

## **Project 1: *Playing with MATLAB Image Toolboxes***

*Guruprasad Raghavan*

**Date:** 07 – Feb – 2016

### **Introduction:**

As the title claims, this project assignment is primarily designed to acquaint students with MATLAB's inbuilt toolboxes (Computer Vision and Image processing). Along with that, the use of open-source software ImageJ is introduced.

This document includes detailed instructions on how to execute the code as well as a primer to understand the results generated.

### **Question 1:**

#### **Part 1.1:**

- Code Execution Protocol:
  1. Open the Script named "Question1Part1.m" on MATLAB
  2. Place cursor on the First section of the script
  3. Navigate to the Editor tab, and click "Run Section"
- Results:

The first image is the visualization of the original image, and the histogram of the original image, depicts that all the intensities are clustered at a particular set of values. This highlights that, there is less contrast in the original image.

On bettering the contrast, the histogram of the modified image is more spread out, and the increased contrast can be viewed in the modified image (Figure 2).



Figure 1: Original Image

Image Histogram (Original Image)

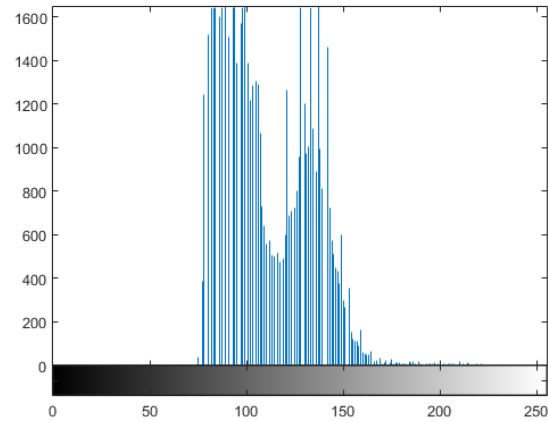
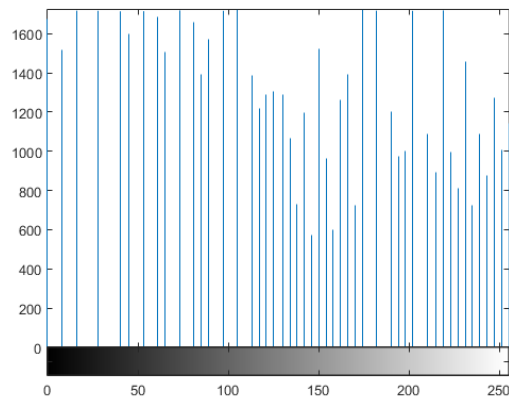


Figure 2: Bettering Contrast

Image Histogram (Modified Image)

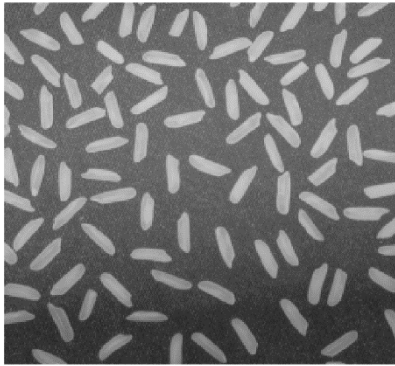


## Part 1.2:

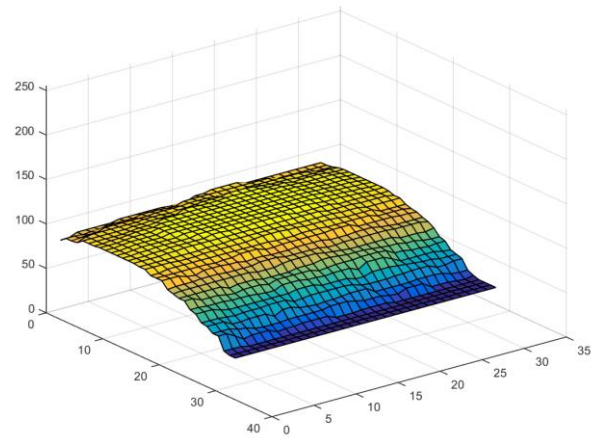
- Code Execution Protocol:
  1. Open the Script named "Question1Part1.m" on MATLAB
  2. Place cursor on the Second section of the script
  3. Navigate to the Editor tab, and click "Run Section"
- Results:

The first image is the visualization of the original image, and the surf plot of background shows that the center of the image is more illuminated (wrt intensity values). In order to remove this artifact, the image is subtracted from the background, and the modified image is also presented in the results that follow.

On leveraging the connected components functionality from MATLAB Image processing toolbox, we can calculate the area of the rice grains. A histogram of the areas is presented below for completeness.



Original 'Rice' image



Background surf plot

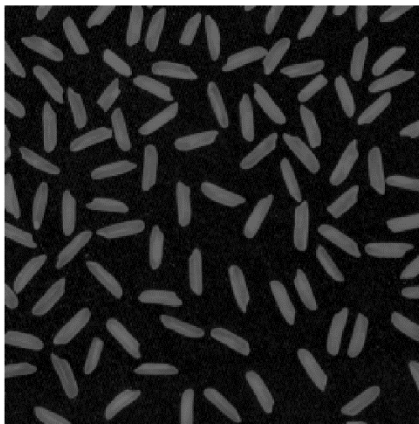
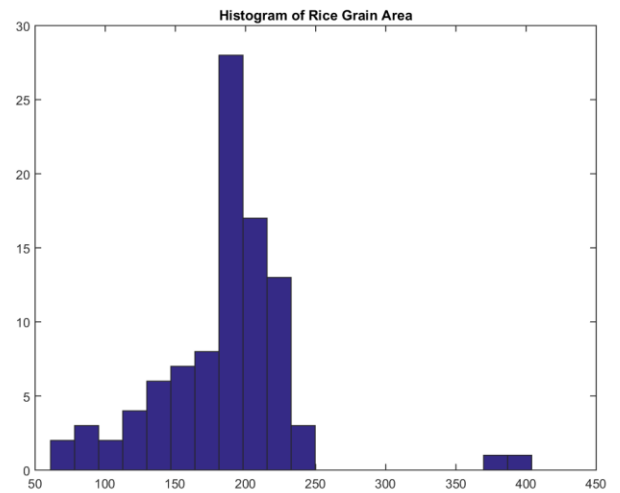


Image with background removed



Histogram of Rice grain Areas (after determining connected points)

## Question 1:

### Part 2.1:

- Code Execution Protocol:
  1. Open the Script named "Question1Part2.m" on MATLAB
  2. Place cursor on the First section of the script titled (`%% Detect SURF Interest Points in a Grayscale Image`)
  3. Navigate to the Editor tab, and click "Run Section"
- Results:

SURF (Speeded Up Robust Features) is used primarily for local feature detection, and has many applications in object recognition, identifying faces, to name a few. This image clearly shows the detection of multiple local features by the SURF function (leveraged from the CV Toolbox).



### Part 2.2:

- Code Execution Protocol:
  1. Open the Script named "Question1Part2.m" on MATLAB R2015b
  2. Place cursor on the Second section of the script titled (`%% Using LBP Features to Differentiate Images by Texture`)
  3. Navigate to the Editor tab, and click "Run Section"

### Part 2.3:

- Code Execution Protocol:
  1. Open the Script named "Question1Part2.m" on MATLAB R2015
  2. Place cursor on the Second section of the script titled `%% Extract and Plot HOG Features`)
  3. Navigate to the Editor tab, and click "Run Section"
- Results:

Using HoG features to detect objects is based on the principle that images can be described by their gradients in the intensity (a distribution) or their edge directions. Here, a histogram of oriented gradients (HoG) is developed.



## Part 2.4:

- Code Execution Protocol:

1. Open the Script named "Question1Part2.m" on MATLAB R2015
2. Place cursor on the Second section of the script titled (`%%Find Edges in an Image`)
3. Navigate to the Editor tab, and click "Run Section"

- Results:

The image depicts the original image, and the contour (edges) being detected by the edge-detection algorithm in the CV toolbox.



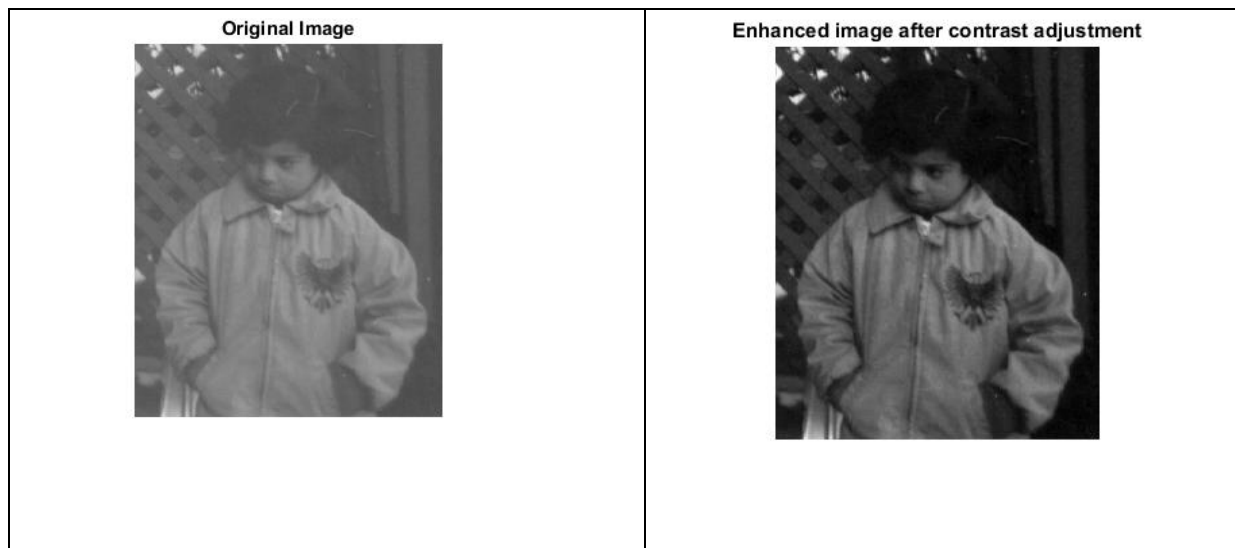
## Part 2.5:

- Code Execution Protocol:

1. Open the Script named "Question1Part2.m" on MATLAB R2015
2. Place cursor on the Second section of the script titled (`%%Enhance Image Quality using Contrast Adjuster`)
3. Navigate to the Editor tab, and click "Run Section"

- Results:

The Contrast Adjuster is used to better the contrast of the input image. The images below, depict the same.



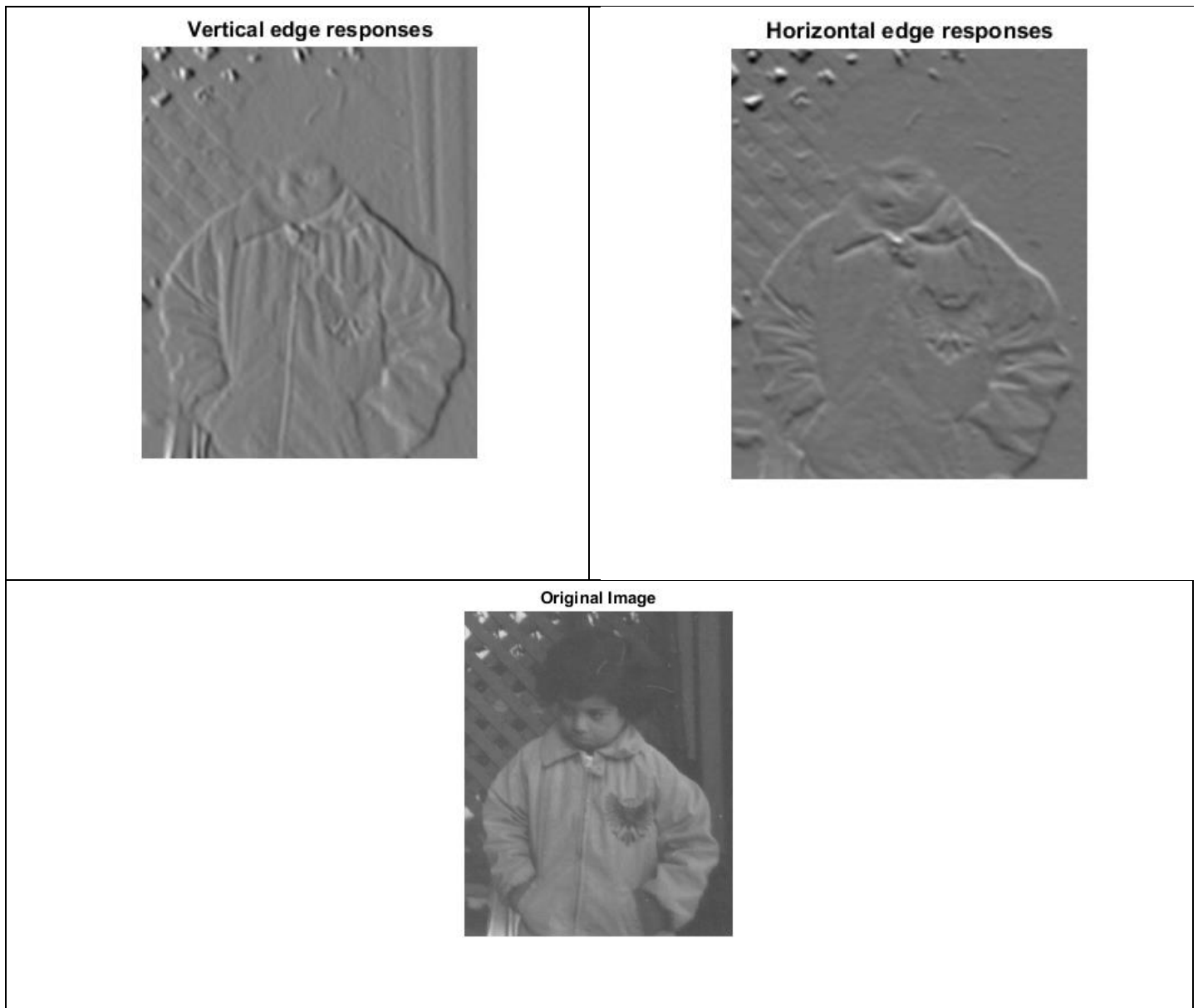
## Part 2.6:

- Code Execution Protocol:

1. Open the Script named "Question1Part2.m" on MATLAB R2015
2. Place cursor on the Second section of the script titled (`%%Find Vertical and horizontal Edges in image`)
3. Navigate to the Editor tab, and click "Run Section"

- Results:

On using a Haar-like wavelet filter, one can construct a horizontal and vertical filter to selectively obtain horizontal and vertical edges of an image. The following images are a depiction of the working algorithm.

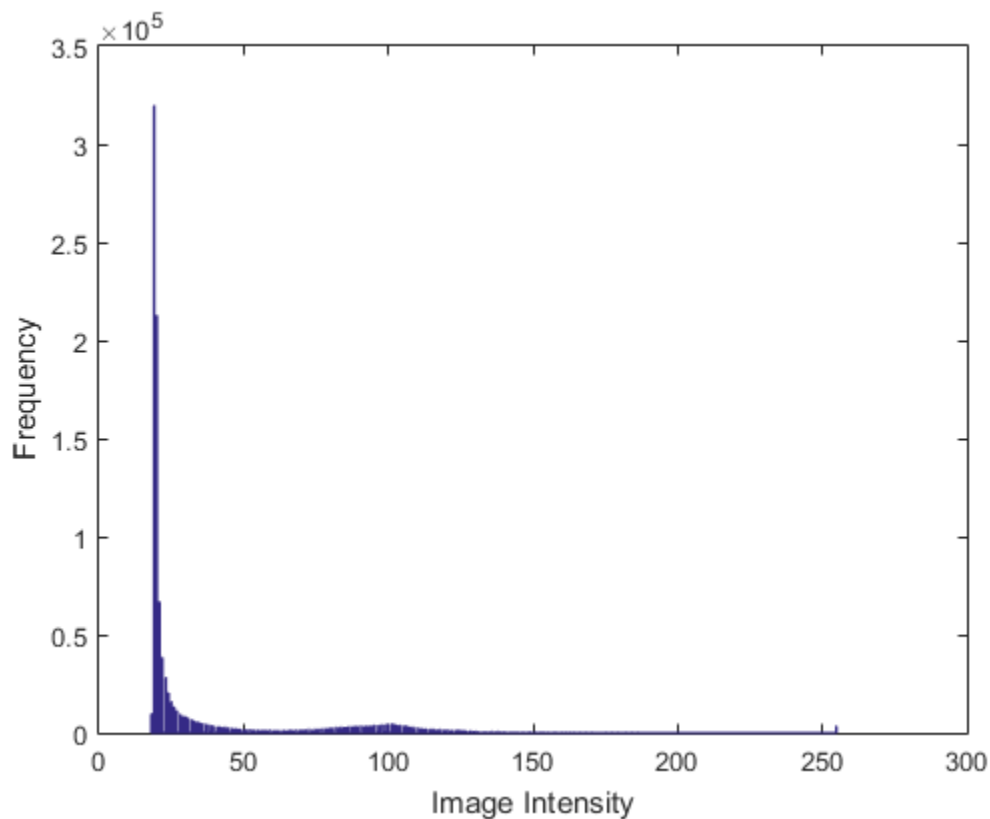


## Question 2:

### Part 2.1:

- Code Execution Protocol:
  1. Open the Script named "Question2.m" on MATLAB
  2. Place cursor on the First section of the script titled (`%% Part 2.1`)
  3. Navigate to the Editor tab, and click "Run Section"
- Results:

The Image histogram is obtained without using the `imhist` function, and is instead obtained by converting the image (from `uint` to `double`), and from a 2D Matrix to a single column of intensity values.



### Part 2.2:

- Code Execution Protocol:
  1. Open the Script named "Question2.m" on MATLAB
  2. Place cursor on the second section of the script titled (`%% Part 2.2`)
  3. Navigate to the Editor tab, and click "Run Section"
- Results:

A draggable rectangle can be interactively drawn to crop the image at any Region of interest. The cropped image is presented below:

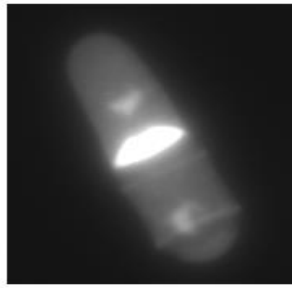


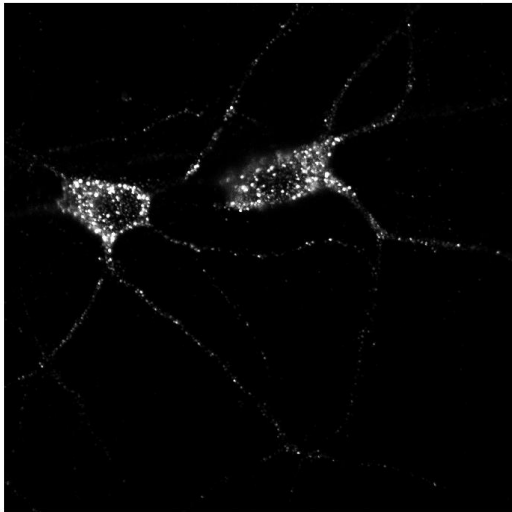
Figure 3: Cropped ROI

### Question 3:

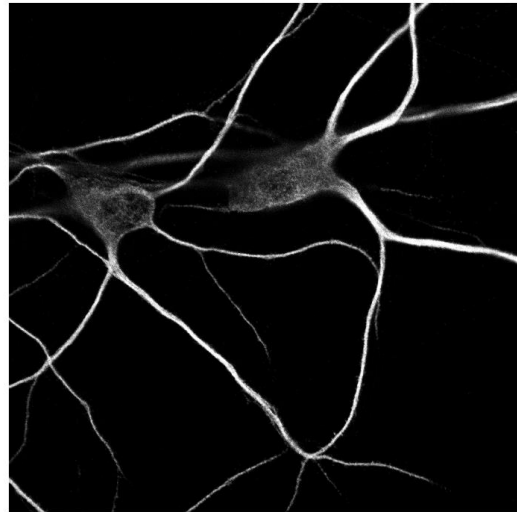
#### Part 3.1:

- Code Execution Protocol:
  1. Open the Script named “Question3.m” on MATLAB
  2. Place cursor on the First section of the script titled (`%% Part 3.1`)
  3. Navigate to the Editor tab, and click “Run Section”
- Results:

The multi-frame image is read into MATLAB separately, and each of its channels are displayed below. A histogram of both the channels are presented below (using the hist function, as built in Question 2).

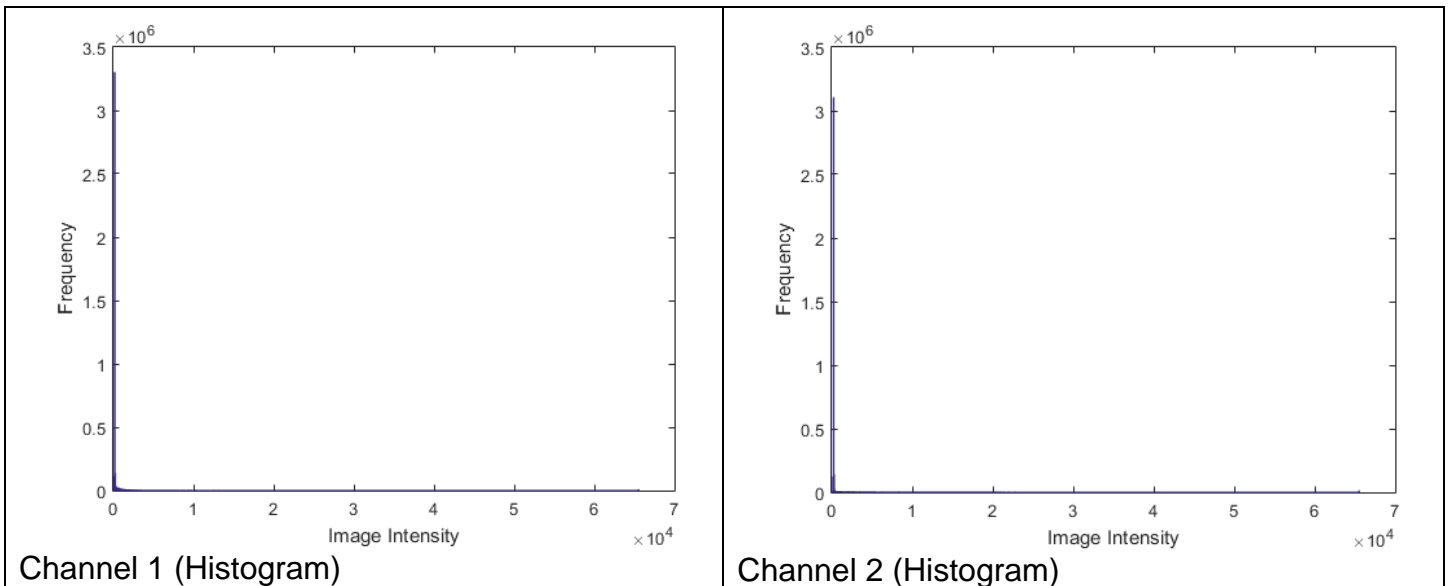


Channel 1 (Image)



Channel 2 (Image)

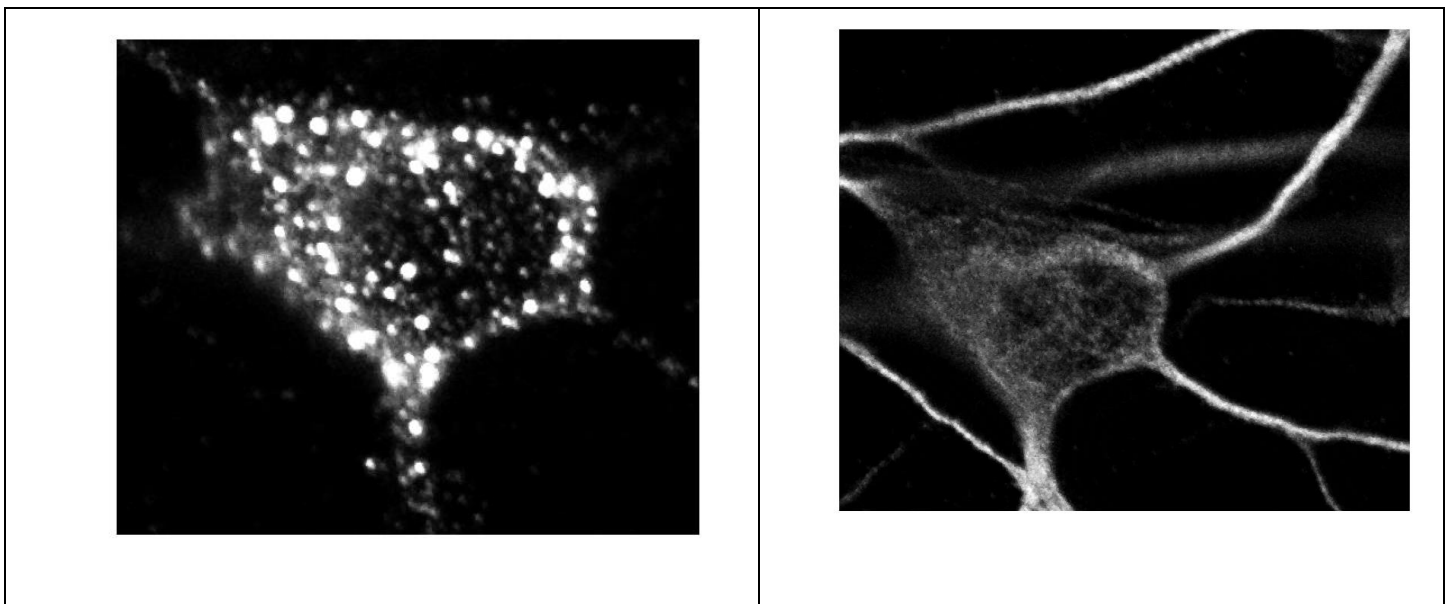




### Part 3.2:

- Code Execution Protocol:
  1. Open the Script named "Question3.m" on MATLAB
  2. Place cursor on the First section of the script titled (`%% Part 3.2`)
  3. Navigate to the Editor tab, and click "Run Section"
- Results:

Region of Interest in both the channels are cropped separately, and are presented below.



There are **2 frames** in this specific Tiff Image.

## Question 4:

### Part 4.0/4.1/4.2:

- Code Execution Protocol:
  1. Open ImageJ
  2. File → Import → ImageSequence
  3. File → Saveas → AVI
- Results:

Here, the videos are presented in the folder. The video corresponding to Part 4.1 is titled "Video4\_1" and the video corresponding to Part 4.2 is titled "ROIVideo4\_2"

### Summary:

The key concepts I've learnt are as follows:

1. Learnt how to use ImageJ, and explored some of its different features.
2. Having used MATLAB for other simulation tasks, interacting with images (ie reading/saving/modifying) was a learning experience.
3. Using interactive tools such as imrect, and imcrop to select a particular region of interest
4. Using toolbox functions, like SURF, HoG etc. I had a glimpse at the math behind these operations, but I hope to gain full understanding by the end of this course.

### References:

1. Tuytelaars, T., and K. Mikolajczyk. "Local Invariant Feature Detectors: A Survey." *Foundations and Trends in Computer Graphics and Vision*. Vol. 3, Issue 3, 2007, pp. 177–280.
2. Bay, H., A. Ess, T. Tuytelaars, and L. Van Gool. "SURF: Speeded Up Robust Features." *Computer Vision and Image Understanding (CVIU)*. Vol. 110, No. 3, 2008, pp. 346–359.