A Supervised Classification Algorithm For Note Onset Detection

Alexandre Lacoste
University of Montreal
Computer Science Department
Montreal, QC, H3T 1J4 CANADA
lacostea@iro.umontreal.ca

Douglas Eck
University of Montreal
Computer Science Department
Montreal, QC, H3T 1J4 CANADA
eckdoug@iro.umontreal.ca

Abstract

This paper presents a novel approach to detecting onsets in music audio files. We use a supervised learning algorithm to classify spectrogram frames extracted from digital audio as being onsets or non-onsets. Frames classified as onsets are then treated with a simple peak-picking algorithm based on a moving average. In this paper we present two versions of this approach. The first version uses a single neural network classifier. The second version combines the predictions of several networks trained using different hyperparameters. In the paper we describe the details of the algorithm and summarize the performance of both variants on the MIREX 2005 note onset detection contest, where the two variants were awarded first and second place. We also examine our choice of hyperparameters by describing results of cross validation experiments done on a custom dataset. We conclude that a supervised learning approach to note onset detection warrants further investigation.

1 Introduction

This paper concerns finding the onset times of notes in music audio. Though conceptually simple, this task is deceivingly difficult to perform automatically with a computer. Consider for example the naïve approach of finding amplitude peaks in the raw waveform. This strategy would fail almost completely except in the case of monophonic percussive instruments played slowly. At the same time, onset detection is implicated in a number of important music information retrieval (MIR) tasks.

Areas involving the analysis of temporal structure in music such as *tempo identification* and *meter identification* benefit from onset detection. For example, an inter-onset histogram built from onsets could be used as a density estimate of likely tempos. *Music classification*

and *music fingerprinting* are two other relevant areas. For both, we need to extract low-dimensional features from music. In the case of classification, onset locations could be used to drastically reduce the number of frame-level features retained. For example, only frames near an onset could be analyzed. Alternately, they could be used to merge frame-level features in an intelligent way. West and Cox [30] for example uses a related segmentation strategy for genre classification. In fingerprinting, the goal is to generate a vector that uniquely identifies a song. In this case a list of onset times could be used directly as a robust fingerprint.

Finally the areas of *music editing* and *automatic music transcription* can benefit from onset detection. For music editing, onsets could be used to break a waveform into logical parts. This could allow song editors and sequencers to present recorded audio as something more easily edited than a raw waveform. Automatic music transcription presents perhaps the most direct use of note onset detection. Finding note onsets remains a fundamental challenge in transforming digital audio into a more structured, discrete representation suitable for notation analysis.

Onsets detection algorithms can generally be divided into three steps:

1. Transformation of the waveform to isolate different frequency bands, in general using either a filter bank or a spectrogram.

2. Enhancement of bands such that note onsets are more salient; this could involve, for example, a filter that detects positive slopes.

3. Peak-picking to select discrete note onsets.

Our approach includes enhancements at each of these steps. In the first step, we look at different methods for computing and representing the spectrogram as well as at strategies for merging spectrogram frames. In the second step, where we focus most of our attention, we introduce a supervised learning approach that tries to identify peaks in the output of the first step. Specifically we use neural networks to provide the best possible onset trace for the peak-picking part. In the third step we take advantage of a tempo estimate in order to integrate some aspect of rhythmic structure into the peak-picking decision process.

In this paper, we first review the work done in this field with special attention paid to other works done on onset detection using machine learning. In Section 3, we describe our algorithm including details about the simpler and more complex variants. In Section 4, we describe a dataset that we built for testing the model and we describe a Matlab-based tool we built to aid in labeling onsets. Finally in Section 5, we present experiment results that report on our investigation of different spectrogram representations and on different network architectures.

2 Previous work

Earlier algorithms developed for onset detection focused mainly on the variation of the signal energy envelope in the time domain. Scheirer [27] demonstrated that much information from

the signal can be discarded while still retaining the rhythmical aspect. On a set of test songs Scheirer filtered out different frequency bands using a filter bank. He extracted the energy envelope for each of those bands, using rectification and smoothing. Finally, with the same filter bank, he modulated a noisy signal with each of those envelopes and merged everything by summation (Figure 1). With this approach, rhythmical information was retained. On the other hand, care must be taken when discarding information. In another experiment he shows that if the envelopes are summed before modulating the noise, a significant amount of information about rhythmical structure is lost.

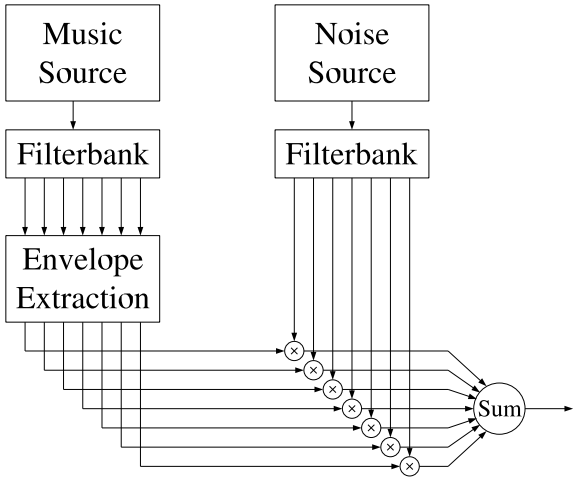

Figure 1: Modulating noise with the energy envelope of different bands from a filter bank retains the rhythmical content of the song.

Klapuri [19] used the psychoacoustical model developed by Scheirer to develop a robust onset detector. To get better frequency resolution, he employed a filter bank of 21 filters. The author points out that the smallest detectable change in intensity is proportional to the intensity of the signal. Thus $\frac{\Delta I}{I}$ is a constant where I is the signal's intensity. Therefore, instead of using $\frac{d}{dt}A$ where A is the amplitude of the envelope, he used

$$\frac{1}{A}\left(\frac{d}{dt}A\right) = \frac{d}{dt}\log\left(A\right). \tag{1}$$

This provides more stable onset peaks and allows lower intensity onsets to be detected. Later, Klapuri used the same kind of preprocessing [20] and won the ISMIR 2004 tempo induction contest [17].

2.1 Onset detection in phase domain

In contrast to Scheirer's and Klapuri's work, J. P. Bello and C. Duxbury [11, 10, 2, 1] took advantage of phase information to track the onset of a note. They found that at steady

state, oscillators tend to have predictable phase. This is not the case at onset time, allowing the decrease in predictability to be used as an indication of note onset. To measure this, they collected statistics on the phase acceleration, as estimated by the following equation:

$$\alpha_{k,n} = \text{princarg}\left[\varphi_{k,n} - 2\varphi_{k,(n-1)} + \varphi_{k,(n-2)}\right] \tag{2}$$

where $\varphi_{k,n}$ is the kth frequency bin of the nth time frame from the short time Fourier transform of the song. The operator princarg maps the angle to the $[-\pi, \pi]$ range. To detect onset, different statistics were calculated across the range of frequencies including mean, variance and kurtosis. These provide an onset trace, which can be analyzed by standard peak-picking algorithms. The authors also have combined phase and energy on the complex domain for more robust detection. Results on monophonic and polyphonic music shows an increase in performance for phase against energy, and even better performance when combining both.

2.2 Onset detection using supervised learning

Only a small amount of work has been done on mixing machine learning and onset detection. In recent work, Kapanci and Pfeffer [18] used support vector machine (SVM) on a set of frame features to estimate if there is an onset between two selected frames. Using this function in a hierarchical structure, they can find the position of onsets. Their approach mainly focuses on finding onsets in signals with slowly-varying change over time such as solo singing.

Marolt et al. [23] used a neural network approach for note onset detection. This approach is similar to ours in its use of neural networks, but is otherwise very different. The model used the same kind of preprocessing as in Scheirer [27], with a filter bank of 22 filters. An integrate-and-fire network was then applied separately to the 22 envelopes. Finally a multi layer perceptron was applied on the output to accept or reject the onsets. Results were good but the model was only applied to monophonic piano music.

Davy and Godsill [8] developed an audio segmentation algorithm using a support vector machine. They classify spectrogram frames into being probable onsets or not. The SVM was used to find a hypersurface delimiting the probable zone from the less probable one. Unfortunately, no clear test is made to outline the performance of the model.

3 Algorithm Description

3.1 Feature Extraction

In this section we introduce two variants of our algorithm. Both use a neural network to classify frames as being onsets or non-onsets. The first variant, SINGLE-NET, follows the process for onset detection described above and shown in Figure 2. Our second variant MULTI-NET combines information from (A) *multiple* instantiations of SINGLE-NET, each trained with different hyperparameters and (B) tempo traces gained by running a tempo

detection algorithm on the neural network output vector. The multiple sources of evidence are merged into a feature matrix similar to a spectrogram which is in turn fed back into another feed-forward network, peak picker and onset detector. See Figure 3.

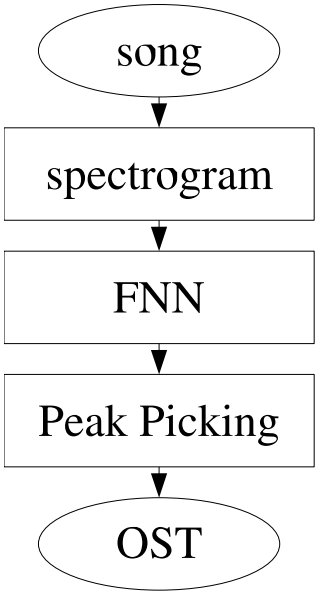

Figure 2: SINGLE-NET flow chart. This simpler variant of our algorithm is comprised of a time-space transform (spectrogram) which is in turn treated with a neural network. The resulting trace is fed into a peak-picking algorithm to find onset times.

3.1.1 Time-frequency domain transform

Aside from the prediction of global tempo done in the MULTI-NET variant of our algorithm, the information provided to the classification step of the algorithm is local in time. This raises the question of how much local information to integrate in order to achieve best results. Using a parameter search we concluded that at least 50ms of sound was necessary to generate good results. For a sampling rate of 22050 Hz, this indicates that the dimensionality of the input for a supervised learning algorithm would be $\sim$1000.

As is commonly done, we decided to us a time-space transform to lower the dimensionality of the representation and to reveal spectral information in the signal. We focused on the Short-Time Fourier Transform (STFT) and the Constant-Q transform [4]. These are discussed separately in the following two sections.

3.1.2 Short Time Fourier Transform (STFT)

The Short Time Fourier transform is a version of the Fourier transform designed for computing short-time duration frames. A moving window is swept through the signal and the Fourier transform is repeatedly applied to portions of the signal inside the window.

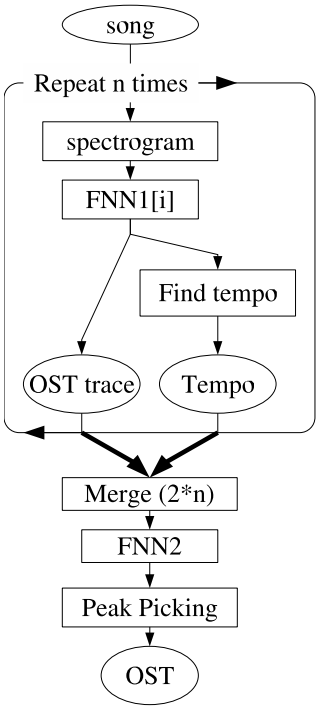

Figure 3: MULTI-NET flow chart. The SINGLE-NET variant is repeated multiple times with different hyperparameters. A tempo detection algorithm is run on each of the resulting FNN outputs. The SINGLE-NET outputs and the tempo detection outputs are then combined using a second neural network.

$$STFT\left(t,\omega\right)=\int_{-\infty}^{\infty}x\left(\tau\right)w^{*}\left(\tau-t\right)e^{-j\omega\tau}d\tau \tag{3}$$

Where $w\left(t\right)$ is the windowing function that isolates the signal for a particular time t. Sequence $x\left(t\right)$ is the signal we want to transform, in this case, an audio signal in PCM format.

The discrete version of the STFT is

$$STFT\left[n,k\right]=\sum_{m=-\infty}^{\infty}x\left[n+m\right]w\left[m\right]e^{-jkm} \tag{4}$$

A Hamming window is applied to the signal. By choosing a bigger window width, we get a better frequency resolution but a smaller time resolution. Reducing the window width produces the inverse effect.

3.1.3 Constant-Q transform

The Constant-Q transform [4] is similar to the STFT but it has two main differences.

- It has a logarithmic frequency scale.

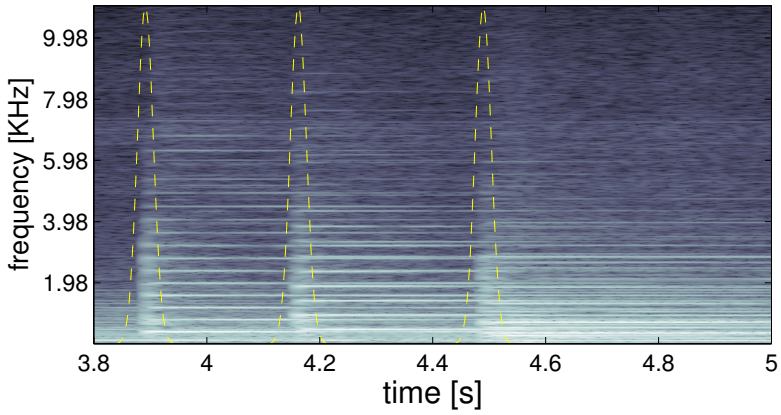

Figure 4: This figure represents the magnitude plane of the STFT of a guitar song. The sampling frequency is 22050 Hz, the window width is 30 ms and the overlapping factor is 0.9. The yellow dashed line represents the labeled onsets positions.

- It has a variable window width.

The logarithmic frequency scale provide a constant frequency-to-resolution ratio for a particular bin.

$$Q = \frac{f_k}{f_{k+1} - f_k} = \left(2^{\frac{1}{b}} - 1\right)^{-1} \tag{5}$$

where b represents the number of bins per octave and k, the frequency bin. For $b = 12$, and by choosing a particular f_0 then k is equal to the midi note number. See Figure 5 for an example of a Constant-Q transform.

As the frequency resolution is smaller at high frequencies, we can shrink the window width to yield better time resolution, which is very important for onset detection.

Like the Fast Fourier Transform (FFT), there is an efficient algorithm for Constant-Q transform. See [5] for implementation details.

3.1.4 Phase planes

Both STFT and Constant-Q are complex transforms. Therefore, we can separate their outputs into phase and magnitude planes. Obviously, the magnitude planes contain relevant information; see Figure 4 and Figure 5. But can we do something with the phase plane? A visual observation (Figure 6) reveals that the phase plane of an STFT is quite noisy.

One potentially useful way to process the phase plane is according to the following equation from [2]:

$$\alpha_{k,n} = \text{princarg}\left[\varphi_{k,n} - 2\varphi_{k,(n-1)} + \varphi_{k,(n-2)}\right] \tag{6}$$

Where $\alpha(k, n)$ is the phase acceleration at frequency bin k and time bin n. The function princarg maps the value of phase acceleration onto the interval $[-\pi, \pi]$. Their experiments

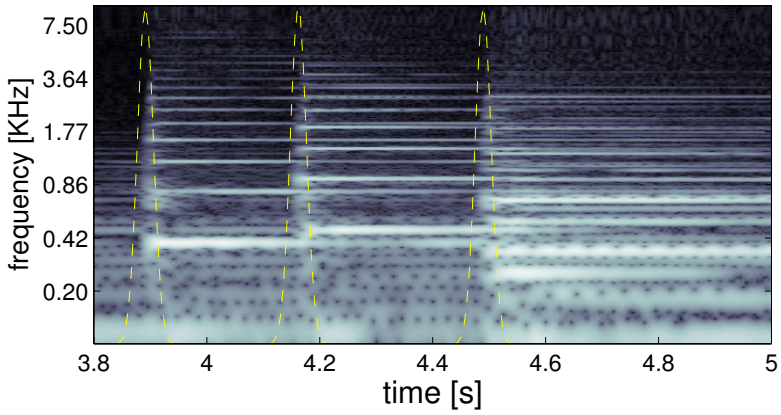

Figure 5: This figure represents the magnitude plane of the Constant-Q transform of the same guitar song as in Figure 4. The sampling frequency is 22050 Hz, the window width is 30 ms and the number of bin per octave is 48. The yellow dashed line represents the labeled onset positions.

shows that at onset time the probability distribution of phase acceleration over all frequency is changing. However, Figure 7 shows that onset patterns are almost absent and table 1 shows that the FNN was not able to see those patterns.

So far we have little evidence that the phase plane information will be useful for our task. However, by taking the phase difference along frequency instead of time (column-wise rather than row-wise in the matrix) yields much more promising results.

$$\varpi_{k,n} = \text{princarg}\left[\varphi_{k,n} - \varphi_{(k-1),n}\right] \tag{7}$$

where $\varpi_{k,n}$ represents the phase difference between frequency bin k and frequency bin $k-1$ for a particular time bin n.

In Figure 8 patterns are visible that correlate highly with onset times. Table 1 show that the phase plane alone, when processed this way, is able to perform almost as well as the magnitude plane.

3.2 Supervised Learning for Onset Emphasis

We employ a feed-forward Neural Network to combine evidence from the different transforms in order to classify the frames. Our goal is to use the neural net as a filtering step in order to provide the best possible trace for the peak-picking part. The network predicts the class membership (onset or no-onset) of each frame in a sequence. The evidence available to the network for each prediction consists of the different spectral features extracted from the PCM signal as described above. For a given frame, the network has access to the features for the frame in question as well as nearby frames. In this section, we use the term "window" to refer to the size of the input window defining which feature frames are fed into the FNN

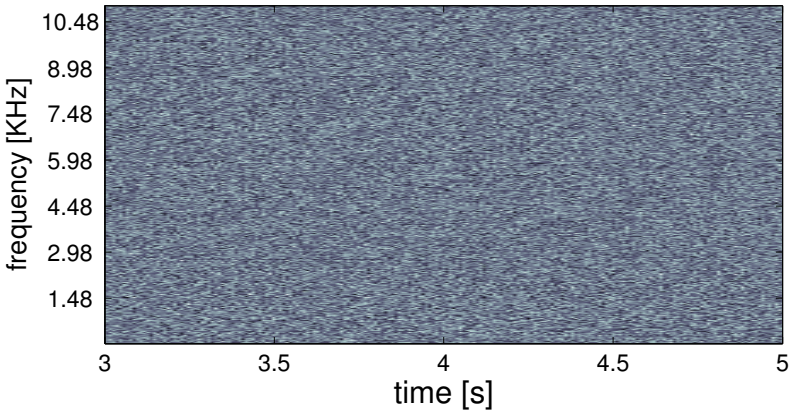

Figure 6: This figure shows the phase plane of the STFT calculated in Figure 4. Unmanipulated, such a phase plane looks very much like a matrix of noise.

(Figure 9). This is in contrast to the spectral window used to calculate the spectrogram in section 3.1.1.

3.2.1 Input variables

Onsets patterns are translation invariant on the time axis. That is, the probability distribution over all the possible patterns presented to the network does not depend on the time value.

$$p\left(X = x | T = t\right) = p\left(X = x\right) \tag{8}$$

$$x \in \mathbb{R}^n \tag{9}$$

where n is the number of inputs variables, x represents a particular input to the network and t is the central time of the window.

Unfortunately, the frequency axis does not exhibit this same shift invariance.

$$p\left(X = x | F = f\right) \neq p\left(X = x\right) \tag{10}$$

where f is the central frequency of the input window. For example, when using the STFT, an onset with a fundamental at a higher frequency will have more widely-spaced harmonics than a low-frequency onset. For the case of Constant-Q transform, the distances between harmonics are indeed shift invariant. However for low frequencies the patterns are highly blurred over frequency and time.

Also, for a small frequency shift, the probability distributions are very similar.

$$|f_1 - f_2| < d \Rightarrow p\left(x | f_1\right) \simeq p\left(x | f_2\right) \tag{11}$$

where d is a hyper-parameter. The solution was to choose a window height that covers 90% of the frequencies range. Therefore, the frequency translation is smaller than 10% of the spectrum.

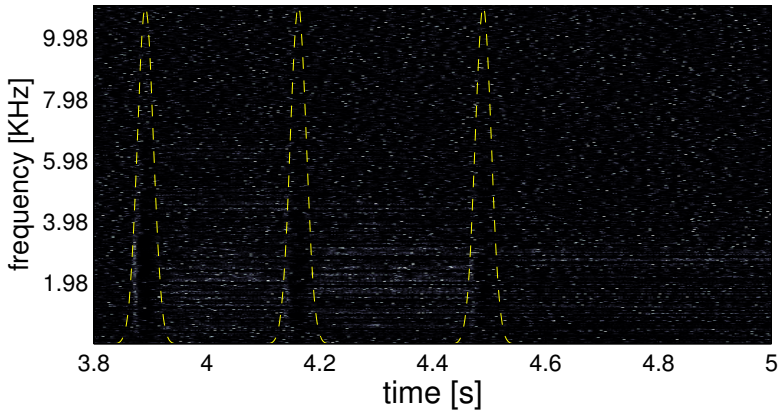

Figure 7: This figure shows the phase plane of the STFT of Figure 4, transformed according to the equation 6. The yellow dashed line represents the labeled onsets positions. In this representation, the onset patterns are hard to see.

Unfortunately, for a decent spectrogram resolution and an input window of ∼50 ms, we still have too many variables in the input window. The solution was to randomly select some variables from the window, specifically ∼100 variables from a continuous distribution over frequency and a Gaussian distribution over time.

3.2.2 Neural network structure

We used a Feed Forward Neural Network (FNN) with two hidden layers and only one neuron in the output layer. The hidden layers used tanh activation functions and the output layer used the logistic sigmoid activation function.

The architecture used for the contest employed 160 inputs, 18 units for first hidden layer and 15 units for second hidden layer.

3.2.3 Target, error function and learning

The goal of the network is to produce the ideal trace for the peak picking part. Such a target trace can be a mixture of very peaked Gaussians, centered on the labeled onset time.

$$T_s(t) = \sum_i \exp^{-(\tau_{s,i}-t)^2/\sigma^2} \tag{12}$$

where $\tau_{s,i}$ is the i^{th} labeled onset time of song s and σ is the width of the peak and is chosen to be 10 ms.

For each time step the FNN predicted the value given by the target trace. The error function is the sum of squared error over all patterns.

$$E = \sum_{s,j} \left(T_s(t_j) - O_s(t_j)\right)^2 \tag{13}$$

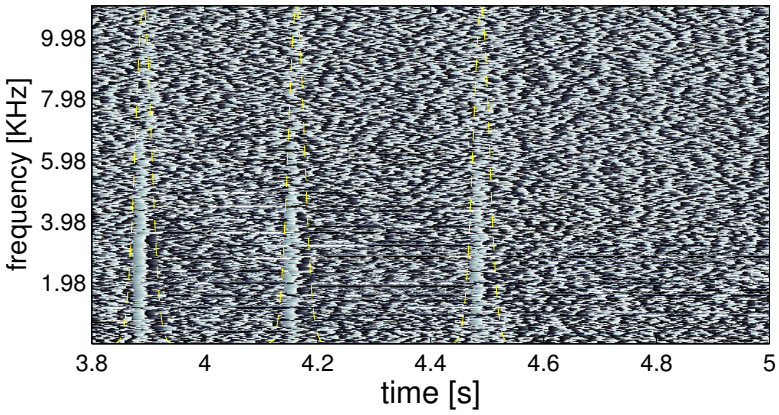

Figure 8: This figure represents the phase plane of the STFT of Figure 4, transformed according to the equation 7. The yellow dashed line represents the labeled onsets positions.

where $O_s(t_j)$ is the output of the network for pattern j of song s.

The learning function is the Polak-Ribiere version of conjugate gradient descent as implemented in the Matlab Neural Network Toolbox. It uses early stopping on a small validation set for regularization. For more details on cross-validation, see section 5. For details on the dataset, see section 4.

3.3 Peak-picking

As explained in section 3.2.1, the input window is translated across the frequency axis using a random selection of points across a moving window. This strategy yields more than one set of values to process for a particular time frame. Those values are simply merged by averaging, generating the onset trace. See Figure 10 for an example.

To isolate the peaks, we first smooth the onset trace using a Gaussian spatial filter of 500 ms of standard deviation. Then we build a peak trace by subtracting the filtered trace plus a small threshold from the original onset trace.

$$\rho_s(t) = O_s(t) - u_s(t) + K \tag{14}$$

where

$$u_s(t) = g * O_s(t) \tag{15}$$

where g is the Gaussian filter, K is the threshold and ρ_s is the peak trace of song s. Using this approach each zero crossing with positive slope represents the beginning of an onset and each zero crossing in a negative slope represents the end of an onset.

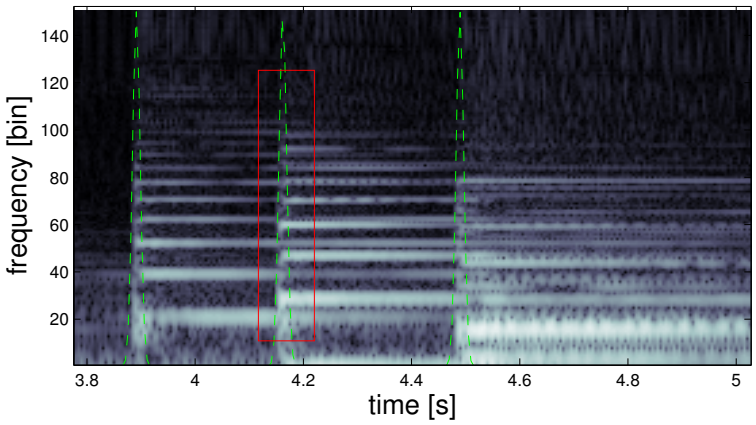

Figure 9: The Constant-Q transform of a piano song with labeled onsets. The green dashed line is the onset trace, it corresponds to the ideal input for the peak-picking algorithm. The red box is a window seen by the neural network for a particular time and particular frequency.

The position of the onset is taken by calculating the center of mass of all points inside the peak.

$$\tau_{s,i} = \frac{\sum\limits_{j \in p_i} t_j \rho_s\left(t_j\right)}{\sum\limits_{j \in p_i} \rho_s\left(t_j\right)} \tag{16}$$

where $\tau_{s,i}$ is the i^{th} onset time of song s and j is element of all the points contained in peak i.

To optimize performance, the value of the threshold K in equation 14 is learned using samples from the training set. In order to make such an optimization, we require a way to gauge overall performance. For this we used the same F-measure used in the MIREX 2005 onset detection contest.

$$\begin{aligned} P &= \frac{n_{ca}}{n_{ca} + n_{HP}} \\ R &= \frac{n_{cd}}{n_{cd} + n_{fn}} \\ F &= \frac{2PR}{P + R} \end{aligned} \tag{17}$$

where n_{cd} is the number of correctly detected onsets, n_{fn} is the number of false negative, n_{fp} is the number of false positives and F is the F-measure. A perfect score gives a F-measure of 1 and for a fixed number of error, the F-measure is optimal when the number of false positive equals the number of false negative.

However, the peak-picking part is not a continuous function, which prevents us from using gradient descent for optimizing K. Fortunately, we have only one parameter to optimize. Such an optimization is usually done using a line search algorithm like the golden section

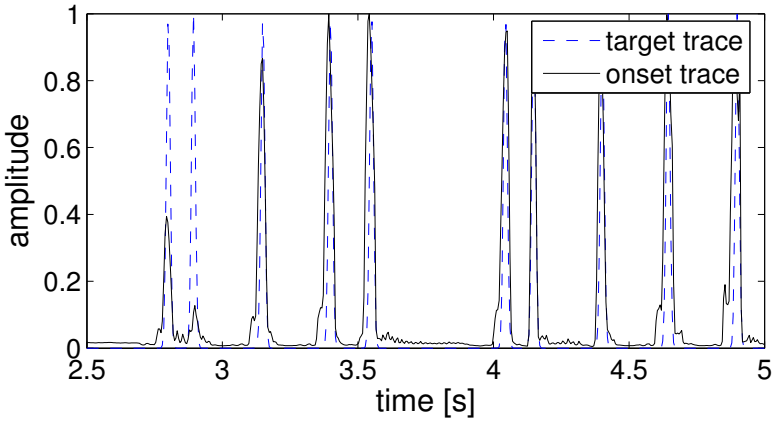

Figure 10: The target trace represents the ideal curve for the peak picking part of the algorithm. The onset trace shows the merged output of the neural network.

([26] section 10.1). However, to speed up development time, we have evaluated the peak-picking function on the training dataset for 25 different values of K, ranging from 0.02 to 0.5 and picked the best value of K.

3.4 Multi-net variant

In order to improve the performance of our algorithm we decided to use a mixture of neural networks. Specifically, we trained the SINGLE-NET algorithm 7 times with different hyper-parameters. For each of the SINGLE-NET networks, a trace representing the tempo was calculated from the predicted onsets. Finally another FNN was used to combine the onset traces tempo traces

This model is more complex than the SINGLE-NET but did work better in the MIREX 2005 contest, outperforming SINGLE-NET by 1.7% of F-measure and winning first place. Details of the tempo trace computation and of the merging procedure are explained in the following sections.

3.4.1 Tempo Trace

The SINGLE-NET variant has access only to short-timescale information available from near-neighbor frames. As such it is unable to discover regularities that exist at longer timescales. One important regularity is tempo. The rate of note production is useful for predicting note onsets. Our approach in MULTI-NET variant is to calculate a tempo trace and to use that tempo trace to condition the probability that a particular point in time is an onset by checking whether it is in phase with the predicted tempo.

Our approach is to compute the inter-onset histogram of a particular point in the onset trace and compare it with the inter-onset histogram of all other onsets. If the two histograms

are correlated, this indicates that this point is in phase with the tempo.

$$\Gamma(t) = h\left(\{\mu_i - \mu_j\}_{ij}\right) \cdot h\left(\{\mu_i - t\}_i\right) \tag{18}$$

where $\Gamma(t)$ is the tempo trace at time t, $h(S)$ is the histogram of set S and μ_i is the i^{th} onset. The dot product between the two histograms is the measure of correlation.

This method calculates n histograms, with each of them requiring time $O(n)$ to compute. Therefore, the algorithm is $O(n^2)$. Moreover, if errors occur in the peak extraction, they directly affect the results of these histograms. To compensate for this, Section 3.5 introduces a way to calculate the tempo trace directly on the onset trace by applying autocorrelation twice. This yields an algorithm with complexity $O(n \log n)$. See Figure 11 for an example.

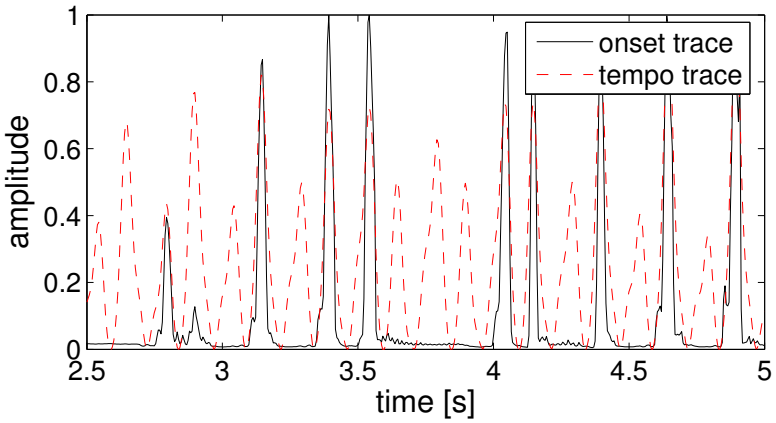

Figure 11: The onset trace shows the merged output of the neural networks as in Figure 10. The tempo trace reveals the tri-correlation of the onset trace.

3.4.2 Tempo trace confidence

The tempo trace allows the final FNN to perform categorization based not only on the ambiguity of a peak but also on whether we are *expecting* a peak or not at this particular time. In addition we provide the network with the normalized entropy of the inter-onset histogram as a measure of rhythmicity.

$$H(T) = \frac{1}{\log_2 n} \sum_{i=1}^{n} p(t_i) \log_2 p(t_i) \tag{19}$$

where the normalization factor serves to map every measure of entropy between 0 and 1. This provides the network with a measure of confidence when weighing the relative influence of the tempo.

3.4.3 Merging information

In order to merge information for the MULTI-NET variant of our approach we simply concatenate all the onset traces from our multiple networks, along with their tempo traces (including the entropy-based prediction about rhythmicity).

This merged information yields a matrix with a sampling rate equal to the original spectrogram, but containing different information. We continue with the SINGLE-NET variant using this new feature frame in place of the original spectrogram. Unlike the SINGLE-NET variant, the input window takes into account 100% of the frequency spectrum. That is, no sliding window over frequency is used because there is no longer any continuity over frequency in the features we extracted.

3.5 Tempo trace by autocorrelation

In this section, we review autocorrelation and tempo induction. We then show that Equation 18 can be calculated directly on the onset trace by applying the autocorrelation function twice to the onset trace.

3.5.1 Autocorrelation and tempo

Autocorrelation works by transforming a signal from the time domain into the frequency domain and back again, discarding phase along the way. Autocorrelation provides a high-resolution picture of the relative salience of different periodicities, thus motivating its use in tempo and meter related music tasks. However, the autocorrelation transform discards all phase information, making it impossible to *align* salient periodicities with the music. Thus autocorrelation can be used to predict, for example, that music has something that repeats every 1000ms but it cannot say *when* the repetition takes place relative to the start of the music.

Autocorrelation is certainly not the only way to compute a tempo trace. Adaptive oscillator models [21, 12] can be thought of as a time-domain correlate to autocorrelation based methods and have shown promise, especially in cognitive modeling. The integrate-and-fire neural network from [23] can be viewed as a such an oscillator-based approach. Multi-agent systems such as those by Dixon [9] have been applied with success, as have Monte-Carlo sampling [6] and Kalman filtering methods [7].

Many researchers have used autocorrelation to find tempo in music. Brown [3] was perhaps the first to use autocorrelation to find temporal structure in musical scores. Scheirer [27] extended this work by treating audio files directly. Tzanetakis and Cook [29] used autocorrelation to generate a Beat Histogram as a feature for music classification. They perform peak-picking as part of computing the beat histogram whereas peak-picking is our primary goal here. Toiviainen and Eerola [28] and Eck [13] both used autocorrelation to predict the meter in musical scores. Klapuri et al. [20] incorporate the signal processing approaches of Goto [16] and Scheirer in a model that analyzes the period and phase of three levels of the metrical hierarchy. Eck [14] introduced a method that combines the computation

of phase information and autocorrelation so that beat induction and tempo prediction could be done directly in the autocorrelation framework.

3.5.2 Tempo trace by autocorrelation

We shall now prove that a tempo trace based on inter-onset histograms can be calculated via autocorrelation. To start, let us assume that the inter-onset histogram is equal the autocorrelation of the onset trace. (In fact this is the case, as is shown below).

$$h(t) = \gamma \star \gamma \tag{20}$$

where $h(t)$ is the inter-onset histogram for inter-onset time t, γ is the original onset trace and $\star$ is the cross correlation operator. Using this to rewrite equation (18) gives:

$$\begin{aligned}
\Gamma(t) &= \int h(t'')\,(\gamma \star \delta_t)\,dt'' \\
&= \int h(t'') \left(\int \gamma(t')\,\delta(t' - t + t'')\,dt' \right) dt'' \\
&= \int h(t'')\gamma(t + t'')\,dt'' \\
&= (\gamma \star \gamma) \star \gamma
\end{aligned} \tag{21}$$

where $\Gamma(t)$ is the tempo trace at time t and $\delta_t \equiv \delta(\tau - t)$ where δ is the delta Dirac.

Therefore, the tempo trace can be calculated by correlating the onset trace 3 times with itself. This operation takes now time $O(n \log n)$ which is much faster than the $O(n^2)$ required by Equation 18.

3.5.3 Inter-onset histogram by autocorrelation

What remains is to demonstrate that the inter-onset histogram of a peaked trace is in fact equal to the autocorrelation of a peaked trace. To achieve this we first show that the autocorrelation of the sum of a function is the pairwise cross-correlation of all functions.

$$\begin{aligned}
f(t) &\equiv \sum_i g_i(t) \\
f(t) \star f(t) &= \mathfrak{F}\left[|F(k)|^2\right] \\
&= \mathfrak{F}\left[\sum_{ij} \overline{G_i(k)} G_j(k)\right] \\
&= \sum_{ij} g_i(t) \star g_j(t)
\end{aligned} \tag{22}$$

where $F(k)$ and $G_i(k)$ are respectively the Fourier transform of $f(t)$ and $g_i(t)$

It is a known result that the cross-correlation of two Gaussians is another Gaussian with the new mean given by $\mu_1 - \mu_2$ and the new variance is $\sigma_1^2 + \sigma_2^2$.

$$N(t; \mu_1, \sigma_1) \star N(t; \mu_2, \sigma_2) = N\left(t; (\mu_1 - \mu_2), \sqrt{\sigma_1^2 + \sigma_2^2}\right) \tag{23}$$

where

$$N(t; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(t-\mu)^2}{\sigma^2}} \tag{24}$$

If we approximate the onset trace as being a mixture of Gaussians

$$\gamma(t) = \sum_i \alpha_i N(t; \mu_i, \sigma_i) \tag{25}$$

then, using Equation 22 and Equation 25, we can rewrite the autocorrelation of the onset traces

$$\gamma(t) \star \gamma(t) = \sum_{ij} (\alpha_i N(t; \mu_i, \sigma_i)) \star (\alpha_j N(t; \mu_j, \sigma_j)) \tag{26}$$

and with Equation 23, Equation 26 becomes

$$\sum_{ij} \alpha_i \alpha_j N\left(t; (\mu_i - \mu_j), \sqrt{\sigma_i^2 + \sigma_j^2}\right) \tag{27}$$

which is a more general case of a Parzen window histogram. The traditional case is where α_i and σ_i remain constant across points. This loss of information occurs when we extract the peaks from the onset trace, keeping only the position and ignoring the width and the height.

4 Dataset

For this project, two different datasets were used. For the MIREX 2005 note onset detection contest, we used the publicly-available Pierre Leveau's dataset [25] to train our model. The dataset can be found at `http://www.lam.jussieu.fr/src/Membres/Leveau/SOL/SOL.htm`. However this dataset is quite small and mainly composed of monophonic songs. To extend our tests and to achieve better robustness, we developed our own dataset. It is composed of 59 hand labeled songs, extracted from the "Ballroom" dataset from the ISMIR 2004 tempo contest [17].

4.1 The ISMIR 2005 audio onset detection dataset

To train our model for the ISMIR 2005 contest we used a dataset comprised of 22 songs. 17 of these songs come from Pierre Leveau's dataset [25]. Though mostly monophonic, these songs cover a wide range of instruments types and two of them contain singing. In addition we added 5 complex polyphonic MIDI songs rendered as PCM audio. These MIDI songs were popular punk songs found on the Internet. Combining these songs yields a dataset with 935 labels and a total duration of 292 seconds.

4.2 A larger dataset

Our focus is on creating an algorithm that can detect onsets over a wide range of musical styles, particularly popular music styles. Therefore, to explore the behavior of the model more completely than was possible at ISMIR 2005, we needed a more appropriate dataset. We chose the "Ballroom" dataset from ISMIR 2004.

The "Ballroom" dataset is composed of 698 wav files of approximately 30 seconds each. Annotating the complete dataset takes too much time and is not necessary for our tests. We therefore have annotated 59 random segments of 10 seconds each. Most of them are complex polyphonic with singing, mixed with pitched and noisy percussion.

To help us label this dataset, we have built a Matlab graphical user interface (Figure 12). This interface is similar to the one used in [25], but is more suited to our needs. This tool can be used used both to label datasets directly and to verify where a labeling algorithm has made errors and whether a mismatch stems from model error or incorrect labeling of the music.

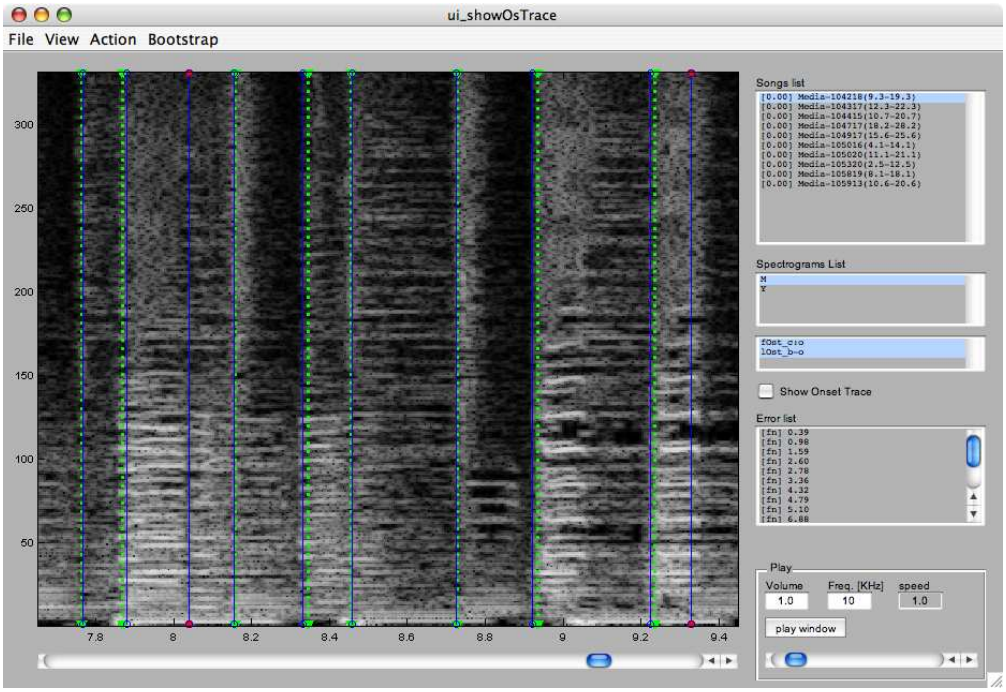

Figure 12: This is a Matlab graphical user interface developed for annotating onsets. One can add, move or delete onsets labels by clicking on the spectrogram. The window can be played at different speed and a cursor shows the current playing position on the spectrogram. The SINGLE-NET can be used to bootstrap the labeling.

The interface provides different ways of analyzing a music segment in order to determine the location of onsets. Different spectrogram types can be selected: STFT magnitude, STFT phase difference, and Constant-Q. One can zoom to a particular window in time and listen to this section at different speeds. A moving cursor helps identifying the location of the

Plane	Spectral window size	F-meas train	F-meas valid
STFT log mag	10 ms	86 ± 2	86 ± 5
STFT log mag	30 ms	86 ± 1	86 ± 5
STFT log mag	100 ms	84 ± 2	83 ± 8
C-Q log mag	10 ms	86 ± 2	86 ± 5
C-Q log mag	30 ms	**87 ± 2**	**87 ± 5**
C-Q log mag	100 ms	84 ± 2	84 ± 6
STFT ph accel	10 ms	49 ± 2	49 ± 4
STFT ph accel	30 ms	47 ± 1	47 ± 5
STFT ph accel	100 ms	49 ± 4	47 ± 6
STFT ph freq-diff	10 ms	62 ± 2	61 ± 6
STFT ph freq-diff	30 ms	80 ± 1	79 ± 4
STFT ph freq-diff	100 ms	74 ± 2	73 ± 6
Noise	—	40 ± 2	40 ± 6

Table 1: Results for running the FNN on different kinds of representations. Constant-Q performed the best, but the difference between Constant-Q and STFT is not significant. Phase acceleration did slightly better than noise, and phase difference across frequency yielded results almost as good as STFT.

onset. Labeling is done by double-clicking at the desired place on the spectrogram. Existing labels can be selected to either be moved or deleted. To bootstrap the labeling, an existing onset detector like SINGLE-NET can be used.

5 Results

To choose among different methods and different hyperparameters, we tested the SINGLE-NET algorithm using 3-fold cross validation on the "Ballroom" dataset (Section 4.2).

For those tests, parameters not specified are assumed to be the default as specified here: Input window size is 150 ms, sampling rate is 200 Hz, number of input variables is 150, number of hidden units in layer one is 18, number of hidden units in layer two is 15 and the Hamming window size is 30 ms.

The first test we made is to determine which plane is appropriate for detecting onsets. We tested the logarithm of the magnitude of the STFT, the logarithm of the amplitude of the Constant-Q transform, the phase acceleration and the phase difference along the frequency axis. For each of these, we evaluated model performance for different window widths. Table 1 shows the results for these tests. The best performance was achieved with the Constant-Q transform, but the difference between Constant-Q and STFT is not significant. The exact window width is not crucial provided it is small enough. The phase acceleration performed only slightly better than noise; however the phase difference along frequency axis worked much better, performing almost as well as the STFT magnitude plane.

Input Window width	Nb input variables	F-meas train	F-meas valid
450 ms	200	86 ± 2	86 ± 6
300 ms	200	86 ± 2	86 ± 6
150 ms	200	86 ± 2	86 ± 5
75 ms	200	85 ± 2	84 ± 5
300 ms	100	84 ± 2	84 ± 6
300 ms	200	86 ± 2	86 ± 6
300 ms	400	87 ± 2	87 ± 5
300 ms	800	87 ± 2	87 ± 6

Table 2: Results for testing different input window sizes and different numbers of input variables. Above the number of input variables is held constant at 200. Below the input window width is held constant at 300 ms. It is shown that the input window width is not crucial provided it is large enough. However, the number of input variables is important.

planes	Nb input variables	Hamming window Size	F-meas train	F-meas valid
STFT log mag + ph freq-diff	100	30 ms	85 ± 2	84 ± 5
STFT log mag + ph freq-diff	100	50 ms	85 ± 1	84 ± 7
STFT log mag + ph freq-diff	100	100 ms	80 ± 2	79 ± 8
STFT log mag + ph freq-diff	200	30 ms	86 ± 2	86 ± 5
STFT log mag + ph freq-diff	200	50 ms	86 ± 2	85 ± 6
STFT log mag + ph freq-diff	200	100 ms	84 ± 2	84 ± 7

Table 3: Results from tests combining different planes as input to the network. Unfortunately, the addition of phase difference in the frequency axis does not yield better results than the STFT log magnitude alone.

We then evaluated the input window width and the number of input variables on the magnitude plane of the STFT. Table 2 shows that the input window width size is not crucial provided it is not too small. However, the number of input variables is indeed important, with saturation occurring at around 400.

Table 1 suggests that combining the magnitude plane with the phase plane might provide better results. The test was performed with different numbers of input variables and different Hamming window sizes. The number of input variables corresponds to the number of points for each plane. Unfortunately, this combination doesn't yield better results.

Both variants of our algorithm were entered in the MIREX 2005 audio onset detection contest. The MIREX 2005 dataset is composed of 30 solo drum songs, 30 solo monophonic pitched songs, 10 solo polyphonic pitched songs and 15 complex mixes. On this dataset, the Multi-net algorithm performed slightly better than the Single-net algorithm. Multi-net yielded an F-measure of 80.07% while Single-Net yielded an F-measure of 78.35%. See Table 4.

Both variants of the algorithm were designed to perform well on a wide range of music,

Variant	MULTI-NET	SINGLE-NET
Overall Average F-measure	80.07%	78.35%
Overall Average Precision	79.27%	77.69%
Overall Average Recall	83.70%	83.27%
Total Correct	7974	7884
Total False Positives	1776	2317
Total False Negatives	1525	1615
Total Merged	210	202
Total Doubled	53	60
Runtime (s)	4713	1022

Table 4: Overall results of the MIREX 2005 onset detection contest for our two variants. Their F-measures were the two highest. They also had the best balance between the Precision and Recall. This is probably due to to the learned threshold in the peak-picking part.

so were less efficient than other algorithms on monophonic songs. But when all songs are considered, MULTI-NET and SINGLE-NET were the two best-performing entries in the contest. Both variants also showed a good balance between Precision and Recall. This advantage is likely due to the learned threshold in the peak picking part (Section 3.3).

6 Discussion

An in-depth analysis of model errors shows that most of the false negatives are produced by pitched onsets with thin harmonics. This is surprising, because such onsets are easily perceived by human. Our failure here is likely due to the fact that we only pick a random subset of the variables from the input window. Picking more variable helps, but for some pitched sounds so few variables are responsible for coding the onset that the FNN still fails.

This problem explains why our entry performed poorly on the category solo bars and bells. We had an F-measure of 86.55% where the best for this category was 99.28%.

False positives, as expected, are mainly generated by singing or vibrato. But the algorithm is still quite robust for those events.

There are also some boundary effects. At the beginning and at the end of sequences, the network was often unable to adequately resolve onsets. One solution to this problem could be to train three different networks, one that predicts onset using only information from the past, a second that uses only information from the future and a third one (like the current model) that incorporates past and future frames. For the first few frames, we could use the "future-only" version, for the last frames, the "past-only" model and for all other frames the "past-future" version. Moreover, the causal "past-only" version could also be used for online detection.

On a more general note, we mentioned several tasks that might benefit from good audio onset detection, such as tempo detection, classification and fingerprinting . This is not to say that onset detection is required for tasks like these. In fact, the MIR community seems

mixed on the usefulness of onset detection in this domain: of the 13 entries in the MIREX 2005 Tempo Contest, only 4 of them used detected onsets or onset energy functions [24]. This may be due to a philosophical rejection of onset detection as a part of tempo finding. Scheirer [27] argued, for example, that explicit note detection was not evident in the auditory system and not necessary for tempo and beat analysis. However it could also be simply due to the fact that onset detection algorithms have, to date, not worked very well. For example, this was the main reason that the second author of this paper did not use an onset detector in his MIREX entry [15].

6.1 Future work

Though our results are relatively good, there is still much room for improvement. The ability to perform good pitch detection would definitively improve model performance for notes that have thin harmonics. Another way would be to train a second network on a dataset of pitched onsets.

Different kind of machine learning approaches can also be used for this problem. Convolutional networks [22] would be able to use a wider window and take advantage of all input variables while still employing a reasonable amount of parameters. Moreover, as the convolution is done in the frequency domain, it could be done quickly.

Working on a low-dimensional set of feature instead of the entire spectrogram could provide speed improvements and could yield good results with a lower capacity network.

6.2 Execution speed

Despite the fact that our algorithm performed well at MIREX, it was also significantly slower than other entries. It is worth noting that this is not inherent to our approach but mainly due to our implementation and some limits on the evaluation procedure.

First the Multi-net variant required running Single-net seven times but only gave slightly better results. Therefore, if speed is of concern, one should avoid Multi-net.

A second major point is that there was a bug in the M2K framework for the Matlab itinerary on Windows. For this reason, we had to remove the -nosplash -nodesktop options, requiring that the complete Matlab desktop be loaded and closed for each song.

Finally, we believe that implementing the Single-net algorithm in C++ would yield performance comparable to the other algorithms.

7 Conclusions

We have presented an algorithm that adds a supervised learning step to the basic onset detection framework of signal transformation, feature enhancement and peak picking. Our Single-net variant used a single feed-forward neural network to enhance spectrogram frames for peak picker. Our Multi-net variant combined the predictions of several Single-net networks with tempo traces to improve performance. Though both algorithms

were relatively slow when compared to other MIREX 2005 entries, we have identified some ways in which they can be improved. We have also identified some ways in which they can be improved. Though both models show promise we believe that the SINGLE-NET model warrants more attention due to its relative simplicity. We conclude that the general approach of supervised learning makes sense in the domain of audio note onset detection.

8 Appendix

8.1 Summary of MIREX 2005 Audio Onset Detection Results

The goal of the contest was to evaluate and compare onset detection algorithms applied to audio music recordings. The dataset consisted of 85 audio files (14.8 minutes total) from 9 classes: complex, poly pitched, solo bars and bells, solo brass, solo drum, solo plucked strings, solo singing voice, solo sustained strings, solo winds. This information is summarized from `http://www.music-ir.org/evaluation/mirex-results/audio-onset/index.html`

Rank	**Participant**	**Avg F-measure**	**Avg Precision**	**Avg Recall**
1	Lacoste & Eck (MULTI-NET)	80.07%	79.27%	83.70%
2	Lacoste & Eck (SINGLE-NET)	78.35%	77.69%	83.27%
3	Ricard, J.	74.80%	81.36%	73.70%
4	Brossier, P.	74.72%	74.07%	81.95%
5	Röbel, A. (2)	74.64%	83.93%	71.00%
6	Collins, N.	72.10%	87.96%	68.26%
7	Röbel, A. (1)	69.57%	79.16%	68.60%
8	Pertusa, Klapuri, & Iñesta	58.92%	60.01%	61.62%
9	West, K.	48.77%	48.50%	56.29%

Table 5: Overall scores from MIREX 2005 Audio Onset Detection contest. Overall Average F-Measure, Overall Average Precision and Overall Average Recall are weighted by number of files in each class.

References

[1] Bello, J. P., Duxbury, C., Davies, M., and Sandler, M. (2004). On the use of phase and energy for musical onset detection in the complex domain. In *IEEE Signal Processing Letters*, volume 11.

[2] Bello, J. P. and Sandler, M. (2003). Phase-based note onset detection for music signals. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing. ICASSP-03*.

[3] Brown, J. (1993). Determination of meter of musical scores by autocorrelation. *Journal of the Acoustical Society of America*, 94:953–1957.

[4] Brown, J. C. (1991). Calculation of a constant q spectral tansform. *Journal of the Acoustical Society of America*, 89(1):425–434.

[5] Brown, J. C. and Puckette, M. S. (1992). An efficient algorithm for the calculation of a constant q transform. *J. Acoust.Soc.Am.*, 92(5):2698–2701.

[6] Cemgil, A. T. and Kappen, H. J. (2003). Monte Carlo methods for tempo tracking and rhythm quantization. *Journal of Artificial Intelligence Research*, 18:45–81.

[7] Cemgil, A. T., Kappen, H. J., Desain, P., and Honing, H. (2001). On tempo tracking: Tempogram representation and Kalman filtering. *Journal of New Music Research*, 28:4:259–273.

[8] Davy, M. and Godsill, S. (2002). Detection of abrupt spectral changes using support vector machines. an application to audio signal segmentation. In *IEEE ICASSP 2002*, Orlando, USA.

[9] Dixon, S. E. (2001). Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research*, 30(1):39–58.

[10] Duxbury, C., Bello, J. P., Davies, M., and Sandler, M. (2003a). A combined phase and amplitude based approach to onset detection for audio segmentation. In *Proceedings of the 4th European Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS-03)*, London, UK.

[11] Duxbury, C., Bello, J. P., Davies, M., and Sandler, M. (2003b). Compled domain onset detection for musical signals. In *Proc. of the 6th Int. Conference on Digital Audio Effects (DAFx-03)*, London, UK.

[12] Eck, D. (2002). Finding downbeats with a relaxation oscillator. *Psychol. Research*, 66(1):18–25.

[13] Eck, D. (2004). A machine-learning approach to musical sequence induction that uses autocorrelation to bridge long timelags. In Lipscomb, S., Ashley, R., Gjerdingen, R., and Webster, P., editors, *The Proceedings of the Eighth International Conference on Music Perception and Cognition (ICMPC8)*. Causal Productions.

[14] Eck, D. (2005). Meter and autocorrelation. In *10th Rhythm Perception and Production Workshop (RPPW) 2005*, Alden Biesen, Belgium.

[15] Eck, D. and Casagrande, N. (2005). A tempo-extraction algorithm using an autocorrelation phase matrix and shannon entropy. MIREX tempo extraction contest (www.music-ir.org/evaluation/mirex-results).

[16] Goto, M. (2001). An audio-based real-time beat tracking system for music with or without drum-sounds. *Journal of New Music Research*, 30(2):159–171.

[17] Gouyon, F., Klapuri, A., Dixon, S., Alonso, M., Tzanetakis, G., Uhle, C., and Cano, P. (2005). An experimental comparison of audio tempo induction algorithms.

[18] Kapanci, E. and Pfeffer, A. (2004). A hierarchical approach to onset detection. In *Proceedings of the International Computer Music Conference*, Miami, Florida.

[19] Klapuri, A. (1999). Sound onset detection by applying psychoacoustic knowledge.

[20] Klapuri, A., Eronen, A., and Astola, J. (2006). Analysis of the meter of acoustic musical signals. *IEEE Trans. Speech and Audio Processing*, 14(1).

[21] Large, E. W. and Kolen, J. F. (1994). Resonance and the perception of musical meter. *Connection Science*, 6:177–208.

[22] LeCun, Y. and Bengio, Y. (1995). Convolutional networks for images, speech, and time-series. *In M. A. Arbib, editor, The Handbook of Brain Theory and Neural Networks. MIT Press.*

[23] Marolt, M., Kavcic, A., and Privosnik, M. (2002). Neural networks for note onset detection in piano music. In *Proceedings of the International Computer Music Conference.*

[24] McKinney, M. and Moelants, D. (2005). Mirex 2005: Tempo contest. In *Proc. 6th International Conference on Music Information Retrieval (ISMIR 2005).*

[25] Pierre Leveau, L. D. and Richard, G. (2004). Methodology and tools for the evaluation of automatic onset detection algorithms in music. In *Proc. of the 5th International Conference on Music Information Retrieval (ISMIR)*, Barcelona.

[26] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1993). *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, 2nd edition.

[27] Scheirer, E. (1998). Tempo and beat analysis of acoustic musical signals. *Journal of the Acoustical Society of America*, 103(1):588–601.

[28] Toiviainen, P. and Eerola, T. (2004). The role of accent periodicities in meter induction: a classificatin study. In Lipscomb, S., Ashley, R., Gjerdingen, R., and Webster, P., editors, *The Proceedings of the Eighth International Conference on Music Perception and Cognition (ICMPC8)*, Adelaide, Australia. Causal Productions.

[29] Tzanetakis, G. and Cook, P. (2002). Musical genre classification of audio signals. *IEEE Trans. on Speech and Audio Processing*, 10(5).

[30] West, K. and Cox, S. (2005). Finding an optimal segmentation for audio genre classification. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005).*