EECS 504 Challenge Project Report
Hong Moon, Kevin Ong

Description of the method and Rationalization

Our initial step is to segment the image using foreground-background graph-cut to distinguish the background and the objects. The only information given about the datasets is that in one of the datasets (IMAGES) the background is white, and in the other dataset (CARDS) the background is green and there are some variations in intensities for white pixels inside the circle.

To use the white color as our background in the graph-cut, the first step of our algorithm is to distinguish which dataset the images are from (CARDS or IMAGES dataset) by inspecting the first pixel, i.e., the top left corner, of the image. If the first pixel is within some threshold (in this case 0.2) of the color white ([1 1 1]), then we determine that the image is from the IMAGES dataset, otherwise, it is from the CARDS dataset, because the first pixel of an image from the CARDS dataset is always green. If the image is from the CARDS dataset, the image is preprocessed such that the green background is replaced by white pixels and reduce the variation in white background by setting pixels that are within a threshold of 0.2 of the color white to pure white.

After preprocessing the images, we perform SLIC segmentation with 800 superpixels and use graph cut to extract the foreground. The reason for using 800 superpixel is that if the number of superpixels are too low (~200), some objects cannot be detected as foreground because the size of the object is too small (e.g. the dolphin in cards_002) and if the number of superpixels are too large (>1000), the image would be over-segmented and some objects would be detected as several objects (e.g. the clown in cards_003).

After applying the graph-cut, we get a binary matrix of the same size as the image where 1 indicates background and 0 indicates foreground.We then find the connected components, or islands, of foreground pixels on the image and find the k largest islands, where k = min(# of islands, 10). Furthermore, we created boxes around each of the k islands to determine the location and identify each object in the image.

To find features of each object, we first detect SURF keypoints on each of the RGB channels and extract the SURF features for each channel. The reason we use SURF is that SURF is both scale- and rotational- invariant. Then, we match the features of each object based on the sum of squared distances and generate a score based on SURF matching metrics and number of matches. After computing the scores, we set the scores that are not in the top 10 to -1 to penalize the lack of matches. Because SURF only performs feature detection and matching on one color channel, we also used color histograms to compensate. We first compute the color histogram of each object around the box and compute histogram dissimilarities using Euclidean distances. Similar to the score computed from SURF, we set the scores computed from color histogram dissimilarity that are not in the top 10 to -1 to penalize the vast difference in color scheme. Finally, we compute the final score of each match by combining the score from SURF and color histogram dissimilarities as follows:

$$Score = b_1 * exp(metric_{SURF}/c_1) + b_2 * exp(-dissimilarity_{color}/c_2)$$

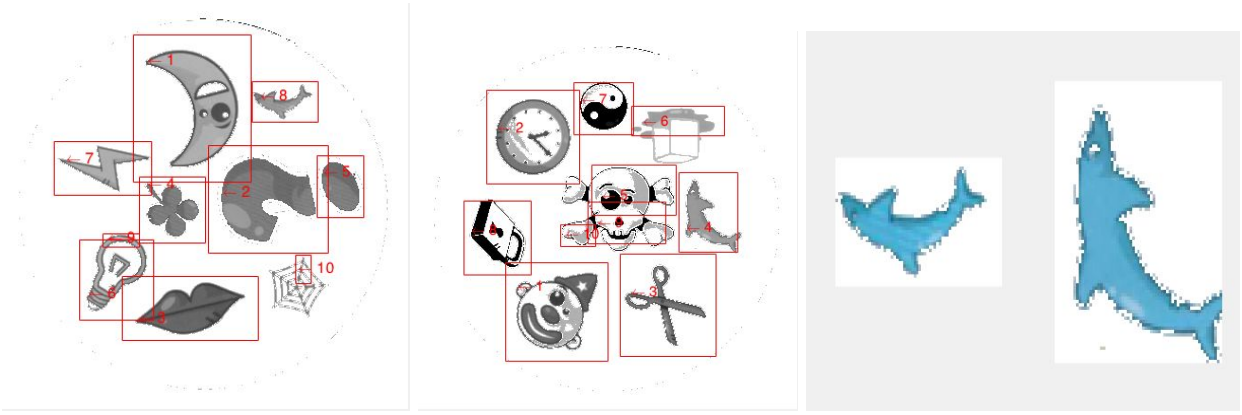
where $b_1 = 1.2,\ c_1 = 1,\ b_2 = 10,\ c_2 = 0.05$. The value of the scaling parameters are determined through numerous experiments.

The matching pair is determined by finding the maximum of the score.

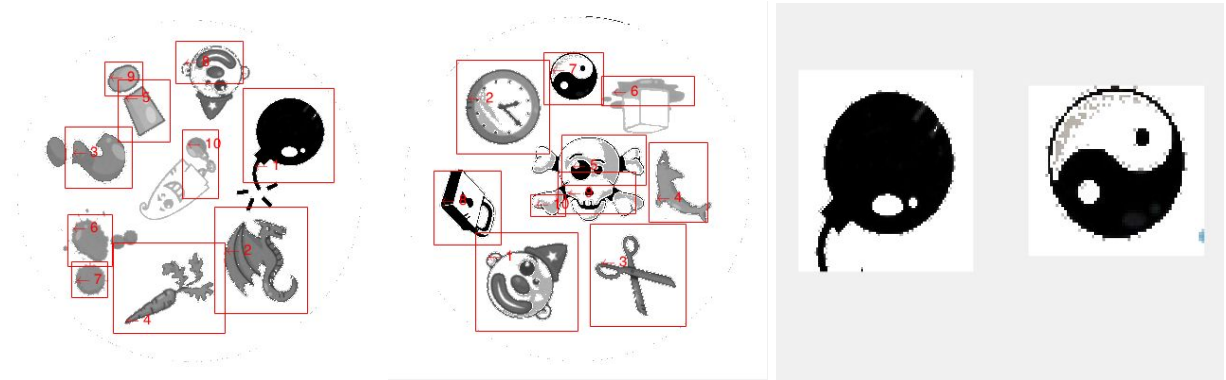
Results and Explanations

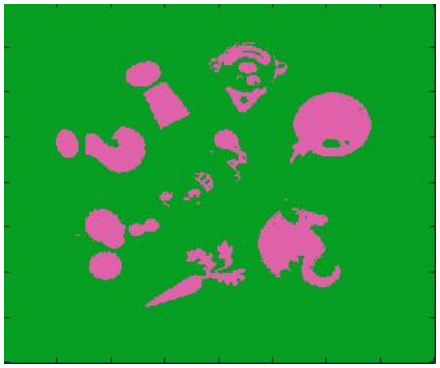
We are able to correctly match objects 19 out 20 pairs: 10/10 in IMAGES and 9/10 in CARDS.

In the following example (cards_002 vs. cards_003), there are very few SURF features extracted from the dolphin, thus the SURF score is set to -1. However, the color similarity between the two dolphins from the two images is very high and dominated the undesirable SURF scores; therefore we are still able to match the dolphins from the two images correctly.



In our next example (card_001 vs. card_003), we were not able to correctly match the clowns from the two images and instead matched the bomb with the yin yang symbol. This is because the clown from card_001 is not segmented correctly. After the preprocessing and SLIC segmentation, the face of the clown is separated into several islands and were not grouped into the same superpixel because of the disconnectedness among parts of the face of the clown and over-segmentation and thus only the lip and the nose was grouped into one object. Furthermore, the properties between the bomb and the yin yang symbol was overwhelmingly similar (shape and color), which is the reason they are classified as matching pairs.





Deviation in performance between the cards and the images

Graph cut segmentation performs significantly better in IMAGES than in CARDS because there is no illumination variation in IMAGES and the background color is the same. Meanwhile, in CARDS, the images are noisy and some objects are not detected as foreground and since they require preprocessing to remove the background, some useful information might be lost. Also in IMAGES, each object is larger so more features can be detected and is easier to find a match. For example, in card_002, the number of SURF features captured for the dolphin and the spider web are very small because of the small sizes of the objects. Thus, there are not enough SURF features available to perform matching well. This is why we incorporated color histogram dissimilarity to compute the final matching score to compensate this.

NOTE: To run the program, run main.m