List of algorithms for hackathon April 27th 2016

1. Background subtraction

1. We will use the image Rat_Hippocampal_Neuron.png for testing.
2. Apply rank filter (http://rsb.info.nih.gov/ij/developer/api/ij/plugin/filter/RankFilters.html) to the image. The rank filter could be max, min, median, gray scale erosion, gray scale dilation etc. We could focus on only a few filters now and we can give the ability to add more at a later stage. This is the background image. The user must be allowed to choose between various filters.
3. Subtract the rank filtered image from the original. This is the background subtracted image. The algorithm code is available at http://rsb.info.nih.gov/ij/plugins/background.html.

2. Anisotropic Diffusion

A filtering technique aimed at reducing image noise without removing significant parts of the image content, typically edges, lines or other details that are important for the interpretation of the image. Most filtering techniques will blur edges along with noise.

Anisotropic diffusion resembles the process that creates a scale space, where an image generates a parameterized family of successively more and more blurred images based on a diffusion process. Each of the resulting images in this family are given as a convolution between the image and a 2D isotropic Gaussian filter, where the width of the filter increases with the parameter. This diffusion process is a linear and space-invariant transformation of the original image.

With a constant diffusion coefficient, the anisotropic diffusion equations reduce to the heat equation which is equivalent to Gaussian blurring. This is ideal for removing noise but also indiscriminately blurs edges too. When the diffusion coefficient is chosen as an edge seeking function, such as in Perona and Malik, the resulting equations encourage diffusion (hence smoothing) within regions and prohibit it across strong edges. Hence the edges can be preserved while removing noise from the image.

MATLAB code: http://www.mathworks.com/matlabcentral/fileexchange/14995-anisotropic-diffusion--perona---malik-/content/anisodiff_Perona-Malik/anisodiff2D.m

ImageJ code: http://rsb.info.nih.gov/ij/plugins/download/Anisotropic_Diffusion_2D.java

Possible Python code: https://mail.scipy.org/pipermail/scipy-user/2009-February/020125.html

3. Frangi Filter

The Frangi filter identifies vessel-like or tube-like structures. This is important in many parts of biology, used to identify gray-matter tracks, blood vessels, neuron shapes, and endoplasmic reticulum tracing, among other fields. The Matlab code link below has an example image that we could use for testing.

MATLAB code: http://uk.mathworks.com/matlabcentral/fileexchange/24409-hessian-based-frangi-vesselness-filter

ImageJ code:
https://github.com/fiji/Feature_Detection/blob/master/src/main/java/fiji/features/Frangi_.java and imagej.net/Frangi (note it may be incomplete or inaccurate)

4. Max entropy Segmentation

1. Code available in http://rsb.info.nih.gov/ij/plugins/download/AutoThresholder.java
2. We will use the image Rat_Hippocampal_Neuron.png for testing.

5. Moments Segmentation

1. Code available in http://rsb.info.nih.gov/ij/plugins/download/AutoThresholder.java
2. We will use the image Rat_Hippocampal_Neuron.png for testing.

6. Shanbhag Segmentation

1. Code available in http://rsb.info.nih.gov/ij/plugins/download/AutoThresholder.java
2. We will use the image Rat_Hippocampal_Neuron.png for testing.

Once you have completed the code, please check it in to https://github.com/zenr/ippy

I have created folder, “filters” and “segmentation”. So please place the code in the appropriate folder.