Computer Vision

Image segmentation

November 8, 2023

1 Environment setup

In this assignment, you will

1) implement the mean-shift algorithm and validate it on an image. **Only the libraries specified in the provided script are permitted, and no others may be used**.

2) implement a simplified version of SegNet (https://arxiv.org/abs/1511.00561) and train/validate your model on the provided multi-digit MNIST dataset. Running PyTorch on a CPU would suffice for this assignment and it should take around 10 minutes to train the simplified SegNet on CPUs. However, you are free to use GPU if you have access to one.

2 Mean-Shift Algorithm (Total: 40 pts)

You can find the skeleton code in the provided `mean-shift.zip` file. You will need to implement four placeholder functions to make the skeleton code runnable. If you import additional packages, you will get 0 (out of 40) points.

2.1 Implement the distance Function (10 pts)

The `distance` function should compute the distance between a given point and all other points that are within a specific radius. For simplicity, here we consider this radius to be $+\infty$ such that you don't have to explicitly handle it.

2.2 Implement the gaussian Function (10 pts)

The `gaussian` function should compute the weights of points according to the distance computed by the `distance` function.

2.3 Implement the update_point Function (10 pts)

The `update_point` function should update the point position according to weights computed from the `gaussian` function.

2.4 Experiment with different bandwidth (10 pts)

The `bandwidth` is set to 2.5 in the skeleton code. Experiment with different values (bandwidth = [1, 3, 5, 7]) by yourself and report your findings. Some values may make the code not runnable. Explain why and provide possible solutions.

3 Implement and Train a Simplified Version of SegNet (Total: 60 pts)

You can find the skeleton code in the provided `seg-net.zip` file. Please follow the instructions in `README.md` to install the package and implement a simplified version of SegNet (40 pts). You also need to train the model and validate its accuracy on the validation set (20 pts).

3.1 Implement a Simplified Version of SegNet (40 pts)

You will need to implement a simplified SegNet model in `lib/models/seg_net_lite.py`. Other modules, including dataloader, pre-/post-processing, and training/evaluation scripts are provided with proper comments. Although you will not need to modify other parts of the codebase if you implement SegNet properly, it is still highly recommended to read through the codebase to get a good understanding of how the full pipeline works.

3.2 Train and Validate Your Model (20 pts)

Train and validate your model using the provided training and validation scripts. Details can be found in `README.md` of the provided codebase. For this task, you get 20 points if your final validation accuracy (mean IoU) surpasses 0.8. The points will be scaled linearly between 0-20 if your final validation accuracy is less than 0.8. Please hand in your trained model (a *single* file named `model_best.pth.tar`) along with your modified `seg_net_lite.py` file. Note that if your `seg_net_lite.py` file cannot load your model, then you will get 0 (out of 20) points for this subtask. It is recommended to use a version control tool (e.g. Git) to keep track of your modifications so that you can make sure your final version before hand-in does not modify anything other than `lib/models/seg_net_lite.py`.

4 Hand-in

Please hand in 1) `mean-shift.py` 2) `seg_net_lite.py` 3) `model_best.pth.tar` 4) A short report (2 pages max.) describing your implementation of mean-shift, your findings and results, all in a *single* zip file. You do not need to report anything on the SegNet task, we will just check your implementation and model accuracy.