Dieu My Nguyen
Due: 07 April 2019

Homework 5

CSCI 5722
Computer Vision

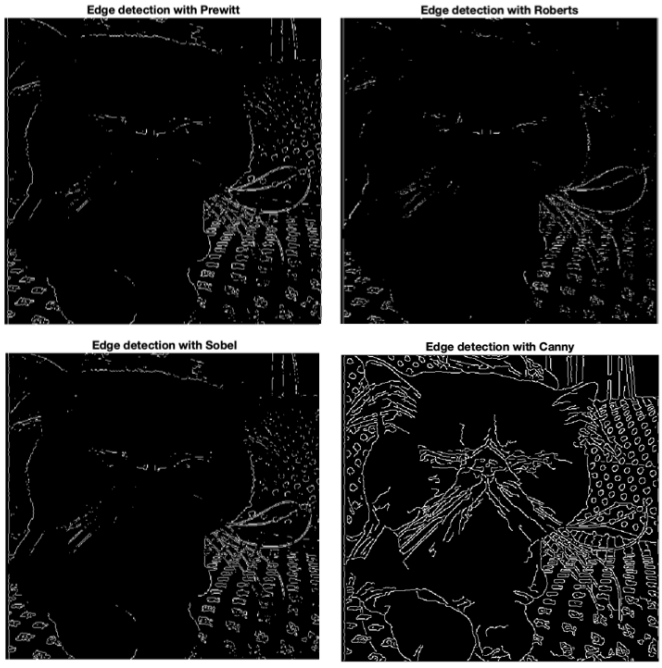
A. Methods for feature vectors

1. Color: This is the simplest but important feature. The RBG channels are invariant to transformations of an image (e.g. scaling, translating, etc.). Color is a basic descriptor that can help differentiate between the different objects in an image.

2. Color & Position: Position is another simple feature for a pixel that we can concatenate with the color feature. Each pixel in the image has the feature vector [R, G, B, x, y]. The mean accuracy for the set of parameters {feature=color, normalization=true, clustering method=k-means, number of clusters=5, max pixel=10,000} is 0.8740. Keeping all parameters constant but adding position to the feature vector, the mean accuracy is 0.8784. Testing on the set of parameters using the HAC algorithm instead of k-means {feature=color, normalization=true, clustering method=HAC, number of clusters=10, max pixel=2500}, the mean accuracy is 0.8831. Adding the position feature increases the accuracy to 0.9113. (Note: This info is also in the table in E.) We'd need to run more repeated tests, but with the given results, it seems that the position feature matters more to improve the HAC algorithm results than for k-means. That's especially interesting since I need to reduce the max pixels dramatically to run the HAC algorithm in a reasonable time.

3. Gradient: This is the directional change in pixel intensity in an image. I added both the gradient magnitude and direction to the feature vector (using default Sobel technique). Image gradients might help with matching textures in an image. However, if we have different lightings or some pixels in the foreground being very similar to some in the background, the image gradient might not help much in segmenting the foreground object. Using the gradient magnitude and direction as the only features, get only reach 0.7490 mean accuracy (Parameters: {feature=gradient, normalization=true, clustering method=HAC, number of clusters=10, max pixel=1000}).

4. Edges: Since we have the image gradient as a feature, I also used the Canny edge detection method to add edges as an additional feature to help the clustering algorithms better separate the cats in the foregrounds from the backgrounds. When we accumulate all the four features, we get a mean accuracy of 0.8805 (Parameters: {feature=all, normalization=true, clustering method=kmeans, number of clusters=5, max pixel=10,000}). This is a 1% increase compared to only using color or color and position, above-mentioned. On the right is a comparison between different edge detection methods. Canny was the best and the chosen method.

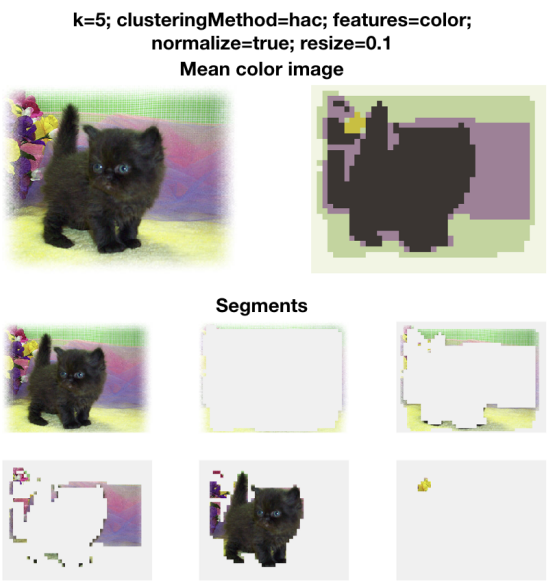


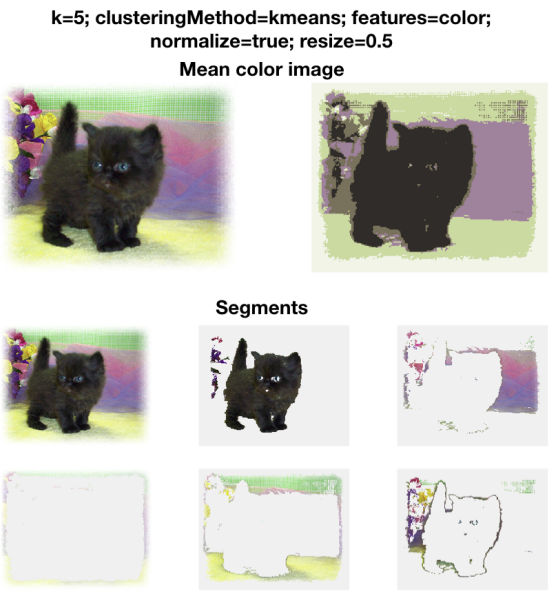
5. Normalization: The normalization

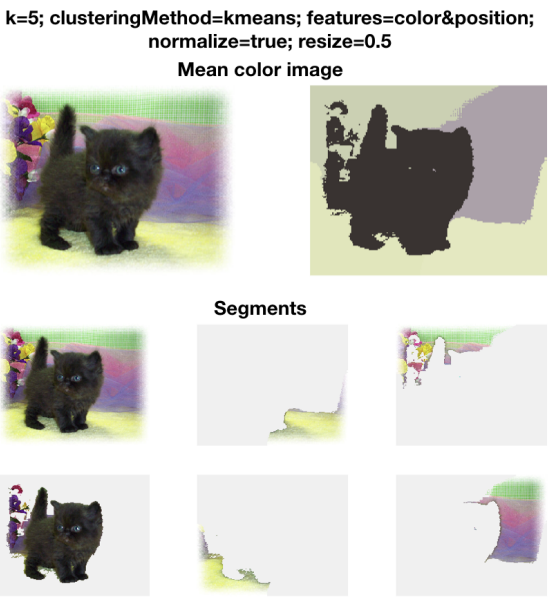
method I employed throughout this assignment is simply the method of standardization to zero mean and unit variance.

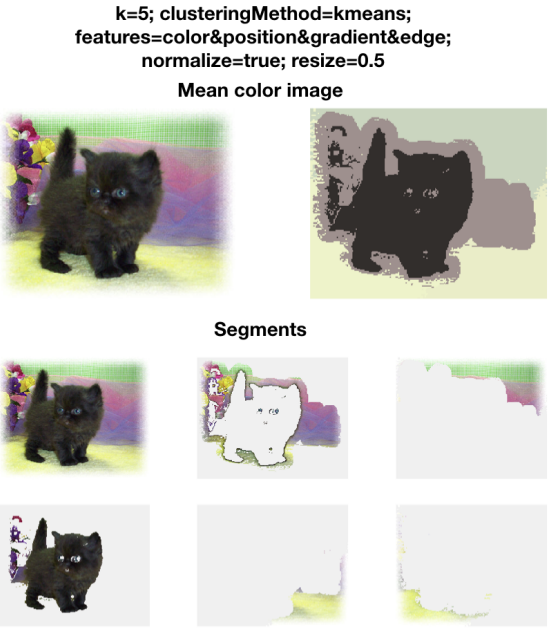
B. Segmentation visualizations

Successful segmentations: Below are a few examples of successful segmentations using different combinations of parameters. It seems that 5 clusters are generally sufficient for both the k-means and HAC algorithms. Also, we achieve decent segmentations using only the color channels as the features (top 2 results).

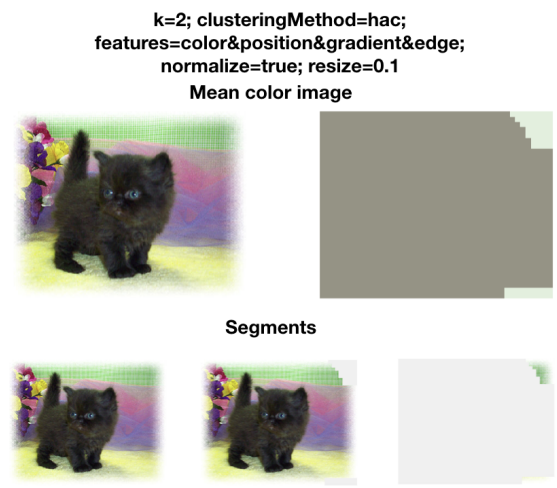


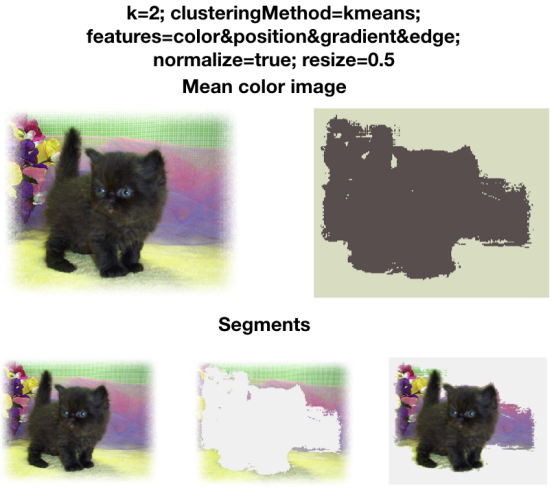


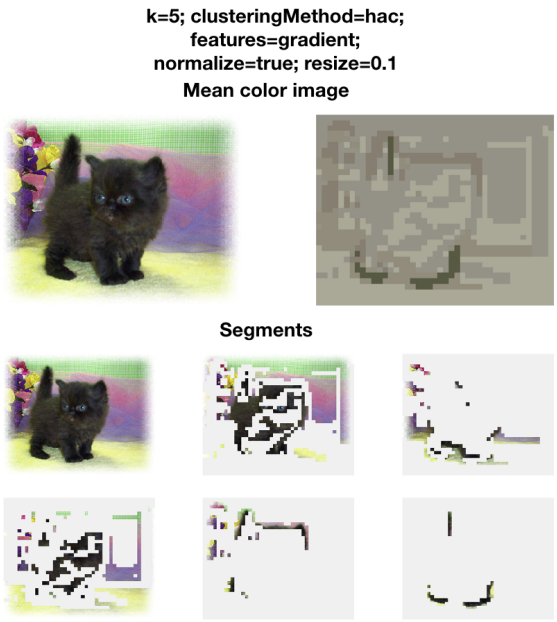


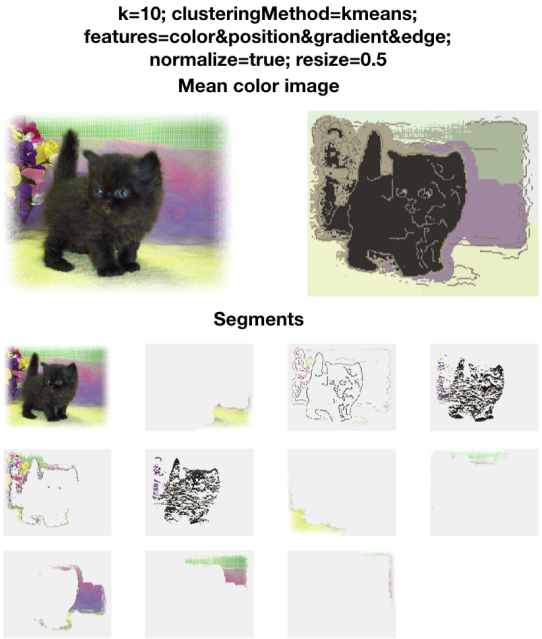


Unsuccessful segmentations: Generally, it seems that both the HAC and k-mean algorithms can yield unsuccessful segmentations when k is too small or too large. Also, using the gradient (or edge) feature alone as in the bottom left result does not yield good segmentation at all.









C. Effects of parameters

1. What effect do each of the segmentation parameters (feature transform, feature normalization, number of clusters, clustering method, resize) have on the quality of the final segmentation?

Using color alone gives surprisingly good final segmentations. Using gradient or edges alone does not. Combining all features may or may not give good results depending on other parameters. Qualitatively, normalization on or off does not seem to affect the segmentation

results. However, the number of clustering and method do, as seen in the visualizations above. Resizing is a necessity for getting results in a timely manner, and surprisingly it does not have significant effect on the quality of segmentations either.

2. How do each of these parameters affect the speed of computing a segmentation?

HAC is definitely a lot slower than k-means, which makes sense since HAC starts with each pixel being its own cluster. K-means takes 1.028351 seconds to run for the black_kitten.jpg image for parameters {feature=all, normalization=true, number of clusters=5, resize=0.1}, whereas HAC takes 46.408310 seconds. For k-means, higher k takes longer, but for HAC, higher takes less time since it can stop running when it hits this number and needs not cluster the points further. Normalization takes a bit longer since it is extra computations; same with the number of features in the feature vector. Resizing definitely affects speed, as larger images will take very long for both algorithms, especially the HAC, and might even cause memory issues in MATLAB.

3. How do the properties of an image affect the difficulty of computing a good segmentation for that image?

The segmentation results definitely vary with the input image. Below is one quantiative evaluation for the parameters {feature=color, normalization=true, clustering method=kmeans, number of clusters=5, resize=0.1}. Generally we don't get good results for cat_mouse.jpg (which has a complicated background), the-black-white-kittens.jpg (in which the white kitten blends in with the white background), young-calico-cat.jpg (which doesn't have a lot of background).

Accuracy for imgs/black-white-kittens2.jpg is 0.9029
Accuracy for imgs/black_kitten.jpg is 0.9718
Accuracy for imgs/black_kitten_star.jpg is 0.9867
Accuracy for imgs/cat-jumping-running-grass.jpg is 0.9521
Accuracy for imgs/cat_bed.jpg is 0.9584
Accuracy for imgs/cat_grumpy.jpg is 0.8675
Accuracy for imgs/cat_march.jpg is 0.9811
Accuracy for imgs/cat_mouse.jpg is 0.6740
Accuracy for imgs/cutest-cat-ever-snoopy-sleeping.jpg is 0.8895
Accuracy for imgs/grey-american-shorthair.jpg is 0.9804
Accuracy for imgs/grey-cat-grass.jpg is 0.9623
Accuracy for imgs/kitten16.jpg is 0.8581
Accuracy for imgs/kitten9.jpg is 0.8243
Accuracy for imgs/stripey-kitty.jpg is 0.8089
Accuracy for imgs/the-black-white-kittens.jpg is 0.7237
Accuracy for imgs/tortoiseshell_shell_cat.jpg is 0.8043
Accuracy for imgs/young-calico-cat.jpg is 0.5998
The mean accuracy for all images is 0.8674

D. Transferring segments

The parameters are listed in the image below for each composite image. Each composite image uses different cat (and dog) and background images. I notice that in a lot of segmentations, the eyes of the animal are missing, probably because they are segmented as a background. The 3rd cat is missing some pixels on its face as well. Overall, a 100% clean segmentation is probably impossible, but these might be good enough as proof of concepts.

k=5; clusteringMethod=kmeans; features=color; normalize=true; resize=0.5

k=5; clusteringMethod=kmeans; features=color; normalize=true; resize=0.5

k=8; clusteringMethod=kmeans; features=color; normalize=true; resize=0.1

k=5; clusteringMethod=kmeans; features=color; normalize=true; resize=0.2

E. Parameter evaluation

Feature Transform	Feature Norm	Clustering Method	Number of Clusters	Max Pixel	Mean Accuracy
Color	True	k-means	2	10000	0.8289
Color	True	k-means	5	10000	0.8740
Color	True	k-means	10	10000	0.8958
Color	False	k-means	2	10000	0.8306
Color	False	k-means	5	10000	0.8770
Color	False	k-means	10	10000	0.8926
Color	True	HAC	10	2500	0.8831
Color	False	HAC	10	2500	0.8859
Color + Position	True	k-means	5	10000	0.8784
Color + Position	False	k-means	5	10000	0.8598
Color + Position	True	HAC	10	2500	0.9113
Color + Position	False	HAC	10	2500	0.8919
All	True	HAC	5	2500	0.9091
All	True	k-means	5	10000	0.8786
Color	True	k-means	5	5000	0.8693
Color	True	k-means	5	50000	0.8668

1. Based on your quantitative experiments, how do each of the segmentation parameters affect the quality of the final foreground-background segmentation?

Note that due to the very long runtime of the HAC algorithm, I limited the evaluations that use it to low max pixel and high k value of 10. Generally, more features in the feature vector increase mean accuracy, though using only color and position work best for both k-means and HAC algorithms (see Color + Position rows). Increasing k also benefits accuracy for the k-means algorithm (see first 3 rows), while turning on or off normalization does not have a great effect (see first 6 rows). The same goes for HAC regarding normalization (see rows 7-8). Changing max pixel also does not affect the mean accuracy by much (0.8693 for 5000 max pixels compared to 0.8668 for 50,000). Interestingly, using all features for the feature vector does not improve accuracy (0.9091 and 0.8786). Overall, the best mean accuracy is 0.9113 using the parameters {feature=color & position, normalization=true, clustering method=HAC, number of clusters=10, max pixels=2500}.

2. Are some images simply more difficult to segment correctly than others? If so, what are the qualities of these images that cause the segmentation algorithms to perform poorly?

I provided the answer to this in C Question 3. Below are more results for a few more sets of parameters to show some trends that persist. The 3 images that have the lowest accuracy are italicized.

Parameters 1: **{feature=color, normalization=true, clustering method=kmeans, number of clusters=5, max pixels=5000}**

Accuracy for imgs/black-white-kittens2.jpg is 0.8977
Accuracy for imgs/black_kitten.jpg is 0.9695
Accuracy for imgs/black_kitten_star.jpg is 0.9837
Accuracy for imgs/cat-jumping-running-grass.jpg is 0.9521
Accuracy for imgs/cat_bed.jpg is 0.9553
Accuracy for imgs/cat_grumpy.jpg is 0.8755
Accuracy for imgs/cat_march.jpg is 0.9822
Accuracy for imgs/cat_mouse.jpg is 0.7148

Accuracy for imgs/cutest-cat-ever-snoopy-sleeping.jpg is 0.8862
Accuracy for imgs/grey-american-shorthair.jpg is 0.9707
Accuracy for imgs/grey-cat-grass.jpg is 0.9712
Accuracy for imgs/kitten16.jpg is 0.8627
Accuracy for imgs/kitten9.jpg is 0.8258
Accuracy for imgs/stripey-kitty.jpg is 0.8089
Accuracy for imgs/the-black-white-kittens.jpg is 0.7214
Accuracy for imgs/tortoiseshell_shell_cat.jpg is 0.8017
Accuracy for imgs/young-calico-cat.jpg is 0.5981
The mean accuracy for all images is 0.8693

Parameters 2: **{feature=color & position, normalization=false, clustering=HAC, number of clusters=10, max pixels=2500}**

Accuracy for imgs/black-white-kittens2.jpg is 0.9035
Accuracy for imgs/black_kitten.jpg is 0.9620
Accuracy for imgs/black_kitten_star.jpg is 0.9810
Accuracy for imgs/cat-jumping-running-grass.jpg is 0.9736
Accuracy for imgs/cat_bed.jpg is 0.9654
Accuracy for imgs/cat_grumpy.jpg is 0.8784
Accuracy for imgs/cat_march.jpg is 0.9752
Accuracy for imgs/cat_mouse.jpg is 0.7603
Accuracy for imgs/cutest-cat-ever-snoopy-sleeping.jpg is 0.8954
Accuracy for imgs/grey-american-shorthair.jpg is 0.9681
Accuracy for imgs/grey-cat-grass.jpg is 0.9876
Accuracy for imgs/kitten16.jpg is 0.8998
Accuracy for imgs/kitten9.jpg is 0.8613
Accuracy for imgs/stripey-kitty.jpg is 0.8313
Accuracy for imgs/the-black-white-kittens.jpg is 0.7304
Accuracy for imgs/tortoiseshell_shell_cat.jpg is 0.8607
Accuracy for imgs/young-calico-cat.jpg is 0.7280
The mean accuracy for all images is 0.8919

3. Also feel free to point out or discuss any other interest in observations that you made.

If we have good contrast and clean delineation of foreground from background, the task of separating the two would not be too difficult. Also, there's proportional relationship between k and speed for k-means, while it is inverse for HAC.