MUSICAL STRUCTURE SEGMENTATION WITH CONVOLUTIONAL NEURAL NETWORKS

Tim O'Brien
Center for Computer Research in
Music and Acoustics (CCRMA)
Stanford University
660 Lomita Drive
Stanford, CA 94305
tsob@ccrma.stanford.edu

ABSTRACT

We approach the task of automatic music segmentation by musical form structure. After reviewing previous efforts which have achieved good results, we consider the rapidly evolving application of convolutional neural networks (CNNs). As CNNs have revolutionized the field of image recognition, especially since 2012, we investigate the current and future possibilities for such an approach to music, and specifically the task of structure segmentation. We implement a straightforward example of such a system, and discuss its preliminary performance as well as future opportunities. [1] [2]

1. INTRODUCTION

This paper describes our ongoing attempts to automatically segment songs according to musical song structure. To accomplish this, convolutional neural networks are trained on spectral audio features via human-annotated structural "ground truth" segment times. Our system's input is a song, and its outputs are predicted times of structure boundaries (*i.e.* the start or end or a section, such as a verse, bridge, or chorus in Western popular music terminology).

Reliable automatic music segmentation is worthwhile for several reasons. If we characterize the structures of arbitrarily large amounts of recorded music, we can use statistics to conduct musicological analysis at a huge scale. This is one aspect of the field of computational musicology [3, 5]. Perhaps by seeing the forest instead of the trees, we can gain new insight into the role of music structure as a compositional element.

Additionally, consumer applications such as music recommendation systems benefit by taking into account song structure, as it is a salient aspect of human appreciation of music. One could even employ music structure boundaries to automatically generate music "thumbnails" or summaries, short snippets of music that include examples of all the sections of the larger work (see, for example, [1]).

More broadly, the essence of this task is interesting in and of itself. Humans can perceive musical sections and their boundaries quite quickly and easily, even without prior instruction in music. However, just like image classification, natural language processing, or speech recognition, this is no easy task for a computer. This is partly because music structure is inherently tied to human perception; the ultimate judge of music structure is the human auditory and cognitive system. Like other perceptual attributes, this is unavoidably subjective.

Structural segmentation is a well-known task in the domain of music information retrieval (MIR), and has been an official task at MIREX [3] since 2009. Approaches have included self-similarity matrix evaluation [17], and flavors of unsupervised non-negative matrix factorization [8, 19], to name a couple. Typically, spectral features such as chroma (pitch classes) or MFCCs are used as the input. One popular variant is beat-synchronous time warping [9], in which temporal frames are nonuniform and dictated by beat detection, as opposed to a more typical uniform frame size.

We discuss related work in more detail in the following section. We then take a deep dive into the methods we employ here (§3), explore our dataset and the features we utilize (§4), and discuss our current results (§5). We conclude with some final remarks and look toward future work in §6.

[1] Our efforts on this project are combined jointly with CS231N (http://cs231n.stanford.edu/) and Music 364 (https://ccrma.stanford.edu/courses/364/). Blair Kaneshiro, instructor for Music 364, and Andrej Karpathy, instructor for CS231N, both agreed to this arrangement.

[2] Code for this project is available, such as it is, at https://github.com/tsob/cnn-music-structure.

[3] MIREX, the Music Information Retrieval Evaluation eXchange, is an annual competition run by ISMIR, the International Society for Music Information Retrieval. Website: http://www.music-ir.org/mirex/wiki/MIREX_HOME.

2. RELATED WORK

Musical structure analysis, as an outgrowth of music theory, has existed for centuries. In the Western classical tradition, musical form is as much a dimension of invention and refinement as any other, from perhaps the Renaissance to the present day. For as long as composers and performers were creating and manipulating musical forms, scholars have been analyzing their structure—albeit mostly by hand (and ear). (See [22] for expert commentary relating to musical form as it evolved from the Medieval through the 20th Century periods of Western classical music.)

More recently, attempts to automatically segment musical structure have begun to show promise. For an exhaustive treatment of the task and its history, we found Nieto's Ph.D. thesis [18] to be invaluable. Additionally, Smith and Chew [28] performed a useful meta-analysis of the task up to 2013.

2.1 Nonnegative Matrix Factorization

While a comprehensive summary of NMF techniques is beyond the scope of this paper, we provide some intuition so as to compare our approach with the competition. Some of the mathematical formalism which is often applied to music structure segmentation can be found in [8].

Nieto and Jehan [19] offer an example application of convex NMF in music structure segmentation, though its use for this task dates back to 2010 [13]. Essentially, any piece of music may be transformed into a feature matrix (using features like FFT, MFCC, pitch chroma, *etc.*) This feature matrix may then be factored into lower dimensional matrices whose outer product reconstructs the original feature matrix, more or less. One of the factored matrices may be viewed as a collection of basis features which may be combined to reassemble the song. The other factored matrix represents the activations of our basis features in time, throughout the song. This is illustrated graphically in Fig. 1, from [20].

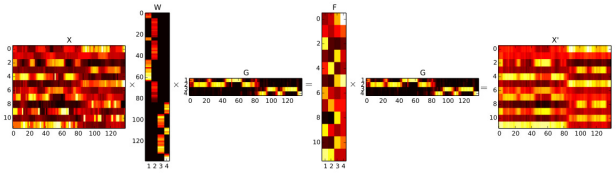


Figure 1: An illustration of convex NMF applied to music structure segmentation from [20, Fig. 1].

From this lower-dimensional representation of song features and their activations, it becomes easier to draw conclusions regarding song structure. Additionally, and fortuitously, the boundary identification and segment association are both straightforward after factorization, since segments with the same basis features can reasonably be assumed to come from the same segment type (*e.g.* first verse and second verse of a pop song.)

2.2 Convolutional Neural Networks

While artificial neural networks have existed since at least the 1960s, and notionally since 1943 [7], the pace of innovation and performance improvements has increased dramatically in the past decade. This is perhaps most evident in the field of image- and vision-related tasks. In 2012, a deep convolutional neural network system won the ImageNet Challenge [14]; every winning ImageNet system since then has also been based on CNNs. (See [23] for detailed information on the ImageNet challenges, as well as a chronicle of the turning point in 2012).

However, usage of CNNs in music and audio has been fairly limited. An early example, [15], uses a CNN to extract musical patterns from audio, as a path to genre classification. Li *et al.* used the dataset GTZAN, common for music genre classification, and extracted the first 13 MFCC features upon which to build their 4-layer CNN system. By today's standards, it is a fairly small CNN: three convolution layers with 3, 15, and 65 convolution kernels, respectively, followed by a fourth fully-connected layer. While yielding interesting results, this is not exactly music structure segmentation.

Karen Ullrich, Jan Schlüter and Thomas Grill, however, have published several papers in recent years regarding music and CNNs, including music structure segmentation. We model a great deal of our work on their 2014 paper [29], upon which their well-performing MIREX submission [25] was based. Recently [11, 12], Grill *et al.* have achieved improved results by combining spectrograms and sliding self-similarity matrices, and using those concatenated features as the input to their CNN systems [24].

3. METHODS

Our initial approach is inspired by [25, 29] and related work. This focuses on the task of boundary retrieval. The subsequent task of associating segments within a song, *e.g.* identifying each verse in a song with the same label, is discussed in §3.2, but left to future work.

We should note that this is a major drawback of the CNN approach to this task. In nonnegative matrix factorization, for example, the task of boundary identification and segment similarity/labelling are accomplished simultaneously. However, the CNN approach should be much faster at test time, since NMF approaches require factoring a huge matrix for each tested song, and much better at segment boundary retrieval (as evidenced in the MIREX 2015 results [4]).

Our method, at a high level, takes a set of audio features related to a particular moment in a song, and outputs a single number which we regard as a segment boundary score. Higher values indicate a higher likelihood of a boundary occurring at that moment. As we will discuss in §4, during training, each of these moments have a corresponding

[4] See `http://www.music-ir.org/mirex/results/2015/mirex_2015_poster.pdf`, under Structure Segmentation. GS1 and GS3 [24] both return state-of-the-art results in the second column, which corresponds to hit rate, or correct identification of boundaries, within 3 seconds of their human-annotated occurrence.

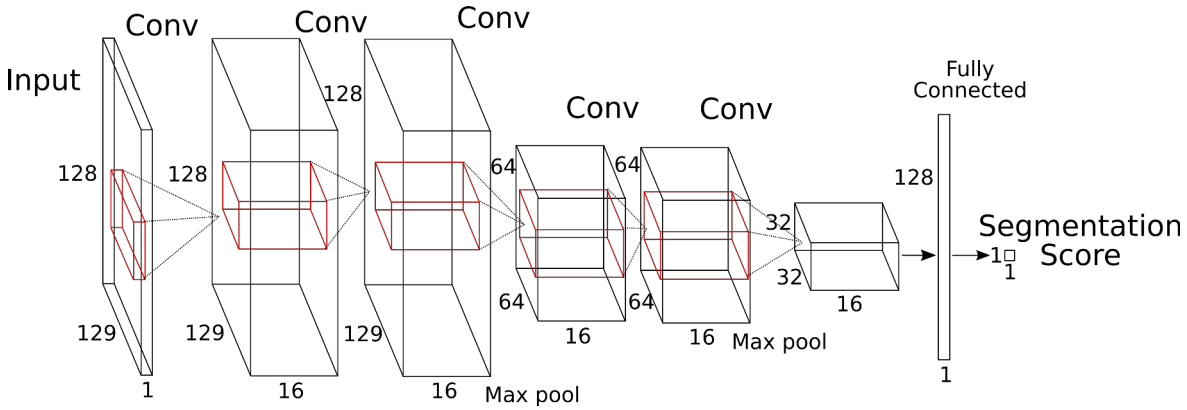


Figure 2: An illustration of our convolutional neural network.

ground-truth score between 0 and 1, where 1 corresponds to a human-annotated segment boundary. Thus, at each time step in a song, the CNN performs a regression on the segment boundary score.

One might ask: why not pose this as a classification task? After all, we are interested in the segment boundary times as well as their associations (*e.g.* verse, chorus). However, this strikes us as an ill-advised approach, since we aim to produce a system which works regardless of genre or type of music. Even within a genre, the musical variability and plasticity of song parts makes us skeptical that classification of song part would yield generalizable models.

3.1 Network architecture

We implemented a small-scale convolutional neural network, shown in Fig. 2, inspired by VGGNet [26] as well as [29]. We do not claim that such a small architecture is optimal or even sufficient; indeed, as we will discuss with our results, we likely require a network which is larger in either or both the number of convolution kernels, or the size of the dimensions, to allow adequate capacity to generalize the notion of a segment boundary. However, we regard this as a good start, in the sense that the small model is less time- and computation-intensive during training, and yields evidence as to whether we are on the right track.

This a sequential CNN, which is similar to vanilla feed-forward neural networks with the exception that lower-dimensional kernels are convolved over the input volume, with the dot product of the convolution kernel and the particular region of input being one output into the next hidden layer. The convolution kernels (weights and biases) are learned via gradient descent.

At each layer, activations are fed through a ReLU (rectified linear unit) nonlinearity. Batch normalization is also applied at each layer. To aid in regularization, 50% dropout is applied at the penultimate fully-connected (*i.e.* non-convolution) layer. We use a mean squared error loss function (with L2 regularization on weights) on minibatches of training input and annotated ground truth scores. Gradients are back-propagated through every level of the CNN, which contains all differentiable units. Our particular means of optimization is stochastic gradient descent with Nesterov momentum.

We implemented this network in Python with Keras [5] using Theano [2, 4] as a backend. We utilized Theano's GPU capabilities, interfacing with NVIDIA's cuDNN 4 library [6] on an NVIDIA GeForce 980M GPU.

3.2 Post-processing network output

As the output of the CNN described above is a scalar score for each time step, we generate a prediction signal for each song, made up of predicted segment boundary scores at each time step in the song. However, this requires two levels of post-processing to arrive at our desired output.

First, we must implement a peak-picking algorithm on the song's prediction signal, as in [29], to arrive at discrete times of predicted segment boundaries. Second, now that we've defined our segment predictions, we need to cluster our segments based on some audio features in order to predict labels. Segment labels need not be as explicit as "verse" and "chorus;" simple alphabetical labels such as A, B, *etc.* are acceptable. The important aspect is to correctly associate the first occurrence of a section with any subsequent occurrences. This may be done by computing average spectral features for the segments, for example, and assigning the same labels to those segments which are closer than a given similarity distance threshold.

Once we have the discrete segment predictions, and/or their predicted labels, we may apply several evaluation metrics, as in the MIREX task. These evaluation metrics are conveniently implemented and available as the Python package `mir_eval` [21]. We should note, however, that these post-processing procedures are currently beyond the scope of our inital efforts, and thus won't be evaluated here.

4. DATASET AND FEATURES

Our datasets fall into two categories. First, we require human annotations of music structure segmentation. Second, we require audio of songs with those human annotations.

[5] `http://keras.io/`

Furthermore, we need to compute audio features and assemble them into a form suitable for input into the CNN described above.

4.1 Dataset

The ground truth on which we train our system must consist of human annotations, since structure and segmentation are perceptual distinctions. To that end, we chose to use the SALAMI (Structural Analysis of Large Amounts of Music Information) dataset [27], which is the largest single set of human song structure annotations of which we are aware, and is commonly used in the music structure segmentation literature. SALAMI contains human-annotated song structure segmentations for some 1,164 songs taken from several sources and genres. An example of functional segmentation for a given track in the SALAMI dataset is reproduced in Fig. 3. 395 are from the Internet Archive, from which the freely available audio tracks were downloaded. Additionally, 74 of the publicly available SALAMI annotations are sourced from the RWC Music Databases [10]. These are high-quality studio recordings of various genres, meant for music research, to which we gained access through Stanford University libraries. Although these works are under copyright, we are allowed to use them for research as affiliates of Stanford University.

Time (s)	**Segment**
0.0	silence
43.56063492	Intro
66.992426303	Verse
89.808163265	Bridge
107.144058956	Chorus
118.560272108	Verse
141.28047619	Bridge
153.137029478	Chorus
176.392086167	Instrumental
187.880385487	Verse
210.856281179	Bridge
228.176712018	Chorus
228.176712018	Outro
277.912562358	no function
303.83154195	End
303.83154195	Silence

Figure 3: Example SALAMI function annotations for the song with SALAMI ID 1003.

We note that, for the vast majority of SALAMI constituents, there are two human annotations. This adds a minimal level of variance to the ground truth, reflecting differences in human perception.

Thus in total, we have audio and human annotations for 469 songs. Features are extracted from audio in Python, with some help from the popular Librosa package [16].

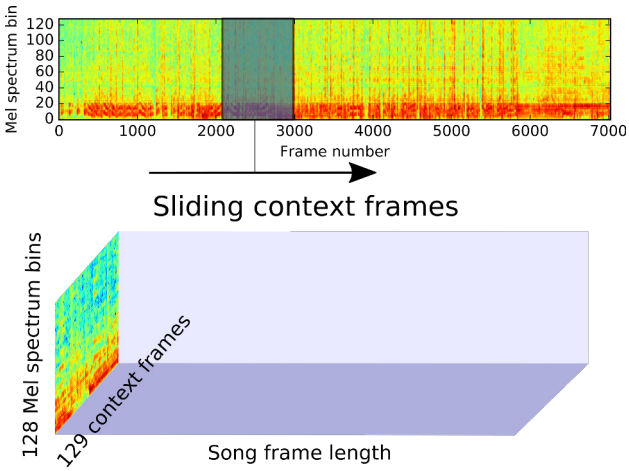


Figure 4: An illustration of audio feature preprocessing for input to our CNN, for an example song.

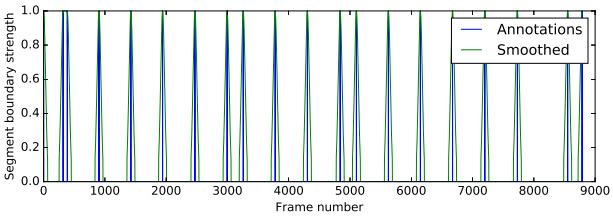


Figure 5: Segment boundary ground truth labelling, per frame. Note that the blue spikes represent the binary labels derived from SALAMI annotations, whereas the green signal shows our smoothed ground-truth achieved by convolving a Gaussian kernel over the blue signal.

4.2 Audio Features

We implemented functionality to retrieve a song, given its SALAMI ID number and availability in our SALAMI audio subset, and compute features such as spectra, Mel-scale spectra, MFCC, and others. For our initial efforts, we decided to use Mel-scale spectrograms. Mel spectra may be thought of like FFT spectra, but the frequency bins correspond to the perceptually-warped Mel scale. The Mel scale is an attempt to transform the linear frequency scale into a mostly logarithmic one which better reflects the way humans perceive pitch. Thus, equally spaced pitches on the Mel scale should correspond to an equal pitch difference in semitones, regardless of the register (low or high).

The frame length and hop size are chosen to be typical values (2048 samples, or 46 ms, per frame, and 50% hop, or 23 ms, between frames), but may also be treated as hyperparameters. We also constrain our Mel spectra to 128 mel-frequency bins, representing a range of 0 Hz to 16 kHz. Finally, each Mel-spectrogram is expressed in dB and normalized on a per-song basis. The top plot of Fig. 4 shows an example Mel-spectrogram.

4.3 Feature Pipeline

After the audio feature computation step of §4.2, we have each song in a large two-dimensional feature matrix. The

horizontal axis is frame number, and the vertical is feature index (Mel spectrum bin, in our case). As in [25], we break each 2D feature matrix into a volume of meta-frames corresponding to each time step. We do this by sliding an "image" of some number of frames (*i.e.* temporal context), and associate each "image" with a single ground-truth value indicating whether a segment boundary occurs at the middle of this context. We decided to make this meta-frame 129 frames wide, *i.e.* 3 seconds long. This is a hyperparameter, and intuitively seems suitable: if we were played 3 seconds of audio and asked whether a segment boundary had occurred at the middle, it strikes us as reasonable. If we listened to just a tenth of a second, on the other hand, we would not expect to predict the correct answer. Thus, for a song with 10,000 frames (slightly less than 4 minutes) we have 10,000 individual 128×129 training examples.

As mentioned above, we may transform our ground truth segment boundaries from discrete times, as in Fig. 3, to signals which represent the presence of a segment boundary at every computed feature frame. We do this by assigning float values between 0 and 1, where 1 indicates the presence of a boundary within that particular frame, and 0 indicates its absence. To account for the sparse occurrence of segment boundaries in a song, as well as the perceptual variance in ground truth, we convolve these labels with a Gaussian kernel (see Fig 5). This differs from [25], who assigned a binary value of 1 within a certain time around the ground truth, 0 otherwise, but further assigned each example lower weight or "importance" depending on the temporal distance from the ground truth label.

Four examples from our dataset are shown in Fig. 6. Thus, we expect results broadly similar to that reported by [25], an example of which is reproduced in Fig. 7.

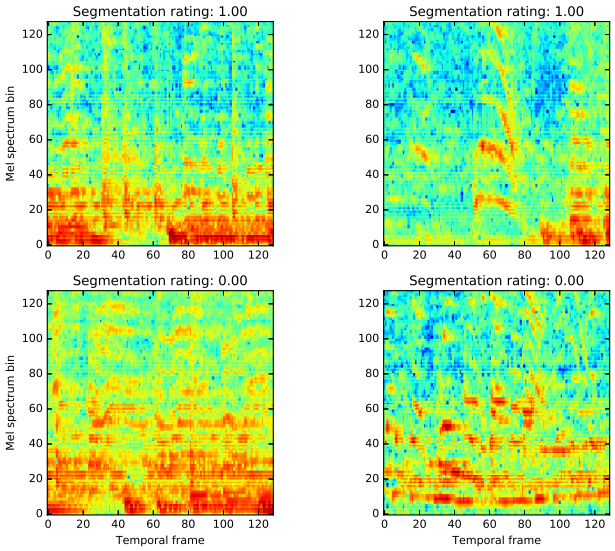


Figure 6: Four examples of Mel-spectrogram context frames. The top two are centered temporally at a human-annotated segment boundary, whereas the bottom two are not. Note that visual inspection of the center of the top two examples shows novel material in relation to the preceding and succeeding context, whereas the bottom two show somewhat homogeneous examples.

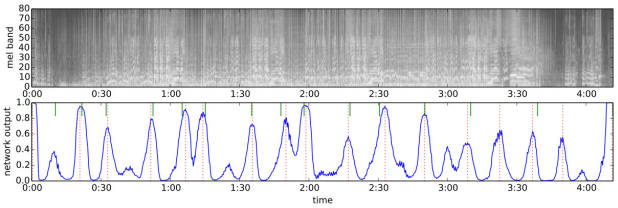


Figure 7: Example results reported by Schlüter *et al.* [25, Fig. 1]. The top graph shows the Mel-spectrogram for a test example, while beneath the corresponding CNN output is shown in blue. In that bottom plot, human-annotated segment boundaries are shown as red dotted lines, whereas the predicted segment boundaries, after peak-picking, are shown as green dashes at the top of the plot.

5. EXPERIMENTAL RESULTS AND DISCUSSION

Results remain somewhat preliminary, as we did not have time to train our model on the full set of 469 songs. We used a training set of 15 songs, a validation set of 1 song, and a test set of 10 songs, all of which were randomly chosen without replacement. As discussed above, depending on the length of a song, it may have tens of thousands of training examples; thus, a full run remains to be performed.

5.1 Test song predictions

Although we fear that the training set was not varied enough to produce a fully generalizable model, we do see evidence in our test predictions that the model is retaining some generalizable hallmarks of music structure boundaries. Several example plots are shown in Figs. 8 and 9. Fig. 8a shows perhaps the best performance in the test set. Upon visual inspection, 4 or 5 of 8 boundaries have corresponding prediction signal peaks which are at least reasonably close to the ground truth and above the background noise in the signal. We expect the signal noise to subside with increased training time and an increased number of training songs. Fig. 8b seems to show at least two boundary identifications, but also perhaps two spurious boundary identifications. Fig. 9b also appears to show a correct boundary identification, as well as a couple spurious boundaries following it. Fig. **??** shows relatively well-behaved predictions, except for the intermittent plunges to large negative predictions.

We may contrast these admittedly anecdotal observations with our previous small-scale training runs on one song. These were mainly for quick system tests. However, it was evident that training on one song does not produce prediction signals with any sort of reliable peaks. That is, as we'd expect, the system did not learn to generalize the notion of segment boundary when it only saw examples from one context (a single song). The fact that, with 15 songs, we start to see halfway decent predictions gives us hope that we may be able to achieve much better results when training with some significant fraction of our corpus of 469 songs.

However, based on the sizes of other, similar systems, we expect to have to enlarge our model. Given our intuitive knowledge of the breadth of sonic phenomena that constitute segment boundaries, we plan to at least double the number of convolution kernels at each layer, as well as the size of the hidden fully-connected layer. Additionally, to capture more complex patterns, we plan to add additional convolution layers; more complex graph structures may also be beneficial.

5.2 Model visualization

One of the most interesting ways to interrogate our model is to visualize the weights. That is, given a particular convolution neuron, we optimize an image (or in this case, the 3-second context Mel-spectrogram) to maximally activate that neuron. [6]

Visualizations of several of our weights from the first convolution layer are shown in Fig. 10. The top row appears to show consecutive vertical lines, which would translate to broadband impulsive sounds such as successive drum hits. Indeed, the regular patterns suggest rhythmic temporal hits. This makes sense; firstly, broadband rhythmic hits seem to be a reasonable low-level feature of music; secondly, and intuitively, transitions between structural sections in music are often marked by pronounced and accentuated rhythmic content.

In the bottom row of Fig. 10, we see two examples of a more complex phenomenon. They suggest perhaps a rising harmonic trajectory, though not in a straightforward manner. Perhaps it is sufficient to characterize them as smooth harmonic trajectories over time. This also makes sense, as musical structure boundaries are often characterized by broad and continuous sweeps over harmonic or melodic terrain, thereby connecting disparate structural elements.

Finally, we may remark that these low-level patterns seem analogous to the low-level patterns such as edges which we expect to find in visual recognition systems. This makes us optimistic, since our model appears to be learning relevant patterns.

6. CONCLUSIONS AND FUTURE WORK

We have chronicled our efforts in implementing a convolutional neural network to automatically segment music by song structure. After reviewing the task and relevant background, we introduced our system, and showed preliminary evidence that it has returned promising results. In the immediate future, we plan to train on a set of songs which are an order of magnitude larger than our current experiment. Simultaneously, we plan to enlarge our network architecture to allow enough capacity to model this large set.

[6] Our code for this section was adapted from the following post to the Keras blog by François Chollet: `http://blog.keras.io/how-convolutional-neural-networks-see-the-world.html`

6.1 Rebalancing the training examples

We should note that most context-frame examples will not be boundaries, leading to an unbalanced set of training examples. We should perhaps boost the number of positive examples (*i.e.* those context frames centered at segment boundaries) shown during training. Indeed, [25] report boosting the probability of a positive training example by a factor of 3. We will accomplish this, quite easily, by randomly inserting some number of positive examples to the training set. Indeed, we may center each context frame exactly at the annotated segment boundary, leading to additional context frames that are not only centered at the boundary frame, but whose boundary frames are exactly centered at the segment time. Whether this is at all beneficial remains to be seen, but it does allow us to add context frames that are not exact duplicates to the training set.

6.2 MIREX-style evaluation

The ultimate system evaluation should follow the MIREX task evaluation procedures, [7] as implemented in [21], and discussed above in §3.2.

However, we should acknowledge Nieto's [18] remarks about the pitfalls of any individual metric. These evaluation metrics are necessarily imperfect because they seek to objectively measure subjective perception. Thus, better performance in the metrics is certainly a goal, but not the only one. We should carefully interpret the details of our ultimate model, as a better-performing system might be quite valuable in any insight its individual weights and elements can provide.

6.3 Transfer learning with pre-trained models

Finally, in the field of image processing, we note the prevalence of systems which leverage transfer learning on pre-trained CNN models. For example, the Caffe Model Zoo [8] features many state-of-the-art models which any investigator can freely use for a subsequent system. Although systems such as the one described in this paper are already in use at companies such as Google and Spotify, though their models are currently proprietary. Sources tell us that this may soon change, in which case a transfer learning project would be extremely interesting and compelling.

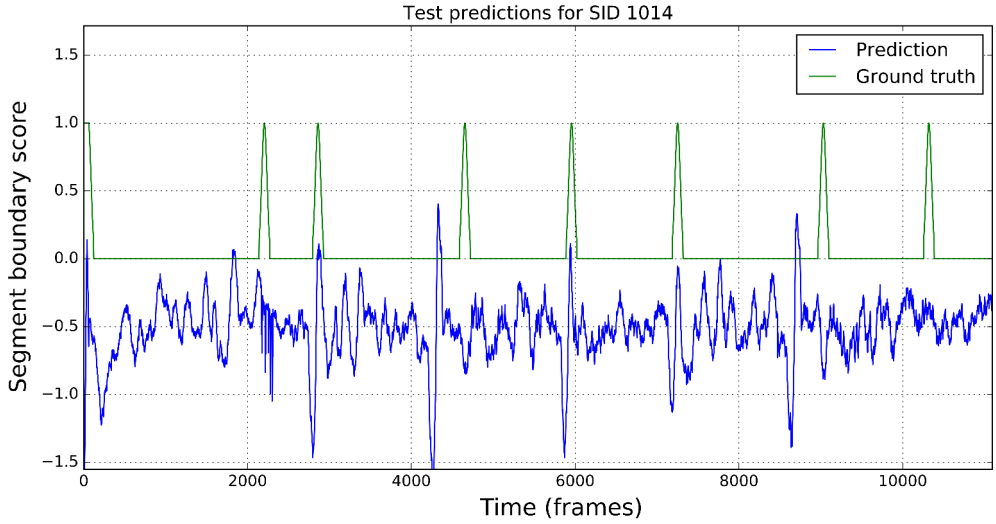
7. REFERENCES

[1] Mark A. Bartsch and Gregory H. Wakefield. Audio Thumbnailing of Popular Music Using Chroma-Based Representations. *IEEE Transactions on Multimedia*, 7(1):96–104, feb 2005.

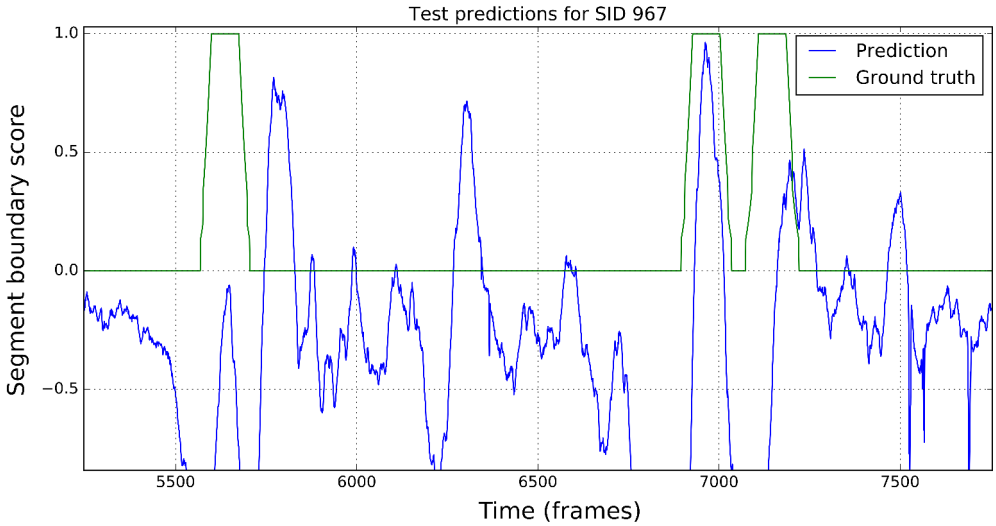
[2] F Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian Goodfellow, Arnaud Bergeron, Nicolas Bouchard, David Warde-Farley, and Yoshua Bengio. Theano: new features and speed improvements. In

[7] See `http://www.music-ir.org/mirex/wiki/2015:Structural_Segmentation\#Evaluation_Procedures`

[8] `https://github.com/BVLC/caffe/wiki/Model-Zoo`



(a) Predictions for the song with SALAMI ID number 1014..



(b) Predictions for about 1 minute of the song with SALAMI ID number 967.

Figure 8: Two examples of segment boundary score prediction. The blue signals are our CNN prediction, and the green signals show the smoothed ground truth.

Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, pages 1–10, 2012.

[3] Bernard Bel and Bernard Vecchione. Computational musicology. *Computers and the Humanities*, 27(1):1–5, 1993.

[4] James Bergstra, Olivier Breuleux, Frederic Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math compiler in Python. *Proceedings of the Python for Scientific Computing Conference (SciPy)*, (Scipy):1–7, 2010.

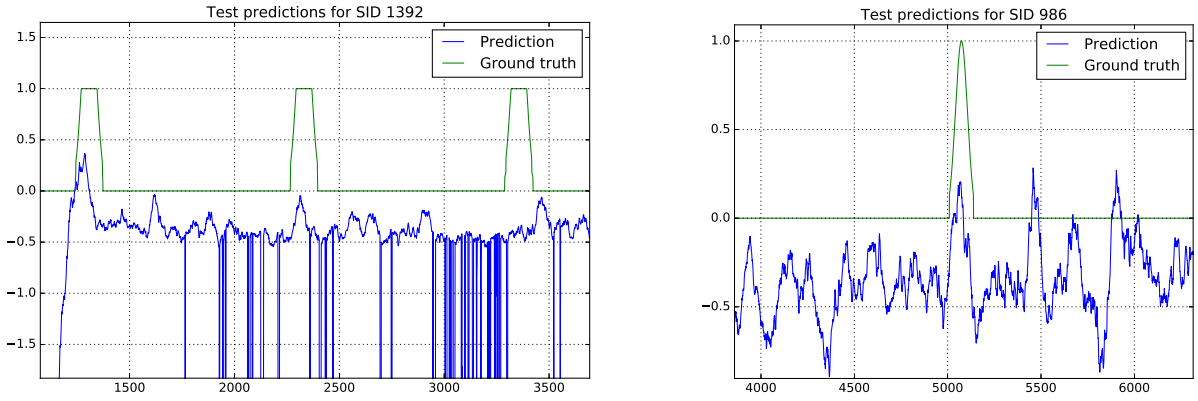
[5] Lelio Camilleri. Computational Musicology: A Survey on Methodologies and Applications. *Revue Informatique et Statistique dans les Sciences humaines*, XXIX(4):51–65, 1993.

[6] Sharan Chetlur, Cliff Woolley, Philippe Vandermersch, Jonathan Cohen, John Tran, Bryan Catanzaro, and Evan Shelhamer. cuDNN: Efficient Primitives for Deep Learning. *CoRR*, abs/1410.0, 2014.

[7] Jack D Cowan. Neural Networks: The Early Days. *Advances in Neural Information Processing Systems*, pages 828–842, 1990.

[8] Chris Ding, Tao Li, and Michael I Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):45–55, 2010.

[9] Daniel P. W. Ellis and Graham E. Poliner. Identifying 'cover songs' with chroma features and dynamic programming beat tracking. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4:IV–1429 – IV–1432, 2007.



(a) (b)

Figure 9: Two examples of segment boundary score prediction. The blue signals are our CNN prediction, and the green signals show the smoothed ground truth.

[10] Masataka Goto. Development of the RWC music database. In *Proceedings of the 18th International Congress on Acoustics (ICA 2004)*, pages 553–556, 2004.

[11] Thomas Grill and Jan Schlüter. Music boundary detection using neural networks on combined features and two-level annotations. In *16th International Society for Music Information Retrieval Conference (ISMIR 2015)*, 2015.

[12] Thomas Grill and Jan Schlüter. Music boundary detection using neural networks on spectrograms and self-similarity lag matrices. In *Signal Processing Conference (EUSIPCO), 2015 23rd European*, pages 1296–1300. IEEE, 2015.

[13] Florian Kaiser and Thomas Sikora. Music Structure Discovery in Popular Music using Non-negative Matrix Factorization. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, pages 429–434, 2010.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In F Pereira, C J C Burges, L Bottou, and K Q Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[15] Tom L H Li, Antoni B Chan, and Andy H W Chun. Automatic Musical Pattern Feature Extraction Using Convolutional Neural Network. In *Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS 2010)*, volume I, pages 546–550, 2010.

[16] Brian McFee, Matt McVicar, Colin Raffel, Dawen Liang, Oriol Nieto, Eric Battenberg, Josh Moore, Dan Ellis, Ryuichi Yamamoto, Rachel Bittner, Douglas Repetto, Petr Viktorin, João Felipe Santos, and Adrian Holovaty. librosa: 0.4.1, oct 2015.

[17] Unjung Nam. *A Method of Automatic Recognition of Structural Boundaries in Recorded Musical Signals*. PhD thesis, Stanford University, 2004.

[18] Oriol Nieto. *Discovering Structure in Music: Automatic Approaches and Perceptual Evaluations*. PhD thesis, New York University, 2015.

[19] Oriol Nieto and Tristan Jehan. Convex Non-Negative Matrix Factorization for Automatic Music Structure Identification. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 236–240, 2013.

[20] Oriol Nieto and Tristan Jehan. MIREX 2014 Entry: Convex Non-Negative Matrix Factorization. *Music Information Retrieval Evaluation eXchange*, 2014.

[21] Colin Raffel, Brian McFee, Eric J. Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, and Daniel P. W. Ellis. mir_eval: A Transparent Implementation of Common MIR Metrics. In *Proc. of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, pages 367–372, 2014.

[22] Leonard G Ratner. *Music, the listener's art*. McGraw-Hill, 1966.

[23] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, and Others. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[24] Jan Schlüter and Thomas Grill. Structural segmentation with convolutional neural networks MIREX submission. *Music Information Retrieval Evaluation eXchange*, 2015.

[25] Jan Schlüter, Karen Ullrich, and Thomas Grill. Structural Segmentation with Convolutional Neural Net-

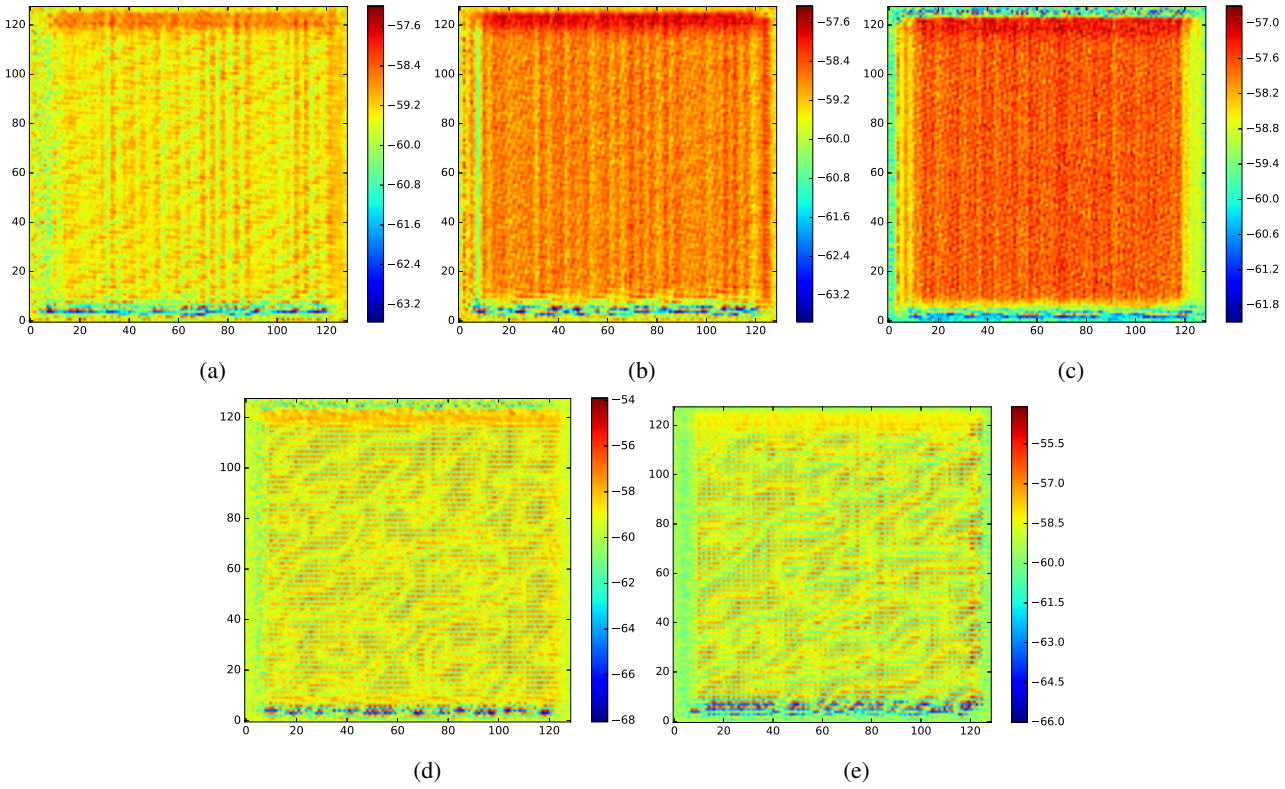


Figure 10: Five example weight visualizations for our first convolution layer.

works MIREX Submission. *Music Information Retrieval Evaluation eXchange*, pages 3–4, 2014.

[26] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *Iclr*, pages 1–14, 2015.

[27] Jordan B L Smith, Ja Burgoyne, and Ichiro Fujinaga. Design and creation of a large-scale database of structural annotations. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, pages 555–560, 2011.

[28] Jordan B L Smith and Elaine Chew. A meta-analysis of the MIREX Structure Segmentation task. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 251–256, 2013.

[29] Karen Ullrich, Jan Schlüter, and Thomas Grill. Boundary Detection in Music Structure Analysis Using Convolutional Neural Networks. *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, 2014.