Course Project

Code of Honor. All external resources used in the project, including research papers, open-source repositories, datasets, and any content or code generated using AI tools, e.g., ChatGPT, GitHub Copilot, Claude, Gemini, must be *clearly cited* in the final submission. The final report must also include *a clear breakdown of individual group member contributions.* Any lack of transparency in the use of external resources or in reporting group contributions will be considered academic dishonesty and will significantly impact the final evaluation.

Topic	Personalized Text-to-Speech using VAE or Diffusion Models
Category	Tiny AI Products
Supervisor	Amirhosein Rostami

Objective Design and implement a simplified text-to-speech (TTS) system that generates speech audio conditioned on speaker identity [5]. The project should use a generative model, e.g., a VAE or diffusion-based, to synthesize speech features, which can then be converted into audio using available Vocoders.

Motivation Personalized speech synthesis is an active area of research with wide applications, e.g., voice assistants and virtual agents. This project provides hands-on experience in integrating generative models with real-world sequence generation tasks, allowing the same text to be spoken in different speaker styles.

Requirements The final submission should address the following requirements while allowing creative freedom in technical details.

1. Implement a generative model, e.g., VAE or diffusion model, to generate intermediate speech representations.
2. Condition generation on speaker identity using speaker embeddings.
3. Construct a small-scale speech dataset with diverse speaker identities from open resources with *public permission*, e.g., VCTK [3] or LibriTTS [4].
4. Use a pretrained Vocoder, e.g., HiFi-GAN [1] or WaveGlow[2] to convert predicted embeddings into audio.
5. Evaluate the model based on standard quality metrics.

Milestones

1. *Literature review and dataset preparation.* Understand generative models in the context of sequence generation. Prepare a small speaker-labeled dataset and extract audio features.
2. *Speaker representation and conditioning.* Select a representation for speaker identity, and integrate it into the generative model to enable speaker-conditioned synthesis.

3. *Model implementation.* Implement the generative model and integrate speaker conditioning. Train the model.
4. *Synthesis and evaluation.* Use a pretrained Vocoder to synthesize audio. Evaluate quality and personalization using suitable metrics.
5. *Final report.* Document all experiments, model design, and elaborate final performance and practical limitations.

Optional Enhancements

- Explore fine-tuning on a very small number of speaker samples ("low-shot personalization").
- Use contrastive loss or clustering techniques to improve speaker conditioning quality.
- Add a user-facing interface to enter text and select speaker identity (e.g., simple Streamlit app).

Submission Guidelines The main body of work is submitted through Git. In addition, each group submits a final paper and gives a presentation. In this respect, please follow these steps.

- Each group must maintain a Git repository, e.g., GitHub or GitLab, for the project. By the time of final submission, the repository should have
  - Well-documented codebase
  - Clear `README.md` with setup and usage instructions
  - A `requirements.txt` file listing all required packages or an `environment.yaml` file with a reproducible environment setup
  - Demo script or notebook showing sample input-output
  - *If applicable,* a `/doc` folder with extended documentation
- A final report (maximum *5 pages*) must be submitted in a PDF format. The report should be written in the provided formal style, including an abstract, introduction, method, experiments, results, and conclusion.
  Important: Submissions that do not use template are considered *incomplete.*
- A 5-minute presentation (maximum *5 slides including the title slide*) is given on the internal seminar on Week 14, i.e., *Aug 4 to Aug 8,* by the group. For presentation, any template can be used.

Final Notes While planning for the milestones please consider the following points.

1. You are encouraged to explore innovative approaches to conditioning or generation as long as the core objectives are met.
2. While computational resources are limited, carefully chosen datasets and training setups can make even diffusion models feasible. Trade-offs, e.g., resolution, training steps, are expected and should be justified.
3. Teams are expected to manage their computing needs and are advised to perform early tests to estimate runtime and training feasibility. As graduate students, team members can use facilities provided by the university, e.g., ECE Facility. Teams are expected to inform themselves about the limitations of the available computing resources and design the model accordingly.

References

[1] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 33:17022–17033, 2020.

[2] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. WaveGlow: A flow-based generative network for speech synthesis. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621. IEEE, 2019.

[3] Junichi Yamagishi, Christophe Veaux, and Kirsten MacDonald. CSTR VCTK Corpus: English multi-speaker corpus for CSTR voice cloning toolkit, 2019.

[4] Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. Libritts: A corpus derived from librispeech for text-to-speech. *arXiv preprint arXiv:1904.02882*, 2019.

[5] Chenshuang Zhang, Chaoning Zhang, Sheng Zheng, Mengchun Zhang, Maryam Qamar, Sung-Ho Bae, and In So Kweon. A survey on audio diffusion models: Text to speech synthesis and enhancement in generative AI. *arXiv preprint arXiv:2303.13336*, 2023.