Theoretical Thinking II **Kleanthis Avramidis**
CSCI 670 - Advanced Analysis of Algorithms USC ID: 2680-5898-81
PhD Program in Computer Science Date: 11/29/2021, Fall

1 Image Segmentation & Clustering

One of the main issues of interest in Computer Vision involves coming up with image representations that are semantically compact and expressive, so that they are easily detected and analyzed. The process of partitioning a digital image into multiple segments is called *Image Segmentation*. The term is essentially used for a wide range of similar-defined tasks, but in the context of this report, Image Segmentation will denote the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics (objects or semantic entities), a task also referred to as *pixel clustering*. The result is a set of disjoint segments that collectively cover the entire image and each of the pixels in a region has their similarity measured with respect to some quantifiable property, such as color, texture or intensity, usually defined by the respective application of the segmentation algorithm.

2 Algorithmic Formulation

A rule of thumb for Image Segmentation by clustering is to represent each pixel with a feature vector. This vector contains all measures that are considered relevant in sufficiently describing a pixel. Examples of such feature vectors include the intensity value of the pixel; its location in the image, and also information about its neighboring pixels, from which one could define texture or shape. We then just cluster these vectors and, to this end, one could define two base algorithms, referred to as *agglomerative clustering* and *decisive clustering*. In the first case we follow a bottom-up approach by defining each pixel to be a separate cluster and then iteratively merging clusters of minimum inter-cluster distance. Symmetrically, in decisive clustering we begin by considering the whole image as a single cluster and then iteratively splitting clusters of maximum distance. However, this formulation poses challenging algorithmic questions as well:

- *What is an optimized measure of inter-cluster distance?* A mathematical definition is needed, since both algorithms depend on that definition. Again, approaches vary in the literature according to the task in hand. Some of the commonly used methods include nearest-neighbor selection, euclidean distance between extreme cluster points, or distances inferred from the centerpoints of each cluster.
- *How many clusters are there?* There is an inherent variability in the content presented in different images or even in the perceived content within the same image. The basic algorithms we described produce a hierarchy of clusters that can be parsed to define their optimal number. But since this would involve an enormous number of pixels, other heuristic methods have been introduced, based on features such as boundaries on cluster size and inherent variance.

An early segmentation algorithm that has had wide impact in the field is the watershed algorithm, which computes the gradient magnitude $||\nabla I||$ of an image I. Zeros in this map are locally extreme intensity values that we take as seeds and assign unique labels. Next, we assign pixels to seeds (like filling a height map with water) where each pixel gets the label of the seed that is hit first. The algorithm has been criticized for producing way too many segments, but a few variants continued to be used as they are efficient and robust models. Another method stems from the correspondence of clustering as defined here with vector quantization. The so-called k-means algorithm assigns clusters by iteratively minimizing the within cluster distance of the feature vectors and updating the cluster centerpoints accordingly. In each epoch, each pixel goes to the segment represented by the cluster centerpoint that claims its feature vector. Despite being NP-Hard, k-means has proven to be fast and efficient given certain heuristics. The main consequence of using k-means is that we know a-priori how many segments there will be as we initialize the algorithm with their number. This might be helpful on some tasks, for example in image compression, but usually the optimal number of clusters cannot be defined, as noted above. Nevertheless, the most significant advances in the field emerged by mathematically formulating segmentation as a graph problem.

2.1 Graph-based Formulation

Since clustering is shown to depend on similarity of data points, it is plausible to assume that this pixel comparison should be done in locally, effectively reducing the problem to graph (grid) modeling, where each pixel is a vertex, connected to its neighbors through an edge of a weight analogous to the measured similarity. Clustering then reduces to segmenting the graph into connected components. Similar to the base algorithms we can define decisive and agglomerative approaches in segmenting graphs:

2.1.1 Normalized Cuts

A decisive clustering approach would introduce cuts to the image graph, where the weight of the cut (i.e., induced similarity) is minimized. In graph theory this is well defined as the min-cut problem. However, this approach in its own does not work well, since one can get very good cuts by just cutting off small groups of pixels, thus not balancing the difference between segments with the coherence within them. Shi and Malik (2000) responded by proposing Normalized Cuts, which cut the graph into two connected components such that the cut cost is a small fraction of the total similarity within each group. We formalize this as decomposing a graph G into components A and B, with their decomposition score s being

$$s = \frac{\mathrm{cut}(A,B)}{\mathrm{assoc}(A,G)} + \frac{\mathrm{cut}(A,B)}{\mathrm{assoc}(B,G)}\,.$$

Here, $\mathrm{cut}(A,B)$ defines the sum of weights of all edges in G that have one end in A and one in B and $\mathrm{assoc}(A,G)$ is just the sum of weights of all edges that have one end in A. s is small if the cut separates two components that have few edges of low weight between them and many internal edges of high weight. Hence, we would like to find the cut that minimizes this criterion, called a normalized cut. While being successful, this combinatorial optimization problem is NP-Complete, even for grid graphs like images.

2.1.2 The Agglomerative Method

Felzenszwalb and Huttenlocher (2004) showed a rather neat and more efficient approach of utilizing graph structure to build an agglomerative image segmenter, where the edge weight now measures dissimilarity. Initially every pixel forms a cluster and each created cluster is a graph component, formed from all edges that start and end inside itself. Hence the difference between two components C_1, C_2 can be the minimum weight (w) edge (v_1, v_2) connecting them: $\mathrm{diff}(C_1, C_2) = w(v_1, v_2)$. We also define the internal difference of a component C to be the largest weight in its minimum spanning tree M: $\mathrm{int}(C) = \max_{e \in M(C)} w(e)$. The algorithm iteratively merges clusters by sorting all edges in order of non-decreasing weight. For each edge, starting from the smallest, we consider the clusters at either end of the edge. If there are distinct clusters at each end, then they are merged, only if the edge weight is small compared to the intra-cluster differences. This algorithm is notably fast and accurate, while requiring only some care for small clusters, whose internal distances might be implausibly small. The issue is addressed by an additive bias $\epsilon > 0$.

3 State of the Art & Future

Deep Learning has revolutionized the common practice in Computer Vision with the introduction of Convolutional Neural Networks (CNN) to efficiently handle image data. CNNs employ representation learning based on convolutional kernels to automatically find relevant features that are spatially correlated. Layers close to the input generate low-level features like edges, while layers close to the output generate high-level features like complex shapes and objects, crucial for Image Segmentation. State of the art learning algorithms share intuition with the aforementioned methods, as they take advantage of pixel correlations in local level: Fully Convolutional and Deconvolutional Networks like U-Net have been introduced to transfer spatially correlated information through their layers and eventually reconstruct a segmented image through transpose operations. Mask-RCNN utilizes a helper region detection algorithm to locate objects, while the most recent DeepLab family introduces depth-wise convolutions to infer information from image channels. Despite the vast improvements in the field, open questions and directions remain for future work, one of the most prominent being the utilization of Graph Neural Networks in representing and enhancing the traditional graph-based formulation of the task and the plausible intuitions outlined.