Citation: Haiquan Fang and Dian Yu. The real-time and high-resolution iInteractive digital human. *Journal of Harbin Institute of Technology (New Series)*. DOI: 10.11916/j.issn.1005-9113.24056.

The Real-time and High-resolution Interactive Digital Human

Haiquan Fang and Dian Yu*

(*School of Public Administration, Zhejiang University of Technology, Hangzhou 310023, China*)

Abstract: Synthesizing real-time, high-resolution, and lip-sync digital human is a challenging task. Although the Wav2Lip model represents a remarkable advancement in real-time lip-sync, its clarity is still limited. To address this, we enhanced the Wav2Lip model in this study and trained it on a high-resolution video dataset produced in our laboratory. Experimental results indicate that the improved Wav2Lip model produces digital humans with greater clarity than the original model, while maintaining its real-time performance and accurate lip-sync. We implemented the improved Wav2Lip model in a government interface application, generating a government digital human. Testing revealed that this government digital human can interact seamlessly with users in real-time, delivering clear visuals and synthesized speech that closely resembles a human voice.

Keywords: digital human; lip-sync; high-resolution; video generation; talking face generation

CLC number: TP39 **Document code**: A **Article ID**: 1005-9113(2024)00-0000-11

0 Introduction

A "digital human" is a digital technological designed to simulate real-world human characteristics. An "interactive digital human" is one that can engage and converse with users in real time. The interactive digital human systems comprise three main components: the digital human itself, a question-answering (QA) system, and a text-to-speech (TTS) system. Optionally, a speech recognition (SR) system can be included or replaced with text input. Digital humans have been extensively applied across sectors such as government, e-commerce, healthcare, cultural tourism, and education. A "government interactive digital human" in this context is designed for public administration, where it functions as an intelligent customer service tool for governance-related inquiries. This technology offers swift, convenient access to public services, and has helped to facilitate the digital transformation of government affairs.

The core technology supporting digital human is speech-driven talking face synthesis, the most challenging aspect of which is lip-sync accuracy. While recent advancements have greatly enhanced this technology, certain challenges persist. The Wav2Lip[1] model, a leading solution in terms of real-time performance and lip-sync accuracy, still lacks clarity as it processes facial images with a resolution of only 96×96 pixels; further, the videos used to train it are low-resolution and not sufficiently sharp. We improved the Wav2Lip model in this study to improve the clarity of its digital-human visuals. The improved model can process facial images with a resolution of 384×384 pixels and was trained on high-resolution video data produced in our laboratory.

The key contributions are as follows:

1) We improved the Wav2Lip model by adding a six-layer convolutional neural network (CNN) structure to the lip-sync discriminator's face encoder. We also introduced two extra blocks to both the face encoder block and face decoder block of the generator. We recorded a high-resolution video dataset to train the model, enabling it to subsequently generate high-resolution digital human in real time. The face resolution of generated faces increased from 96×96 pixels to 384×384 pixels while retaining the original model's lip-syncing accuracy and real-time

Received 2024-09-06.

Sponsored by the Collaborative Education Projects Between Industry and Academia by the Ministry of Education (Grant No. 230801065261444), and the Humanities and Social Sciences Pre Research Fund Project of Zhejiang University of Technology (Grant No.SKY-ZX-20220207)

* Corresponding author: Haiquan Fang, Ph.D. E-mail: fanghaiquan@zjut.edu.cn

performance.

2) We assembled and organized a government knowledge base (KB), a government QA system, and a fine-tuned TTS system. We integrated the digital human, government QA system, and TTS to form a comprehensive and interactive digital human platform.

1 Related Work

1.1 Digital Human

Existing digital human technologies can be either two-dimensional (2D) or three-dimensional (3D). This research primarily focuses on 2D digital human, which rely on speech-driven talking-face synthesis as their core technology. Current research on this topic falls into two main categories: speaker-specific models and speaker-independent models. Speaker-specific models require identity-specific training and extensive data on a particular speaker. In this study, we focused on speaker-independent models.

There are many algorithmic models for digital humans, including Wav2Lip[1], Speech2Vid[2], Kim et al.'s model[3], ATVG[4], FOMM[5], MakeItTalk[6], FACIAL[7], AnyoneNet[8], Audio2Head[9], EVP[10], PC-AVS[11], AD-NeRF[12], StyleTalker[13], DFA-NeRF[14], DFRF[15], and DiffTalk[16]. These can be divided into 5 categories according to their generation methods. Methods based on audio features include Speech2Vid, Wav2lip, PC-AVS, StyleTalker. Among them, Wav2Lip, PC-AVS, and StyleTalker adopt generative adversarial networks (GAN)[17]. Methods based on 2D facial representation include ATVG MakeItTalk and AnyoneNet. Methods based on 3D morphable models (3DMM)[18] include Kim et al.'s, FACIAL, EVP, AD-NeRF, DFA-NeRF, and DFRF. Among them, AD NeRF, DFA NeRF, and DFRF use neural radiation fields (Nerfs)[19]. Methods based on dense motion fields[20] include FOMM and Audio2Head, while those based on diffusion models[21] include DiffTalk.

Wav2Lip[1] is widely regarded as the state-of-the-art for its synthesis speed and lip-sync accuracy. However, its output resolution is low and the videos it generates are fairly blurry. Gupta[22] created improved high-resolution talking-face videos by training a lip-sync generator in a compact vector quantized space. They also used the GPEN face-enhancement method for enhanced resolution, which is effective but overly time-consuming and not conducive to real-time synthesis. Muaz[23] applied shift-invariant learning to generate photo-realistic, high-resolution videos with accurate lip-sync; this approach is also computationally costly. Ideally, a digital human model needs to balance both high clarity and processing speed.

1.2 Question Answering System

Many types of QA systems have been established to date; our focus here is on systems supported by question-answer pairs. These systems operate by calculating the similarity between a user-submitted question and the questions stored in a knowledge base, returning the answer associated with the most similar question as a candidate answer. To complete this calculation, both the user's question and the stored questions must first be converted into vector forms. There has been significant research on text vectorization, which can be used to facilitate this conversion.

Early text vectorization methods include the word2vec model[24], a type of neural network model. After 2018, the field shifted towards pre-training models, marked by the advent of large language models such as GPT[25-27], BERT[28], and ERNIE[29]. These models fundamentally rely on the Transformer[30] deep learning structure.

1.3 TTS

TTS synthesis is designed to produce human-like speech from text. Typically, TTS models first generate a mel-spectrogram from input text[31-38] which is then converted into a speech waveform using a separately pre-trained vocoder[39-42]. Alternatively, some models directly generate the waveform from text in an end-to-end manner[43-46].

Traditional TTS systems were trained on relatively small datasets. In contrast, today's large-scale TTS systems[47-49] are trained on tens of thousands of hours of speech data. These advanced systems can be fine-tuned with as little as 1 min or even 5 s of training data. The resulting speech closely matches the original voice and sounds more natural.

2 Methods

2.1 Digital Human for Real-time and High-resolution

The Wav2Lip[1] model includes two parts: a lip-sync discriminator and a generator. To accommodate the increased input image resolution in our model,

additional convolutional layers were necessary.

2.1.1 Lip-sync discriminator and its training principle

The lip-sync discriminator consists of a face encoder and an audio encoder. Both predominantly utilize CNN architectures. The face encoder of the original Wav2Lip model accepts an input image size of 96x96 pixels, whereas our improved version uses an input size of 384×384 pixels. This four-fold increase in both length and width necessitated the addition of six layers to the convolutional structure of the face encoder beyond that of the original Wav2Lip model. Details regarding the face encoder model are described in Appendix 1. The design of the original audio encoder was kept unchanged.

The original dataset consists of recorded video files that are segmented into shorter clips during preprocessing, each containing only one sentence. Each short video is then processed to frame the video, detect faces, and extract the audio, yielding multiple facial images and an accompanying audio file. This preprocessing transforms the original video data into a dataset suitable for training.

The training process begins by randomly selecting a starting frame number and extracting five consecutive images from that point. The mel-spectogram matrix of the audio corresponding to these five images is calculated. These images, along with their associated mel-spectogram matrix, form a positive sample. Conversely, a negative sample is created when the audio does not correspond to the images.

In this system, a positive sample is labeled with a ground truth of "1" and a negative sample with a ground truth of "0". The face encoder processes each image to produce a vector $\boldsymbol{v}$, while the audio encoder processes the mel-spectogram matrix to also produce a vector $\boldsymbol{s}$. Both vectors have shapes of 1×512. The output value of the lip-sync discriminator is the cosine similarity between these two vectors, as shown in Eq.(1).

$$p = \frac{\boldsymbol{v} \cdot \boldsymbol{s}}{\|\boldsymbol{v}\|_2 \cdot \|\boldsymbol{s}\|_2} \tag{1}$$

During model training, multiple samples are collected into a batch (e.g., a batch size of 10). The cosine similarities p of these 10 samples are computed to create an array, as are their corresponding ground truths y. The loss between these two arrays is calculated using the cross-entropy loss function, as shown in Eq.(2). The parameters of the face encoder and audio encoder models are optimized based on this loss.

$$\text{Loss} = -\frac{1}{n}\sum_{i=1}^{n} y\log(p) \tag{2}$$

2.1.2 Generator and its training principle

The generator includes three key components: a face encoder block model, a face decoder block model, and an audio encoder. All primarily utilize CNN structures. Compared to the traditional Wav2Lip model, we increased the depth of both the face encoder block model and face decoder block model from seven blocks to nine blocks for enhanced performance. Each block includes one deconvolution layer and two convolutional layers. Appendices 2 and 3 provide details of the face encoder block model and face decoder block model, respectively. The design of the audio encoder was, again, kept consistent with the original model.

To train the generator, begin by selecting a random number i to obtain five consecutive images starting from this index. Replace the pixels in the lower half of these five images with zero values. Then, randomly select another five images. Combine these ten images, denoted as x. Calculate the mel-spectogram matrix corresponding to the five consecutive images starting from the i-th image, denoted as mel. Additionally, calculate separate mel-spectogram matrices for the five consecutive images starting from the $(i-1)$-th, i-th, $(i+1)$-th, $(i+2)$-th, and $(i+3)$-th image, denoted as indiv_mel. The ground truth, denoted as gt, is the sequence of five consecutive images starting from the i-th image.

Model processing includes four steps:

1) Process the audio *indiv_mel* using the audio encoder model;

2) Process image x using the face encoder block model;

3) Use the face decoder block model to process the outputs of the first two steps;

4) Use the output block model to process the results of the third step to obtain the generated image, denoted as L_g.

The loss function includes two parts: 1) Calculating the loss between the generated image L_g and the ground truth L_G using the L_1 norm loss function, as shown in Eq.(3), and 2) employing the pre-trained expert lip-sync discriminator to calculate the loss value. Input the generated image L_g and mel-spectogram matrix into the discriminator to compute

the loss using the cross-entropy loss function. Then, combine these two loss values to calculate the total loss and optimize the generator model parameters accordingly.

$$L_1 = \frac{1}{n}\sum_{i=1}^{n} \| L_g - L_G \|_1 \quad (3)$$

2.1.3 High-resolution video data

High-resolution video data is required for model training to create high-resolution digital humans. We recorded a dataset of 4K resolution videos totaling 300 min in duration and featuring ten individuals. Each individual participant contributed 30 min of spoken content in Chinese, with only one person speaking per video.

We used the S3FD[50] algorithm for face detection, obtaining cropped face images. Typically, such cropped images are rectangular and be directly converted to a square format[1]. However, this can cause distortion. We addressed this problem by using the bottom edge of the cropped face image as a reference, selecting the square down, and then resize to 384×384 pixels.

2.2 Government Question Answering System

2.2.1 Construction of government knowledge base

We compiled relevant governmental data to establish a government question answering system, collecting 400 question-answer pairs from the Zhejiang Provincial Government Service Network. This network covers a wide range of information including residence registry, ID card services, immigration, social insurances and housing funds, vehicle-related services, corporate groups, professional certifications, and departmental responsibilities. Relevant question-and-answer pairs were organized into a knowledge base, structured to include both questions and corresponding answers. Using a Python program, we constructed a dictionary wherein each question serves as a key and its respective answer as the value.

2.2.2 Construction of government question answering system

We built a government question answering system based on the knowledge base. This primarily included computing the similarity between a user-submitted question and the questions contained within the knowledge base. The system then selects the answer corresponding to the most similar question as a candidate answer. To complete this process, both the user's question and the knowledge base questions must be converted into vector format. We employed the BERT model, a robust text vectorization tool, to transform the text into vectors.

2.3 TTS

Existing TTS technology is relatively mature. We employed the open-source speech synthesis system GPT-SoVITS to fine-tune our TTS model. Fine-tuning requires audio data from a target speaker. By recording just one minute of audio from a speaker, we were able to fine-tune our model so that the synthesized voice closely approximates the original voice.

3 Experimentation and Practical Application

3.1 Technical Route

The objective of this study was to first develop a real-time and high-resolution digital human, followed by the creation of a government interactive digital human.

The core of the digital human is based on the improved Wav2Lip model (as shown in Fig. 1). The technical route to establishing it first involved capturing and preparing video data for subsequent processing, followed by developing the lip-sync discriminator model and the generator model as components of the improved Wav2Lip. This improved model was then trained using preprocessed data, focusing on both the lip-sync discriminator and generator. Once the generator model was fully trained, inputting audio into the system enabled the generation of digital human videos in real-time.

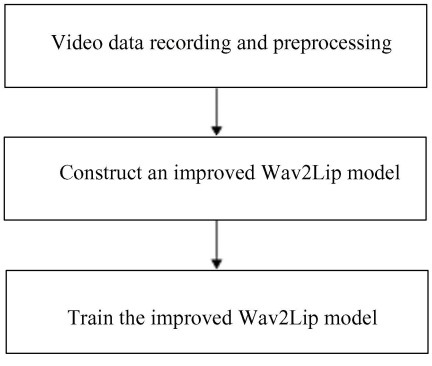


Fig. 1 The technical route of digital human

The technical route for government digital human interaction is depicted in Fig. 2. First, the user inputs a question into the government question answering system. If the system finds a matching answer, it outputs the corresponding text. The answer text is then

fed into the TTS system, which converts it into audio. Finally, the audio is input into the improved Wav2Lip model, which generates a digital human video providing a visual and auditory response to the user.

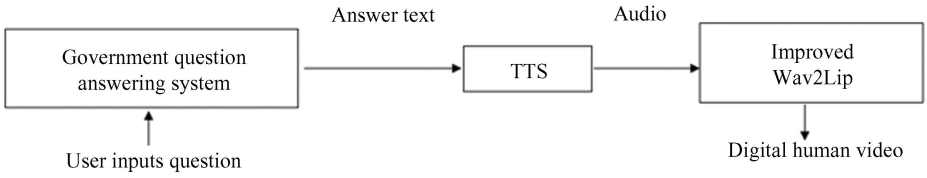


Fig. 2 The technical route of digital human in government interaction

3.2 Program Implementation and Deployment

The digital human, government question answering system, and TTS components were all developed using the PyTorch deep learning framework and implemented in the Python program. The system was deployed with the Docker method to ensure a scalable and isolated environment. The three components were integrated into a cohesive interactive platform using the Vue framework. The hardware configuration supporting the system comprises a NVIDIA RTX 3060 Ti graphics card with 12 GB memory and i9-10900k CPU, on a computer operating with Linux.

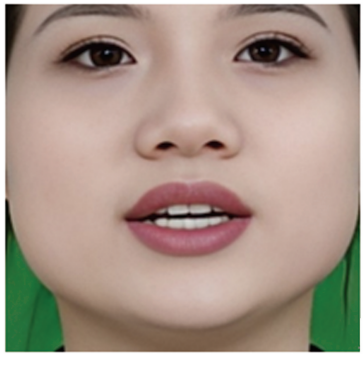
3.3 Real-time and High-resolution Digital Human Generation

The improved Wav2Lip model successfully generates high-quality digital humans after training, exhibiting two main advantages: 1) The generated face image is clear, as shown in Fig. 3; 2) the model swiftly produces video content (e.g., a 10 s video returned in only 5 s).

(a)A randomly selected negative sample image

(b)The image which needs to be generated

(c)Image generated by the improved Wav2Lip model, illustrating our model's ability to generate clear images



(d)A ground truth image of the positive sample

Fig. 3 Images saved during training phase

3.4 Quantitative Evaluations

To evaluate the quality of lip-sync, we use "Lip Sync Error-Confidence" (LSE-C) and "Lip Sync Error-Distance" (LSE-D) metrics introduced in

Wav2Lip[1]. We also use "Fréchet Inception Distance" (FID) to measure the quality of the generated faces[51]. The mean LSE-D, LSE-C and FID scores are shown in Table 1.

Table 1 Quantitative comparison of different methods on our new datasets

Method	LSE-C ↑	LSE-D ↓	FID ↓
Wav2Lip-96×96	5.65	8.23	4.25
Wav2Lip-384×384 (ours)	6.06	7.68	2.89
Real videos	7.17	6.76	—

As shown in Table 1, our method outperforms previous approaches. We see that the lip-sync accuracy of the videos generated using our method is almost as good as real synced videos, and our method produces lip-sync videos at very high-resolutions.

3.5 Practical Application

3.5.1 Text-generated digital human

Text can be transformed into digital-human video by combining the TTS and digital human model, as depicted in Fig. 4. We employed the open-source speech synthesis system GPT-SoVITS to fine-tune our TTS model, resulting in a high-quality synthesized voice closely approximating the original human voice. The speech synthesis also operates at high speed, generating 1 s of audio in only 0.5 s, allowing for real-time synthesis.

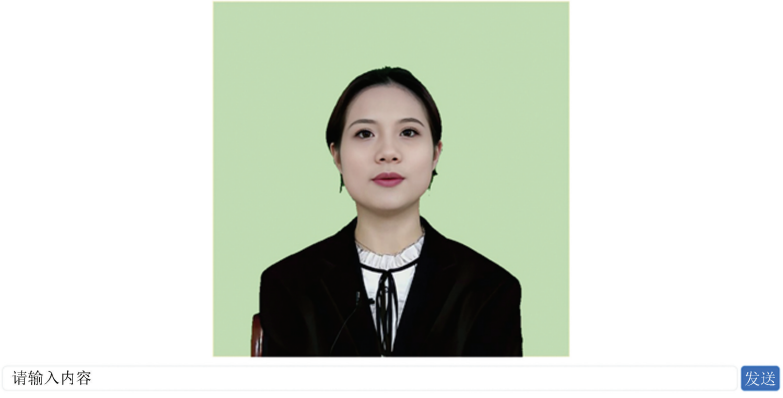


Fig. 4 Text-generated digital human

3.5.2 Government interaction digital human

In this study, we integrated a government question answering system, TTS, and digital human to create a government interactive digital human system. When a user submits a question, if there is an answer in the knowledge base, the digital human can quickly vocalize a response (as shown in Fig. 5).

Fig. 5 Government interaction digital human

The government digital human we are developing is currently in the experimental phase and has not been formally implemented in any practical application of governance operations. Government

departments have dedicated data storage and management departments to ensure security. If our digital human was adopted by these departments, the associated QA and KB would need to be carefully screened to include only publicly accessible data, further safeguarding data security. Furthermore, the number and distribution of users could be counted and user feedback could be recorded to further optimize the digital human model. Optimized digital humans can provide users with a better experience, increase user engagement, and ultimately contribute to broader societal benefits.

4 Conclusions

In this study, we improved the Wav2Lip model to enable it to process resolutions up to 384 × 384 pixels, compared to the original 96×96 pixels. We also created a set of high-resolution video data for training, allowing the model to generate high-resolution, accurately lip-synced digital human in real time, achieving clear results without the need for facial enhancement algorithms. We constructed the digital human, government QA and TTS, then integrated these three components to create a government digital human capable of real-time interaction with clear, high-quality output. This digital human system is well-suited for deployment in public service sectors and may significantly advance the digital transformation of government operations.

The government QA system utilized in this work operates on a fixed set of question-and-answer pairs, limiting its ability to handle a broader array of inquiries. Moving forward, we plan to explore the development of a large government model to expand its capabilities. The system could also respond to a wider variety of questions after integrating the digital human with large models like Zhipu Qingyan's ChatGLM4. Additionally, while the proposed government interaction digital human currently supports only Chinese, it may be extended to other languages after gathering video data in those respective languages.

Note: The facial images used in this study were provided by an author of this article and are presented here with her consent.

References

[1] Prajwal K R, Mukhopadhyay R, Namboodiri V P, et al. A lip sync expert is all you need for speech to lip generation in the wild. Proceedings of the 28th ACM International Conference on Multimedia. New York: ACM Press, 2020: 484-492.

[2] Chung J S, Jamaludin A, Zisserman A. You said that? [2017-05-08]. https://arxiv.org/abs/1705.02966.

[3] Kim H, Garrido P, Tewari A, et al. Deep video portraits. ACM Transactions on Graphics, 2018, 37(4): Article No.163. DOI: 10.1145/3197517.3201283.

[4] Chen L, Maddox R K, Duan Z Y, et al. Hierarchical cross-modal talking face generation with dynamic pixel-wise loss. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2019: 7824-7833. DOI: 10.1109/CVPR.2019.00802.

[5] Siarohin A, Lathuilière S, Tulyakov S, et al. First order motion model for image animation. Proceedings of the 33rd International Conference on Neural Information Processing Systems. New York: ACM Press, 2019: 7137-7147.

[6] Zhou Y, Han X T, Shechtman E, et al. MakeltTalk: speaker-aware talking-head animation. ACM Transactions on Graphics, 2020, 39(6): Article No.221. DOI: 10.1145/3414685.3417774.

[7] Zhang C X, Zhao Y F, Huang Y F, et al. FACIAL: synthesizing dynamic talking face with implicit attribute learning. Proceedings of the IEEE/CVF International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2021: 3847-3856. DOI: 10.1109/ICCV48922.2021.00384.

[8] Wang X S, Xie Q C, Zhu J H, et al. AnyoneNet: synchronized speech and talking head generation for arbitrary persons. IEEE Transactions on Multimedia, 2022, 25: 6717-6728. DOI: 10.1109/TMM.2022.3214100.

[9] Wang S Z, Li L C, Ding Y, et al. Audio2Head: audio-driven one-shot talking-head generation with natural head motion. Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence. Montreal: IJCAI, 2021: 1098-1105. DOI: 10.24963/ijcai.2021/152.

[10] Ji X Y, Zhou H, Wang K Y, et al. Audio-driven emotional video portraits. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2021: 14075-14084.

[11] Zhou H, Sun Y S, Wu W N, et al. Pose-controllable talking face generation by implicitly modularized audio-visual representation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2021: 4174-4184.

[12] Guo Y D, Chen K Y, Liang S, et al. AD-NeRF: audio driven neural radiance fields for talking head synthesis. Proceedings of the IEEE/CVF International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2021: 5764-5774.

[13] Min D C, Song M, Ko E, et al. StyleTalker: one-shot style-based audio-driven talking head video generation. [2022-08-23]. https://arxiv.org/abs/2208.10922.

[14] Yao S Y, Zhong R Z, Yan Y C, et al. DFA-NeRF: personalized talking head generation via disentangled face attributes neural rendering. [2022-01-03]. https://arxiv.org/abs/2201.00791.

[15] Shen S, Li W H, Zhu Z, et al. Learning dynamic facial radiance fields for few-shot talking head synthesis. Proceedings of the 17th European Conference on Computer Vision. Heidel-berg: Springer, 2022: 666-682.

[16] Shen S, Zhao W L, Meng Z B, et al. DiffTalk: crafting diffusion models for generalized audio-driven portraits animation. [2023-01-10]. https://arxiv.org/abs/2301.03786.

[17] Goodfellow I J, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets. Proceedings of the 27th International Conference on Neural Information Processing Systems. New York: ACM Press, 2014, 2: 2672-2680.

[18] Blanz V, Vetter T. A morphable model for the synthesis of 3D faces. Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques. New York: ACM Press, 1999: 187-194. DOI: 10.1145/311535.311556.

[19] Mildenhall B, Srinivasan P P, Tancik M, et al. NeRF: repre-senting scenes as neural radiance fields for view synthesis. Communications of the ACM, 2022, 65(1): 99-106. DOI: 10.1145/3503250.

[20] Siarohin A, Lathuilière S, Tulyakov S, et al. Animating arbitrary objects via deep motion transfer. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2019: 2372-2381. DOI: 10.1109/CVPR.2019.00248.

[21] Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models. Proceedings of the Neural Information Processing Systems. New York: ACM Press, 2020: 6840-6851.

[22] Gupta A, Mukhopadhyay R, Balachandra S, et al. Towards generating ultra-high resolution talking-face videos with lip synchronization. Proceedings of Winter Conference on Applications of Computer Vision. Piscataway: IEEE, 2023: 5198-5207. DOI: 10.1109/WACV56688.2023.00518.

[23] Muaz U, Jang W, Tripathi R, et al. SIDGAN: High-resolution dubbed video generation via shift-invariant learning. Proceedings of the IEEE/CVF International Conference on Computer Vision. Piscataway: IEEE, 2023: 7799-7808. DOI: 10.1109/ICCV51070.2023.00720.

[24] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality. NIPS'13: Proceedings of the 26th International Conference on Neural Information Processing Systems. New York: ACM Press, 2023, 2: 3111-3119.

[25] Radford A, Narasimhan K, Salimans T, et al. Improving language understanding by generative pre-training. arXiv: 1301.3781, 2018.

[26] Radford A, Wu J, Child R, et al. Language models are unsupervised multitask learners. OpenAI blog, 2019, 1(8): 8-9.

[27] Tom B Brown, Benjamin Mann, Nick Ryder, et al. Language models are few-shot learners. NIPS'20: Proceedings of the 34th International Conference on Neural Information Processing Systems. New York: ACM Press, 2020: 1877-1901.

[28] Devlin J, Chang M-W, Lee K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis: Association for Computational Linguistics, 2018, 1: 4171-4186.

[29] Sun Y, Wang S, Li Y, et al. ERNIE: Enhanced representation through knowledge integration. arXiv: 1904.09223, 2019.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. Attention is all you need. arXiv: 1706.03762, 2017. https://doi.org/10.48550/arXiv.1904.09223.

[31] Wang Y, Skerry-Ryan R, Stanton D, et al. Tacotron: Towards end-to-end speech synthesis. arXiv: 1703.10135, 2017.

[32] Ö Arık S, Chrzanowski M, Coates A, et al. Deep voice: Real-time neural text-to-speech. Proceedings of Machine Learning Research: International Conference on Machine Learning. PMLR, 2017, 139: 195-204.

[33] Li N, Liu S, Liu Y, et al. Neural speech synthesis with transformer network. AAAI'19/IAAI'19/EAAI'19: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence. New York: ACN Press, 2019, 33: 6706-6713. DOI: 10.1609/aaai.v33i01.33016706.

[34] Ren Y, Ruan Y, Tan X, et al. FastSpeech: Fast, robust and controllable text to speech. Conference on Neural Information Processing Systems. Vancouver: NeruIPS, 2019.

[35] Kim J, Kim S, Kong J, et al. Glow-TTS: A generative flow for text-to-speech via monotonic alignment search. Advances in Neural Information Processing Systems. Vancouver: NeruIPS, 2020, 33: 8067-8077.

[36] Ren Y, Liu J, Zhao Z. Portaspeech: Portable and high-quality generative text-to-speech. Advances in Neural Information Processing Systems. Vancouver: NeruIPS, 2021.

[37] Liu J, Li C, Ren Y, et al. Diffsinger: Singing voice synthesis via shallow diffusion mechanism. Proceedings of the AAAI Conference on Artificial Intelligence, 2022,

36: 11020-11028.DOI:10.1609/aaai.v36i10.21350.
[38] Huang R, Zhao Z, Liu H, et al. Prodiff: Progressive fast diffusion model for high-quality text-to-speech. Proceedings of the 30th ACM International Conference on Multimedia. New York: ACM Press, 2022: 2595-2605. DOI:10.48550/arXiv.2207.06389.
[39] van den Oord A, Dieleman S, Zen H, et al. Wavenet: A generative model for raw audio. arXiv:1609.03499, 2016.
[40] Kong J, Kim J, Bae J. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. 34th Conference on Neural Information Processing Systems. Vancouver: NeruIPS, 2020, 33: 17022-17033.
[41] Yamamoto R, Song E, Kim J-M. Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. IEEE International Conference on Acoustics, Speech and Signal Processing. Piscataway: IEEE, 2020: 6199-6203.
[42] Huang R, Lam M W Y, Wang J, et al. FastDiff: A fast conditional diffusion model for high-quality speech synthesis. arXiv: 2204.09934, 2022. https://doi.org/10.48550/arXiv.2204.09934.
[43] Ren Y, Hu C, Tan X, et al. Fastspeech 2: Fast and high-quality end-to-end text to speech. International Conference on Learning Representations. Appleton: ICLR, 2020.
[44] Donahue J, Dieleman S, Binkowski M, et al. End-to-end adversarial text-to-speech. In 9th International Conference on Learning Representations. Appleton: ICLR, 2021.
[45] Kim J, Kong J, Son J. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In International Conference on Machine Learning. PMLR, 2021: 5530-5540.
[46] Liu Y, Xue R, He L, et al. Delightfultts 2: End-to-end speech synthesis with adversarial vector-quantized auto-encoders. arXiv: 2207.04646, 2022.
[47] Wang C, Chen S, Wu Y, et al. Neural codec language models are zero-shot text to speech synthesizers. arXiv: 2301.02111, 2023. https://doi.org/10.48550/arXiv.2301.02111.
[48] Zhang Z, Zhou L, Wang C, et al. Speak foreign languages with your own voice: Cross-lingual neural codec language modeling. arXiv: 2303.03926, 2023. https://doi.org/10.48550/arXiv.2303.03926.
[49] Kharitonov E, Vincent D, Borsos Z, et al. Speak, read and prompt: High-fidelity text-to-speech with minimal supervision. Transactions of the Association for Computational Linguistics, 2023, 11: 1703-1718. DOI: 10.1162/tacl_a_00618.
[50] Zhang S, Zhu X, Lei Z, et al. S3fd: Single shot scale-invariant face detector. 2017 IEEE International Conference on Computer Vision. Piscataway: IEEE, 2017: 192-201.
[51] Heusel M, Ramsauer H, Unterthiner T, et al. GANs trained by a two time-scale update rule converge to a local nash equilibrium. NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems. New York: ACM Press, 2017: 6629-6640.

Appendix 1: The face encoder model

(1) Input layer: The input image consists of 5 facial images, each with 3 channels and 384 × 384 pixels, totaling 15 channels.

(2) Face encoder model:

(2.1) Convolutional layer: there are 16 channels, the kernel is 7×7, stride is 1, padding is 3.

(2.2) Convolutional layer: there are 32 channels, the kernel is 5×5, stride is (1,2), padding is 1.

(2.3) Convolutional layer: there are 32 channels, the kernel is 3×3, stride is 1, padding is 1.

(2.4) Convolutional layer: there are 32 channels, the kernel is 3×3, stride is 1, padding is 1.

(2.5) Convolutional layer: there are 64 channels, the kernel is 3×3, stride is 2, padding is 1.

(2.6) Convolutional layer: there are 64 channels, the kernel is 3×3, stride is 1, padding is 1.

(2.7) Convolutional layer: there are 64 channels, the kernel is 3×3, stride is 1, padding is 1.

(2.8) Convolutional layer: there are 128 channels, the kernel is 3×3, stride is 2, padding is 1.

(2.9) Convolutional layer: there are 128 channels, the kernel is 3×3, stride is 1, padding is 1.

(2.10) Convolutional layer: there are 128 channels, the kernel is 3×3, stride is 1, padding is 1.

(2.11) Convolutional layer: there are 256 channels, the kernel is 3×3, stride is 2, padding is 1.

(2.12) Convolutional layer: there are 256 channels, the kernel is 3×3, stride is 1, padding is 1.

(2.13) Convolutional layer: there are 256 channels, the kernel is 3×3, stride is 1, padding is 1.

(2.14) Convolutional layer: there are 256 channels, the kernel is 3×3, stride is 2, padding is 1.

(2.15) Convolutional layer: there are 256 channels, the kernel is 3×3, stride is 1, padding is 1.

(2.16) Convolutional layer: there are 256 channels, the kernel is 3×3, stride is 1, padding is 1.

(2.17) Convolutional layer: there are 512 channels, the kernel is 3×3, stride is 2, padding is 1.

(2.18) Convolutional layer: there are 512 channels, the kernel is 3×3, stride is 1, padding is 1.

(2.19) Convolutional layer: there are 512 channels, the kernel is 3×3, stride is 1, padding is 1.

(2.20) Convolutional layer: there are 512 channels, the kernel is 3×3, stride is 2, padding is 1.

(2.21) Convolutional layer: there are 512 channels, the kernel is 3×3, stride is 1, padding is 0.

(2.22) Convolutional layer: there are 512 channels, the kernel is 1×1, stride is 1, padding is 0.

Appendix 2: The face encoder block model

(1) Face encoder block 1:

(1.1) Convolutional layer: there are 8 channels, the kernel is 7×7, stride is 1, padding is 3.

(2) Face encoder block 2:

(2.1) Convolutional layer: there are 16 channels, the kernel is 3×3, stride is 2, padding is 1.

(2.2) Convolutional layer: there are 16 channels, the kernel is 3×3, stride is 1, padding is 1.

(2.3) Convolutional layer: there are 16 channels, the kernel is 3×3, stride is 1, padding is 1.

(3) Face encoder block 3:

(3.1) Convolutional layer: there are 32 channels, the kernel is 3×3, stride is 2, padding is 1.

(3.2) Convolutional layer: there are 32 channels, the kernel is 3×3, stride is 1, padding is 1.

(3.3) Convolutional layer: there are 32 channels, the kernel is 3×3, stride is 1, padding is 1.

(4) Face encoder block 4:

(4.1) Convolutional layer: there are 64 channels, the kernel is 3×3, stride is 2, padding is 1.

(4.2) Convolutional layer: there are 64 channels, the kernel is 3×3, stride is 1, padding is 1.

(4.3) Convolutional layer: there are 64 channels, the kernel is 3×3, stride is 1, padding is 1.

(5) Face encoder block 5:

(5.1) Convolutional layer: there are 128 channels, the kernel is 3×3, stride is 2, padding is 1.

(5.2) Convolutional layer: there are 128 channels, the kernel is 3×3, stride is 1, padding is 1.

(5.3) Convolutional layer: there are 128 channels, the kernel is 3×3, stride is 1, padding is 1.

(6) Face encoder block 6:

(6.1) Convolutional layer: there are 256 channels, the kernel is 3×3, stride is 2, padding is 1.

(6.2) Convolutional layer: there are 256 channels, the kernel is 3×3, stride is 1, padding is 1.

(6.3) Convolutional layer: there are 256 channels, the kernel is 3×3, stride is 1, padding is 1.

(7) Face encoder block 7:

(7.1) Convolutional layer: there are 512 channels, the kernel is 3×3, stride is 2, padding is 1.

(7.2) Convolutional layer: there are 512 channels, the kernel is 3×3, stride is 1, padding is 1.

(7.3) Convolutional layer: there are 512 channels, the kernel is 3×3, stride is 1, padding is 1.

(8) Face encoder block 8:

(8.1) Convolutional layer: there are 512 channels, the kernel is 3×3, stride is 2, padding is 1.

(8.2) Convolutional layer: there are 512 channels, the kernel is 3×3, stride is 1, padding is 1.

(9) Face encoder block 9:

(9.1) Convolutional layer: there are 512 channels, the kernel is 3×3, stride is 2, padding is 0.

(9.2) Convolutional layer: there are 512 channels, the kernel is 3×3, stride is 1, padding is 0.

Appendix 3: The face decoder block model

(1) Face decoder block 1:

(1.1) Convolutional layer: there are 512 channels, the kernel is 1×1, stride is 1, padding is 0.

(2) Face decoder block 2:

(2.1) Deconvolution layer: there are 512 channels, the kernel is 3×3, stride is 1, padding is 0.

(2.2) Convolutional layer: there are 512 channels, the kernel is 3×3, stride is 1, padding is 1.

(3) Face decoder block 3

(3.1) Deconvolution layer: there are 512 channels, the kernel is 3×3, stride is 2, padding is 1, output padding is 1.

(3.2) Convolutional layer: there are 512 channels, the kernel is 3×3, stride is 1, padding is 1.

(3.3) Convolutional layer: there are 512 channels, the kernel is 3×3, stride is 1, padding is 1.

(4) Face decoder block 4:

(4.1) Deconvolution layer: there are 256 channels, the kernel is 3×3, stride is 2, padding is 1, output padding is 1.

(4.2) Convolutional layer: there are 256 channels, the kernel is 3×3, stride is 1, padding is 1.

(4.3) Convolutional layer: there are 256 channels, the kernel is 3×3, stride is 1, padding is 1.

(5) Face decoder block 5:

(5.1) Deconvolution layer: there are 128 channels, the kernel is 3×3, stride is 2, padding is 1, output padding is 1.

(5.2) Convolutional layer: there are 128 channels, the kernel is 3×3, stride is 1, padding is 1.

(5.3) Convolutional layer: there are 128 channels, the kernel is 3×3, stride is 1, padding is 1.

(6) Face decoder block 6:

(6.1) Deconvolution layer: there are 64 channels, the kernel is 3×3, stride is 2, padding is 1, output padding is 1.

(6.2) Convolutional layer: there are 64 channels, the kernel is 3×3, stride is 1, padding is 1.

(6.3) Convolutional layer: there are 64 channels, the

kernel is 3×3, stride is 1, padding is 1.

(7) Face decoder block 7:

(7.1) Deconvolution layer: there are 32 channels, the kernel is 3×3, stride is 2, padding is 1, output padding is 1.

(7.2) Convolutional layer: there are 32 channels, the kernel is 3×3, stride is 1, padding is 1.

(7.3) Convolutional layer: there are 32 channels, the kernel is 3×3, stride is 1, padding is 1.

(8) Face decoder block 8:

(8.1) Deconvolution layer: there are 16 channels, the kernel is 3×3, stride is 2, padding is 1, output padding is 1.

(8.2) Convolutional layer: there are 16 channels, the kernel is 3×3, stride is 1, padding is 1.

(8.3) Convolutional layer: there are 16 channels, the kernel is 3×3, stride is 1, padding is 1.

(9) Face decoder block 9:

(9.1) Deconvolution layer: there are 8 channels, the kernel is 3×3, stride is 2, padding is 1, output padding is 1.

(9.2) Convolutional layer: there are 8 channels, the kernel is 3×3, stride is 1, padding is 1.

(9.3) Convolutional layer: there are 8 channels, the kernel is 3×3, stride is 1, padding is 1.

The number of parameters in the model was determined through trial-and-error. A larger number of channels increases the complexity of the model and decelerates its inference speed, making real-time synthesis increasingly challenging. Conversely, too few channels leads to an overly simplistic model that cannot synthesize a digital human effectively.