Computer Vision Assignment 2

Nicolas Wicki

November 2021

1 1 Environment setup

All worked according to the description.

2 Mean-Shift Algorithm

2.1 Implement the distance Function

Easily implemented with torch norm.

2.2 Implement the gaussian Function

At first, I could not find the definition of the gaussian function in this context (where we use the bandwidth as the variance). However, having a closer look at [1] made it clear what was actually meant (also having a look at the Wikipedia page certainly helped).

2.3 Implement the update_point Function

This can nicely be done by using the vector notation `v[:, None]`, which converts a vector to matrix shape and allows for row-wise multiplication.

2.4 Accelerating the Naive Implementation

Implementing a batchified version was certainly more complex. I appreciate the better understanding of tensors I gained along the way. Please do include this exercise in future years, but maybe assign more points, as I think it is a crucial part of setting up an architecture to ensure low runtime. Running the slow implementation of the Mean-Shift Algorithm for 20 iterations takes approximately 30.36 seconds, while the batchified fast implementation using a batch size of 16 takes approximately 19.77 seconds. Both were run on the CPU. If we change the implementation to additionally use the GPU available, we need approximately 7.21 seconds for the slow implementation and for the batchified version with batch size of 16 we need approximately 2.16 seconds.

3 Implement and Train a Simplified Version of SegNet

3.1 Implement a Simplified Version of SegNet

Implementing the architecture was great. To finally construct one as seen in the lecture was very satisfying.

3.2 Train and Validate Your Model

Training the model using a GPU and 12 workers gave me a processing speed of approximately 1700 samples/s. Training for 20 epochs resulted in a final mIoU score of 0.869.

References

[1] A. Chaudhuri, D. Kakde, C. Sadek, L. Gonzalez, and S. Kong, “The mean and median criterion for automatic kernel bandwidth selection for support vector data description,” *CoRR*, vol. abs/1708.05106, 2017. arXiv: `1708.05106`. [Online]. Available: `http://arxiv.org/abs/1708.05106`.