# TUNET: ADAPTING THE WAVE-U-NET ARCHITECTURE FOR VOCAL ENGINEERING

**Elena Georgieva, Iran Roman, and Brian McFee**
Music and Audio Research Lab (MARL), New York University, USA
email: elena@nyu.edu

## ABSTRACT

In this paper, authors modified Stollner's Wave-U-Net architecture for vocal engineering. Vocal engineering is a step in the music production process where an engineer tunes and aligns all the vocal stems in a song such that they sound polished and precise. In this paper, 106 pairs of isolated vocal stems from eight all-vocal songs were used as training data, and 20 pairs of stems from two songs were used as validation. One stem in each pair was a raw vocal performance, and the second was hand tuned and aligned. The Wave-U-Net architecture has potential to work for this task, but so far it has not quite generalized to the validation set. Authors suggest future steps.

## 1. INTRODUCTION

Every modern pop song goes through a process of recording, production, mixing and mastering. There is also the little talked about step of vocal engineering. Vocal engineering is a process of tuning and aligning all the vocal stems in a song such that they sound polished and precise. Often, a specialized vocal engineer is hired for the role and they spend 5-10 hours going through this process for the lead and backing vocals in order to get the desired, modern sound. The vocal engineering process includes aligning vocals in time with other vocals and with the backing track, and touching up tuning for a modern sound. It's a tricky process that takes a trained ear, as the performance should sound accurate and polished, yet not robotic. Almost all modern pop songs go through this process, regardless of how skilled the singers are. Pop music simply requires an inhuman level of precision.

The modern tools for the vocal engineering process are two software products: Celemony's Melodyne [1] and Antares' Auto-Tune [2]. Using Melodyne or Auto-Tune, an engineer adjusts pitch note-by-note, as well as elements like pitch modulation/vibrato and pitch drift, as well as where a consonant like an 's' or a 't' is placed. Melodyne allows for pitch and timing adjustments down to 1 cent in pitch and milliseconds for timing. Auto-Tune can be used as the infamous auto-tune effect or to touch up pitch more naturally. It does not allow for timing adjustments. Overall, vocal engineering is a tedious process that can benefit from automation. I especially believe automation would be excellent for backing vocals, which may require less of a human touch than the lead vocal.

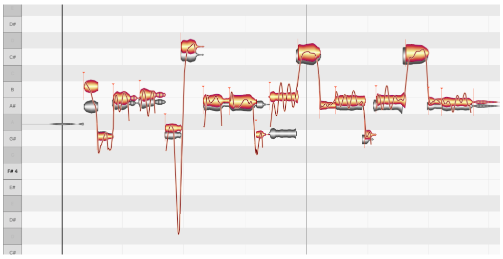


**Figure 1**. A screenshot from the software Melodyne to visualize 7 seconds of a raw and edited audio file. In this screenshot, the orange blobs represent the original vocal line and the grey represents the edited vocal line. The vertical axis is pitch, horizontal is time.

For this project, I think it would be interesting to automate this vocal engineering process. Automating vocal engineering would save time, money, and resources, and allow amateur producers and musicians to create a more decent product without hiring professional help. Additionally, I believe this automation would result in a negligible decrease in quality for backing vocals, especially.

## 2. RELATED WORK

Broadly, machine learning and deep learning have been entering the music production space in recent years, as described in part by authors of the 2020 text "Intelligent Music Production" [1].

Intelligent music production is a fairly new research area that explores using machines for automatic tasks to aid in the recording, mixing, and mastering processes. The goals of the field can be summarized in two categories. First, to automate repetitive and objective, technical tasks, to free up the engineer to think about the art. An example of this is iZotope RX [3], a tool that automatically does tasks like removing plosives or sibilance from a vocal, allowing the engineer to run a music stem through the software and then swiftly move on to mixing the stem into a song. A second goal of intelligent music production is to do a blackbox task to get a decent, acceptable mix or master for an artist who does not have music production skills, or is simply in a rush to get a demo of their song done. An

[1] https://www.celemony.com/en/melodyne/what-is-melodyne

[2] https://www.antarestech.com/

[3] /urlhttps://www.izotope.com/en/products/rx.html

example of this blackbox-stye tool is LANDR[4], an automatic mastering service that converts your mix into an acceptable master. Tools like these effectively "democratize" music production, allowing amatures to make decent content earlier on.

In 2020, A team of researchers published an ICASSP paper called "Deep Autotuner" [2]. It is a pitch corrector network for singing performances that outputs constant pitch shift values up to 100 cents (one semitone) up or down. It can also apply the shifts to the audio. They use a deep neural network including convolutional and recurrent layers. They use in-tune karaoke performances from the Digital Archive of Mobile Performances (DAMP)[5] dataset, collected by researchers at Smule, as 'tuned' data and out-of-tune karaoke performances as 'un-tuned' data. This process could be improved if the process is done with recordings before going through the vocal production process and after, because all singers' vocals are produced, not just those who are more out of tune. Additionally, the Deep Autotuner work does not include timing adjustments, only pitch.

Pitch tracking is a known challenge in the field of music information retrieval for vocals [3]. Another team of researchers released SPICE: a model for self-supervised pitch estimation, though they do not attempt the task of tuning [4]. The SPICE model is trained to recognize relative pitch without access to labelled data and it can also be used to estimate absolute pitch by calibrating the model using just a few labelled examples. The model is able to estimate pitch at a level of accuracy comparable to state-of-the-art fully supervised models including CREPE [5]. Neither of these pitch estimation models perform pitch shifting.

## 3. METHODS

### 3.1 Data

Elena Georgieva is a vocal producer, and data was taken from her archive of vocal recordings. 126 pairs of vocal stems were selected, and each pair was comprised of a raw .wav file and a hand tuned and aligned .wav file. An excerpt of one audio file is illustrated in Figure 1. The orange blobs represent the original vocal line and the grey represents the edited vocal line. Variations in pitch (vertical) and timing (horizontal) can be observed. The most drastic pitch adjustment in this segment is two semi-tones.

The 126 pairs of audio recordings in the dataset are isolated vocal recordings from singers ages 18-30, of all gender identities. Audio files were recorded with a 44.1kHz sampling rate, 24 bit depth and industry standard condenser or dynamic microphones and audio interfaces.

The data are from around 60 individual vocalists performing 10 songs. 8 songs consisting of 106 stems were used as training, and 2 songs comprised of 20 stems were used as validation. All songs were all-vocal arrangements in a pop/contemporary style.

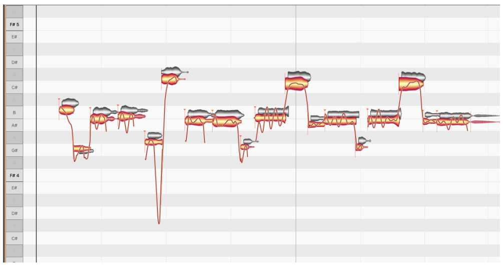


**Figure 2**. A screenshot from the software Melodyne to visualize 7 seconds of a raw audio file and an audio file that has been pitch shifted systematically for the purpose of data augmentation. In this screenshot, the orange blobs represent the original vocal line and the grey represents the augmented vocal line. It has been systematically shifted up by a semi-tone.

For data augmentation, authors used the pitch shift function of Spotify's Pedalboard [6]. Each of the 126 audio files was pitch shifted 10 times: 10, 20, 30, 40, and 50 cents up in pitch and the same amounts down in pitch, where a cent represents 1/100th of a musical whole tone. In total, each raw, un-tuned stem was presented to the model 11 times: as performed, and in each of the 10 pitch-modified forms. That's 1,166 audio files total. One shifted audio file is illustrated in a screenshot of the Melodyne software in Figure 2.

Data were presented to the model in .hdf5 format– an efficient format for managing large datasets.

The author has chosen not to release the dataset at this time, though hopefully we can do so in a future iteration of the project with more data that is more polished and anonymized.

### 3.2 Architecture

To achieve the goal of automatic vocal tuning and alignment, authors adapted an existing deep learning model called Wave-U-Net [6]. The Wave-U-Net is a convolutional neural network, originally used for audio source separation, which works directly on the raw audio waveform, separating an audio mix into four components. The Wave-U-Net, introduced in 2018, is a notable architecture because it operates in the time-domain, which allows modeling phase information. Previously, most audio/machine learning algorithms operated in the frequency domain. An illustration of Stollner's Wave-U-Net model is illustrated in Figure 3.

The Wave-U-Net, in turn, was an adaptation of the spectrogram-based U-Net architecture to the one-dimensional time domain [7]. The original U-Net was used for singing voice separation: a process of separating the vocals from the rest of the audio mix. Through a series of downsampling and upsampling blocks, which also involve 1D convolutions, features are computed on multiple levels of abstraction and time resolution, and combined to make a prediction.

[4] /urlhttps://www.landr.com/

[5] https://ccrma.stanford.edu/damp

[6] `https://github.com/spotify/pedalboard`

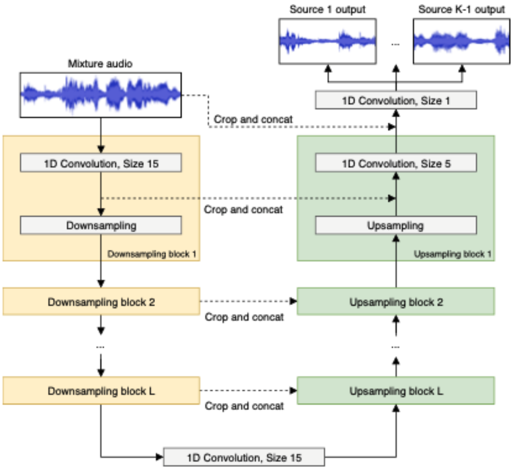


**Figure 3**. A diagram of Stoller's Wave-U-Net model. The Wave-U-Net uses a process of downsampling to compute higher-level features on coarser time scales, followed by upsampling, combining these features with earlier local features, yielding features that are used to make predictions.The network has L layers in total, with each level operating at half the time resolution of the previous one. Authors modified this architecture to output a single audio file.

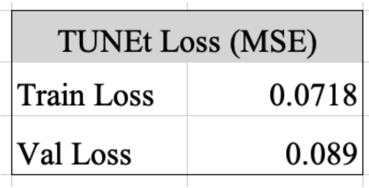
The Wave-U-Net computes more and more high-level features on coarser time scales using downsampling blocks. These are combined with earlier features using upsampling, resulting in multi-scale features which are used for predictions. The network has a change-able number of levels, with each successive level operating at half the time resolution. Each step is a 1-dimensional convolution and includes zero-padding, and is followed by a LeakyReLU activation. The final convolution uses tanh activation. A decimate step discards features for every other time stamp to half the time resolution. Next, upsampling upsamples by a factor of 2 and the high-level features are concatenated with more local features.

To modify the network for vocal engineering data, it was adjusted to accept a mono input (as opposed to stereo), and to output one mono file (vs 4 stereo files).

### 3.3 Evaluation Metrics

As loss, the Wave-U-Net model uses mean squared error (MSE) over all source output samples. Authors use the ADAM optimizer and define 2000 iterations as one epoch. The patience is set to 20, so the model stops training after 20 iterations without improvement on the validation set, measured by the MSE loss. The learning rate was set to the default: 0.0003. Authors ran the model once with a lower learning rate: 0.00003, and did not see a significant change in results.

Additionally, authors used the signal-to-distortion ratio (SDR) metric to evaluate tuning performance [8]. SDR is an overall measure of how good a source sounds, and it is reported as a single number, where a larger positive number represents a good-sounding source. Technically, the SDR is computed when an audio track is partitioned into non-overlapping segments, and segment-wise metrics are averaged over each audio track to evaluate the model's performance.



| TUNEt Loss (MSE) | |
|---|---|
| Train Loss | 0.0718 |
| Val Loss | 0.089 |

**Figure 4**. Mean squared error (MSE) loss over the training set (106 vocal stems) and the validation set (20 vocal stems).

## 4. RESULTS

Initial results show the Wave-U-Net is a promising architecture for this task, but additional data and modifications are necessary to achieve strong results. When the model was trained on 106 audio files and validated on 20, it ran for 22 epochs. This means it improved on the validation set twice and then ran 20 epochs without improvement on the validation set before terminating. At epoch 0, the training MSE loss was 0.08122, and the validation loss was 0.0857. At epoch 21, the training loss was 0.0718 and the validation loss was 0.0890, as shown in Figure 34.

### 4.1 Limitations

Unfortunately, due to time and computational power limitations, authors were not able to train on the augmented dataset of 1166 stems. The author is seeking an efficient way to transfer the 70GB hdf file on to NYU's High Performance Computing environment. Second, the SDR metric proved to be faulty and authors need to re-visit the process of calculating the signal to distortion ratio and consider if it is the right metric for this task in the first place.

## 5. DISCUSSION AND CONCLUSION

In this paper, I modified the Wave-U-Net architecture for vocal tuning and alignment. 106 pairs of isolated vocal stems from eight all-vocal songs were used as training data, and 20 pairs of stems from two songs were used as validation. One stem in each pair was a raw vocal performance, and the second was hand tuned and aligned to create a more polished performance.

These results indicate that the model is not properly generalizing to the validation set. The patience set to 20 stops the training because it is focused on the validation performing. The model needs more data to capture the training data, so the validation set can respond.

Notably, in an earlier training iteration, with the patience set to 1000, the model was able to overfit to a validation set of two stems.

Training with the larger, augmented dataset is the next step. Additionally, we can gather more data from other vocal music industry professionals.

To fully assess what the model is doing, the next steps are to calculate a second evaluation metric in additional to the MSE loss. The signal to distortion ratio was discussed, though other researchers have suggested it may be a flawed metric. The SDR computation is problematic when the source is silent or near-silent. In case of silence, the SDR is undefined (log(0)), which happens often for vocal tracks, especially in this dataset.

## 6. ACKNOWLEDGMENTS

Thanks to singers from ALIAS A Cappella, FLARE Vocals, Stanford Counterpoint, The Stanford Mendicants, and Stanford O-Tone for their vocal performances.

Thanks to Saksham Singh at NYU MARL for hdf5 processing resources.

Thanks to Iran Roman and Brian McFee for mentorship and Joan Bruna, Yann LeCun, Alfredo Canziani, and the NYU Data Science 1008 teaching team for a great course.

## 7. REFERENCES

[1] B. D. Man, R. Stables, and J. D. Reiss, *Intelligent Music Production*, 2020.

[2] S. Wager, G. Tzanetakis, C.-i. Wang, and M. Kim, “Deep autotuner: A pitch correcting network for singing performances,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[3] E. Humphrey, S. Reddy, P. Seetharaman, and A. Kumar, “An introduction to signal processing for singing-voice analysis,” in *IEEE Signal Processing Magazine*, 2019.

[4] B. Gfeller, C. Frank, D. Roblek, M. Sharifi, M. Tagliasacchi, and M. Velimirovic, “Spice: Self-supervised pitch estimation,” in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2020.

[5] J. Kim, J. Salamon, P. Li, and J. Bello, “Crepe: A convolutional representation for pitch estimation,” 2018.

[6] S. D. Daniel Stoller, Sebastian Ewert, “Wave-u-net: A multi-scale neural network for end-to-end audio source separation,” in *International Society for Music Information Retrieval Conference(ISMIR)*, 2018.

[7] A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, and T. Weyde, “Singing voice separation with deep u-net convolutional networks,” in *International Society for Music Information Retrieval Conference (ISMIR)*, 2017.

[8] J. L. Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, “Sdr – half-baked or well done?” 2019.