CNN-Based Models for Pitch Estimation, Modification, and Auto-Tuning

by

Jiazhuo Jiang

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF COMPUTER SCIENCE

In the Department of Computer Science

CNN-Based Models for Pitch Estimation, Modification, and Auto-Tuning

by

Jiazhuo Jiang
B. Sc. University of Victoria, 2023

Supervisory Committee

Dr. George Tzanetakis, Supervisor
(Department of Computer Science)

Dr. Alex Thomo, Departmental Member
(Department of Computer Science)

ABSTRACT

Pitch estimation and pitch modification are fundamental audio processing tasks that are used in a variety of applications. An important example is the auto-tuning of vocals in which pitch estimation is applied, deviations from a desired target pitch are calculated, and the pitch of input vocal signal is modified to match the target pitch. Most existing approaches to auto-tuning are based on traditional digital signal processing (DSP) techniques for both the pitch detection and the pitch modification of the signal. In this thesis, the use of Convolutional Neural Networks (CNNs) is explored as a possible replacement of traditional DSP methods for pitch estimation, pitch modification as well as end-to-end autotuning. CNNs can model complex intput and output relationships and are more efficient than deep learning methods that take into account time/sequence information such as Long Term/Short Term (LSTM) networks and Recurrent Neural Networks (RNNs). The results show the potential of this approach as well as some of the challenges that need to be overcome. The experimental results indicate that larger data sets can result in better accuracy but they also tend to bring in more noise.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I would like to thank:

George Tzanetakis for mentoring, support, encouragement, and patience.

DEDICATION

Just hoping this is useful!

Chapter 1

Introduction

There is an increasing trend to move away from traditional Digital Signal Processing (DSP) methods for audio processing to deep learning approaches. For example, monophonic pitch estimation is an audio task that traditionally has been tackled using DSP approaches based on various representations such as time-domain, frequency-domain, and autocorrelation. In addition, some times pitch detection systems also take into account also information about the perceptual characteristics of the human auditory system. An alternative approach to traditional DSP approaches is to train deep learning networks by directly providing large amounts of audio data annotated with the corresponding ground truth of what the desired answer should be. A key influential example in this direction is the CREPE (Constant-Q Transform-based Pitch Estimation) model, as it has been observed to enhance the accuracy of pitch estimation when applied in various situations[13]. CREPE is also able perform the inference in real time as the audio is being processed.

Pitch detection is a fundamental problem in audio processing, and it is used in various applications, including the transcription of music and speech and bioacoustics [11, 12]. Pitch detection by conventional techniques gives rise to certain problems, particularly when the pitch is detected from noisy signals or signals containing more than one tone. These challenges are partly overcome by the CREPE model, which uses a deep convolutional neural network (CNN) that integrates the Constant-Q Transform (CQT) and is capable of making precise pitch predictions from raw audio waveforms in a short time without the need for extensive pre-processing as has been employed in previous works [13].

In this thesis, we show how an existing pitch detection method (whether DSP-based or deep learning based) can be used to train a deep learning model for pitch

detection that is tailored to a specific type of input audio or dataset. In a similar fashion we also explore how pitch modification can modeled in the same fashion by using an existing DSP system to train a corresponding deep learning model. Auto-tuning consists of pitch detection, pitch adjustment calculation, and pitch modification. Finally, the possibility of using a single end-to-end deep learning approach for auto-tuning is explored.

1.1 Preliminaries

Auto-tuning is a key task in audio processing, where pitch detection servers as a fundamental component. As the name suggests, pitch detection involves detecting the first harmonic present in a sound signal that is variable with time. Since pitch is a key ingredient in the subtle processing of a signal, many such applications exist [17]. Different characteristics exist to evaluate pitch detection methods, the most common of which are effectiveness, reliability, and real-time processing capabilities.

The CREPE model represents a breakthrough in pitch detection in that it is a fully deep-learning model. Because of the Constant-Q Transform, CREPE is able to capture the time and frequency information of audio signals, therefore making it adaptable to different acoustic environments, as stated in [23].

The introduction mentions previous research that provides a foundation upon which pitch detection was built, whereas the background section provides an explicit summary of the background which led to the development and evaluation of the different systems described in the thesis. Methods based on autocorrelation and cepstral analysis have been the most popular ones for pitch detection. While these approaches have some advantages for real-world applications owing to their traditions, they pose performance problems, especially in highly noisy or complex audio signal conditions. The literature discussed shows that such conventional pitch estimation methods have a limitation of accuracy on estimates of pitch since they depend on a set of engineered features with prior put-in processing that is not well suited to the signal's quality and acoustics [12, 6].

To overcome the limitations of previously existing models of pitch detection, recent trends in learning have started the establishment of several sophisticated pitch detection techniques. Several reports on RNNs and CNNs have suggested that these networks have reported better classification performance because it is possible to learn from raw audio data directly. In this paradigm, the advantage of the accurate rep-

resentation of the complex temporal and spectral structures present in the hysteresis signals is also effectively put to use [22]. These improvements can be found in the CREPE model that efficiently employs the Constant-Q Transform (CQT) embedded Diagrams in a Convolutional Neural Network (CNN) architecture, which enables the model to produce fairly accurate pitch predictions even in pitch-obscuring noisy environments. This stream of research is worth mentioning because it not only enhances the accuracy of this component but also increases the usability of such models for other audio analysis purposes.

1.2 Thesis Structure

This thesis is organized as follows:

- **Chapter 2: Problem Definition and Challenges** — This chapter discusses the specific challenges of pitch detection in noisy environments and reviews the limitations of traditional methods.
- **Chapter 3: Proposed Model** — Introduces the CNN-based approach used in this research, detailing the architecture and describing how it addresses pitch detection, modification, and auto-tuning.
- **Chapter 4: Methodology** — Describes the experimental setup, datasets, and evaluation metrics used to compare the CNN-based approach with traditional and recurrent models.
- **Chapter 5: Results and Evaluation** — Presents the outcomes of the experiments, providing comparisons of the CNN model with recurrent models and traditional methods and an analysis of performance in terms of accuracy, speed, and generalization.
- **Chapter 6: Conclusions and Future Work** — Summarizes the key findings, highlights the potential of CNN models for real-time pitch detection, and suggests directions for future research in audio processing.

This thesis investigates CNN-based models as a promising solution for frame-based pitch detection, modification and auto-tuning. By examining various configurations and applications, this research aims to advance the understanding. This thesis examines CNNs in audio processing and identifies practical solutions for efficient pitch

adjustment. Through empirical evaluations, it demonstrates the potential for CNNs to offer a balance of accuracy and efficiency that supports both traditional and real-time applications in audio processing.

Chapter 2

Problem Definition and Challenges

Although many methods [5, 2, 7] have been proposed for the identification of pitch in an audio processing system, the process is still complex; by integrating the CREPE model and leveraging the unique features of various machine learning models, this research explores innovative solutions for auto-tuning. Subsequent chapters will discuss the model architecture, evaluation methodologies, and empirical results, as well as demonstrate how the combined strengths of different models enhance pitch accuracy and adaptability in complex audio contexts.

2.1 Understanding the Challenges

To effectively define the problem of pitch detection and auto-tuning, it is essential to examine the key challenges that impede traditional algorithms:

2.1.1 Noise Interference

Specifically, in practical scenarios, the audio speech signal is often accompanied by background noise, which poses a great challenge when determining the fundamental frequency [18, 20]. Interference can be of many types and can come from different sources, such as noise interference, intermodulation interference, or the recording system. Conventional pitch detection techniques may also fail to work well under such circumstances by producing wrong pitch estimates.

2.1.2 Complex Signal Characteristics

Since different audio signal sources include polyphonic music, speech, and natural sounds, the nature of pitch detection becomes even more challenging [16, 9]. These aspects of the signal mean that what is optimal in one case can be suboptimal in another, such that standard techniques cannot easily transfer from one signal type to another. For instance, polyphonic music will feature not one pitch but rather simultaneous pitches, and speech will feature assorted pitches and intonations. Such considerations require finer methods with the ability to track a hypothesized pitch given interfering frequencies.

2.1.3 Real-Time Processing Requirements

Most practical uses of pitch detection, such as live music performance and real-time speech recognition, call for real-time algorithms. For real-time applications, more sophisticated techniques may not be feasible due to the time taken for pre-processing or, in the case of signal processing techniques, complex signal analysis.Therefore, models with high accuracy and quick response to shifts in the audio environment are much desired.

2.2 Research Motivation

The motivation for this study stems from the persistent challenges traditional algorithms face in high-pitch detection and auto-tuning, especially in the presence of noise and other interference within the audio stream. While techniques like autocorrelation and cepstral analysis have a long-standing history in this field, their feature extraction and preprocessing steps are often manual, introducing potential errors due to variations in audio characteristics and noise environments [1, 3]. These methods can also be time-consuming, limiting their applicability in real-time contexts [8, 24]. With advances in machine learning, new techniques will be explored to improve accuracy, processing efficiency, and adaptability under diverse and noisy conditions.

An approach to address these limitations is to employ deep learning methods and training using large datasets. An example of such an approach is the CREPE model, which employs convolutional neural networks (CNNs) together with the Constant-Q Transform (CQT). To this end, this modern strategy helps the CREPE model to

decode as many temporal and spectral characteristics of the coupled audio signals as it overcome the drawbacks of the traditional approaches to analysis.

This research evaluates the performance of the CREPE system and the underlying CNN architecture in high-pitch detection and beyond, integrating auto-tuning in the presence of noise. The model's adaptability shows promise for applications in fields such as music transcription, speech analysis, and bioacoustics, utilizing CNNs to enhance both pitch detection accuracy and auto-tuning capabilities.

As revealed below, the importance of this research goes beyond the academy. Enhancing pitch detection capabilities can lead to substantial advancements in several fields: Music Transcription: Application of the proposed CREPE model can enhance the accuracy of digital tools used by musicians and composers and thus provide better transcription of difficult musical pieces. **Speech Analysis:** Better pitch detection can be helpful in improving voice recognition systems by including user-friendly interaction with devices. **Bioacoustics:** Effective identification of animal vocalizations enhances the process of monitoring animal species and the health of ecosystems.

2.3 Literature Review

Pitch detection has been a central issue in audio processing for quite some time now, and a vast number of works have been dedicated to the development of new methods for enhancing the quality of the detection algorithms based on their accuracy and their ability to cope with real-life conditions. To provide the background of the proposed model CREPE, this literature review focuses on key developments of pitch detection methods and pitch estimation key findings together with the difficulties related to pitch detection, especially when the background noise is high.

2.3.1 Traditional Pitch Detection Methods

It is significant to realise that pitch detection algorithms have been, in the past, based on concepts that generally used signal processing, which focused on the mathematical treatment of signals [4, 10]. Examples of the techniques include autocorrelation and Cepstral analysis. For instance, Drugmanet al. [6] used autocorrelation to estimate pitch under a noiseless environment but pointed out that when noise is added, the algorithm's performance is highly distorted. The cepstral analysis applies a cepstral transformation to the signal for periodicity detection, but neither of them provides

an accurate solution when background noise interferes with the audio signal [17].

2.3.2 Challenges in Noisy Environments

Previous studies suggest that conventional approaches deteriorate in performance when tested on noisy signals of speech [21]. In the same regard, Jain et al. pointed out that pitch detection algorithms may drop their accuracy by a huge margin when a certain level of background noise is achieved and further underlined the impact of this finding in such related areas as speech recognition and music transcription. Scholars [11, 14] have tried to improve conventional methods with the help of preliminary steps, including spectral subtraction and the Wiener filter, in order to minimize noise before pitch estimation. However, all these approaches involve additional computation and still need to be more capable of providing reasonably accurate estimates for real-world applications.

2.3.3 Advancements in Machine Learning Techniques

Pitch detection has increasingly benefited from machine learning approaches rather than the former standard means of processing sounds. Out of these techniques, RNN and CNN have been shown to be the most suitable for learning the features of the audio signal. For example, Yi et al. [22] employed CNN as well to improve the performance of pitch tracking and to achieve a lower incidence of voicing errors. This was possible because of the growing popularity of deep learning techniques, where some issues posed by the traditional approaches were eluded, and deep learning enabled training from raw audio. Similarly, in research, Zhang et al. [23] examined the complex deep pitches classification systems that were developed without any feature hand selection, which have been improved tremendously over the previous systems.

These models appear effective but may need large quantities of labelled training data and can be time-consuming, which limits their use in real-time systems [19].

2.3.4 The Pitch Detection

The Pyin Model

The PYIN model is an improvement of monophonic pitch detection mechanisms, as it combines the classical YIN algorithm with more probabilistic modeling, thus performing better when there is an on-pitch tracking error situation, more so in

noisier environments. The use of probabilistic filters also leads to a reduction of octave errors and refinement of the pitch estimates, making it suitable for these applications involving voice activity detection and mono recordings of instruments. Preliminary results illustrate the strength and precision of PYIN by showing its superiority to the conventional approach based on YIN techniques in real-world scenarios.

The CREPE Model

The CREPE model illustrated the considerable progress in pitch detection by combining deep learning with the constant-Q transform (CQT). What CREPE actually does is leverage a CNN on the CQT representation of audio signals to extract and predict pitch under various conditions. CREPE has been preliminarily tested, and the results demonstrate higher rates of accuracy than standard approaches, starting from the noisy conditions and different types of audio materials. In a similar experimental study, Kim and Lee (2021) showed that CREPE can retain high efficiency in real-time applications, which makes this tool most beneficial for musicians, speech analysts, and ecologists [13]. The fact that CREPE is hence able to run without requiring substantial preprocessing is an added advantage in that it can seamlessly fit into application domains that require real-time response.

As a result, the literature on pitch detection presents a process that is continually developing with greater complexities as traditional algorithms struggle to estimate pitch accurately in noisy contexts. The introduction of new machine learning algorithms makes it possible to develop new solutions like the CREPE – a model based on deep learning and CQT that improves accuracy and makes pitch detection less sensitive to noise. This review also sets the stage for this research by pointing out the need for better approaches in pitch detection, especially for real application scenarios. The following chapters of the thesis will provide a detailed description of the applied methodology and the empirical analysis of the CREPE model, whose contributions to the audio processing field are his main focus.

In summary, this chapter discusses the multifaceted challenges of accurate pitch detection, particularly in noisy environments. By establishing the CREPE model as an innovative solution, this research aims to advance the discourse in audio processing. Subsequent chapters will delve into using CNNs for pitch detection and pitch modification, the methodology employed for its evaluation, and the empirical results that demonstrate the potential of the proposed approach.

2.3.5 Pitch Adjustment

Pitch adjustment changes the fundamental frequency (f_0) of an audio signal to match a target pitch, based on standard musical notes and their frequencies (e.g., A4 = 440 Hz). The process starts by detecting the current f_0 of the audio. Then, it adjusts the pitch to the closest standard note, ensuring it aligns with the musical scale. This method is widely used in auto-tuning to correct pitch while keeping the audio quality natural.

2.3.6 Pitch Modification

Pitch modification primarily focuses on adjusting the pitch of audio signals and is widely used in audio processing applications. This involves modifying the pitch to make it closer to a standard frequency, ensuring consistency and harmony in the audio. Common achieved by pitch shifting and pitch multiplication, which are applied to systematically alter the pitch based on required targets. These techniques form the foundation for auto tuning which is the next stage of the thesis. By enabling precise adjustments, pitch modification not only enhances the naturalness of the audio but also supports the broader framework of automated pitch correction and adjustment.

2.3.7 Auto Tuning

Auto tuning is the process of automatically correcting or adjusting the pitch of audio signals to match a desired musical scale or target frequency. It brings together the key steps of **pitch detection**, **pitch adjustment calculation**, and **pitch modification** into one unified process. By combining these steps, auto tuning fine-tunes audio segments to make them sound closer to the user's intended result. This approach not only improves the quality and harmony of the audio but also simplifies the pitch correction process, making it an essential tool in modern music and audio production. In later stages, neural networks such as CNNs will be employed to attempt to achieve the same outcomes as these three steps, aiming for a more efficient and integrated solution to auto tuning.

Chapter 3

Proposed Model

This chapter explains the proposed systems, especially the integration of components such as the Constant-Q Transform (CQT), CNN architecture, and the basic tuning strategy with novel deep learning methods. This model aims to respond to the issues related to pitch detection that were the focus of the previous chapters.

3.1 Basic Auto-tuning Method

3.1.1 Audio Segmentation and Overlapping

The first stage of the process consists of breaking the audio into equal segments or frames for the subsequent frames-based pitch alteration and final recombination processes. In order to reduce the discontinuity between frames and achieve seamless blending, overlap is used. For instance, let us consider that there are 10 samples and the frame size is 4; 25% overlap means the first frame contains samples 1, 2, 3 and 4. The second frame overlaps 25 percent and includes samples 4, 5, 6 and 7. Similarly, the third frame contains samples 7, 8, 9 and 10. This overlapping technique helps transition between frames and enhances the quality of the output after processing.

3.1.2 Pitch Detection with Pyin and CREPE

Estimating the fundamental frequency (f_0) for each frame serves as a reference for tunable pitch alterations. The two appropriate f_0 estimation methods are PYIN and CREPE. PYIN is based on a probabilistic approach for singing pitch detection in monophonic recordings, and CREPE is a high-accuracy technique using a CNN

regardless of underlying noise or a mix of sounds.

3.1.3 Pitch Matching and Shifting

At the initial stage of pitch alteration, the fundamental frequency f_0 for every audio frame is identified and used as a baseline during the modification process. Any detected f_0 is then adjusted to the nearest pitch of a pre-determined set pitch values in order to avoid excessive movement beyond the allowable values. The modification needed to pitch the f_0 to f_{target} in semitones is derived from the following expression:

$$n_{\text{steps}} = 12 \times \log_2 \left(\frac{f_{\text{target}}}{f_0} \right)$$

where f_{target} is the target pitch, we will use the standard frequencies in this research , and f_0 is the intoned value in a given frame. Where it is not possible to change the amplitude of the volume on a per frame basis, root mean square (RMS) of the frame after the processing is done is also produced in order to support achieving a certain sound level constant. The RMS definition is provided below.

$$\text{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} x_i^2}$$

where x_i refers to the individual sample values in the frame and N is the total count of samples. This ensures that every single frame can be pitched to the required level while maintaining the overall loudness of the output, and therefore ensuring that the accuracy of tuning and the quality of the audio produced is not compromised.

3.2 Frame-Based Pitch Adjustment with Machine Learning

This thesis describes a new frame-based pitch shifting technique from the point of view of deep learning, where a pitch notation is predicted and adjusted for each frame. Newer than common methods of pitch shifting, this method uses a neural network model akin to that of CREPE, which directly predicts the pitch notation for every individual frame considerably facilitating pitch adjustment for varied acoustic conditions.

3.2.1 Contribution: Pitch detection and modification in a single network

The approach presented here is different from the usual pitch shift because the presented deep learning model allows direct frame-level pitch manipulations. This modification of the method is also in contrast to CREPE, which is concerned with pitch measurements only, in that it is now possible to alter the frames in a single pipeline without going through any pitch modifications in between. The model thus avoids the burdensome preprocessing and feature design stages as it is trained directly on the inputs and outputs to be transformed in order to reach the specific pitches. This feature increases the usefulness of the technique as it can be applied in real-time situations with minimal hassle.

This study introduces two incremental contributions within the pitch adjustment framework. The first contribution is the use of deep learning to predict pitch notation for each frame, allowing for precise pitch alignment without conventional pitch-shifting algorithms. The second contribution builds upon this by extending the deep learning model to directly predict the modified frames, creating a one-step adjustment process. This shift from pitch detection to pitch transformation leverages CNN-based architectures and the preliminary exploration of LSTM layers to capture both temporal and frequency patterns, presenting a more cohesive and efficient solution for dynamic audio environments.

The CREPE model is based on previous works in pitch detection, e.g. the work of Kim et al. (2021) that proposed the use of deep learning for pitch estimation but did not include the CQT for the optimal pitch detection. Combined with these methodologies, CREPE improves not only accuracy but also extends the range of using pitch detection technologies wider.

3.3 Architecture of the CREPE Model

The following sub-section reveals the detailed architecture of the CREPE model with focus onto each of its component and the way it works to perform pitch detection.

3.3.1 Pitch Detection Process in CREPE

1. **Audio Preprocessing:** The audio is segmented into fixed-length frames (e.g., 2048 samples) with overlapping windows to maintain continuity during pitch detection.

2. **CQT Transformation:** Perform a Constant-Q Transform (CQT) on Confidence and Post-processing:Confidence and Post-processing: each audio frame to extract frequency-domain features.

3. **Model Input:** Feed the CQT features into the CNN. If raw waveform data is also used, combine the CQT features and waveform data before inputting them into the model.

4. **Prediction Output:** The model maps the extracted features to pitch classes through fully connected layers and outputs a softmax probability distribution, where each class corresponds to a specific pitch frequency.

5. **Confidence and Post-processing:** The class with the highest probability is selected as the pitch prediction for the current frame. A confidence score is generated to indicate the reliability of the prediction. Median filtering is applied to stabilize the pitch sequence.

6. **Final Output:** The predicted pitch class is mapped to its frequency in Hz, producing a time-series of pitch frequencies with confidence scores.

3.3.2 Constant-Q Transform (CQT)

The CQT is a time frequency analysis tool that offers a logarithmic frequency scale that is more natural than the linear scales.

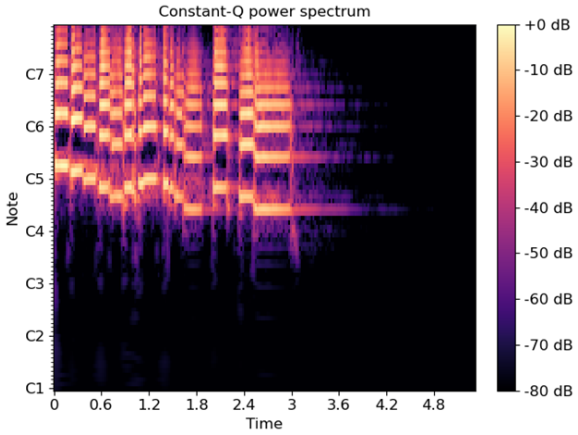

Figure 3.1: Constant-Q Spectrum

The CQT is defined as:

$$X(q,k) = \sum_{n=0}^{N-1} x[n]w[n-k]e^{-j\frac{2\pi q}{Q}n}$$

where:

- X(q, k) is the CQT coefficient at frequency bin q and time bin k,
- x[n] is the input audio signal,
- w[n] is the window function, and
- Q is the quality factor that determines the frequency resolution.

The output of the taken CQT is a matrix with the energy as shown in **Figure 3.1** [15] of the given signal at different frequencies during the time, in which this matrix is the input to the CNN.

3.4 Processing Architecture of Auto-Tuning and Training

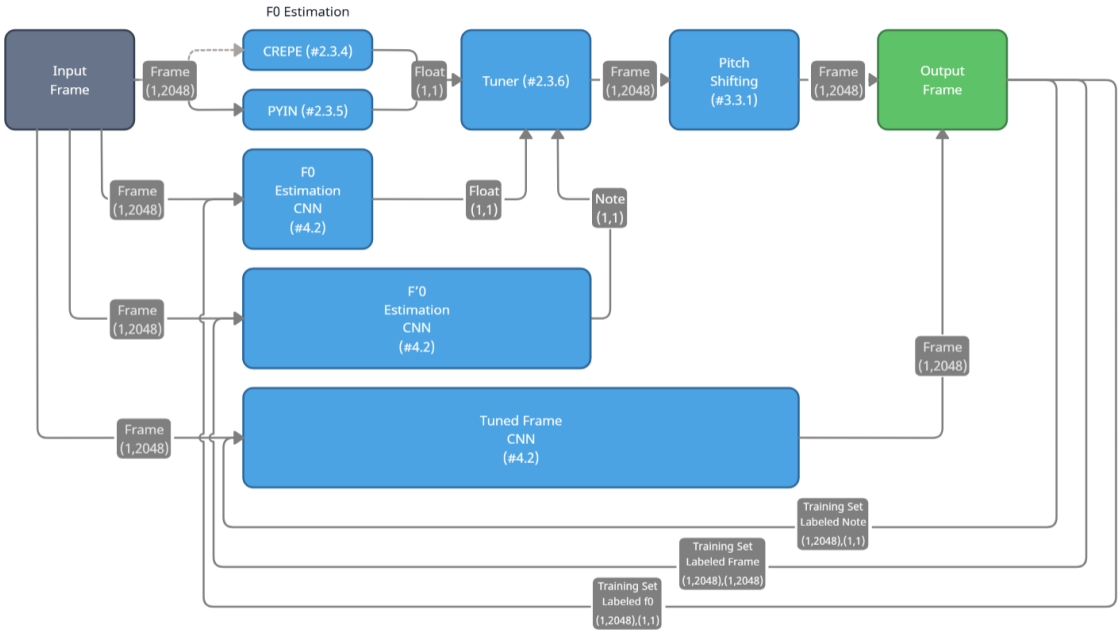

Figure 3.2: Processing Architecture

Figure 3.2 shows the entire auto-tuning process and explains how the resulting outputs are utilized to train new CNN models. The diagram includes each step of the process along with references to the corresponding sections of the thesis. Connections between the stages indicate the type and specifications of the outputs generated at each phase.

The top section represents the auto-tuning pipeline: starting with the input audio frame, the pipeline uses either PYIN or CREPE to detect the fundamental frequency (f_0). This is followed by auto-tuning, which adjusts the detected f_0 to determine the target frequency (f_0'), representing the desired pitch. Subsequently, the pitch shift stage modifies the audio frame accordingly, and the final tuned frame is produced as the output.

In this process, f_0' estimation consumes the majority of computational resources and time. Therefore, two approaches are proposed to improve efficiency:

1. **Direct prediction of the target f_0':** Using a CNN model which takes a frame

with 2048 samples then predicts the corresponding note and then converts it into the target frequency.

2. **Direct prediction of the modified frame** Using a CNN model that takes a frame with 2048 samples as input, directly predicting the adjusted frame without the explicit pitch modification step.

Additionally, for comparative purposes during experimentation, we design a CNN model to predict f_0. This model functions similarly to CREPE but simplifies the process by removing the CQT-related operations, reducing computational overhead.

During the workflow, we record the intermediate results, including the detected f_0, the corresponding f_0', the associated musical note, and the modified frame. These outputs are systematically packaged into datasets. These datasets are subsequently used to train CNN models, such as predicting target frequencies or directly generating tuned audio frames.

Chapter 4

Methodology

This chapter outlines the experimental methodology used to evaluate the proposed approach. It covers the **Datasets** used, detailing varied audio samples; the **Experimental Setup**, including environment, model configuration, training process, and the pitch detection algorithm's role in auto-tuning and frame-based adjustments; and the **Evaluation Metrics**, where pitch accuracy is assessed through listening tests and visual analysis, with basic auto-tuning as a baseline for comparison.

4.1 Datasets

The DAMP-S-AG dataset used in this research study is public available on the Zenodo website and is used in this thesis for pitch detection, modification and auto-tuning. This dataset consists of various performance of the song Amazing Grace recorded using the mobile phone karaoke application Smule.

The dataset encompasses of vocal recordings: Thus, the recordings include different singers and have slight pitch and time differences from a canonical performance of Amazing Grace. Because mobile phones are used for recording, the background noised in each recording is typical of situations in which traditional pitch detection methods can fail.

The dataset used for subsequent training is based on frame processing. Specifically, audio data is first sampled by selecting several audio files and extracting five 20-second segments from each. Each segment is then further divided into individual frames. To maintain continuity and reduce gaps between frames, an overlap-add technique, similar to the basic method mentioned previously, is applied. For each frame, the

fundamental frequency (f_0) is computed, matched to the desired target pitch, the pitch is modified accordingly, and the processed frame is subsequently recorded and saved.

This dataset ensures that the CNN model is well trained and tested under various conditions. The dataset is accessible at the following link:

https://zenodo.org/records/3596940https://zenodo.org/records/3596940.

4.2 Experimental Setup

The process of testing the CNN and LSTM models implied an accurate experimental setup to provide validity and repeatability of the procedure. The following components were integral to the setup:

4.2.1 Environment

The experiments were performed in a controlled environment using Google Colab platform that suits computing needs for deep learning tasks. The Colab notebook utilized for implementation is accessible at: https://colab.research.google.com/drive/1HADErYaCi0pF8cDAfUWxbDsi8eoFbctP?usp=sharing#scrollTo=XThuzatosqZF.

4.2.2 Model Configuration

The CNN model architecture featured the following principal parameters:

- Input Layer: The model takes input in the form of reshaped audio frames, preparing the data for efficient processing by the convolutional layers.
- Convolutional Layers: A sequential architecture with different filter sizes is applied to capture features at multiple levels of detail from the audio frames, allowing the model to recognize important audio patterns for pitch detection.
- Activation Functions: The ReLU (Rectified Linear Unit) function is selected and used to introduce non-linearity into the model, enhancing its ability to capture complex patterns essential for distinguishing pitch variations.
- Output Layer: The proposed model generates estimated pitch values, which serve as the basis for comparison with the actual pitch for evaluation purposes.

The preliminary blueprint serves as a primary structure for the model but may be modified where applicable to meet particular input-output specifications thereby accommodating various tasks of audio processing. In contrast to that a single layer LSTM network is employed in isolation to capture the temporal dependencies of the audio frames hence improving pitch detection by utilizing the available data in a sequential manner. This initial configuration forms the foundation of the model, but adjustments will be made as necessary to align with specific input and output requirements, ensuring adaptability for different audio processing tasks.

4.2.3 Training Process

The modeling was accomplished with the use of a variety of supervised learning techniques and a plethora of audio samples that were annotated by people. The training process involved optimizing the following loss function:

$$\text{Loss} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

where the observed variable is y_i and is compared to the predicted variable indicated by $\hat{y}_i$ in the case of the current paper for each sample in the dataset.

In addition, the Relative Squared Error (RSE) is applied to assess the model's performance relative to the variance of the data:

$$\text{RSE} = \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}$$

where $\bar{y}$ denotes the mean of the observed values. This function captures the model's error relative to the variability of the target data.

4.2.4 Pitch Detection Algorithm

The fundamental frequency ($f0$) dataset is formulated using the pitch detection algorithm that also generates pyin model output. The output of a pyin model gives first f_0 estimates which are important in determining the pitch accuracy at different parameters.

4.2.5 Pitch Modification

Pitch modification is an essential step that adjusts the frequency of an audio signal to match a target pitch f_0'. Two critical components are **hop size** and **phase adjustment**:

- **Hop Size**: Defines the overlap between adjacent frames. A smaller hop size ensures smoother transitions in the modified signal, crucial for maintaining audio continuity during adjustments.
- **Phase Adjustment**: Aligns the phase across frames to avoid artifacts or distortion, ensuring that the modified signal retains its natural quality.

This process works alongside pitch detection and pitch shift techniques to enable precise auto-tuning and accurate pitch correction.

4.2.6 Feature Vector

The feature vector consists of four parts: f_0, the corresponding target pitch, the original audio frame, and the modified audio frame. Each frame contains 1024 audio samples. For the experiments, the datasets are divided into three major groups, each serving specific experimental purposes:

Group 1: Dataset Size and Its effects on Training

This group consists of three sub-datasets, each varying in size to explore the effect of dataset volume on the model's training performance:

- **Small Dataset**: Contains 21,450 frames, this dataset includes F0, the corresponding note, the original frame, and the modified frame. These frames were extracted from 5 tracks randomly selected from the DAMP Amazing Grace dataset.
- **Medium Dataset**: An expanded version of the small dataset, includes 85,800 frames extracted from 20 different audio tracks of Amazing Grace. The frames retain the same structure, providing additional volume.
- **Large Dataset**: This dataset consists of 175,890 frames extracted from 40 unique audio recordings of Amazing Grace.

The goal of forming these groups is to investigate the relationship between dataset size and the resulting training outcomes.

Group 2: Dataset with Modified Frames (nsteps Adjustments)

This group comprises two sub-datasets that focus on the effects of modifying the `nsteps` parameter during frame adjustments. Unlike the original dataset, where frames were modified using their native `nsteps` values, these sub-datasets alter the `nsteps` values to introduce variations in pitch correction:

- **Nsteps +1 Dataset**: Includes 85,800 frames modified with `nsteps +1` adjustments, representing a higher pitch shift compared to the original values.
- **Nsteps -1 Dataset**: Comprising 85,800 frames modified with `nsteps -1` adjustments, this dataset applies a downward pitch shift relative to the original.

The frames in this group are sourced from 20 randomly selected performances of Amazing Grace. The main goal of this group is to assess how variations in the training dataset influence the model's predictions, particularly in relation to pitch adjustments.

Group 3: Dataset by Vocal Characteristics

This group is designed to explore how the vocal characteristics of training data—specifically gender-based differences—affect the model's performance. It consists of three sub-datasets, each containing 42,900 frames, with F0, note, original frame, and modified frame components:

- **Male-Only Dataset**: Contains frames sourced exclusively from male voices, providing a dataset tailored to male vocal characteristics.
- **Female-Only Dataset**: Includes frames sourced exclusively from female voices, offering a complementary focus on female vocal characteristics.
- **Mixed-Gender Dataset**: Combines male and female voices in equal proportions to create a balanced dataset.

The primary goal of this group is to evaluate the influence of gender-specific training data on the model's predictions and generalization capabilities.

4.3 Evaluation Metrics

In order to evaluate the performance of the models accurately, the following evaluation metrics were used:

4.3.1 Frame-to-Frame Prediction:

The evaluation criteria primarily used in the assessment of the pitch detection performance is, simply put, the actual pitch values and the predicted pitch values difference. The Gross Pitch Error (GPE) is determined as follows:

$$\text{GPE} = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100$$

where N denotes the total number of pitch frames.

4.3.2 Frame-to-Note Prediction

Accuracy: For frame-to-note prediction, accuracy measures the percentage of frames where the predicted note matches the actual note. This provides a clear measure of the model's effectiveness in categorizing each frame into the correct note, making it well-suited for evaluating overall performance in note prediction.

Chapter 5

Results and Evaluation

This chapter reports the findings of the experiments.

5.1 Basic F0 Estimation

The accuracy of pitch detection of the CREPE pitch model was assessed also against the PYIN pitch model. According to experimental results, CREPE demonstrates higher stabilities and accuracies in the pitch detection, in effect showing greater accuracy in the detection of pitch in the presence of surrounding noise or congestion.

Contained within Figure 5.1 is the comparison of pitch detection performance between the PYIN and CREPE models, both applied to the same 20-second audio sample. From the figure, it is shown that in regions where the two algorithms align closely, the primary frequencies predominantly range between 150 Hz and 250 Hz. However, notable differences in stability and sensitivity emerge upon closer inspection.

PYIN exhibits instability in several regions, failing to detect the f_0 and outputting values of zero in many instances. Thus PYIN's sensitivity to noise, which significantly affects its reliability in such contexts. On the other hand, CREPE consistently provides values across the pitch track, highlighting its ability to maintain stable even in the presence of noise. Nevertheless, certain segments display jumps in the estimation, indicating that CREPE can oversensitive for noise in the audio.

This comparison highlights the trade-offs between the two methods. While PYIN's sensitivity to noise limits its reliability in less controlled settings, CREPE provides more consistent detection but occasionally introduces artifacts due to its responsiveness to noise. These findings demonstrate CREPE's effectiveness for continuous pitch

estimation, with minor challenges in handling noise sensitivity.

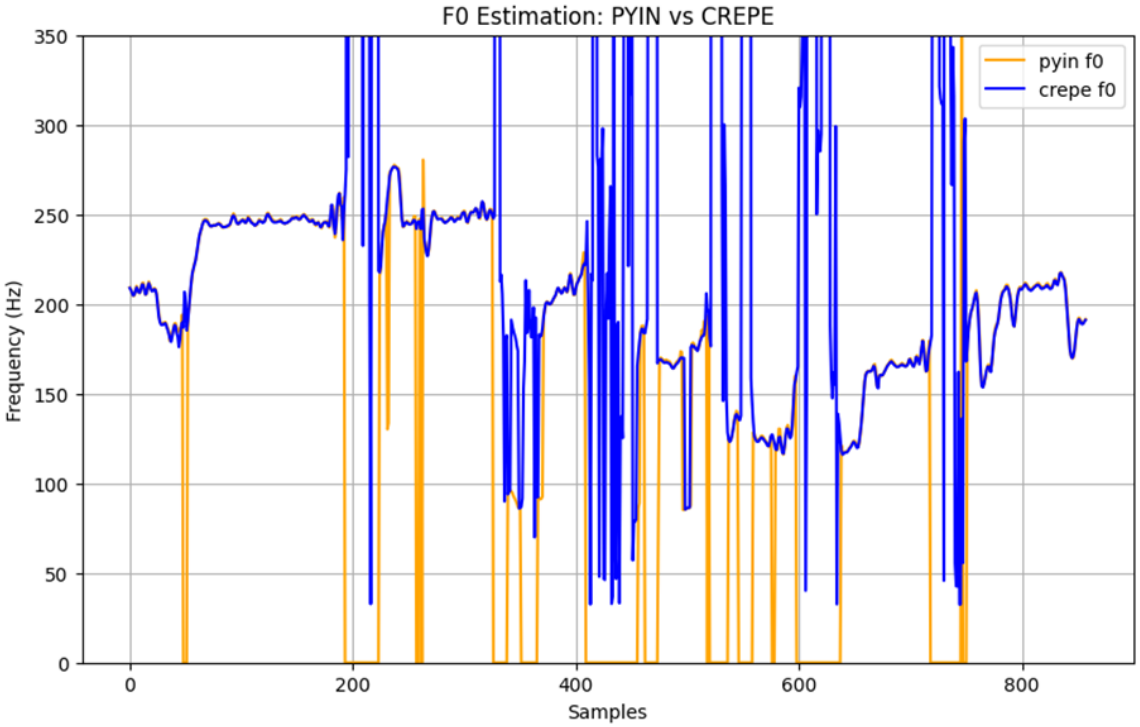

Figure 5.1: PYIN vs CREPE

5.2 Basic Auto Tuning

In the Basic Auto Tuning experiment which uses similar explained in section 3.4, the audio is cut into equal length frames at first. Then, every frame is analyzed in terms of its fundamental frequency (F0) using the PYIN algorithm. Given the found F0, the one closest to the found pitch is found, and the corresponding pitch shift (nsteps) is found. Finally, the frames are pitch shifted to achieve the intended tuning.

Four control groups were formed for this experiment. All the groups employ overlapping frames to create smooth transitions between frames hence minimizing gaps and creating a better general effect.

1. **Control Group 1**: The original, unmodified audio.

2. **Control Group 2**: This group uses only the detected F0 and the calculated pitch shift (nsteps) to adjust each frame without any additional smoothing.

3. **Control Group 3**: This group uses the raw F0 values in combination with the RMS (Root Mean Square) method, which helps reduce sudden jumps between frames and provides a smoother transition.

4. **Control Group 4**: This group builds on Group 2 by applying a median filter to smooth the F0 values and using the RMS method, further reducing abrupt jumps and enhancing overall smoothness.

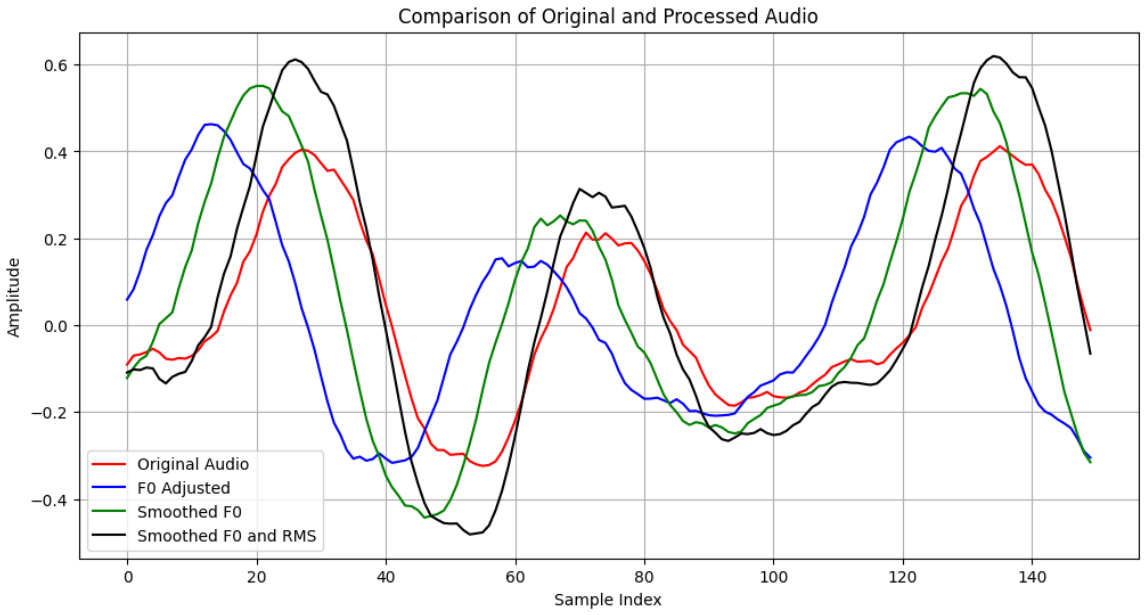

Figure 5.2: Comparison of Original and Processed Audio

In **Figure 5.2**, we compare the original audio with three processed versions to assess the effects of different auto-tuning methods:

1. **Original Audio (Red)**: This serves as the baseline for comparison.

2. **F0 Adjusted (Blue)**: This method adjusts the audio based on raw F0 and nsteps without smoothing. It introduces a noticeable "electronic" sound with abrupt transitions, making the audio sound less natural compared to the original.

3. **Smoothed F0 (Green)**: Using a median filter on F0 reduces these electronic artifacts and smooths transitions, though sections with large pitch changes can still sound slightly unnatural.

4. **Smoothed F0 and RMS (Black)**: Combining F0 smoothing with RMS adjustment produces the most natural-sounding output, effectively minimizing electronic noise and providing smooth transitions that closely resemble the original audio.

During pitch adjustment, the frequency of the waveform changes to match the target frequency. This process shifts the waveform forward or backward along the time axis to align with the desired pitch, altering its shape. This change ensures the audio signal matches the target frequency but causes it to no longer perfectly overlap with the original waveform.

As shown in **Figure 5.2**, using only F0 adjustment (blue) can introduce electronic noise, while RMS adjustment alone (black) reduces some artifacts but still lacks smoothness in rapid transitions. The **Smoothed F0 and RMS** approach (black) achieves the best results, balancing accurate pitch correction with natural, seamless transitions, making it the optimal method for auto-tuning.

5.3 F0 Prediction With CNN

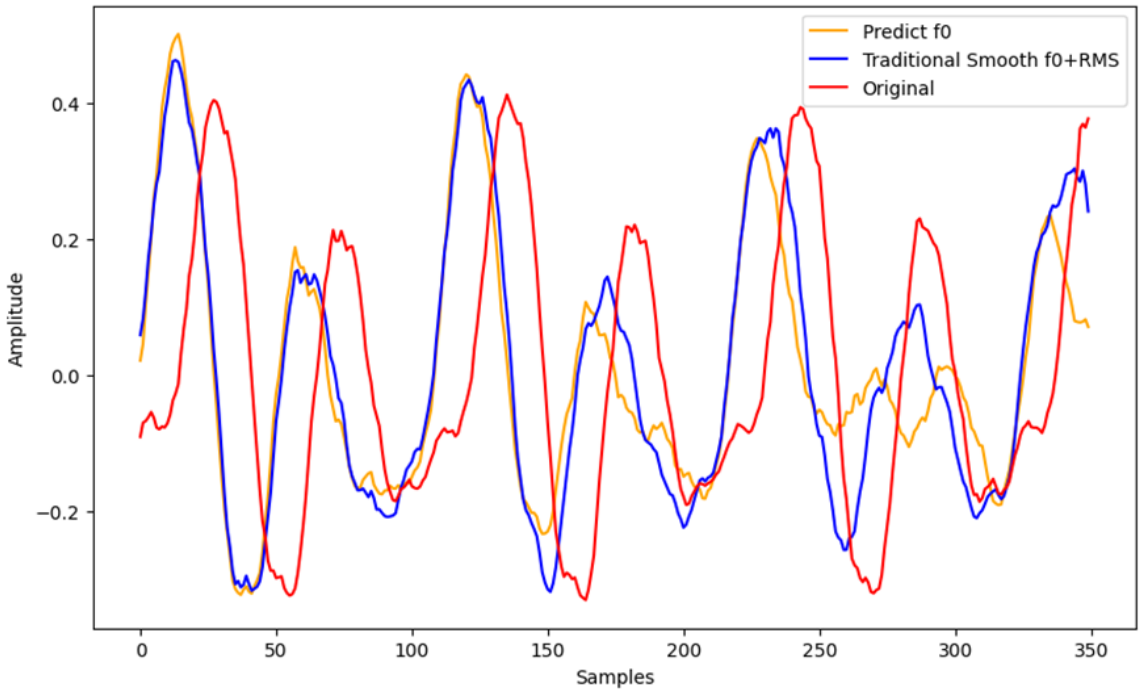

Figure 5.3: Predict F0 vs Traditional vs Original

In this section, the CNN is used to predict f_0 as a replacement for PYIN and CREPE. As shown in **Figure 5.3**, the red line represents the original audio, the blue line represents the traditional method, and the orange line represents the CNN-based prediction. In this comparison, aside from using different methods for f_0 estimation, all other steps were kept consistent. The results demonstrate that using CNN alone for f_0 prediction yields a relatively close match, with significant of overlap. While the absence of CQT in the CNN approach slightly affects accuracy compared to CREPE, the results confirm that CNNs are indeed effective for f0 detection.

5.4 Pitch Note Prediction With CNN

This CNN model is designed to classify an input audio frame of length 2048 into one of 61 classes, each representing a standard pitch. Here's a simplified breakdown of the layers:

1. **Reshape Layer**: Prepares the 1D input for processing by convolutional layers.
2. **Convolutional Layers**: Extract features from the input frame, learning patterns relevant to pitch classification.
3. **MaxPooling Layers**: Downsample the data to reduce computation and focus on prominent features.
4. **Flatten Layer**: Converts the 2D data to 1D, preparing it for the fully connected layers.
5. **Dense Layers**: Learn higher-level features and finalize the classification.
6. **Output Layer**: A softmax layer with 61 units, one for each pitch class, providing the probability for each pitch.

Since the frame-to-note method can only identify the f_0', additional steps such as pitch shifting are required for further processing. In this setup, the model takes a 2048-length audio frame as input and outputs one of the 61 classes, each corresponding to a standard pitch. Based on the classified pitch, a target frequency is determined, allowing the frame to be converted or adjusted to match this pitch.

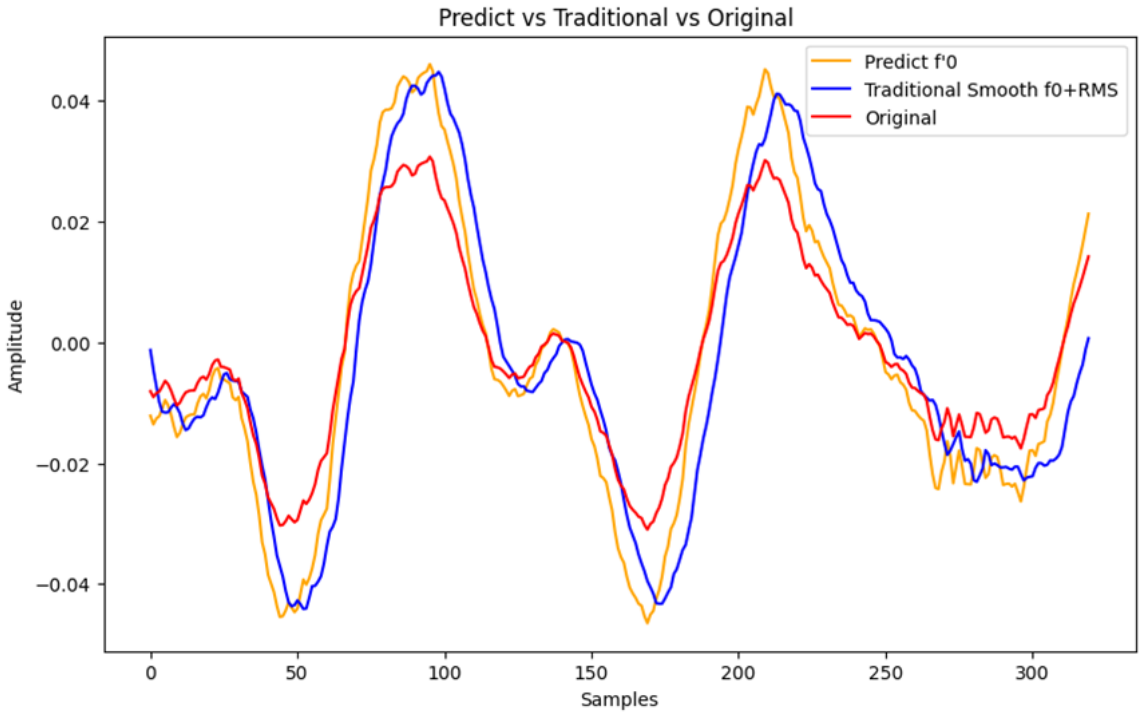

Figure 5.4: Predict vs Traditional vs Original

As shown in Figure 5.4, the red line represents the original audio data, while the blue line indicates the output from traditional pitch detection. The yellow line, representing the pitch predicted by the CNN model, closely aligns with the original pitch, demonstrating that the CNN-predicted pitch is quite similar to the traditional pitch detection result adjusted for auto-tuning. This alignment indirectly suggests that the CREPE model used for pitch estimation is fairly accurate.

Additionally, the runtime for both the CNN-based prediction and the traditional pitch detection method is comparable, making the CNN approach a viable alternative for accurate and efficient pitch estimation + tuning adjustment.

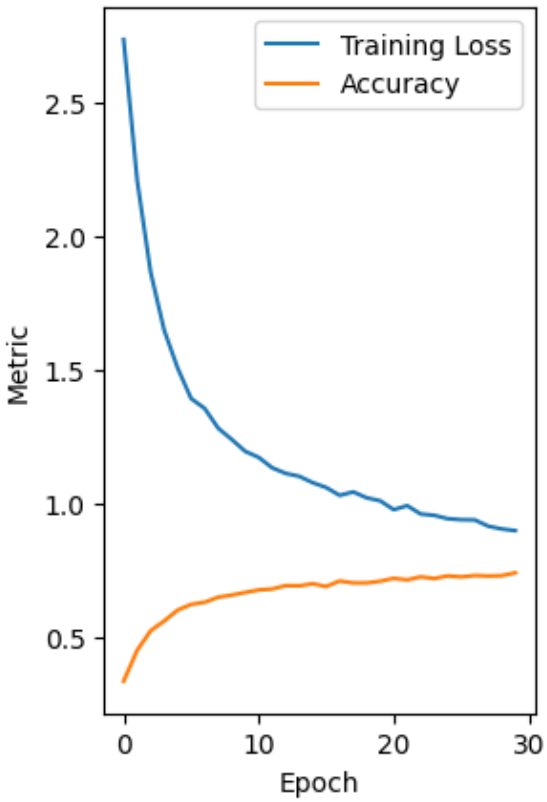

Figure 5.5: Epochs and Accuracy and Training Loss

Figure 5.5 shows a steady rise in accuracy over 30 epochs, with the model reaching around 80-85% accuracy and leveling off around 20-25 epochs. At the same time, the training loss drops quickly during the first few epochs and then flattens as the model learns, showing effective progress. The steady drop in training loss means the model is reducing errors between predictions and actual values, showing good optimization. However, the flattening of training loss after 20-25 epochs suggests limited improvements with the current setup, meaning more advanced models or better tuning may be needed to improve further.

5.5 Frame Predictions With CNN and LSTM

This section shows the results of the training and prediction.

5.5.1 Predictions vs Traditional

The CNN-predicted output (yellow) is effective in capturing the main structure of the audio, as shown in **Figure 5.6**; its quality is suboptimal. Noticeable deviations from the original waveform (red) are apparent, even on a rough visual inspection, During playback, the reproduction of the human voice lacks clarity and fidelity. These discrepancies suggest that the CNN model may not be capturing the fine details needed for a fully accurate audio reconstruction. However, the key advantage of this CNN-based approach is its significantly faster processing speed compared to CREPE, making it a more efficient option when speed is a priority despite the quality trade-offs.

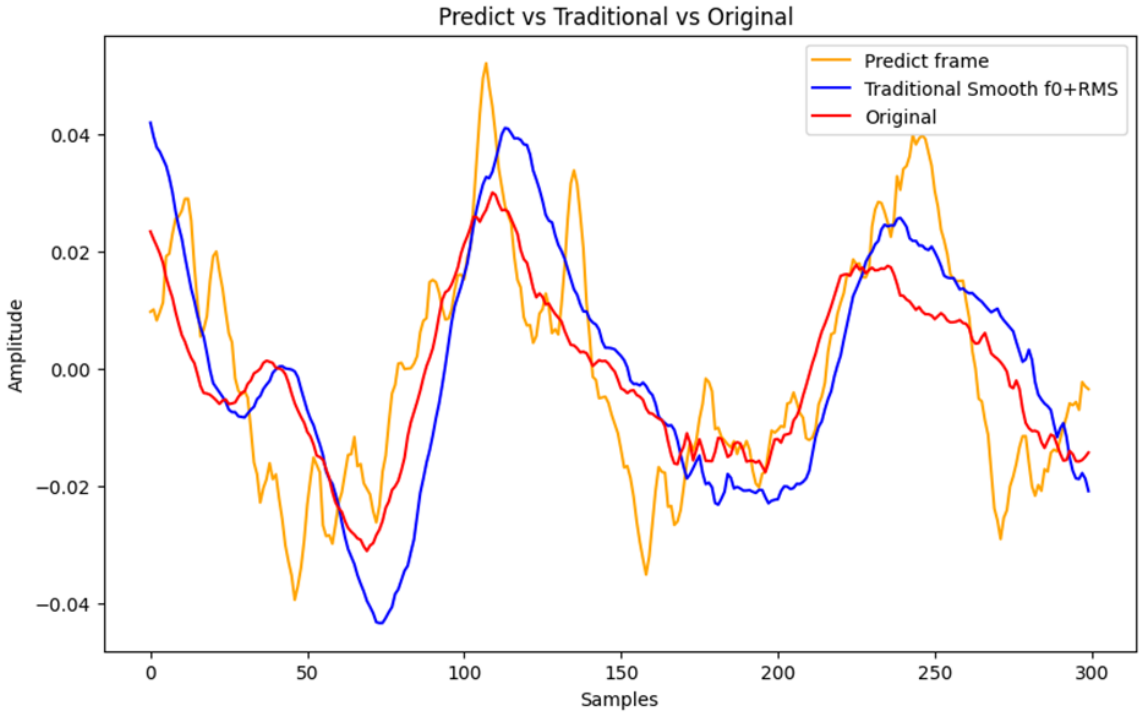

Figure 5.6: Predict Frame vs Traditional vs Original

On the other hand, Figure 5.7 shows that while the training loss decreases steadily over 20 epochs, indicating effective learning, the validation loss which is a metric used to evaluate a model's performance on a separate validation dataset during training and shows well the model generalizes to unseen data. It plateaus early around the 3rd epoch and remains relatively constant. This widening gap between training and validation loss suggests the model may start to over-fit the training data as it continues to improve on training loss without a corresponding improvement in validation

performance.

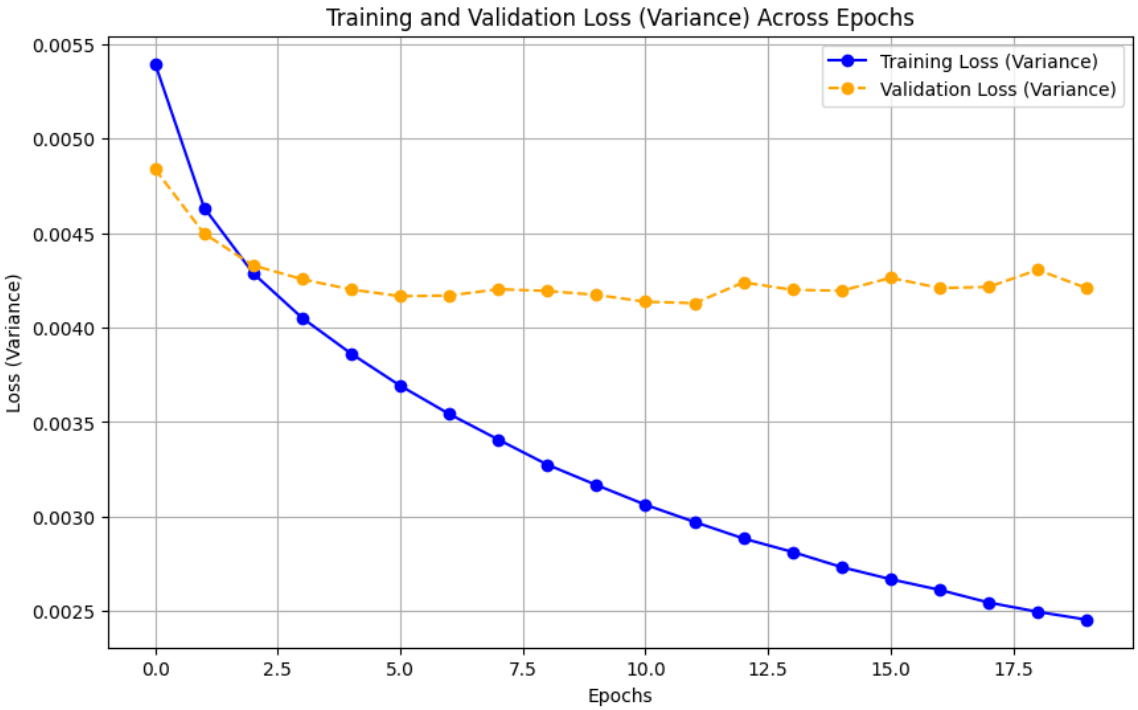

Figure 5.7: Training and Validation Loss (Variance) Across Epochs

5.5.2 Impact of Training Data Volume on Model Performance

This section analyzes the impact of dataset size on model performance by comparing training and validation loss curves. Figures 5.7 and 5.8 display the training dynamics of the same model trained on datasets of different sizes, illustrating how data quantity affects generalization.

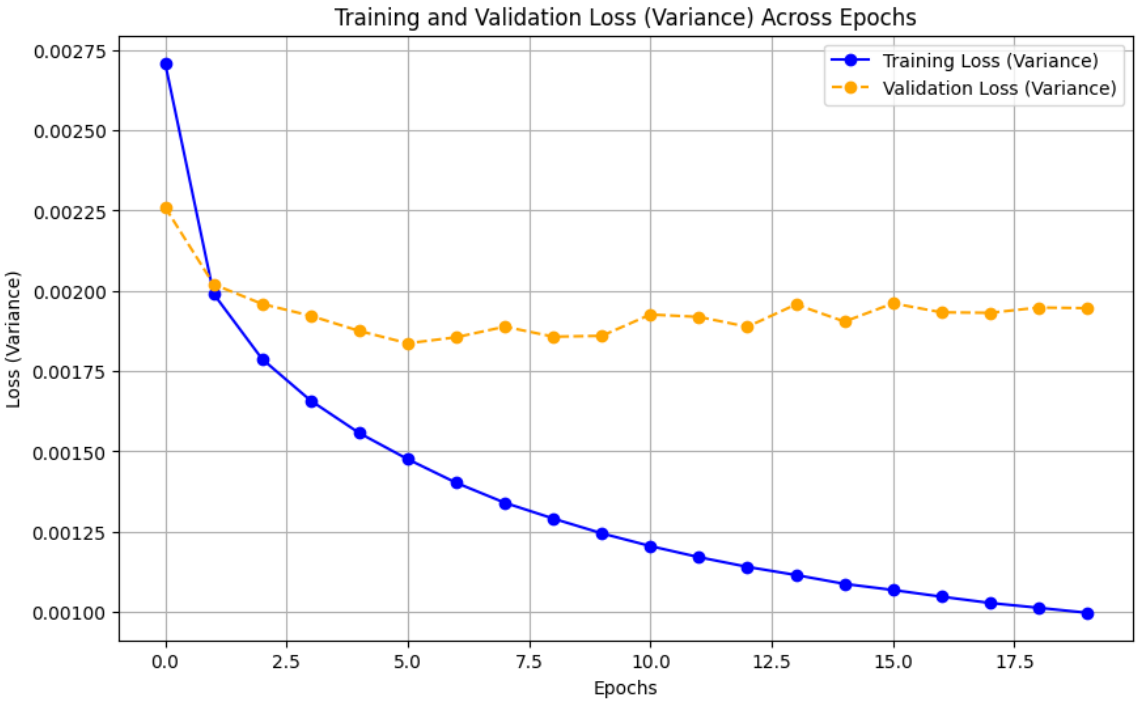

Figure 5.8: Training and Validation Loss (Variance) Across Epochs with Large Datasets

In both figures, the training loss (blue line) consistently decreases across epochs, indicating effective learning as the model reduces error on the training set. However, the validation loss (orange line) exhibits different patterns depending on dataset size. In Figure 5.7, trained on a smaller dataset, the validation loss plateaus at a higher level, signaling limited generalization and a tendency toward over-fitting. In contrast, Figure 5.8, which uses a larger dataset, shows a lower and more stable validation loss, suggesting that additional data enhances generalization and reduces over-fitting.

On the other hand, a larger dataset led to better predictive accuracy, but it also seemed to introduce more noticeable noise, resulting in a slightly degraded audio quality despite the improved accuracy.

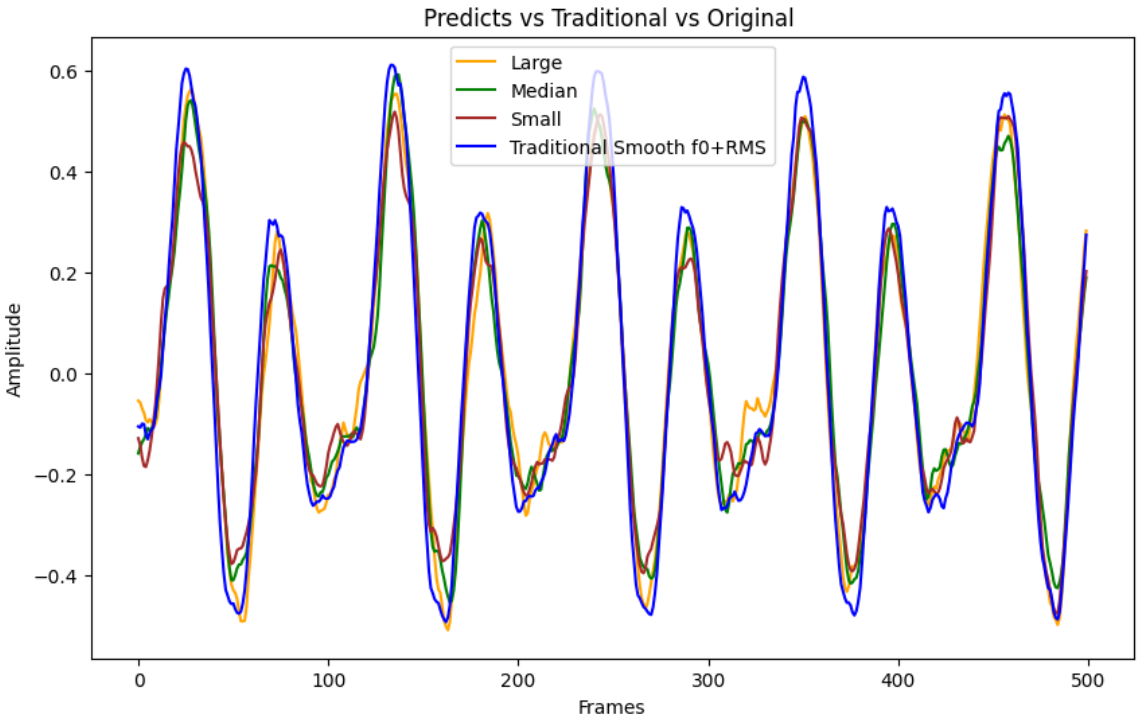

Figure 5.9: Different Sizes Datasets

As shown in **Figure 5.9**, the visual differences between the groups are not particularly pronounced. Therefore, we compare the MSE values for a more detailed analysis. The **small group** achieved an MSE of **0.0139**, the **medium group** recorded **0.0131**, and the **large group** reached **0.0129**. These subtle differences suggest that increasing the amount of training data improves the model's performance, albeit incrementally.

5.5.3 Time Consumption in Different Auto tuning Methods

This experiment evaluates the time consumption of three different autotuning methods, comparing their efficiency under identical hardware conditions. Each method was tested three times, and the average time was recorded.

1. **Pyin-Based Autotuning**: Using Pyin for F0 prediction followed by frame adjustment

2. **CREPE-Based Autotuning**: Using CREPE for F0 prediction followed by frame adjustment

3. **CNN-Based Frame Prediction**: Directly predicting the modified frames using a CNN model

Method	Test 1	Test 2	Test 3	Average Time
Pyin-Based Autotuning	1m 7s	1m 7s	1m 6s	1m 7s
CREPE-Based Autotuning	2m 40s	2m 41s	2m 40s	2m 40s
CNN-Based Frame Prediction	59s	58s	58s	58s

Table 5.1: Table of Time Consumption

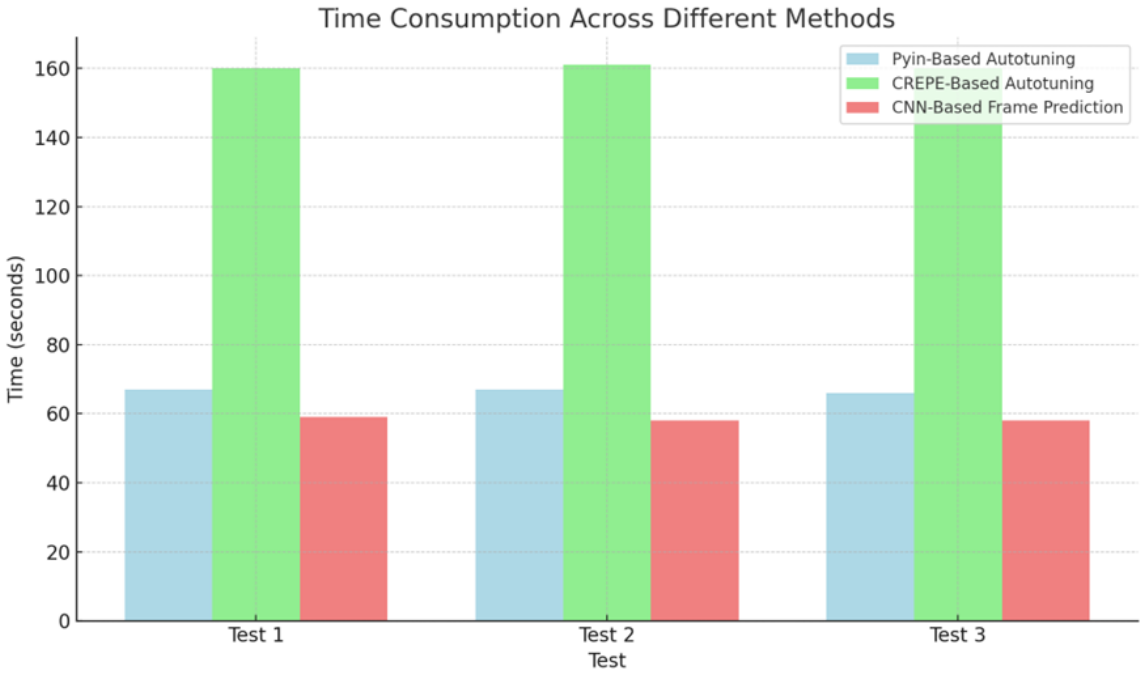

Figure 5.10: Time Consumption

The results in Figure 5.10 demonstrate that CREPE is significantly slower compared to the other two methods, particularly when no specialized hardware acceleration is available. While CREPE provides robust F0 detection, its computational demands make it less advantageous in scenarios requiring real-time performance. On the other hand, the CNN-based approach, by skipping the pitch modification step entirely, achieves the best time efficiency, offering a practical advantage in applications where processing speed is critical.

5.5.4 Nsteps Validation

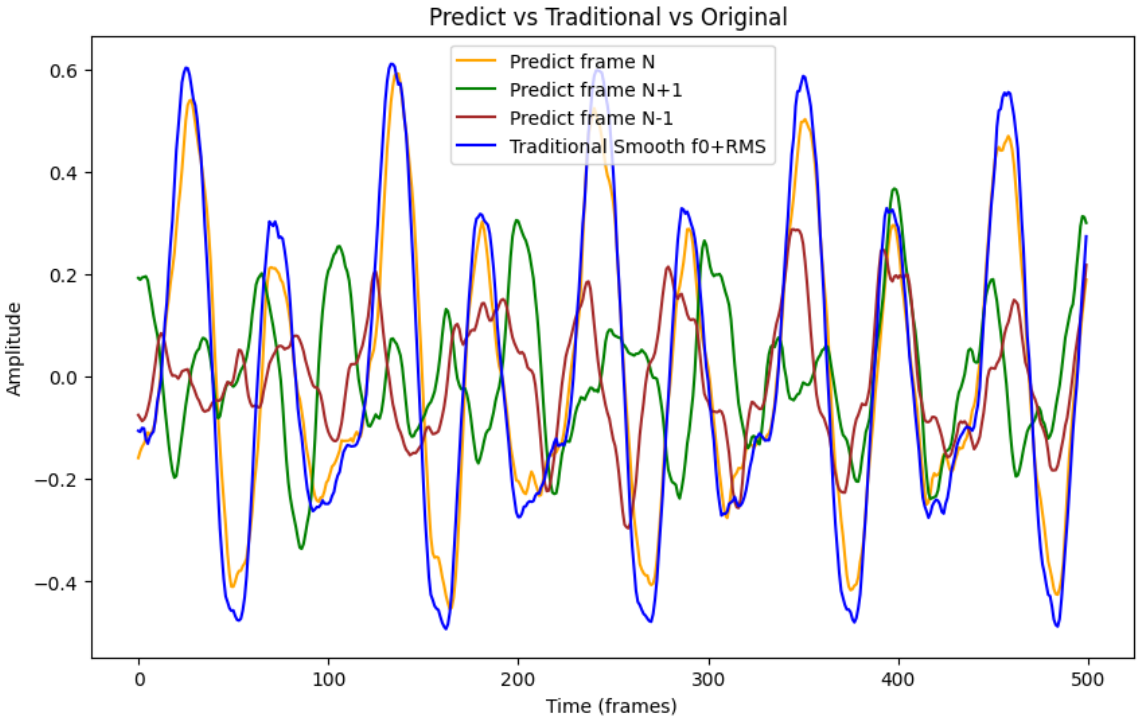

Figure 5.11: Nsteps Effects

In **Figure 5.11** shows a comparison between the original file (red) and predicted outputs using different nsteps values. The results indicate that the prediction with normal nsteps (yellow) closely aligns with the original waveform, suggesting that the model effectively captures the intended pitch. In contrast, the adjustments with nsteps + 1 (blue) and nsteps - 1 (green) introduce noticeable distortions, with nsteps + 1 producing a higher pitch and nsteps - 1 resulting in a lower pitch compared to the normal value.

This outcome demonstrates that slight adjustments in nsteps significantly affect pitch accuracy, indicating that even small biases in the training set have a corresponding impact on predictions. The effectiveness of the CNN model in capturing pitch variations highlights its suitability for this task.

5.5.5 Impact of Training Data Selection on Model Performance

In this section, the impact of dataset selection on the model's ability to generalize and effectively perform pitch correction is evaluated. Three control groups were established, each focusing on a distinct subset of the training data: male voices and female voices. The results demonstrate how the choice of training data influences the model's predictions and performance metrics.

In the first control group, all training data consisted of male voices, which were acoustically similar to the target audio for autotuning. The results, shown in **Figure 5.12** Note the sections marked with gray bidirectional arrows in the figure, indicate that the model predictions closely align with traditional pitch correction methods. The calculated Mean Squared Error (MSE) for this group was **0.0157**, suggesting the model effectively learned to generalize and apply accurate corrections based on the male training data.

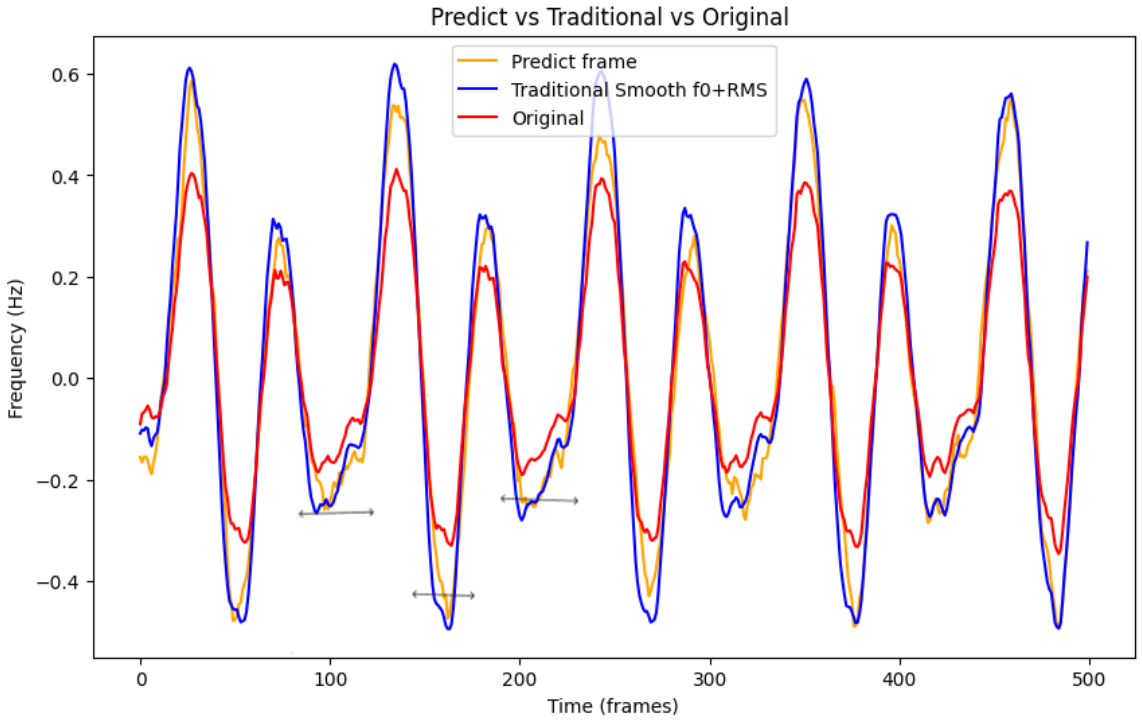

Figure 5.12: Male Dataset

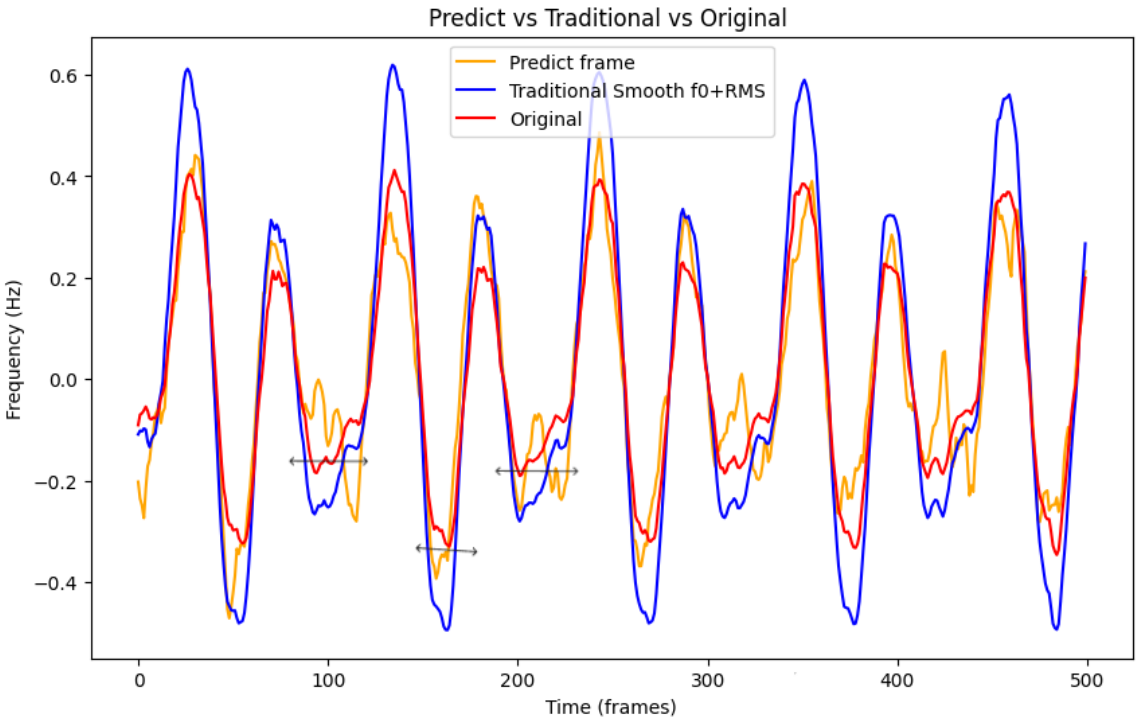

Figure 5.13: Female Dataset

In contrast, the second control group used only female voices for training. As shown in **Figure 5.13**, and the bidirectional arrows in the figure show the difference from the Figure 5.12 and 5.14. The model failed to make meaningful pitch corrections, with its predictions showing minimal deviation from the original input. The MSE for this group was **0.0292**, nearly twice that of the male training dataset. This stark difference underscores the importance of dataset selection, as mismatched training data can significantly impair model performance.

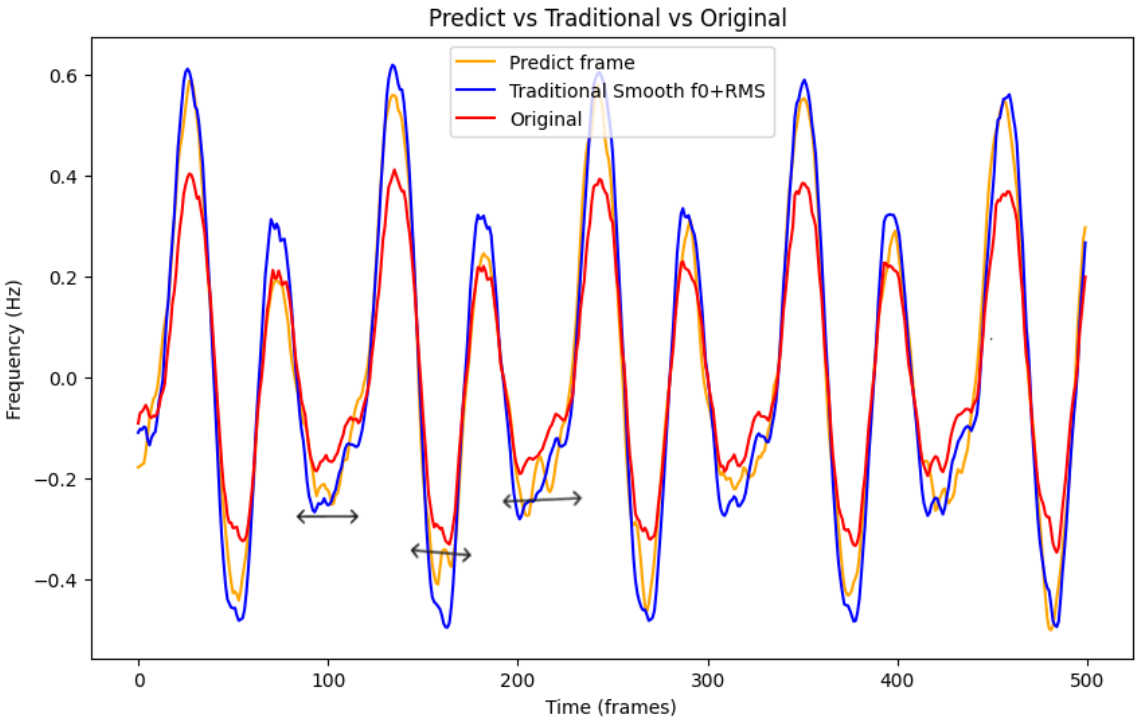

Figure 5.14: Female and Male Dataset

To further investigate, a third control group was introduced with a balanced dataset comprising equal parts male and female voices. The results, shown in **Figure 5.14**, demonstrate improved performance compared to the female-only dataset. The MSE for this group was **0.0166**, reflecting a compromise between the two extremes. While the balanced dataset improved predictions relative to the female-only dataset, it still did not match the accuracy achieved with the all-male dataset.

These results suggest that training data should closely match the characteristics of the target audio for optimal results. The male dataset gave the model patterns that fit the test audio better, leading to more accurate predictions. In contrast, the female dataset caused inconsistencies, resulting in lower performance. The MSE values clearly show that using a dataset with the same gender as the target audio improves the outcome. This is also supported by the visual results, where predictions from gender-matched datasets are closer to the target audio.

5.5.6 Challenges of Recurrent Models in Pitch Prediction

Unexpected Results With LSTM

This task is inappropriate for the application of the LSTM model because, as can be observed from the training loss, it seems to need to learn something from the data, leading to no change in the loss over training. The output is mostly meaningless noise, which indicates that the model has not been able to learn any interesting patterns. The same problem is observed in the case of the RNN model, which is also a time-stepping architecture where no improvement in loss is seen, and output is relevant but needs more clarity for the required end. The Figure 5.15 is ineffective after all.

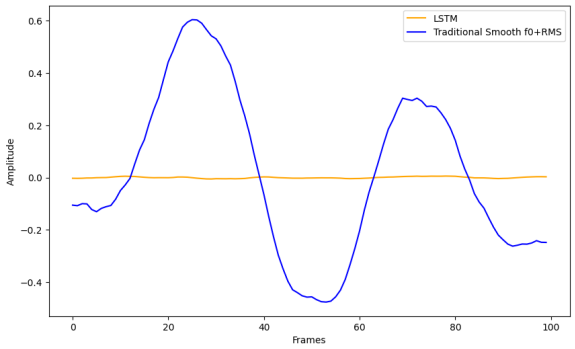

Figure 5.15: LSTM vs Traditional

Another Approach: Predict Frame with Note

In this batch of experiments, an effort was made to predict the frame and the F0 simultaneously and use the subsequently predicted F0 to modify the frame. The results, however, could have been more impressive since the method created excessive noise. The adjustment process increased the noise level within the adjusted frames of prediction, thereby producing a less coherent output, which consequently degraded the quality of the sound.

5.6 Result Conclusions

Table 5.2 highlights the performance of different datasets and configurations in terms of MSE and accuracy. Larger datasets consistently improve performance, as evidenced by decreasing MSE values. Gender-specific datasets reveal that male-only data performs better than female-only, with a balanced dataset showing intermediate results.

Nstep adjustments result in higher MSE, indicating a negative impact on prediction quality. For f_0' and f_0 predictions, the accuracy remains low at 11.81%, with MSE values indicating room for improvement. suggests that the model performs better in terms of overall numeric closeness. Overall, the findings emphasize the importance of dataset size, characteristics, and targeted adjustments for enhancing model performance.

	MSE	Accuracy
Small-size	0.0139	-
Median-Size	0.0131	-
Large-Size	0.0129	-
Nstep+1	0.0967	-
Nstep-1	0.0973	-
Male	0.0152	-
Female	0.0290	-
Half-M&F	0.0160	-
F0	0.0985	-
F'0	0.1012	11.81%

Table 5.2: Results

Chapter 6

Conclusions and Future Work

This chapter integrates the results of a variety of experiments that have been carried out in order to investigate the same problem, model and methods in pitch detection and frame prediction. The experiments mainly consisted of audio frame and pitch prediction using CNNs, LSTMs, etc. and their comparisons with the other methods. Moreover, this chapter suggests possible avenues for future work in order to improve pitch detection and frame prediction models.

6.1 Summary of Key Findings

The experiments provided valuable insights into the strengths and limitations of various models and methods used in pitch detection and frame prediction. Key findings include:

1. **Effectiveness of CNN-based Autotuning**: The CNN model demonstrated a reasonable ability to capture the main structure of the audio signal, albeit with some quality issues in clarity. While the CNN approach was not perfect, it was much faster than CREPE, making it an efficient choice for real-time applications where speed is essential (Figure 5.6).

2. **Impact of Dataset Size**: Increasing the dataset size led to improved predictive accuracy and better generalization, as evidenced by lower and more stable validation loss in the larger dataset experiment (Figure 5.8). However, using a larger dataset also appeared to introduce slightly more noise, potentially affecting the audio quality despite the enhanced accuracy.

3. **Sensitivity to Nsteps Adjustments**: The experiment on varying nsteps values highlighted the sensitivity of pitch prediction to small parameter changes. Using nsteps + 1 and nsteps - 1 caused noticeable pitch deviations, with higher and lower pitch shifts, respectively, compared to the original (Figure 5.9). This outcome suggests that the CNN model is effective in capturing pitch but is also susceptible to small biases in training data.

4. **Challenges with Recurrent Models**: LSTM and RNN models did not perform well in this task, as indicated by the lack of change in loss during training. Both models produced incomprehensible noise as output, failing to capture meaningful patterns for pitch prediction. This indicates that recurrent architectures may be unsuitable for frame-based pitch detection in this context.

5. **Simultaneous Frame and F0 Prediction**: Attempting to predict both the frame and F0 simultaneously was not successful, as it introduced excessive noise and amplified distortions in the predicted frames. This approach yielded little audio output, suggesting that separate prediction methods may be more effective.

These experiments demonstrate the effectiveness of CNNs in certain aspects of pitch prediction and frame generation but also highlight the challenges faced when using recurrent models and combined prediction methods. The findings provide a foundation for further exploration into improving model performance and addressing limitations.

6.2 Future Research Directions

While the results of this study have been positive, one can also indicate a number of possible directions for future work. Possible directions are:

- **Expanding the Dataset**: Using a larger and well-built dataset – for instance, one that classifies the voice as that of a man or woman or even using standard frequencies made synthetically – could improve the accuracy of the model. The actual results are not satisfactory, but it is plausible that the reorganized design of the sound, which incorporates some elements of noise, will, in due course, improve the quality and intelligibility of the produced audio.

- **Adopting More Complex Model Structures:** We can integrate additional sophisticated model designs. For instance, extensive incorporation of more layers in convolutional networks (CNN), addition of Transformers, or application of hybrid systems will enhance the models' understanding of sound content, helping produce accurate predictions of pitch even for complex content. Such models might have better generalization abilities and lower the amount of distortion present in the produced output.

- **Improving the Existing Model** : It might be that seeking to apply various other loss functions could deal with the issue of forecasting that is plagued with noise. In the pursuit of model fitting, the application of different types of loss functions may help reduce noise and improve overall noise performance, including different factors of the forward process and the outcome at the end of training. To conclude, such perspectives, in turn, are quite encouraging in further enhancements of the model targeting the pitch accuracy as well as audio quality processing aspects that are based on this study. In conclusion, these future directions offer promising paths for improving the model's ability to accurately capture pitch and enhance audio processing quality, building on the foundations established in this study.

Bibliography

[1] Meihui Ba, Zhongzhe Li, and Jian Kang. The multisensory environmental evaluations of sound and odour in urban public open spaces. *Environment and Planning B: Urban Analytics and City Science*, 50(7):1759–1774, 2023.

[2] Vivek Bhardwaj and Vinay Kukreja. Effect of pitch enhancement in punjabi children's speech recognition system under disparate acoustic conditions. *Applied Acoustics*, 177:107918, 2021.

[3] Francesc Busquet, Fotis Efthymiou, and Christian Hildebrand. Voice analytics in the wild: Validity and predictive accuracy of common audio-recording devices. *Behavior Research Methods*, 56(3):2114–2134, 2024.

[4] Lucille Calmon, Michael T. Schaub, and Ginestra Bianconi. Dirac signal processing of higher-order topological signals. *New Journal of Physics*, 25(9):093013, 2023.

[5] Anandhan Dhanasingh and Ingeborg Hochmair. Signal processing & audio processors. *Acta Oto-Laryngologica*, 141(sup1):106–134, 2021.

[6] T. Drugman and A. Alwan. A new approach to pitch tracking for speech and music. *IEEE Transactions on Audio, Speech, and Language Processing*, 27(8):1305–1316, 2019.

[7] Jesse Engel, Rigel Swavely, Lamtharn Hanoi Hantrakul, Adam Roberts, and Curtis Hawthorne. Self-supervised pitch detection by inverse audio synthesis. In *ICML 2020 Workshop on Self-supervision in Audio and Speech*, 2020.

[8] Omer Melih Gul, Michel Kulhandjian, Burak Kantarci, Azzedine Touazi, Cliff Ellement, and Claude D'amours. Secure industrial iot systems via rf fingerprinting under impaired channels with interference and noise. *IEEE Access*, 11:26289–26307, 2023.

[9] Ben Hayes, Jordie Shier, György Fazekas, Andrew McPherson, and Charalampos Saitis. A review of differentiable digital signal processing for music and speech synthesis. *Frontiers in Signal Processing*, 3:1284100, 2024.

[10] Reinhard Hochmuth and Jana Peters. On the analysis of mathematical practices in signal theory courses. *International Journal of Research in Undergraduate Mathematics Education*, 7(2):235–260, 2021.

[11] Y. Huang and H. Wu. Noise-robust pitch detection using adaptive filtering techniques. *IEEE Signal Processing Letters*, 29:233–237, 2022.

[12] A. Jain, M. Shahnawaz, and S. Jain. Robust pitch detection in noisy environments using deep learning. *Journal of Signal Processing Systems*, 93(3):399–409, 2021.

[13] J. Kim and S. Lee. Real-time pitch detection using the crepe model in varied acoustic conditions. *Journal of the Acoustical Society of America*, 150(6):4127–4136, 2021.

[14] T. D. Le and H. P. Shum. Enhancing pitch detection accuracy through spectral subtraction methods. *Speech Communication*, 119:80–89, 2020.

[15] Brian McFee, Colin Raffel, Dawen Liang, Daniel P. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python, 2015. Version 0.10.2.

[16] Mohaddeseh Mirbeygi, Aminollah Mahabadi, and Akbar Ranjbar. Speech and music separation approaches: A survey. *Multimedia Tools and Applications*, 81(15):21155–21197, 2022.

[17] R. A. Morrison and L. Boucher. Cepstral-based pitch detection in noise: Performance metrics and methods. *Journal of the Acoustical Society of America*, 147(5):3240–3250, 2020.

[18] Boaz Rafaely, Vladimir Tourbabin, Emanuel Habets, Zamir Ben-Hur, Hyunkook Lee, Hannes Gamper, Lior Arbel, Lachlan Birnie, Thushara Abhayapala, and Prasanga Samarasinghe. Spatial audio signal processing for binaural reproduction of recorded acoustic scenes: Review and challenges. *Acta Acustica*, 6:47, 2022.

[19] M. F. Siddiqui and Z. H. Tan. Computational challenges in real-time pitch detection using deep learning models. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4521–4525, 2021.

[20] Berrak Sisman, Junichi Yamagishi, Simon King, and Haizhou Li. An overview of voice conversion and its challenges: From statistical modeling to deep learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:132–157, 2020.

[21] Natalie Yu-Hsien Wang, Hsiao-Lan Sharon Wang, Tao-Wei Wang, Szu-Wei Fu, Xugan Lu, Hsin-Min Wang, and Yu Tsao. Improving the intelligibility of speech for simulated electric and acoustic stimulation using fully convolutional neural networks. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 29:184–195, 2020.

[22] Y. Yi, X. Wang, and Y. Yang. Convolutional neural network for pitch detection: A case study. *Journal of Audio Engineering Society*, 70(3):148–158, 2022.

[23] J. Zhang and W. Chen. Pitch detection using deep learning: A review. *Journal of Machine Learning Research*, 21(64):1–34, 2020.

[24] Haiquan Zhao and Zian Cao. Robust generalized maximum blake–zisserman total correntropy adaptive filter for generalized gaussian noise and noisy input. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2023.