WAFFL: WAVEFORM AUDIO FUNDAMENTAL FREQUENCY LEARNER

Sean Goldie* **Gregor McWilliam*** **Jack Tipper***
New York University
{goldie, gregor, tipper}@nyu.edu
*Equal contribution

ABSTRACT

Musical pitch estimation is a core task in the fields of music information retrieval (MIR) and speech analysis. In recent years, deep learning-based techniques for pitch estimation have surged to the forefront, with the convolutional approach CREPE earning its place as the current industry standard. In this paper, we introduce WAFFL, a novel and efficient machine learning method for estimating the fundamental frequency of monophonic vocal recordings. WAFFL differs from existing techniques by employing a model trained on labeled Mel-spectrogram frames of single-voice singing excerpts. Unlike the leading methods CREPE and pYIN, which function entirely in the time domain, WAFFL is trained and operates on data in the frequency domain. At its foundation, WAFFL utilizes a straightforward neural architecture centering on a multilayer perceptron (MLP). Our proposed training and data preprocessing strategy performed well in our test trials, and may prove generalizable for future research tasks in this field. We plan to offer a pretrained implementation of WAFFL as an open-access Python package to facilitate community experimentation.

1. INTRODUCTION

Pitch is a quality of musical sound wholly defined by human perception. Like frequency, pitch delineates sonic order from low to high and is measured in Hz. Fundamental frequency (f0) refers to the first and lowest harmonic in the series of stacking harmonics that make up a single-voice sound. In scientific settings, pitch is commonly conflated with fundamental frequency, as these values often coincide [1]. Although pitch and fundamental frequency are separate measurements and observing their distinctions can be important in the field of psychoacoustics, this paper will conform to the convention within pitch estimation and henceforth treat the terms as interchangeable [2].

Estimating musical pitch has been a topic of interest in acoustics since 1967, when Noll introduced the cepstrum approach for fundamental frequency prediction [3]. In the decades since, a wide variety of techniques have been proposed, building up to the handful of leading methods that are commonly used today. The YIN algorithm and its newer probabilistic version pYIN are at the forefront of computational approaches, and the convolutional method behind CREPE is the current vanguard of neural network-based techniques [4]. All of these pitch estimation systems pursue a common goal: to accurately extract the development of a monophonic audio signal's fundamental frequency value as it unfolds over time [5]. The YIN algorithm operates in the time domain, utilizing the autocorrelation function alongside a variety of error-filtering modifications [6]. The pYIN method builds on this to calculate pitch probabilities for YIN outputs and select the best candidates using a Hidden Markov model [7]. CREPE also operates in the time domain, using a deep convolutional neural network to achieve state-of-the-art results [4].

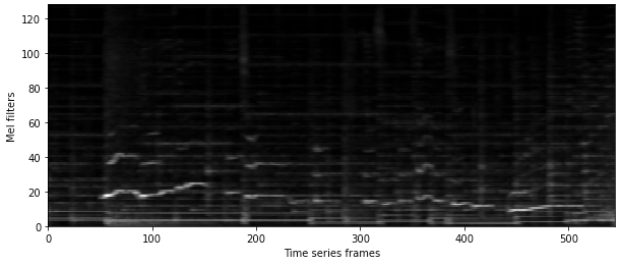

Figure 1. Mel-scaled spectrogram from MIR-1K dataset with our frame slicing displayed on the X-axis and the Mel filter bank displayed on the Y-axis.

Our WAFFL method offers unique advantages by introducing a new approach to pitch data preprocessing. We sought to reduce the training data resource load by converting the training dataset of monophonic audio files into Mel-spectrograms, thus discarding signal phase information and standardizing frequency resolution. The time series of each Mel-spectrogram is then broken into individual frames matching a corresponding frequency label. Our shallow neural network is then trained on these data pairs. This process enables rapid inference from a relatively small model. The resulting pitch estimation pipeline maps inputs to the same Mel-spectrogram space as the training data, iterates over the frames of the time series, and calculates predictions for each frame. Our training process also infers voicing data from the training labels by recognizing frames with zero pitch values as unvoiced, as this is how both of our selected training datasets delineated unvoiced frames.

2. METHOD

2.1 Data Pre-Processing

Our methodology for training WAFFL was to expose the model to isolated frames of Mel-scaled spectrogram data from monophonic audio recordings alongside their corresponding fundamental frequency labels in Hz. We tested

two different datasets for training, in addition to training on both combined; these were the MIR-1k [8] and Vocadito [9] datasets. Both datasets contain pitch-annotated audio clips of individual people singing. In MIR-1k, the annotations are given as semitone values above C0, which we converted to Hz:

$$y_{freq} = 440 \cdot 2^{(semitone-69)/12} \ . \quad (1)$$

Spectrograms emerged as an appropriate choice for our model's input data type, following our hypothesis that this format would be relatively lightweight in terms of file size, and also consistent with our goal for the machine learning model to interpret/infer pitch information. Mel-scaling was selected to calibrate the model to the perceptual curve of human hearing (i.e. pitch), consequently simplifying adaptation to a uniform frequency axis.

The two datasets used for training have different periods for their frequency label annotations (320 samples at 16kHz for MIR-1k and 256 samples at 44.1kHz for Vocadito). In order to create Mel-spectrograms with stable dimensions (number of Mel filters by number of labels) for each training sample, we truncated the time-domain signals by the remainder of their length divided by the annotation period for their respective dataset. This process was performed prior to generating the spectrogram data:

$$x'[n] = x[0, N - (N \bmod h) - 1] \ . \quad (2)$$

Next, we generated Mel-scaled spectrograms of each full-resolution sample with 128 Mel filters, to yield spectrogram data of dimensions $128 \times n$ (where n denotes the number of labels) using library functions from Librosa [10]. The spectrograms were created using the Short Time Fourier Transform (STFT):

$$X[m,k] = \sum_{n=0}^{N-1} x[n + mH] \cdot w[n] \cdot e^{-j2\pi kn/N} \ . \quad (3)$$

Mel-scaling was applied to the spectrograms as follows:

$$\mathrm{f_{mel}} = 1127.01028 \cdot \log_2(\,1 + \mathrm{f}/700\,) \ . \quad (4)$$

This order of operations allows for the maximum retention of frequency data resolution. We then compiled the sliced frames of each spectrogram and their corresponding ground-truth pitch labels for our preprocessed training dataset. This approach yielded training samples of dimensions 128×1 paired with their fundamental frequency value labels in Hz. Our data preprocessing technique is relatively efficient when compared alongside popular time-domain machine learning techniques such as CREPE and pYIN, and may prove to be generalizable for other machine and deep learning audio tasks. Moreover, this approach also allows for new inputs to the trained model to have any length and sample rate, provided the model's prediction method is wrapped in a function that calls it in a loop iterating over the input data.

2.2 Model Selection and Training

We selected an MLP as the underlying model architecture for WAFFL. The Scikit-learn [11] MLP regressor was chosen for its accessible implementation and usage. Future work could involve replication of this technique using deeper models or versions found in other machine learning libraries. The model was initialized with an input layer of 128 neurons to match the 128 Mel filters used in the data preprocessing and the resulting dimensions of the training samples. A hidden layer was added with 12 neurons for dimensionality reduction. We selected this neuron count semi-arbitrarily, because there are 12 pitches per chromatic octave in Western music, and because 12 is roughly 10% of the input dimension of 128. The output layer was a single node that predicts floating-point representations of pitch in Hz. The model trained on the pre-processed data and pitch labels, where each label corresponds to a single 128-element vector of Mel-spectrogram magnitudes.

2.3 Metrics

Three models were created in total: one trained for each individual preprocessed dataset, and one trained on the two preprocessed datasets combined. Model evaluation was performed on the same dataset as training via a standard 80/20 train-test-split, comparing against samples the model had not yet been exposed to. Evaluation metrics were gathered using the Python library mir_eval [12]. All metrics are adaptations of traditional F-scores that target different attributes of the predictions. Voicing Recall (VR) is defined as the portion of frames with non-zero pitch labels correctly predicted as having a non-zero pitch, while Voicing False Alarm (VFA) is the portion of frames incorrectly predicted as having non-zero pitch. Raw Pitch Accuracy (RPA) refers to the portion of frames correctly predicted within half a semitone of the labelled pitch in Hz, while Raw Chroma Accuracy (RCA) is the same metric mapped to a single octave of chroma. Overall Accuracy (OA) is an average of these metrics and represents the overall F-score.

2.4 Evaluation

The model that was trained on only the MIR-1k dataset vastly outperformed the others in every metric, in addition to performing the best during a round of subjective perceptual analysis after listening to output sonifications. We hypothesize that the addition of new and more diverse training data (i.e. from the Vocadito dataset) worsened the model's performance, making it excessively sensitive to small pitch fluctuations in its input signals. All models performed well with regard to Voicing Recall and Voicing False Alarm, suggesting that the training data accurately captured the difference between voiced and unvoiced frames (i.e. zero vs. non-zero energy). The best model achieved an overall accuracy score of 0.673, which bearing in mind the lightweight, shallow neural network and relatively low number of training samples (~6000 after preprocessing) we believe to be a noteable achievement.

3. RESULTS

The results obtained from the MIR-1k-trained WAFFL model are shown in Table 1 and graphed in Figure 2. Considering the approachable methodology of our preprocessing and training technique, these results were largely promising, showing a voicing recall of 0.893 and an overall accuracy of 0.673. Though these results are generally lower than existing methods such as CREPE and pYIN (which achieved overall accuracy results of 0.874 and 0.843 respectively when evaluated for the Vocadito dataset), the metrics derived from the MIR-1k-trained WAFFL model largely validate the use of Mel-spectrogram data as a training input for monophonic pitch estimation. This satisfactory result may be due to properties of the MIR-1k dataset, such as its clear separation of lead vocal and accompaniment tracks, lack of bleed, relatively large scale, and consistency of language, recording environment, and performer type.

Data	VR	VFA	RPA	RCA	OA
MIR-1K	**0.893**	**0.005**	**0.584**	**0.586**	**0.673**
Vocadito	0.722	0.010	0.230	0.232	0.293
Both	0.731	0.015	0.303	0.305	0.340

Table 1. Comparison of mir_eval metrics across WAFFL models trained on each dataset.

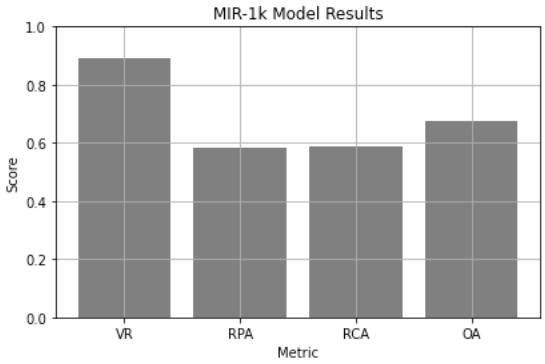

Figure 2. WAFFL MIR-1K model results.

3.1 Dataset Comparison

The evaluation of the WAFFL models and the way in which overall accuracy and other metrics were impacted by different training data further highlights the importance of dataset selection in the development of machine learning-based prediction models. While the model trained solely on the MIR-1k dataset performed well considering its specificity, all models that incorporated the Vocadito dataset were found to be significantly inferior across all measured scores.

There are many potential reasons for this decrease in key metrics, the most obvious of which is perhaps the dataset's small size relative to MIR-1k, with Vocadito providing just 40 total excerpts of monophonic singing. It should perhaps not be surprising, then, that the model performed poorly when trained on this dataset alone, as the dataset's scale may limit its effectiveness in accurately estimating the fundamental frequency of novel audio data. This is despite Vocadito's relative increase in both sample rate and time-stamp resolution, at an original 44.1 kHz and 256 samples-per-label compared to the 16 kHz and 320 samples-per-label offered by MIR-1k.

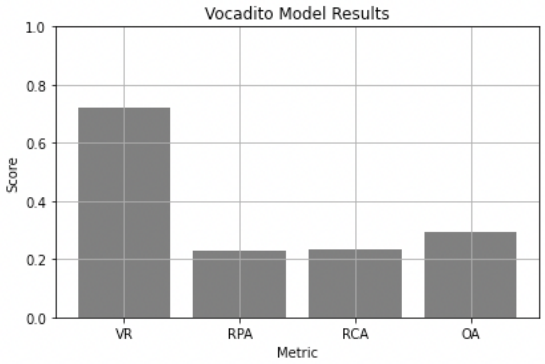

Figure 3. WAFFL Vocadito model results.

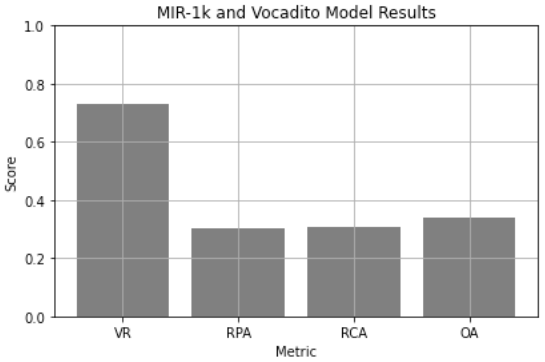

Figure 4. Results of the combined MIR-1K and Vocadito WAFFL model.

3.2 Comparison to Existing Methods

We conducted a prediction speed test by measuring the time required to generate output estimates for the entirety of Vocadito's 40-track dataset, and subsequently calculating the mean processing time required for each set of outputs. Running on the non-optimized Intel i9 CPU via which the models were compared, WAFFL's mean processing time was just 0.95 seconds. CREPE required an average of 23.90 seconds for each track—a staggering latency increase of over 2515% when compared to WAFFL. Our model also outperformed pYIN in terms of processing time, with the latter imposing a mean estimation delay of 12.98 seconds for each Vocadito audio track. Indeed, it was only the computational method YIN—the most traditional of the compared approaches—that computed the output in less time, taking a mere 0.326 seconds on average to deliver an estimation.

Despite WAFFL's subordinate scores on metrics such as overall accuracy, its adequate pitch estimation performance combined with its vastly improved inference speed indicates that it may still prove useful, and that Mel-spectrogram-focused pitch estimation models such as WAFFL could merit further research. Though a more thorough speed comparison that takes into consideration the variety of hardware systems specifically optimized for such machine learning tasks is left for future work, our investigation underlines WAFFL's efficacy as a potential candidate for low-latency monophonic pitch estimation tasks.

3.3 Subjective Evaluation

Notwithstanding the lower overall accuracy of our WAFFL models when compared alongside existing machine learning-based methods, a subjective perceptual evaluation of a variety of resynthesized outputs from the WAFFL MIR-1k model found that these sonifications accurately replicated the perceived pitch information contained within the original recording. Pitch contours were predicted for randomly selected vocal recordings between 10 and 30 seconds in length—limited as such so as to avoid listener fatigue—and then converted to audio by way of the mir_eval library's sonify() function [12]. Congruent with the metrics generated during our objective analysis, our WAFFL model trained solely on the MIR-1k dataset was deemed best suited to faithfully recreate the original recording's perceived melody contour. On the other hand, the model trained on the Vocadito dataset alone was found to provide the poorest sonified outputs, with ubiquitous perceptible frequency fluctuation errors undermining its general adherence to the pitch of the original audio.

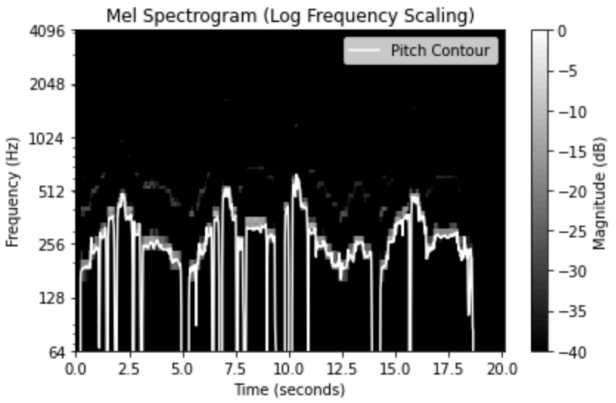

Figure 5. Inference output pitch contour from the WAFFL MIR-1K model.

4. CONCLUSION

In this paper, we presented WAFFL: a machine learning model for monophonic pitch estimation of vocal recordings. In doing so, we have validated the effectiveness of leveraging our novel approach of using Mel-spectrogram data representations to training fundamental frequency extraction models. In spite of the limited data on which this model was trained, the early results are promising, achieving accuracy scores comparable to existing methods. It is the speed of this approach, however, that sets it apart from the state-of-the-art machine learning models. WAFFL requires a fraction of the time to generate an output when compared to industry leaders CREPE and pYIN. Through future research, particularly into pre-processing methodologies or the use of WAFFL as a low-latency pitch estimation tool, we hope that our model garners utilization in user-facing applications and on mobile/embedded devices.

5. ACKNOWLEDGEMENTS

This work was conducted for our final project in the graduate-level Music Information Retrieval class at NYU Steinhardt. We would like to thank Dirk Vander Wilt, our professor, for his time and helpful feedback.

6. REFERENCES

[1] R. F. Lyon, "Human Hearing Overview," in *Human and Machine Hearing*, Cambridge University Press, 2017, pp. 46-77.

[2] B. Gfeller, C. Frank, D. Roblek, M. Sharifi, M. Tagliasacchi, and M. Velimirović, "Pitch Estimation Via Self-Supervision," in *2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3527-3531. IEEE, 2020.

[3] A. M. Noll, "Cepstrum pitch determination," *The Journal of the Acoustical Society of America*, vol. 41, no. 2, pp. 293-309, 1967.

[4] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "Crepe: A convolutional representation for pitch estimation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 161-165, 2018.

[5] O. Das, J. O. Smith III, and C. Chafe, "Improved Real-Time Monophonic Pitch Tracking with the Extended Complex Kalman Filter," *Journal of the Audio Engineering Society*, vol. 68, no. 1/2, pp. 78-86, 2020.

[6] A. De Cheveigné, and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917-1930, 2002.

[7] M. Mauch, and S. Dixon, "pYIN: A fundamental frequency estimator using probabilistic threshold distributions," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 659-663, 2014.

[8] C. L. Hsu and J. S. H. Jang, "On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset", *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no.2, pp. 310-319, 2009.

[9] R. M. Bittner,, K. Pasalo, J. J. Bosch, G. M. Meseguer-Brocal, and D. Rubinstei, "Vocadito: A dataset of solo vocals with F0, note, and lyric annotations", *Arxiv*, 2021.

[10] B. McFee, C. Raffel, D. Liang, D. P. W. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "Librosa: Audio and music signal analysis in Python", in *Proceedings of the 14th Python in Science Conference*, Austin, USA, 2015, pp. 18-24.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and

E. Duchesnay, “Scikit-learn: Machine learning in Python”, *Journal of Machine Learning Research*, vol. 12, no. 1, pp. 2825-2830, 2011.

[12] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, “mir_eval: A transparent implementation of common MIR metrics,” in *Proceedings of the 15th Int. Society for Music Information Retrieval (ISMIR) Conference*, Taipei, Taiwan, 2014.