EE 5175 Image Signal Processing

Mini Project

Mohammed Khandwawala EE16B117

Contents

1 Image Segmentation

The paper implemented in this project is ™Normalized Cuts and Image Segmentation" by Shi and Malik. The approach used in the paper to segment the image is to realize the image as a weighted undirected graph. Each pixel is connected by every nearby pixel (in a spatial neighbourhood of r) with a weight dependent of difference in intensity values and distance between the pixels. Now segmenting the image id the same problem as finding min cut of this graph.

Minimum Cut is defined as the disjoint partition of the graph that minimizes the sum of weights of edges connecting the two partitions. Suppose we have graph G and the two partitions are A and B and minimizes

$$cut(A,B) = \sum_{u \in A, v \in B} w(u,v)$$

Since we want to keep segmenting this graph in the problem , the authors noticed problem with using the above value of cut as it favours segmenting small set of nodes. To avoid this the authors used normalized cut instead V is the set of all vertices in the graph where as A and B are set of vertices after partition.

$$Ncut = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(V,B)}$$

$$assoc(A,V) = \sum_{u \in A, t \in V} w(u,t)$$

We all define another measure to see how strong is the connection of vertices in the same partitions, Normalized Association

$$Nassoc = \frac{assoc(A,A)}{assoc(A,V)} + \frac{assoc(B,B)}{assoc(V,B)}$$

with some algebric manipulation we can show that

$$Ncut(A,B) = 2 - Nassoc(A,B)$$

Hence the problem we are solving is to minimize disassociation or maximizing normalized association.

This is a NP hard problem and only approximate solution can be found.

The weight of the edge between two pixel i and j in defined as

$$W(i,j) = \begin{cases} e^{\frac{||F(i)-F(j)||^2}{\sigma_i^2}} \times e^{\frac{||X(i)-X(j)||^2}{\sigma_x^2}} & ||X(i) - X(j)|| \leq r \\ 0 & else \end{cases}$$

F(i) is the intensity of pixel i and X(i) in the coordinate of pixel i in the image. $sigma_i$ and $sigma_x$ are the hyperparameter to control the sensitivity of the weight component of image intensity and distance respectively. r

The normalized min cut by substituting D ,W and y can be simplified as

$$Ncut(y) = \frac{y^T(D-W)y}{y^T Dy} - (1)$$

D is a diagonal matrix of dimension n*m x n*m , where n,m are image dimensions.Diagonal entries contains the sum of weights of outgoing edges from pixel corresponding to that diagonal index. W is weight matrix and y = (1 + x) - b(1 - x) and b = k/1-k , x is the indicator vector 1 for A and -1 for B. which makes y indicator vector with value 1 or -b.

To solve (1) , we can observe that it is the Rayleigh quotient if we replace y by $D^{\frac{1}{2}}x$ and D - W is real and symmetric, and it can be minimized by the smallest eigenvalue of the equation.
Theorem *Let A be a real symmetric matrix. Under the constraint that xx is orthogonal to the j-1 smallest eigenvectors* x_1*,...*x_{j-1}*,the quotient* $\frac{x^T Ax}{x^T x}$ *is minimized by the next smallest eigenvector* x_j *and its minimum value is the corresponding eigenvalue* λ_j*.*

And hence the equation 1 will be minimized by the eigenvector corresponding to the second smallest eigenvalue. We can find appropriate threshold on this eigenvector values to partition the image into two, For more partitions similar procedure can be repeated on the each partition. For stopping we will use two hyperparameters segmentArea to ensure minimum size of partition and minNcut to ensure maximum minimum value to be a valid cut.

2 Algorithm

- Initialize image and Hyperparameters
- create sparse matrices D and W
- Iterate on each pixel
  - create a grid of r^2 values around the selected pixel
  - select valid indices from the grid , applying boundary conditions
  - compute distance of the given pixel within this grid
  - keep the items within r^2 distance
  - compute difference in intensity in pixel values in this neighbourhood
  - update the weights of the edges outgoing from this pixel
  - sum of the weights outgoing will be updated in diagonal of D corresponding to this pixel.
- Solve the equation (D-W)y = Dy , (Used inbuit eigns function of matlab) for the second smallest eigenvalue.
- Select a threshold t , to partition. initialize t with median of the second eigenvalue and then search for t that minimizes NcutValue.
- Use the above threshold and partition the pixels with value higher than threshold in the corresponding index of the eigenvector.
- For the two partitions , recursively repeat the above steps. W is already computed and W for the segmented can be extracted from the same.
- For stopping , check the size of the partition to be above a threshold and NcutValue is below a threshold.

3 Segmented Images

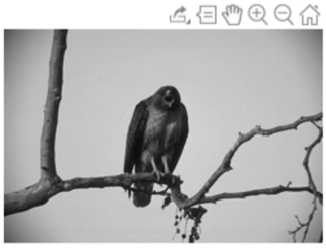

Figure 1: Full Image of Greyscale Crow

Figure 2: Segmented Image of Greyscale Crow , r = 5 ,$\sigma_x = 4, \sigma_i = 5, segArea = 1750, segNcut = 0.7$

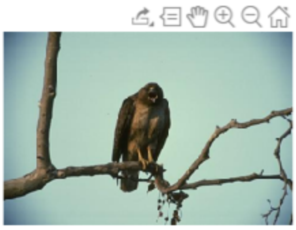

Figure 3: Full Image of RGB crow

Figure 4: Segmented Image of RGB crow , r = 5 ,$\sigma_x = 5, \sigma_i = 8, segArea = 1750, segNcut = 0.7$

Figure 5: Full Image of Race Track RGB

Figure 6: Segmented Image of Race Track RGB , r = 4 ,$\sigma_x = 5, \sigma_i = 4, segArea = 2500, segNcut = 0.3$

Figure 7: Full Image of Church

Figure 8: Segmented Image of church , r = 5 ,$\sigma_x = 3, \sigma_i = 4.5, segArea = 5000, segNcut = 0.2$

Figure 9: Full Image of a building

Figure 10: Segmented image of a building , r = 5 ,$\sigma_x = 5, \sigma_i = 8, segArea = 2000, segNcut = 0.23$

4 References

J. Shi and J. Malik, $^{\mathrm{TM}}$Normalized Cuts and Image Segmentation, 2000