Exercise 05 for MA-INF 2201/MA-MOROB-M04 - Computer Vision WS25/26
24.11.2025
Submission on 14.12.2025

1 Multi-Label MRFs

1.1 Alpha-Expansion Graph Construction

(5 points) Provide a graph structure using the Alpha Expansion algorithm for 5 nodes (a,b,c,d,e) with the initial states: $\boxed{\alpha|\beta|\gamma|\beta|\alpha}$ where the label α is being expanded. Clearly indicate all nodes, edges and capacities of edges in terms of unary U and pairwise P costs.

1.2 The Minimum Cut

(5 points) Show the specific cut for the graph in Part 1.1 that results in the new state: $\boxed{\alpha|\beta|\alpha|\beta|\alpha}$. Write down the mathematical expression for the total cost of this configuration.

2 Interactive Object Segmentation with Graph Cuts

In this part of the exercise, you will implement an interactive image segmentation tool based on the Graph Cut algorithm. You must implement the graph construction and solve the graph using the Python package `PyMaxflow`. Before you start coding, carefully read the tutorial to understand how to use `PyMaxflow`. [1]

You are **not allowed to call** any built-in segmentation, graph-cut and GrabCut functions, i.e. no high-level function that directly returns a segmentation for you. This includes, but is not limited to:

- `cv2.grabCut`, `cv2.watershed`,
- `skimage.segmentation.graph_cut`, `random_walker`, etc.,
- any external MRF/graph-cut wrapper libraries such as `pygco`, `PyStruct`, etc.
- any deep learning based segmentation model such as `Segment Anything Model`, etc.

As a dataset, we will use 6 images from the Geodesic Star Convexity dataset [1]. The `/dataset` folder have 3 subfolders:

1. `/images`: RGB images.
2. `/images-gt`: Ground truth (GT) segmentation masks.
3. `/images-labels`: User annotations (scribbles) for background and foreground.

2.1 The Core Graph Cut Algorithm (without interaction)

(15 points) Implement a class that takes an image and a predefined annotation map (scribbles) to produce a binary segmentation mask. You will also use provided ground truth masks to evaluate accuracy of produced masks. Requirements are as following:

1. **Graph Construction:** You must build a graph where pixels are nodes. You will construct the graph using `PyMaxflow` by defining unary and pairwise cost calculation functions **explicitly**.

[1] `https://pmneila.github.io/PyMaxflow/tutorial.html`

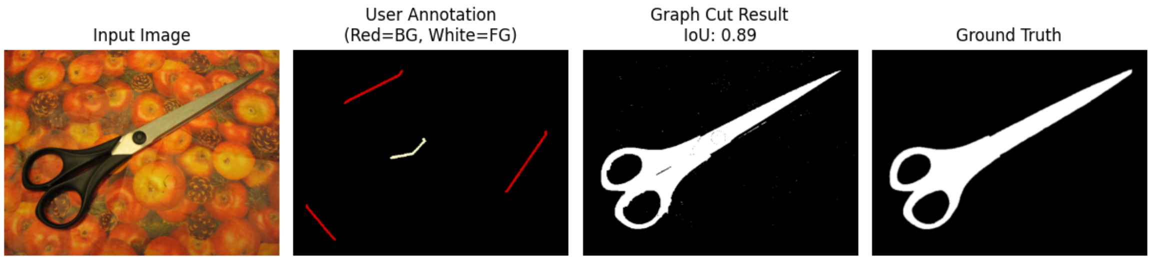

Figure 1: **The Graph Cut Input and Results.** (a) is input image, and (b) is annotated mask which has background (BG) and foreground (FG) scribbles. While (c) is the obtained mask from Graph Cut, (d) is Ground Truth Mask. The evaluation result is 0.89 IoU for this example.

- Unary cost computation: Model the foreground and background color distributions using color histogram and Gaussian Mixture Models (GMMs) based on the provided scribbles. You are allowed to use `sklearn.mixture.GaussianMixture` to learn the GMM parameters. You will compare the results in the evaluation part.
- Pairwise cost computation: Connect neighboring pixels. The weight should be high if pixels are similar and low if they are different.
- User Constraints (scribbles): Set capacities for the annotated pixels to Infinity to enforce user input.

2. **Max-Flow/Min-Cut:** Use the PyMaxflow library to solve the optimization problem.
3. **Evaluation:** Compare your results with the Ground Truth image using the Intersection over Union (IoU) metric. You need to obtain at least average 0.50 IoU score for the dataset.

2.2 Interactive Segmentation Tool (GUI)

(15 points) Build a Python application that allows a user to open an image and interactively segment it. You can use `cv2.setMouseCallback` or any other library for interaction. Minimum functional requirements are as following:

1. **Scribble Interaction:** Allow the user to paint Foreground and Background scribbles in different color.
2. **On-Demand Update:** When the user presses a key, collect the current scribbles, update the GMMs or only the user constraints, and rerun the Graph Cut.
3. **Visualization:** Overlay the binary mask on the image.
4. **Iterative Refinement:** Allow the user to add new strokes and rerun segmentation without restarting.
5. **Reset and Save:** Clear all annotations to reset and save the resulting binary mask.

You are expected to try to obtain better results with your interactive tool. Save your qualitatively best mask for each image, and compare with the ground truth by calculation IoU accuracy. You need to gather your results for your report.

3 Bonus

(5 points) The selected at most 5 submissions will receive 5 extra points:

- **Accuracy:** Highest average IoU score on dataset.
- **Usability:** Most user-friendly, responsive and feature-rich GUI (e.g., real-time updates, brush size control, undo functionality).

4 Deliverable

You are expected to submit the following files:

1. `graphcut_core.py`: Your implementation of the core Graph Cut algorithm.
2. `interactive_tool.py`: The executable script for the GUI.
3. `report.pdf`: A short report which includes:
   - Drawing of the alpha-expansion graph structure and cut for Part 1.
   - Result comparison (qualitative results and IoU score) between the core graph cut algorithm and interactive tool.
   - Brief discussion on which color modeling (histogram vs GMM) and what parameters worked best. What is your conclusion from comparison results?

Notes. Please exclude the data folder from your submission. For grading, in addition to evaluating the report and code correctness, we also value the cleanliness and organization of the code. Most importantly, write your **own** code.

Important: You are not allowed to use any additional python packages beyond the ones imported in the template. (except from GUI-related libraries for this exercise)
Grading: You must submit a code that runs and produces reasonable results.
Plagiarism: Do not cheat and copy the solution from anywhere. We need to verify that the code is yours and that you fully understand it. Any violation of these rules will result in receiving zero points.
Submission: You can complete the exercise in a group of two, but only one submission per group is allowed. Include a README.txt file with your group members in each solution. Points for solutions without a README file will only be given to the uploader.

References

[1] Varun Gulshan, Carsten Rother, Antonio Criminisi, Andrew Blake, and Andrew Zisserman. Geodesic star convexity for interactive image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.