Time-Aware Neural Networks for Onset Detection

Master in Computer Engineering - Mobile Computing

João Pedro Lima Ramos

Leiria, September of 2024

Time-Aware Neural Networks for Onset Detection

Master in Computer Engineering - Mobile Computing

João Pedro Lima Ramos

Dissertation under the supervision of Professor Carlos Grilo, Professor Gustavo Reis, Professor Patrício Domingues and Professor Rolando Miragaia.

Leiria, September of 2024

Originality and Copyright

This dissertation report is original, made only for this purpose, and all authors whose studies and publications were used to complete it are duly acknowledged.

Acknowledgments

As I complete this dissertation, I would like to express my sincere gratitude to the many people who have supported me throughout this academic journey.

I am deeply indebted to my supervisors for their guidance, expertise, and patience. Your mentorship has been crucial in shaping both this dissertation and my development as a researcher.

To my family, thank you for your constant support and encouragement. Your unwavering belief in me has been a source of strength throughout this process.

To my friends, I appreciate your encouragement, understanding, and the moments of relaxation you provided when I needed them most. Your support has been an essential part of this experience.

This achievement is the result of the collective support and wisdom of all those mentioned here and many others. I am truly grateful to everyone who has been part of this journey.

Abstract

This dissertation investigates the application of time-aware neural networks to musical onset detection, building upon Böck and Schlüter's (2014) work. The research explores modifications of neural network architectures to better capture temporal aspects of music for improved onset detection accuracy. Starting with a replication of Böck and Schlüter's convolutional neural network (CNN) model, the study explores incremental modifications including batch normalization, Long Short-Term Memory (LSTM) layers, and adjustments to convolutional layer capacities.

Experiments conducted on the Böck dataset, using 8-fold cross-validation, reveal that increasing the number of feature maps in convolutional layers yields the most significant improvement. The best-performing model, a CNN with increased feature maps and batch normalization, achieves an F1-score of 0.905, outperforming variations with recurrent elements. The research also highlights the importance of multi-channel spectrogram representations for providing multi-scale temporal information.

While improvements are incremental, this study offers insights into designing time-aware neural networks for onset detection, contributing to ongoing research in music information retrieval. It reaffirms the strength of Böck and Schlüter's approach while demonstrating potential for refinement, underscoring the challenges in advancing onset detection techniques and the importance of temporal dynamics in music analysis.

Keywords: Onset Detection, Time-Aware Neural Networks, Convolutional Neural Networks, Music Information Retrieval, Deep Learning.

Resumo

Esta dissertação investiga a aplicação de redes neuronais conscientes do tempo à deteção de onsets musicais, baseando-se no trabalho de Böck e Schlüter (2014). A investigação explora modificações de arquiteturas de redes neuronais para melhor capturar os aspetos temporais da música, visando melhorar a precisão na deteção de onsets. Começando com uma replicação do modelo de rede neuronal convolucional (CNN) de Böck e Schlüter, o estudo explora modificações incrementais, incluindo normalização de lotes, camadas de Long Short-Term Memory (LSTM) e ajustes nas capacidades das camadas convolucionais.

As experiências realizadas no conjunto de dados de Böck, utilizando validação cruzada de 8 dobras, revelam que o aumento do número de mapas de características nas camadas convolucionais proporciona a melhoria mais significativa. O modelo com melhor desempenho, uma CNN com mapas de características aumentados e normalização de lotes, atinge uma pontuação F1 de 0,904, superando variações com elementos recorrentes. A investigação também destaca a importância das representações de espectrogramas multicanal para fornecer informações temporais em múltiplas escalas.

Embora as melhorias sejam incrementais, este estudo oferece insights sobre o design de redes neuronais conscientes do tempo para deteção de *onsets*, contribuindo para a investigação em curso na área de recuperação de informação musical. Reafirma a solidez da abordagem de Böck e Schlüter, demonstrando simultaneamente o potencial de refinamento, sublinhando os desafios no avanço das técnicas de deteção de *onsets* e a importância da dinâmica temporal na análise musical.

Palavras-chave: Deteção de *Onsets*, Redes Neuronais Conscientes do Tempo, Redes Neuronais Convolucionais, Recuperação de Informação Musical, Aprendizagem Profunda.

Contents

List of Figures

List of Tables

List of Abbreviations and Acronyms

AI	Artificial Intelligence
BiLSTM	Bi-directional Long Short-Term Memory
CNN	Convolutional Neural Network
DFT	Discrete Fourier Transform
DNN	Deep Neural Networks
FNN	Front Forward Neural Network
LSTM	Long Short-Term Memory
MFCCs	Mel-Frequency Cepstrum Coefficients
MIR	Music Information Retrieval
ML	Machine Learning
ODF	Onset Detection Function
RNN	Recurrent Neural Network
STFT	Short-Time Fourier Transform

1. Introduction

Music transcription, the process of converting audio recordings into musical notation, has long been a challenging task in the field of Music Information Retrieval (MIR). Within this broader challenge, onset detection - the identification of the precise moments when musical notes or events begin - plays a crucial role. Accurate onset detection forms the foundation for many MIR tasks, including beat tracking, rhythm analysis and, ultimately, the creation of accurate musical scores from audio recordings.

The advent of deep learning techniques has opened new avenues for addressing the complexities of music transcription and onset detection. This dissertation explores the application of advanced neural network architectures to the specific task of musical onset detection, building upon and extending the seminal work of Böck and Schlüter in this domain.

The importance of this research lies in its potential to enhance the ability to automatically analyze and transcribe music across diverse genres. Improved onset detection can lead to more accurate music transcription systems, benefiting areas such as musicology, music education, and audio production. Furthermore, advancements in this field contribute to a broader understanding of how machines can be trained to perceive and interpret complex auditory signals.

The primary objectives of this study are to replicate and analyze the Convolutional Neural Network (CNN) approach introduced by Böck and Schlüter for onset detection; to explore and evaluate modifications to this architecture, incorporating other advancements in deep learning such as batch normalization and recurrent neural network layers; to develop and test a novel CNN architecture that improves upon the state-of-the-art in onset detection performance; and to provide a comprehensive analysis of the strengths and limitations of different model architectures for onset detection.

To achieve these goals, the study employs a rigorous methodology involving preprocessing of audio data into suitable spectral representations; implementation and training of various neural network architectures; evaluation using standard metrics such as F1-score, precision, and recall; and cross-validation to ensure robustness of results across different data subsets.

The experiments conducted in this study use the Böck dataset, a widely recognized benchmark in the field of onset detection. This dataset comprises approximately 100 minutes of diverse musical recordings with over 25,000 manually annotated onset events, providing a comprehensive testbed for evaluating onset detection algorithms.

This work contributes a detailed replication study of the Böck and Schlüter CNN model, exploring architectural modifications such as batch normalization and recurrent layers. An enhanced CNN with increased convolutional capacity achieved a modest improvement over the original results, underscoring the enduring strength of Böck and Schlüter's approach. The study offers insights into input representations and the effectiveness of convolutional architectures in capturing temporal context for onset detection, adding to the ongoing discourse in music information retrieval.

This rest of the dissertation is structured as follows:

- **Chapter 2** provides essential background information on audio signal processing, onset detection techniques, and relevant machine learning concepts;
- **Chapter 3** reviews the state-of-the-art in onset detection, focusing on the evolution of machine learning approaches in this field;
- **Chapter 4** details the methodology employed, including data preprocessing, model architectures, and evaluation techniques;
- **Chapter 5** explores the effects of various pre-processing techniques applied to the input data;
- **Chapter 6** analyzes the modeling process, experiments, and results of the musical onset detection system, evaluating its performance and comparison to state-of-the-art methods;
- **Chapter 7** summarizes and comments on the developed work, as well as its main results, and suggests possible directions for future work.

By thoroughly examining and building upon existing approaches in onset detection, this research aims to contribute to the ongoing discourse in music information retrieval. The insights gained from this study can inform future research directions and potentially inspire new approaches to the challenging task of automatic music transcription.

2. Background

The intricate nature of audio signals and their analysis is foundational to advancements in digital audio processing, particularly in the context of onset detection. This chapter delves into the essential aspects of audio signals, their key characteristics, the theoretical foundations of their digital representation, and the transformative techniques used to extract meaningful information for onset detection.

2.1. Audio Signal Introduction

An audio signal is a representation of sound that typically uses a variation in air pressure, voltage, or digital values to represent the air pressure variations caused by the sound waves. When represented digitally, as it commonly is in modern applications, an audio signal is usually a sequence of numerical samples taken at regular intervals [1].

Audio signals can be either monophonic or polyphonic. Monophonic signals contain only one note at a time, while polyphonic signals can contain multiple notes simultaneously. This distinction is important in the context of onset detection, as detecting onsets in polyphonic music is typically more challenging than in monophonic music [2].

2.1.1. Audio Signal Characteristics

The characteristics of an audio signal - amplitude, frequency, phase, and timbre - play pivotal roles in its perception and processing:

Amplitude: This represents the strength or loudness of the signal. In the time domain, the amplitude is the height of the waveform, reflecting the degree of displacement of the sound pressure level from its equilibrium state. A higher amplitude corresponds to a louder sound. When measuring digital signals, the amplitude usually refers to the value of each sample, often represented as a digital number within a certain range [3].

Frequency: The frequency of a sound wave refers to the number of cycles it completes in a second, typically measured in Hertz (Hz). It is directly related to the perceived pitch of a sound: a higher frequency results in a higher perceived pitch. In music, different notes correspond to different frequencies [3].

Phase: Phase provides information about the position of a waveform relative to a reference point in time. Two identical sound waves that are "in phase" will add together to create a sound wave of increased amplitude. If they are "out of phase", they can cancel each other out, decreasing the overall amplitude. This can play a significant role in complex audio signals, where multiple frequencies interact, and can impact the timbre and spatial positioning of sounds [1].

Timbre: This refers to the quality or color of a sound that makes it possible to distinguish different types of sound, even when they have the same pitch and loudness. The timbre is primarily determined by the harmonic content of a sound and the dynamic characteristics of the sound such as vibrato and the attack-decay envelope of the sound [4].

All these properties can be influenced by various factors, including the environment in which the sound propagates, the nature of the sound source, and any processing the sound undergoes. For instance, real-world sounds often contain noise and reverberation that can make the onset detection task more challenging [1].

2.1.2. Sampling Theorem

The process of converting a continuous analog audio signal into a digital audio signal is called *sampling*. During sampling, the value of the signal is measured at regular intervals, creating a series of samples. The sampling rate, denoted by f_s, refers to the number of these samples taken per second and is typically measured in Hertz (Hz). For example, CD-quality audio has a sampling rate of 44.1 kHz, meaning 44.100 samples are taken per second [1]. An illustration of this process is presented in Figure 1.

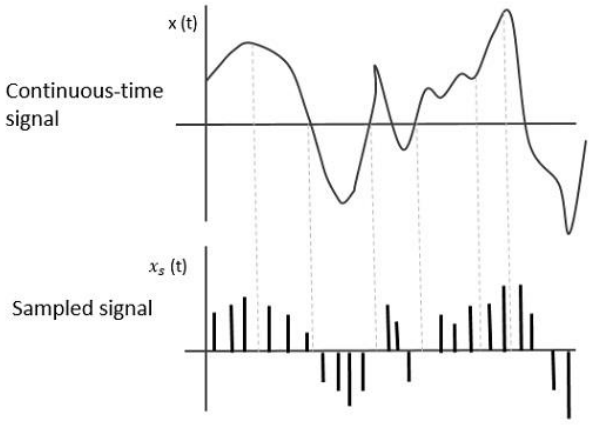

Figure 1: Illustration of the sampling process, showing both the continuous-time signal and the corresponding sampled signal [5].

The *Nyquist-Shannon sampling theorem*, a fundamental theorem in the field of information theory, provides a specific criterion to ensure the accurate digital representation of the original analog signal. The theorem states that to perfectly capture all the information in an analog audio signal without any loss or aliasing (creating misleading signals), the sampling rate must be at least twice the highest frequency contained in the original signal [6] [7].

Mathematically, the Nyquist-Shannon theorem can be expressed as:

$$f_s \geq 2f_{max},$$

where f_s is the sampling rate and f_{max} is the highest frequency present in the original analog signal.

This minimum rate, $2f_{max}$, is commonly known as the Nyquist rate, and the corresponding frequency, f_{max}, is the Nyquist frequency. If the sampling rate is lower than the Nyquist rate, a phenomenon called aliasing can occur, where high-frequency components in the signal are incorrectly perceived as lower-frequency components, leading to a distortion of the signal [6] [7].

It is important to note that the Nyquist-Shannon theorem assumes ideal conditions where the signal is perfectly bandlimited (i.e., all frequency components above f_{max} are zero) and the sampling and reconstruction processes are perfect. In practice, some additional considerations, such as filtering and quantization noise, need to be considered for high-quality audio signal processing [7].

2.2. Audio Signals Representation

Audio signals, in their raw form, are a complex amalgamation of various frequency components and temporal dynamics. To analyze, process, or even simply understand these signals, they need to be represented in forms that elucidate their underlying characteristics. This section introduces and explains the primary methods used for audio signal representation, diving into the nuances of both time and frequency domains, thereby offering a comprehensive perspective on some of the many ways in which sound can be portrayed and understood [8].

2.2.1. Time-Domain Representation

The *time-domain* representation of an audio signal allows us to visualize changes in a signal's amplitude over time, giving a bird's eye view of the signal's behavior. This is typically depicted as a waveform, as shown in the Figure 2. In this figure, the x-axis represents time, and the y-axis represents amplitude, illustrating how the signal fluctuates within a specified time frame [9].

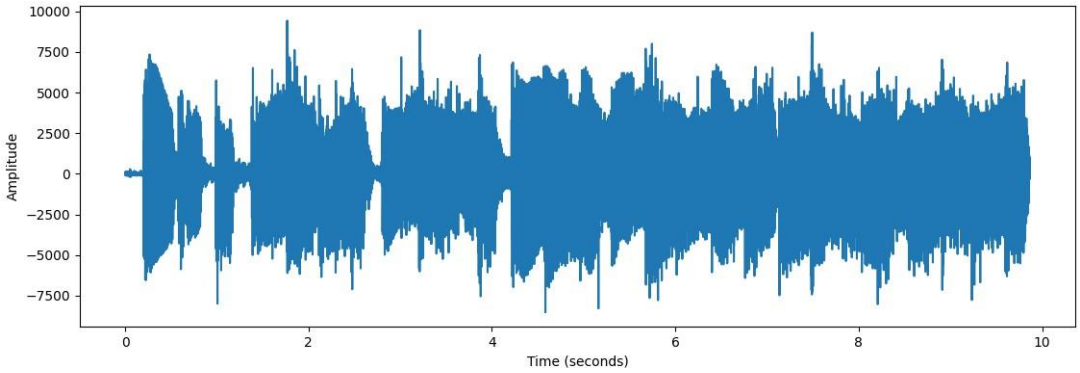

Figure 2: A time-domain waveform.

While the waveform lays down a foundational understanding of the audio signal's behavior, it is by no means the only representation in the time domain. Other notable techniques include [9]:

Envelope: This focuses on capturing the broader amplitude trends of a signal by delineating its peak values, granting an understanding of the signal's macroscopic behavior.

Zero-Crossing Rate (ZCR): This metric provides insights into the frequency of sign changes in a signal, and it is particularly useful in speech signal analysis to roughly gauge the predominant frequency.

Autocorrelation: Employed to unearth periodicities within a signal, this technique becomes pivotal, especially when pinpointing pitches or understanding rhythmic patterns.

Filtered Signal Plots: Visualizing a signal post-filtering (e.g., after applying a low-pass filter) can elucidate how specific frequency components manifest themselves in the time domain.

Each of these representations, while rooted in the time domain, provides unique lenses to analyze and interpret audio signals, especially when the objective is to discern specific characteristics or events within the signal [9].

However, the time-domain, for all its directness, comes with an intrinsic limitation. While it excels at highlighting amplitude dynamics, such as loud and quiet periods, it does not inherently divulge the frequency content embedded within these fluctuations [3]. Given the richness and importance of frequency information, especially in contexts like music or speech analysis where frequencies play pivotal roles, a mere time-domain representation often does not suffice. It necessitates a complementary frequency-domain analysis to offer a complete perspective of the audio signal [10].

2.2.2. Frequency-Domain Representation

The *frequency domain* represents an integral part of understanding the composition of an audio signal. Unlike the time-domain representation that charts how a signal changes over time, the frequency domain delineates how much of the signal lies within each given frequency band over a range of frequencies. To delineate it simply, it is like reading the musical score of a composition, wherein each note corresponds to a specific frequency [11].

The first step in translating a time-domain signal into the frequency domain is to apply the Fourier Transform. This mathematical process decomposes a function (in this case, the time-domain signal) into its constituent frequencies. In essence, it allows for the visualization of the 'ingredients' of the signal; each individual frequency component, its amplitude, and phase offset contribute to the overall audio signal [12].

Looking at a plot of a frequency-domain representation, the x-axis typically represents frequency, most often in Hertz (Hz), and the y-axis represents amplitude. Each peak in the plot indicates the presence of a frequency component in the signal. The height of each peak corresponds to the amplitude of that component, and the location of the peak along the frequency axis indicates the frequency of the component [13].

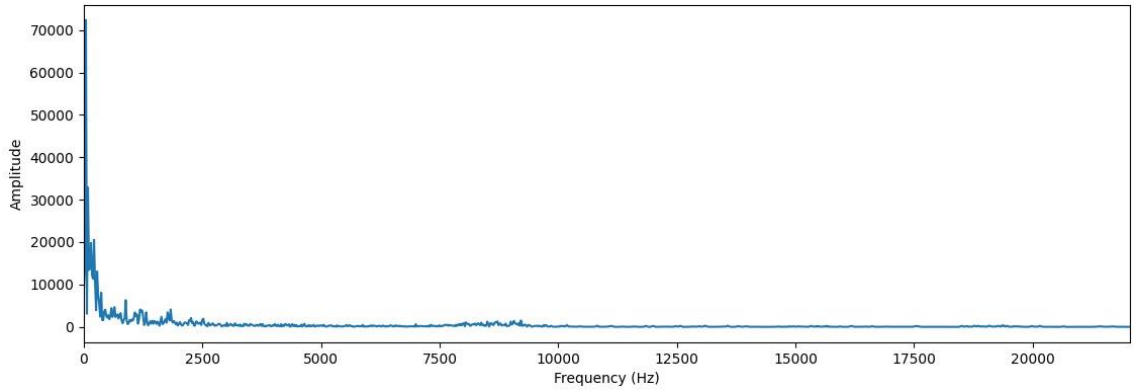

Figure 3: Same sound as Figure 2 but in the frequency domain.

Figure 3 displays the same sound as Figure 2. To read this graph, one looks at the peaks and identifies the corresponding frequencies. The highest peak, for instance, is often referred to as the dominant frequency as it represents the frequency component with the highest amplitude. In music, this frequency would often correspond to the perceived pitch of the note [1].

In essence, frequency-domain representation provides a detailed snapshot of the 'musical ingredients' in an audio signal. It provides an in-depth perspective on the audio signal's makeup, which proves invaluable when analyzing complex sounds, decomposing them into their fundamental frequencies, and detecting onsets in music [14].

2.2.3. Spectrogram Representation

Spectrograms are a fundamental tool in the analysis of audio signals, offering a visual representation of the spectrum of frequencies as they vary with time. The spectrogram is a three-dimensional display where the x-axis typically represents time, the y-axis indicates frequency, and the intensity of colors or shades of gray represents the amplitude or energy of a particular frequency at a particular time [9]. Figure 4 shows an example of a normalized spectrogram:

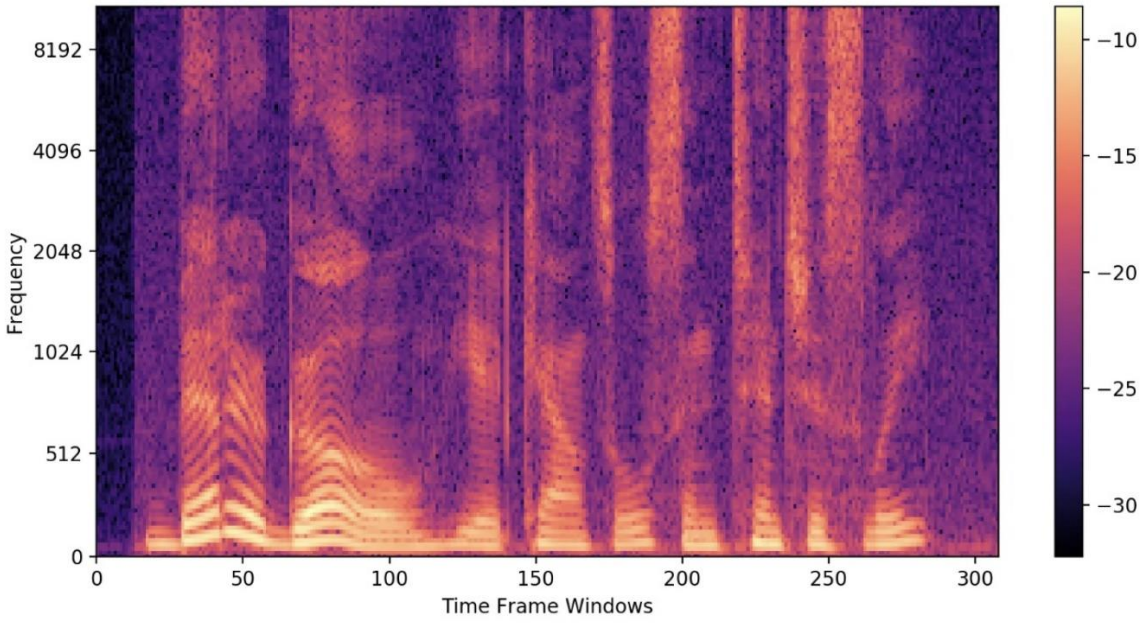

Figure 4: A normalized spectrogram [15].

For onset detection, a spectrogram can reveal sudden bursts of energy across frequencies which often correspond to the start of a sound. Techniques such as peak picking can be employed on the spectrogram to identify these onsets. By tracking changes in energy over time, it becomes possible to detect the beginnings of musical notes, speech phonemes, or other sound events [16].

Advanced methods such as logarithmic frequency scaling or Mel-frequency scaling are sometimes used to better match human auditory perception. This can make the detection of onsets more intuitive and can be particularly useful in complex audio environments [17].

2.2.4. Chromagram Representation

The *chromagram* is an essential tool in music information retrieval that represents the energy within each of the twelve different pitch classes across time. This representation is particularly useful for analyzing music where pitch content is more informative than raw frequency content, such as in harmonic and tonal music. Observing Figure 5, it is possible to see that chromagrams condense the frequency spectrum into 12 bins corresponding to the 12 semitones of the musical octave, providing a compact representation of the musical pitch content of a signal. This pitch-based approach to signal analysis is especially pertinent for tasks such as chord recognition, key detection, and, crucially, onset detection, where the beginnings of musical notes must be identified [18].

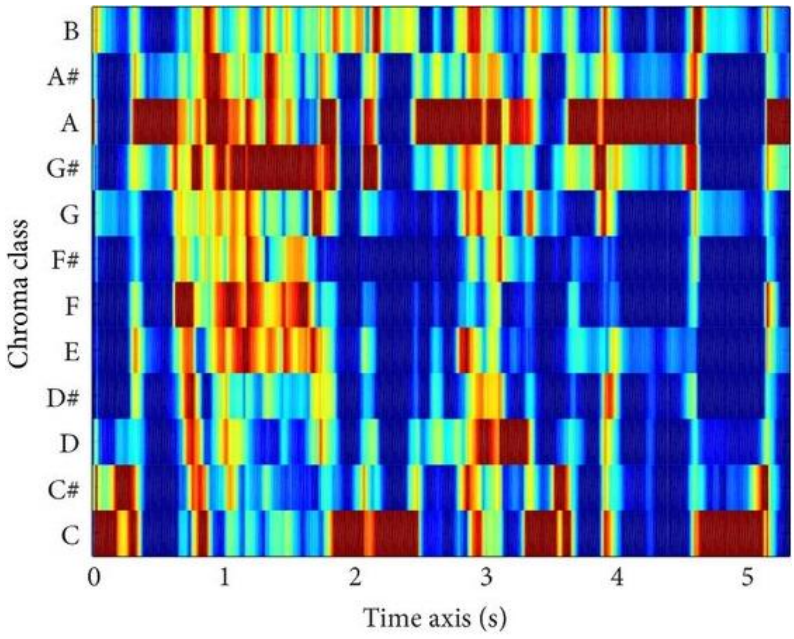

Figure 5: Example of a chromagram of a piece of music [19].

For onset detection, chromagrams can be instrumental in identifying the start of a note within a harmonic framework. By monitoring changes in chroma energy, onsets can be inferred when a particular pitch class suddenly increases in intensity, indicating the start of a new note or chord. This is reinforced by the notion that onsets in music often correspond to harmonic changes, which can be effectively captured by chromagram analysis. Furthermore, given that chroma features are invariant to octave and timbre changes, they provide a robust way to detect onsets even in polyphonic music where multiple instruments and notes are present [20].

2.3. Audio Signals Transformations

Audio signal transformations involve a diverse set of mathematical techniques designed to analyze, modify, and improve the understanding of sound data. These transformations convert audio signals from their original time-domain representation into formats that unveil underlying patterns, frequencies, and features that are not immediately apparent [21].

2.3.1. Fourier Transform

The *Fourier Transform*, named after French mathematician Jean-Baptiste Joseph Fourier, is a mathematical technique fundamental to many areas of science and engineering, including signal processing. The Fourier Transform is used to decompose a time-domain signal into its frequency components, forming a frequency-domain representation [12].

The Fourier Transform of a continuous-time signal, often termed as *Continuous Fourier Transform* (CFT), is given by the following formula [12]:

$$F(f) = \int f(t) e^{-2\pi i f t} dt.$$

Here, $f(t)$ is the original time-domain signal, $F(f)$ is the frequency-domain representation of the signal, f is frequency, t is time, i is the imaginary unit, and the integral runs over all time from $-\infty$ to ∞.

CFT allows us to see all the frequencies that make up the original time-domain signal. It essentially breaks down a complex waveform into a series of simpler sinusoidal waves that, when combined, reproduce the original signal [22].

In practice, however, most signals encountered are digital, sampled at discrete intervals in time. To handle such signals, the *Discrete Fourier Transform* (DFT) is used, which is given by [23]:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-\frac{2\pi i k n}{N}}.$$

Here, $x(n)$ represents the n-th sample of the time-domain signal, $X(k)$ is the k-th sample of the frequency-domain representation, N is the total number of samples, i is the imaginary unit, and k and n are indices of the frequency and time arrays, respectively.

The *Inverse Fourier Transform* serves as a crucial counterpart to the Fourier Transform, enabling the reconstruction of the original time-domain signal from its frequency-domain representation. This process is essential in many signal processing tasks where one needs to analyze the frequency components and then reconstruct the original signal.

For the Continuous Fourier Transform (CFT), the inverse is given by the equation:

$$f(t) = \int F(f) e^{2\pi i f t} df.$$

Here, $f(t)$ is the reconstructed time-domain signal, $F(f)$ is the frequency-domain representation, i is the imaginary unit, f is frequency, t is time, and the integral runs over all frequency from $-\infty$ to ∞. Essentially, this integral sums up all the individual sinusoidal components, each with its unique frequency f, to rebuild the original continuous-time signal [12].

For the *Discrete Fourier Transform* (DFT), the inverse transform is formulated as:

$$x(n) = \sum_{k=0}^{N-1} X(k) e^{\frac{2\pi i k n}{N}}.$$

In this equation, $x(n)$ is the n-th sample of the reconstructed time-domain signal, $X(k)$ is the k-th sample of the frequency-domain representation, N is the total number of samples, i is the imaginary unit, and k and n are indices for frequency and time arrays, respectively. This summation effectively synthesizes the original discrete-time signal by adding up the individual sinusoidal components [23].

The ability to move between time and frequency domains and back, facilitated by the Fourier Transform and its inverse, plays a pivotal role in understanding and manipulating signals. In audio processing, these tools allow for sophisticated analysis and manipulation, enabling developments in areas like audio compression, equalization, and notably, the detection of onsets in musical signals.

2.3.2. Fast Fourier Transform

The *Fast Fourier Transform* (FFT) is an algorithmic implementation of the Discrete Fourier Transform (DFT) that reduces its computational complexity, making it more efficient for large datasets. FFT is a cornerstone of digital signal processing and finds applications in several fields such as image analysis, data compression, and, of course, audio processing.

DFT, as discussed before, involves a series of complex multiplications and additions, resulting in a computational complexity of $O(N^2)$ where N is the number of data points. However, FFT reduces this complexity to $O(N \log N)$, significantly decreasing computation time for large N, by cleverly reusing results from previous computations [24].

The most common implementation of FFT, the Cooley-Tukey algorithm, assumes that the number of data points, N, is a power of 2. The algorithm works by recursively dividing DFT into smaller DFTs of size $\frac{N}{2}$, leading to a divide-and-conquer strategy that greatly speeds up the computation.

FFT is given by the following formulas, which are very similar to DFT but involve fewer computations:

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} x(n) e^{-\frac{2\pi i k n}{N}} \ (for\ even\ n),$$

$$X\left(k + \frac{N}{2}\right) = \sum_{n=0}^{\frac{N}{2}-1} x(n) e^{-\frac{2\pi i \left(k+\frac{N}{2}\right) n}{N}} \ (for\ odd\ n).$$

Here, $x(n)$ represents the n-th sample of the time-domain signal, $X(k)$ is the k-th sample of the frequency-domain representation, N is the total number of samples, i is the imaginary unit, k and n are indices for frequency and time arrays, respectively [23].

By using FFT, the computational burden of transforming a signal from the time domain to the frequency domain can be significantly reduced, making it a highly valuable tool in real-time audio processing and onset detection tasks.

2.3.3. Short-Time Fourier Transform

The *Short-Time Fourier Transform* (STFT) is a vital tool used to analyze non-stationary signals, whose spectral content changes over time, such as audio signals. STFT provides a way to understand how the frequency content of a signal evolves over time by applying the Fourier Transform to successive short segments or windows of the signal.

The mathematical expression for the STFT is as follows:

$$STFT\{f\}(m,\omega) = \sum_{n=-\infty}^{\infty} f(n)w(n - m)e^{-i\omega n}.$$

Here, $f(n)$ is the signal to be analyzed, $w(n - m)$ is a window function centered at time m, n is the discrete time index, ω is the frequency, and i is the imaginary unit. The window function $w(n - m)$ serves to isolate a portion of the signal at time m, and this windowed signal segment is then transformed into the frequency domain [25].

Figure 6 illustrates the STFT process, showing how a signal is segmented, windowed, and transformed into the frequency domain.

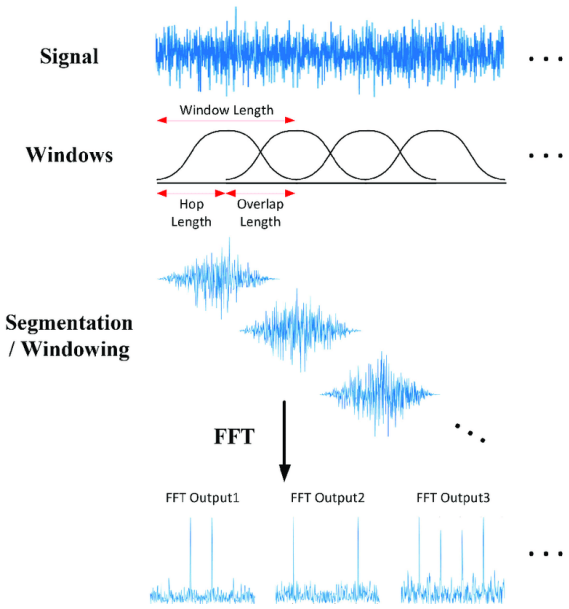

Figure 6: STFT process overview [26].

The window's length and shape can be tailored to the specific analysis needs. Common window functions include the Hamming window, the Hanning window, and the Gaussian window. The choice of the window function affects the time-frequency resolution trade-off of STFT, where a longer window provides better frequency resolution but poorer time resolution, and vice versa [27].

STFT is useful in applications like speech and music analysis, where it is crucial to understand how different frequency components evolve over time. The visual representation of STFT is often depicted as a spectrogram, where the horizontal axis represents time, the vertical axis represents frequency, and the color or intensity represents the amplitude or magnitude of the frequency component. Reading such a plot offers insights into the time-frequency characteristics of the analyzed signal, and it is extensively used in onset detection and other audio processing tasks [28].

The application of STFT in music and audio processing has proven invaluable in revealing the time-dependent spectral content of complex signals, enabling detailed analysis and manipulation for various music applications.

2.3.4. Constant-Q Transform

The *Constant-Q Transform* (CQT) is a specialized transformation widely used in music and audio signal processing. Unlike the Fourier Transform, which divides the frequency spectrum into linearly spaced bins, CQT creates bins that are spaced logarithmically. This aligns more closely with the human perception of pitch, as the "Q" factor, or quality factor, remains constant across all frequencies, hence the name Constant-Q [29].

The mathematical expression for CQT is given by:

$$CQT(k) = \sum_{k=0}^{K-1} x(n) e^{-2\pi i * (2^k * Q) * \frac{n}{N}} .$$

Here, $x(n)$ is the time-domain signal, n is the time index, N is the total number of samples, i is the imaginary unit, Q is the quality factor, and K is the total number of bins in the transform. The quality factor Q is typically defined as the ratio of the center frequency to the bandwidth for each bin and remains constant across all bins [30].

CQT provides a multi-resolution analysis that affords high frequency resolution at low frequencies and high time resolution at high frequencies. This characteristic is especially

beneficial for analyzing music signals, where different musical notes correspond to different frequencies and are perceived logarithmically [29].

A common application of CQT is the visualization of musical content using a chromagram, where the frequency bins correspond to musical notes, and the energy in each bin is mapped to a color or intensity. This provides a powerful representation for tasks such as chord recognition and onset detection [30].

2.3.5. Wavelet Transform

The *Wavelet Transform* is a mathematical tool that provides a multi-resolution analysis of signals, making it especially valuable in processing non-stationary or time-varying signals like audio. Unlike the Fourier Transform, which uses sinusoidal basis functions, the Wavelet Transform employs wavelets: functions that are localized in both time and frequency [31].

The *Continuous Wavelet Transform* (CWT) of a signal $f(t)$ is given by the following expression:

$$CWT(a,b) = \frac{1}{\sqrt{a}} \int f(t)\psi\left(\frac{t-b}{a}\right) dt.$$

Here, a is the scaling parameter, b is the translation parameter, and $\psi(t)$ is the wavelet function or mother wavelet. The scaling parameter a allows for the stretching and compressing of the wavelet, while the translation parameter b shifts the wavelet along the time axis. Together, these parameters enable the Wavelet Transform to provide detailed insights into both the frequency and time characteristics of a signal [32].

The *Discrete Wavelet Transform* (DWT) is a sampled version of the CWT and is represented as:

$$DWT(k,m) = \sum_{n=0}^{N-1} f(n)\psi(e^k * n - m).$$

Here, k and m are discrete scale and translation parameters, respectively, and n is the discrete time index. DWT is computationally more efficient and forms the basis of the wavelet decomposition and reconstruction algorithms commonly used in signal processing [33].

Wavelet transforms have become instrumental in many fields, from image compression to noise reduction in audio signals. Their ability to provide a localized frequency analysis makes them particularly suitable for analyzing signals with transient or time-varying characteristics, such as musical onsets or speech signals [34].

In addition, wavelets can be chosen or even designed to match specific signal characteristics, allowing for a more adaptive and tailored analysis [35]. This flexibility has led to the development of various types of wavelets, such as the Haar wavelet, Daubechies wavelets, and Morlet wavelets, each with unique properties suited to different applications [31].

2.3.6. Mel-Frequency Cepstrum Coefficients Transform

Mel-Frequency Cepstrum Coefficients (MFCCs) are widely used in speech and audio processing as a representation of the spectral characteristics of a signal. The computation of MFCCs consists of several stages that transform a signal into a format that resembles the human auditory system's perception [36].

Pre-emphasis: The signal is first passed through a pre-emphasis filter, which accentuates higher frequencies, typically using a simple first order filter. This is expressed by the equation:

$$F(s) = s - \alpha * s^{-1},$$

where α is commonly set to 0.97, and s represents the signal in the time domain [37].

Windowing: The signal is then segmented into overlapping frames and each frame is multiplied by a window function, often a Hamming window:

$$w(n) = x(n) * h(n).$$

Here, $h(n)$ represents the window function, and $x(n)$ and $w(n)$ are the original and windowed signals respectively [38].

Fourier Transform: The Fast Fourier Transform (FFT) is applied to each windowed frame, resulting in the frequency spectrum of each frame:

$$X(k) = \sum_{n=0}^{N-1} w(n) * \frac{e^{-2\pi ikn}}{N},$$

where $X(k)$ represents the Fourier Transform of the windowed signal [12].

Mel Filter Bank: The spectrum is converted to the Mel scale, which is a perceptual scale of pitches, using a set of triangular filters:

$$m(f) = 2595 * \log_{10}\left(1 + \frac{f}{700}\right).$$

Here, f represents the frequency in Hertz and $m(f)$ is the frequency in Mel scale [17].

Discrete Cosine Transform (DCT): Finally, a DCT is applied to the log energies in the Mel scale to obtain the MFCCs:

$$c(m) = \sum_{k=1}^{K} \log E(k) * \cos\left(\pi m * \frac{k - 0.5}{K}\right).$$

Here, $c(m)$ is the m-th MFCC and K is the number of Mel filters. $E(k)$ is the energy in the k-th Mel filter [36].

MFCCs are a robust representation of the spectral shape of an audio signal and have been widely used in various speech and audio analysis tasks [36].

2.2. Onset Detection

Onset detection, a pivotal component in Music Information Retrieval (MIR), facilitates countless tasks including beat tracking, rhythm analysis, and music transcription. An onset marks the initiation of a musical note or sound, serving as a critical element for deciphering a piece's temporal structure. The precision in identifying these onsets plays a significant role in the accurate analysis and interpretation of musical compositions, influencing the subsequent processing steps in MIR tasks [39] [40].

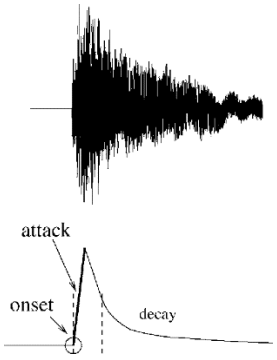

Figure 7: A visual representation of onset detection in an audio waveform [16].

As illustrated in Figure 7, onset detection involves identifying the precise moments when musical notes or sounds begin within an audio waveform. This visual representation underscores the importance of accurate onset detection for understanding and analyzing the structure of musical pieces.

The realm of onset detection extends far beyond traditional MIR tasks, finding utility in diverse fields such as music performance analysis, where it aids in the evaluation of a performer's timing and articulation, and music education, where it supports rhythm training and instrument learning. Interactive music systems, which rely on user input to generate or modify music in real-time, leverage onset detection to ensure responsive and engaging user experiences. Furthermore, the technique has been applied in clinical settings, offering insights into speech disorders by analyzing the temporal characteristics of vocal onset, thus contributing to diagnostic and therapeutic processes [3] [41].

2.3. Traditional Onset Detection

Traditional onset detection methods primarily use signal processing techniques. The process typically involves two steps: computing an *onset detection function* (ODF) and applying *peak picking* to the ODF [16], as outlined in Figure 8.

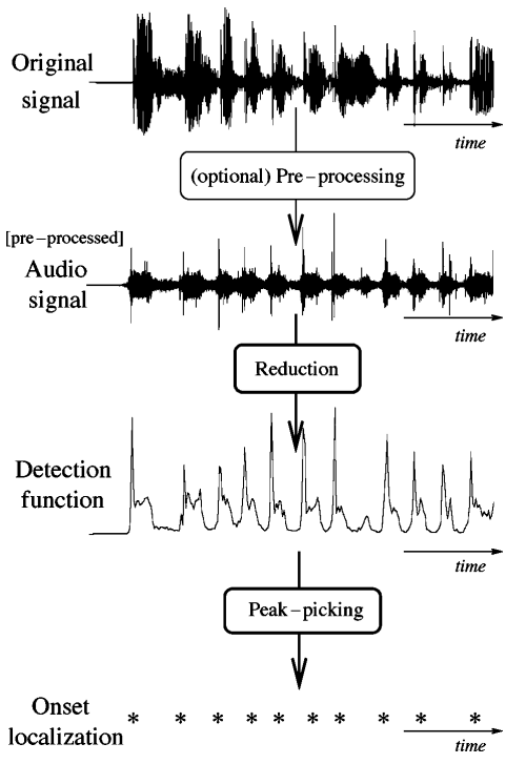

Figure 8: Traditional onset detection algorithm [16].

ODFs are functions of time which provide a probabilistic estimation of an onset at each time point. The underlying principle is that an onset in an audio signal usually corresponds to a noticeable change in its properties, which could be picked up by the ODF. There are multiple types of ODFs, each emphasizing a different characteristic of the audio signal [16]:

Energy-based ODFs: These detect onsets by looking for a significant increase in the energy of the signal. The ODF, $E(n)$, can be calculated as the difference in energy between successive frames:

$$E(n) = X(n) - X(n-1),$$

where $X(n)$ is the energy of the frame at time n [16].

Spectral-based ODFs: These focus on changes in the spectral content of the signal. One of the most common is the Spectral Flux, defined as the 2-norm of the positive part of the spectral difference:

$$F(n) = \left|\left|H(n) - H(n-1)\right|\right|_2,$$

where $H(n)$ is the magnitude spectrum at frame n.

Phase-based ODFs: These ODFs detect onsets by examining changes in the phase of the signal. The Complex Domain Method is a popular approach that combines phase and magnitude information:

$$C(n) = R(n) - R(n-1) - I(n),$$

where $R(n)$ is the real part of the spectrum, and $I(n)$ is the imaginary part [16].

High-Frequency Content ODF: This method is particularly effective for percussive onsets as it calculates the amount of high-frequency content in a signal [42]:

$$HFC(n) = \sum_{k=0}^{K-1} k\,|X_k(n)|^2,$$

where K is the number of frequency bins, k is the frequency bin index, and $X_k(n)$ is the Fourier transform of the signal.

After computing the ODF, the second step is to apply peak picking to detect the exact onset times. This is often achieved by setting a threshold and identifying points in the ODF where the function exceeds this threshold [43].

The use of traditional methods, however, comes with a range of challenges. While these methods can perform well on clean, monophonic signals, they often struggle with more complex, polyphonic music which contains overlapping notes from different instruments. Additionally, these methods can require manual fine-tuning of parameters to perform well for different types of music [40].

2.4. Onset Detection Using Machine Learning

This section examines the application of machine learning techniques to musical onset detection. It covers fundamental machine learning concepts, deep neural networks, feature extraction, model training, and evaluation methods. Special focus is given to recurrent and convolutional neural networks due to their effectiveness in processing musical data. The section aims to illustrate how machine learning enhances onset detection accuracy and adaptability compared to traditional approaches.

2.4.1. Machine Learning

Machine Learning (ML) is a field of artificial intelligence (AI) that involves the development of algorithms which allow computers to learn from and make decisions or predictions based on data. In essence, machine learning algorithms identify patterns in data and then create mathematical models based on these patterns [45].

There are several types of machine learning including supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. For a task like onset detection, supervised learning is typically used, as it involves training the algorithm on a dataset where the correct labels, indicating the presence (1) or absence (0) of onsets at specific times, are known [44].

Machine Learning provides the foundation for many complex systems, including neural networks, and it continues to be a vibrant and evolving field of research.

2.4.2. Deep Neural Networks

Deep Neural Networks (DNNs) are a specific type of machine learning model that have proven to be particularly effective at many tasks, including onset detection. They are called 'deep' because they have multiple layers of neurons. These layers enable the network to learn hierarchical representations of the data, with each layer learning to recognize more complex patterns based on the output of the previous layer [45].

The architecture of a typical DNN consists of an input layer, multiple hidden layers, and an output layer. Each layer contains several nodes, or neurons, which perform simple computations on the data. The results of these computations are passed on to the next layer. This structure allows DNNs to model complex relationships between inputs and outputs, as illustrated in Figure 9.

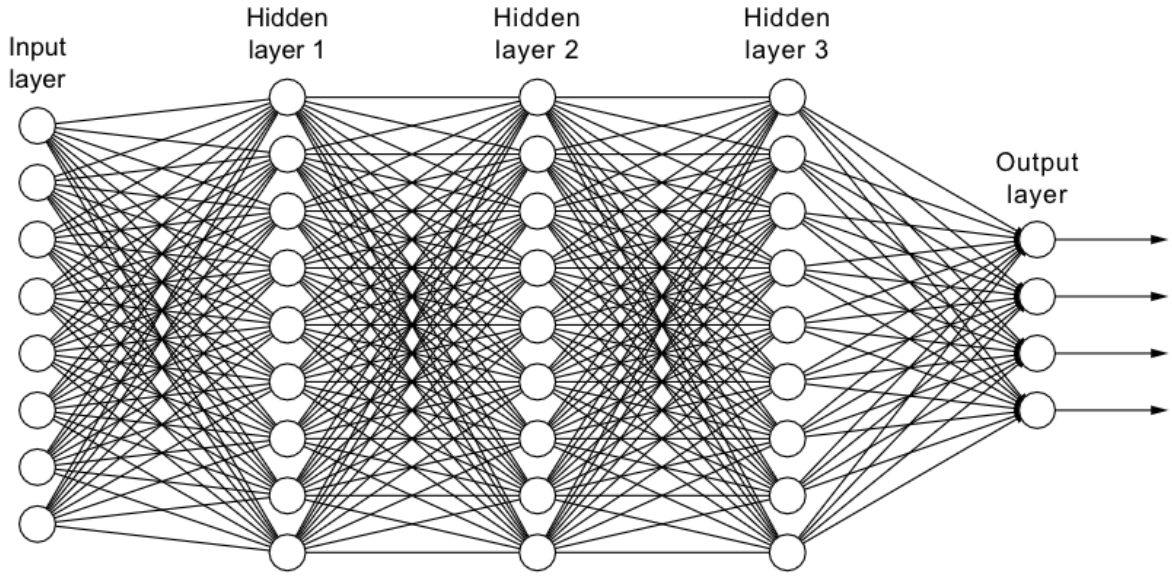

Figure 9: An example of a fully connected neural network with three hidden layers [46].

In the case of onset detection, the inputs could be various features extracted from the audio signal, such as spectral content or amplitude envelope, and the output would be a binary classification for each time frame, indicating whether an onset is present (1) or not (0). The DNN is trained on a dataset of labeled examples and learns to associate the input features with the correct binary output for each frame. The actual onset times can then be derived from these binary predictions in a post-processing step.

2.4.3. Feature Extraction

Before training a machine learning model, it is often necessary to extract relevant features from the data. Feature extraction is the process of transforming raw data into a format that can be effectively processed by the model. In the context of onset detection, features could be extracted from the audio signal which highlight the characteristics that are most relevant for identifying onsets.

Common features for onset detection include spectral contrast, spectral flux, zero-crossing rate, and chroma features, among others. These features can provide important information

about the frequency content, amplitude, and harmonic structure of the audio signal, all of which can help the model identify onsets [16].

2.4.4. Training a Machine Learning Model

Training a Machine Learning (ML) model involves iteratively adjusting the model's internal parameters to optimize its performance on a dataset. This process, often referred to as learning, is crucial for the model to accurately map input data (features) to the correct outputs (labels) [47].

One common approach to training ML models, particularly neural networks, involves Stochastic Gradient Descent (SGD) to minimize a loss function that measures the model's error [48]. This optimization process relies on the backpropagation algorithm, introduced by Rumelhart et al. [49], which efficiently computes the gradient of the loss function with respect to the network's weights. By propagating errors backward through the network, backpropagation enables SGD to update model parameters, allowing the network to learn and improve its performance iteratively.

In each iteration or 'epoch' of the training process, the model generates predictions for a batch of data, and the loss function is calculated. The gradient of the loss function, which represents the direction of steepest ascent, is then computed. However, since the goal is to minimize the loss, the model parameters are updated in the opposite direction, i.e., the direction of steepest descent [45].

Each parameter in the model is then updated by subtracting a fraction (determined by a hyperparameter called the learning rate) of the computed gradient. This adjustment shifts the parameters towards a region in the parameter space where the model's loss is lower.

The process of prediction, loss calculation, gradient computation, and parameter update is repeated over multiple epochs until the model's performance on the training data converges, i.e., the change in loss becomes negligible, or some other stopping criterion is met.

Note that while SGD and its variants, like Adam, are commonly used, they are not the only optimization algorithms available for training ML models. Other algorithms like AdaGrad, RMSProp, and BFGS also find use depending on the specific requirements and nature of the problem [50].

Through training, ML models learn to generalize from the provided examples, enabling them to make accurate predictions when presented with unseen data.

2.4.5. Evaluation and Testing

Evaluation and testing are crucial stages in the ML pipeline, as they provide an indication of the model's generalization capability, i.e., its ability to perform accurately on unseen data. After training, models are evaluated on a separate dataset, referred to as the test set, which is distinct from the data used for training [51]. The model's predictions on the test data are compared with the true values, and this comparison is quantified using specific evaluation metrics. The choice of metrics typically depends on the problem at hand. For onset detection tasks, popular metrics include precision, recall, and the F1-score [40].

Precision (P) is the proportion of correctly identified onsets to the total number of predicted onsets. It is calculated as follows:

$$P = \frac{TP}{TP + FP},$$

where TP stands for True Positives (correctly identified onsets), and FP stands for False Positives (incorrectly identified onsets).

Recall (R), also known as sensitivity or true positive rate, measures the proportion of actual onsets that the model correctly identified. It is computed as:

$$R = \frac{TP}{TP + FN},$$

where FN stands for False Negatives (actual onsets that the model failed to identify).

However, both precision and recall provide only a partial view of the model's performance. To obtain a more balanced measure, the *F1-score* is used. The F1-score is the harmonic mean of precision and recall, giving equal weight to both. It is defined as:

$$F1 = \frac{2 * P * R}{P + R}.$$

A model with perfect precision and recall would have an F1-score of 1, while a model with poor precision and recall would have an F1-score closer to 0. Hence, the F1-score provides a single number that can be used to compare the performance of different models.

By using these metrics to evaluate the model on the test set, insights can be gained into how the model might perform under real-world conditions and areas where further improvement may be needed can be identified.

2.4.6. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) represent a specific category of Deep Neural Networks (DNNs) that have demonstrated impressive efficacy in a wide range of applications, including onset detection [45]. The primary strength of CNNs comes from their unique architecture which is specifically designed to process data that possess a grid-like structure, such as time-series data (audio signals) or 2-dimensional data (images).

The architecture of a CNN is characterized by the incorporation of convolutional layers, which employ a set of trainable filters or kernels. Each filter is convolved across the input data, producing a feature map that represents the presence of learned features or patterns in the input [52]. Figure 10 illustrates the typical structure of a convolutional neural network.

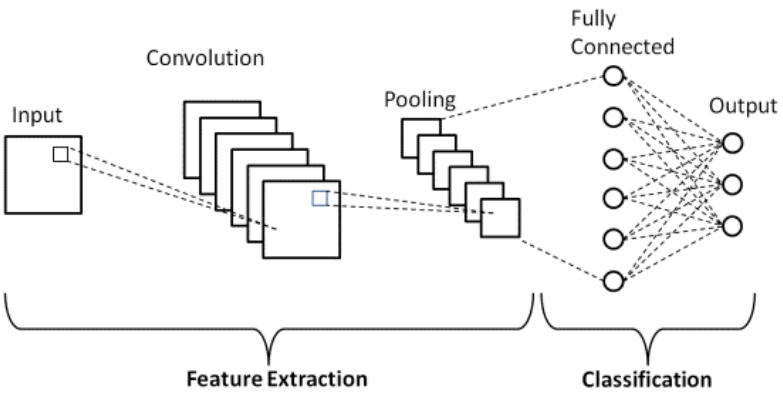

Figure 10: Typical structure of a convolutional neural network [53].

In the context of onset detection, these filters can learn to recognize and capture localized patterns in the audio signal, such as a sudden change in amplitude or frequency content, that signifies an onset. As such, the convolution operation enables CNNs to capture translational invariances in the data, that is, the ability to detect an onset regardless of its location in the input signal [45].

Furthermore, CNNs incorporate pooling layers following the convolutional layers. Pooling layers serve to progressively reduce the spatial dimensions (i.e., downsample) of the input data while retaining the most salient features, thereby reducing the computational demands of

the network, and mitigating the risk of overfitting. Common types of pooling operations include max pooling and average pooling [54].

Overall, the integration of convolutional and pooling layers in a hierarchical manner allows CNNs to learn complex and abstract representations of the input data at multiple levels of abstraction, thereby enabling robust and efficient onset detection.

2.4.7. Recurrent Neural Networks

Recurrent Neural Networks (RNNs) stand out as a class of DNNs particularly well-equipped to handle sequential or temporal data. They are constructed to inherently consider the temporal dimension by incorporating feedback connections in their architecture, enabling the passage of information from one time step in the sequence to the next [50].

This inherent memory makes RNNs a fitting choice for tasks such as onset detection in music, where the occurrence of an onset may depend on the context provided by previous time steps in the audio signal. However, while the sequential nature of RNNs provides their strength, it also introduces a critical challenge. Traditional RNNs can struggle with learning long-range dependencies due to the infamous "vanishing gradient" problem, where the contribution of information decays geometrically over time, making it challenging to connect cause and effect over time [51].

This issue has been significantly alleviated through the introduction of more sophisticated RNN architectures, namely Long Short-Term Memory (LSTM) units and Gated Recurrent Units (GRUs). Both these variants introduce gating mechanisms that control the flow of information in the network, enabling the model to selectively remember or forget information, thereby making it capable of learning long-term dependencies [52] [53].

LSTMs and GRUs have shown to significantly improve performance in tasks involving sequential data, including audio processing tasks like onset detection, making them an important tool in the toolbox of modern machine learning practitioners.

2.4.8. Long Short-Term Memory and Bidirectional Long Short-Term Memory Networks

Long Short-Term Memory (LSTM) and *Bidirectional Long Short-Term Memory* (BiLSTM) networks represent advanced variants of RNNs that have demonstrated significant efficacy in sequence modeling tasks [55].

LSTMs were introduced to address the vanishing gradient problem inherent in traditional RNNs, which limits their ability to capture long-term dependencies in sequential data [56]. The key innovation of LSTMs lies in their gating mechanism, which allows the network to selectively remember or forget information over long sequences. This mechanism comprises three gates: the input gate, the forget gate, and the output gate, which work in concert to control the flow of information through the cell state.

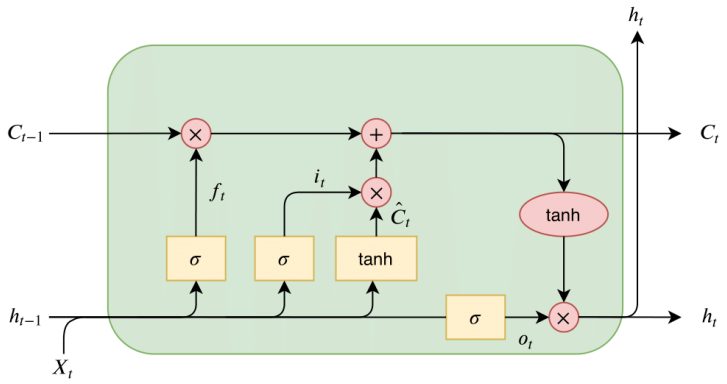

Figure 11: LSTM cell architecture [57].

In Figure 11, the structure of a single LSTM cell is shown, highlighting the flow of information. The *forget gate* (denoted by f_t) determines what portion of the previous cell state C_{t-1} should be discarded, by applying a *sigmoid* function, which produces values between 0 and 1 (where 1 retains all information, and 0 discards it entirely). The *input gate* (denoted by i_t) regulates how much new information from the current input X_t and the previous hidden state h_{t-1} should be added to the cell state. This new information is processed through a *tanh* layer, generating a candidate value $\hat{C}_t$, which is scaled by u_t.

The cell state C_t is updated by combining the retained portion of the previous cell state and the scaled new information. Finally, the *output gate* (denoted by o_t) determines how much of the updated cell state C_t is passed on to the next hidden state h_t. The hidden state h_t, produced by applying a *tanh* activation to C_t and scaling it by o_t, is then propagated forward to the next time step [55].

In the context of onset detection, LSTMs can effectively model the temporal dependencies in audio signals, capturing both short-term and long-term patterns that may be indicative of onsets. This capability is particularly valuable in music with complex rhythmic structures or in polyphonic recordings where onsets may be influenced by preceding musical events [63].

Bidirectional LSTMs (BiLSTMs) further extend this concept by processing the input sequence in both forward and backward directions [64]. This bidirectional approach allows the network to incorporate both past and future context when making predictions at each time step, potentially leading to more accurate onset detection.

The architecture of a BiLSTM network consists of two separate LSTM layers: one that processes the input sequence from left to right (forward layer) and another that processes it from right to left (backward layer). The outputs of these two layers are typically concatenated or summed to produce the final output for each time step [66].

In onset detection applications, BiLSTMs can leverage both preceding and subsequent audio frames to make more informed decisions about the presence of an onset at any given point in time. This bidirectional processing is particularly advantageous in scenarios where the onset characteristics are influenced by both the lead-up to and the aftermath of the event [67].

3. State of the Art

Onset detection remains a vibrant research area within the field of Music Information Retrieval (MIR), reflecting several years of ongoing exploration and discovery. Over time, diverse methodologies and techniques have emerged, each aiming to enhance the accuracy and reliability of onset detection. This task, however, is fraught with complexities, one of the most significant being the use of disparate datasets across different studies, which can make the comparison of results challenging.

The application of different datasets across various studies is largely due to the uniqueness of musical pieces and the inherent subjectivity of human annotation, resulting in a diverse array of datasets each with its own characteristics. Therefore, comparing the performance of different approaches can be difficult as these datasets vary in genre, complexity, recording quality, and even the definition of what constitutes an onset.

Despite these challenges, this chapter provides a comprehensive review of the state-of-the-art methods in onset detection. While making direct comparisons may be difficult due to the use of different datasets, valuable insights into the performance and robustness of each approach can still be gleaned by examining their techniques and the reported results. The chapter aims to detail the technical aspects of these methods, their performance in the context of the datasets used, and how they have contributed to advancing the field of onset detection.

3.1. Early Neural Network Approaches

The application of neural networks to onset detection marked a significant advancement in the field of music information retrieval. These early approaches laid the groundwork for more sophisticated techniques that would follow, leveraging the pattern recognition capabilities of neural networks to develop more accurate and robust onset detection systems. This section explores the pioneering approaches that used different neural network architectures to tackle the challenge of onset detection, tracing their evolution and growing complexity.

Integrate-and-Fire Neural Network

One of the earliest applications of neural networks to onset detection was introduced by Smith in 1996 [58]. This approach combined physiological insights with advanced signal processing techniques to create a novel onset detection system. The method's foundation lies in its use of an auditory front end, which processes the input signal in a manner inspired by the human auditory system.

Smith's system begins by applying a series of bandpass filters to the input audio signal, segmenting the sound into multiple channels, each corresponding to a specific frequency range. This segmentation is based on psychoacoustic principles, mimicking the frequency discrimination capabilities of the human ear. A key innovation in this approach is the introduction of an onset-offset filter, designed to emphasize the onset of tones by detecting rapid increases and decreases in energy characteristic of note onsets.

Following the onset-offset filtering, the system applies logarithmic compression to the signal, enhancing the system's dynamic range and mimicking certain nonlinear processing effects observed in biological auditory systems. The heart of Smith's approach lies in its use of an integrate-and-fire neural network, responsible for integrating onset and offset signals across both frequency bands and time. This network, inspired by the behavior of biological neurons, effectively groups onset signals that occur close together in time and across adjacent frequency channels.

Smith's approach demonstrated impressive results when tested on a variety of sound sources, including both speech and musical sounds. By combining insights from auditory physiology with the pattern recognition capabilities of neural networks, this early work laid an important foundation for future research in onset detection.

Multilayer Perceptron Neural Network

Building upon Smith's groundwork, Marolt et al. introduced a more advanced neural network approach to onset detection in 2002 [59]. This method combined elements of Smith's model with a multilayer perceptron (MLP) neural network, introducing a more sophisticated machine learning component to the onset detection process.

Like its predecessor, Marolt's system begins with an auditory front end that processes the input signal using a bank of auditory filters. However, they expanded on this concept, dividing the signal into 22 overlapping frequency bands, each covering half an octave. This finer-grained frequency analysis allows for more detailed examination of the spectral characteristics of potential onsets.

The system retains the integrate-and-fire neural network component from Smith's model but introduces a multilayer perceptron neural network as a post-processing stage. The MLP takes inputs from the activities of the integrate-and-fire neurons, as well as other parameters such as the amplitudes of individual frequency bands. Using these inputs, the MLP is trained to distinguish between true onsets and false positives, outputting a binary signal indicating whether an onset has occurred.

The addition of the MLP represents a significant advancement over previous approaches. By leveraging the power of supervised learning, the system can be trained on a large dataset of labeled onsets, allowing it to learn complex patterns that distinguish true onsets from other signal fluctuations. Marolt et al. evaluated their system using a combination of synthesized and real piano recordings, demonstrating the ability to accurately identify about 98% of all onsets while maintaining a low 2% rate of false detections.

The success of this approach demonstrated the potential of combining traditional signal processing techniques with more advanced machine learning methods. By using the integrate-and-fire network to generate initial onset candidates and then refining these candidates with an MLP, the system was able to achieve a level of accuracy that would have been difficult to attain with either approach alone.

Single-Net and Multi-Net Approaches

Further expanding the application of neural networks in onset detection, Lacoste and Eck introduced a novel approach in 2005 [60] that used feed-forward neural networks (FNNs) for frame-level onset classification. Their work introduced two related models: Single-Net and Multi-Net, both of which processed spectrogram frames to identify onsets.

The Single-Net approach involves a single feed-forward neural network trained to analyze the spectral content of each frame and determine whether it contains an onset. Building on

this, the Multi-Net approach integrates multiple instances of the Single-Net model, each trained with different hyperparameters. By combining the outputs of these multiple networks, the Multi-Net approach can potentially achieve greater robustness and accuracy in onset detection.

A key innovation in the Multi-Net approach is the incorporation of tempo information. The system includes tempo traces derived from a separate tempo detection algorithm. By considering tempo information alongside the spectral features, the Multi-Net approach can better contextualize potential onsets within the rhythmic structure of the music.

Both the Single-Net and Multi-Net approaches begin with a feature extraction phase that transforms the input audio data into a time-frequency domain representation, typically using methods such as the Short-Time Fourier Transform (STFT) or the Constant-Q transform. Lacoste and Eck emphasized the importance of parameter optimization in their approach, conducting extensive experiments to determine the optimal frame size and input variables for their models.

The neural network architecture employed in both approaches consists of two hidden layers with hyperbolic tangent (*tanh*) activation functions, chosen to enhance classification accuracy while minimizing computational complexity. When evaluated using diverse musical datasets, including those from the MIREX[1] 2005, their results demonstrated the efficacy of their neural network-based approach, achieving high F-measures and showing a good balance between precision and recall in onset detection tasks.

The work of Lacoste and Eck represented a significant step forward in the application of neural networks to onset detection. By demonstrating the effectiveness of both single-network and multi-network approaches, their research opened up new avenues for exploration in the field. The incorporation of tempo information in the Multi-Net approach was particularly innovative, showing how contextual musical information could be integrated into the onset detection process.

[1] MIREX (Music Information Retrieval Evaluation eXchange) is an annual evaluation campaign for Music Information Retrieval (MIR) algorithms. Established in 2005, MIREX provides a framework for the formal evaluation of MIR systems and algorithms, allowing researchers to compare their methods on a common set of tasks and datasets.

3.2. Support Vector Machine Approach

In 2004, Kapanci and Pfeffer [61] introduced a novel approach to onset detection using Support Vector Machines (SVMs), offering a fresh perspective on the problem. Their method reframed onset detection as a determination of whether two frames at a certain temporal distance could stem from the same event. This shift in perspective allowed for a more nuanced analysis of the audio signal, particularly beneficial for capturing soft onsets that may span multiple frames.

The core of their system is an adaptive algorithm that dynamically adjusts the comparison function based on the temporal distance between frames. This adaptability allows the system to effectively distinguish between genuine onsets and other transient changes in the audio signal, addressing the challenge of differentiating onsets from variations within individual note events, particularly in expressive performances.

Central to the algorithm is the comparison graph, a hierarchical structure that organizes regions of the audio signal into subregions and determines the comparisons to be made within each region. This graph is processed bottom-up, starting from leaf nodes representing adjacent frames, allowing efficient identification of potential onsets at different levels of granularity.

The onset recognition function, a key component of the system, considers features such as fundamental frequency, amplitude, and harmonic strengths of each frame pair to determine the likelihood of an onset. Support Vector Machines are then employed to classify these frame pairs as onsets or non-onsets based on the extracted features.

SVMs offer several advantages in this context, including their ability to find optimal separating hyperplanes in high-dimensional feature spaces through the use of kernel functions. This allows for effective classification even when the data is not linearly separable. The SVM classifier was trained on a manually labeled corpus of solo singing segments, chosen for the particular challenges vocal onsets present.

The SVM-based approach demonstrated strengths in handling soft onsets and complex musical textures, areas where earlier energy-based and neural network methods often struggled. Its ability to consider pairwise comparisons between frames allowed for a more context-aware analysis of the audio signal, while the hierarchical nature of the comparison graph enabled adaptation to different temporal scales.

Kapanci and Pfeffer's work represented a significant contribution to onset detection, showcasing the potential of SVMs in this domain. Their approach highlighted the importance of feature engineering in audio analysis and opened new avenues for research in music information retrieval. The insights gained from this method, such as considering pairwise frame comparisons and using hierarchical structures to capture multi-scale information, would go on to influence subsequent work in the field.

3.3. Deep Neural Networks Approaches

As computational power increased and machine learning techniques evolved, researchers began exploring more sophisticated neural network architectures for onset detection. This progression marked a significant leap forward in the field, with each new approach building upon the foundations laid by earlier work while introducing novel ideas and techniques.

Recurrent Neural Networks (RNNs)

In 2012, Böck et al. [62] challenged the *status quo* by demonstrating the power of neural networks for onset detection. At the heart of their system was a RNN, a model architecture inherently suited to handling sequential data like audio signals. RNNs possess an internal memory-like mechanism, enabling them to process current input in the context of previous timesteps – crucial for accurately predicting onsets based on temporal patterns. This represented a significant departure from techniques that made decisions based solely on a single frame of audio data.

A notable achievement of Böck et al. was their focus on real-time onset detection. Designing accurate onset detectors under real-time computational constraints presents specific challenges. Their proposed system, outlined in Figure 12, demonstrates careful consideration for streamlining processing. This achievement highlighted the potential of neural networks not only for improved accuracy but also for practical applications such as live music synchronization or performance analysis.

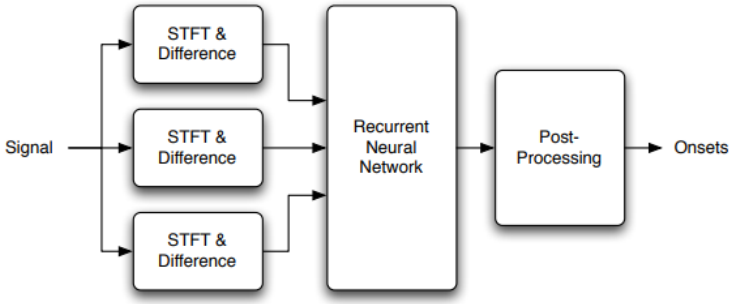

Figure 12: Online real-time onset detection system overview [62].

The model developed by Böck et al. attained an average F1-score of 0.818 with a margin of error of ±25ms. Their contributions occupy a significant position in the trajectory of onset detection research, as their pioneering use of neural networks catalyzed innovation within the Music Information Retrieval (MIR) community.

Convolutional Neural Networks (CNNs)

Building on the success of RNNs, Schlüter and Böck introduced the first application of Convolutional Neural Networks (CNNs) to onset detection in 2014 [63]. CNNs, known for their effectiveness in image recognition tasks, proved to be equally powerful in analyzing spectrograms of audio signals.

To process the audio, they computed three magnitude spectrograms using different window sizes and applied an 80-band Mel filter, which allowed them to transform the audio signal into a more suitable representation for analysis. The magnitudes were logarithmically scaled and then normalized across frequency bands, ensuring consistent and meaningful comparisons. The normalization constants were computed using a hold-out set, enhancing the reliability of the results.

To enable the network to make accurate decisions, they devised a context window of ±70ms (15 frames in total) around the frame being classified. This context window was extracted from all three spectrograms, ensuring that the network could take advantage of broader contextual information. This approach allowed the network to capture relevant temporal dependencies and improve its performance in onset detection tasks.

The architecture of the network, as illustrated in Figure 13, was relatively straightforward, consisting of two convolutional layers each followed by a max-pooling layer and two fully

connected layers. Despite the simplicity of the network, it provided an important advantage: the convolutional layers were able to learn localized features from the input audio signal, while the fully connected layers were responsible for integrating these local features into a global prediction.

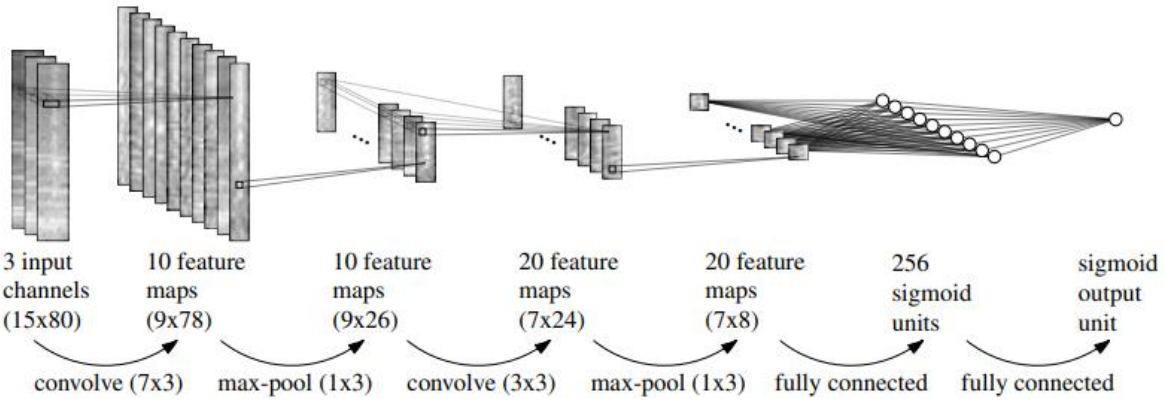

Figure 13: Initial CNN architecture used by Schlüter and Böck [63].

They trained the CNN on a large-scale dataset consisting of manually annotated onsets. This extensive training facilitated the model's understanding of the complexities and subtleties inherent in real-world audio signals. Remarkably, Schlüter and Böck's model surpassed several existing methods of that time.

For evaluating the results, the researchers adopt a precision/recall curve approach. An onset is considered correctly detected if it falls within 25ms of an unmatched target annotation. Any additional detections beyond the matched targets are classified as false positives, and unmatched targets are considered false negatives. The metrics for the optimal F1-score reach up to 0.903 in the Böck Dataset.

Furthermore, Schlüter and Böck's work provided important insights into how the CNN model made its predictions by visualizing the learned filters and activations. These visualizations showed that the CNN had indeed learned to recognize the typical changes in an audio signal that correspond to an onset, validating the effectiveness of the model.

The work of Schlüter and Böck represented a significant step forward in the use of CNNs for onset detection, demonstrating the potential of deep learning techniques in this domain and setting a new standard for subsequent research.

Hybrid and Advanced Architectures

In 2018, Gong and Serra [64] investigated various neural network architectures for musical onset detection, including a Convolutional Recurrent Neural Network (CRNN) approach. This hybrid model combined convolutional layers for local feature extraction with recurrent layers for temporal context awareness.

The CRNN architecture consisted of convolutional layers followed by bidirectional Long Short-Term Memory (LSTM) units. This design allowed the network to process local spectral patterns effectively while maintaining an understanding of the broader temporal context. The authors experimented with different input sequence lengths to optimize the model's performance.

Gong and Serra's work also addressed important issues in deep learning research for music information retrieval, such as reproducibility and model efficiency. They identified a 5-layer CNN architecture that achieved comparable performance to more complex models while using only 28.3% of the trainable parameters. This finding highlighted the potential for simpler, more efficient architectures in onset detection tasks.

In 2023, Tomczak and Hockman [65] introduced even more sophisticated architectures, combining the strengths of different neural network types. They proposed two models: a Convolutional Recurrent Neural Network (CRNN) and a Bidirectional Temporal Convolutional Network (BTCN).

The CRNN architecture leveraged convolutional layers to process local spectral patterns, followed by a bidirectional gated recurrent unit (GRU) layer to maintain awareness of the temporal context. This combination allowed the network to capture both spatial and temporal information effectively. The BTCN, on the other hand, incorporated dilated convolutions that extended both backwards and forwards in time. This design enabled the model to capture long-range dependencies in the audio signal, which is particularly useful for detecting onsets in complex musical contexts.

Both models demonstrated superior performance compared to existing state-of-the-art algorithms, particularly on string instrument recordings. When tested on the QTDS dataset, their BTCN and CRNN models achieved F-measure scores above 0.9. Importantly, the models maintained competitive performance on general music datasets, indicating their versatility across different types of music.

4. Methodology

This chapter outlines the methodologies and processes involved in developing and evaluating the onset detection model. It encompasses data preprocessing, model architecture design, training processes, and the evaluation techniques used.

4.1. Detailed Workflow Process

This section delineates the workflow process employed in the research on musical onset detection. Figure 14 presents a flowchart illustrating the key stages of the methodology. The process follows a cyclical pattern, emphasizing its iterative nature.

The workflow consists of four main stages: Data Collecting, Preprocessing, Modeling, and Evaluation. The process begins with Data Collecting, progresses through Preprocessing and Modeling, and concludes with Evaluation. Notably, the flowchart depicts feedback loops from both Evaluation and Modeling to Preprocessing, indicating that insights from later stages inform refinements in data preparation techniques.

The following subsections provide a detailed examination of each stage, elucidating the methods, techniques, and considerations integral to this onset detection research.

Figure 14: Methodology flowchart.

4.1.1. Data Collection

The Böck dataset was employed for this dissertation, comprising a comprehensive collection of musical recordings annotated for onset detection. This dataset includes approximately 100 minutes of diverse music genres, featuring 26,000 labeled onset events. The annotations were manually curated, ensuring high accuracy and consistency, which is critical for training robust machine learning models. This dataset was selected for its balance of polyphonic and monophonic music, providing a well-rounded foundation for evaluating the performance of onset detection algorithms.

This dataset was used to build upon the work of Jan Schlüter and Sebastian Böck, whose contributions to musical onset detection using convolutional neural networks have set a

benchmark in the field. The diversity and detailed annotations of the Böck dataset make it an ideal choice for training and evaluating advanced models for musical onset detection, ensuring that the models developed are both comprehensive and versatile.

4.1.2. Preprocessing

In the preprocessing phase, Mel spectrograms of different sizes were created using various parameters to convert the raw audio signals into a visual representation that emphasizes perceptually relevant frequency components. This transformation facilitates more effective feature extraction for subsequent modeling steps, ensuring the retention of essential information from the audio data.

Additionally, various spectral representations such as constant-Q transform (CQT) and chromagrams were experimented with, although they did not yield satisfactory results compared to Mel spectrograms.

Data labeling involved splitting the spectrograms into frames and annotating them based on the presence of onsets. This was done by aligning the frames with the annotated onset times from the dataset, marking frames as either containing an onset or not. Both training and validation datasets were created, with frames annotated accordingly to ensure that the model could learn to distinguish between onset and non-onset frames accurately. This labeling process ensured that the training data provided a robust foundation for the model to learn temporal patterns associated with musical onsets effectively.

The preprocessing stage is crucial for transforming the raw audio data into a format suitable for input into neural network models, ensuring that the most relevant features are extracted and highlighted for effective learning.

4.1.3. Modeling

The modeling phase involved experimentation with various neural network architectures, including Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, double-layer LSTMs (2xLSTMs), and Bidirectional LSTMs (BiLSTMs), to optimize musical onset detection. The initial approach replicated the CNN model proposed by Schlüter and Böck [63], with subsequent iterations introducing incremental changes to enhance accuracy and robustness.

These architectures were selected for their proven effectiveness in handling sequential data and capturing temporal dependencies. The process encompassed extensive testing of different model configurations, optimizers, and hyperparameters. Techniques such as mini-batch gradient descent, learning rate scheduling, and momentum were employed to improve training efficiency.

To mitigate overfitting, strategies like dropout and early stopping were implemented. This iterative modeling and training process aimed to achieve an optimal balance between precision, recall, and computational efficiency while ensuring good generalization to unseen data. The comprehensive approach sought to develop a versatile onset detection system applicable to diverse real-world scenarios.

4.4.4. Evaluation

The evaluation phase employed an 8-fold cross-validation approach to assess the model's performance. Custom evaluation metrics were developed, incorporating a margin of error, peak picking, and smoothing as post-processing steps to refine the onset detection results. A tolerance window of ±25 milliseconds was established to accommodate minor temporal discrepancies.

Key performance indicators such as true positives, false positives, false negatives, precision, recall, and F1-score were calculated to provide a comprehensive assessment of the model's accuracy. This detailed evaluation framework ensured that the model's performance was thoroughly vetted, highlighting areas of strength and opportunities for further improvement.

4.2. Hardware, Software, and Libraries

The development and implementation of the onset detection models for this dissertation leveraged a suite of advanced technologies and tools. Python 3.9 was the primary programming language used, providing a robust and versatile environment for data processing, model development, and evaluation. Keras[2], a high-level neural networks API, facilitated the construction and training of the deep learning models, enabling rapid prototyping and experimentation with various neural network architectures. For audio processing, madmom[3] was employed to handle the transformation of audio signals into

[2] Keras: Chollet, F., & others. (2015). Keras. GitHub. https://github.com/keras-team/keras
[3] madmom: [75] https://github.com/CPJKU/madmom

spectral representations such as Mel spectrograms, which are crucial for feature extraction. Additionally, Weights & Biases[4] was used for experiment tracking, ensuring systematic documentation of the modeling process.

The computational demands of training deep learning models for onset detection were met with a high-performance hardware setup. The system was powered by an Intel® Core™ i5-13600K processor, providing multi-core processing capabilities essential for handling extensive data preprocessing and training workloads. A NVIDIA® GeForce RTX® 4090 graphics card was employed to accelerate the training of the neural networks. The system also included 32GB of DDR5 RAM, ensuring data handling and manipulation, particularly for the large dataset and complex model architectures. This hardware configuration significantly reduced training times.

4.3. Challenges

The development and implementation of an effective onset detection system using deep learning techniques presented several significant challenges. This section outlines the key obstacles encountered during the research process.

4.3.1. Data Complexity and Variability

One of the primary challenges was dealing with the inherent complexity and variability of musical data. Music spans a wide range of genres, instruments, and recording qualities, each presenting unique characteristics for onset detection:

- Diverse Onset Types: Different instruments and musical styles produce vastly different types of onsets. For example, the sharp attack of a drum hit contrasts dramatically with the gradual onset of a bowed string instrument. Creating a model capable of accurately detecting this full spectrum of onset types proved challenging;
- Polyphonic Complexity: In polyphonic music, where multiple instruments or notes sound simultaneously, identifying individual onsets becomes significantly more difficult. The overlapping of sounds can mask or distort onset signals, making them harder to detect;

[4] Weights & Biases: Biewald, L. (2020). Experiment Tracking with Weights and Biases. Software available from wandb.com. https://www.wandb.com/

- Recording Quality Variations: The dataset included recordings of varying quality, from professional studio recordings to amateur live performances. Inconsistencies in audio quality, background noise levels, and recording techniques added an extra layer of complexity to the onset detection task.

4.3.2. Class Imbalance

The nature of onset detection leads to a severe class imbalance in the training data:

- Sparse Onsets: In most musical pieces, onsets occur relatively infrequently compared to the total duration of the audio. This results in a dataset where non-onset frames vastly outnumber onset frames;
- Model Bias: This imbalance can lead to models that are biased towards predicting non-onsets, potentially missing critical onset events and reducing the overall effectiveness of the detection system.

4.3.3. Temporal Context and Resolution

Determining the optimal approach to capture temporal context presented a significant challenge:

- Context Window Size: Choosing the appropriate size for the input context window was crucial. Too small a window might miss important contextual information, while too large a window could introduce irrelevant data and increase computational complexity;
- Resolution Trade-offs: Higher temporal resolution (shorter frame sizes) can provide more precise onset localization but may introduce noise and increase computational demands. Lower resolution, while more efficient, risks missing rapid successions of onsets.

4.3.4. Model Architecture Optimization

Designing and optimizing the neural network architecture posed several challenges:

- Complexity *vs.* Performance: Balancing model complexity with performance was an ongoing issue. More complex models often yielded better results but at the cost of increased computational requirements and potential overfitting;

- Architecture Selection: With numerous potential architectures (CNNs, RNNs, LSTMs, etc.) and hybrid approaches available, determining the most effective structure for onset detection required extensive experimentation and evaluation.

4.3.5. Computational Resources and Efficiency

The computational demands of training and evaluating deep learning models for onset detection were significant:

- Training Time: Complex models required substantial training time, limiting the number of experiments and iterations that could be performed within the research or hardware availability timeframe;
- GPU Memory Constraints: The size of the models and the dimensionality of the input data often pushed the limits of available GPU memory, necessitating compromises in batch sizes or model complexity.

4.3.6. Replication of Previous Work

A significant challenge encountered during this research was the difficulty in exactly replicating the results reported by Böck and Schlüter in their seminal 2014 paper on onset detection using convolutional neural networks [63]:

- Lack of Precise Implementation Details: While the paper provided a general overview of the model architecture and training process, it lacked some of the fine-grained implementation details necessary for exact replication. This included specifics about data preprocessing, weight initialization, and certain hyperparameters.
- Dataset Discrepancies: Although the same Böck dataset was used, subtle differences in how the data was preprocessed or augmented could have led to discrepancies in results. For instance, the specific audio files used as a hold-out set during the normalization process were not provided.
- Undocumented Optimizations: It is possible that the original implementation included optimizations or tweaks that were not fully documented in the paper, making exact replication difficult.

These challenges collectively represented significant hurdles in the development of an effective onset detection system. Addressing them required a combination of careful

experimental design, and iterative refinement of both the data preprocessing pipeline and the model architecture.

5. Data Preprocessing

The data preprocessing phase is a critical component in the development of an effective onset detection system using neural networks. This stage involves transforming raw audio signals into a format that is both suitable for neural network input and rich in relevant features for onset detection. The preprocessing steps described in this chapter were meticulously designed to capture the complex characteristics of audio signals, ensuring that the model was trained on informative and well-structured data.

5.1. Böck Dataset

The Böck Dataset[5], proposed by Böck et al. [63], is the dataset used in this onset detection study. It consists of 321 audio files, totaling 102 minutes of music with more than 25,000 annotated onsets. The dataset encompasses a diverse range of musical content, including solo and mixed music excerpts categorized into complex mixtures (193 files), pitched percussive (60 files), non-pitched percussive (17 files), wind instruments (25 files), bowed strings (23 files), and vocal recordings (3 files).

This composition ensures a comprehensive evaluation of onset detection algorithms across various musical contexts and instrument types. An additional 84 audio files without annotations are included, typically used for normalization statistics or as a hold-out set. The Böck Set's diversity and comprehensive nature provide a robust foundation for training and evaluating CNN-based onset detection models, allowing for the assessment of algorithm performance across a wide spectrum of musical inputs.

5.2. Pre-processing Methodology

The onset detection system employs an advanced pre-processing pipeline to transform raw audio signals into a suitable format for neural network analysis. This pipeline, largely inspired by the work of Böck et al. [63], consists of several key stages, each designed to extract and emphasize relevant features from the audio data. Figure 15 presents a high-level overview of the pre-processing steps.

[5] Can be found at https://github.com/CPJKU/onset_db

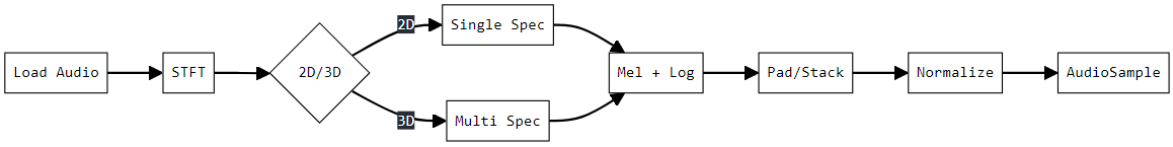

Figure 15: Flowchart of the audio pre-processing pipeline for onset detection.

As illustrated in Figure 15, the process begins with loading the audio signal and proceeds through various transformations, including Short-Time Fourier Transform (STFT), spectrogram computation, and application of Mel filter banks. The pipeline accommodates both 2D and 3D spectrogram representations, allowing for flexibility in the input format. The final stages involve normalization and creation of an AudioSample object, which serves as the input to the neural network model.

5.3. Spectral Representations

This section provides a detailed explanation of each step in the pre-processing pipeline, including the rationale behind specific choices and the mathematical foundations of the employed techniques.

5.3.1. Fast Fourier Transform (FFT) and Mel Spectrograms

One of the fundamental techniques employed in the preprocessing pipeline is the conversion of time-domain audio signals into frequency-domain representations. This transformation is achieved through the application of the Fast Fourier Transform (FFT), which decomposes the audio signal into its constituent frequency components.

The implementation uses both single and stacked Mel spectrograms to provide comprehensive frequency representations. Mel spectrograms are particularly useful in audio processing tasks as they approximate the human auditory system's perception of sound, emphasizing lower frequencies where human hearing is more sensitive.

Single Mel Spectrograms

For the single Mel spectrogram approaches, the following parameters were employed:

- Number of Mel bands: 80 or 120;
- Frequency range: 27.5 Hz to 16 kHz;
- FFT window size: 2048 samples;
- Hop length: 441 samples (10 ms).

This configuration provides a balance between frequency resolution and temporal precision. The use of 120 Mel bands allows for detailed frequency analysis while maintaining computational efficiency. The frequency range of 27.5 Hz to 16 kHz covers the most relevant part of the audible spectrum for music and speech signals.

Stacked Mel Spectrograms

To capture a broader range of temporal and spectral details, a stacked spectrogram approach as used in [63] was also implemented. This method involves creating multiple spectrograms with varying window sizes and combining them into a single, multi-channel representation. The window sizes used were:

- Number of Mel bands: 80;
- 1024 samples (approximately 23 ms);
- 2048 samples (approximately 46 ms);
- 4096 samples (approximately 93 ms).

By using multiple window sizes, it becomes possible to capture both fine-grained temporal details (with smaller windows) and more stable frequency information (with larger windows). This multi-resolution approach allows the neural network to learn from features at different time scales, potentially improving its ability to detect onsets in various musical contexts.

Figure 16 shows the waveform and corresponding stacked spectrogram of one of the samples of the dataset.

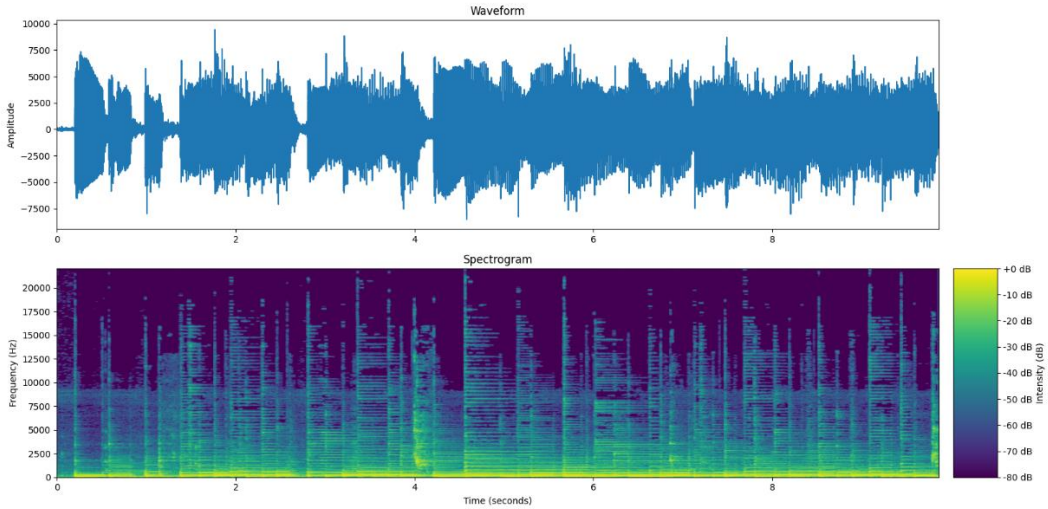

Figure 16: A waveform and a 3D spectogram of one of the samples.

5.3.2. Logarithmic Scaling

After the computation of Mel spectrogram(s), logarithmic scaling was applied to the magnitude values. This step is crucial as it better aligns with human perception of loudness, which follows a logarithmic rather than a linear scale. Logarithmic scaling also helps to compress the dynamic range of the spectrogram, making softer sounds more prominent and preventing louder sounds from dominating the representation.

The logarithmic scaling was implemented using the following formula:

$$S_{log} = 10 * log_{10}(S + \varepsilon),$$

where S is the original spectrogram, ε is a small constant to avoid taking the logarithm of zero, and S_{log} is the resulting log-scaled spectrogram.

5.3.3. Spectrogram Normalization

To ensure consistent scaling across different audio samples and to aid in the convergence of neural network training, a robust normalization technique was implemented in the preprocessing of audio spectrograms for the CNN model. This approach employs a method that leverages a holdout set to compute global normalization parameters, adapting to both 2D and 3D spectrogram configurations.

The process begins with the designation of a subset of audio files without annotations or fold assignments as a holdout set. This set serves as the basis for calculating the mean and standard deviation of the spectrograms, which are subsequently used as global normalization parameters. The computation of these statistics differs based on the spectrogram type:

For 2D spectrograms, the mean and standard deviation are calculated across the time dimension for each frequency band. This results in a single set of normalization parameters that capture the average spectral characteristics of the audio data.

In the case of 3D spectrograms, which consist of three different spectral representations computed with varying frame sizes (2048, 1024, and 4096 samples), the statistics are computed independently for each representation. This results in three sets of normalization parameters, preserving the unique characteristics of each time-frequency resolution.

The normalization formula applied to both 2D, and 3D spectrograms is:

$$S_{norm} = \frac{S - \mu}{\sigma + \varepsilon},$$

where S represents the original spectrogram, S_{norm} denotes the normalized spectrogram, μ is the mean, σ is the standard deviation, and ε is a small constant (10^{-7}) added for numerical stability.

This normalization technique offers several key benefits:

- It centers the data around zero and scales it to unit variance, facilitating the convergence of gradient-based optimization algorithms used in neural network training;
- It equalizes the scale of different frequency bands, preventing certain frequencies from dominating due to their naturally higher amplitudes;
- It enhances the model's robustness to variations in overall volume between different audio samples;
- By using a holdout set for normalization parameters, it prevents data leakage and ensures that the test set remains unseen, maintaining the integrity of the evaluation process.

The adoption of this normalization strategy, adaptable to both 2D and 3D spectrogram configurations, contributes to the preparation of high-quality input data for the CNN model. This approach potentially improves the model's performance and generalization capabilities in onset detection tasks across various spectrogram representations.

5.3.4. Alternative Spectral Representations

In addition to the Mel spectrograms used as the primary input representation, experiments were conducted with alternative spectral representations to explore their efficacy in onset detection. Specifically, the Constant-Q Transform (CQT) and chromagrams were investigated as potential alternatives or complementary features to the Mel spectrograms.

The Constant-Q Transform was explored due to its logarithmic frequency scaling, which aligns closely with human auditory perception and musical pitch organization. This property was hypothesized to potentially capture onset-related features more effectively, especially

for pitched instruments. Implementation of the CQT followed standard practices, with a range of frequency bins and time resolutions tested to optimize performance.

Chromagrams, which represent the energy distribution across the twelve pitch classes of Western music, were also examined. The motivation behind this approach was to capitalize on the harmonic structure of music, potentially allowing the model to detect onsets through changes in harmonic content, even in cases where energy-based features might be less pronounced.

These alternative representations were processed and fed into the neural network architectures in a manner similar to the Mel spectrograms, with appropriate adjustments made to accommodate their specific dimensions and characteristics.

Despite the theoretical advantages of these representations, the experimental results did not demonstrate significant improvements over the Mel spectrogram-based approach. The CQT-based models showed comparable but slightly lower performance metrics across various architectural configurations. Chromagram-based models, while capturing some onset events effectively, generally underperformed compared to both Mel spectrogram and CQT-based approaches, particularly in detecting percussive or non-pitched onsets.

The underperformance of these alternative representations may be attributed to several factors:

1. Information loss: The dimensionality reduction inherent in chromagrams may have resulted in the loss of critical spectral information necessary for precise onset detection;
2. Temporal resolution: The time-frequency trade-off in the CQT, while beneficial for pitch-related tasks, may have compromised the temporal precision required for accurate onset detection;
3. Model optimization: The neural network architectures and hyperparameters were initially optimized for Mel spectrogram inputs. Despite efforts to adapt these for CQT and chromagram inputs, it is possible that the models were not fully optimized for these alternative representations.

While these experiments did not yield superior results, they provided valuable insights into the robustness of Mel spectrograms for onset detection tasks. The findings suggest that the combination of frequency resolution and temporal precision offered by Mel spectrograms remains highly effective for this specific application in music information retrieval.

It is worth noting that future work could explore more sophisticated fusion techniques, combining multiple spectral representations to potentially leverage the strengths of each. Additionally, further investigation into architecture modifications specifically tailored to CQT or chromagram inputs may yield improved results in subsequent studies.

5.4. Temporal Context and Frame Slicing

To provide temporal context for onset detection, a frame-based approach was implemented where the network processes 15 consecutive frames of the spectrogram at a time. This method allows the neural network to consider the temporal evolution of the signal around potential onset points. Unlike the previous implementation, where spectrograms were pre-sliced during preprocessing and fed separately into the CNN, the new approach dynamically feeds sequences of frames into the model during training, without the need for pre-slicing.

5.4.1. Frame Parameters

The following parameters were used for frame slicing:

- Frame width: 15 frames (150 ms);
- Frame jump (step size): 1 frame (10 ms).

This configuration results in a high degree of temporal overlap between adjacent frames, providing fine-grained temporal resolution for onset detection. The 150 ms frame width was chosen to capture sufficient context around potential onsets while maintaining a manageable input size for the neural network as see in other works.

5.4.2. Data Augmentation Through Frame Shifting

To increase the robustness of the model and artificially expand the dataset, a form of data augmentation through frame shifting was implemented. Instead of slicing spectrograms during preprocessing, the training data is dynamically created by sliding the frame window over the spectrogram with small temporal offsets. This technique introduces slight temporal variations in the input, making the model more robust to small shifts in timing, which is crucial for accurate onset detection across various musical styles and recording conditions.

5.5. Labeling Strategies

The labeling process is a critical component of supervised learning for onset detection, as it determines how the model interprets and learns from the data. In this context, labeling

involves marking each frame within an audio signal as either containing an onset or not, creating a binary classification task.

5.5.1. Fuzzy Training Samples

To address the inherent imprecision in onset annotations and allow for flexibility in the exact timing of detected onsets, a fuzzy labeling strategy was implemented. This approach offers two labeling types: 'exact' and 'fuzzy'.

In the 'exact' labeling mode, each frame corresponding to an annotated onset time is assigned a label of 1, while all other frames are labeled 0. This creates a binary classification task where each frame is either an onset or not.

The 'fuzzy' labeling mode introduces a more nuanced approach:

- The frame corresponding to the annotated onset time is labeled 1;
- The immediately preceding and following frames are also labeled 1 but are assigned a reduced weight of 0.25 in the loss function;
- All other frames are labeled 0 and maintain a full weight of 1.

This fuzzy labeling strategy creates a small window of positive labels around each onset, acknowledging that the exact onset time may fall between frames. The reduced weight for adjacent frames ensures that the model is not overly penalized for slight temporal shifts in onset detection.

Other methods, such as 'super_fuzzy' and 'gaussian', were also explored. The 'super_fuzzy' approach extended the window of positive labels to include two frames before and after the onset, providing greater tolerance for onset timing inaccuracies. The 'gaussian' method, also tried in [63], applied a weighted Gaussian distribution around the onset, giving progressively lower weights to neighboring frames. However, these methods did not yield satisfactory results and were not pursued further in the model's development.

5.5.2. Label Assignment

The process of label assignment plays a crucial role in transforming raw audio data into a format suitable for neural network training. For onset detection tasks, this step is particularly vital as it bridges the gap between the continuous nature of audio signals and the discrete, frame-wise predictions made by neural networks. The labeling strategy employed can

significantly impact the model's performance, affecting its ability to accurately detect onsets and generalize to various musical contexts.

The label assignment process for onset detection involves a delicate balance between precision and flexibility. On one hand, precise labeling ensures that the model learns to identify onsets at exact time points. On the other hand, a more flexible approach can account for the inherent ambiguity in onset perception and annotation, potentially leading to more robust models. To address this trade-off, two primary labeling modes are often employed: 'exact' and 'fuzzy' labeling.

The 'exact' labeling mode assigns a binary label to each frame, marking it as either containing an onset (1) or not (0). This approach provides clear, unambiguous targets for the model to learn. However, it can be sensitive to small temporal shifts in onset annotations, which may occur due to human error in manual labeling or slight variations in onset perception.

In contrast, the 'fuzzy' labeling mode introduces a degree of tolerance around each onset. This approach acknowledges that the precise moment of an onset can be subjective and that slight temporal variations should not necessarily be penalized. By assigning positive labels to frames immediately adjacent to the annotated onset, with reduced weights, the fuzzy labeling creates a small window of leniency around each onset event.

The labeling process, regardless of the chosen mode, follows a systematic procedure to convert temporal onset annotations into frame-wise labels suitable for neural network training. This procedure involves several key steps:

a. Convert annotated onset times to frame indices by multiplying the time (in seconds) by the frame rate (100 frames per second in this implementation);
b. Initialize a zero vector for onset labels and a unit vector for weights, both with length equal to the number of frames in the audio file;
c. For each onset frame:
    - In 'exact' mode, set the corresponding label to 1;
    - In 'fuzzy' mode:
        1) Set the label to 1 for the onset frame and its immediate neighbors;
        2) Set the weight to 0.25 for the neighboring frames.

This labeling scheme effectively transforms the onset detection problem into a frame-wise classification task, which is well-suited for neural network training. The fuzzy labeling

option provides a more forgiving learning target that can potentially improve the model's generalization and robustness to slight temporal variations in onset occurrences.

The choice between 'exact' and 'fuzzy' labeling allows for experimentation with different levels of temporal precision in the training data, potentially leading to models with different characteristics in terms of temporal accuracy and robustness. This flexibility in labeling strategies enables researchers to tailor the learning process to the specific requirements of their onset detection task, balancing the need for precise onset localization with the desire for models that can generalize well across various musical contexts and recording conditions.

6. Modeling and Results

This chapter presents a comprehensive analysis of the modeling process, and the results obtained from the musical onset detection system. It delves into the evaluation metrics used to assess the model's performance, discusses the various experiments conducted, and analyzes the outcomes. The chapter aims to provide a clear understanding of how well the model performs in detecting musical onsets and how it compares to existing state-of-the-art approaches.

6.1. Time-Aware Neural Networks for Onset Detection

Music is inherently a temporal phenomenon, with its structure and meaning deeply rooted in the progression and interaction of sounds over time. As observed in the background and state-of-the-art sections, the temporal nature of music plays a crucial role in onset detection. The literature reveals several approaches that have sought to exploit the temporal aspects of music for onset detection. Eyben et al. [66] introduced the use of bidirectional Long Short-Term Memory (LSTM) networks for onset detection, allowing the model to consider both past and future context. Böck et al. [62] developed an online real-time onset detection system using recurrent neural networks, demonstrating the importance of sequential processing in onset detection. Schlüter and Böck [63] employed convolutional neural networks (CNNs) with multiple input channels representing different time scales, implicitly capturing some temporal information.

Of particular note is the work by Serra et al. [64], which sought to leverage temporal information by using bidirectional LSTMs after the convolutional layers. In their approach, each spectrogram was treated as a single point in the sequence that the LSTM would process, allowing for a broader temporal context to be considered. However, an important observation is that when a convolutional network receives a spectrogram, each column of pixels represents a period in time, and the final volume produced by that part of the network also contains temporal information.

This research aims to explore how onset detection can take advantage of this characteristic. The proposed network architecture takes as input a 3D volume representing a 150ms context window (15 frames × 80 frequency bins × 3 channels) centered around each 10ms frame. This input first passes through convolutional layers, which act as feature extractors,

transforming it into a new volume that preserves the temporal sequence while highlighting spectral-temporal patterns. For example, the output may take the shape of (7, 8, 20), where 7 represents the time steps retained after convolution and pooling.

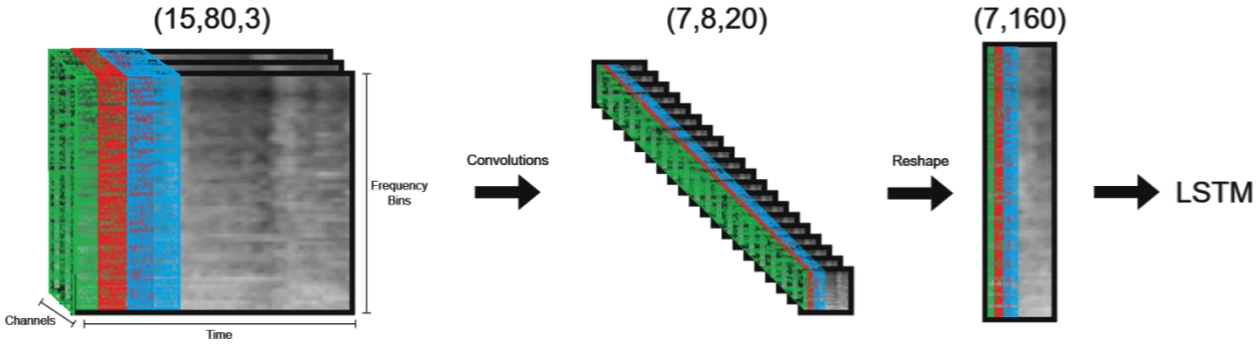

Figure 17: Sample transformation across the network.

As illustrated in Figure 17, the output is then reshaped into a 2D matrix, typically of shape (7, 160), where the first dimension represents feature vectors, each one corresponding to a single time step within the 150ms context. This reshaping flattens the frequency and channel dimensions while retaining the temporal structure, preparing the data for sequential processing by the LSTM. The LSTM processes this sequence of feature vectors, capturing how features evolve over the 150ms window and modeling temporal dependencies that are important for onset detection. By focusing on this short timeframe, the model can detect subtle changes that may indicate the start of a musical event.

This frame-by-frame approach provides several practical advantages. It enables real-time processing, as predictions can be made as each new frame becomes available. It also maintains a consistent context for each frame, as all predictions are based on the same window of surrounding audio. Additionally, this structure allows the LSTM to focus on temporal patterns specific to musical onsets within the brief time span. By leveraging the temporal information preserved through the convolutional layers and explicitly modeling it with the LSTM, this approach aims to more effectively capture the time-dependent nature of musical onsets, potentially leading to improved detection accuracy.

6.2. Evaluation Methodology

The evaluation methodology for the onset detection model employs a systematic approach to assess its performance across various musical contexts. This methodology combines

signal processing techniques with statistical analysis to provide a comprehensive assessment of the model's capabilities.

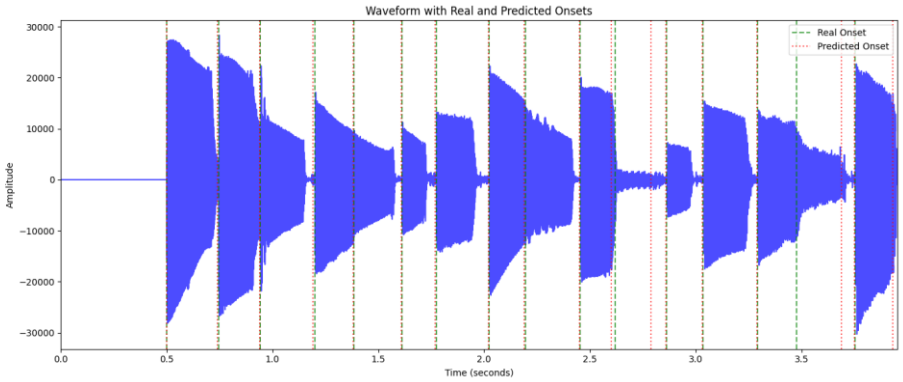

Figure 18: Waveform with real and predicted onsets.

As illustrated in Figure 18, onset detection involves identifying specific points in a waveform where new musical events begin. The figure shows both the actual (ground truth) onsets and the onsets predicted by the model, providing a visual representation of the detection process and the challenges involved in accurate onset identification.

Prior to evaluation, the onset detection function undergoes a smoothing operation. This is achieved by convolving the function with a Hamming window of 5 frames. The smoothing process helps to reduce noise and minor fluctuations in the onset detection function, leading to more robust peak detection.

Following smoothing, a peak-picking algorithm is applied to identify onset candidates. Local maxima in the smoothed onset detection function that exceed a specified threshold are considered as potential onsets. This threshold is a critical parameter that balances the trade-off between detecting true onsets and avoiding false positives.

The evaluation process then centers on comparing these detected onsets against ground truth annotations. A tolerance window of ±25 milliseconds is used to determine whether a detected onset matches an annotated onset. This window accounts for temporal discrepancies in annotations and onset perception. A combination window of 30 milliseconds is implemented to merge closely spaced annotated onsets, addressing the issue of rapid successive events in the ground truth.

The evaluation system categorizes each detected onset into one of three classes:

1. True Positive (TP): A detected onset that falls within the tolerance window of an annotated onset;
2. False Positive (FP): A detected onset that does not correspond to any annotated onset within the tolerance window;
3. False Negative (FN): An annotated onset that has no corresponding detected onset within its tolerance window;

These categorizations form the basis for computing the primary evaluation metrics: Precision, Recall, and F1-score. Precision quantifies the proportion of detected onsets that are correct, while Recall measures the proportion of actual onsets that are successfully detected. F1-score provides a balanced assessment of the model's performance by combining Precision and Recall.

The evaluation process includes a parameter optimization phase, which involves exploring key parameters such as the detection threshold. The F-measure serves as the primary metric for selecting the optimal parameter configuration.

To ensure result robustness, an 8-fold cross-validation strategy is implemented. This approach involves dividing the dataset into eight subsets, using seven for training and one for validation in each iteration. This method provides a more reliable estimate of the model's performance on unseen data and helps identify potential overfitting.

The evaluation methodology aligns with standard practices in the field of Music Information Retrieval for onset detection tasks. This approach aims to provide an accurate assessment of the onset detection model's capabilities. The methodology not only offers an understanding of the model's performance but also allows for comparisons with other onset detection systems reported in the state-of-the-art chapter.

6.3. Optimization and Tuning

This study involved extensive experimentation with various hyperparameters to optimize the performance of the onset detection system. Numerous setups were tested, exploring different combinations of optimizers, learning rates, momentum values, dropout rates, and architectural parameters. Among the optimizers evaluated, including Adam, SGD, and Adadelta, Stochastic Gradient Descent (SGD) consistently demonstrated superior performance for this specific task. This aligns with recent findings in deep learning research

suggesting that SGD often leads to solutions that generalize better, possibly due to its ability to find flatter minima in the loss landscape.

The learning rate proved to be a critical hyperparameter. After thorough testing of various values, a learning rate of 0.05 was found to be optimal, providing a good balance between convergence speed and stability. A dynamic approach to momentum adjustment was implemented, involving a gradual increase from 0.45 to 0.9 over the course of training. Specifically, the momentum remained at 0.45 for the first 10 epochs, then linearly increased to reach 0.9 by epoch 20. This strategy aimed to leverage the benefits of lower momentum in the early stages of training for better exploration of the parameter space, transitioning to higher momentum for faster convergence in later stages.

Dropout rates were also carefully tuned to prevent overfitting. After testing a range of values, a dropout rate of 0.5 was found to be optimal. This value provided the best balance between model capacity and generalization ability, effectively reducing overfitting without overly limiting the model's learning capacity. In addition to these general hyperparameters, extensive exploration was conducted to determine the optimal sizes of feature maps for the CNN layers and the number of units for various LSTM configurations. For the CNN, various combinations of feature map sizes were tested, with 32 filters in the first layer and 64 filters in the second layer emerging as the most effective configuration. This offered a good balance between model capacity and computational efficiency.

For the LSTM layers, different unit configurations were explored. In the single LSTM setup, 64 units were found to be optimal. For the two-layer LSTM (2LSTM) configuration, 64 units in the first layer and 32 units in the second layer yielded the best results. In the case of bidirectional LSTM (BiLSTM), 64 units in each direction provided the highest performance. These optimal configurations were carried forward in subsequent experiments, ensuring that each architectural variation was evaluated using its most effective setup.

6.4. Model Development and Experiments

This section details the process of developing a convolutional neural network (CNN) model for musical onset detection and the experiments conducted to refine its performance. The work builds upon the groundbreaking approach of Böck and Schlüter [63], iteratively refining and extending their methodology.

6.4.1. Initial Experiments

The initial experiments focused on recreating the CNN architecture described by Böck and Schlüter [63]. Their model achieved state-of-the-art performance at the time, outperforming previous approaches including recurrent neural networks.

Base Architecture

The base architecture consisted of the following layers:

- Input: 3-channel spectrogram excerpts (15 frames x 80 Mel bands);
- Convolutional layer 1: 10 feature maps with 7x3 filters;
- Max pooling layer 1: 1x3 pooling;
- Convolutional layer 2: 20 feature maps with 3x3 filters;
- Max pooling layer 2: 1x3 pooling;
- Dropout layer 1;
- Fully connected layer: 256 units;
- Dropout layer 2;
- Output layer: 1 unit with *sigmoid* activation.

Training Methodology

The model was trained using stochastic gradient descent with dynamic momentum, optimizing binary cross-entropy loss. Two training strategies were initially tested:

1. Exact Labelling Strategy:
    - Learning rate: 0.05;
    - Initial momentum: 0.45, linearly increased to 0.9 between epochs 10 and 20;
    - No dropout layers;
    - Fixed number of 100 epochs.
2. Fuzzy Labelling Strategy:
    - Initial learning rate: 1.0, multiplied by 0.995 after each epoch;
    - Dropout layers included;
    - Fixed number of 300 epochs.

These hyperparameters were chosen to closely match those reported in the original paper.

Table 1: Results table: Original vs replication vs early stopping

Model	F1-score (Exact)	F1-score (Fuzzy)
Original Böck CNN	0.885	0.903
CNN Replica (R)	0.859	0.900
CNN Replica with Early Stopping (ES)	-	0.898

As shown in Table 1, the replicated model achieved F1-scores of 0.859 and 0.900 for exact and fuzzy labelling strategies, respectively. These results are slightly lower than the original model's reported F1-scores of 0.885 and 0.903. Notably, other researchers have encountered similar challenges in reproducing the exact results reported by Böck and Schlüter.

For instance, Gong and Serra [64] attempted to replicate the original CNN model and reported an F1-score of 0.867, which falls between the replication results presented in this study and the original paper's figures.

To optimize training efficiency and potentially improve performance, an additional experiment was conducted using early stopping and a lower initial learning rate. This approach used an initial and fixed learning rate of 0.05 instead of the 1.0 with the custom scheduler and implemented early stopping. This method achieved an F1-score of 0.898, slightly underperforming the initial fuzzy labelling implementation while taking half of the time to train.

The improved performance and reduced training time of this approach are significant. Training for 300 epochs per fold, as in the original implementation, would be time-consuming and potentially unnecessary. The early stopping mechanism allows the model to terminate training when no further improvement is observed, saving computational resources.

Based on these results, and considering the constraints of limited time and computational resources available for this study, all subsequent experiments exclusively use the fuzzy labelling approach with early stopping and a learning rate of 0.05. This decision aims to balance improved performance and faster training times, while working within the practical limitations of the research environment. The approach offers a compromise between model robustness and efficient use of available resources, enabling a more comprehensive exploration of onset detection techniques within the scope of this study.

6.4.2. Model Refinement

Following the establishment of the baseline, a series of refinements to the model architecture and training process were undertaken. These refinements focused on incorporating elements from more recent deep learning research and exploring variations on the original architecture.

Experiment 1: Batch Normalization

The first refinement involved the addition of batch normalization (BN) layers after each convolutional layer. Batch normalization, introduced by Ioffe and Szegedy [67], helps to stabilize the distribution of layer inputs during training. This technique often leads to faster convergence and better generalization by reducing internal covariate shift.

The implementation of batch normalization involved the following changes:

1. A Batch Normalization layer was added after each convolutional layer;
2. The order of operations was adjusted to: Convolution -> Batch Normalization -> Activation -> Max Pooling.

Table 2: Results table: Added batch normalization

Model	F1-score	Description
Böck CNN (Original)	0.903	Reported performance in Böck and Schlüter [63]
CNN (R) ES	0.898	Replication of Böck and Schlüter [63]
CNN (R) ES BN	0.900	Added Batch Normalization

As seen in Table 2, the addition of batch normalization improved the model's F1-score from 0.898 to 0.900. While this improvement is modest, it demonstrates the potential benefits of batch normalization in stabilizing the learning process. Moreover, the training time was reduced by half. This reduction in training time underscores the efficiency gains achieved through batch normalization, making it a valuable enhancement in the training process.

As before, considering the constraints of limited time and computational resources, batch normalization was incorporated in subsequent experiments. This decision aimed to leverage the improved efficiency and potentially more stable learning process, allowing for a more comprehensive exploration of onset detection techniques within the practical limitations of the research environment. The use of batch normalization in subsequent experiments enabled

a balanced approach between model performance and resource utilization, crucial for maximizing research outcomes within the given constraints.

Experiment 2: LSTM Layer

Inspired by the success of recurrent neural networks in sequence modeling tasks, the next experiment replaced the dense layer with a Long Short-Term Memory (LSTM) layer. LSTM networks, introduced by Hochreiter and Schmidhuber [68], are designed to process sequential data, which could potentially enhance the model's ability to capture short-term temporal patterns relevant for onset detection, even within the limited context of our 7-frame input sequences.

The integration of a Long Short-Term Memory (LSTM) layer into the Convolutional Neural Network (CNN) architecture represents a shift in the approach to onset detection by incorporating temporal dependencies in musical audio. This modification allows the LSTM layer to process the output of the convolutional layers on a frame-by-frame basis, in contrast to approaches that analyze entire spectrograms as a single input.

The modifications for this experiment included:

1. Replacing the Flatten layer with a Reshape layer to prepare the output of the convolutional layers for the LSTM;
2. Replacing the Dense layer of 256 units with an LSTM layer of 128 units.

Table 3: Results table: Added an LSTM layer

Model	**F1-score**	**Description**
Böck CNN (Original)	0.903	Reported performance in Böck and Schlüter [63]
CNN (R)	0.898	Replication of Böck and Schlüter [63]
CNN (R) ES BN	0.900	Added Batch Normalization
CNN (R) ES BN LSTM	0.897	Added an LSTM layer

The introduction of the LSTM layer aimed to enhance the model's capacity to capture temporal dependencies essential for onset detection in audio streams. Although the F1-score slightly decreased to 0.897, as illustrated in Table 3, this result remains however very close to the previous configurations.

This small change indicates that the LSTM layer maintains a comparable performance level, suggesting that the model's ability to learn from previous time steps is still valuable, even if the overall score did not show improvement. The LSTM's role in modeling sequences can be particularly beneficial in more complex datasets or tasks where temporal dynamics are crucial.

Overall, while the immediate performance did not surpass expectations, the inclusion of the LSTM highlights a promising direction for future iterations of the model. With further tuning or adjustments, it may unlock even greater potential for capturing intricate patterns in audio data.

Experiment 3: Double LSTM Layers

Given the success of the single LSTM layer, the hypothesis was formed that adding a second LSTM layer might allow the model to capture even more complex temporal patterns. This experiment aimed to test whether a deeper recurrent structure could further improve onset detection performance.

The modification for this experiment was:

1. Adding a second LSTM layer of 64 units after the first LSTM layer.

Table 4: Results table: Added a second LSTM layer

Model	F1-score	Description
Böck CNN (Original)	0.903	Reported performance in Böck and Schlüter [63]
CNN (R)	0.898	Replication of Böck and Schlüter [63]
CNN (R) ES BN	0.900	Added Batch Normalization
CNN (R) ES BN LSTM	0.897	Added an LSTM layer
CNN (R) ES BN 2LSTM	0.898	Added a second LSTM layer

The addition of a second LSTM layer resulted in an F1-score of 0.898, which is practically identical to the performance of the model with a single LSTM layer (0.897) and the initial CNN replication (0.898), as shown in Table 4. This result suggests that increasing the depth of the recurrent part of the network by adding a second LSTM layer does not lead to a meaningful improvement in onset detection accuracy. Instead, it introduces additional computational complexity without providing a significant performance gain. This

observation highlights the importance of carefully considering the trade-off between model complexity and performance improvements when designing neural network architectures for onset detection tasks.

Experiment 4: Bidirectional LSTM

The next refinement explored the use of bidirectional LSTM layers. Bidirectional LSTMs, introduced by Schuster and Paliwal [69], process the input sequence in both forward and backward directions. This allows the network to capture both past and future context, which could be particularly useful for onset detection where the surrounding context in both directions can provide important cues.

The modification for this experiment was:

1. Replacing the unidirectional LSTM layers with a bidirectional LSTM layer.

Table 5: Results table: Added a BiLSTM layer

Model	F1-score	Description
Böck CNN (Original)	0.903	Reported performance in Böck and Schlüter [63]
CNN (R)	0.898	Replication of Böck and Schlüter [63]
CNN (R) ES BN	0.900	Added Batch Normalization
CNN (R) ES BN LSTM	0.897	Added an LSTM layer
CNN (R) ES BN 2LSTM	0.898	Added a second LSTM layer
CNN (R) ES BN BiLSTM	0.898	Added a bi-directional LSTM layer

As shown in Table 5, the combination of bidirectional LSTM and batch normalization achieved an F1-score of 0.898. While this does not outperform the best model so far, it does match the performance of the double LSTM, suggesting that the bidirectional architecture can achieve similar performance with potentially fewer parameters. This architecture presents a good balance between model complexity and performance, which could be advantageous in scenarios where computational resources are limited.

Experiment 5: CNN with Increased Feature Maps and BiLSTM

Building upon the modifications introduced in previous experiments, this iteration focused on enhancing the CNN architecture by increasing the number of feature maps in the convolutional layers. This experiment aimed to explore the benefits of increased model

capacity while retaining the improvements made in earlier stages of the study. The architecture for this experiment can be observed in the Figure 19.

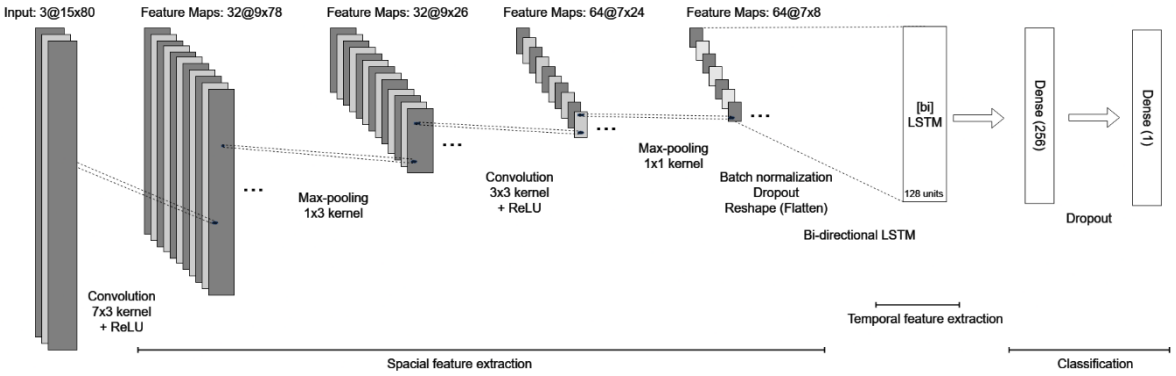

Figure 19: Custom Network Architecture

The key modification in this experiment was the substantial increase in the number of feature maps in the convolutional layers. Specifically, the number of feature maps in the first convolutional layer was increased from 10 to 32, and in the second convolutional layer from 20 to 64. This change was implemented while maintaining the same input format as the previous experiments, ensuring consistency in the data representation across all models.

Table 6: Results table: CNN with Increased Feature Maps and BiLSTM

Model	**F1-score**	**Description**
Böck CNN (Original)	0.903	Reported performance in Böck and Schlüter [63]
CNN (Replication)	0.898	Replication of Böck and Schlüter [63]
CNN (R) ES BN	0.900	Added Batch Normalization
CNN (R) ES BN LSTM	0.897	Added an LSTM layer
CNN (R) ES BN 2LSTM	0.898	Added a second LSTM layer
CNN (R) ES BN BiLSTM	0.898	Added a bi-directional LSTM layer
CNN (+FM) ES BN BiLSTM	0.902	Increased CNN feature maps with BiLSTM

Table 6 presents a comparison of model iterations, showcasing the progression from the initial replication to the current enhanced model.

The model with increased filters achieved an F1-score of 0.902, which is a modest improvement over both the initial replication (0.898) and the previous experimental iterations (0.897-0.898). This performance is closer to that reported in the original Böck and

Schlüter paper [63], though the gain is relatively small considering the increased computational complexity.

The improved performance of this model can be primarily attributed to the increased convolutional capacity. By increasing the number of feature maps to 32 and 64 in the respective convolutional layers, the model can perform more complex feature extraction. This enhancement potentially allows the network to capture subtler onset-related patterns in the input spectrograms.

The current model, with its increased number of filters, shows considerable promise. It achieves a 0.4 percentage point absolute improvement in F1-score over the initial replication, bringing the performance very close to the state-of-the-art result reported by Böck and Schlüter. This improvement demonstrates the potential benefits of increasing the model's capacity to learn more nuanced features from the input data.

Experiment 6: Enhanced CNN without LSTM's

Due to the success of the previous experiment, this following one aimed to isolate the impact of increased convolutional capacity without the LSTM layers. The motivation was to determine whether the improved performance could be achieved solely through enhancements to the CNN architecture.

The architecture in this experiment maintained the increased number of feature maps (32 and 64) in the convolutional layers, as implemented in Experiment 5. However, all LSTM layers were removed, resulting in a pure CNN model with enhanced capacity very similar to the one specified in [63].

Table 7: Results table: CNN with Increased Feature Maps without LSTM

Model	**F1-score**	**Description**
Böck CNN (Original)	0.903	Reported performance in Böck and Schlüter [63]
CNN (Replication)	0.898	Replication of Böck and Schlüter [63]
CNN (R) ES BN	0.900	Added Batch Normalization
CNN (R) ES BN LSTM	0.897	Added an LSTM layer
CNN (R) ES BN 2LSTM	0.898	Added a second LSTM layer
CNN (R) ES BN BiLSTM	0.898	Added a bi-directional LSTM layer
CNN (+FM) ES BN BiLSTM	0.902	Increased feature maps with BiLSTM

CNN (+FM) ES BN	0.905	Increased feature maps without LSTM

As shown in Table 7, the CNN with increased feature maps and without LSTM achieved an F1-score of 0.905, surpassing the results reported in Böck's original paper [63]. This is a significant finding, as it demonstrates that a relatively small modification to Böck's original architecture – namely, increasing the number of feature maps – can lead to state-of-the-art performance.

This result suggests that the increased convolutional capacity alone is sufficient to capture the complex patterns necessary for accurate onset detection. The ability to achieve this performance without LSTM layers indicates that the enhanced CNN is capable of effectively modeling both local and broader temporal contexts within the input spectrograms.

Experiment 7: Alternative input formats

Throughout the previous experiments, alternative input dimensions of 15x120x1 and 15x80x1 were tested alongside the standard 15x80x3 format. This experiment aimed to systematically analyze the impact of these different input formats on model performance.

Three input formats were compared across all previous model architectures:

- 15x80x3 (standard);
- 15x120x1;
- 15x80x1.

It is important to note that each channel in the 15x80x3 format represents the same spectrogram but computed with different window sizes, providing multiple perspectives of the same audio input.

Consistently across all experiments, the alternative input formats (15x120x1 and 15x80x1) yielded inferior results compared to the standard 15x80x3 format. This performance degradation can be attributed to several factors:

- Loss of Multi-Scale Information: The 3-channel input in the 15x80x3 format provides the model with multi-scale spectral information. Each channel, computed with a different window size, captures temporal and frequency characteristics at varying resolutions. This multi-scale representation allows the model to

simultaneously consider both fine-grained and broader spectral patterns, which is crucial for accurate onset detection across diverse musical contexts.

- Reduced Input Richness: The single-channel inputs (15x120x1 and 15x80x1) contain less overall information compared to the 3-channel input. Even with increased frequency resolution in the 15x120x1 format, the lack of multi-scale perspective limits the model's ability to discern onset patterns that may be more apparent at different time-frequency resolutions.
- Model Architecture Optimization: The CNN architectures, originally designed for the 15x80x3 input, may not be optimal for processing single-channel inputs. The convolutional filters and subsequent layers are likely tuned to exploit the multi-channel structure of the standard input format.
- Temporal Context: The 3-channel input potentially provides a richer temporal context, allowing the model to better distinguish between transient onset events and sustained musical elements.

This experiment underscores the importance of input representation in onset detection tasks. The superior performance of the 15x80x3 format demonstrates that providing diverse perspectives of the same audio input, through multiple spectrogram channels, is more beneficial than increasing resolution in a single channel or simplifying to a single-channel input.

These findings highlight the critical role of input design in music information retrieval tasks and suggest that future work should carefully consider multi-scale and multi-perspective input representations to maximize model performance.

6.5. Results Summary

This study aimed to enhance the performance of musical onset detection through the refinement and extension of convolutional neural network (CNN) architectures. Building upon the seminal work of Böck and Schlüter, a series of experiments was conducted to progressively improve model performance. The research culminated in the development of an enhanced CNN with increased feature maps and batch normalization, but without LSTM layers, that demonstrated modest improvements over the original Böck CNN model, achieving a state-of-the-art F1-score of 0.905.

Key findings of this study include:

- Baseline Replication: The initial replication of Böck and Schlüter's CNN model achieved F1-scores of 0.859 and 0.900 for exact and fuzzy labelling strategies, respectively. These results were slightly lower than the original model's reported F1-scores of 0.885 and 0.903. This discrepancy is consistent with replication attempts by other researchers, such as Gong and Serra [66], highlighting the challenges in exact reproduction of deep learning models;
- Early Stopping and Learning Rate Adjustment: Implementing early stopping and using a fixed learning rate of 0.05 achieved an F1-score of 0.898. While slightly underperforming the initial fuzzy labelling implementation, this approach significantly reduced training time, demonstrating a good balance between performance and computational efficiency;
- Batch Normalization: The addition of batch normalization improved the model's F1-score from 0.898 to 0.900. More importantly, it reduced training time by half, showcasing significant efficiency gains;
- LSTM Layers:
    - Single LSTM: Replacing the dense layer with an LSTM layer resulted in an F1-score of 0.897, slightly lower but comparable to the batch-normalized CNN;
    - Double LSTM: Adding a second LSTM layer improved the F1-score to 0.898, suggesting potential benefits of deeper recurrent structures;
    - Bidirectional LSTM: The BiLSTM configuration achieved an F1-score of 0.898, matching the performance of the double LSTM but potentially with fewer parameters.
- Increased Convolutional Capacity: An improvement in the results was achieved by increasing the number of feature maps in the convolutional layers from 10 to 32 in the first layer and from 20 to 64 in the second layer.
    - With BiLSTM: This configuration achieved an F1-score of 0.902, a substantial improvement over the initial replication;
    - Without LSTM: Remarkably, removing the LSTM layers and relying solely on the enhanced CNN architecture resulted in the best performance, with an F1-score of 0.90. This surpassed the original Böck and Schlüter results, demonstrating that increased convolutional capacity alone can effectively capture both local and broader temporal contexts for onset detection.

- Input Representation: Experiments with alternative input formats (15x120x1 and 15x80x1) consistently yielded inferior results compared to the standard 15x80x3 format. This highlights the importance of multi-scale spectral information in onset detection tasks.

The progression of model performance across different architectural modifications was evaluated by comparing F1-scores for all experimental configurations across the 8 folds. Figure 20 illustrates this comparison, presenting the F1 Score results for each experiment.

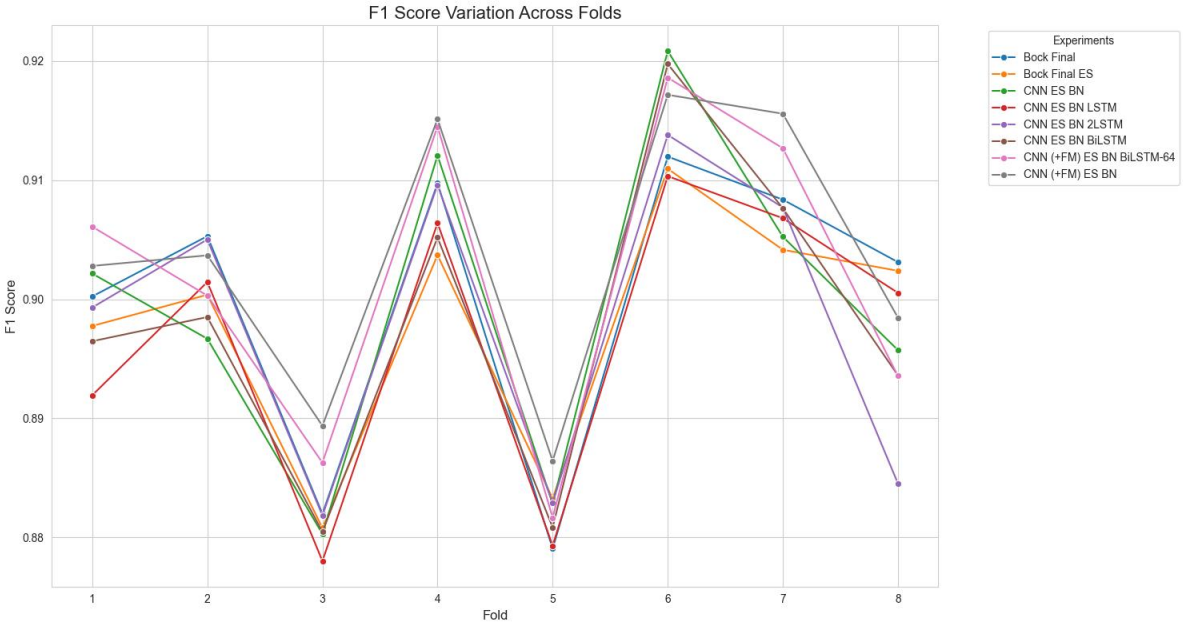

Figure 20: F1 Score Comparison Across experiments

These results collectively demonstrate that while LSTM layers showed promise in capturing temporal dependencies, the most significant improvements came from increasing the convolutional capacity of the network. The final model, a CNN with increased feature maps and batch normalization but without LSTM layers, achieved state-of-the-art performance, surpassing even the original Böck and Schlüter results.

This study highlights the importance of convolutional capacity in capturing complex patterns necessary for accurate onset detection. The fact that the best performance was achieved without LSTM layers suggests that a well-designed CNN can effectively model both local and broader temporal contexts within the input spectrograms for this task. Furthermore, the consistent superiority of the 15x80x3 input format underscores the critical role of multi-scale spectral information in onset detection.

7. Conclusions and Future Work

This dissertation has explored the application of time-aware neural networks to the task of musical onset detection, building upon the foundational work of Böck and Schlüter (2014). The research focused on investigating how neural network architectures can be optimized to better capture the temporal aspects of music, which are crucial for accurate onset detection.

The study began with a replication of Böck and Schlüter's convolutional neural network (CNN) model, which remains a strong baseline in the field. This replication achieved F1-scores of 0.859 and 0.900 for exact and fuzzy labelling strategies, respectively. While these scores were slightly lower than those originally reported, they confirmed the robustness of Böck and Schlüter's approach and provided a solid foundation for further experimentation.

A series of incremental modifications were then explored, aimed at enhancing the model's ability to process temporal information. These included the addition of batch normalization, experimentation with Long Short-Term Memory (LSTM) layers, and adjustments to the convolutional layer capacities. The most notable improvement came from increasing the number of feature maps in the convolutional layers, which allowed the network to capture a richer set of time-frequency patterns.

Interestingly, while LSTM layers were investigated for their potential to model long-term dependencies in the audio signal, the best performing model ultimately did not include these recurrent elements. Instead, a CNN with increased feature maps (32 in the first layer, 64 in the second) and batch normalization achieved the highest F1-score of 0.905. This result, while a modest improvement over the baseline, demonstrates that carefully tuned convolutional architectures can effectively capture the temporal context necessary for onset detection.

The research also highlighted the critical role of input representation in model performance. Experiments with alternative input formats consistently showed that the standard multi-channel spectrogram representation (15x80x3) outperformed single-channel alternatives. This finding underscores the importance of providing the network with multi-scale temporal information, aligning with the time-aware focus of the study.

While the improvements achieved in this study were incremental, they provide valuable insights into the design of time-aware neural networks for onset detection. The success of

the enhanced CNN model suggests that increasing the capacity of early convolutional layers can lead to better temporal feature extraction, potentially obviating the need for more complex recurrent structures in this specific task.

Future work in this area could explore several promising directions. One avenue would be to investigate more sophisticated time-aware architectures, such as temporal convolutional networks or attention mechanisms, which might offer new ways to model the multi-scale temporal dependencies in music. Another direction could be to explore adaptive input representations that dynamically adjust to different temporal scales present in various musical genres or instruments.

The potential of transfer learning in onset detection also warrants further investigation. By pre-training models on large, diverse datasets and fine-tuning them for specific musical contexts, it might be possible to develop more robust and generalizable onset detection systems.

Additionally, future research could focus on making these models more interpretable, potentially revealing insights into which temporal patterns are most salient for onset detection across different musical styles. This could not only improve model performance but also contribute to the understanding of musical structure and perception.

In conclusion, while this study has made modest contributions to the field of onset detection through careful tuning and experimentation with time-aware neural network architectures, it also reaffirms the strength of Böck and Schlüter's original approach. The work underscores the challenges involved in significantly advancing the state-of-the-art in this domain and highlights the ongoing importance of considering temporal dynamics in music information retrieval tasks. As research in this field progresses, the continued exploration of time-aware neural network architectures promises to yield further insights and incremental improvements in the ability to automatically analyze and understand musical structure.

8. References

[1] U. Zölzer, Digital Audio Signal Processing, 1st ed. Chichester, UK: John Wiley & Sons, 1997.

[2] N. . Degara, M. E. P. Davies, A. . Pena and M. D. Plumbley, "Onset Event Decoding Exploiting the Rhythmic Structure of Polyphonic Music," *IEEE Journal of Selected Topics in Signal Processing,* vol. 5, no. 6, pp. 1228-1239, 2011.

[3] A. Lerch, An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics, John Wiley & Sons, 2012.

[4] S. Handel, "Timbre perception and auditory object identification," in Hearing, B. C. J. Moore, Ed. San Diego, CA: Academic Press, 1995, pp. 425-461.

[5] Tutorials Point, "Digital Communication - Sampling," TutorialsPoint. [Online]. Available: https://www.tutorialspoint.com/digital_communication/digital_communication_sampling.htm. [Accessed: Sept. 28, 2024].

[6] H. Nyquist, "Certain Topics in Telegraph Transmission Theory," Transactions of the American Institute of Electrical Engineers, vol. 47, no. 2, pp. 617-644, Apr. 1928.

[7] C. E. Shannon, "Communication in the Presence of Noise," Proceedings of the IRE, vol. 37, no. 1, pp. 10-21, Jan. 1949.

[8] L. Cohen, Time-Frequency Analysis. Englewood Cliffs, NJ: Prentice Hall, 1995.

[9] A. V. Oppenheim and R. W. Schafer, Discrete-Time Signal Processing, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 1998.

[10] K. C. Pohlmann, Principles of Digital Audio, 5th ed. New York, NY: McGraw-Hill Professional, 2005.

[11] R. E. Crochiere and L. R. Rabiner, Multirate Digital Signal Processing. Englewood Cliffs, NJ: Prentice Hall, 1983.

[12] R. N. Bracewell, The Fourier Transform and Its Applications, 2nd ed. New York, NY: McGraw-Hill, 1986.

[13] A. D. Poularikas, Ed., The Transforms and Applications Handbook, 1st ed. Boca Raton, FL: CRC Press, 1996.

[14] L. Wyse, "Audio spectrogram representations for processing with Convolutional Neural Networks," in Proceedings of the First International Workshop on Deep Learning for Music, Anchorage, AK, USA, 2017, pp. 1-5.

[15] K. Chaudhary, "Understanding Audio data, Fourier Transform, FFT and Spectrogram features for a Speech Recognition System," Drops of AI, 24 Jul. 2024. [Online]. Available: https://dropsofai.com/understanding-audio-data-fourier-transform-fft-and-spectrogram-features-for-a-speech-recognition-system/. [Accessed: 8 Feb. 2024].

[16] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler, "A Tutorial on Onset Detection in Music Signals," IEEE Transactions on Speech and Audio Processing, vol. 13, no. 5, pp. 1035-1047, Sep. 2005.

[17] S. S. Stevens, J. Volkmann, and E. B. Newman, "A Scale for the Measurement of the Psychological Magnitude Pitch," The Journal of the Acoustical Society of America, vol. 8, no. 3, pp. 185-190, Jan. 1937.

[18] M. A. Bartsch and G. H. Wakefield, "To catch a chorus: using chroma-based representations for audio thumbnailing," in Proceedings of the 2001 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, NY, USA, 2001, pp. 15-18.

[19] Y.-H. Chin, "Music Emotion Detection Using Hierarchical Sparse Kernel Machines," Ph.D. dissertation, Dept. Elect. Eng., National Taiwan University, Taipei, Taiwan, 2014.

[20] M. Müller and S. Ewert, "Chroma Toolbox: MATLAB Implementations for Extracting Variants of Chroma-Based Audio Features," in Proceedings of the 12th

International Society for Music Information Retrieval Conference (ISMIR 2011), Miami, FL, USA, 2011, pp. 215-220.

[21] J. O. Smith III, Spectral Audio Signal Processing. Stanford, CA: W3K Publishing, 2011.

[22] D. W. Kammler, A First Course in Fourier Analysis, 2nd ed. Cambridge, UK: Cambridge University Press, 2010.

[23] S. W. Smith, The Scientist and Engineer's Guide to Digital Signal Processing, 2nd ed. San Diego, CA: California Technical Publishing, 1999.

[24] J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," Mathematics of Computation, vol. 19, no. 90, pp. 297-301, Apr. 1965.

[25] J. B. Allen and L. R. Rabiner, "A Unified Approach to Short-Time Fourier Analysis and Synthesis," Proceedings of the IEEE, vol. 65, no. 11, pp. 1558-1564, Nov. 1977.

[26] H. Jeon, Y. Jung, S. Lee, and Y. Jung, "Area-Efficient Short-Time Fourier Transform Processor for Time–Frequency Analysis of Non-Stationary Signals," IEEE Access, vol. 8, pp. 137546-137557, 2020.

[27] M. R. Portnoff, "Time-Frequency Representation of Digital Signals and Systems Based on Short-Time Fourier Analysis," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 28, no. 1, pp. 55-69, Feb. 1980.

[28] P. Flandrin, Time-Frequency/Time-Scale Analysis. San Diego, CA: Academic Press, 1998.

[29] J. C. Brown, "Calculation of a Constant Q Spectral Transform," Journal of the Acoustical Society of America, vol. 89, no. 1, pp. 425-434, Jan. 1991.

[30] C. Schörkhuber, A. Klapuri, N. Holighaus, and M. Dörfler, "A Matlab Toolbox for Efficient Perfect Reconstruction Time-Frequency Transforms with Log-Frequency

Resolution," in Proceedings of the AES 53rd International Conference on Semantic Audio, London, UK, 2014, pp. 1-8.

[31] I. Daubechies, Ten Lectures on Wavelets. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1992.

[32] S. Mallat, A Wavelet Tour of Signal Processing: The Sparse Way, 3rd ed. Burlington, MA: Academic Press, 2009.

[33] G. Strang and T. Nguyen, Wavelets and Filter Banks. Wellesley, MA: Wellesley-Cambridge Press, 1996.

[34] C. K. Chui, An Introduction to Wavelets. San Diego, CA: Academic Press, 1992.

[35] Y. Meyer, Wavelets: Algorithms & Applications. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1993.

[36] S. B. Davis and P. Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 28, no. 4, pp. 357-366, Aug. 1980.

[37] L. R. Rabiner and R. W. Schafer, Digital Processing of Speech Signals. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1978.

[38] F. J. Harris, "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform," Proceedings of the IEEE, vol. 66, no. 1, pp. 51-83, Jan. 1978.

[39] B. Fuentes, R. Badeau, and G. Richard, "Blind Harmonic Adaptive Decomposition Applied to Supervised Source Separation," in Proceedings of the 20th European Signal Processing Conference (EUSIPCO), Bucharest, Romania, 2012, pp. 2654-2658.

[40] S. Böck, F. Krebs, and M. Schedl, "Evaluating the Online Capabilities of Onset Detection Methods," in Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR), Porto, Portugal, 2012, pp. 49-54.

[41] M. Mauch and S. Dixon, "A Fundamental Frequency Estimator Using Probabilistic Threshold Distributions," in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 2012, pp. 369-372.

[42] P. Masri, "Computer Modeling of Sound for Transformation and Synthesis of Musical Signals," Ph.D. dissertation, Dept. of Elect. and Electron. Eng., University of Bristol, Bristol, UK, 1996.

[43] S. Dixon, "Onset detection revisited," in Proceedings of the 9th International Conference on Digital Audio Effects (DAFx-06), Montreal, Canada, 2006, pp. 133-137.

[44] T. M. Mitchell, Machine Learning. New York, NY: McGraw-Hill, 1997.

[45] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA: MIT Press, 2016.

[46] O. Dürr, B. Sick, and E. Murina, Probabilistic Deep Learning With Python, Keras and TensorFlow Probability. Birmingham, UK: Packt Publishing, 2020.

[47] C. M. Bishop, Pattern Recognition and Machine Learning. New York, NY: Springer, 2006.

[48] L. Bottou, "Large-Scale Machine Learning with Stochastic Gradient Descent," in Proceedings of COMPSTAT'2010, Paris, France, 2010, pp. 177-186.

[49] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," Nature, vol. 323, no. 6088, pp. 533-536, Oct. 1986.

[50] S. Ruder, "An overview of gradient descent optimization algorithms," arXiv preprint arXiv:1609.04747, 2016.

[51] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," in Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI), Montreal, Canada, 1995, pp. 1137-1143.

[52] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[53] Q. Xie, "Development of a Highly Transferable Urban Winter Road Surface Classification Model: A Deep Learning Approach," M.S. thesis, Dept. of Civil and Environmental Engineering, University of Waterloo, Waterloo, ON, Canada, 2022.

[54] D. Scherer, A. Müller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in Artificial Neural Networks – ICANN 2010, K. Diamantaras, W. Duch, L. S. Iliadis, Eds. Berlin, Heidelberg: Springer, 2010, pp. 92-101.

[55] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735-1780, Nov. 1997.

[56] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," IEEE Transactions on Neural Networks, vol. 5, no. 2, pp. 157-166, Mar. 1994.

[57] T. M. Ingolfsson, "Insights into LSTM architecture," [Online]. Available: https://thorirmar.com/post/insight_into_lstm/. [Accessed: Sept. 21, 2024].

[58] L. S. Smith, "Onset-based Sound Segmentation," in Advances in Neural Information Processing Systems 8 (NIPS 1995), D. S. Touretzky, M. C. Mozer, M. E. Hasselmo, Eds. Cambridge, MA: MIT Press, 1996, pp. 729-735.

[59] M. Marolt, A. Kavcic, and M. Privosnik, "Neural Networks for Note Onset Detection in Piano Music," in Proceedings of the International Computer Music Conference (ICMC), Gothenburg, Sweden, 2002, pp. 39-42.

[60] A. Lacoste and D. Eck, "A Supervised Classification Algorithm for Note Onset Detection," EURASIP Journal on Advances in Signal Processing, vol. 2007, Article ID 43745, 2007.

[61] E. Kapanci and A. Pfeffer, "A Hierarchical Approach to Onset Detection," in Proceedings of the International Computer Music Conference (ICMC), Miami, FL, USA, 2004, pp. 438-441.

[62] S. Böck, M. Schedl, A. Arzt, and F. Krebs, "Online Real-time Onset Detection with Recurrent Neural Networks," in Proceedings of the 15th International Conference on Digital Audio Effects (DAFx-12), York, UK, 2012, pp. 1-4.

[63] S. Böck and J. Schlüter, "Improved Musical Onset Detection with Convolutional Neural Networks," in 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 2014, pp. 6979-6983.

[64] R. Gong and X. Serra, "Towards an Efficient Deep Learning Model for Musical Onset Detection," 2018. [Online]. Available: https://arxiv.org/abs/1806.06773

[65] M. Tomczak and J. Hockman, "Onset Detection for String Instruments Using Bidirectional Temporal and Convolutional Recurrent Networks," in Proceedings of Audio Mostly 2023 (AM '23), Edinburgh, United Kingdom, Aug. 2023. DOI: https://doi.org/10.1145/3616195.3616206.

[66] F. Eyben, S. Böck, B. Schuller, and A. Graves, "Universal Onset Detection with Bidirectional Long Short-Term Memory Neural Networks," in Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR), Utrecht, Netherlands, 2010, pp. 589-594.

[67] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in Proceedings of the 32nd International Conference on Machine Learning (ICML), Lille, France, 2015, pp. 448-456.

[68] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[69] M. Schuster and K. K. Paliwal, "Bidirectional Recurrent Neural Networks," IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673-2681, Nov. 1997.

[70] P. Burk, L. Polansky, D. Repetto, M. Roberts, and D. Rockmore, "Music and Computers," [Online]. Available: https://musicandcomputersbook.com/chapter3/03_01.php. [Accessed: Feb. 7, 2024].

[71] A. L. Samuel, "Some Studies in Machine Learning Using the Game of Checkers," IBM Journal of Research and Development, vol. 3, no. 3, pp. 210-229, Jul. 1959.

[72] J. L. Elman, "Finding Structure in Time," Cognitive Science, vol. 14, no. 2, pp. 179-211, Apr. 1990.

[73] Y. Bengio, P. Simard, and P. Frasconi, "Learning Long-Term Dependencies with Gradient Descent is Difficult," IEEE Transactions on Neural Networks, vol. 5, no. 2, pp. 157-166, Mar. 1994.

[74] K. Cho et al., "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 2014, pp. 1724-1734.

[75] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer, "madmom: a new Python Audio and Music Signal Processing Library," in Proceedings of the 24th ACM International Conference on Multimedia, Amsterdam, Netherlands, 2016, pp. 1174-1178.