Optical Music Recognition with Neural Networks

Scott McCaulay

School of Informatics and Computing, Indiana University

Introduction

This project is an evaluation of a neural network as a method for performing Optical Music Recognition. The goal is to try to use a neural network alone to parse images of musical scores, with as little help as possible from other computer vision techniques.

Background

A big part of the challenge in this field is the heterogeneity of the data. This is especially true in retrieving data from scanned musical scores from different historical eras, many originally hand written, in various conditions, sizes and using inconsistent notations. This project makes no attempt to address those challenges. Since all the data utilized here will be from the same source, it will be as though the image pre-processing stage is already completed, and done flawlessly. Given this sanitized and consistent data, it may be possible for a machine learning algorithm with no understanding of the domain to produce results at least better than random guessing. That is what the project intends to test.

Typical methods of optical music score recognition use a combination of routines for individual tasks such as detection of staff lines, recognition of musical symbols, etc. There has been some work using neural networks as part of a solution, in combination with the aforementioned methods. Neural networks do not seem as widely used today for musical scores as they are for character recognition or speech. One goal of this work is to explore the possibilities and challenges in applying neural networks to this application, specifically where they can help and where they may not be as helpful.

Data and Methods

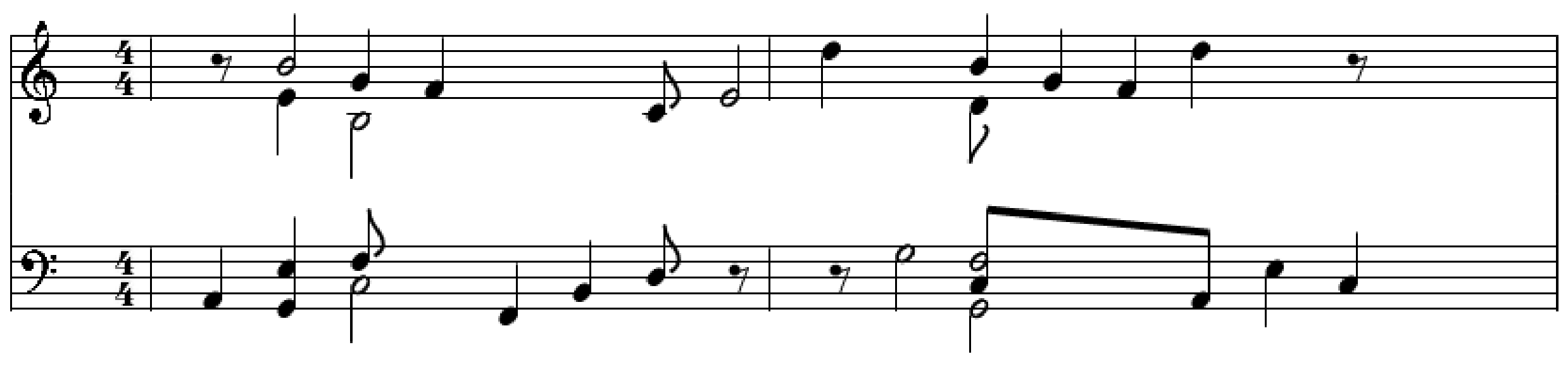

Image 1: An example Score file, a corresponding key is produced as a CSV file

Generating the Data:

Test data for this project has been generated programmatically. A set of 200 MIDI files has been generated, along with an “answer key” for each file in CSV format. Score images were generated from the MIDI files using the MIDI Sheet Music application. The images in PNG format are input to the Neural Network application, with the CSV file which contains location, pitch and duration for each note for training and testing.

Simplifying the Process:

Limitations were placed on the data to simplify recognition. All notes are in the key of C, so there are no sharp and flat icons to decipher. All files are in 4/4 time signature. Notes are within a range of E2 to E5, only three note durations are used. Each time step can have one or more notes. Given temporal placement, pitch and duration, there are 1,008 unique note events which can be present in a score.

Preprocessing the Data:

The input files are 840 x 312 pixels, making a possible 262,080 nodes in the input layer of the neural network. A preliminary process filters this data to standardize and minimize input to the neural network. First the staff positions are aligned. Second, the value of each pixel across the entire corpus is calculated. Locations that contain black pixels in more than 25% of scores are eliminated from inclusion. This greatly reduces the number of input nodes reducing memory and calculation requirements for our network. Figure 2 shows the results of this process, common values are eliminated, darker pixels in this image indicate more unique occurrences, lighter grays show areas that may be common to multiple events.

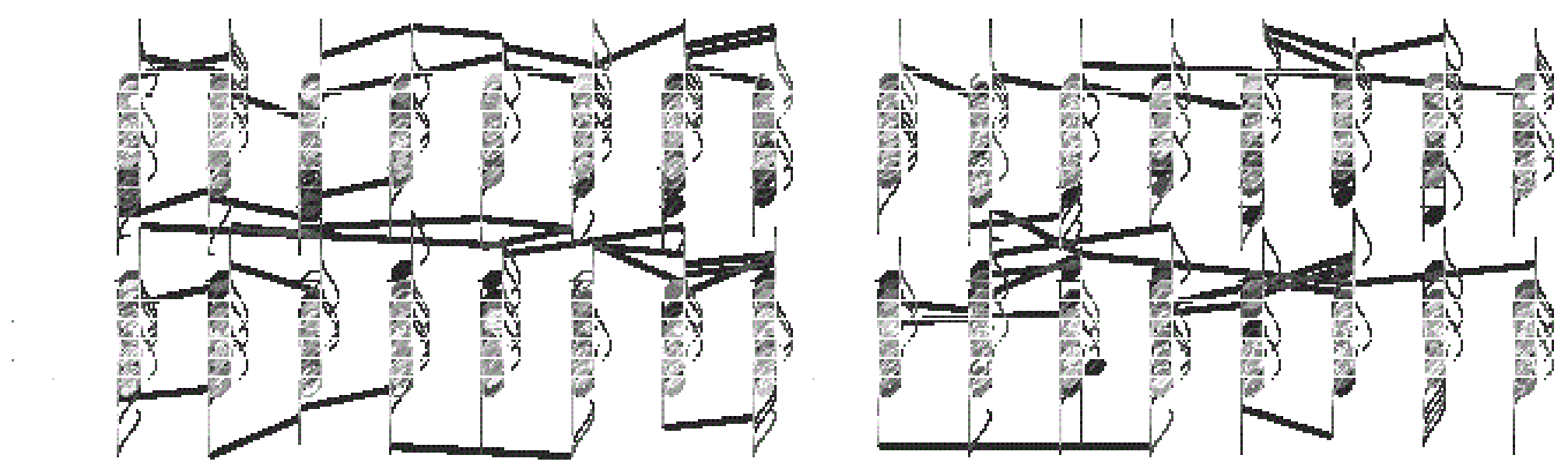

Image 2: Identifying most significant pixel locations in the corpus

The Neural Network:

The network is a simple, traditional feed-forward neural network. It has approximately 32K nodes and potentially 30 million edges, which presents a moderate challenge for calculation, but is trivial in comparison to networks developed at Google or Stanford with billions or 10’s of billions of edges. Without the pre-processing, the number of edges would have been over 250 million.

Given the temporal compartmentalization of both the input and output layers, it was possible to split the network into isolated frames so that inputs from within a frame only feed outputs in the same frame. The divisions of the frames are calculated as the midpoints between the denser clusters of pixels seen in Image 2. Along with the preprocessing of the input data, this reduces the calculation required for the network.

Output Layer:

The output layer of the neural network is a multidimensional grid representing all the unique note events. The goal of the network is to correctly calculate the cells of this grid to match exactly the notes present in the score. This is done by iteratively adjusting the weights of the network edges connecting input pixels to output cells, until the calculation of input values, adjusted by the excitatory and inhibitory weights of the connecting edges, converge on the correct values in the output layer.

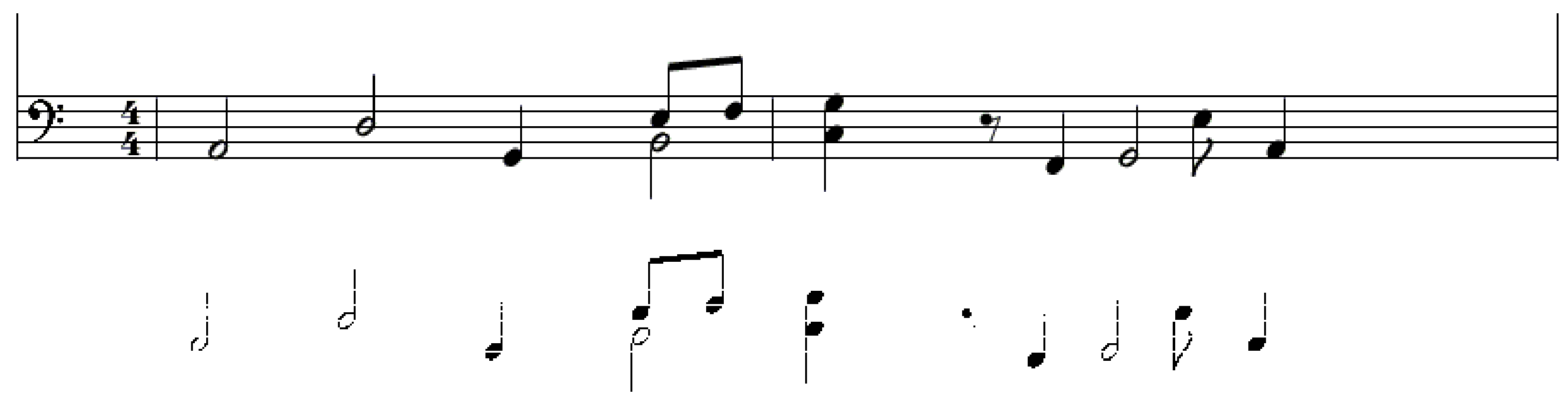

Image 3: Input layer before and after pre-processing

Results

A Benchmark for Comparison:

For Comparison purposes, 50,000 random guesses were computer generated and compared to the ground truth of the scores of the training set. Guesses were based on the likelihood of note events in the overall population. Given that the actual output layers are mostly empty (most note events are not present), random guesses based on likelihood perform well, in the mid-90% range.

As of press time, the network is still training, and adjustments are still being made to the scoring algorithm and activation functions. While progress has been slow, some success is evident, in that the network has been able to calculate a set of weights that produce outputs from the training set in which 97.43% of output nodes have the correct value. This high number may exaggerate the appearance of success, since most of the correct calculations, as in the random guesses, are made of true negative cases. Still the network outperforms the best of 50,000 random guesses by nearly 2%.

Given the heavily sanitized data, there was an expectation that performance would be much better, and it may still be with continued parameter tweaking and training. Observing the input data and evaluating the ambiguity of the pixels does highlight the difficulty of the problem, without regard to the methods used to address it.

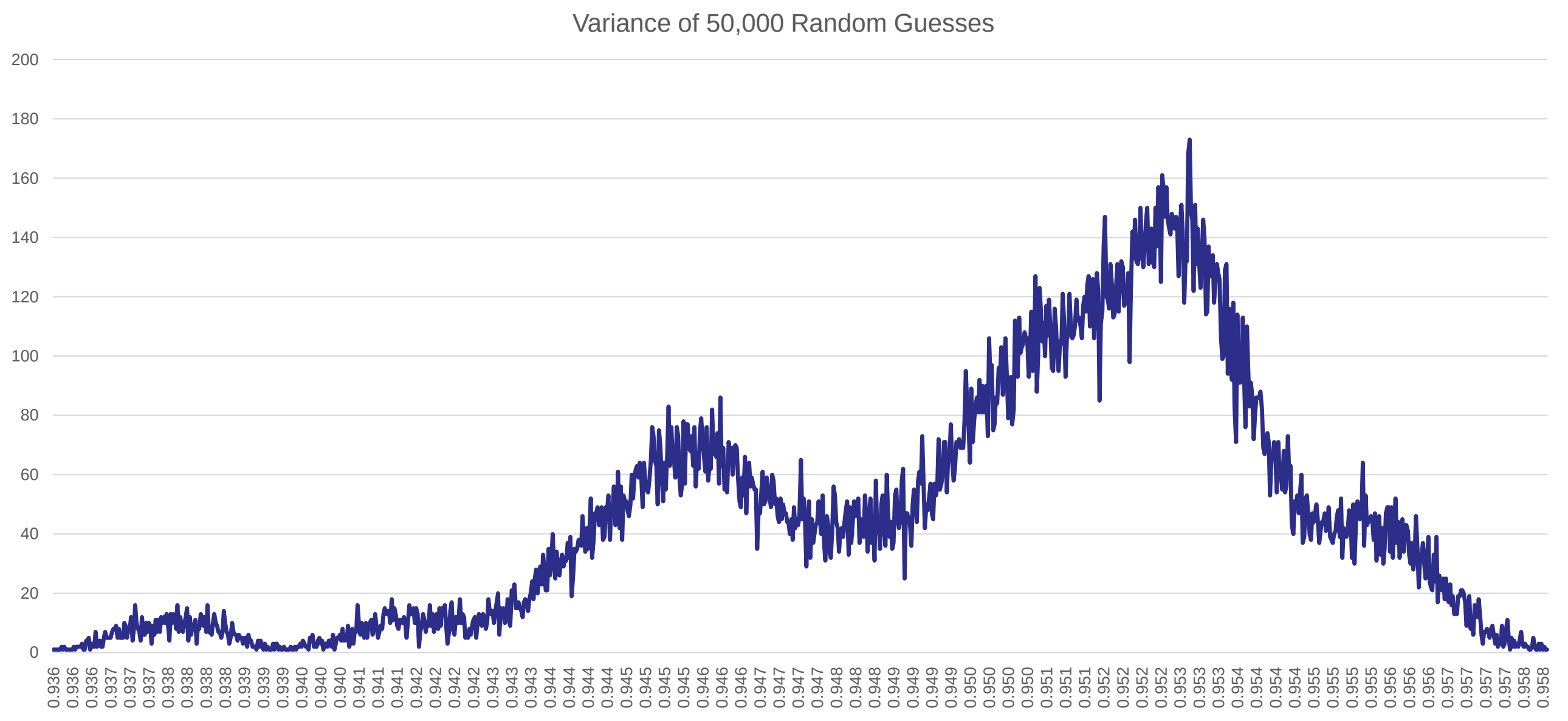

Image 2: Computer generated guesses against the population of images

Conclusions

Preliminary results indicate that a neural network could be a helpful addition to an optical music recognition toolbox, in combination with other tools. Given a uniform set of scores and adequate training data, a neural network could potentially supplement other tools to locate and identify symbols in a score. This work does highlight the ambiguity present even in machine generated audio scores with severely limited types of data. Given complex, inconsistently presented data, considerable preliminary work would be required to get adequate results from a neural network.

References

1. **Byrd, Donald, and Jakob Grue Simonsen. "Towards a standard testbed for optical music recognition: Definitions, metrics, and page images." Journal of New Music Research 44.3 (2015): 169-195.**
2. **Fornés, Alicia, et al. "CVC-MUSCIMA: a ground truth of handwritten music score images for writer identification and staff removal." International Journal on Document Analysis and Recognition (IJDAR) 15.3 (2012): 243-251.**
3. **Raphael, Christopher, and Rong Jin. "Optical music recognition on the international music score library project." IS&T/SPIE Electronic Imaging. International Society for Optics and Photonics, 2013.**
4. **Rebelo, Ana, et al. "Optical music recognition: state-of-the-art and open issues." International Journal of Multimedia Information Retrieval 1.3 (2012): 173-190.**
5. **Vaidyanathan, M. (2007-2013). "Midi Sheet Music." 2.6. 2016, from https://sourceforge.net/projects/midisheetmusic/.**
6. **Wen, Cuihong, et al. "A new optical music recognition system based on combined neural network." Pattern Recognition Letters 58 (2015): 1-7.**