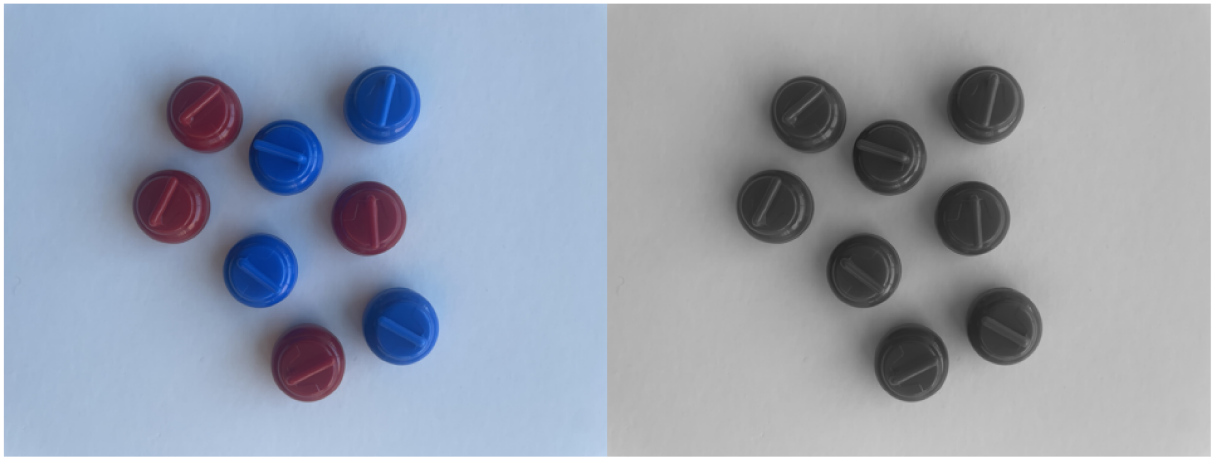
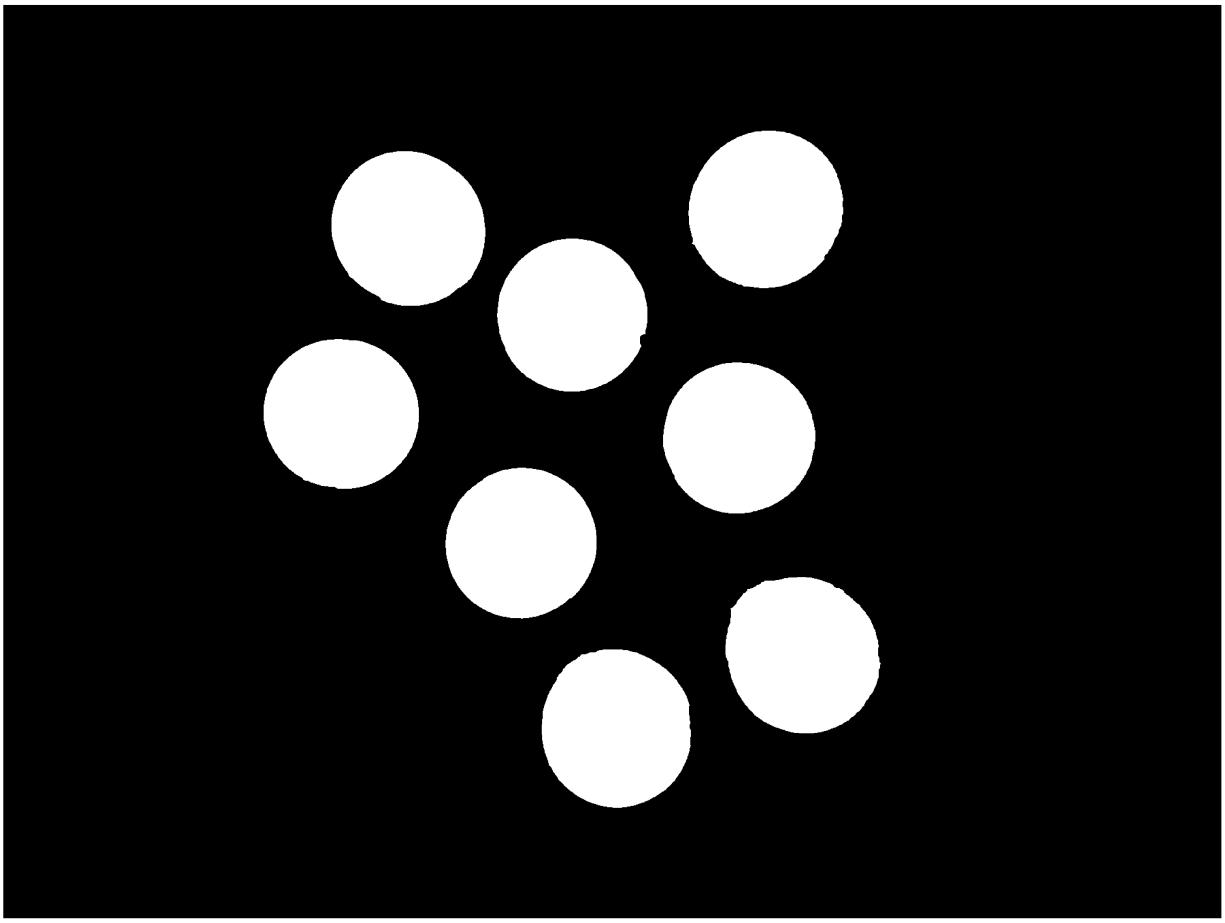
Project: Segment an Image

For this problem, you are given two lines of code which read in the image and convert it to grayscale. You will need to add your own code to provide a mask which accurately segments the curling stones from the background. Use variable name **`curlingMask`**.

```
curlingImg = imread("curlingImage.jpg");
curlingImgGray = im2gray(curlingImg);
montage({curlingImg,curlingImgGray})
```



```
curlingMask = segmentImage(curlingImg);
imshow(curlingMask)
```



```
function [BW,maskedImage] = segmentImage(X)
%segmentImage Segment image using auto-generated code from Image Segmenter app
%  [BW,MASKEDIMAGE] = segmentImage(X) segments image X using auto-generated
%  code from the Image Segmenter app. The final segmentation is returned in
%  BW, and a masked image is returned in MASKEDIMAGE.

% Auto-generated by imageSegmenter app on 20-Jan-2023
%----------------------------------------------------
X = im2gray(X);

% Threshold image - adaptive threshold
BW = imbinarize(im2gray(X), 'adaptive', 'Sensitivity', 0.800000, 'ForegroundPolarity', 'bright

% Invert mask
BW = imcomplement(BW);

% Fill holes
BW = imfill(BW, 'holes');
```

```
% Dilate mask with disk
radius = 18;
decomposition = 8;
se = strel('disk', radius, decomposition);
BW = imdilate(BW, se);

% Erode mask with disk
radius = 18;
decomposition = 8;
se = strel('disk', radius, decomposition);
BW = imerode(BW, se);

% Create masked image.
maskedImage = X;
maskedImage(~BW) = 0;
end
```