy: The fraction of all samples that were predicted correctly.
- Precision: The fraction of predictions for a particular label that are correct. Formally defined as $\frac{TP}{TP+FP}$.
- Recall: The fraction of samples with ground truth label=L that we correctly predicted as L. Formally defined as $\frac{TP}{TP+FN}$.
- F1 Score: Harmonic mean of precision and recall, used as a quick way to interpret both precision and recall at the same time.

Our initial success metric was to achieve above 70% for accuracy, precision, and recall for each of 3 tasks: single note, interval, and chord classification. We describe our final results in a later section, but overall we succeed on these criteria.

3 Datasets

Throughout the training process, models were solely trained on notes and augmentations from the University of Iowa dataset. Testing was done on a dataset we gathered playing piano notes and chords in the University of Washington Music Building.

3.1 University of Iowa (Iowa) Dataset

The University of Iowa dataset [2] contains a sound sample for each key of an 88-key piano, played at three dynamics (pianissimo, mezzoforte, fortissimo). This sound sample is a clean `.aiff` file, recorded in a quiet room with high quality microphones. However, this can also be a limitation as real data likely includes noise and is not as clean as the samples in this dataset.

Using code written by our TA, Victoria Ebert [1], this set of clean notes could be combined, forming the basis of our training set. The dataset was a convenient set of all piano notes across a variety of dynamics that could be easily combined into all intervals, and into many common chords between C3 and C6, supporting a training set for multi-pitch estimation. The choice to only randomly generate chords between C3 and C6 was done to significantly reduce data processing time, especially since adding more notes exponentially increases the number of combinations. Most common chords are contained in the middle of the piano range, and do not range beyond two octaves. The combinations were generated using Victoria's helper function calling `librosa` to combine spectrograms and produce combined audio. Ultimately, this augmentation was motivated by the goal to support piano pitch transcription beyond single notes.

3.2 Music Building (MB) Dataset

In order to claim that our model can perform well in a real-world scenario, we decided to create our own test dataset. Therefore, we recorded samples of single notes, intervals, and chords played on various pianos in the UW practice rooms. The rationale behind creating a test dataset in this setting was threefold: (1) practice rooms are one of the most frequent areas where music is played, (2) the location is accessible, and (3) the noise environment is just right (not too little or too much noise). Specifically, the dataset includes:

- All 88 single notes
- All perfect intervals in the C3 to C6 range, as well as some random intervals
- Two to three dozen random chords with 3 to 5 notes each
- Silence interspersed with each of the above samples

This dataset is limited by its coverage. Due to time constraints (and respect for our neighbors in the adjacent practice rooms), we did not record all possible chords or intervals. Also, this test set is only reflective of a single environment - piano practice rooms at UW in the evening.

3.3 Data Augmentation

Once we observed poor generalization to real-world piano data (discussed in Section 4), we employed two methods of data augmentation.

1. Silence was not predicted as a class. To obtain silence from both training and testing samples, we used a `librosa` helper function to classify any 50 ms slice with less than 60 dB as silence.
2. We decided to try mixing noise into our spectrograms. This involved reusing the technique to combine chords and pairing the already-generated single notes, intervals, and chords with the noise. For each noise sample, we randomly selected an interval of the same length as the piano sound and combined them.

We collected noise from two sources: professional noise found on the internet and real noise that our team recorded at real locations. The professional noises came from samples of a bird park, clapping, footsteps, library noise, quiet city cars, room noise, and people in a hall. These samples were high quality, lasting between 30 seconds and 1 minute. The samples were chosen for their high likelihood of occurring near music played in the real world. Real noises came from recordings of ambient noises around Bellevue Square Mall, especially around areas with pianos (main walkway, restaurants, etc). The main limitation of both of these datasets is the relatively small number of noise sources and lack of diversity in recording equipment.

4 Results

4.1 Original Data

Model	Train Set	Test Set	Single Note Accuracy	Interval Accuracy	Chord Accuracy
1D-CNN	Iowa	MB	22%	1.4%	$< 1\%$
1D-CNN	MB	MB	53%	-	-
VGG	Iowa	MB	47.17%	1.9%	$< 1\%$
VGG	MB	MB	99%	-	-

Figure 1: Initial Accuracies of 1D-CNN and VGG models, trained and tested on same Pitch Type

Our initial hypothesis was that a relatively simple vision model, a 1D-CNN with 2 convolutions, would be enough to achieve 70% accuracy, precision, and recall.However, from Figure 1, we see that the model has very low accuracy across all three pitch types. Additionally, when trained on the test set, the mode was unable to overfit. This experiment was convincing evidence that a simple model is not powerful enough to capture representations of noisier data. We upgraded to a more powerful version of a 1D-CNN, a VGG (Very Deep Convolutional Network) [3], which involves many more convolutional layers.

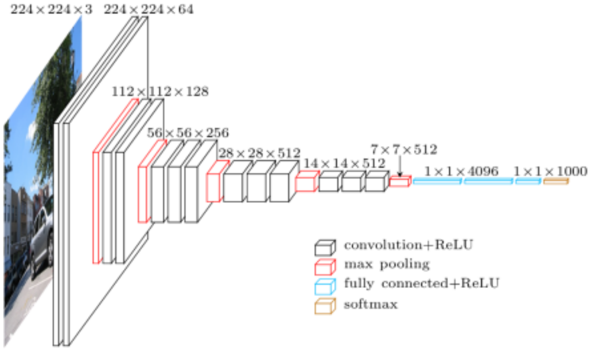

Figure 2: VGG Model Architecture

We then hypothesized that a more powerful model would be able to improve performance a reasonable amount, but might still fall short of our goal because our training data is too clean. As shown in Figure 1, the VGG does perform much better on the single note task, but achieves very little progress in the other two tasks. However, the model is able to overfit, indicating that it is large enough to capture the multi-pitch representations. And yet, we conjectured that there was still some issue with the difference between the clean Iowa data and noisy MB data.

4.2 Augmented Data

To address the issue of data mismatch, we introduced data augmentation (described in Section 3.3) and trained the VGG model on this augmented Iowa dataset, evaluated on MB test set. We can see in Figure 3 that accuracy for Interval classifications has noticeably improved from below 2% to nearly 12%. However, performance on Chords is still disappointing.

Pitch Type	Accuracy	Precision	Recall	F1 Score
Single Note	0.528	0.536	0.212	0.304
Interval	0.122	0.621	0.259	0.365
Chord	0.003	0.075	0.030	0.043

Figure 3: Performance of the VGG model trained and tested on the same Pitch Type

4.3 Joint Training

From Figure 3, we see that the recall scores are noticably low, indicating that the model has a hard time distinguishing between single notes and clusters of notes. Importantly, each result so far came from a task framed a "one-on-one" scheme of training and evaluation. For each pitch type, we only trained the model on the training set of that pitch type and only tested on the test set of that pitch type. As such, we conjectured that training the model on both single notes and intervals could help the model distinguish the two. This led to some surprising results, as indicated in Figure 4.

Pitch Type	Accuracy	Precision	Recall	F1 Score
Single Note	0.547	0.690	0.590	0.636
Interval	0.699	0.834	0.604	0.700
Chord	0.910	0.957	0.796	0.869

Figure 4: Performance of VGG model trained on augmented Single Notes and Intervals

From the table, we see that precision and recall for the single note and interval classes have risen significantly. Even more surprising, however, we also see that metrics for chord classification have also risen by an enormous amount, despite not having trained on any chords in the training set.

5 Discussion

We propose the following analysis for this latest experiment. First, the increasing recall from Single Note to Chord gives weight to the argument that the model typically struggles with distinguishing between single and multi-note samples. Secondly and most surprisingly, the increase in performance on chord classification indicates that a model trained on just single notes and intervals can extend well to general multi-pitch estimation, as the model is able to identify chords using some joint representation.

Data acquisition and labeling is expensive and time consuming. Our results imply that building a training set consisting solely of single notes and 2-note intervals augmented with noise is much more practical and inexpensive than building a train set of chords. (e.g. $\binom{88}{2} \ll \binom{88}{3}$). Training on this augmented dataset of single notes and intervals is also computationally cheap because there are much fewer batches of data.

Overall, we achieve the following results and takeaways:

1. Data augmentation is extremely important. Performing augmentation can alleviate the data mismatch between synthetic "clean" audio samples and real-world audio samples.
2. Joint training (training on multiple Note Types) is a promising way to tackle the generalizability of multi-pitch estimation. It is both less expensive and attains higher performance across all metrics when compared with only training on our chords training set.
3. We generally succeed in achieving 70% accuracy, precision, and recall for the three pitch types, except for single notes. However, future work may address this with a separate model trained specifically for single note classification.
4. Because we use a very small VGG model on small input slices, our system is very fast at inference time, with end-to-end processing from an audio file to pitch predictions for every 50 ms slice taking less than 1 second.

6 Future Work

Throughout our experiments, we have identified both overtones in all recordings and noise present in lower-quality recordings to be significant challenges to our model's accuracy. Since data augmentation and joint training were able to significantly improve our model performance, we believe increasing the number of samples in our currently limited noisy datasets in future work may lead to better generalizability. Alternatively, our current evaluation metric utilizes an "all-or-nothing" scheme where the model must accurately determine all performed pitches to be classified as "correct". Additional exploration of different evaluation metrics that provide partial credit for the model's correctness could better benefit the analysis of the model's strengths and weaknesses. Lastly, since the inseminating goal of this report was to perform automatic transcription for any combination of instrumentation, we believe future work should take the next step toward simultaneous pitch identification by expanding training to multiple instruments.

Acknowledgements

We would like to acknowledge Victoria Ebert for her spectrogram and chord generation code, as well as her support with obtaining the dataset. We would also like to thank the University of Iowa for the open access to their MIS Piano dataset. We would like to thank the UW Music Building for letting us record in the practice rooms. And finally, we thank the students in the course, the TAs, and Noah Smith for their support and feedback.

References

[1] Victoria Ebert. chordgen. `https://github.com/ebertv/chordgen`.

[2] University of Iowa Electronic Music Studios. UIowa MIS Piano Dataset. `https://theremin.music.uiowa.edu/MISpiano.html`.

[3] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

Appendix

All code and data is contained in the Shared Drive.

Dataset

- `chordgen_main`: Clone of Victoria's Repository, with regenerated intervals and chords, real noise, and raw noise
- `clean_spectrograms`: Generated clean spectrograms of single note, intervals, and chords
- `demo`: Demos of full pipeline from audio file to spectrogram to MIDI-converted mp3 of model prediction
- `extracted_data`: json-readable format of clean and noisy spectrograms
- `noisy_spectrograms`: Generated noisy spectrograms of single note, intervals, and chords
- `real_data`: Music Building and YouTube datasets

Codebase

- `extract-dataset.ipynb`: Code for extracting spectrograms into json format
- `make_midi.ipynb`: Code for full end-to-end process converting audio file to spectrogram to MIDI-converted mp3 of model prediction
- `one-to-one-exp.ipynb`: Experiment for 1D-CNN and VGG models trained and tested on the same Pitch Type
- `joint-exp.ipynb`: Experiment for VGG model on Joint Training
- `spectrogram_generation_clean.ipynb`: Code to generate clean spectrograms
- `spectrogram_generation_noisy.ipynb`: Code to generate noisy spectrograms