Machine Learning for Sketch Recognition and Sketch Generation

CS3308 Machine Learning

December 17, 2025

1. INTRODUCTION

- Three students at most form a group.
- You only need to complete one of the two tasks (Unimodal task or Multimodal task).
- Deadline for this project is January 4, 2026, 23:59. Each group needs to submit a report PDF on Canvas, and the source code is also required by providing the link to your GitHub repo.
- This project will be evaluated from workload (20%), model performance (20%), results analysis (40%), and report writing (20%). It is more crucial to conduct a reasonable analysis of the experimental results.
- The data required for the Unimodal Task is available in

  Sketch Recognition: QuickDraw414k.zip

https://pan.sjtu.edu.cn/web/share/9941a986e116499ec9a733e6f5114a61

  Sketch Generation: QuickDraw_generation.zip

https://pan.sjtu.edu.cn/web/share/c608de1be86dfc845d4b7b8e74a963d3

2. Unimodal Task: Recognition and Generation of Freehand Sketches

Freehand sketches are characterized by sparsity and abstractness, while the lack of texture makes them more challenging to recognize compared to natural images. Sketches can be represented in forms such as drawing *sequences, images, or graphs*, offering flexibility in developing neural networks for sketch-based applications. You may extract features from a single representation or fuse features from different representations. This task requires you to develop a sketch recognition model and implement a sketch generation model.

A. Sketch Recognition:

Utilize the QuickDraw-414k dataset [1], provided in cloud storage, which is a subset of the large-scale sketch dataset QuickDraw [2] comprising 345 classes. You are required to develop a classification model that can accurately recognize sketches in the test set. For data loading methods of different representations, you can refer to the implementation at:

https://github.com/PengBoXiangShang/torchsketch/tree/master/torchsketch/data/dataloaders/quickdraw/quickdraw_414k

The following references [1, 3-6] may be useful for addressing this problem.

B. Controllable Sketch Generation:

You are required to train a multi-category sketch generation model capable of producing sequential sketches in a controlled manner (models that output pixel images directly do not meet the requirements). Controllability can be achieved by implementing one or more of the following aspects:

- Reconstruction: Given a sketch sequence, image, or graph, reconstruct the drawing sequence of that sketch.
- VAE-like Models [2][7]: Construct a VAE-like model to flexibly obtain latent space encodings, enabling the decoder to generate desired sketches. A simple form of this is interpolation.
- Condition-Guided Diffusion Models [8]: Utilize diffusion models guided by specific conditions.
- Stroke-Level Editing [9][10]: Perform edits at the level of individual sketch strokes.

Evaluation Metrics:

- Model performance can be evaluated through sketch reconstruction.
- At the category level, you may use a sketch recognition model to classify the generated results and determine whether they belong to the same category as the input sketch.

- In addition, you should identify or design at least one instance-level metric for evaluation.

Dataset: Select 5 to 10 category files from the QuickDraw dataset provided in the cloud storage as your training data. For data loading methods, you can refer to:

https://github.com/CMACH508/SketchEdit/blob/master/Dataset.py

and https://github.com/CMACH508/SketchEdit/blob/master/Utils.py

3. Multimodal Task: Implementing Sketch Generation Using Pre-trained Models

With the advent of pre-trained models like CLIP [11] and Stable Diffusion [12], a wide range of intriguing controllable sketch generation applications can be realized. This task challenges you to leverage these models to propose a training-free sketch generation method, focusing on achieving fine-grained control beyond simple text-to-image conversion.

Core Objective:

Design and implement a method that uses pre-trained models to generate freehand sketches without any additional training on sketch data. Your primary goal is to achieve precise controllability over the generation process.

Suggested Directions for Controllability:

- Stroke-Controlled Text-to-Sketch [13][14]: Generate sketches from text prompts while allowing explicit or implicit control over the number of strokes or sketch complexity.
- Sketch Generation with Specific Style [15]: Generate new sketches that match the style of exemplar sketches but with different content. You may provide either a single style or multiple styles as examples.

`Metrics:

- Primary Evaluation Method: If you can identify or design a quantitative metric (e.g., using a pre-trained feature extractor to measure consistency with a target attribute) that aligns with your control objective, use it.
- Alternative Method (User Study): If a suitable quantitative metric cannot be found, you must design and conduct a small-scale user study. Prepare a clear protocol to collect subjective evaluations on criteria like output quality, adherence to the control condition.

4. Project Report

Each group is required to turn in a project report with your main ideas, utilized methods and algorithms, experimental settings, experimental results, and your discussion about the results. The project report (.pdf) can be written either in English or in Chinese. At the end of the report, please attach the contribution of each member as a percentage. And the work done by each student needs to be clarified. For example,

Name	Student ID	Score	Work
A	00001	30%	-
B	00002	30%	-
C	00003	40%	-

You are also required to submit the source code of your model by providing the link to your GitHub repo in the report. If you do not know how to use GitHub, please visit its tutorial (https://guides.github.com/activities/hello-world/) for some advice.

5. Reference Material

[1] Xu, Peng, Chaitanya K. Joshi, and Xavier Bresson. "Multigraph transformer for free-hand sketch recognition." IEEE Transactions on Neural Networks and Learning Systems 33.10 (2021): 5150-5161.

[2] Ha, David, and Douglas Eck. "A neural representation of sketch drawings." *arXiv preprint arXiv:1704.03477* (2017).

[3] Yu, Qian, et al. "Sketch-a-net: A deep neural network that beats humans." *International journal of computer vision* 122.3 (2017): 411-425.

[4] Yang, Lan, et al. "S3Net: Graph Representational Network For Sketch Recognition." *ICME*. 2020.

[5] Li, Lei, et al. "Sketch-R2CNN: An RNN-rasterization-CNN architecture for vector sketch recognition." *IEEE transactions on visualization and computer graphics* 27.9 (2020): 3745-3754.

[6] Li, Tengjie, Shikui Tu, and Lei Xu. "SketchMLP: effectively utilize rasterized images and drawing sequences for sketch recognition." Machine Learning 114.3 (2025): 1-18.

[7] Zang, Sicong, Shikui Tu, and Lei Xu. "Controllable stroke-based sketch synthesis from a self-organized latent space." *Neural Networks* 137 (2021): 138-150.

[8] Wang, Qiang, et al. "Sketchknitter: Vectorized sketch generation with diffusion models." *The eleventh international conference on learning representations*. 2023.

[9] Li, Tengjie, Shikui Tu, and Lei Xu. "SketchEdit: Editing Freehand Sketches at the Stroke-level."

[10] Zang, Sicong, Shuhui Gao, and Zhijun Fang. "Generating Sketches in a Hierarchical Auto-Regressive Process for Flexible Sketch Drawing Manipulation at Stroke-Level." *arXiv preprint arXiv:2511.07889* (2025).

[11] Radford, Alec, et al. "Learning transferable visual models from natural language supervision." *International conference on machine learning*. PmLR, 2021.

[12] Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022.

[13] Vinker, Yael, et al. "Clipasso: Semantically-aware object sketching." *ACM Transactions on Graphics (TOG)* 41.4 (2022): 1-11.

[14] Xing, Ximing, et al. "Diffsketcher: Text guided vector sketch synthesis through latent diffusion models." *Advances in Neural Information Processing Systems* 36 (2023): 15869-15889.

[15] Li, Tengjie, Shikui Tu, and Lei Xu. "Text to Sketch Generation with Multi-Styles." *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.