

CV: Lab 05 Writeup

Benedict Armstrong
benedict.armstrong@inf.ethz.ch

Assignment

This weeks assignment was split into two parts:

- Implementing mean-shift for image segmentation
- Implementing a simplified version of SegNet

This writeup only covers the first part.

Mean-Shift

The mean-shift algorithm can be split into five steps:

- Calculate distances between pixels
- Calculate weights for each pixel based on distance
- Calculate the mean of the pixels weighted by the weights
- Update the pixels to the mean
- Repeat until convergence (or number of steps reached)

Calculating Distances

I simply used numpy's `linalg.norm` function to calculate the distances for all pixels from a given pixel.

Calculating Weights

We use the Gaussian kernel to calculate the weights for each pixel. The kernel is defined as:

$$K(x) = e^{-\frac{x^2}{2b}}$$

where b is the bandwidth and x is the distance between the pixels.

Calculating the Mean and Updating Pixels

The mean can be easily be calculated using the following code:

```
np.sum(weight.reshape(-1, 1) * X, axis=0) / np.sum(weight)
```

this result is then used to update the pixels.

Results

Now for the interesting part, the results. I ran the algorithm on the provided image:



Figure 1: Original Image



Figure 2: Mean-shift with bandwidth = 1



Figure 3: Mean-shift with bandwidth = 3



Figure 4: Mean-shift with bandwidth = 5



Figure 5: Mean-shift with bandwidth = 7

A bandwidth of one does not lead to good results after 15 steps. Three, five

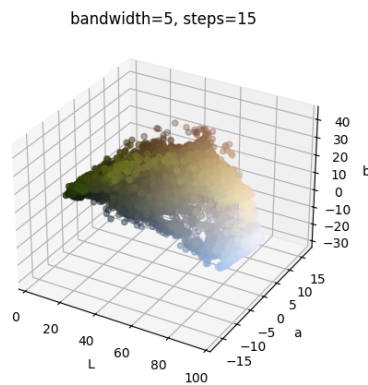


Figure 6: 3D Scatter plot of the colors before mean-shift

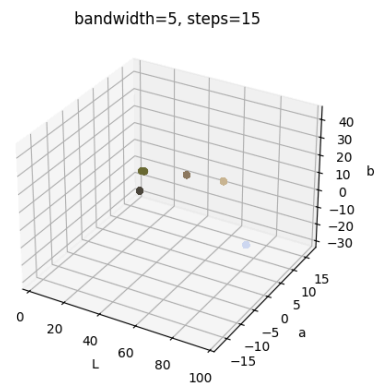


Figure 7: 3D Scatter plot of the colors after 15 steps of mean-shift with bandwidth = 5

We can plot the color of each pixel on a 3D scatter plot to see how the colors are grouped. The first image shows the colors before mean-shift is applied, and the second shows the colors after 15 steps of mean-shift with a bandwidth of 5. We can see that the colors are grouped into 5 clusters.

I've also made fun little [GIF](#) showing the evolution of the algorithm.