31/5/2021
Variations of w (white) and b (black):

1) wa=0, wb=0.5, wc=1; ba=0, bb=0.5, bc=1
2) wa=0.5, wb=0.75, wc=1; ba=0, bb=0.5, bc=1
3) wa=0, wb=0.25, wc=1; ba=0, bb=0.5, bc=1
4) wa=0, wb=0.5, wc=1; ba=0.5, bb=0.75, bc=1
5) wa=0, wb=0.5, wc=1; ba=0, bb=0.25, bc=0.5

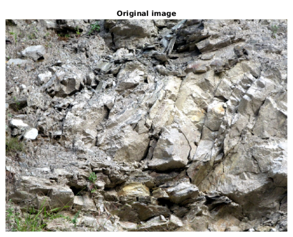
graph_1, fig_1 correspond to the 1)
Graph_2, fig_2 correspond to the 2).. etc

Minutes of the meet (Feb 5 2021)

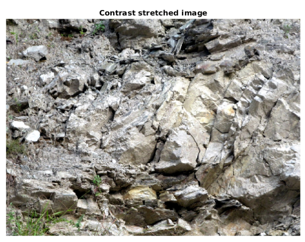
1. For the purpose of finding the sinusoidal curves in the images we observed from the images Cae and Cbe that bilateral filteraling after the dilated binary gradiant mask is most appropriate
2. But for the third image 26_u.jpg we observed that it's not that great as the curves are no longer clearly visible from a layman's view point. Refer figure Cce.
3. The second method that's edge detection using fuzzy logic seems better in a few cases as compared to the one mentioned before. Refer to figure Bce.

A) Image Segmentation:

a) 1_u.jpg

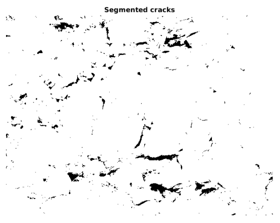


Aaa

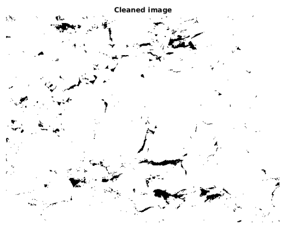


Aab

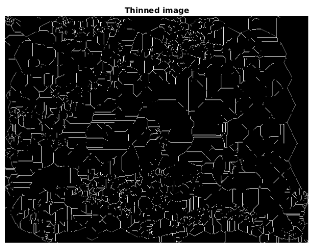
Aac



Aad

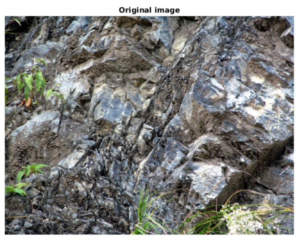


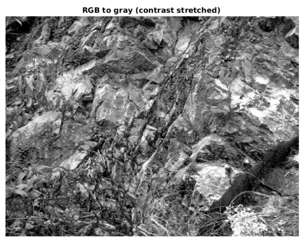
Aae



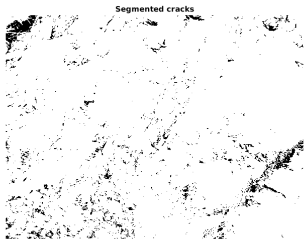
Aaf

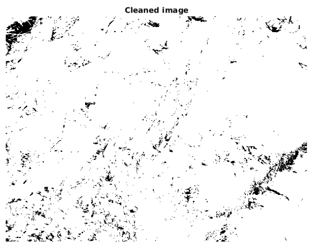
b) 25_u.jpg

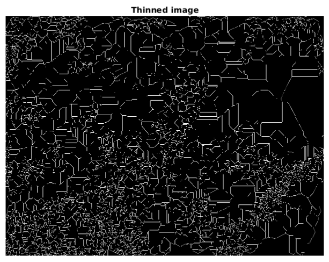




Aba **Abb** **Abc**

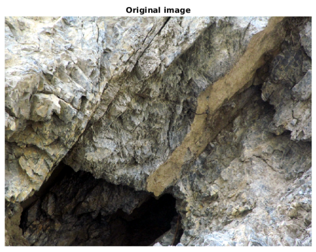


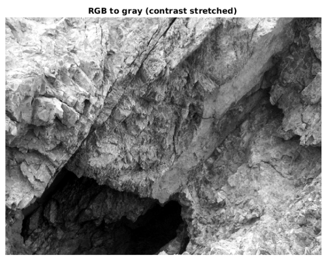




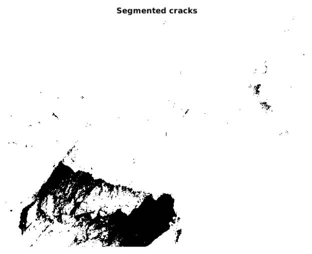
Abd **Abe** **Abf**

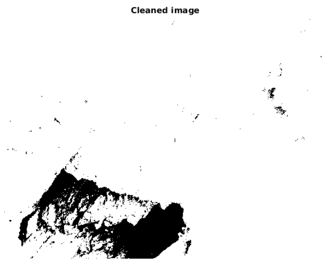
c) 26_u.jpg

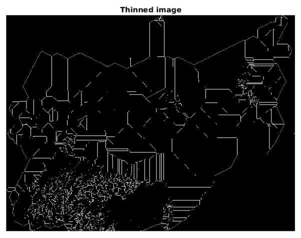




Aca **Acb** **Acc**







Acd **Ace** **Acf**

MethodA):

a) Original Uninterpreted Image

b) Contrast Stretched Image: Improve the contrast in an image by stretching the range of intensity values to span a desired range of values.

c) RGB to Gray image: Convert RGB Images to Gray scale images by taking the average value of the 3 colors at every pixel.

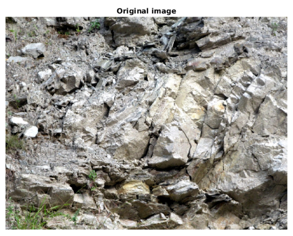
d) Segmented Cracks: Convert the grayscale image to binary image , by replacing all pixels in the input image with luminance greater than a threshold level with the value 1 (white) and replacing all other pixels with the value 0 (black)

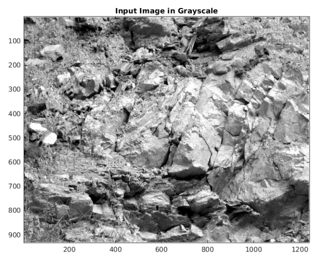
e) Cleaned image: Remove *isolated* pixels (individual 1's that are surrounded by 0's or individual 0’s that are surrounded by 1’s)

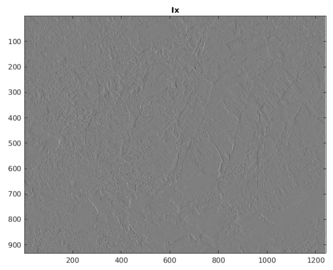
f) Thinned image: Thin objects to lines. It removes pixels so that an object without holes shrinks to a minimally connected stroke, and an object with holes shrinks to a connected ring halfway between each hole and the outer boundary.

B) Fuzzy Logic Image Processing

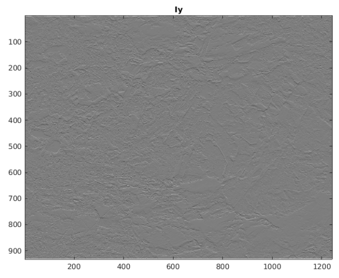
a) 1_u.jpg

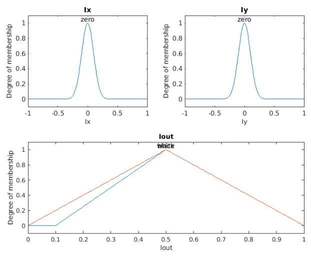


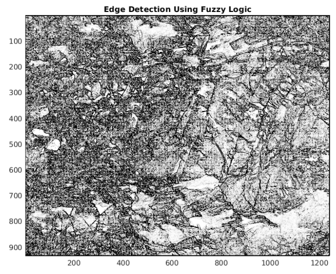




Baa **Bab** **Bac**

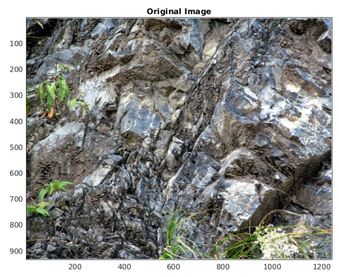


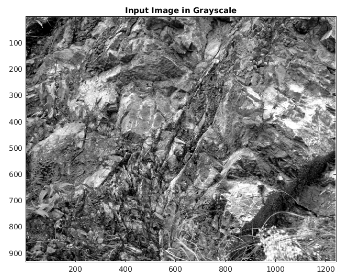


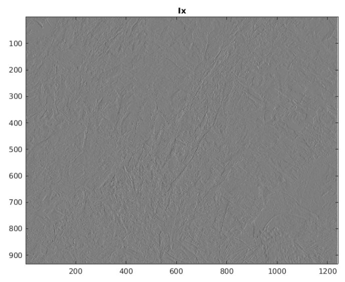


Bad **Bae** **Baf**

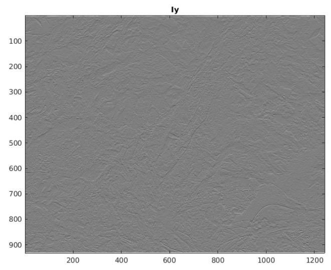
b) 25_u.jpg

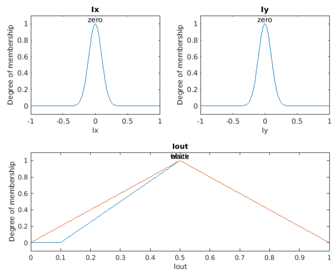


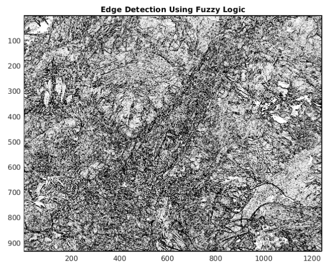




Bba **Bbb** **Bbc**

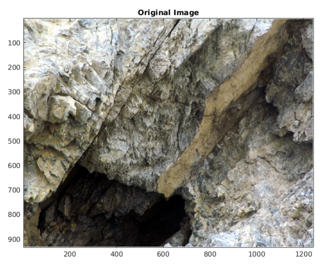


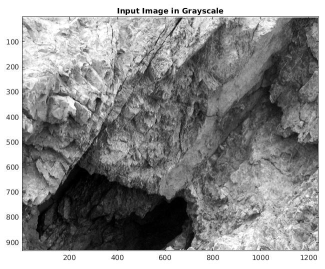


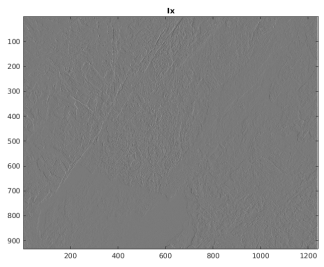


Bbd **Bbe** **Bbf**

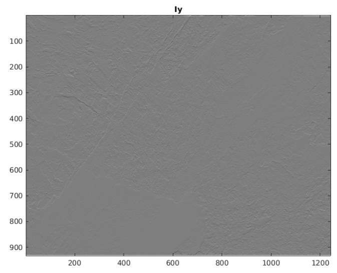
c) 26_u.jpg

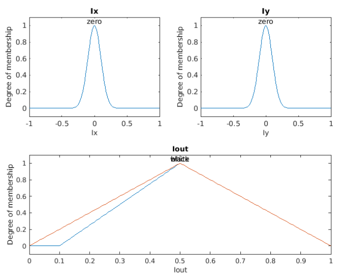


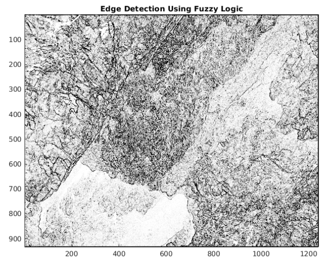




Bca **Bcb** **Bcc**







Bcd **Bcf** **Bcg**

Method B):

a) Original Uninterpreted Image

b) RGB to Gray Scale Image: Convert RGB Images to Gray scale images by taking the average value of the 3 colors at every pixel.

c) Ix: Gradient of the intensities of Image pixels in x direction

d) Iy: Gradient of the intensities of Image pixels in y direction

e) Degree of membership vs I: adds a membership function with the specified type and parameters. Specify a zero-mean Gaussian membership function for each input. If the gradient value for a pixel is 0, then it belongs to the zero membership function with a degree of 1.

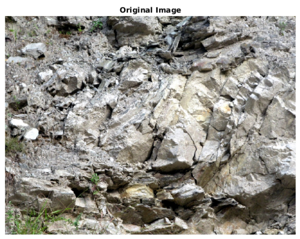
If sx and sy specify the standard deviation for the zero membership function for the Ix and Iy inputs, then to adjust the edge detector performance, you can change the values of sx and sy. Increasing the values makes the algorithm less sensitive to the edges in the image and decreases the intensity of the detected edges.

As you can with sx and sy, you can change the values of start, peak, and end of the triangles of the membership functions to adjust the edge detector performance. These parameters influence the intensity of the detected edges.

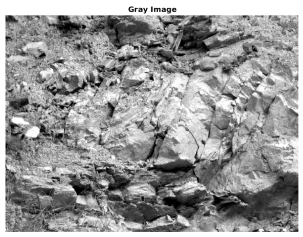
f) Edge Detections: Use Ix and Iy values to detect edges and mark them as white pixels in the final Image. Pixel is colored white if it belongs to a uniform region and black otherwise. A pixel is in a uniform region when the image gradient is zero in both directions. If either direction has a nonzero gradient, then the pixel is on an edge.

C) Bilateral Filtering

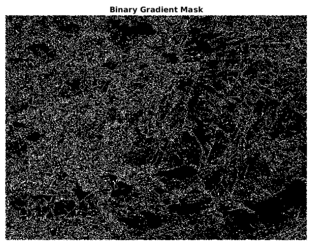
a) 1_u.jpg



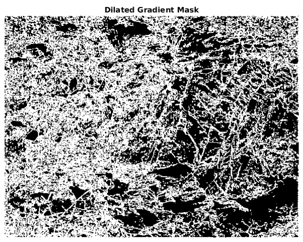
Caa



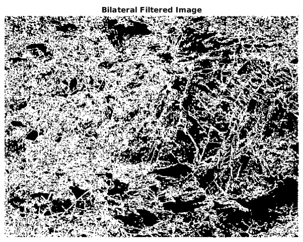
Cab



Cac

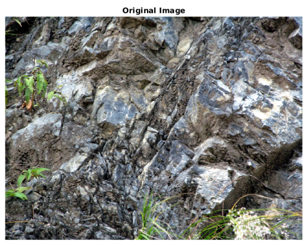


Cad

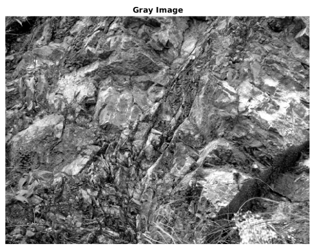


Cae

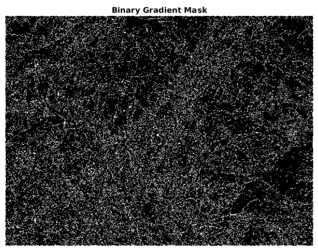
b) 25_u.jpg



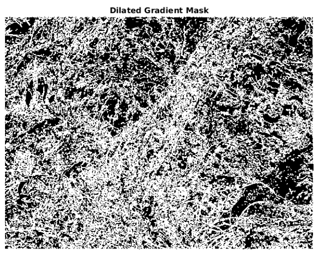
Cba



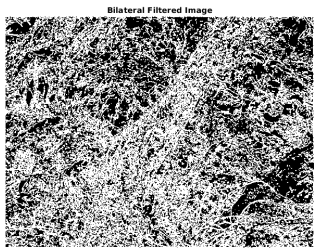
Cbb



Cbc

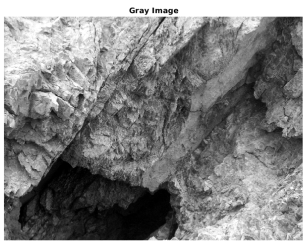


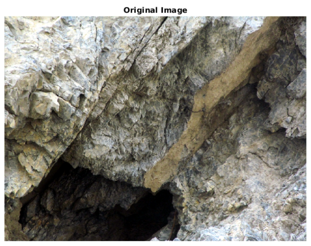
Cbd

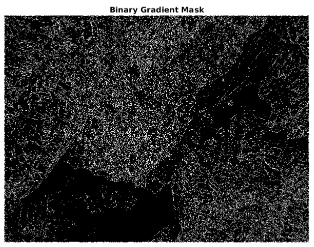


Cbe

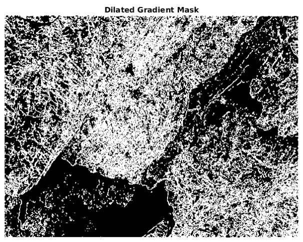
c) 26_u.jpg

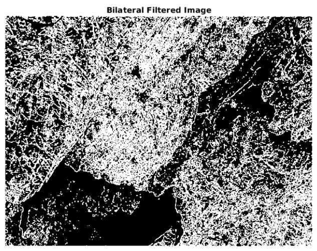






Cca **Ccb** **Ccc**





Ccd **Cce**

Method C):
a) Original Uninterpreted Image

b) RGB to Gray Image: Convert RGB Images to Gray scale images by taking the average value of the 3 colors at every pixel.

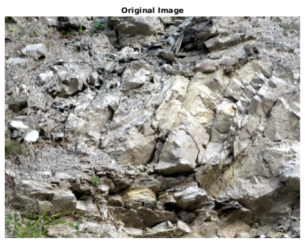
c) Binary Gradient Mask: Convert the grayscale image to binary image , by replacing all pixels in the input image with luminance greater than a threshold level with the value 1 (white) and replacing all other pixels with the value 0 (black)

d) Dilated Gradient Mask: Dilate the binary image, i.e add pixels to the boundaries of objects in an image

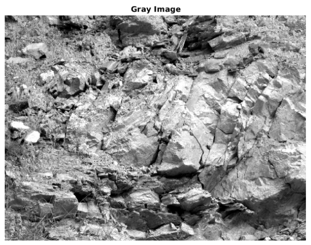
e) Bilateral Filtered Image: This is an edge preserving smoothing method where we make a mask with weights for surrounding pixels and convolve it with the original image. The smoothed intensity at every pixel location x1, would be the weighted average of the surrounding pixels. The weight for a pixel location x2, for the intensity to be calculated at x1, is based on
i) spatial distance between x1 and x2 (larger the distance, less the weight)
ii) dissimilarity between the intensity values at x1 and x2 (larger the dissimilarity, less the weight)

D) Comparison between various Edge/ Fracture detection filter techniques

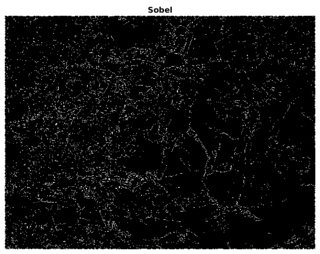
a) 1_u.jpg



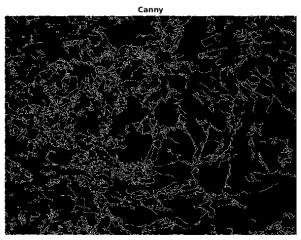
Daa



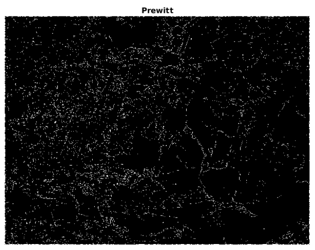
Dab



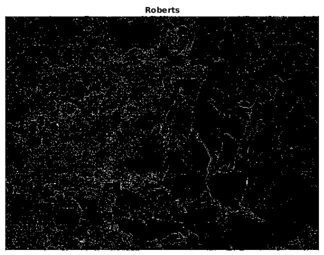
Dac



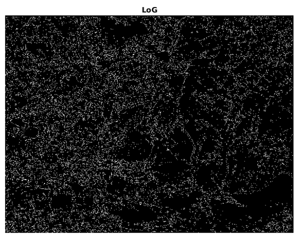
Dad



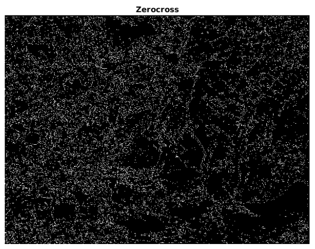
Dae



Daf

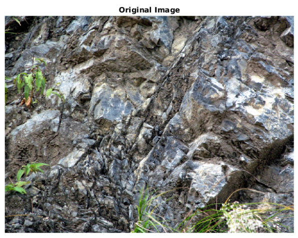


Dag

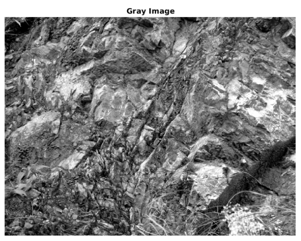


Dah

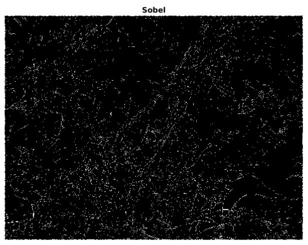
b) 25_u.jpg



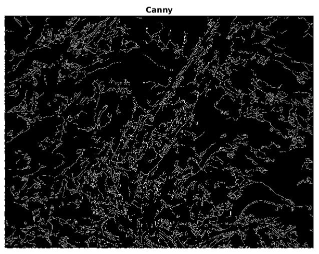
Dba

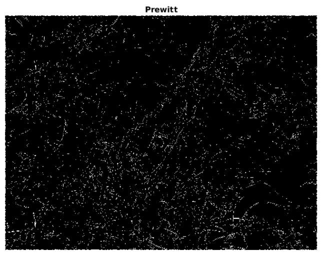


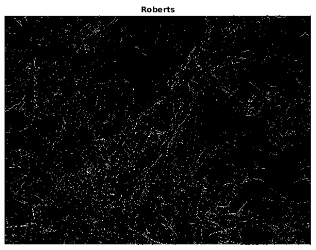
Dbb



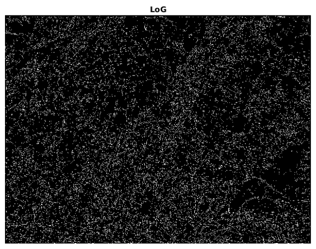
Dbc

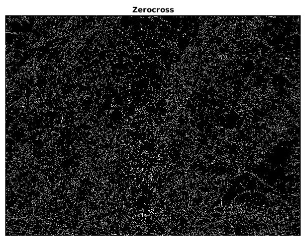






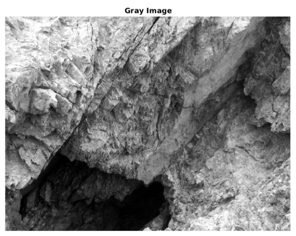
Dbd **Dbe** **Dbf**

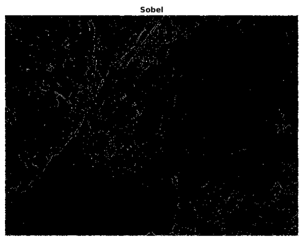




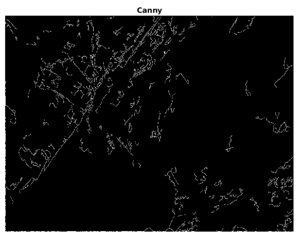
Dbg **Dbh**

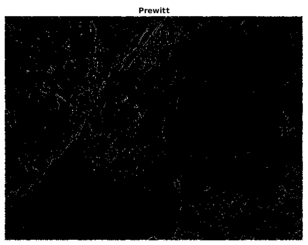
c) 26_u.jpg

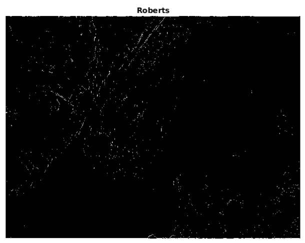




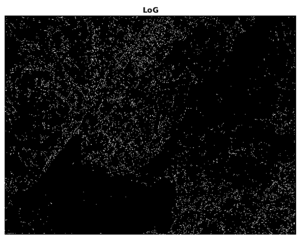
Dca **Dcb** **Dcc**



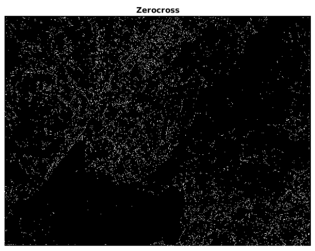




Dcd **Dce** **Dcf**



Dcg



Dch

Different filters D):
a) Original Uninterpreted Image
b) RGB to Gray Image: Convert RGB Images to Gray scale images by taking the average value of the 3 colors at every pixel.

These are some algorithms for edge detection used commonly in Image processing:

c) Sobel filter: uses matrix math to calculate areas of different intensities of an image

d) Canny filter: uses a multi-stage *algorithm* to detect a wide range of edges in images

e) Prewitt filter: uses a derivative mask and can detect only horizontal and vertical edges

f) Roberts Filter: performs a simple, quick to compute, 2-D spatial gradient measurement on an image. It thus highlights regions of high spatial frequency which often correspond to edges. The matrices used are Gx=[[1 0] [0 -1]] and Gy=[[0 1] [-1 0]]

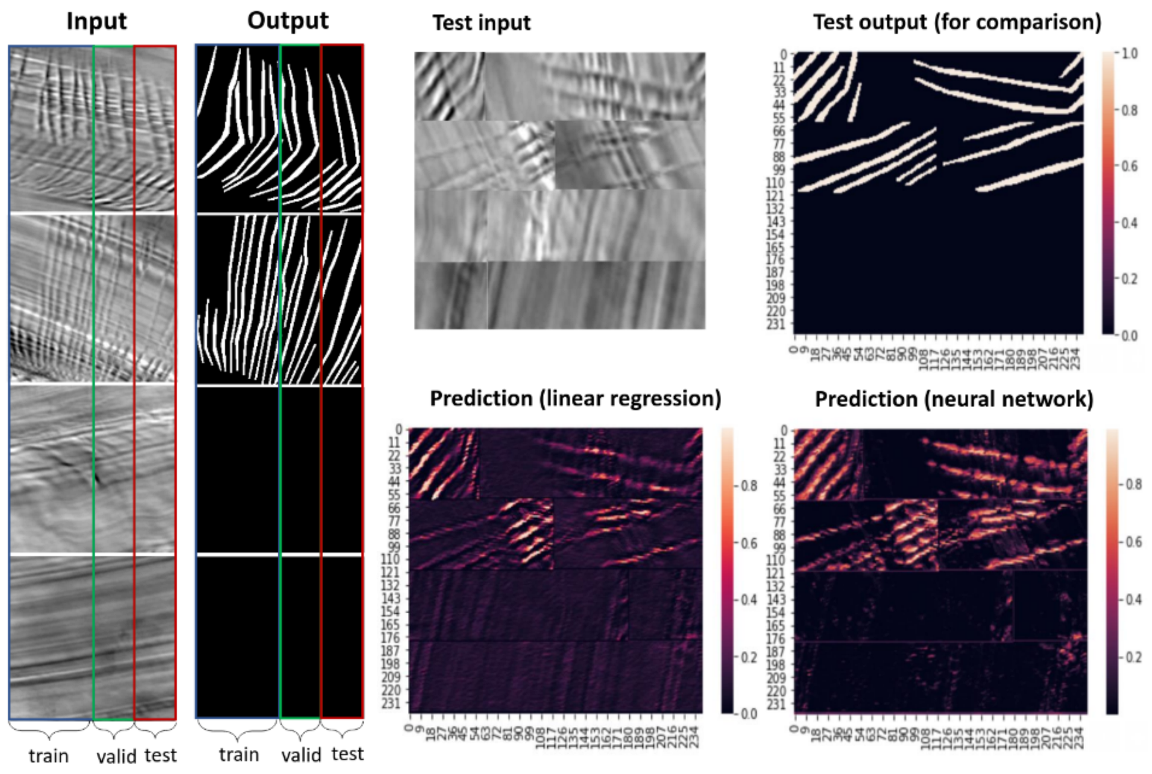
g) LoG filter: Finds edges by looking for zero-crossings after filtering I with a Laplacian of Gaussian (LoG) filter.

h) Zero Cross filter: Finds edges by looking for zero-crossings after filtering I with a filter that you specify, unlike fixed LoG filter

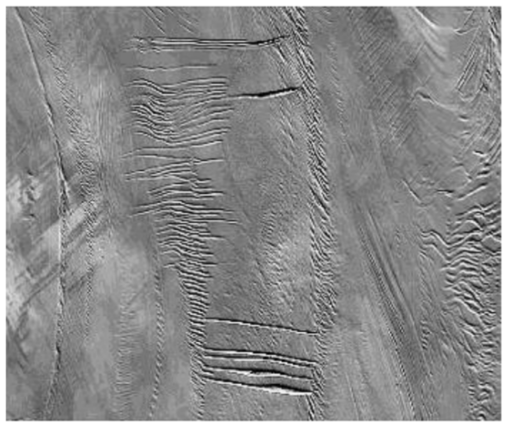
----

NOT RELEVANT FOR NOW

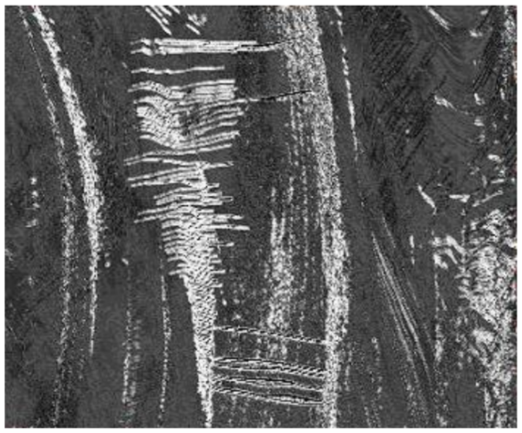
Sample framework:



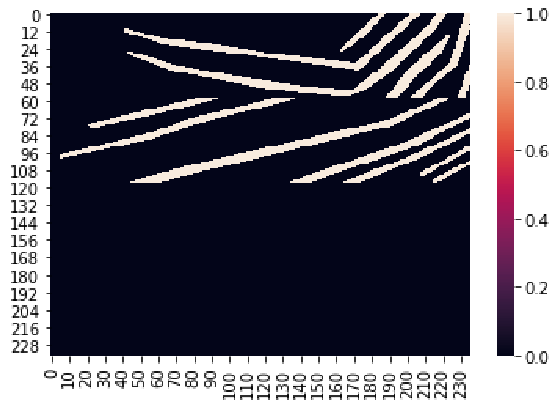
Test input:



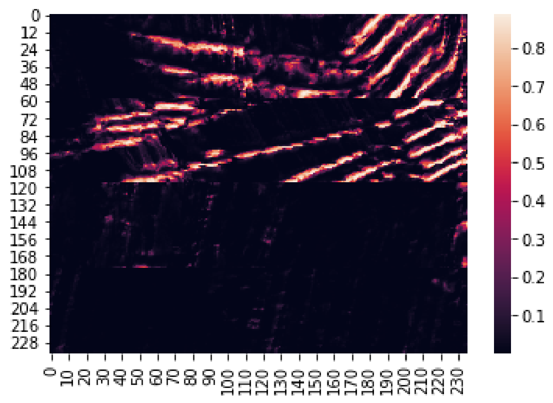
Prediction after Neural Networks:



Test y (original)



Predicted y



Ipynb notebooks:

1. Fractures_sample_test
2. Practice (takes sir's given images and tries to identify and reconstruct the fractures)
   Todo: needs cropping and resizing of the given images

Reference:
Google earth images: access denied
2 Narayan bghs: we transfer files: transfer expired and is not available anymore

Paper: https://www.nap.edu/read/2309/chapter/4#84
https://github.com/HoustonJ2013/Capstone_DL_Object_detection

Di2018 Developing a seismic pattern interpretation network (SpiNet) for automated seismic interpretation

Literature survey: https://github.com/f0nzie/deep-seismic-fav-docs
Important: https://github.com/chingyaolai/Fracture-detection/blob/master/Fracture_detection.pdf
https://github.com/chingyaolai/Fracture-detection/blob/master/Fractures%20Investigation_apply%20model_finepattern_LargeInputShape.ipynb

Quantification of Fracture Patterns MATLAB scripts to quantify patterns of fractures in rocks and other materials: https://github.com/DaveHealy-Aberdeen/FracPaQ

https://github.com/TerminusEst/geo_frac_analysis