[Fall 2024] CS172 Final Project

November 2, 2024

Acknowledgements

1. **Important Events**:
   - **Milestone Presentation**
     - Time: **Week 12 Mon. in class, 2024/12/2**
     - Presentation Duration: 5min
   - **Presentation Materials Submission**
     - Deadline: **Week 15 Sun., 2024/12/29 23:59:00**
     - Submission Contents:
       (a) Presentation Slides and Contribution List
         * Contribution List is a ONE-page PDF to clarify the individual contribution
         * submit in Blackboard in PDF format

       (b) Presentation and Demo Video
         * Record Requirements: 5min (in English), and each team member should be involved in the presentation for his/her part.
         * submit to ShanghaiTech cloud disk [link] and named in GroupID_LeaderName.zip
   - **Final Presentation**
     - Time: **Week 16 Mon. in class, 2024/12/30**
     - Presentation Duration: 5min
     - Requirements: Each team member should be involved in the presentation for his/her part.
   - **Report and Code Submission**
     - Deadline: **Week 16 Sun., 2025/1/5 23:59:00**
     - Submission Contents:
       (a) Report
         * in CVPR template (uploaded in Blackboard), 4-10 pages (not including reference)
         * submit in Blackboard in PDF format

       (b) Code zip
         * includes all files and README.md
         * submit to ShanghaiTech cloud disk [link], and named in GroupID_LeaderName.zip

2. **Team Requirements**:
   - Limit to 3-5 students per team.
   - Team registration rules will be released soon in the WeChat group.

3. **Scoring Criteria**:
   - report and code 30%
   - novelty 40%
   - milestone presentation 15%
   - final presentation 15%
4. **Code of Conduct**:
   - Plagiarism or cheating is strictly prohibited!
   - NO fake result is allowed! Make sure your codes can run and are consistent with your solutions.
   - If you use any open-source code, it is important to provide proper attribution and give credit to the original authors. Proper citation and acknowledgment of sources are crucial in research, so be sure to follow best practices and give credit where credit is due.

1 Conditioned Human Motion Generation

Conditioned human motion generation focuses on synthesizing realistic and diverse human movements that align with specific contextual signals, such as text descriptions, object trajectories, or environmental settings. The goal is to produce natural and physically plausible motions while satisfying various conditioning requirements. Key challenges include the complexity of modeling human motion dynamics, ensuring consistency with the given context, and capturing nonverbal communication aspects such as individual styles, intentions, and cultural expressions. Recent advances in deep learning and data-driven approaches have enabled significant progress in generating human-like movements, with applications spanning entertainment, robotics, and virtual environments. We recommend using at least one of the following conditions: text, scene, or dynamic objects.

1.1 Basic

- Select a generative model backbone (e.g., VAE, GAN, Flow, or Diffusion), choose a conditioning signal and corresponding dataset, and train your model.
- Conduct both qualitative and quantitative evaluations of your generated results, following standard practices in SOTA generative model research.

1.2 Advanced Options

- Combine multiple conditioning signals (e.g., text with dynamic object trajectories, or text with environmental scenes).
- Incorporate an optimization module to refine and improve the quality of the generated outputs.
- Experiment with simulation techniques to enhance generation quality (e.g., PHC, ICCV 2023, Zhengyi Luo).

1.3 Reference

1. (siggraph asia 2023) Object Motion Guided Human Motion Synthesis
2. (Nips 2023) HUMANISE: Language-conditioned Human Motion Generation in 3D Scenes
3. (ICCV 2023) Perpetual Humanoid Control for Real-time Simulated Avatars
4. (ICCV 2021) Stochastic Scene-Aware Motion Prediction
5. (ECCV2022) TEMOS: Generating diverse human motions from textual descriptions

2 Video Style Transfer

Video style transfer is a technique that applies the artistic style of an image or painting to each frame of a video, transforming it to match the visual aesthetics of a reference image. Inspired by image-based neural style transfer, video style transfer builds upon these principles but presents additional challenges, particularly with temporal coherence across frames. Flickering, jittering, and inconsistency can disrupt the viewer experience when each frame is styled independently, leading researchers to explore methods that ensure style continuity between frames. Recent advancements in neural networks have allowed for significant progress in video style transfer, enabling real-time processing and improving aesthetic outcomes. This project provides both basic and advanced styling options.

2.1 Basic

- Applies style frame-by-frame without additional coherence mechanisms, leading to potential flickering but faster results.
- Popular neural networks such as convolutional neural networks (CNNs) and feedforward networks are commonly used here.
- Utilize some temporal loss function to maintain the frame consistency.

2.2 Advanced Options

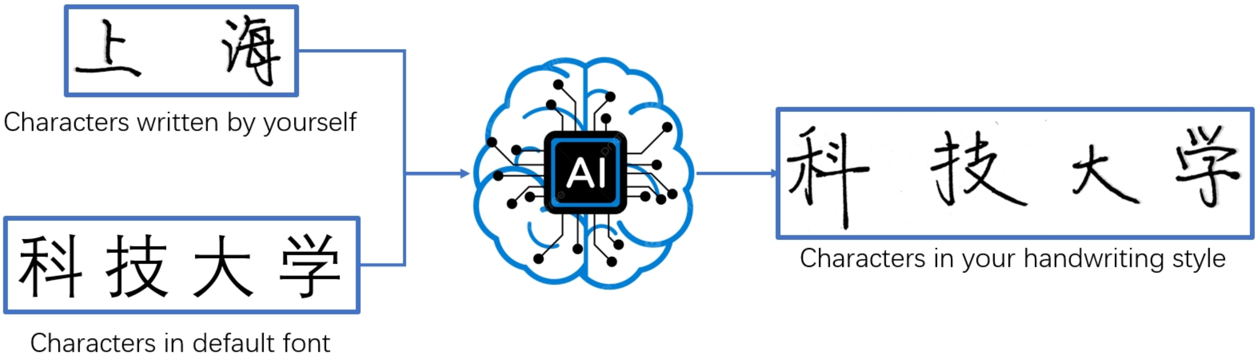
- Advanced options in video style transfer focus on enhancing temporal coherence and speed.
- Advanced neural networks (RNNs, 3D-CNN, Transformer) and temporal coherence constraints further improve results by analyzing dependencies over multiple frames.
- Implement smooth transitions between multiple styles within a video. Try to create seamless style blending without abrupt changes.
- Applying style transfer selectively (e.g., only on the foreground or specific objects) while keeping the background unchanged.

2.3 Reference

1. (AAAI 2020) Consistent Video Style Transfer via Compound Regularization
2. **(CVPR 2016) Image style transfer using convolutional neural networks**
3. (ACM TOG 2022) Vtoonify: Controllable high-resolution portrait video style transfer
4. (AAAI 2021) Arbitrary video style transfer via multi-channel correlation
5. (TNNLS 2023) Exploring the temporal consistency of arbitrary style transfer: A channelwise perspective
6. (ICCV 2017) Coherent Online Video Style Transfer
7. (CVPR 2023) Inversion-based Style Transfer with Diffusion Models
8. (ECCV 2020) Optical Flow Distillation: Towards Efficient and Stable Video Style Transfer

3 Handwritten Text Generation

Every individual's handwriting is unique. Factors such as physiological attributes, psychological states, educational backgrounds, and personal habits all significantly contribute to the distinctiveness of one's penmanship. Consequently, as the proverbial saying goes, "handwriting mirrors the person," indicating that by examining someone's handwriting, we can glean certain insights about the scribe. In today's digital age, where typing often supersedes handwriting in numerous contexts, the uniformity of printed typefaces, although neat, lacks the personal uniqueness that handwriting offers. To solve aforementioned problem, handwriting text generation is proposed, which blends the personalization of traditional handwriting with the efficiency of automated processes, offering a digital format to preserve the authenticity of individual handwriting. This task aims to automatically generate the desired handwritten text images that not only correspond to specific text content but also emulate the calligraphic style of a given exemplar writer (e.g., character slant, cursive join, stroke thickness, and ink color). In this project, you are required to generate a personalized typeface reflecting your own handwriting



3.1 Basic

- Collect a dataset containing Chinese characters written by your own hand. The scale of the dataset is to be determined by yourself.
- Train a model that can generate Chinese characters in the same style as those of the data set.
- Design a simple software which can transfer complete sentences or articles to your handwriting style.

3.2 Advanced Options

In the process of generating individual characters, there is negligible disparity between English and Chinese. The task of generating English alphabets might even be deemed simpler, given the mere total of 26 alphabets in the English language, a number significantly lower than that of Chinese characters. Consequently, the model architecture you have designed can be directly applied to train a model for generating English alphabets. However, the scenario shifts markedly when it comes to sentence generation. Due to the prevalence of cursive writing in English, which entails connecting letters in a flowing manner, it becomes impracticable to independently generate each alphabet then amalgamating them. You are required to devise a solution to overcome this challenge to earn extra points.

3.3 Reference

- (CVPR 2023) Disentangling Writer and Character Styles for Handwriting Generation
- (ECCV 2024) One-DM:One-Shot Diffusion Mimicker for Handwritten Text Generation
- (Arxiv 1308) Generating Sequences With Recurrent Neural Networks