A Flexible Speech Feature Converter Based on An Enhanced Architecture of U-net

A Thesis

presented to

the Faculty of the Graduate School

at the University of Missouri-Columbia

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

---

by:

Yanghao Yue

Dr. Yunxin Zhao, Thesis Supervisor

DECEMBER 2020

The unsigned, appointed by the dean of Graduate School, have examined the thesis entitled

A Flexible Speech Feature Converter Based on An Enhanced Architecture of U-net

presented by Yanghao Yue

a candidate for the degree of Master of Science,

and hereby certify that, in their opinion, it is worthy of acceptance.

Dr. Yunxin Zhao

Dr. Jianlin Cheng

Dr. Dominic Ho

ACKNOWLEDGEMENTS

First and foremost, I would express my sincere gratitude to my supervisor Dr. Yunxin Zhao, for her kind and valuable guidance in every stage, not only in this project, but also throughout my whole overseas study, and my life. Time goes really fast, but I will always remember this experience and learn from her keen and vigorous academic observation in my future study.

I shall extend my thanks to all the members in Spoken Language and Information Processing lab for their kindness and help. Working and having discussions with them really enlightens me a lot. I will never forget the sense of accomplishment when we solved the problems and the fun we had together.

Last but not least, I'd like to thank all my family and my friends for their encouragement and support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

In order to analyze speech or audio, many methods are applied to transform the time domain signals into various features such as the mel spectral features and WORLD vocoder features. These two types of features can both be extracted from speech or used to synthesize speech. On the other hand, certain applications call for conversion between different types of features. To convert mel spectral features to WORLD vocoder features, one possible method is to first synthesize time domain signal from mel spectrogram and then do the feature extraction by WORLD vocoder. The goal of this project is to develop a direct way to achieve this transformation, i.e., convert mel spectrogram output of text-to-speech (TTS) system to WORLD vocoder features.

In this project, a feature converter is designed to accomplish our aim. The converter has an enhanced neural network architecture based on the U-net. In our design, except for the basic architecture of U-net, the Res Path composed of residual blocks and linear transformations are added on the skip connection. Our flexible system can complete feature conversion directly at feature level without processing in the time domain. In addition to the function of converting mel spectrogram to WORLD features, the reverse transformation from WORLD features to mel spectrogram is also attainable by a few adjustments. The transformed feature has achieved good performance in objective metrics and the converter generalized well to different speakers, which can be applied to produce high quality speech via vocoder resynthesis.

1. INTRODUCTION

A feature is a measurement or description of data samples that represents the essential information in the data. In speech signal processing, mel spectrogram and WORLD vocoder features are two types of features that are widely used to analyze and synthesize speech.

Mel spectrogram is mel-scaled spectrogram, which is first obtained from the time domain audio signals by STFT (Short-time Fourier transform) and then processed by mel-scale filter banks. In mel spectrograms, normal frequency scale (Hz) is converted to mel frequency scale. Mathematically, the mel scale is the result of non-linear transformation of the frequency scale and it's a perceptual scale of pitches judged by listeners to be equal in distance from one another. Besides the spectrogram structure information, the mel scale aims to mimic the non-linear frequency resolution of human ear perception of sound, which is more discriminative at lower frequencies and less discriminative at higher frequencies.

WORLD vocoder [1] is based on a speech production model and its features consist of three components. The first one is fundamental frequency (F0), which is defined as the lowest frequency of a periodic waveform and human ears identify it as the specific pitch of the speech tone. The second one is spectral envelope which is the envelope curve of the amplitude spectrum. And the last one is aperiodicity (AP) which has the aperiodic information of the speech signal. In the process of feature extraction and synthesis, we need all three features, referred as WORLD vocoder features.

In speech study, we need to do the conversion between different types of features according to some certain requirements and applications. Certainly, it's possible to use an indirect method that is to first synthesize the time-domain signal from the source feature and then do the target feature extraction. But the procedure of converting features to the time domain wave is often lossy and with computation overhead, and it is therefore desired to find a direct way of making conversion at feature level.

1.1 Problem Description

Admittedly, different types of features can represent speech characteristics in different forms, and they might be used for different purposes independently. In the famous text-to-speech system Tacotron2 [2], mel spectrogram is thought to be easier to train when it's used as the predicted output of the neural network model from the text sequence. Next the time-domain audio samples are generated from mel spectrograms by Griffin-Lim algorithm or data-driven vocoder like Waveglow [3]. But the WORLD vocoder features can provide more detailed structural information than mel spectrograms e.g., we can determine whether the tone of TTS output is expressive or flat by fundamental frequency (F0) contour. Because training a new neural TTS model using WORLD vocoder features is a challenging task, we came up with an idea of developing a feature converter from mel spectrograms to WORLD vocoder features instead of going back to the time domain. And of course, we must ensure that the converted features can also be used to produce accurate and high-quality speech.

1.2 Proposed Method

Since we can consider the mel spectrogram as a matrix, if we stack the three types of WORLD feature components to a matrix, then what we actually need is to figure out the relationship between Mel matrix and WORLD matrix. With the continuous advance in deep learning, there is no longer the need for complicated mathematical derivations, instead, deep neural networks can be trained to learn this complex relationship. Among many kinds of matrices, image is easy to process and display. Many famous neural network structures have been applied to the image field, and most of them have achieved good results. With the inspiration of image-to-image transformation architecture [4], an enhanced neural network based on U-net [5] is designed to be our feature converter. To accomplish the Mel to WORLD feature conversion, we take mel spectrograms as a source and WORLD vocoder features as a target to train our neural network and choose a best model as our converter.

2. DATA PREPROCESSING

In this project, we used a public domain dataset named LJSpeech [6]. The LJSpeech dataset consists of 13,100 short audio clips of a single female speaker who speaks American English. Clips recorded in 22.05kHz sampling rate vary in length from 1 to 10 seconds and have a total length of approximately 24 hours. Before designing the architecture of our neural network, we need to prepare and pre-process the data first. Since we want to implement the mel spectrogram to WORLD vocoder features converter, on the source side, we need to turn time-domain audio from LJSpeech into mel spectrogram, and on the target side, we need to turn the audio into WORLD features through WORLD vocoder.

In this section, we will illustrate the process of data preprocessing and related parameters for the two types of features.

2.1 Mel spectrogram

We used the same process and settings as Tacotron2 to get the mel spectrogram. First of all, because each audio file in LJSpeech is a single-channel 16-bit PCM WAV, we normalize all the audio files by dividing it by the 16-bit maximum absolute value 32768 after loading. Then we apply STFT to get 513-dimensional (frequency bin) linear spectrograms with FFT size of 1024, Hann window length of 1024 samples, and frame hop length of 256 samples (number of audio samples between adjacent STFT frames, or matrix columns). Next, we extract the magnitude spectrograms from linear complex spectrograms and create an 80 mel bands filter bank matrix, and generate mel spectrograms by multiplying the filter bank matrix with magnitude spectrograms. Let N denote the number of frames, $M \in R^{513\times N}$ denote magnitude spectrograms and $F \in R^{80\times 513}$ denote the mel filter bank matrix. Then we get the mel spectrograms $Mel \in R^{80\times N} = FM$. In the final step, we take the log operation to get the log mel spectrograms.

As examples, we include two mel spectrograms below in Figs. 1 & 2, extracted from two sentences in LJSpeech. Note that the figures use different shades of color to represent the values of the time-frequency elements in matrices, the brighter the color, the larger the value.

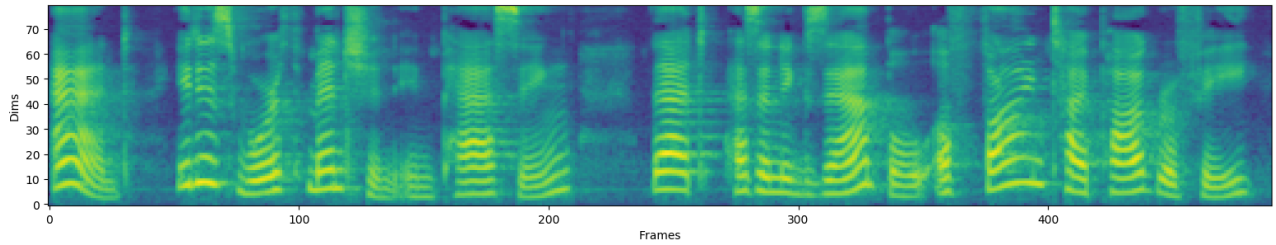

Figure 1 Mel spectrogram of LJ001-0006

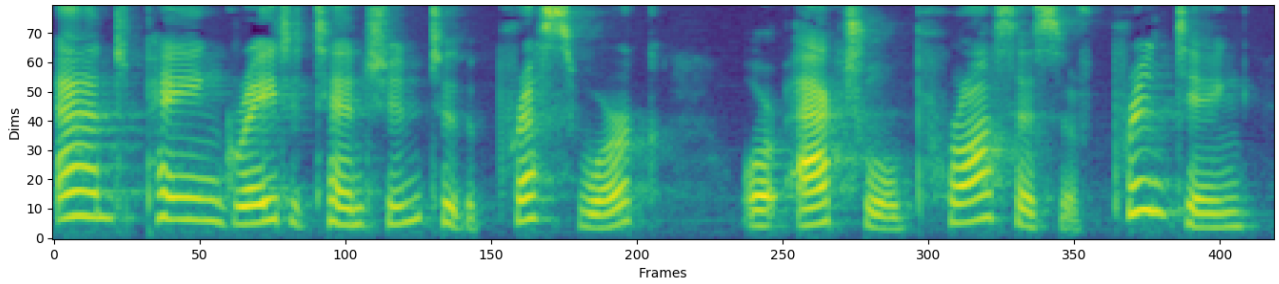

Figure 2 Mel spectrogram of LJ001-0051

2.2 WORLD vocoder features

The WORLD vocoder system consists of three analysis algorithms for determining the three types of WORLD vocoder features and a synthesis algorithm that uses these parameters to generate speech waveform. First the F0 is estimated with Harvest [7]. Second, the spectral envelope is estimated with CheapTrick [8], which uses not only the waveform but also the F0 information. Third, the D4C [9]

algorithm is applied to estimate the aperiodicity with the waveform and the F0 information. Figure 3 illustrates the whole process of the system.

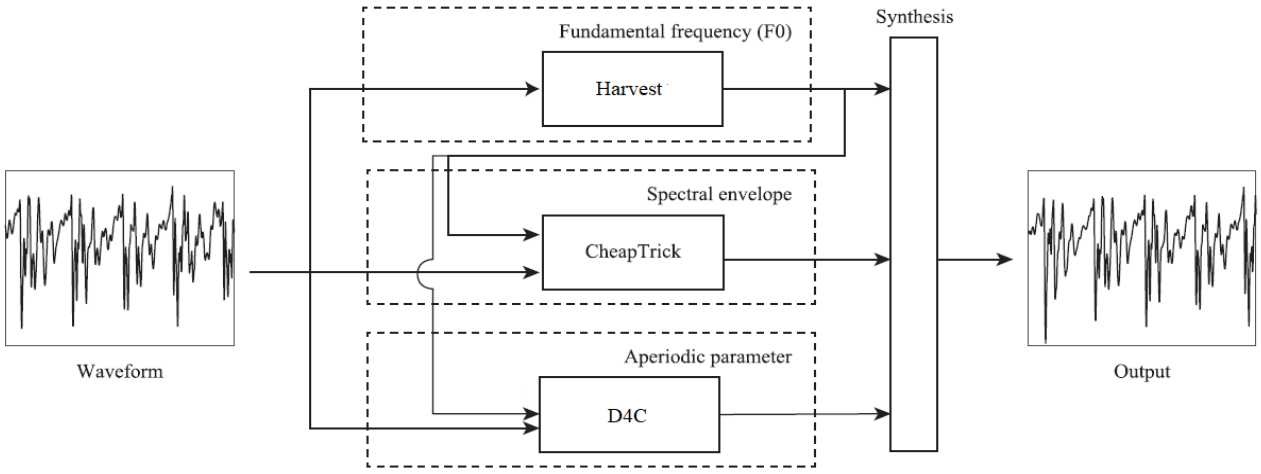

Figure 3 Overview of WORLD analysis/synthesis system

The WORLD vocoder system also uses 1024 samples FFT size and 1024 samples Hann window length, but one difference is that the WORLD uses a parameter called frame period to determine how many frames this audio file has. Because the sampling rate in LJSpeech is 22.05kHz and the hop length is 256 samples, to keep the same number of frames as mel spectrograms, we set the frame period to 11.61 milliseconds. After applying the three algorithms, (N denotes the number of frames) we now have data size of $N \times 513$ for spectral envelope, $N \times 1$ for F0 and $N \times 513$ for aperiodicity. If we concatenate the three matrices in the column direction, the feature matrix will be $N \times 1027$ in size. As examples, in Figs 4 and 5 we show the spectral envelope and F0 features. Note that for better visualization, we take the log operation of the spectral envelope to compress its dynamic range.

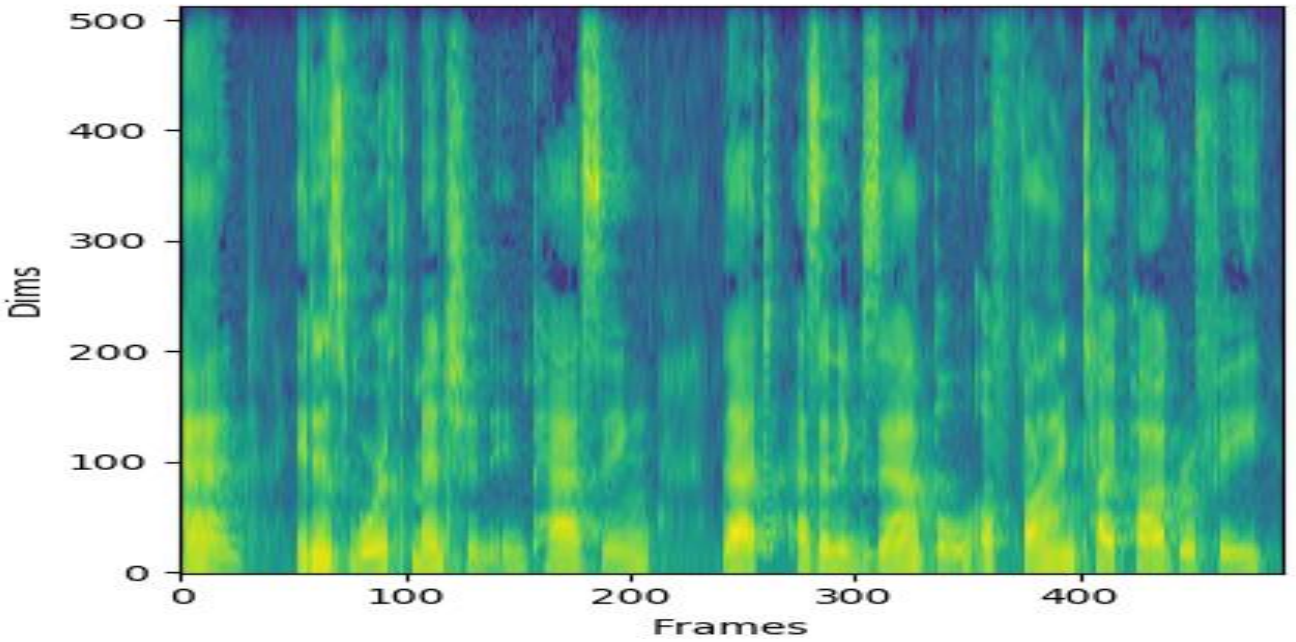

Figure 4 Spectral envelope of LJ001-0006

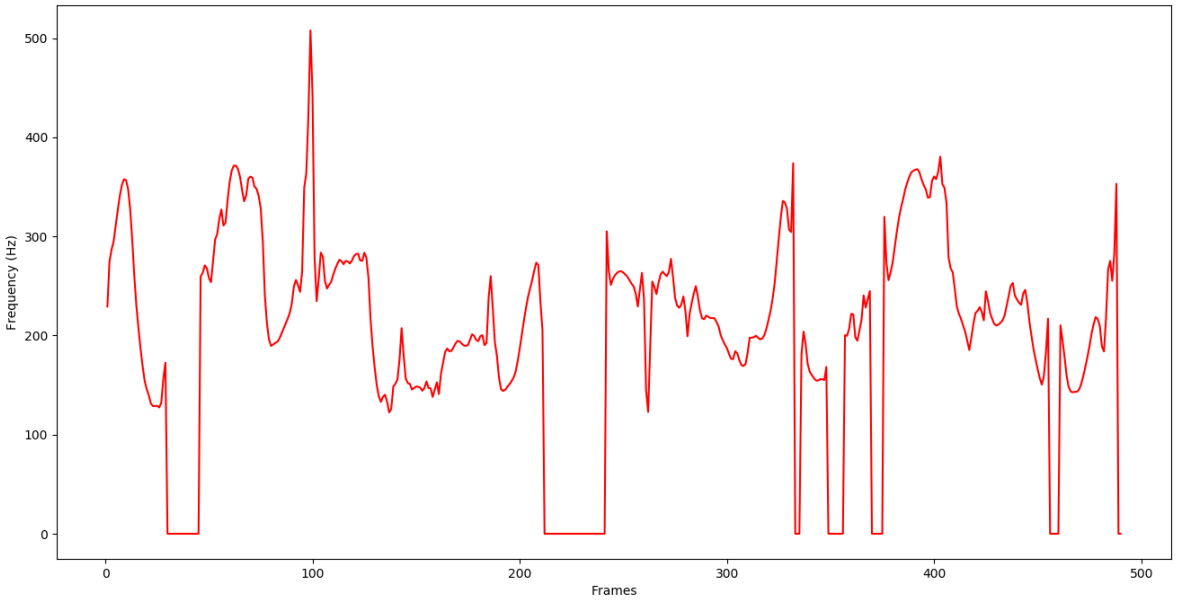

Figure 5 F0 contour of LJ001-0006

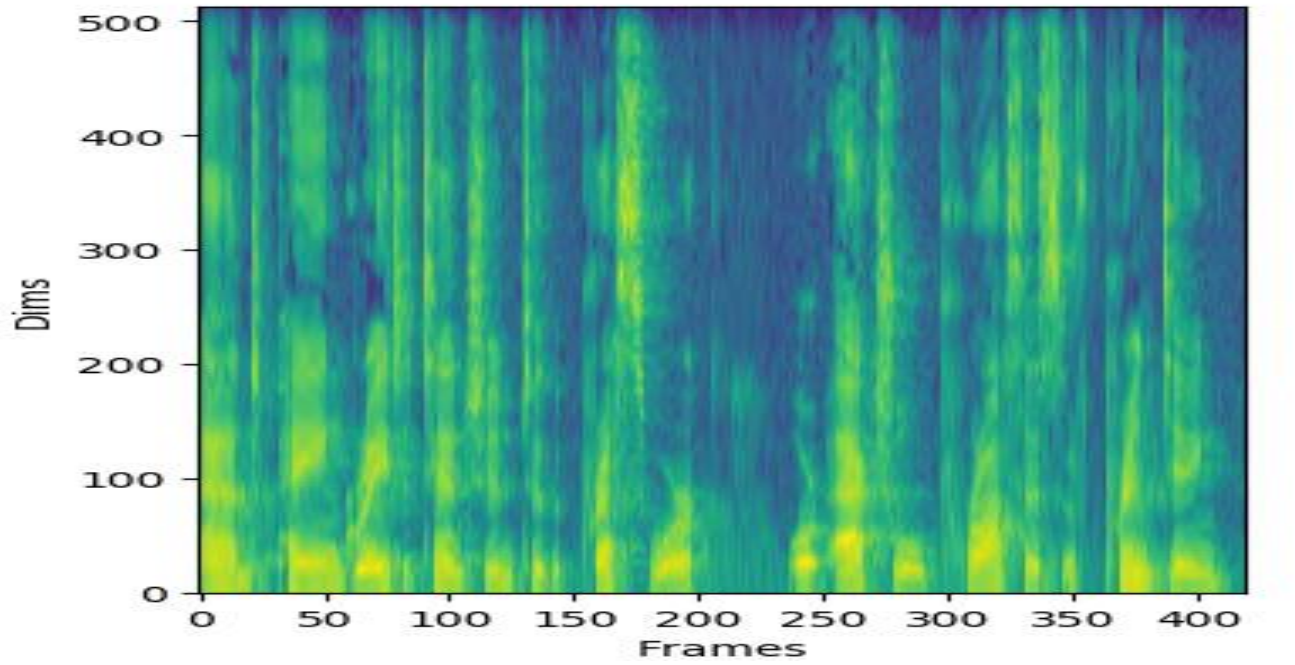

Figure 6 Spectral envelope of LJ001-0051

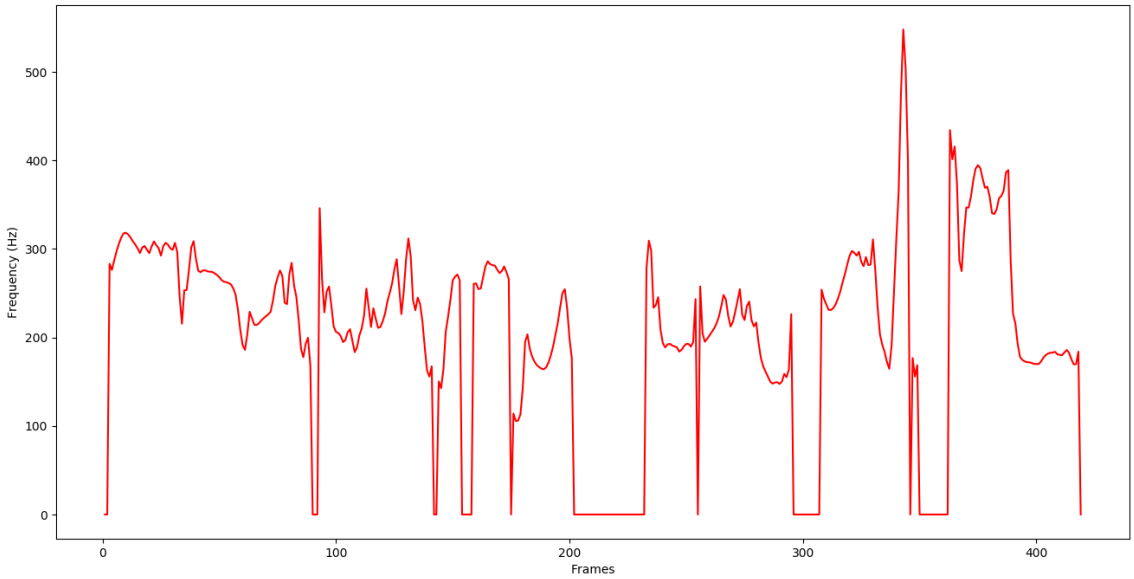

Figure 7 F0 contour of LJ001-0051

When using the high dimensional WORLD features as the conversion target, the converter can be trained to learn the conversion in our pilot experiments. But the speech synthesized by such converted features is somewhat noisy and the training process takes a long time. To overcome this deficiency and save storage space, we transform the 513-D (Dimension) spectral envelope to 60-D mel-generalized-cepstrum (MGC) via SPTK(Speech Signal Processing ToolKit). Then we change the 513-D aperiodicity to the coded 2-D band aperiodicity which is calculated as the power ratio between total power and the power of the sine wave for each frequency band to reduce dimensionality of D4C aperiodicity. These two transformations are reversible. If we can ensure high accuracy and low error in the 60-D MGC and 2-D band aperiodicity output through the converter, then we can still synthesize high quality speech similar to natural recorded speech by WORLD vocoder after the reverse transformation. By including 1-dimensional voiced and unvoiced (VUV) information extracted from F0 or aperiodicity per frame and taking the log operation of F0, we finally get the 64 dimensions WORLD vocoder features which will be used in the subsequent training and synthesis processes. In the synthesis part of WORLD system, the 513-D spectral envelope, F0 and 513-D aperiodicity are used as the input.

3. FEATURE CONVERTER DESIGN

In this section, we will introduce the proposed audio feature converter based on an enhanced architecture of U-net. Firstly, we illustrate the enhanced network architecture for mel spectrogram to WORLD vocoder features transformation. Then we present the design of the converter which can do the reverse conversion from WORLD features to mel spectrogram. Finally, we give the details of the training procedure for the converter networks.

3.1 Network architecture

3.1.1 Overview of the architecture

The U-net based network consists of three main parts: Encoder, Decoder and Skip Connection.

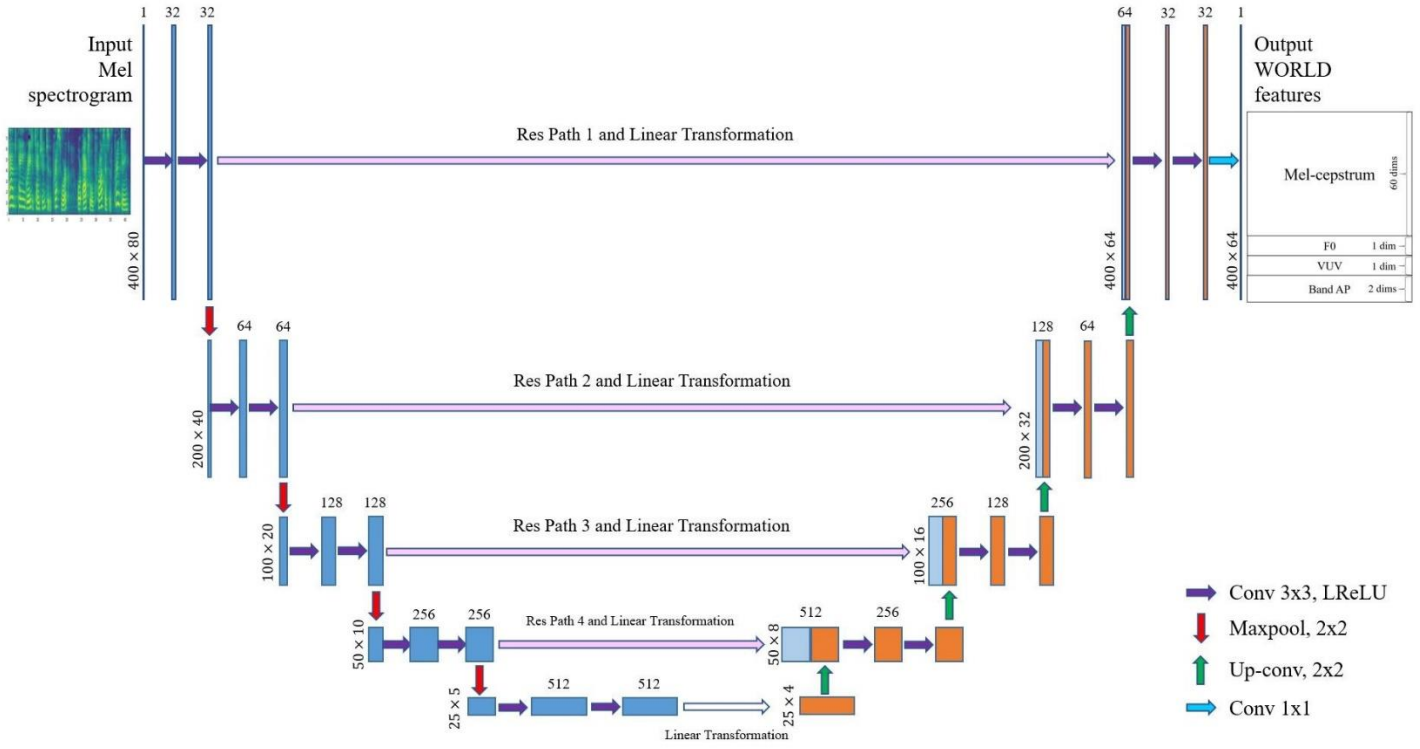

Figure 8 The architecture of Mel to WORLD feature converter network based on U-net

As can be seen from Fig. 8, the network architecture is symmetric with a U-shaped structure. On the left side, it's the encoder. In mel spectrogram to WORLD vocoder features conversion, the task of the encoder is to extract spatial features from the mel spectrogram matrix, e.g., 400×80 size, which means 400 frames of 80-D mel spectral features. The encoder follows the typical formation of a convolutional network. It involves a block of two layers of 3×3 convolution operations, followed by a max-pooling operation with a pooling size of 2×2 and stride of 2. This block is repeated four times, and each time after down-sampling in which the output size will be a quarter of its previous size, the number of filters in the convolutional layers is doubled. Since we set the stride and zero-padding size to 1, as the number of channels increases, the convolution operation does not change the size of the output. Finally, a progression of two 3×3 convolution operations and a linear transformation in our case connects the encoder to the decoder. For the encoder, if we take a 400×80 mel matrix as input, the output size will be 25×5 or 25×4 after linear transformation in each of the 512 channels at the bottleneck between encoder and decoder.

On the other hand, the decoder aims to construct the WORLD vocoder features from the encoded mel spectral features. The decoder on the right side first upsamples the feature map using a 2×2 transposed convolution operation [10], reducing the feature channels by half but doubling the height and width of the feature map. Then a block of two layers of 3×3 convolution operations is performed again. Similar to the Encoder, this succession of upsampling and two convolution operations is repeated four times, halving the number of filters at each stage. Finally, a 1×1 convolution operation used for reducing the number of channels from 32 to 1 is performed to generate the 64 dimensions WORLD features which includes the 60-D mel-cepstrum, 1-D F0, 1-D VUV and 2-D band aperiodicity. Because of the settings in convolution, the WORLD features still retain the same number of frames as the mel spectrogram. All convolutional layers in this architecture, except for the final one, use the LReLU (Leaky Rectified Linear

Unit) activation function [11]. Because we want our output WORLD features to match the target matrix directly in values, we do not need to use the activation function at the end.

Perhaps, the most ingenious aspect of the U-net architecture is the introduction of skip connections between its encoder and decoder. In all the four levels, the output of the convolutional layer, prior to the pooling operation of the encoder is transferred to the decoder. In the basic U-net architecture, these encoded features are then concatenated with the output of the upsampling operation. Specifically, each skip connection concatenates all channels of output feature maps at the same level in encoder and decoder. Then the concatenated feature map is propagated to the successive layers. Both shallow and deep levels of feature information are combined by skip connections and these can also allow the network to retrieve the spatial information lost by previous operations. For the goal of feature conversion, Res Path and linear transformation are added on the skip connection in our enhanced architecture. Next, Res Path and linear transformation in our converter will be introduced respectively.

3.1.2 Res Path

The idea of Res Path comes from MultiResUnet [12]. In convolutional neural networks, due to the loss of information by pooling layers and the possible gap introduced by the processing in deep decoder stages, the basic design of skip connection as in the original U-Net is not optimal. Thus, instead of simply concatenating the feature maps from the encoder stages to the decoder stages, we first pass them through a chain of convolutional neural networks with residual connections and then concatenate (or after linear transformation) them with the decoder features. This proposed shortcut is called ‘Res Path’. More specifically, 3 × 3 filters are used in the convolutional layers and 1 × 1 filters accompany the residual connections [13]. Furthermore, residual connections are also introduced as they make the learning easier and are very useful in deep convolutional networks.

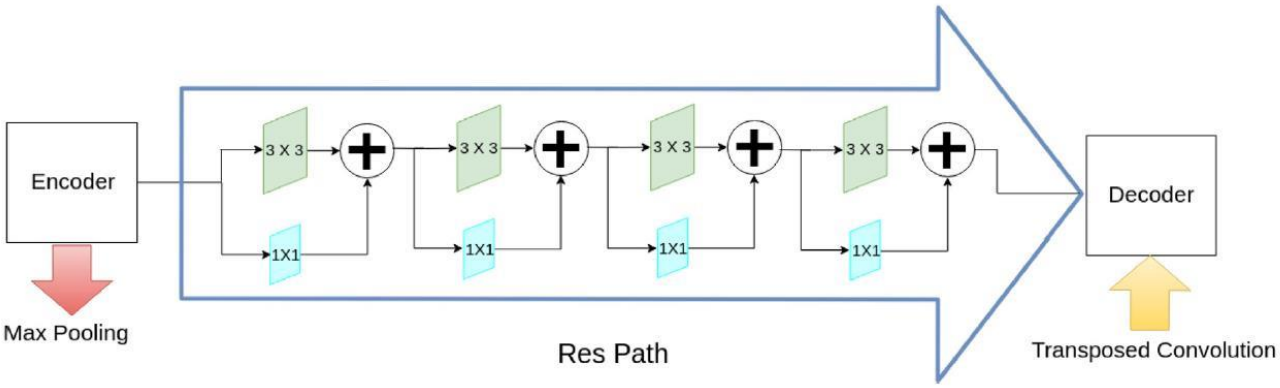

Figure 9 Design of Res Path

As shown in Fig. 9, we replaced the ordinary skip connections with the proposed Res Path. Therefore, we apply some convolution operations on the feature maps propagating from the encoder stage to the decoder stage. We believe that the semantic gap between encoder and decoder feature maps are likely to decrease as we move towards the inner skip paths and deeper levels. Hence, we also gradually reduce the number of the residual convolution blocks used along the Res Paths. In Fig.9, there are 4 residual blocks in total, each with a 3 × 3 convolution layer and a 1 × 1 convolution layer set on the shortcut. We use 4, 3, 2, 1 residual blocks respectively along the four Res Paths in our architecture. Also, in order to match the number of channels of feature maps in encoder and decoder, we use 32,64,128,256 filters in the blocks of the four Res Paths individually.

3.1.3 Linear Transformation

In addition to the Res Path, linear transformation is required to accomplish the feature conversion on account of the dimension difference between mel spectrogram and WORLD vocoder features. In the U-net based architecture, the only operations that change the feature map size or dimension are the max-pooling operation for down-sampling and transposed convolution operation for up-sampling. On the encoder side, the dimensions are 80, 40, 20, 10 from the input mel spectral features to the fourth level

feature maps. After the fourth down-sampling, the dimension becomes 5 at the bottleneck. However, to construct the 64-D WORLD features, if we go backwards from the output to the bottleneck on the decoder side, we can derive that the dimensions to be 64, 32, 16, 8 which correspond to the dimensions at each level of encoder. Thus at the bottleneck, linear transformation is used to transform the 5-D feature maps to 4-D feature maps. Then the 64-D WORLD feature are output through the following operations in decoder.

This linear transformation can be explained by the following formula:

$$y = xA^T + b \tag{1}$$

Here, x is the input data matrix with 5-D features in each channel of the feature maps and y is the output data matrix with 4-D features. A^T is the linear transformation matrix with the size of $in\_dim \times out\_dim$. By multiplying the $N \times in\_dim$ matrix x and the $in\_dim \times out\_dim$ matrix A^T, plus bias b, we can get the $N \times out\_dim$ output y. In the Pytorch framework that we use, a function module called *torch.nn.Linear* which is used as *nn.Linear(in_dim, out_dim)* in the code can help us do the math and find the best A^T and b while training the network model.

On the four skip connections in our feature converter, to concatenate the feature maps from the encoder stages to the decoder stages after Res Paths, four different linear transformations are applied to match the dimensions in each channel of the feature maps on the encoder and decoder stages at the same level. The sizes of matrix A^T from the first to the fourth level are 80×64, 40×32, 20×16, 10×8, and 5×4 at the bottleneck.

Combining these three main parts, our enhanced architecture based on U-net is completed. The architectural details are described in Table 1.

Table 1 Details of the Mel to WORLD feature converter network

<table>
<tr><th colspan="5">Feature Converter Neural Network</th></tr>
<tr><td colspan="5">Input: Mel spectrogram</td></tr>
<tr><th>Layer Name</th><th>Information</th><th>Res Path & Linear Transformation</th><th>Information</th><th>Channels</th></tr>
<tr><td>Conv. 1</td><td>Double Conv2D(3,3,32), stride=1; Batchnorm; LReLU</td><td rowspan="8">Res Path 1</td><td rowspan="8">Conv2D(3,3)
Conv2D(1,1)

Conv2D(3,3)
Conv2D(1,1)

Conv2D(3,3)
Conv2D(1,1)

Conv2D(3,3)
Conv2D(1,1)</td><td rowspan="8">32</td></tr>
<tr><td>Down. 1</td><td>Maxpool2D(2)</td></tr>
<tr><td>Conv. 2</td><td>Double Conv2D(3,3,64), stride=1; Batchnorm; LReLU</td></tr>
<tr><td>Down. 2</td><td>Maxpool2D(2)</td></tr>
<tr><td>Conv. 3</td><td>Double Conv2D(3,3,128), stride=1; Batchnorm; LReLU</td></tr>
<tr><td>Down. 3</td><td>Maxpool2D(2)</td></tr>
<tr><td>Conv. 4</td><td>Double Conv2D(3,3,256), stride=1; Batchnorm; LReLU</td></tr>
<tr><td>Down. 4</td><td>Maxpool2D(2)</td></tr>
<tr><td>Conv. 5</td><td>Double Conv2D(3,3,512), stride=1; Batchnorm; LReLU</td><td>Linear 1</td><td>Linear(80,64)</td><td>32</td></tr>
<tr><td>Up. 6</td><td>DeConv2D(2,2,256), stride=2</td><td rowspan="3">Res Path 2</td><td rowspan="3">Conv2D(3,3)
Conv2D(1,1)
Conv2D(3,3)
Conv2D(1,1)
Conv2D(3,3)
Conv2D(1,1)</td><td rowspan="3">64</td></tr>
<tr><td>Conv. 6</td><td>Double Conv2D(3,3,256), stride=1; Batchnorm; LReLU</td></tr>
<tr><td>Up. 7</td><td>DeConv2D(2,2,128), stride=2</td></tr>
<tr><td>Conv. 7</td><td>Double Conv2D(3,3,128), stride=1; Batchnorm; LReLU</td><td>Linear 2</td><td>Linear(40,32)</td><td>64</td></tr>
<tr><td>Up. 8</td><td>DeConv2D(2,2,64), stride=2</td><td>Res Path 3</td><td>Conv2D(3,3)
Conv2D(1,1)
Conv2D(3,3)
Conv2D(1,1)</td><td>128</td></tr>
<tr><td>Conv. 8</td><td>Double Conv2D(3,3,64), stride=1; Batchnorm; LReLU</td><td>Linear 3</td><td>Linear(20,16)</td><td>128</td></tr>
<tr><td>Up. 9</td><td>DeConv2D(2,2,32), stride=2</td><td>Res Path 4</td><td>Conv2D(3,3)
Conv2D(1,1)</td><td>256</td></tr>
<tr><td>Conv. 9</td><td>Double Conv2D(3,3,32), stride=1; Batchnorm; LReLU</td><td>Linear 4</td><td>Linear(10,8)</td><td>256</td></tr>
<tr><td>Conv. 10</td><td>Conv2D(1,1,1), stride=1</td><td>Linear 5</td><td>Linear(5,4)</td><td>512</td></tr>
<tr><td colspan="5">Output: WORLD vocoder features</td></tr>
</table>

3.2 Reverse feature converter

In the content above, we have described the mel spectrogram to WORLD vocoder features converter in detail. Since we can achieve the transformation from Mel to WORLD features through this converter, and the U-net architecture is highly symmetric, we can also achieve the reverse transformation by our neural network, namely the conversion from WORLD features to mel spectrogram. The architecture of the reverse feature converter is illustrated below.

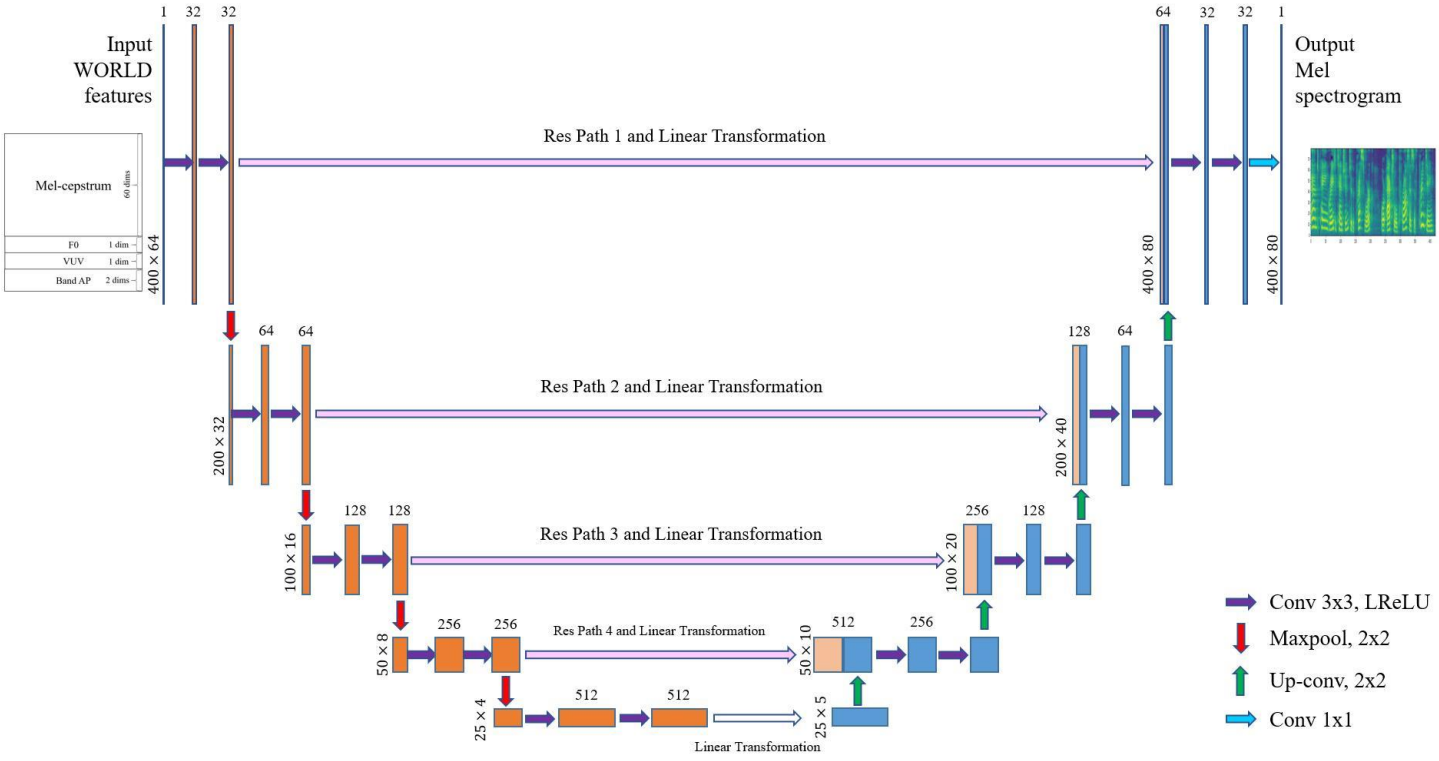

Figure 10 The architecture of WORLD to Mel feature converter network based on U-net

And the details of the reverse feature converter are listed in Table 2.

Table 2 Details of the WORLD to Mel feature converter network

Feature Converter Neural Network				
Input: WORLD vocoder features				
Layer Name	**Information**	**Res Path & Linear Transformation**	**Information**	**Channels**
Conv. 1	Double Conv2D(3,3,32), stride=1; Batchnorm; LReLU	Res Path 1	Conv2D(3,3) Conv2D(1,1) Conv2D(3,3) Conv2D(1,1) Conv2D(3,3) Conv2D(1,1) Conv2D(3,3) Conv2D(1,1)	32
Down. 1	Maxpool2D(2)			
Conv. 2	Double Conv2D(3,3,64), stride=1; Batchnorm; LReLU			
Down. 2	Maxpool2D(2)			
Conv. 3	Double Conv2D(3,3,128), stride=1; Batchnorm; LReLU			
Down. 3	Maxpool2D(2)			
Conv. 4	Double Conv2D(3,3,256), stride=1; Batchnorm; LReLU			
Down. 4	Maxpool2D(2)			
Conv. 5	Double Conv2D(3,3,512), stride=1; Batchnorm; LReLU	Linear 1	Linear(64,80)	32
Up. 6	DeConv2D(2,2,256), stride=2	Res Path 2	Conv2D(3,3) Conv2D(1,1) Conv2D(3,3) Conv2D(1,1) Conv2D(3,3) Conv2D(1,1)	64
Conv. 6	Double Conv2D(3,3,256), stride=1; Batchnorm; LReLU			
Up. 7	DeConv2D(2,2,128), stride=2			
Conv. 7	Double Conv2D(3,3,128), stride=1; Batchnorm; LReLU	Linear 2	Linear(32,40)	64
Up. 8	DeConv2D(2,2,64), stride=2	Res Path 3	Conv2D(3,3) Conv2D(1,1) Conv2D(3,3) Conv2D(1,1)	128
Conv. 8	Double Conv2D(3,3,64), stride=1; Batchnorm; LReLU	Linear 3	Linear(16,20)	128
Up. 9	DeConv2D(2,2,32), stride=2	Res Path 4	Conv2D(3,3) Conv2D(1,1)	256
Conv. 9	Double Conv2D(3,3,32), stride=1; Batchnorm; LReLU	Linear 4	Linear(8,10)	256
Conv. 10	Conv2D(1,1,1), stride=1	Linear 5	Linear(4,5)	512
Output: Mel spectrogram				

We keep the previous settings in the basic U-net architecture and Res Path. The only adjustment is on the linear transformation. This time, the dimension of the input is 64 and output has 80 dimensions. Therefore, the sizes of matrix A^T from the first to the fourth level are 64×80, 32×40, 16×20, 8×10, and 4×5 at the bottleneck. After the architecture is determined, we exchange the source and target data used in Mel to WORLD converter to train and choose a best model as our WORLD to Mel converter.

Now the enhanced U-net architecture is capable of accomplishing the mutual conversion of mel spectrogram and WORLD features, which shows the flexibility of our speech feature converter.

3.3 Network training details

3.3.1 Hyperparameters

The neural network of the speech feature converter is programmed and trained using Pytorch, an open source machine learning framework. We use the Adam optimizer with a learning rate of 0.001 and batch size is usually set to 32 or can be adjusted according to the memory capacity of GPUs. Batch-normalization [14] which enables faster and more stable training of deep neural networks are applied in all the convolutional layers in this network, except for the last output layer. The 13,100 audio clips from LJSpeech are randomly divided into three parts, namely the training set of 12,500 speech utterances, the test set of 500 and the validation set of 100 speech utterances. The model is trained for 100 epochs by using the data. And the total number of parameters which are trainable in the two types of model is listed in Table 3.

Table 3 The number of parameters in models

Model	Parameters
Mel to WORLD feature converter	8,924,225
WORLD to Mel feature converter	8,924,256

3.3.2 Loss function

For the loss function of the network, since we want all the values of our converted output matrix to be as close as possible to the ground truth values, we choose L1 loss that measures the mean absolute error (MAE) between converted output and the target. The L1 loss for our feature converter can be defined as:

$$L_1(\hat{y}, y) = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i| \quad (2)$$

where $\hat{y}$ is the converted output and y is the target, and the absolute error of each value i is computed and the sum of the error is divided by the total number N to get the average. During the training, the smaller the loss, the better the performance of the model. After each epoch, the L1 loss on validation set is used to observe whether overfitting occurs in the model training and if early stopping is needed. Also, the loss curve can give us the feedback to adjust the hyperparameters.

In our experiments, we achieved good results by using L1 loss. At the very beginning, we chose the widely used L2 loss that measures mean square error (MSE) in our Mel to WORLD feature converter, but the results were not good, especially on the 1-D F0 contour of converted the WORLD features. After a close examination of this problem, we found the square operation of the error in L2 loss function to be a possible reason. Due to the fact that our 64-D WORLD feature matrix is composed of four submatrices i.e., MGC, F0, VUV, and band aperiodicity which are different in the range of values, the loss values that each submatrix contribute to the total loss is very different. After the square operation, such differences are magnified and sometimes will mislead the training of our network when the loss is backpropagated.

3.3.3 Size padding

In our speech dataset, the lengths of different audio clips are mostly different which means the number of frames H is also different after being transformed to the W-dimensional features. When we use

the mini-batch method to load the shuffled data in batches, N features in the same batch differ in the frame size, so we can't form batches with a size of (N,C,H,W). Here, C represents the number of channels. Before first layer and after the final layer, C is 1. In order not to damage the integrity and continuity of speech, we will not take the operation like image cropping. Thus, to solve this problem, size padding is needed in data loading.

In a general way, we first find the matrix with the maximum number of frames in a batch where data are loaded randomly, and then use right zero-padding to make the frame numbers of other matrices the same as the maximum length. Because of the randomness of this method, it's sometimes possible to put a very long speech sequence and a very short sequence in the same batch and we need to add long zero-padding on the right ride of the short sequence, which takes up more memory space.

Another feasible way is that we first sort the entire speech dataset by length, and then select the data with similar length according to the batch size for loading. This will avoid introducing overlong or too much padding, but for data loading, it is not a completely random method. We have applied both of ways respectively to the training of converter and the results are almost the same good.

Furthermore, we also have to make the frame number a multiple of 16 by padding. Because in our network, the number of frames is only changed when the feature map is upsampled or downsampled. In the max-pooling operation for downsampling, if the frame number is odd, it will be divided by 2 and rounded down, then the number becomes an even number and can't be restored to the original odd number after up-sampling. Since we want to keep the length of the speech synthesized by converted features and downsampling is done 4 times, our frame number need to be padded to be multiples of 16. When we use a trained model to convert the feature separately, we also apply the padding first, then remove the padding part after conversion, and finally use the feature to synthesize the speech.

4. RESULTS AND EVALUATION

In this section, we evaluate the performance of our proposed feature converter in objective metrics. The two converters for transformations between mel spectrogram and WORLD vocoder features are both evaluated. For comparison purposes, the performance of other architectures is also listed. And some results are shown in figures as well.

4.1 Metrics

To illustrate the performance of feature conversion, we used quantitative measures for evaluation over the test sets. In Mel to WORLD feature converter, we use Mean Absolute Error (MAE) to calculate the error of spectral envelope, F0, aperiodicity, and the global error of the feature matrix between the converted features and the ground truth features. MAE is also applied in WORLD to Mel feature converter to evaluate the converted mel spectrograms. When the total MAE of each test set is figured out, we take the average as our final result.

Since F0 has a one-dimensional contour which can be viewed as a vector, we use the Cosine Similarity as an extra measure for F0 contour. Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space. It is defined to equal the cosine of the angle between them. The cosine of 0° is 1, and it is less than 1 for any angle in the interval (0, π] radians. It is thus a judgment of orientation and not magnitude: two vectors with the same orientation have a cosine similarity of 1, two vectors oriented at 90° relative to each other have a similarity of 0, and two vectors diametrically opposed have a similarity of -1, independent of their magnitude. Because the value of cosine similarity is in the interval [-1, 1], the closer this value gets to 1, the better F0 we get after conversion.

4.2 Mel to WORLD feature converter evaluation

4.2.1 Test sets

In this evaluation, besides 500 speech utterances from LJSpeech, we use another dataset called LibriTTS [15]. LibriTTS is a multi-speaker English corpus of approximately 585 hours of read English speech at 24kHz sampling rate. Because the speaker of LJSpeech is female, we select one male speaker in the speakers of LibriTTS in particular as well as a female speaker. After downsampling the audio files to 22.05kHz, we use 330 audio files from the male speaker and 269 audio files from the female speaker in LibriTTS as our other two test sets. And 500 utterances output by Tacotron2 and Waveglow which we call 'TTS speech' are also used for evaluation.

4.2.2 Results

First, in Table 4, we list the distribution of the values in target spectral envelope, F0, aperiodicity matrices and the global distribution of values in target WORLD feature matrix, which are all extracted from natural recorded speech of the first three test sets. For TTS speech, we regard the waveforms which are synthesized by Waveglow from the mel spectrograms output by Tacotron2 as the 'natural speech'. And we extract WORLD features from these waveforms as our target.

Table 4 Value distribution of the target WORLD feature in different test sets

Test sets	**Spectral envelope**	**F0**	**Aperiodicity**	**Global WORLD feature**
LJSpeech	$[0.618 \times 10^{-13}, 30.596]$ $\mu = 5.351 \times 10^{-3}, \sigma = 0.029$	$[0, 742.885]$ $\mu = 143.243, \sigma = 73.686$	$[0.001, 1.000]$ $\mu = 0.702, \sigma = 0.266$	$[0, 742.885]$ $\mu = 0.493, \sigma = 4.623$
LibriTTS Female speaker	$[6.492 \times 10^{-13}, 12.312]$ $\mu = 3.411 \times 10^{-3}, \sigma = 0.038$	$[0, 768.556]$ $\mu = 157.957, \sigma = 109.961$	$[0.001, 1.000]$ $\mu = 0.736, \sigma = 0.275$	$[0, 768.556]$ $\mu = 0.523, \sigma = 5.405$
LibriTTS Male speaker	$[7.295 \times 10^{-13}, 15.098]$ $\mu = 4.679 \times 10^{-3}, \sigma = 0.016$	$[0, 626.952]$ $\mu = 99.291, \sigma = 130.927$	$[0.001, 1.000]$ $\mu = 0.675, \sigma = 0.584$	$[0, 626.952]$ $\mu = 0.436, \sigma = 5.853$
TTS speech	$[8.912 \times 10^{-13}, 8.767]$ $\mu = 3.103 \times 10^{-3}, \sigma = 0.018$	$[0, 719.684]$ $\mu = 134.224, \sigma = 79.286$	$[0.001, 1.000]$ $\mu = 0.722, \sigma = 0.289$	$[0,719.684]$ $\mu = 0.493, \sigma = 4.569$

Table 5 Results in different test sets of using two different methods for Mel to WORLD conversion

Test sets	Methods	MAE of spectral envelope	MAE of F0	MAE of aperiodicity	Cosine Similarity of F0	Global MAE
LJSpeech	Converter	$\mathbf{(1.002 \pm 0.148) \times 10^{-3}}$	$\mathbf{16.815 \pm 1.822}$	$\mathbf{(4.478 \pm 0.267) \times 10^{-2}}$	$\mathbf{0.954 \pm 0.007}$	$\mathbf{(3.924 \pm 0.303) \times 10^{-2}}$
	Time domain	$(1.403 \pm 0.174) \times 10^{-3}$	17.556 ± 2.083	$(6.805 \pm 0.329) \times 10^{-2}$	0.941 ± 0.009	$(5.179 \pm 0.342) \times 10^{-2}$
LibriTTS female speaker	Converter	$\mathbf{(1.167 \pm 0.413) \times 10^{-3}}$	38.269 ± 5.503	$\mathbf{(6.876 \pm 0.629) \times 10^{-2}}$	0.919 ± 0.014	$(7.219 \pm 0.814) \times 10^{-2}$
	Time domain	$(1.344 \pm 0.405) \times 10^{-3}$	26.545 ± 3.412	$(8.073 \pm 0.523) \times 10^{-2}$	0.923 ± 0.011	$(6.685 \pm 0.570) \times 10^{-2}$
LibriTTS male speaker	Converter	$\mathbf{(0.774 \pm 0.180) \times 10^{-3}}$	$\mathbf{19.209 \pm 1.957}$	$\mathbf{(4.966 \pm 0.395) \times 10^{-2}}$	$\mathbf{0.937 \pm 0.014}$	$\mathbf{(4.390 \pm 0.360) \times 10^{-2}}$
	Time domain	$(1.348 \pm 0.248) \times 10^{-3}$	19.556 ± 2.420	$(9.309 \pm 0.670) \times 10^{-2}$	0.906 ± 0.019	$(6.622 \pm 0.529) \times 10^{-2}$
TTS speech	Converter	$(0.807 \pm 0.066) \times 10^{-3}$	20.802 ± 2.123	$(9.300 \pm 0.607) \times 10^{-2}$	0.955 ± 0.007	$(6.711 \pm 0.485) \times 10^{-2}$
	Time domain	N/A	N/A	N/A	N/A	N/A

In the 'Methods' section of Table 5, 'Converter' means that we apply our converter to do the feature conversion directly and 'Time domain' means that we first synthesize speech utterances from mel spectrogram by Waveglow and then do the feature extraction by WORLD vocoder to finish the whole conversion process. The transformed features obtained by the two methods are then compared with the ground truth features, i.e., the target WORLD features extracted directly from natural speech, to calculate MAE and cosine similarity of F0 contours.

As we can see, in comparison with the evaluation results of going back to and processing in the time domain, our converter has a lower MAE between the converted features and ground truth features, and the converted F0 contour by our converter has a higher cosine similarity than the time domain method on the test sets of LJSpeech and LibriTTS male speaker. On LibriTTS female speaker test set, the performances of two methods are close to each other. In the TTS speech test set, the real natural recorded speech does not exist, and so the time domain method is not available for contrast. In this case, we only

fill in the table with the comparison results between the converted features output by the converter and the so-called ‘target WORLD features’. Note that a bold-faced result indicates that the corresponding method is superior to the other method in this measure and this convention is used in the following tables.

There are still differences in the performance between test sets. Since we use the speech data of a single female speaker in LJSpeech and the TTS model is also trained by using this dataset, the performance on LJSpeech test set and TTS speech is better than on two speakers on LibriTTS, which implies that our converter can achieve better generalization if we use more speech data from more speakers, male, female, or with different English accents for data augmentation.

For the WORLD features, a comparison on F0 contours is easier to visualize than comparisons on spectral envelopes and on aperiodicities. Therefore, in visualizing the effect of conversion, we provide two spectral envelope comparisons in Figs. 11 and 12, but mainly compare F0 contours in Figs. 13-18.

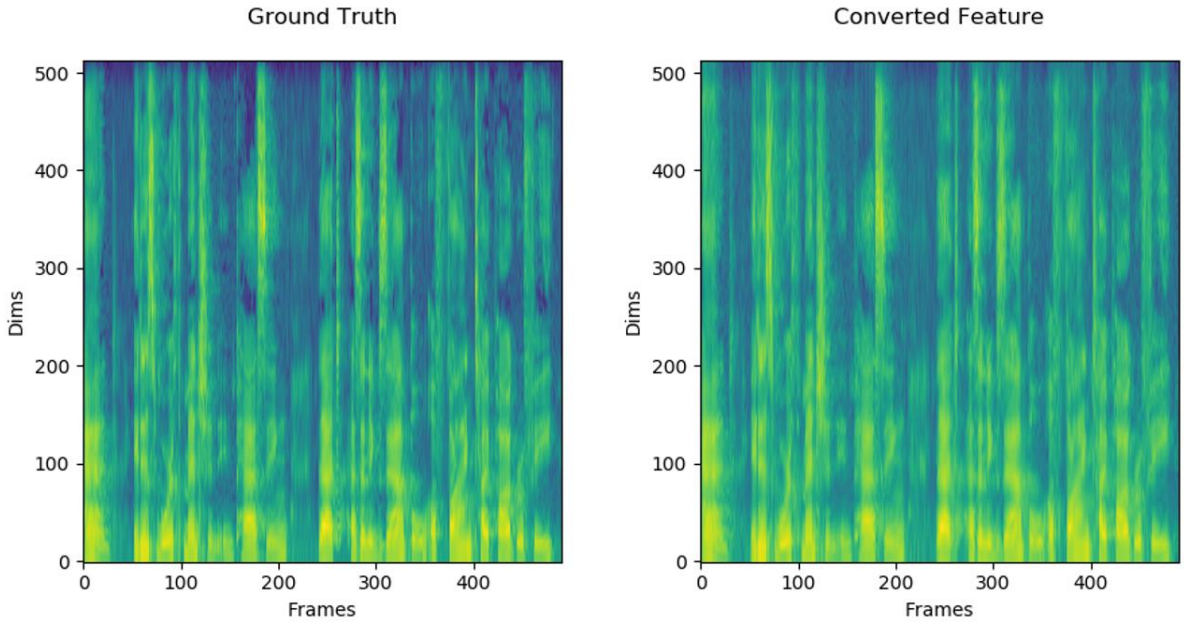

Figure 11 Spectral envelope comparison of LJ001-0006

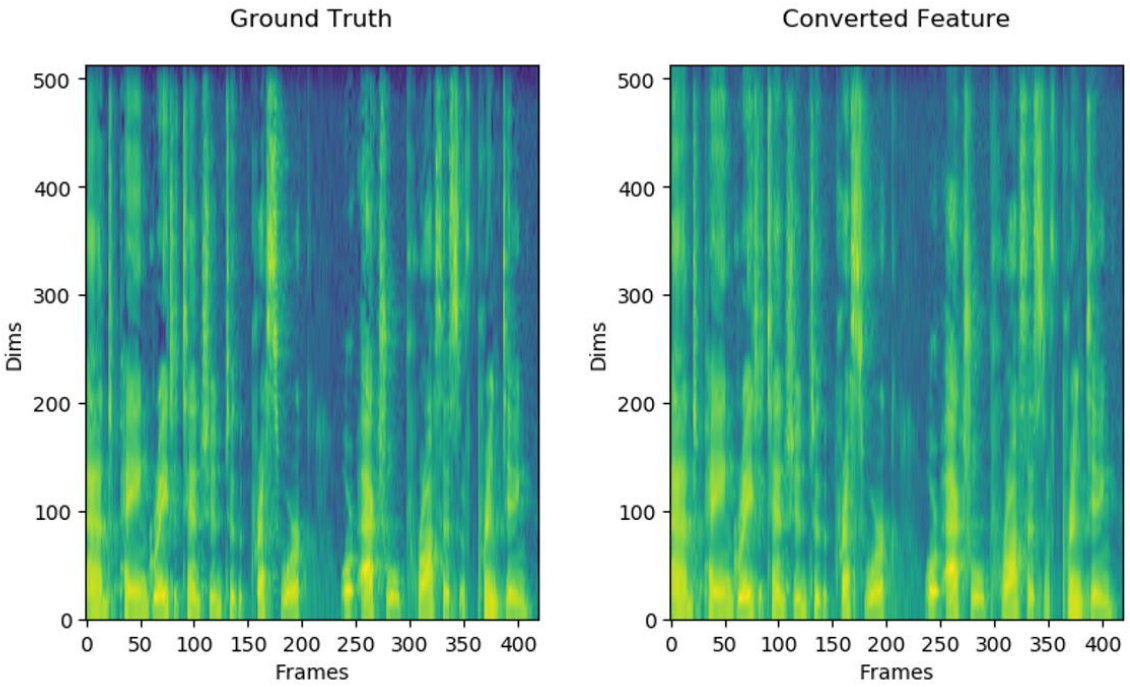

Figure 12 Spectral envelope comparison of LJ001-0051

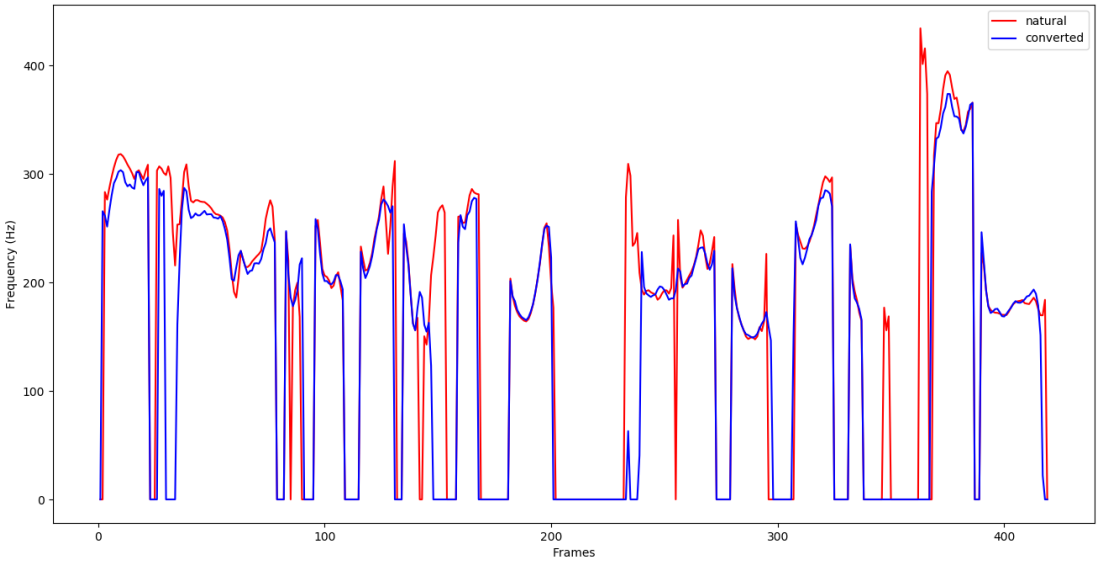

Figure 13 F0 contour comparison of LJ001-0051

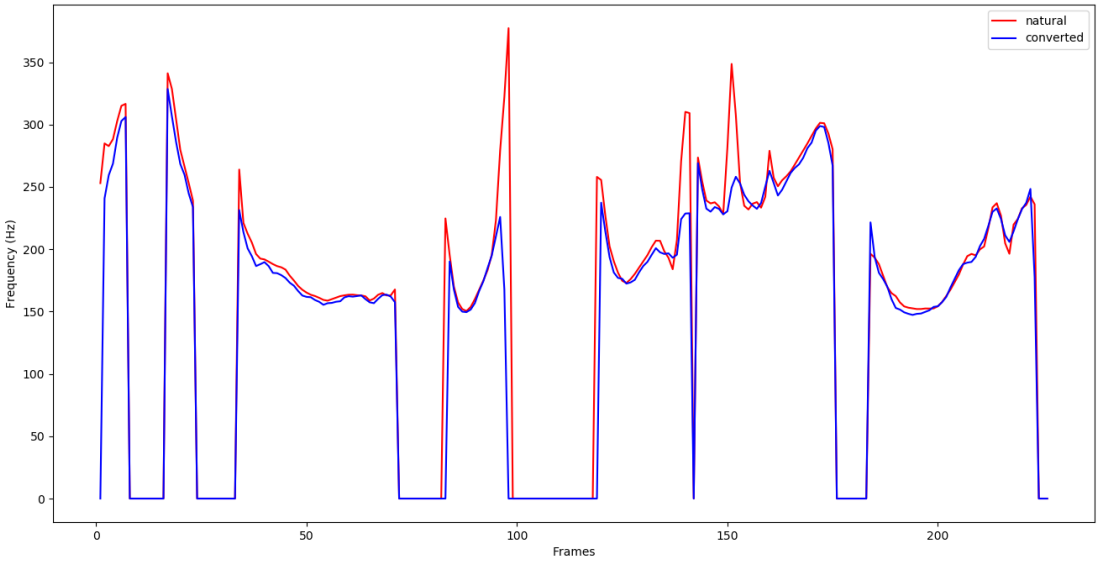

Figure 14 F0 contour comparison of LJ045-0096

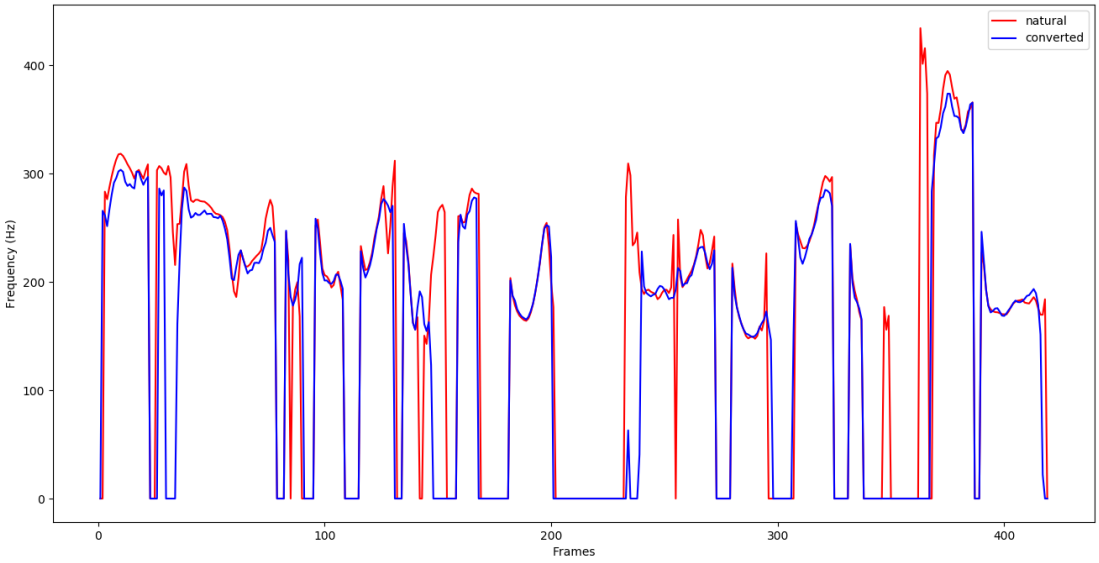

Figure 15 F0 contour comparison of LJ050-0118

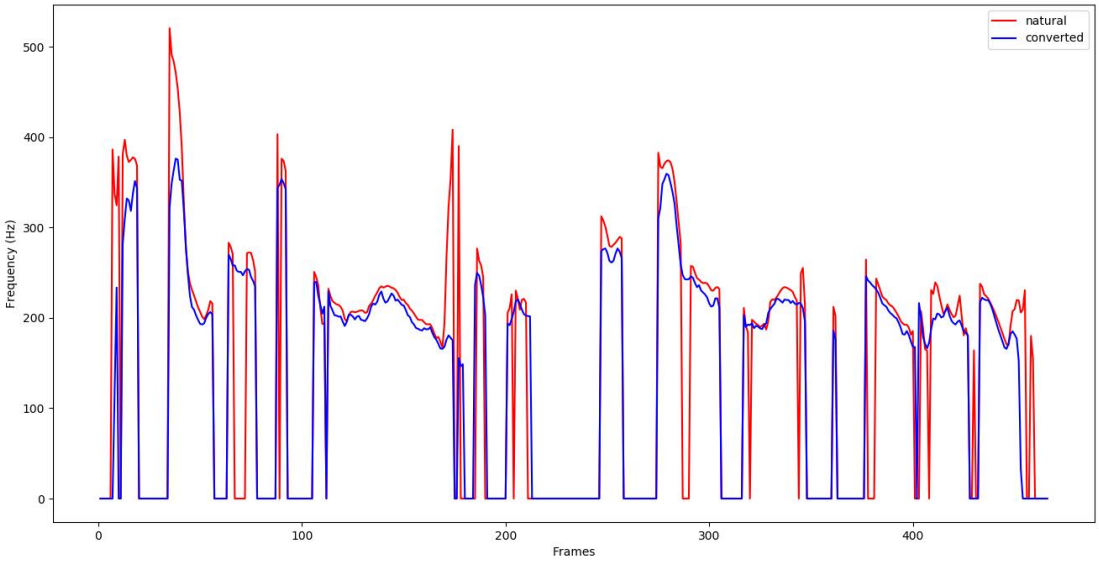

Figure 16 F0 contour comparison of LibriTTS female speaker

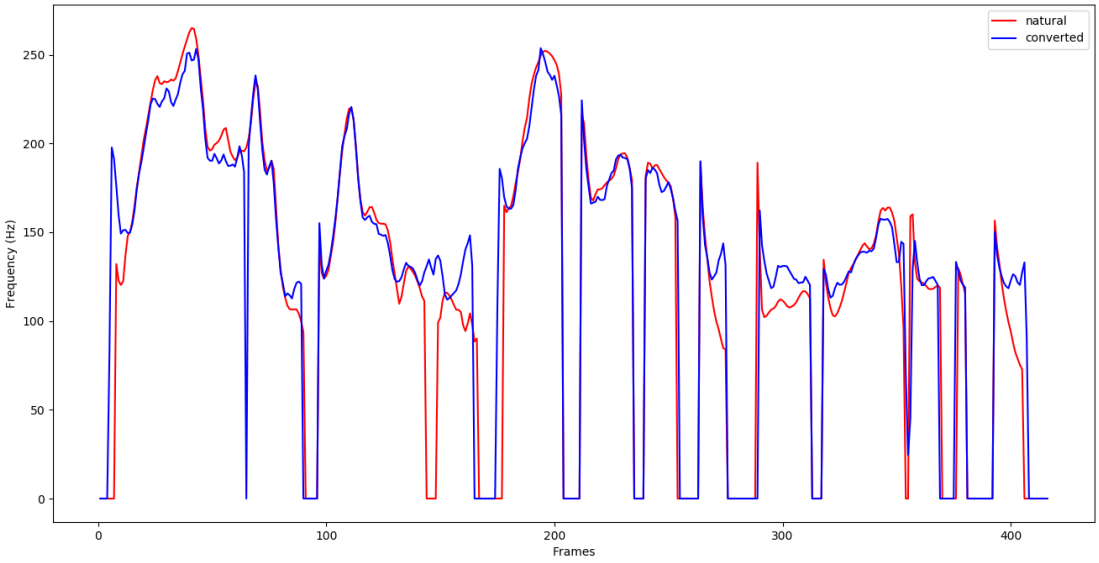

Figure 17 F0 contour comparison of LibriTTS male speaker

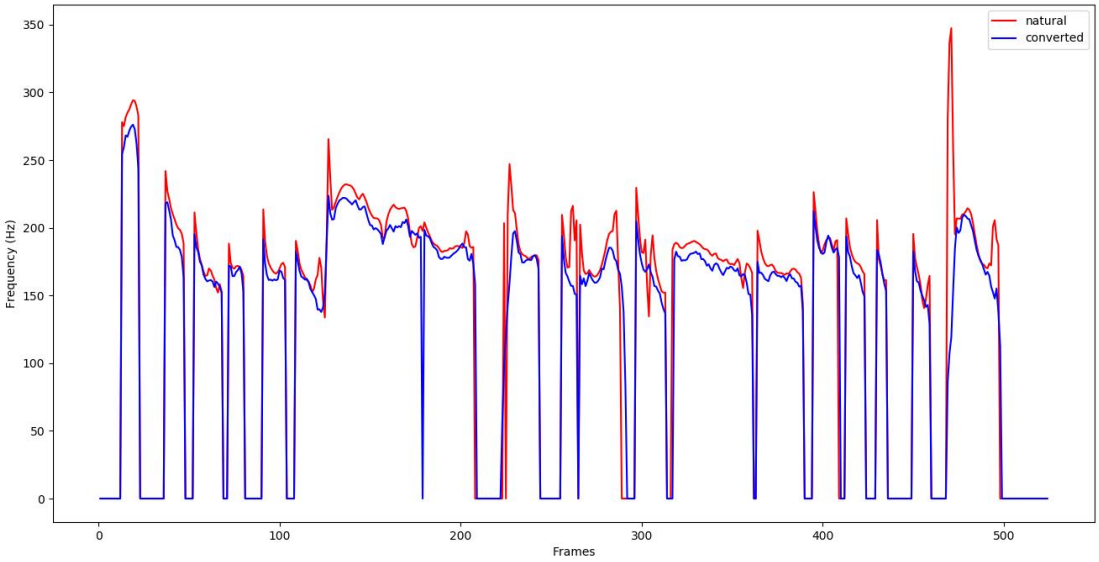

Figure 18 F0 contour comparison of TTS speech

4.3 WORLD to Mel feature converter evaluation

For WORLD to Mel feature converter, we still use the same test sets. Since the converter only outputs mel spectrogram, we use MAE of mel spectrogram for evaluation. The value distribution of the target mel spectrogram in different test sets is listed in Table 6.

Table 6 Value distribution of the target mel spectrogram in different test sets

Test sets	**Value distribution of target mel spectrogram**
LJSpeech	$[-11.513, 1.743]$ $\mu = -5.525, \sigma = 1.380$
LibriTTS female speaker	$[-11.513, 1.229]$ $\mu = -5.408, \sigma = 1.512$
LibriTTS male speaker	$[-11.496, 1.436]$ $\mu = -5.250, \sigma = 3.449$
TTS speech	$[-11.830, 1.467]$ $\mu = -5.588, \sigma = 1.744$

Table 7 Results in different test sets of using two different methods for WORLD to Mel feature conversion

Test sets	**MAE of converted Mel by time domain method**	**MAE of converted mel by our converter**
LJSpeech	0.556 ± 0.017	$\mathbf{0.241 \pm 0.011}$
LibriTTS female speaker	0.615 ± 0.028	$\mathbf{0.381 \pm 0.028}$
LibriTTS male speaker	0.297 ± 0.008	$\mathbf{0.234 \pm 0.009}$
TTS speech	N/A	0.234 ± 0.003

In Table 7, we show results of feature conversion in the reverse direction , here 'converter' is our WORLD to Mel converter, and 'time domain method' means that we first synthesize speech utterances by WORLD vocoder from WORLD features and then extract the mel spectrogram from the resynthesized speech to finish the whole conversion process. The converted mel spectrogram is compared with the target mel spectrogram extracted directly from natural speech, and MAE is calculated for each method.

According to the results, our WORLD to Mel feature converter also has a better performance than the method in time domain over all the test sets that are compared. But like the Mel to WORLD feature converter, more speech data from different speakers are still needed for further improving the conversion performance. In the following, 5 pairs of ground-truth and converted mel spectrogram plots are provided in Figs. 19-23.

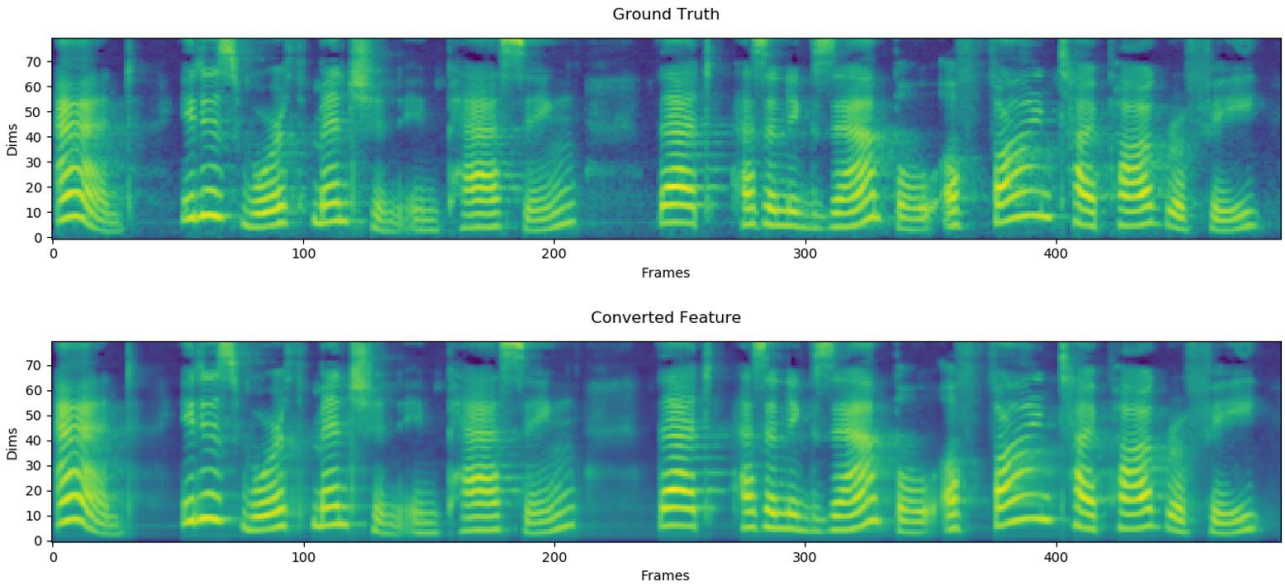

Figure 19 Mel spectrogram comparison of LJ001-0006

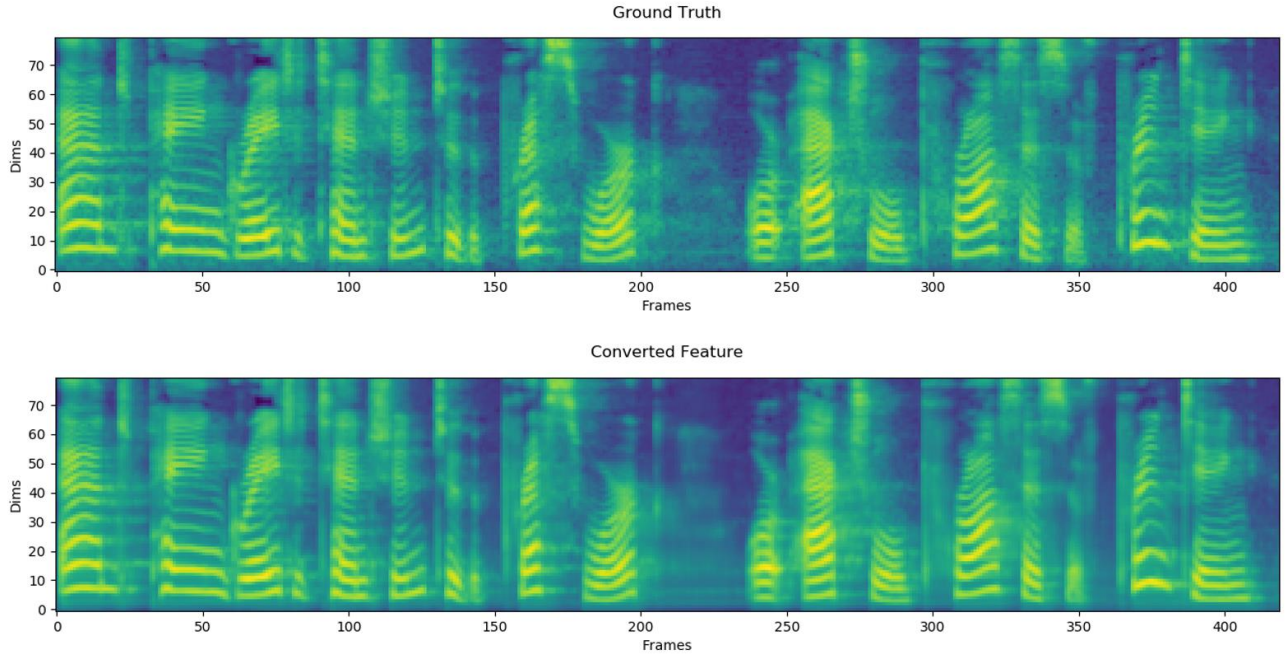

Figure 20 Mel spectrogram comparison of LJ001-0051

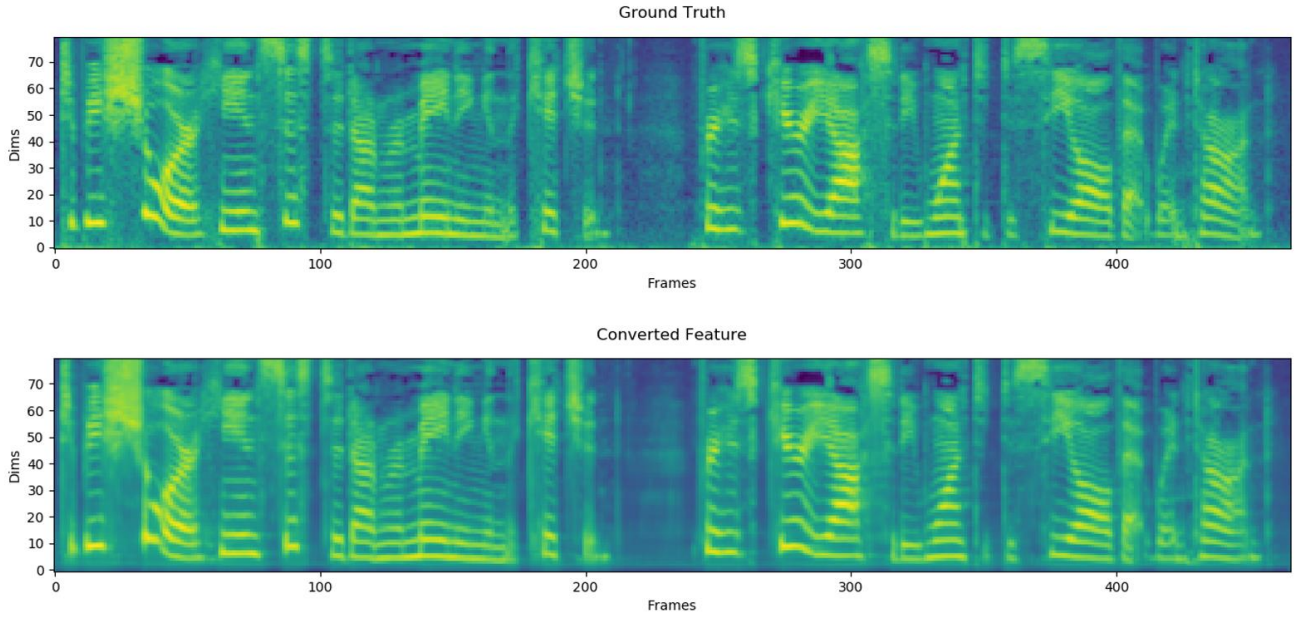

Figure 21 Mel spectrogram comparison of LibriTTS female speaker

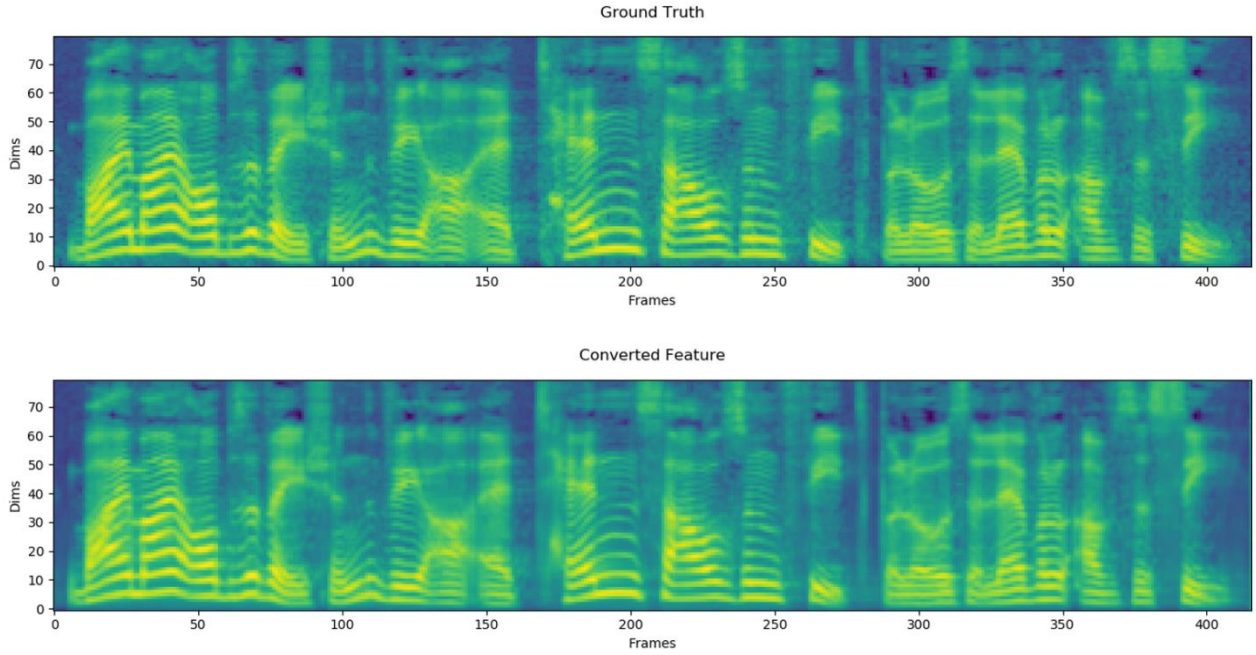

Figure 22 Mel spectrogram comparison of LibriTTS male speaker

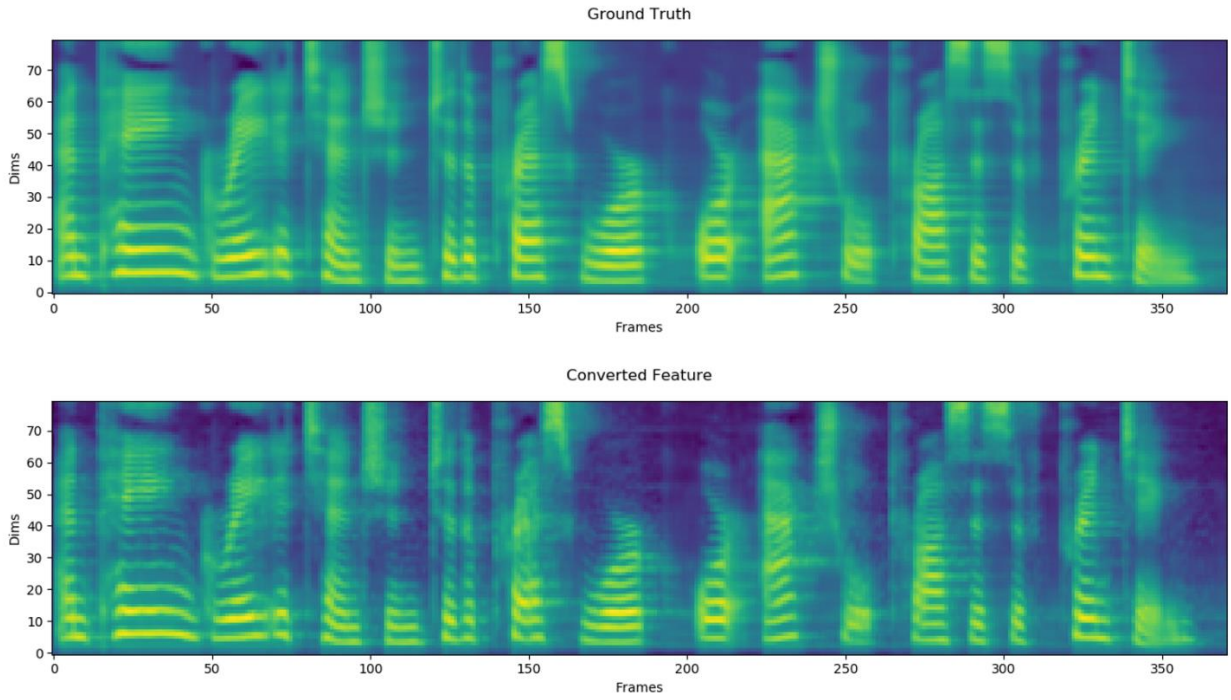

Figure 23 Mel spectrogram comparison of TTS speech

4.4 Investigation on other architectures

As a comparison, other possible architectures for the Mel to WORLD feature converter are evaluated on LJSpeech test set and the speech data of a male speaker from LibriTTS. In Tables 8 and 9, Unet+ResPath+LT is the proposed architecture used for our feature converter.

Basic Linear Transformation is the architecture of using a simple transformation matrix to convert Mel features to WORLD features through a matrix multiplication.

LT with hidden layers means that we add some hidden layers with activation functions to the basic linear transformation architecture which becomes an architecture like the fully connected network for nonlinear feature transformation.

Unet+LT is created by removing the Res Path from Unet+ResPath+LT. And we simply concatenate the feature maps from the encoder stages to the decoder stages on skip connections.

Table 8 Results in LJSpeech test set of different architectures for Mel to WORLD feature converter

Architecture	**MAE of spectral envelope**	**MAE of F0**	**MAE of aperiodicity**	**Cosine Similarity of F0**	**Global MAE**
Basic Linear Transformation	$(4.058 \pm 0.490) \times 10^{-3}$	75.026 ± 6.206	$(8.364 \pm 0.389) \times 10^{-2}$	0.833 ± 0.012	$(11.686 \pm 0.709) \times 10^{-2}$
LT with hidden layers	$(1.986 \pm 0.233) \times 10^{-3}$	22.421 ± 2.104	$(5.777 \pm 0.314) \times 10^{-2}$	0.930 ± 0.008	$(5.168 \pm 0.347) \times 10^{-2}$
Unet+LT	$(1.255 \pm 0.162) \times 10^{-3}$	17.943 ± 1.923	$(4.542 \pm 0.286) \times 10^{-2}$	0.950 ± 0.008	$(4.079 \pm 0.322) \times 10^{-2}$
Unet+Res Path +LT	$\mathbf{(1.002 \pm 0.148) \times 10^{-3}}$	$\mathbf{16.815 \pm 1.822}$	$\mathbf{(4.478 \pm 0.267) \times 10^{-2}}$	$\mathbf{0.954 \pm 0.007}$	$\mathbf{(3.924 \pm 0.303) \times 10^{-2}}$

LT stands for linear transformation.

Table 9 Results in male speaker of LibriTTS test set of different architectures for Mel to WORLD feature converter

Architecture	**MAE of spectral envelope**	**MAE of F0**	**MAE of aperiodicity**	**Cosine Similarity of F0**	**Global MAE**
Basic LT	$(3.386 \pm 0.701) \times 10^{-3}$	69.942 ± 3.954	$(13.929 \pm 0.787) \times 10^{-2}$	0.704 ± 0.040	$(13.937 \pm 0.683) \times 10^{-2}$
LT with hidden layers	$(1.907 \pm 0.405) \times 10^{-3}$	44.416 ± 4.315	$(6.611 \pm 0.457) \times 10^{-2}$	0.897 ± 0.013	$(7.723 \pm 0.561) \times 10^{-2}$
Unet+LT	$(1.052 \pm 0.235) \times 10^{-3}$	22.367 ± 2.293	$(4.886 \pm 0.378) \times 10^{-2}$	0.934 ± 0.014	$(4.671 \pm 0.375) \times 10^{-2}$
Unet+Res Path +LT	$\mathbf{(0.774 \pm 0.180) \times 10^{-3}}$	$\mathbf{19.209 \pm 1.957}$	$\mathbf{(4.966 \pm 0.395) \times 10^{-2}}$	$\mathbf{0.937 \pm 0.014}$	$\mathbf{(4.390 \pm 0.360) \times 10^{-2}}$

LT stands for linear transformation.

For each of the architectures, we choose the best model when the loss curve converges, or use early stopping before overfitting to select the best model. According to the results in Table 8 and Table 9, Unet+ResPath+LT leads in all of the objective metrics, which proves that our enhanced architecture based on U-net has delivered the best overall performance for conversion between mel spectral and World features.

4.5 Speech Quality

We have listened to all the speech utterances resynthesized by converted features in the test sets. These speech utterances are of very high quality and almost indistinguishable from the original audio clips. In the future, we will use Mean Opinion Score (MOS) to carry out a formal perceptual evaluation test in terms of the quality of the feature-converted speech. Here, for purpose of comparison, 5 pairs of speech waveforms of the natural or TTS speech and the corresponding resynthesized speech from feature conversion are provided in Figs. 24-28. Whether the conversion is from mel spectrogram to WORLD feature, or from WORLD feature to Mel, as shown in these figures, the resynthesized waveforms are all very clear and are nearly identical to the corresponding original waveforms, without added artifacts or noise.

All the waveforms on the top are natural recorded speech or TTS speech, while those on the bottom are synthesized by converted features.

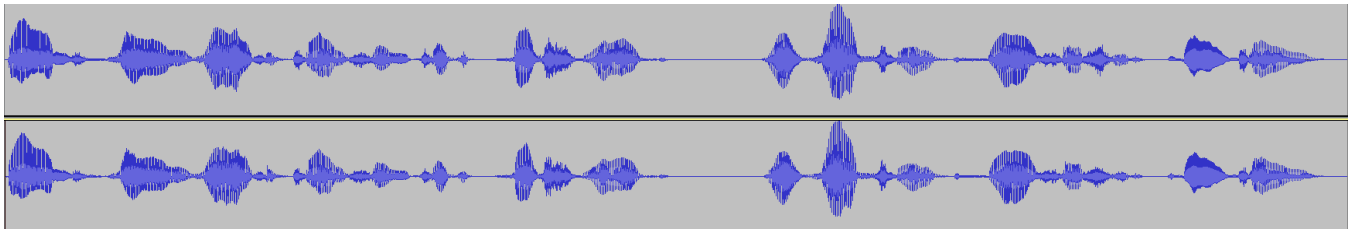

Figure 24 Waveform comparison of LJ001-0051

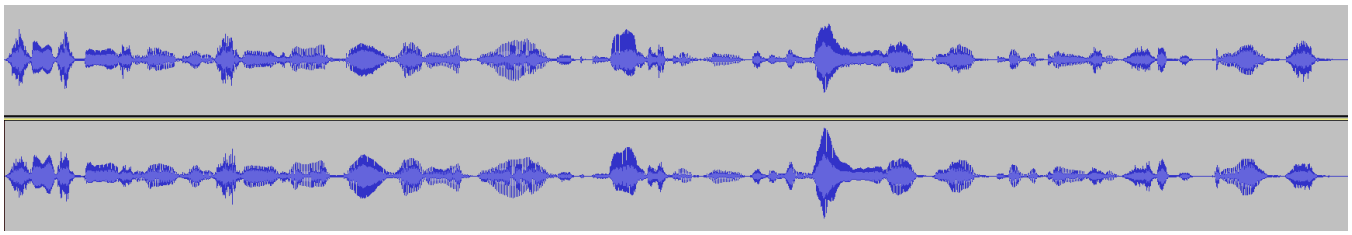

Figure 25 Waveform comparison of LJ050-0118

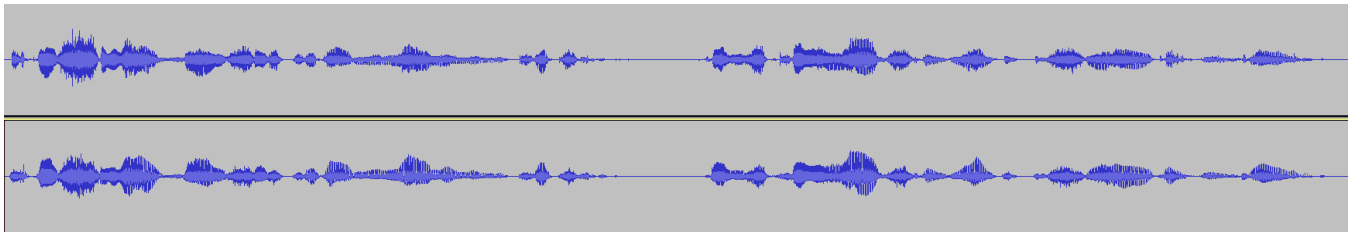

Figure 26 Waveform comparison of LibriTTS female speaker

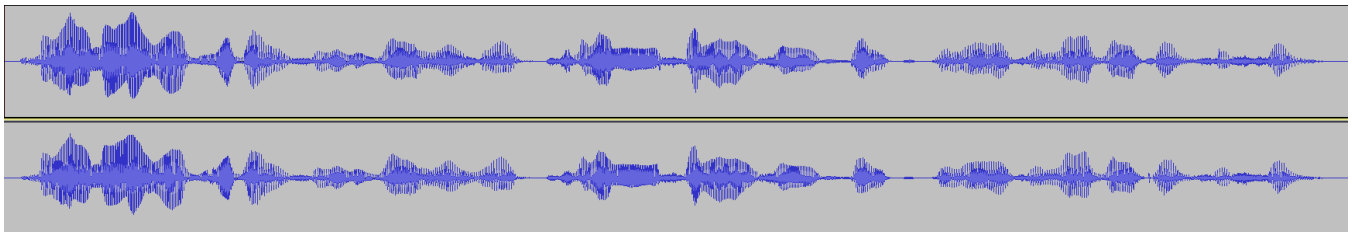

Figure 27 Waveform comparison of LibriTTS male speaker

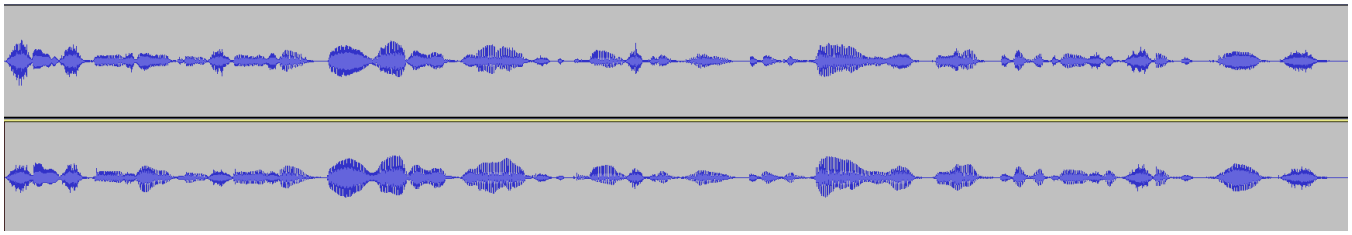

Figure 28 Waveform comparison of TTS speech

5. CONCLUSION AND FUTURE WORK

In this work, we started by introducing the two types of widely used features used in speech analysis and synthesis, i.e., mel spectrogram and WORLD vocoder features. In some application scenarios, for example, if mel spectrograms are output by a neural TTS system like Tacotron2, and we want to analyze the TTS output by some components of WORLD features or make adjustments on these feature components followed by resynthesizing speech by WORLD vocoder, then performing conversion on features is needed. Different from synthesizing the source speech features to the time domain waveforms and then analyzing the target features again, we have successfully designed a novel speech feature converter to accomplish the conversion directly at the feature level based on U-net. Furthermore, with the addition of Res Path and linear transformations on the skip connections to our architecture, this enhanced deep convolutional neural network is able to perform bidirectional feature conversions between mel spectrogram and WORLD features.

Through a series of experimental evaluations, the converted features are found to be very close to the target features and the errors are small. High-quality speech can be resynthesized accurately from the converted features. Our converter not only saves the computation cost of going back to the wave domain with vocoders like Waveglow or WORLD, but it also helps avoid adding in artifacts or noise in the conversion. In comparison with other basic architectures of feature conversion, our proposed architecture is found to be superior, and the enhancement we made to U-Net does bring overall performance improvements. This feature converter is flexible with a good generalization ability to unseen speakers and new datasets. In a nutshell, this proposed speech feature converter is the method that we are looking for.

In the future, we plan to use this proposed speech feature conversion architecture for more downstream applications, such as voice conversion between different speakers and intonation conversion between different speech styles, and intonation enhancement on TTS speech that suffers from

oversmoothing. For all these purposes, having the capability of seamlessly converting features between different domains is highly desirable. With further improvements on our network design, we look forward to implementing additional functions to fulfill more challenging tasks.

6. REFERENCES

[1] Masanori Morise, Fumiya Yokomori, Kenji Ozawa, "WORLD: A Vocoder-Based High-Quality Speech Synthesis System for Real-Time Applications," IEICE Transactions on Information and Systems, July 2016, doi: 10.1587/transinf.2015EDP7457

[2] J. Shen *et al*., "Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions," 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, 2018, pp. 4779-4783, doi: 10.1109/ICASSP.2018.8461368.

[3] R. Prenger, R. Valle and B. Catanzaro, "Waveglow: A Flow-based Generative Network for Speech Synthesis," ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, United Kingdom, 2019, pp. 3617-3621, doi: 10.1109/ICASSP.2019.8683143.

[4] C. Wang, C. Xu, C. Wang and D. Tao, "Perceptual Adversarial Networks for Image-to-Image Transformation," in IEEE Transactions on Image Processing, vol. 27, no. 8, pp. 4066-4079, Aug. 2018, doi: 10.1109/TIP.2018.2836316.

[5] Ronneberger, Olaf, Fischer, Philipp, & Brox, Thomas, "U-net: Convolutional networks for biomedical image segmentation", In International conference on medical image computing and computer-assisted intervention, 2015, page 234--241. Springer International Publishing, Cham, doi: 10.1007/978-3-319-24574-4_28

[6] Keith Ito, Linda Johnson, "The LJ Speech Dataset," [online]. Available: "https://keithito.com/LJ-Speech-Dataset/" [accessed 2017]

[7] M. Morise, "Harvest: A high-performance fundamental frequency estimator from speech signals", in Proc. INTERSPEECH 2017, pp. 2321–2325, 2017. Available: "http://www.isca-speech.org/archive/Interspeech_2017/abstracts/0068.html"

[8] M. Morise, "CheapTrick, a spectral envelope estimator for high-quality speech synthesis", Speech Communication, vol. 67, pp. 1-7, March 2015. Available: "http://www.sciencedirect.com/science/article/pii/S0167639314000697"

[9] M. Morise, "D4C, a band-aperiodicity estimator for high-quality speech synthesis", Speech Communication, vol. 84, pp. 57-65, Nov. 2016. Available: http://www.sciencedirect.com/science/article/pii/S0167639316300413

[10] M. D. Zeiler, D. Krishnan, G. W. Taylor and R. Fergus, "Deconvolutional networks," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, 2010, pp. 2528-2535, doi: 10.1109/CVPR.2010.5539957.

[11] LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. Nature 521, 436–444 (2015). https://doi.org/10.1038/nature14539

[12] Nabil Ibtehaz, M. Sohel Rahman, "MultiResUNet : Rethinking the U-Net Architecture for Multimodal Biomedical Image Segmentation" Neural Networks 121, August 2019. doi: 10.1016/j.neunet.2019.08.025.

[13] Szegedy Christian, Ioffe Sergey and Vanhoucke Vincent "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning."(2017)

[14] Ioffe, Sergey, & Szegedy, Christian Batch normalization: Accelerating deep network training by reducing internal covariate shift, arXiv preprint arXiv:1502.03167.
[15] Zen, H., Dang, V., Clark, R., Zhang, Y., Weiss, R.J., Jia, Y., Chen, Z., Wu, Y. (2019) “LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech”, Proc. Interspeech 2019, 1526-1530, doi: 10.21437/Interspeech.2019-2441.