Lukas Gemar, Ryan Meehan, and the Team

Dog Extraction Report:

The process of how we came upon our dog extraction algorithm is in the final report. As a quick overview, we transformed the color space from RGB to HSV and then did k-means clustering on the coordinates in the HSV color space. The result of this is group of clustered pixels of similar colors (we picked the number of clusters to be 4). We ranked the masks of color-clustered pixels in order to predict which one was most likely to represent the dog.

This pdf will serve as a guide to show some examples of how well the feature extraction works on various images. I will try to include three representative examples.

Dog 1:

Original Picture: Processed Binary Mask:

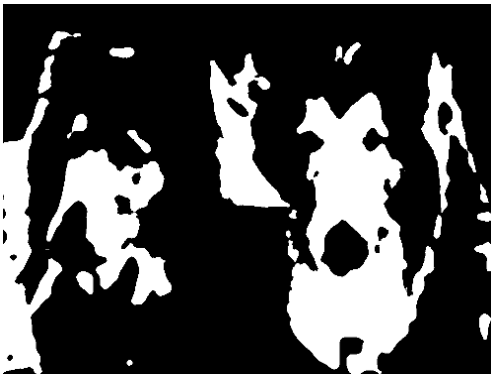

The first image is the original picture and the second is the mask of pixels that our algorithm thinks represent the dog. We then use this mask to extract features about the dog. This example worked particularly well because the original image has a couple solid blocks of color and the dog is well-contrasted against the background.

Dog 2:


The first image is the original picture and the second is the mask we've selected for pixels representing the dog. We only capture part of the dog in this binary mask, and cross-referencing with the original makes it obvious that we are only grabbing one of the dog's many colors. This points out a problem with our clustering algorithm: it imposes that the dog be one color, yet many of the dogs have several colors. Below are the remaining masks ranked in order of likelihood to contain the dog. The mask above is the top choice and these are the outputs of the remaining three clusters:

This is #2:

This is #3:



This is #4:

We can see from these that what we ranked as #3 also represents a huge proportion of the dogs. This is where our algorithm has trouble. Our code does not have a good way of deciding when masks could be combined to produce the whole dog. If we had more time we may implement some kind of algorithm that matches boundaries between different color clusters and combines them if they trace out each other to some extent.

Dog 3:

Original Picture: Processed Binary Mask:


This represents a case when our algorithm completely breaks down. There are too many different colors in the picture and there are many diverse objects in the background that throw off the clustering process. With more comparison across images of dogs, we may be able to recognize more efficiently which colors are most likely to represent the dog and only look for clusters of colors in those regions. This would avoid the error caused by the chair and other large objects in the background of this image.

In Conclusion:
We can say that our algorithm works quite well for simple pictures where the dog is large and in the center of the image, clearly offset from the background. The algorithm struggles for cases when the dog is multicolored because it groups the dogs different colors into different clusters. Also, the algorithm is not able to extract a dog from a widely diverse background.