**Cell Detection Report**

**Aim**

* To determine which edge detector from the set of kernels we have; produces the most accurate result when convolved with an image, via ROC analysis.

**Method**

1. For each kernel available:
2. Convolve the kernel with the image.
3. Threshold image / Smooth image
4. Convert image to binary image
5. Produce ROC analysis value.
6. Compare ROC analysis results of all kernels.

Kernels used: Simple gradient, Roberts, Sobel, first Order Gaussian, Laplacian, Laplacian of Gaussian, Canny.

Smoothing methods: Box/Mean Blur, Gaussian Smoothing

Threshold value (for noise reduction): Varies? (See result of each kernel). Using a varied value may produce better results for different kernel.

**Results**

Results for each filter: (ROC Value, Smoothing method, Threshold value, Gradient function)

**Results for Image 9343 AM**

Sobel

Roberts

First-order Gaussian

Laplacian

Laplacian of Gaussian

Filters ordered by ROC value:

**Results for Image 43590 AM**

Sobel

Roberts

First-order Gaussian

Laplacian

Laplacian of Gaussian

Filters ordered by ROC value:

**Results for Image 10905 JL**

Binary Image produced with kernel:

Sobel

Roberts

First-order Gaussian: Seeing how intense the colour was on the image, and how easy it seems to identify the edges, I decided to use low sigma value (and a filter size sigma\*3)

Laplacian

Laplacian of Gaussian

Filters ordered by ROC value:

**Conclusion**

Graph with ROC plotted values from all 3 images.

Based on this we have deduced that

It may be that when producing gradients, had we used absolute values rather calculating the magnitude, we may be able to conclude other than what we have if the image quality produced was the same; due to the time complexity of the absolute algorithm.

**Glossary**

mb = mean blur

gs = gaussian smoothing

mg = magnitude used for gradient (vs absolute)

**Notes**

* Converting image to binary: using BW = im2bw (I, level) = where I is a **grayscaleImage**.
* Use **Zero-crossing** function for laplacian.
* **Imgradientxy** for gradient.
* Find Matlab **Canny** function.
* BW = im2bw(Image,graythresh(Image)).