# Hard Image Segmentation using Graph Cuts

Brian Mackwan, Ekansh Ravi Shankar

1 May 2024

# Network Flow and Min Cut

We convert our image to a flow network, and we run a Max-Flow Min-Cut algorithm on this network. The algorithm used is Edmonds-Karp, which is strictly polynomial in time, $|V||E|^2$, which is required due to the large input size.

# Conversion from Image to Graph

We will create a unique graph $G = (V, E)$ from the input image as follows

1. Create as many vertices as the pixels in the image, and add two extra vertices as well, one for the source, $s$, and one for the sink, $t$.
2. We will now create two types of edges. $n$-links and $t$-links.
    1. $n$-links are edges that connect two vertices which are neighbours in a 4-neighbourhood system.
    2. $t$-links connect $s$ or $t$ to the vertices depending on whether the pixel is part of the object(obj) or the background(bkg), which will be decided initially using a seed.
3. We will now assign specific weights to all these edges, which will be described in the next slide.

# Assigning Weights to Edges

For $n$-links, we want the weight to be small if there is a boundary between object and background, because we want that edge to be included in our graph cut. Let the weight between vertex $p$ and $q$ be denoted by $B_{p,q}$.

$$B_{p,q} \propto \exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right)$$

For $t$-links, we have to assign infinite weight to all the seeded edges. Ideally, we would have infinite weights, but for the implementation we have kept the value as $max(\{B_{p,q}|(p,q) \in E\}) + 1$, which will not change the answer.

# Evaluating the Cut

We know that the minimum cut is exactly the segmentation we want. To calculate all the edges in this minimum cut, we have used the Depth First Search algorithm after running Edmonds Karp on the graph that has been calculated, which gives us the boundary we need.

# Calculating the Hyperparameters

After some testing, we found the following equation to work best.

$$B_{p,q} = \sigma^2 \exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right)$$

In addition, instead of using a constant value for $\sigma$, we found much better results using $\sigma$ approximated as the average value of $|I_p - I_q|$ for all neighbours $p, q$.

# Results - Image of a Baby

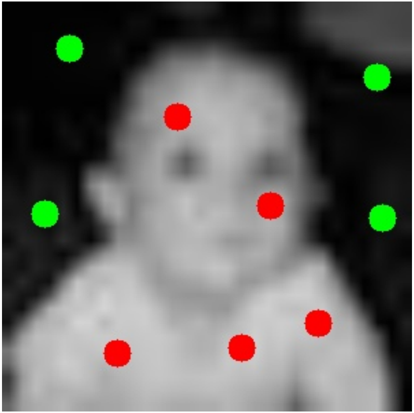

Figure: Seeded Image

Figure: Segmented Image with Cuts

# Results - Brain Image Segmentation

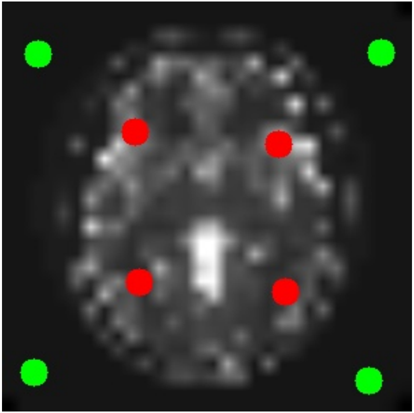

Figure: Seeded Image

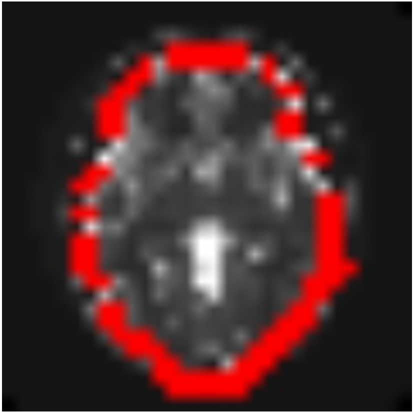

Figure: Segmented Image with Cuts

# Results - Lung Image Segmentation

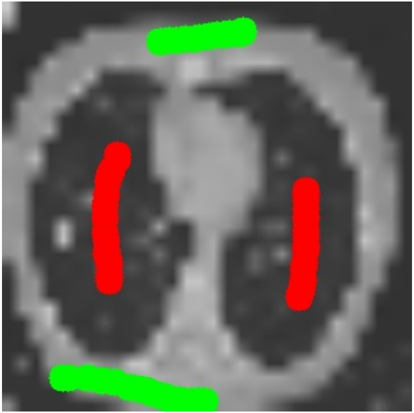

Figure: Seeded Image

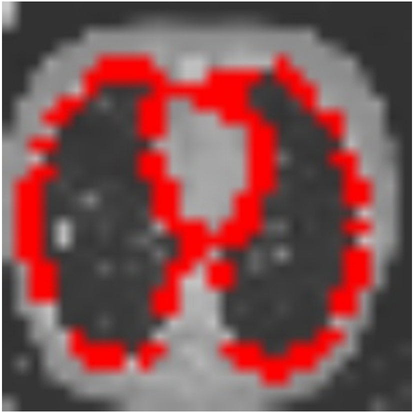

Figure: Segmented Image with Cuts

# Improvements and Other Techniques

One improvement that can be made in the graph construction, is that instead of using infinite weight edges for seeded $t$-links, we could instead assign infinite weight edges for the seeded vertices, and use a Gaussian Mixture Model with respect to these vertices and assign edges from the source or sink to other vertices, proportional to the value using the GMM.

Another possible tweak in the model, is using directed graphs instead of undirected graphs. The directed link between vertices $p$ and $q$, in this case, can be assigned a weight $w_{p,q} = max(0, B_{p,q} - h(I_p - I_q))$. The value of $h$ can be tweaked in this case.

It is also possible to implement better algorithms to solve the max-flow min-cut problem.

# References


1. https://www.csd.uwo.ca/ yboykov/Papers/miccai00.pdf
2. https://link.springer.com/content/pdf/10.1007/s11263-006-7934-5.pdf