Experiment No. 01:
BASICS OF IMAGE PROCESSING WITH MATLAB

Instructor: Dr. Jafar Saniie
ECE 498-07

Acknowledgment: I acknowledge all of the work (including figures and codes) belongs to me and/or persons who are referenced.

Signature: Maggie Barclay & Theo Guidroz

Introduction

A. Purpose

The purpose of this lab is to introduce students to the following concepts:

- MATLAB and its image processing toolbox
- Image segmentation and edge detection
- Image morphological operations and histogram expansion
- Object recognition by extracting shapes
- Video processing using the same principle

B. Background

According to the official website mathworks.com, MATLAB® combines a desktop environment tuned for iterative analysis and design processes with a programming language that expresses matrix and array mathematics directly. It includes the Live Editor for creating scripts that combine code, output, and formatted text in an executable notebook. MATLAB includes toolboxes to its user which focus on reference-standard algorithms and workflow apps for image processing, analysis, visualization, and algorithm development. The user can perform image segmentation, image enhancement, noise reduction, geometric transformations, and image registration using deep learning and traditional image processing techniques in 2D and 3D.

Image segmentation is a commonly used technique in digital image processing and analysis to partition an image into multiple parts or regions, often based on the characteristics of the pixels in the image. There exists a variety of image segmentation techniques, including:

- Thresholding, where Otsu's method is used to create a binary image from a 2D image. Otsu's method returns a single intensity threshold that separate pixels into two classes, foreground and background. This threshold is determined by minimizing intra-class intensity variance, or equivalently, by maximizing inter-class variance. This method can be applied to color images as well, which requires a conversion to grayscale image first. Thresholding is the simplest method among the image segmentation.
- Clustering, where an image is segmented into K number of clusters using K-means clustering-based segmentation. Various results can be generated using different parameters.
- Region growing, where a number of initial seed points are given on the image, and neighboring pixels of these points are examined to iteratively determine whether these pixels should be added to the region.
- Deep learning based techniques, etc.

Edge detection is an image processing technique for finding the boundaries of objects within images. It works by detecting discontinuities in brightness. Edge detection can also be used for image segmentation. Common edge detection algorithms include Sobel, Canny, Prewitt, Roberts, and fuzzy logic methods.

The shape extraction is performed based on image segmentation. After provided with the segmented regions and/or their boundaries, the shapes properties of the regions (ratio of dimensions, roundness) will be determined, and these extracted shapes/features will be used to recognize the object by its shape.

Digital Video Processing In principle, all digital videos consist of several frames, each frame can be considered as one image. In practice, the videos will be encoded to reduce the size of the video, so decoding may need to be applied to the videos and extract the frames. Then, basic principles can be used on those frames to perform video processing

II. Lab Procedure and Equipment List

A. Equipment

Equipment

- 1 x PC installed with MATLAB

B. Procedure

PART A

1. Run the image segmentation script, given in Appendix A.
2. Check results from the image segmentation script.
3. Check the result with personal images.
4. Create and run the edge detection script.

PART B

1. Run the shape extraction script given in Appendix B with the test image.
2. Add noise to the test image.
3. Modify the shape extraction script to run with the image with added noise. Check the results.
4. Redo step 2 and 3, but with a variance of 0.01.

PART C

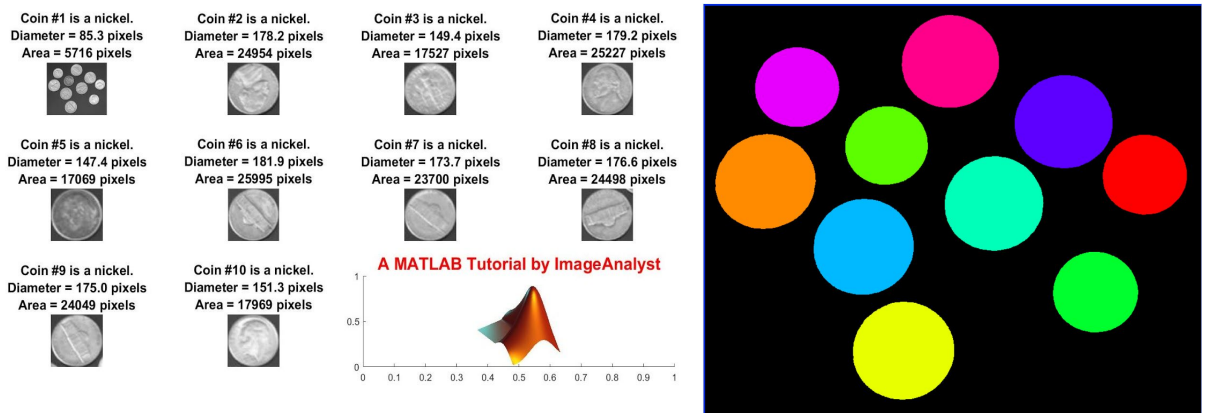
1. Try opening, closing, erosion, and dilation with MATLAB.
2. Histogram expansion with MATLAB. Use your own image to perform histogram expansion and compare the result with the example effect image in the background section.

PART D

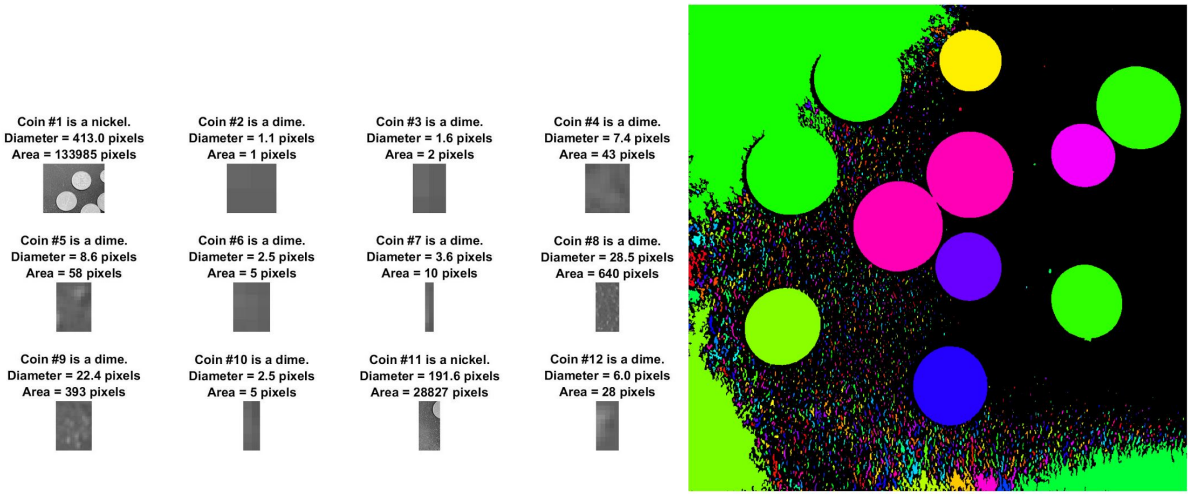
1. Run the video processing script in Appendix C with the test video.
2. Create a script to call upon the Classify function in part B to process these frames iteratively.
3. Use generate video from frame function in the video processing script to generate a video with labeled shapes.

III. Results and Analysis

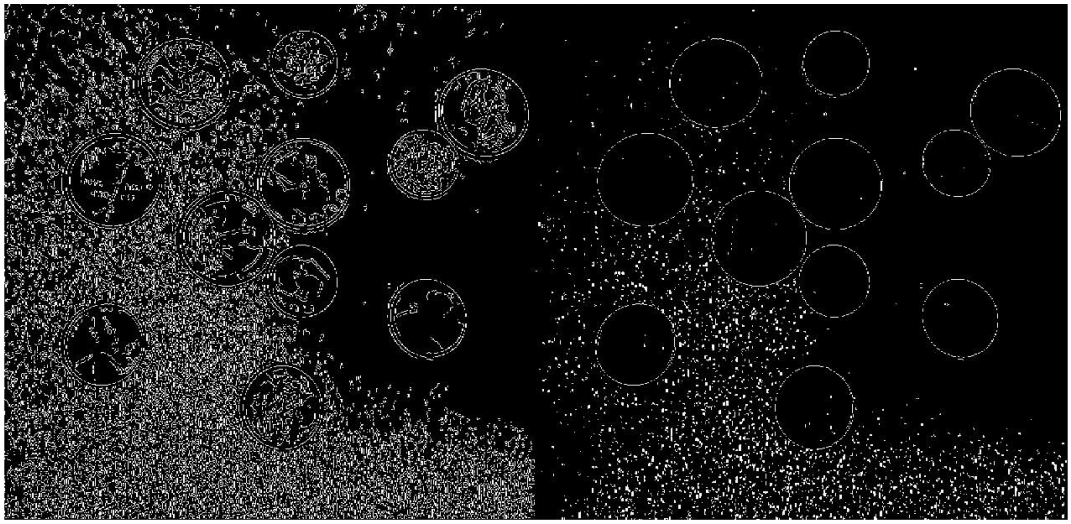
PART A



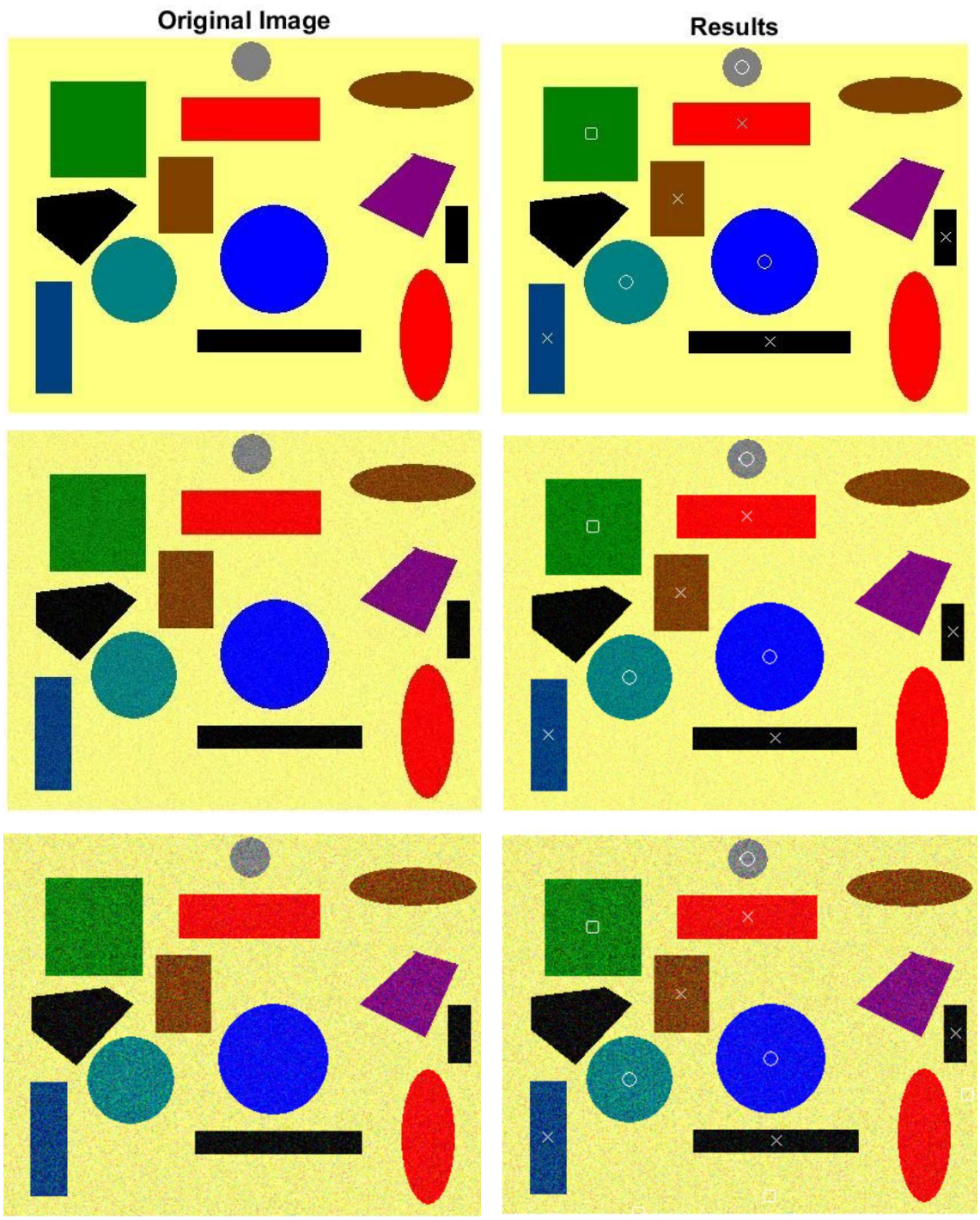
Results of edge detection using the given image



Results of edge detection using personal image



Canny algorithm vs Pewitt algorithm



Results obtained using 3 different degrees of noise

Part C ----------

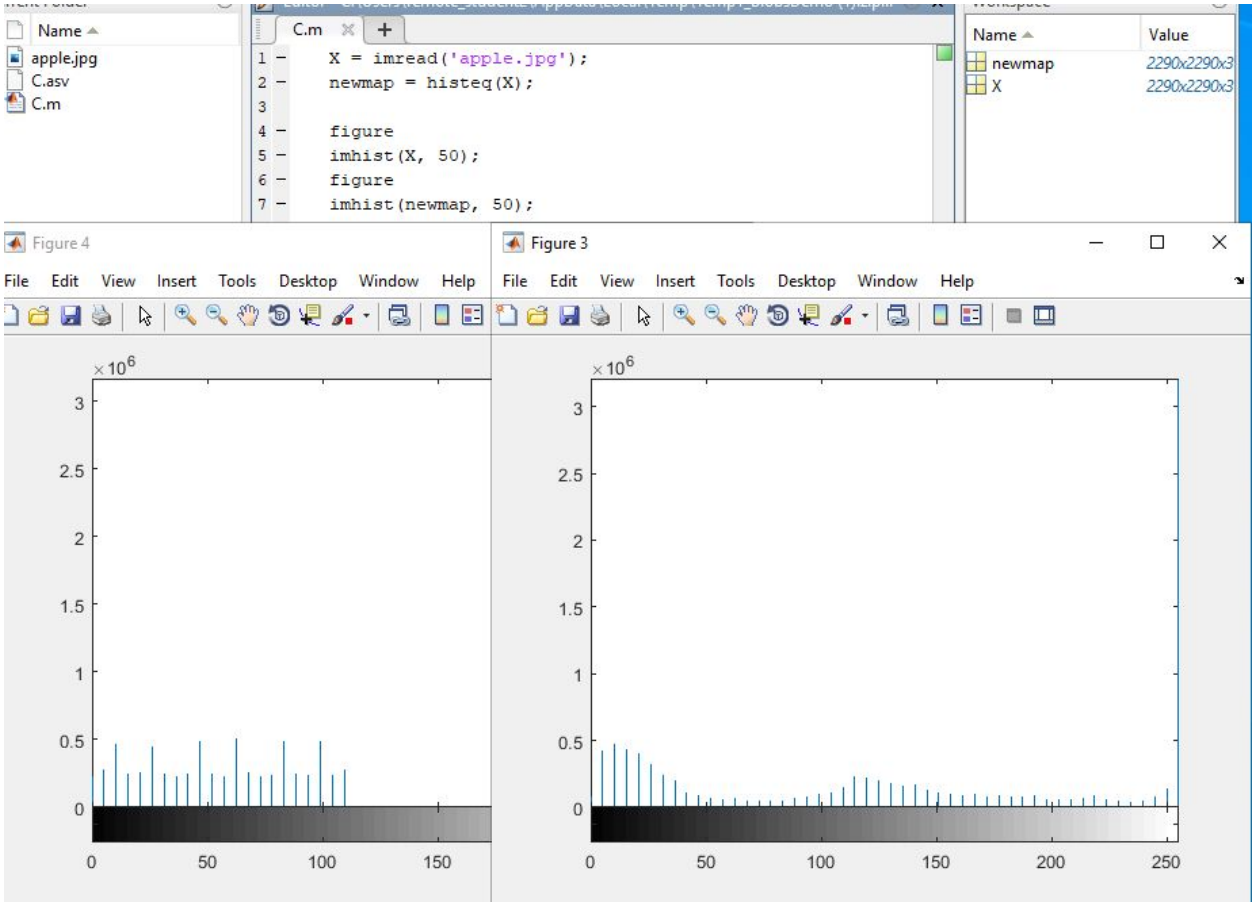
Apple.jpg

```
I = imread('apple.jpg');
imshow(I)
se = strel('disk',100);
I_opened = imopen(I,se);
figure, imshow(I_opened,[])
```

```
I = imread('apple.jpg');
se = strel('disk',100);
I_opened = imclose(I,se);
figure, imshow(I_opened,[])
```

```
I = imread('apple.jpg');
se = strel('disk',100);
I_opened = imerode(I,se);
figure, imshow(I_opened,[])
```

```
I = imread('apple.jpg');
se = strel('disk',100);
I_opened = imdilate(I,se);
figure, imshow(I_opened,[])
```


Name
apple.jpg
C.asv
C.m
C.m
1 - X = imread('apple.jpg');
2 - newmap = histeq(X);
3
4 - figure
5 - imhist(X, 50);
6 - figure
7 - imhist(newmap, 50);
Name
Value
newmap
2290x2290x3
X
2290x2290x3
Figure 4
File Edit View Insert Tools Desktop Window Help
×10^6
3
2.5
2
1.5
1
0.5
0
0 50 100 150
Figure 3
File Edit View Insert Tools Desktop Window Help
×10^6
3
2.5
2
1.5
1
0.5
0
0 50 100 150 200 250

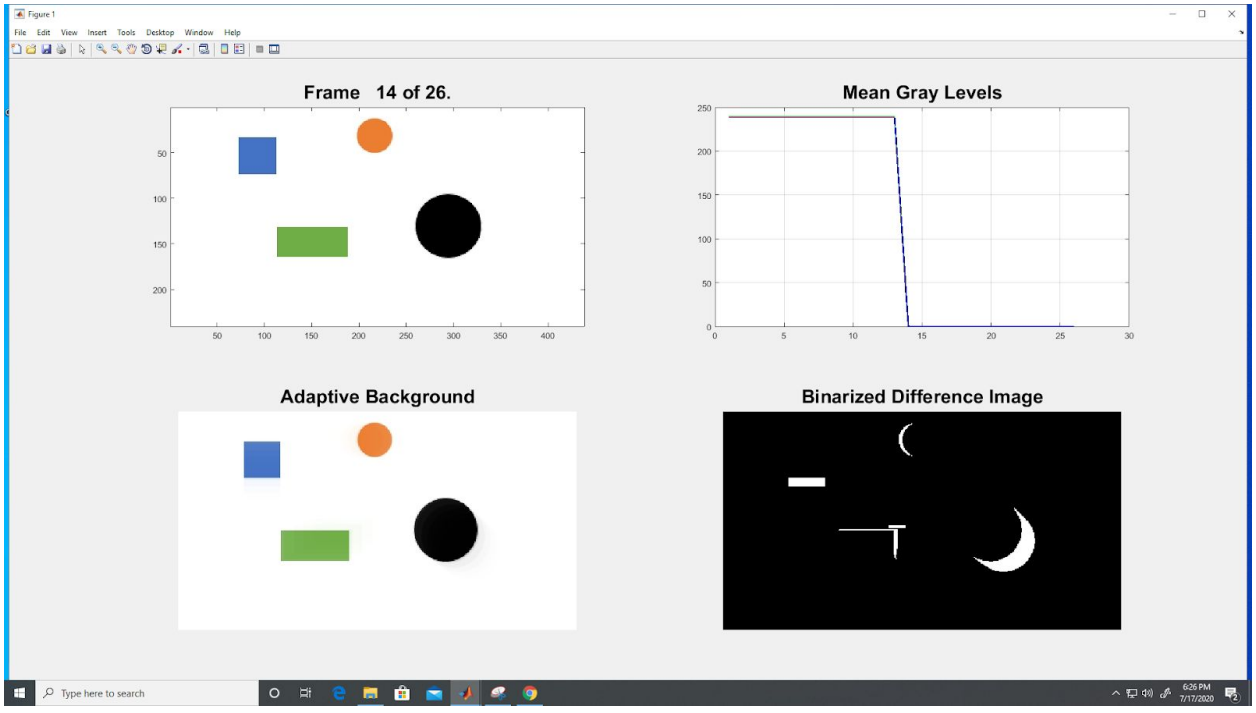
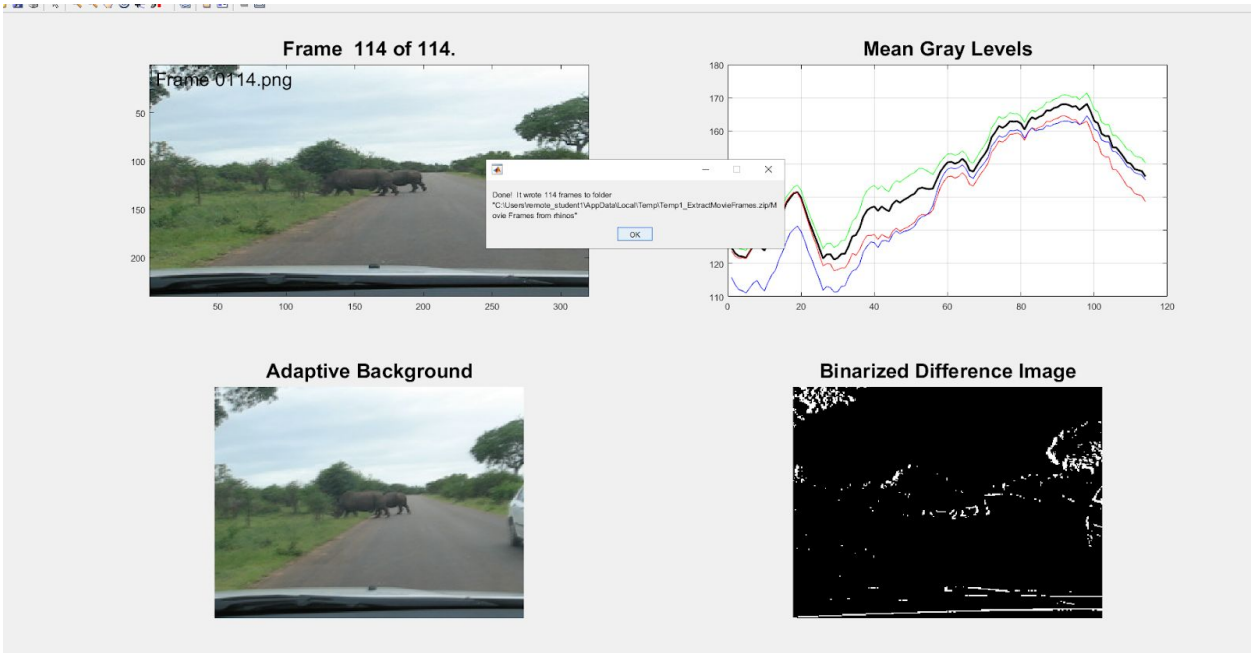


Figure 1
File Edit View Insert Tools Desktop Window Help
Frame 14 of 26.
Mean Gray Levels
Adaptive Background
Binarized Difference Image
Type here to search
6:26 PM
7/17/2020

Part D----------

1.



2.

```
function W = Classify(ImageFile)
% Step 1: Read image Read in
RGB = imread(ImageFile);
figure,
imshow(RGB),
title('Original Image');

% Step 2: Convert image from rgb to gray
GRAY = rgb2gray(RGB);
figure,
imshow(GRAY),
title('Gray Image');

% Step 3: Threshold the image Convert the image to black and white in order
% to prepare for boundary tracing using bwboundaries.
threshold = graythresh(GRAY);
BW = im2bw(GRAY, threshold);
figure,
imshow(BW),
```

```
title('Binary Image');

% Step 4: Invert the Binary Image
BW = ~ BW;
figure,
imshow(BW),
title('Inverted Binary Image');

% Step 5: Find the boundaries Concentrate only on the exterior boundaries.
% Option 'noholes' will accelerate the processing by preventing
% bwboundaries from searching for inner contours.
[B,L] = bwboundaries(BW, 'noholes');

% Step 6: Determine objects properties
STATS = regionprops(L, 'all'); % we need 'BoundingBox' and 'Extent'

% Step 7: Classify Shapes according to properties
% Square = 3 = (1 + 2) = (X=Y + Extent = 1)
% Rectangular = 2 = (0 + 2) = (only Extent = 1)
% Circle = 1 = (1 + 0) = (X=Y , Extent < 1)
% UNKNOWN = 0

figure,
imshow(RGB),
title('Results');
hold on
for i = 1 : length(STATS)
  W(i) = uint8(abs(STATS(i).BoundingBox(3)-STATS(i).BoundingBox(4)) < 0.1);
  W(i) = W(i) + 2 * uint8((STATS(i).Extent - 1) == 0 );
  centroid = STATS(i).Centroid;
  switch W(i)
     case 1
        plot(centroid(1),centroid(2),'wO');
     case 2
        plot(centroid(1),centroid(2),'wX');
     case 3
        plot(centroid(1),centroid(2),'wS');
  end
  return
end
try
          videoObject = VideoReader(movieFullFileName)
          % Determine how many frames there are.
```

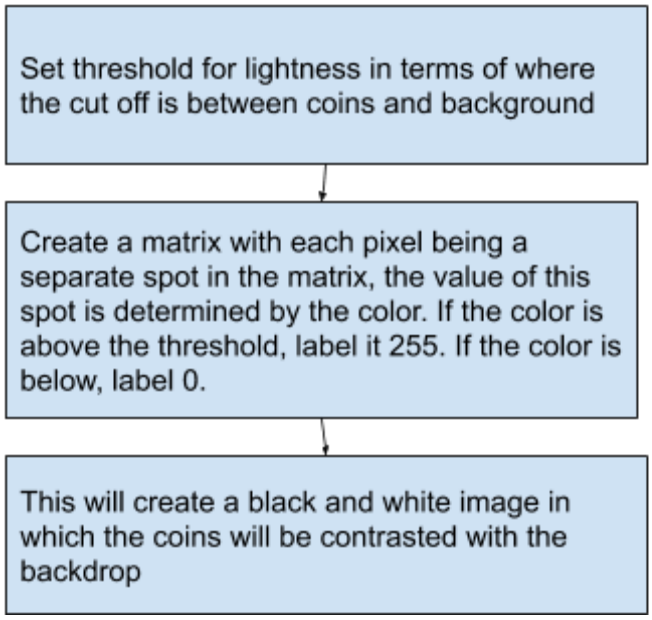
```
        numberOfFrames = videoObject.NumberOfFrames;
        vidHeight = videoObject.Height;
        vidWidth = videoObject.Width;
   for frame = 1 : numberOfFrames
                % Extract the frame from the movie structure.
                thisFrame = read(videoObject, frame)
      Classify(thisFrame)
   end
End
```

Questions----------

1. Comment on the advantages and disadvantages of thresholding and clustering as methods of image segmentation. (10 pts)

- With thresholding, it is less capable of detection of objects without a contrasting or solid background.
- Another disadvantage is the computational time and the memory of the new data created.
- Thresholding also tosses information in the image that cannot be used in the newly created image
- An advantage of thresholding is its simplicity given certain images. It is known as the simplest method of image segmentation.

2. Draw a flow chart and explain how the image segmentation script in part A works. (10 pts)



3. Why are there differences between the results of two edge detection algorithms? You may need to check materials regarding those two algorithms to explain. (10 pts)

Prewit is a simpler method, but it is more sensitive to noise in an image. Canny uses a smoothing effect to remove noise, and it is immune to a noisy environment. Canny takes longer to implement to reach a real time response.

https://www.ijser.org/researchpaper/Edge-Detection-by-Using-Canny-and-Prewitt.pdf

4. Explain the differences in shape extraction with different levels of noise. (10 pts)

The shape extraction still worked with the noise, using the levels of noise shown in the images. The shapes are appropriately labeled for what they are, but if the noise was worse then this may not be the case. When the variance was changed, the variance of the pixel values were changed. This led to more noise within the image. The process of making each pixel have noise (change in color or stay the same) is seemingly random, however the effects of noise can be manually reduced by turning down the variance.

5. In the video processing script, how can you reduce the number of generated frames while still keeping as much information as possible? (10 pts)

In order to reduce the number of generated frames, the camera can be set to a lower frame rate. This will save memory and allow the information to still be captured. Another idea

would be to detect changes within the frame, and only begin to take pictures while there's change. This would keep all of the information while reducing the generated frames.

IV. Conclusions

In this lab, students ran different image processing scripts to familiarize themselves with concepts such as image segmentation and shape extraction. The output of most scripts were as expected and therefore, the lab can be considered successful. The only experiment which had an unexpected result was in Part A when students used a personal image to detect the coins and this could be revisited.

References

[1] Image Segmentation Tutorial
https://www.mathworks.com/matlabcentral/fileexchange/25157-image-segmentation-tutorial
[2] Shape Recognition
https://www.mathworks.com/matlabcentral/fileexchange/15491-shape-recognition
[3] Video Processing Tutorial
https://www.mathworks.com/matlabcentral/fileexchange/47726-video-processing-tutorial
[4] Lab 1 manual