VTTS: VISUAL-TEXT TO SPEECH

Yoshifumi Nakano1, Takaaki Saeki1, Shinnosuke Takamichi1, Katsuhito Sudoh2, Hiroshi Saruwatari1

1The University of Tokyo, Japan. 2Nara Institute of Science and Technology, Japan.

ABSTRACT

This paper proposes a visual-text to speech (vTTS) method, a method for synthesizing speech directly from visual text (i.e., text as an image). vTTS can use visual features in visual text that should be important for speech synthesis such as emphasis and radicals (components in Chinese characters), but they are not available in conventional TTS using discrete symbols as the input. The proposed vTTS method extracts visual features with a convolutional neural network and then generates acoustic features with a non-autoregressive model in an end-to-end manner. Experimental results show that 1) the vTTS method is capable of generating speech with naturalness comparable to or better than a conventional TTS, 2) it can transfer emphasis and emotion attributes in visual text to speech without additional labels and architectures, and 3) it can synthesize more natural and intelligible speech from unseen and rare characters than conventional TTS.

Index Terms—speech synthesis, visual text, visual-text to speech, visual feature

1. INTRODUCTION

Text to speech (TTS) is a task for synthesizing speech from arbitrary text. Along with the development of deep neural networks (DNNs), neural TTS methods have succeeded in synthesizing speech with the same naturalness as human speech [1, 2, 3, 4]. Generally, TTS first converts each character (or each phoneme based on linguistic knowledge) into a discrete symbol and synthesizes a speech waveform from it (Figure 1(a)).

However, when we read a sentence out loud, we do not view each character as a discrete symbol but rather use the visual information of the character. For example, when we read a phonogram (a character representing a speech sound), the character or the combination of sub-characters (components of one character) determines the reading. In addition, we can recognize attributes in visual text and reflect them in speech. For example, when reading a textbook, we can recognize an underlined (or bold, italic) word as important and read the word emphatically [5]. Another example is typeface, which sometimes evokes a certain emotion [6, 7]. Advertisements and comics utilize this to convey desired emotions to readers [8]. From these facts, it is more appropriate to use the visual information of text to synthesize speech.

In this paper, we propose a new task called visual-text to speech (vTTS), which generates a speech waveform from visual text (Figure 1(b)). We propose a method for vTTS, and the method consists of a visual feature extractor and FastSpeech2-inspired model. The visual feature extractor extracts visual features from visual text using a convolutional neural network (CNN). The FastSpeech2-inspired model predicts acoustic features from the extracted visual features.

Our vTTS method has three advantages over conventional TTS. First, our vTTS method can directly reflect emphasis and emotion attributes in speech by recognizing the difference of the typeface of visual text itself whereas the basic TTS methods require additional labels and architectures [9, 10]. Second, our vTTS method does not require a vocabulary (lookup table) of characters, unlike the basic TTS. A pre-determined fixed-size vocabulary cannot be used to embed out-of-vocabulary (OOV) characters (typically, “unknown” symbol is used). Our vTTS method avoids this and obtains visual features from OOV characters. Last, for some languages, our model can efficiently predict the reading from character compositionality (the reading or meaning is determined by the structure of the character). Even if rare and OOV characters emerge, the vTTS method can synthesize more natural and intelligible speech utilizing character compositionality than the basic TTS.

We compared our vTTS method with the basic TTS.
method in three phonetic languages: Japanese (Hiragana), Korean (Hangul), and English (Roman Alphabet) and then revealed the following properties.

- Our vTTS method is comparable to or better than the basic TTS method in terms of naturalness of synthetic speech.
- It can transfer emphasis and emotion attributes in visual text to speech without additional labels and architectures.
- It can synthesize more natural and intelligible speech from OOV and rare phonograms than the basic TTS.

Our implementation\(^1\) and corpus\(^2\) are open-sourced on the project pages for reproducibility and future work.

2. RELATED WORK

2.1. Text to speech (TTS)

As the basic TTS model, we use FastSpeech2 [11], a well-known non-autoregressive model. FastSpeech2 consists of three components: an encoder, variance adapter, and decoder. The input of the original FastSpeech2 is a phoneme sequence, but we use a character sequence instead for fair comparison with the proposed method. Section 4.1 notes the results of preliminary experiments on this change.

Some studies use information other than text such as reference voices [12], face images [13], and articulation [14]. These methods have an additional encoder for such information. However, similar architecture is not suitable for vTTS. Since visual text is not discrete as described in Section 1, text and visual information should not be treated separately. Therefore, we propose a model that jointly treats text and visual information in Section 3.

A simple implementation for vTTS is to concatenate optical character recognition (OCR) and TTS. Strong drawbacks of this implementation are 1) OCR and TTS are isolated; OCR errors are propagated into TTS during inference, and 2) an additional encoder is required to obtain visual features as mentioned above. Our model avoids these and learn direct conversion from visual text to speech features. In Section 4.2.1, we compare vTTS with TTS (i.e., concatenation of OCR and TTS with the 100% accuracy of OCR) and reveals superiority of vTTS.

Although a character is an input unit of speech synthesis that does not rely on language knowledge, some studies use other units, e.g., phrase [15], word [16], subword [17], and byte [18] in descending order of length. In this paper, we use a pixel, which is a smaller input unit than ever before. We can observe a similar trend in natural language processing as described below.

\(^1\)https://revealed_after_acceptance
\(^2\)https://revealed_after_acceptance

2.2. Visual text for natural language processing (NLP)

In NLP, the input unit of language models and machine translation models has been examined. Starting with phrases [19] and words [20] and moving through subwords [21], characters [22], and sub-characters [23], some research proposes the use of pixels (i.e., visual text).

There is one study that extracted character embeddings from visual text [24]. This study utilized logogram compositionality, allowing for better processing of rare words. Similarly, there are studies that have used character compositionality in a word segmentation task [25] or a sentiment classification task [26]. Other than that, the use of sliced visual text, which is analogous to subword text symbols, leads to there being more resistance to various types of noise [27]. These studies mainly used a CNN to extract the visual information from visual text.

These studies are based on the compositionality in logograms and acquire meaning-related features. In contrast, our study aims to acquire sound-related ones contained in characters, especially in phonograms.
3. METHOD

3.1. Architecture comparison between vTTS and TTS

As for the architecture of the proposed vTTS method, we replace the character embedding layer (i.e., text feature extractor) of the original FastSpeech2 with a visual feature extractor (Figure 2(a)). This replacement allows for extracting visual features from visual text. The subsequent encoder, variance adapter, and decoder are the same as the original FastSpeech2. To investigate the ideal performance of the vTTS method, in this paper, we use artificially generated visual text rather than that in in-the-wild images (e.g., advertisements and comics).

3.2. Generating visual text

The first step is to transform text to grayscale sliced images. The overall process is illustrated in Figure 2(c). Since the original FastSpeech2 is a non-autoregressive TTS method utilizing a duration-based upsampler, we must take the temporal alignment between visual text and a speech feature sequence. Therefore, we use visual text with monospace fonts in this work. Each character is of a specified width \(w\), height \(h\), and font size \(f_s\). Therefore, characters of length \(n\) are transformed into an image whose width is \(nw\) and height is \(h\). We extract sliced images from it using sliding windows similar to a previous study [27]. The window is of a specified width \(wc\) and height \(h\) extracted at intervals \(s(=w)\). Here, \(c\) represents the number of characters in one sliced image, and adjusting \(c\) leads to a change in the preceding and following characters to be considered when obtaining visual features from each piece of visual text. For example, if \(c = 3\), convolution is performed including one character before and after. By considering the number of preceding and following characters, reading and prosody can be acquired that depend on the neighboring characters. Blank images are padded to the left and right edges to create sliced images for the number of characters. Finally, we generate \(n\) sliced images of height \(h\) and width \(wc\) from text.

3.3. Visual features

We extract visual features from the sliced images generated in Section 3.2. The basic TTS uses the output from a text feature extractor. In comparison, the vTTS method uses the output of a visual feature extractor instead. The architecture of the extractor is shown in Figure 2(b) and is inspired by a machine translation study using visual text [27]. The extractor mainly comprises 2D convolution, 2D batch normalization [28], and the activation function ReLU [29]. Finally, we apply a linear layer to the reshaped output to obtain a visual feature for each character. The visual features are fed to the FastSpeech2-inspired encoder.

The visual feature extractor utilizes the following information. Figure 3 shows examples of visual text with the following information.

- **Compositionality**: In phonetic languages (e.g., Korean), the character or combination of sub-characters determines the overall reading. The visual feature extractor acquires the correspondence between a character (or sub-character) and the reading. Therefore, even if OOV and rare characters emerge, we expect our vTTS method to be able to accurately predict the readings using the visual features of the characters or sub-characters.

- **Emphasis attribute**: The visual feature extractor distinguishes emphasized typefaces (underline, bold, italic) from normal typefaces. This enables the vTTS method to synthesize word-emphasized speech only from emphasized typefaces.

- **Emotional attribute**: The visual feature extractor acquires the correspondence between typeface and emotion, and it reflects the emotion in speech in accordance with the typeface.

4. EXPERIMENTS AND RESULTS

4.1. Experimental Setup

**Language.** We evaluated the performance of the proposed method on three phonetic languages: Japanese (Hiragana), Korean (Hangul), and English (Roman Alphabet). For Japanese, we converted Katakana and Kanji into 80 Hiragana-characters in advance. In Hiragana, one character corresponds to the specific sound. For Korean, we used 1226 Hangul-characters. In Hungul, the combination of jaamo determines the sound. For English, we applied lowercasing to facilitate training.

**Dataset.** We used the JSUT [30] (Japanese), KSS [31] (Korean), and LJSpeech [32] (English) corpora for naturalness comparison of vTTS and TTS methods as described in Section 4.2.1. The training (validation) data sizes for these languages were 8.3 (0.61), 9.0 (0.38), and 19 (0.89) hours, respectively. The test data sizes were 100 samples for all languages. We used the JECS corpus, which contains non-emphasized Japanese speech, for the emphasis experiments described in Section 4.2.2. The training, validation, and test data sizes were 0.412 hours, 0.03 hours, and 50 samples, respectively. We used the manga2voice corpus [33], which contains emotional speech, for the emotion experiment described in Section 4.2.3. The training, validation, and test data sizes were 0.062 hours, 0.003 hours, and 50 samples, respectively. Since the training data sizes of the JECS and manga2voice corpora were small, we fine-tuned a vTTS model pretrained using the JSUT corpus. We used the KSS corpus in Section 4.2.4. The data sizes were almost the same but slightly different from those in Section 4.2.1. See Section 4.2.4 for details. All audio was downsamped to 22.05 kHz. The alignment between visual text and acoustic features was obtained through forced alignment of visual text and a character se-
Table 1. Results of MOS and 95% confidence interval for naturalness. Bold means significantly better score than TTS. For all languages, vTTS was comparable to or better than TTS in naturalness.

| Lang. | TTS | vTTS |
|-------|-----|------|
|       | $c = 1$ | $c = 5$ | $c = 5$ |
| ja    | 3.41 ± 0.09 | 3.41 ± 0.09 | 3.43 ± 0.09 | 3.49 ± 0.10 |
| ko    | 3.04 ± 0.16 | 3.55 ± 0.15 | 3.18 ± 0.15 | 3.01 ± 0.15 |
| en    | 3.72 ± 0.10 | 3.69 ± 0.10 | 3.70 ± 0.11 | 3.71 ± 0.10 |

 sequence, and a character sequence and a acoustic feature sequence [34].

Visual text. We used pygame\(^3\) to generate visual text from text. For Japanese and English, we used the IPA typeface; for Korean, the Gowun Batang typeface. Visual text of $w = 30$ and $h = 30$ was used for all languages. $fs$ was set to 15 for Japanese and Korean and 20 for English. This is the result of setting the text to be the appropriate size for an image.

Model configuration. The model size and hyperparameters of the original FastSpeech2 and FastSpeech2-inspired architecture in the vTTS method followed the open-source implementation [35]. In the visual feature extractor, the 2D convolution used only one channel, and there was a padding of 1, kernel size of 3, and stride of 1 [27]. The dimension of the output of the visual feature extractor was set to 256, the same as the encoder hidden dimension of FastSpeech2. In the variance adapter, we used character-level pitch and energy features, and we used F0 extracted by the WORLD vocoder [?] for the pitch features. We generated speech waveforms from mel spectrograms using the pretrained HiFi-GAN [36].

Preliminary experiment. As described in Section 2.1, we used characters instead of phonemes as input unit for FastSpeech2. We conducted a subjective evaluation of this substitution under the same conditions as in Section 4.2.1. The results showed that there was no significant difference in naturalness in Japanese and English. On the other hand, the use of phonemes was better than that of characters and visual text in Korean. This comparison is not the main scope of this paper, but deep investigation will be our future work.

4.2. Results

4.2.1. Naturalness comparison between vTTS and TTS

To compare our vTTS method with the conventional TTS method, we conducted a mean opinion score (MOS) evaluation on the naturalness of the synthetic speech. We used three settings for vTTS, $c = 1, 3, 5$, to investigate the impact of the window size. Native speakers of each language participated in the evaluation. 150 and 200 speakers recruited using our crowdsourcing system listened to 20 speech utterances in Japanese and English, respectively. In contrast, 10 speakers recruited by snow-ball sampling\(^6\) listened to 160 speech utterances in Korean. The text content was kept consistent among different methods so that all listeners examined only naturalness without other interfering factors.

The results are shown in Table 1. For all languages, the naturalness of vTTS was comparable to or better than that of TTS. Comparing the highest score of vTTS ($c = 5$ for Japanese and English, $c = 1$ for Korean) with TTS, although there was no significant difference for Japanese and English, there was a significant difference for Korean ($p < 0.05$). This means that the extraction of visual features was effective for Hangul, whose readings are determined by the combination of sub-characters.

It can also be seen that the appropriate window size differed depending on the language. For English, there was no significant difference between $c = 1, 3, 5$. However, for Japanese, the naturalness with a sliding window of $c = 5$ was slightly better than that with $c = 1$ ($p < 0.10$), while for Korean, $c = 1$ was significantly better than $c = 5$ ($p < 0.001$). One possible reason is that the number of phonemes expressed by one character differs depending on the language. One Roman Alphabet usually corresponds to within one phoneme, one Japanese Hiragana character to within two phonemes, while one Korean Hangul character may correspond to three or more phonemes. This difference is expected to have caused the aforementioned differences. The following experiments used the best $c$ value for each language, i.e., $c = 5$ for Japanese and English and $c = 1$ for Korean.

4.2.2. Emphasis attribute

We evaluated whether our vTTS method can transfer the emphasis attribute in visual text to speech. We trained the vTTS model using pairs of word-emphasized speech and visual text. We used three types of word emphasis methods (underline, bold, and italic) as shown in Figure 3(b). We trained the vTTS model using each type, i.e., three vTTS models were trained. The trained models try to synthesize speech with emphasis placed on specified words. We prepared synthetic speech of not only emphasis groups (“Emphasis (*)”) but also control groups (“Control (*)”). Their difference was in the input visual text: word-emphasized visual text in the emphasis groups and non-emphasized text in the control groups.

For the listening test, listeners listened to speech randomly selected from seven kinds (three emphasis groups, three control groups, and ground-truth natural speech) of speech utterances. The listeners answered with which word was emphasized, from word (specifically, noun) candidates. 150 native speakers listened to 20 speech utterances. The average number of candidates per sentence was 3.6; the chance rate was 0.299.

\(^3\)https://www.pygame.org/news
\(^4\)https://github.com/Wataru-Nakata/FastSpeech2-JSUT
\(^5\)https://github.com/jik876/hifi-gan
\(^6\)We could not collect enough number of native listeners in our crowdsourcing system
Table 2. Accuracy of perceived emphasized words. Our method can properly reflect emphasis attribute in visual text in speech.

| Speech         | Accuracy |
|----------------|----------|
| Ground truth   | 0.960    |
| Emphasis (underline) | 0.933    |
| Emphasis (bold) | 0.898    |
| Emphasis (italic) | 0.877    |
| Control (italic) | 0.505    |
| Control (bold)  | 0.400    |
| Control (underline) | 0.381    |

The results are shown in Table 2. For all emphasis methods, the accuracy of the emphasis groups was significantly better than that of the control groups (p < 0.01). This shows that our vTTS method properly recognized the emphasis attribute in visual text. In addition, the accuracy of the emphasis groups was close to that of the ground truth. There was no significant difference between two emphasis methods (underline, bold) and the ground truth. These results show that these two emphasis methods can accurately transfer the emphasis attribute in visual text to speech, without additional emphasis labels and architectures. One possible reason for the lower accuracy of the italic input compared with the two other emphasis methods is that the shape of italic is not as similar to that of normal typeface because of geometric transformation.

4.2.3. Emotion attribute

We evaluated whether our method can transfer emotions evoked by certain typefaces to speech. We trained the vTTS model using pairs of emotional speech and typeface-decorated visual text. Before the training, we conducted preliminary experiments to find typefaces that evoke emotions. As a result, we decided to use the Koruri typeface7 and Aiharahudemokaisyo typeface8, which evoke joy and sadness, respectively. These are typefaces that more than 70 percent of evaluators thought of as corresponding emotion.

We trained one vTTS model using the above two typefaces. In the listening test, listeners listened to the synthetic speech and answered the perceived emotion. To avoid the listener from perceiving emotion from the linguistic content, the content was common between emotions, and it was selected from the “neutral” subset of the manga2voice corpus. 120 listeners participated, and each listener listened to 20 speech utterances.

The results are shown in Table 3. When listening to speech synthesized from the Koruri typeface, 79% of the listeners answered that it sounded joyful, and 89% of the listeners answered that it sounded sad when listening to speech synthesized from the Aiharahudemokaisyo typeface. These scores are comparable to TTS with emotion labels [9]. These results show that the vTTS method can accurately transfer the emotion attribute in visual text to speech, without additional emotion labels and architectures.

4.2.4. Speech synthesis from rare and OOV characters

Finally, we investigated the ability of the vTTS method to synthesize speech from OOV and rare characters. A good example for this purpose is Korean because the character compositionality determines the reading. We designed three test sets: in-vocab, rare, and OOV. The “in-vocab” set consisted of only characters appearing more than three times in the training data. Sentences in the “rare” set included characters appearing less than three times in the training data, and those in the “OOV” set included OOV characters. The numbers of rare and OOV characters are 218 and 260, respectively. Each test set consisted of 50 sentences. For comparison, we trained the TTS model under the same conditions as for the vTTS model. The TTS model had an “unknown” symbol, and an OOV character was encoded in the symbol. We conducted a MOS test on naturalness and a dictation test. The listener listened to speech utterances randomly selected from all test sets and then answered the speech naturalness in the MOS test and dictated the text in the dictation test. The character error rate (CER) was used in the dictation test. 10 listeners participated in each test. Each listener listened to 160 and 64 speech utterances in the MOS test and dictation test, respectively.

The MOS results on naturalness are shown in Table 4. There was a significant difference between vTTS and TTS in the “in-vocab,” “rare,” and “OOV” sets (p < 0.001). For

Table 3. Confusion matrix between reference (true_*) and perceived (pred_*) emotion. Our method can properly reflect emotion attributes in visual text in synthetic speech.

| true_happy | pred_happy | pred_sad |
|------------|------------|----------|
| 0.795      | 0.205      |          |
| 0.114      | 0.886      |          |

Table 4. Results of MOS and 95% confidence interval on naturalness. Δ denotes MOS decrease from “in-vocab.” Our method can synthesize more natural speech than TTS.

|         | in-vocab | rare (Δ) | OOV (Δ) |
|---------|----------|----------|---------|
| TTS     | 3.29 ± 0.16 | 2.32 ± 0.16 (−0.97) | 2.31 ± 0.20 (−0.98) |
| vTTS    | 3.58 ± 0.13 | 3.12 ± 0.16 (−0.46) | 2.95 ± 0.21 (−0.63) |

Table 5. Results on CER. Δ denotes CER increase from “in-vocab.” Our method can synthesize more intelligible speech than TTS.

|         | in-vocab | rare (Δ) | OOV (Δ) |
|---------|----------|----------|---------|
| TTS     | 0.120    | 0.194 (+0.074) | 0.255 (+0.135) |
| vTTS    | 0.080    | 0.114 (+0.034) | 0.163 (+0.083) |

7https://github.com/Koruri/Koruri/releases/tag/20210720
8https://faraway.work/font.html
both TTS and vTTS, the scores decreased from “in-vocab” to “rare” and “OOV,” but the changes in vTTS were smaller than those in TTS. The dictation results for intelligibility are shown in Table 5. The overall tendency was the same as the MOS test. There was a significant difference between TTS and vTTS in the “in-vocab,” “rare,” and “OOV” sets ($p < 0.001$). Also, the CER increase for vTTS was also smaller than that for TTS.

From these results, we can say that our vTTS method can capture character compositionality for speech synthesis and synthesize intelligible speech even from OOV and rare characters.

### 5. CONCLUSIONS

In this paper, we proposed vTTS, a task to generate speech waveforms from visual text, and a vTTS method that can convert visual text to speech features in an end-to-end manner. Experimental results showed that 1) the basic performance of the vTTS method is comparable to or better than that of the TTS method, 2) it can reflect several attributes in visual text in speech, and 3) it can generate more natural speech from OOV and rare characters. In future work, we plan to apply our method to a variety of media such as manga or posters. We hope that this research will serve as a first benchmark in the development of speech synthesis from visual text.

### 6. REFERENCES

[1] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, “WaveNet: A generative model for raw audio,” arXiv preprint arXiv:1609.03499, Sep. 2016.

[2] Yuxuan Wang, R. J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhounan Chen, Samy Bengio, Quoc V. Le, Yannis Agiomyrgiannakis, Robert A. J. Clark, and Rif A. Saurous, “Tacotron: Towards end-to-end speech synthesis,” in Proc. INTERSPEECH, Aug. 2017, pp. 4006–4010.

[3] Jonathan Shen, Ruoming Pang, Ron Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, Rif Saurous, Yannis Agiomyrvgiannakis, and Yonghui Wu, “Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions,” in Proc. ICASSP, Apr. 2018, pp. 4779–4783.

[4] Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu, “FastSpeech: Fast, robust and controllable text to speech,” in Proc. NeurIPS, May. 2019, pp. 3165–3174.

[5] Hendrik Strobelt, Daniela Oelke, Bum Chul Kwon, Tobias Schreck, and Hanspeter Pfister, “Guidelines for effective usage of text highlighting techniques,” IEEE Transactions on Visualization and Computer Graphics, pp. 489–498, May. 2016.

[6] Saemi Choi, Toshihiko Yamasaki, and Kiyoharu Aizawa, “Face2speech: Towards multi-speaker text-to-speech synthesis using an embedding vector predicted from a face image,” in Proc. INTERSPEECH, Oct. 2020.

[7] Joshua Caldwell, “Japanese typeface personalities: Are typeface personalities consistent across culture?,” in Proc. IEEE International Professional Communication 2013 Conference, Jul. 2013, pp. 1–8.

[8] Kosuke Nonaka, Junki Saito, and Satoshi Nakamura, “Music video clip impression emphasis method by font fusion synchronized with music,” in Proc. Entertainment Computing and Serious Games, Nov. 2019, pp. 146–157.

[9] Rui Liu, Berrak Sisman, and Haizhou Li, “Reinforcement learning for emotional text-to-speech synthesis with improved emotion discriminability,” in Proc. INTERSPEECH, Aug. 2021, pp. 4648–4652.

[10] Chenye Cui, Yi Ren, Jinglin Liu, Feiyang Chen, Rongjie Huang, Ming Lei, and Zhou Zhao, “EMOVIE: A mandarin emotion speech dataset with a simple emotional text-to-speech model,” in Proc. INTERSPEECH, Aug. 2021, pp. 2766–2770.

[11] Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu, “FastSpeech 2: Fast and high-quality end-to-end text to speech,” in Proc. ICLR, Sep. 2021.

[12] Yuxuan Wang, Daisy Stanton, Yu Zhang, R. J. Skerry-Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Fei Ren, Ye Jia, and Rif A. Saurous, “Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis,” in Proc. ICML, Mar. 2018, p. 5167–5176.

[13] Shunsuke Goto, Kotaro Onishi, Yuki Saito, Kentaro Tachibana, and Koichiro Mori, “Face2speech: Towards multi-speaker text-to-speech synthesis using an embedding vector predicted from a face image,” in Proc. INTERSPEECH, Oct. 2020.

[14] Tam’s as Gabor Csapo, L. Viktor Toth, Gabo Gosztolya, and Alexandra Marko, “Speech synthesis from text and ultrasound tongue image-based articulatory input,” arXiv preprint arXiv:2107.02003, Jul. 2021.
[15] Yukiya Hono, Kazuna Tsuboi, Kei Sawada, Kei Hashimoto, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda, “Hierarchical Multi-Grained Generative Model for Expressive Speech Synthesis,” in Proc. INTERSPEECH, Oct. 2020, pp. 3441–3445.

[16] Guangzhi Sun, Yu Zhang, Ron J. Weiss, Yuan Cao, Heiga Zen, and Yonghui Wu, “Fully-hierarchical fine-grained prosody modeling for interpretable speech synthesis,” in Proc. ICASSP, May. 2020, pp. 6264–6268.

[17] Masashi Aso, Shinnosuke Takamichi, Norihiro Takamune, and Hiroshi Saruwatari, “Acoustic model-based subword tokenization and prosodic-context extraction without language knowledge for text-to-speech synthesis,” Speech Communication, vol. 125, pp. 53–60, Dec. 2020.

[18] Bo Li, Yu Zhang, Tara Sainath, Yonghui Wu, and William Chan, “Bytes are all you need: End-to-end multilingual speech recognition and synthesis with bytes,” in Proc. ICASSP, May. 2019, pp. 5621–5625.

[19] Richard Zens, Franz Josef Och, and Hermann Ney, “Phrase-based statistical machine translation,” in Proc. Annual Conference on Artificial Intelligence, Sep. 2002, pp. 18–32.

[20] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomáš Mikolov, “Enriching word vectors with subword information,” arXiv preprint arXiv:1607.04606, Jul. 2016.

[21] Rico Sennrich, Barry Haddow, and Alexandra Birch, “Neural machine translation of rare words with subword units,” in Proc. Association for Computational Linguistics, Aug. 2016, pp. 1715–1725.

[22] Hicham El Boukkouri, Olivier Ferret, Thomas Lavergne, Hiroshi Noji, Pierre Zweigenbaum, and Jun’ichi Tsujii, “CharacterBERT: Reconciling ELMo and BERT for word-level open-vocabulary representations from characters,” in Proc. COLING, Dec. 2020.

[23] Longtu Zhang and Mamoru Komachi, “Neural machine translation of logographic language using sub-character level information,” in Proc. the Third Conference on Machine Translation, Oct. 2018, pp. 17–25.

[24] Frederick Liu, Han Lu, Chieh Lo, and Graham Neubig, “Learning character-level compositionality with visual features,” in Proc. ACL, Jul. 2017, pp. 2059–2068.

[25] Falcon Dai and Zheng Cai, “Glyph-aware embedding of Chinese characters,” in Proc. EMNLP, Sep 2017, pp. 64–69.

[26] Baohua Sun, Lin Yang, Patrick Dong, Wenhan Zhang, Jason Dong, and Charles Young, “Super characters: A conversion from sentiment classification to image classification,” in Proc. EMNLP, Oct. 2018, p. 309–315.

[27] Elizabeth Salesky, David Etter, and Matt Post, “Robust open-vocabulary translation from visual text representations,” in Proc. EMNLP, Nov. 2021, pp. 7235–7252.

[28] Sergey Ioffe and Christian Szegedy, “Batch Normalization: Accelerating deep network training by reducing internal covariate shift,” in Proc. ICML, Jul. 2015, pp. 448–456.

[29] Vinod Nair and Geoffrey E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in Proc. ICML, Jun. 2010, pp. 807–814.

[30] Ryoosuke Sonobe, Shinnosuke Takamichi, and Hiroshi Saruwatari, “JSUT corpus: free large-scale Japanese speech corpus for end-to-end speech synthesis,” arXiv preprint arXiv:1711.00354, Oct. 2017.

[31] Kyubyong Park, “KSS dataset: Korean single speaker speech dataset,” https://kaggle.com/bryandpark/korean-single-speaker-speech-dataset, 2018.

[32] Keith Ito and Linda Johnson, “The LJSpeech dataset,” https://keithito.com/LJ-Speech-Dataset/, 2017.

[33] Shinnosuke Takamichi, Yuki Saito, Tomohiko Nakamura, Tomoki Kooriyama, and Hiroshi Saruwatari, “manga2voice: speech analysis towards audio synthesis from comic image,” in Proc. 2020 Spring meeting of the Acoustical Society of Japan, Mar. 2020.

[34] “HMM-based speech synthesis system (HTS),” http://hts.sp.nitech.ac.jp/.

[35] Chung-Ming Chien, Jheng-Hao Lin, Chien-yu Huang, Po-chun Hsu, and Hung-yi Lee, “Investigating on incorporating pretrained and learnable speaker representations for multi-speaker multi-style text-to-speech,” in Proc. ICASSP, Jun. 2021, pp. 8588–8592.

[36] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae, “HiFi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis,” in Proc. NeurIPS, Oct. 2020, pp. 17022–17033.