

## Project 1 - Learned Skeletonization

100 Points Possible

5/1/2025

Attempt 1



In Progress

**NEXT UP: Submit Assignment**

Add Comment

**Unlimited Attempts Allowed**

▼ Details

## Project: Deep Learning-Based Learned Skeletonization of Road Networks

### Overview

Develop a deep learning model that recovers thin skeleton representations from thick, noisy road network images generated using OpenStreetMap (OSM) data. Your approach should focus on handling the inherent ambiguities in skeletonization while leveraging advanced neural network techniques covered in CSE627. This is a group project (1–3 members).

**Submission:**

- Upload your Slides deck via Canvas (export to PDF first).
- Include a link to your Git repository containing the code (organized as Python scripts, not Jupyter notebooks).
- **Note:** Only the slides will be graded.

### Project Details

#### Data Generation

- **OSM Data Extraction:**
  - Select a subset of road types from OSM and retrieve the data.
- **Rasterization:**
  - Render roads as 256×256 pixel images, assigning varying thickness based on attributes (e.g., lane count, width).
- **Distortions:**
  - Introduce realistic distortions (blur, noise, etc.) to simulate real-world imperfections.
- **Ground Truth:**
  - Rasterize the roads with a thickness of one-pixel.

#### Model Approaches

- **Baseline:**
  - Implement a U-Net to directly predict the thin skeleton from the noisy, thick input.
- **Advanced Option (choose one):**
  - **Iterative Thinning:** Train a network to simulate a gradual de-thickening process.
  - **Multi-Task Learning:** Incorporate auxiliary tasks (e.g., distance maps or intersection cues) to support skeleton prediction.

- **Training on Misaligned Data:** Train your model using data where the ground truth skeleton is intentionally misaligned, and design a loss function or post-processing step that compensates for this shift.

## Testing & Verification

- **Verification:**
  - Include a description of how you tested and verified your implementation (e.g., unit tests, visual inspection of intermediate outputs, or controlled experiments with known outcomes).
- **Ensure correctness:**
  - Detail the steps taken to confirm that each component of your pipeline (data generation, network output, metric calculation) is working as expected.

## Evaluation

- **Qualitative Results:**
  - Include slides with visual examples showing the input image, ground truth skeleton, and model prediction.
- **Quantitative Metrics:**
  - **Test Loss:** Measures the quality / generalization using whatever loss you end up minimizing in the project.
  - **Mean Squared Error (MSE):** Computed using the distance transform from the ground-truth thin-line image (not shifted).
  - **Node Precision & Recall:** Evaluate for 1-valent, 2-valent, 3-valent, and 4-valent nodes. First, do a bipartite match between predicted and ground-truth pixels of the same valence. Define a “match” as a prediction within 3 pixels of the ground truth.
  - Optionally, you may also include common segmentation metrics like IoU or the Dice Coefficient.

## Ablation Study

- Perform an ablation study by varying key parameters (e.g., learning rate, loss functions, architecture choices) over abbreviated training runs.
- Present your findings in clear tables/charts.
- Identify the configuration selected for extended training and justify your choice.

## Implementation Requirements

- **Google Slides:**
  - Use the slides as both a progress log and your final presentation.
  - The first slides must display your final qualitative and quantitative results.
  - Slides should cover experimental setup, ablation study, testing/verification, and a concise explanation of your methodology.
  - Use your professional judgment to design the slides—do not ask for additional instructions on slide design.
- **Code:**
  - Organize your project as a Git repository containing Python scripts (no Jupyter notebooks).
  - The code must be runnable on Colab (e.g., via `%run script.py`) and be fast enough for GPU execution.

- Although the code is required, it will not be graded.

## Recommended Search Terms & Resources

- “U-Net architecture PyTorch segmentation”
- “skimage skeletonize usage example”
- “iterative thinning algorithm Python”
- “training on misaligned data deep learning”
- “calculating cross-entropy loss in PyTorch”
- “computing precision and recall for node matching”
- “OpenStreetMap data extraction and rasterization”

## Grading Rubric

Category	Description	Points
<b>Qualitative Results</b>	Clear visual presentation of input, ground truth, and model output.	20
<b>Quantitative Evaluation</b>	Detailed presentation of metrics (XENT, MSE, node precision/recall) with clear explanations.	20
<b>Advanced Option Implementation</b>	Bonus for selecting and properly implementing one advanced option versus only the baseline:  - <b>Baseline Only:</b> Full credit if baseline is well executed.  - <b>Baseline + Advanced:</b> Additional points if one advanced option is implemented (Iterative Thinning, Multi-Task Learning, or Misaligned Data).	20
<b>Ablation Study</b>	Structured analysis of parameter variations and their impact, presented in tables/graphs.	15
<b>Testing &amp; Verification</b>	Clear description of testing methods and verification of implementation correctness.	15
<b>Methodology Explanation</b>	Concise explanation of your approach, data preparation, model design, evaluation strategy, and overall insights.	10
<b>Overall Professional Presentation</b>	The slide deck is organized, coherent, and reflects graduate-level judgment in its design.	10
<b>Total: 100 Points</b>		

### Important Note:

Do not request detailed instructions regarding presentation style. Use your own judgment as graduate students to professionally present and justify your approach.

### Data Generation Code:

Here are 500 images and the script that generates them, you can use this to get started in this project. You would want to do some custom data augmentation but this is a good start.

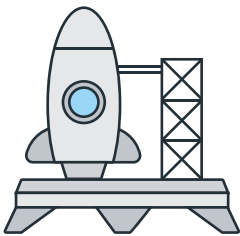
[thinning\\_data.zip \(https://miamioh.instructure.com/courses/230947/files/35396903?wrap=1\)](https://miamioh.instructure.com/courses/230947/files/35396903?wrap=1) ↓  
[https://miamioh.instructure.com/courses/230947/files/35396903/download?download\\_frd=1](https://miamioh.instructure.com/courses/230947/files/35396903/download?download_frd=1)

Keep in mind, this submission will count for everyone in your Project Groups group.

Choose a submission type

Upload

More



Choose a file to upload  
File permitted: PDF

or

 Canvas Files



<https://miamioh.instructure.com/courses/230947/modules/items/5866909>



Assignment

<https://miamioh.instructure.com/courses/230947/modules/iten>