Segmentation via Clustering

Hanwen Zhao

Abstract

In this report we will use clustering algorithms to segment images and use these segmentation to identify foreground and background objects. Eventually, we will transfer foreground objects from one image to another.

1 Clustering Algorithms

Clustering is a machine learning technique which separate data points into groups. With a given data set, we can classify them into specific groups as we demand. In this section, we will discuss two popular clustering algorithms. The approach we used is from bottom up, pixels belong together because they look similar. In this section, we will discuss two different popular clustering algorithms: K-Means Clustering and Hierarchical Agglomerative Clustering.

1.1 K-Means Clustering

The idea behind the K-Means Clustering Algorithm is very simple:

- we randomly initialize the k cluster centers
- for each point p, find and assign to closest cluster c_i
- set c_i to be the mean of the points in cluster i, this is the new center
- repeat steps until c_i remain the same

The following figure shows the iteration steps of K-Means Clustering Algorithm.

Figure 1: Series of K-Means Iterations

1.2 Hierarchical Agglomerative Clustering Algorithm

The hierarchical agglomerative clustering algorithm is also conceptually simple:

- assign each point to its own cluster
- compute the distance between all pairs of clusters, merge the pair of clusters that are closest to each other
- recomputing the centroids of all clusters and the distances between all pairs of centroids
- repeat steps until the number of clusters meets our requirement

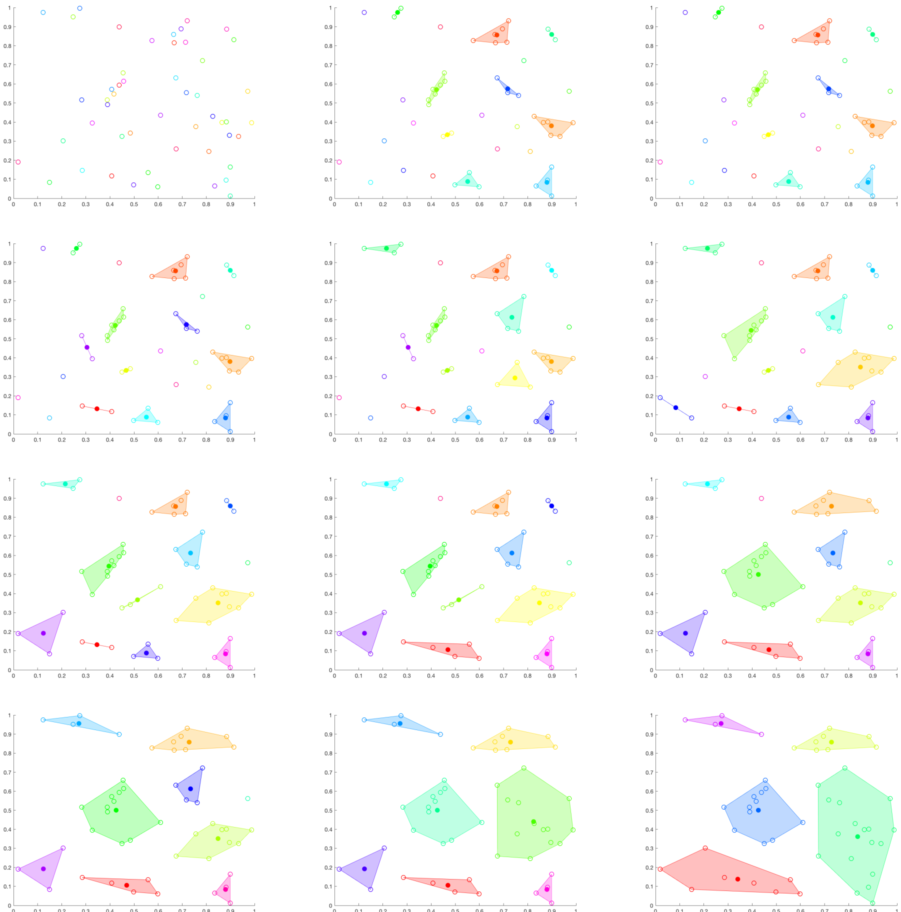

Figure 2: Series of HAC Iterations

2 Pixel Feature Vectors

We need to compute some feature vector for each pixel since we are using clustering algorithm to segment an image. The feature vector for each pixel determine the qualities of segmentation. There are numerous ways to define pixel feature vectors, one of the simplest possible feature vectors for a pixel is the vector of colors for that pixel. In addition, we have the following features can be used to help segment an image.

Color Features (RGB Channels)

The RGB color feature vectors are very important since it contains the primary information of the image, we expect the pixel with similar color could cluster together

Position Features (X, Y Coordinates)
The position feature vector is also an important information in one image, the pixel with similar position (close to each other) can be clustered together

Gradient Features (Magnitudes and Directions)
The gradient feather can be helpful to detecting the any big changes in the gray scale image.

Edge Features
The edge feature is similar to gradient feature, it contains the information of where could be the location of the edges of the objects

Color Features (HSV Channels)
HSV is an alternative representation of the RGB color model, it represents Hue, Saturation, and Lightness

2.1 Feature Normalization

Sometimes we want to combine different types of features into one single feature vector. Usually the features from different source have different rang of values, uneven scaling between different feature vectors may cause clustering algorithms to behave poorly. One way for uneven scaling between different features is to have zeros mean and unit variance. We can use the following equations to calculate normalized feature vector, assume f_{ij} is the value of j^{th} feature for the i^{th} pixel:

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} f_{ij} \tag{1}$$

$$\sigma_j^2 = \frac{1}{n-1}\sum_{i=1}^{n}(f_{ij} - \mu_j)^2 \tag{2}$$

$$\tilde{f}_{ij} = \frac{f_{ij} - \mu_j}{\sigma_j} \tag{3}$$

The method to handle feature normalization is called standardization, there are several other methods, such as re-scaling, mean normalization and scaling to unit length.

3 Image Segmentations

Once we computed feature vector for each pixel, now we can compute a segmentation for the original image by applying a clustering algorithm to compute feature vectors. The following figures shows some example of successful and unsuccessful image segmentations:

3.1 Successful Segmentation

Figure 3: Segmentation K-Means, k=3, Color Feature, Normalized, Resize=1

Figure 4: Segmentation HAC, k=3, Color and Position Feature, Normalized, Resize=0.01

Figure 5: Segmentation K-Means, k=3, Color Feature, Un-Normalized, Resize=1

3.2 Unsuccessful Segmentation

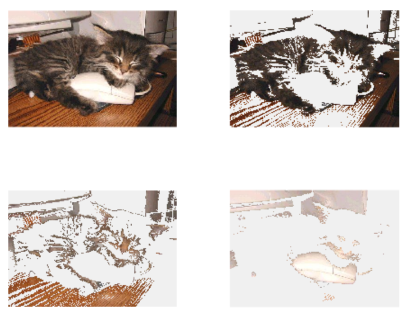

Figure 6: Segmentation K-Means, k=3, Color Feature, Normalized, Resize=1

Figure 7: Segmentation K-Means, k=3, Color and Position Feature, Un-Normalized, Resize=1

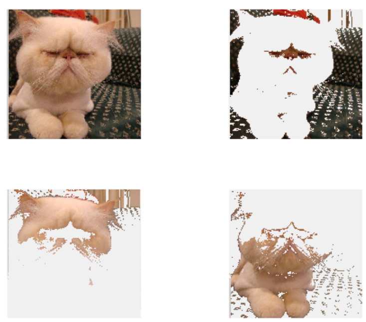

Figure 8: Segmentation K-Means, k=3, Color and Position Feature, Normalized, Resize=1

From figures as shown above, we can see for relative simple image (less objects), a small k would be sufficient. Based on the complexity of the image, different transformation would produce different quality. For more complex image, the color of the cat is not enough to identify the the object (similar color in the scene). With more than one feature transformation, normalization is important since we want each feature have the same weight. Also if we re-sized the image to small, we will not have enough pixel to separate objects.
The selection of clustering algorithm affect the speed of computing significantly. K-Means is noticeable faster than HAC algorithm. However, for complicated situation, HAC outputs better results.

4 Composite Image

Figure 9: Composite Grab Cat Image 1: k=3, K-Means, Position and Color Features, Normalized, Resize=0.2

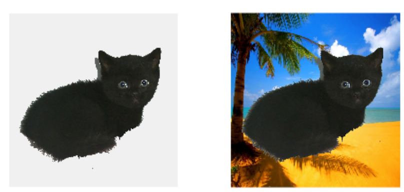

Figure 10: Composite Grab Cat Image 2: k=3, HAC, Color Features, Normalized, Resize=0.02

5 Evaluation of Segmentation Parameters

Feature Transform	Feature Normalization	Clustering Method	Number of Clusters	Max Pixels	Mean Accuracy
Color	Yes	K-Means	10	10000	0.9150
Color	Yes	K-Means	3	10000	0.8566
Color+Position	Yes	K-Means	10	10000	0.9069
Color+Gradient	Yes	K-Means	10	10000	0.8713
Color+Edge	Yes	K-Means	10	10000	0.8924
Color+HSV	Yes	K-Means	10	10000	0.9088
Color	Yes	HAC	10	500	0.8774
Position	Yes	HAC	10	500	0.8273
Color+Position	Yes	HAC	10	500	0.8995
Color+Position+Gradient	Yes	HAC	10	500	0.8585
Color+Position+Gradien+Edge	Yes	HAC	10	500	0.8547
Color+Position+Gradien+Edge+HSV	Yes	HAC	10	500	0.8642

6 Reference

$https://en.wikipedia.org/wiki/Feature\_scaling$