

Scalable Bank of Filters for High Throughput Image Analysis

Shilpika¹, Nicola Ferrier², Venkatram Vishwanath¹
shilpika@anl.gov, nferrier@anl.gov, venkat@anl.gov

¹ Leadership Computing Facility, Argonne National Laboratory

² Mathematics and Computer Science, Argonne National Laboratory



T2

T4

T3

T5

T7

T6

T8

T10

T9

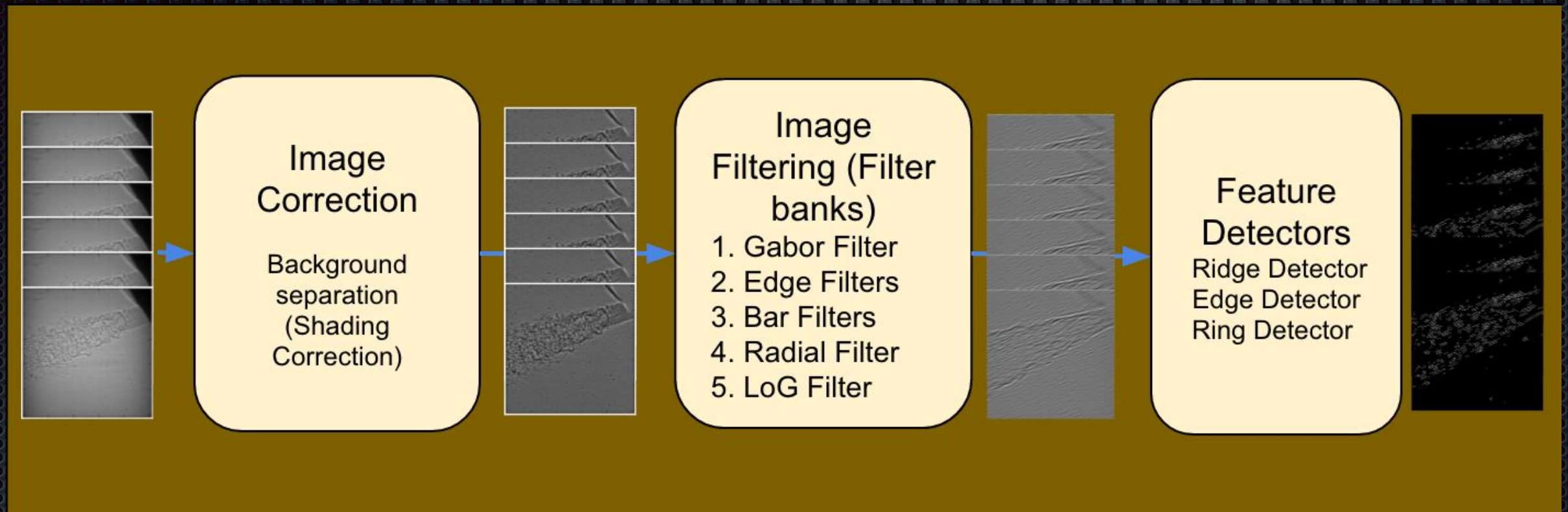
T11

Nozzle Pressure

Fuel Composition

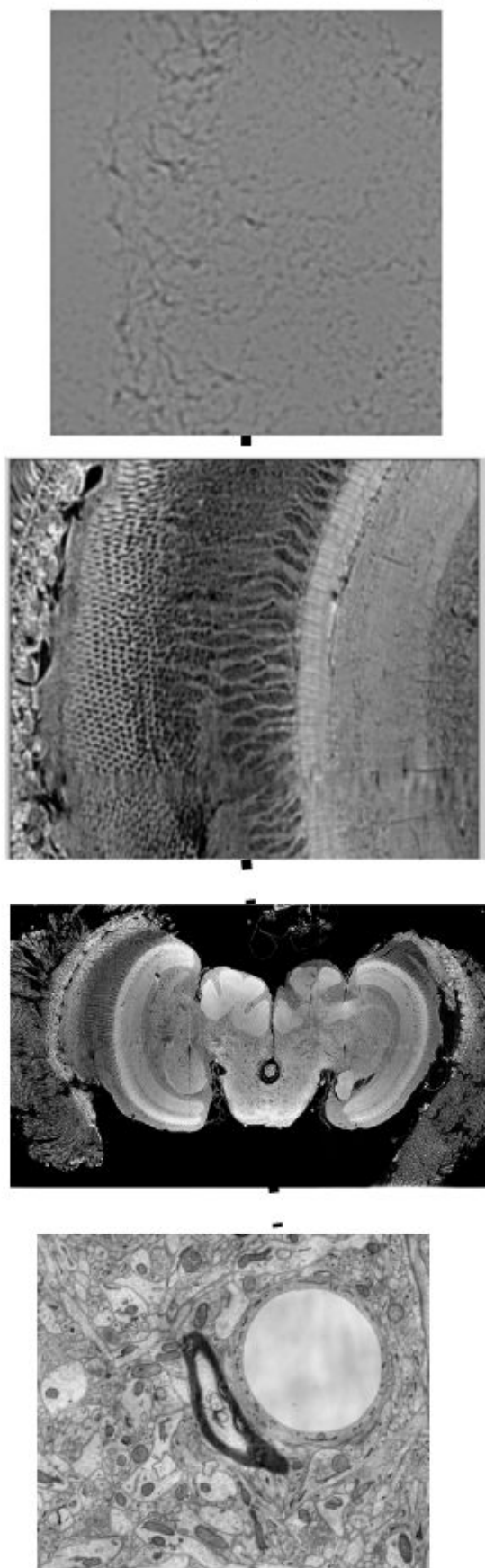
Fuel Quantity

Bank of Filters Workflow Overview

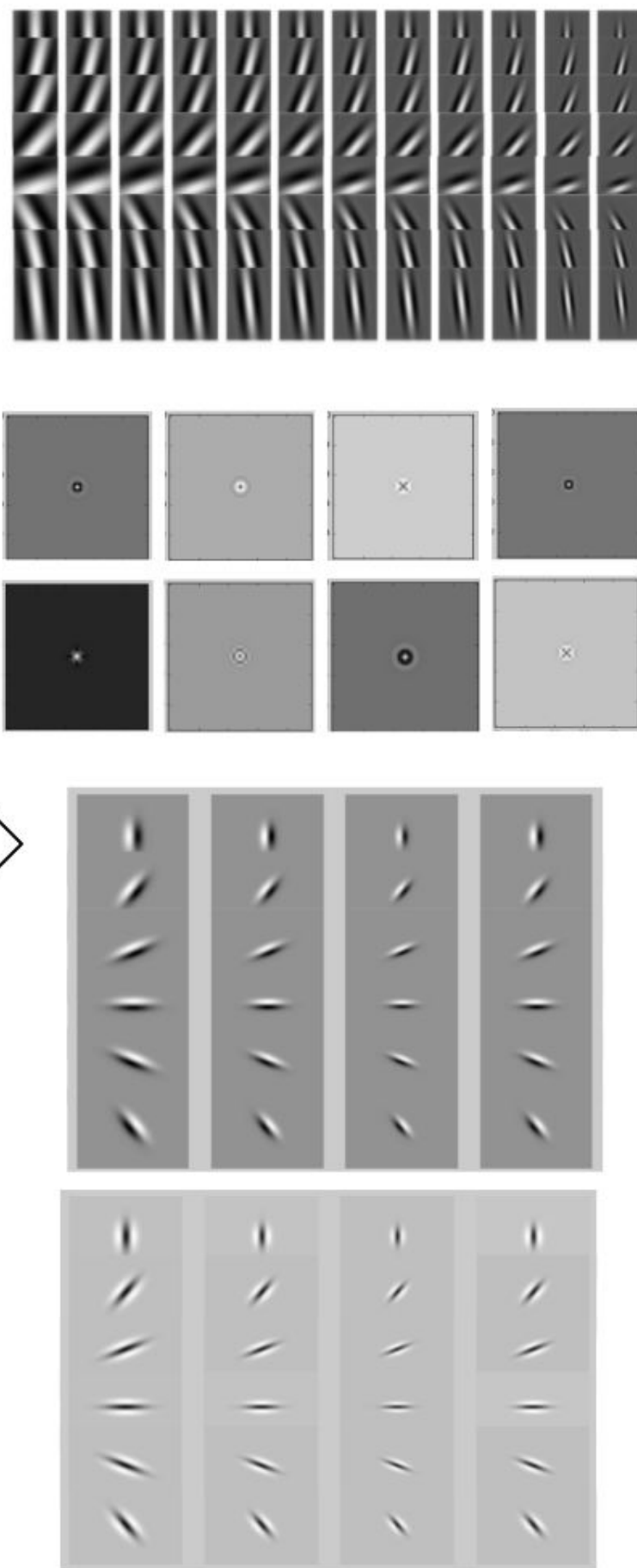


- We present the design and evaluation of using banks of filters on the Cray Urika-GX and the Cooley system and evaluate the performance with combustion engine datasets imaged at the APS
- We identify an optimum set of filter banks, with the ability to switch between different types of filters (gabor, edge/bar, and radial filters) and use this evaluation and analysis in scientific imaging applications.
- We scale the analysis of **210,000 (~100GB) images on 800 cores** on both Sage and Cooley systems improving the performance by **700 X over existing approaches**

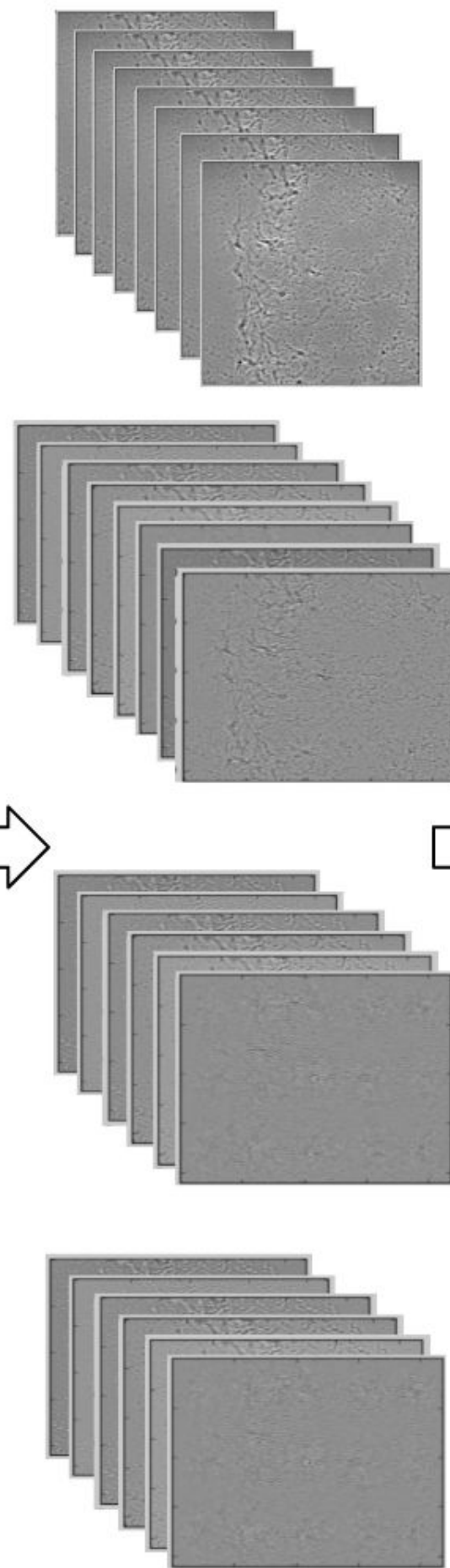
Original images (X-ray, E.M. etc)
Datasets (Combustion,
Neuroscience, Materials, etc.)



Bank of Filters



Filtered images



Images (cropped)
after Ridge Detection

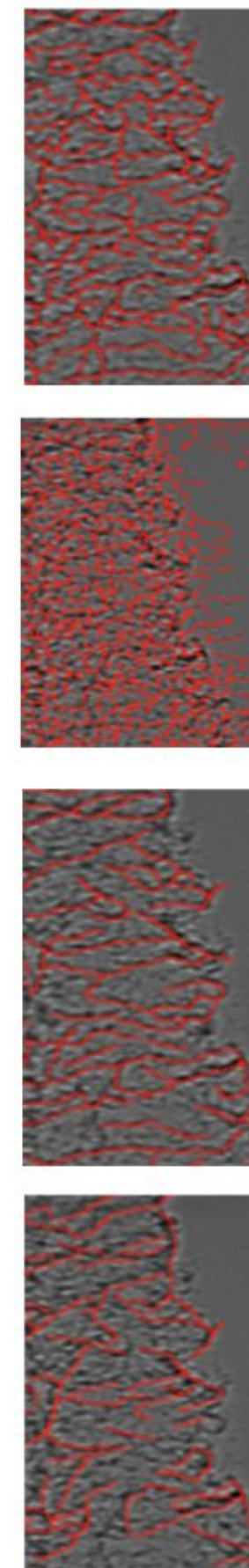
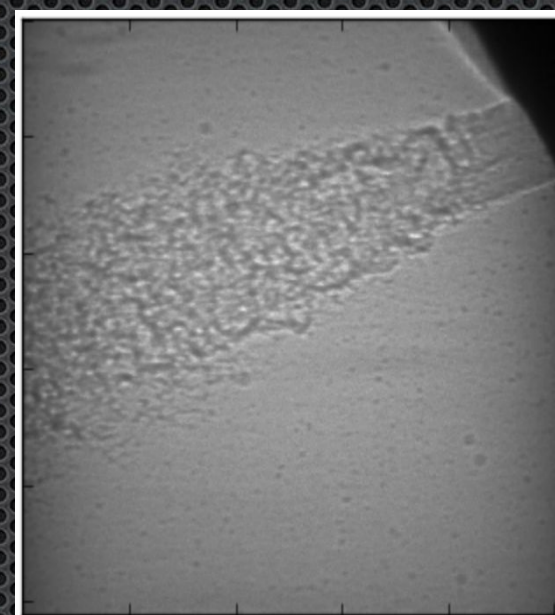


Image Correction

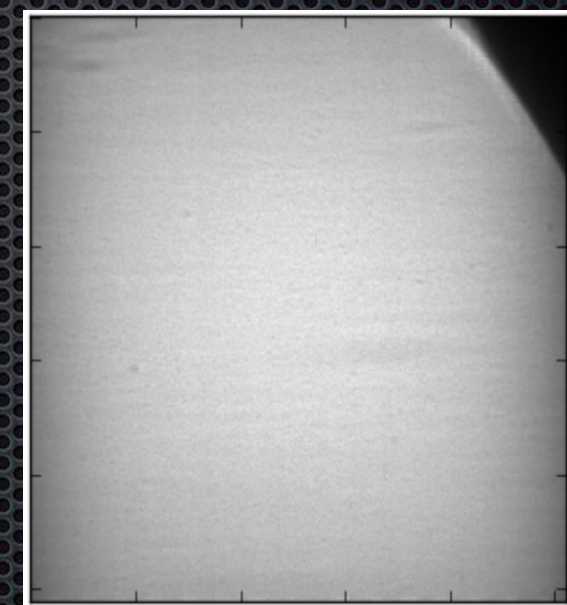
Analysis of the video sequences (each frame is a snapshot of the sequence) of the Fuel Spray Data includes

- Background computation (average of first N frames – usually N=10 to 20 frames)
- Background separation which includes shading correction (each pixel is the ratio of the data value over the “background” image)

Original Image



Background



Final image

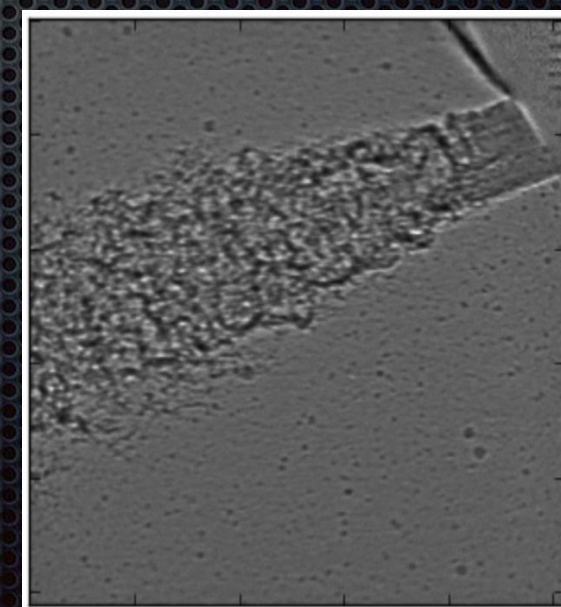
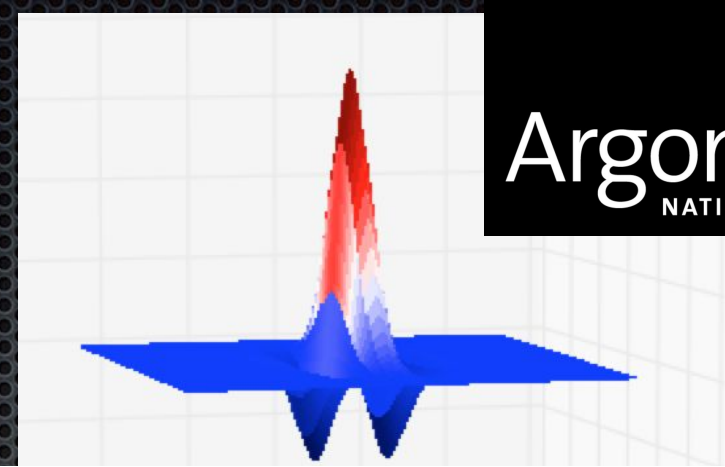
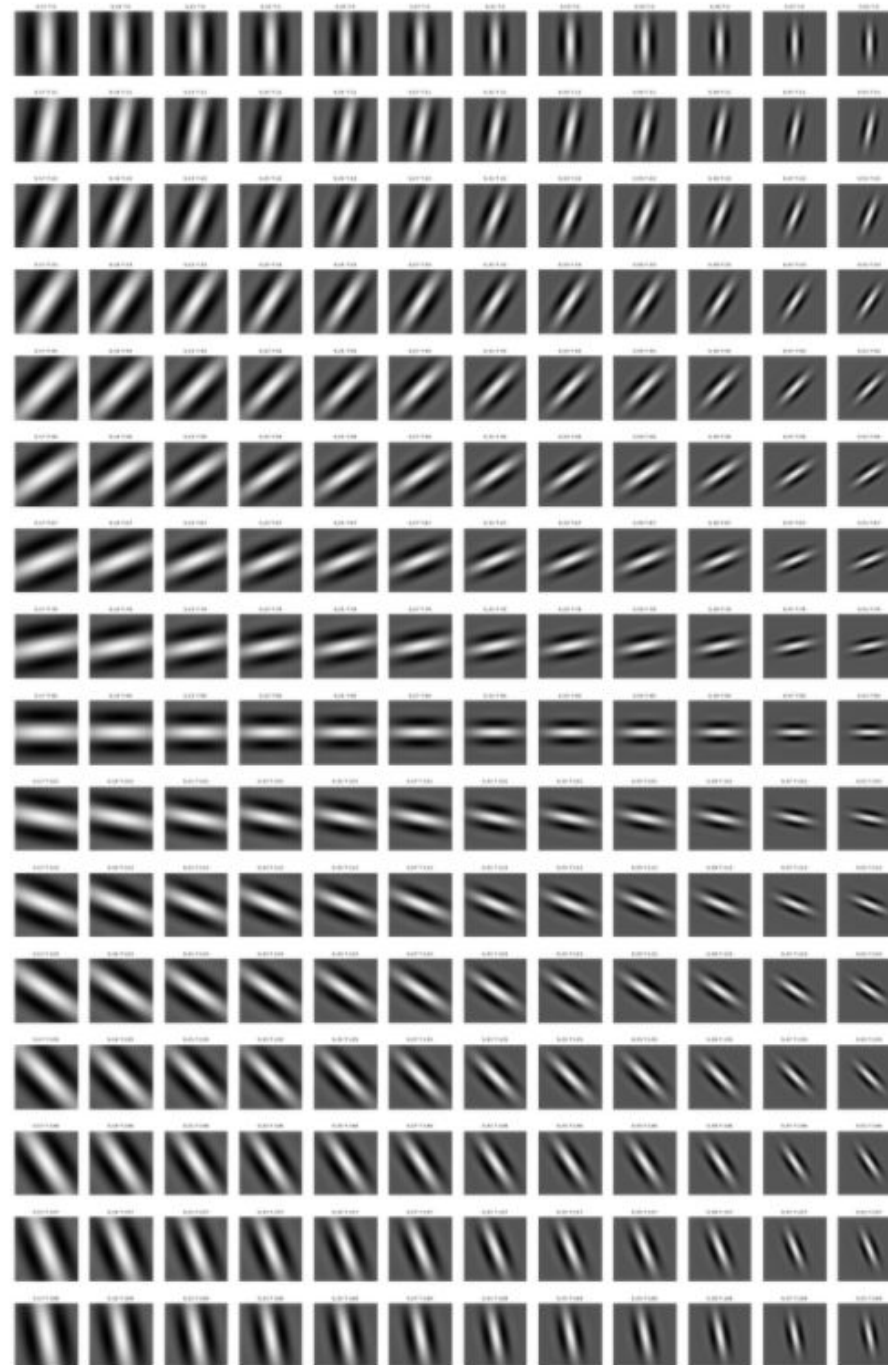


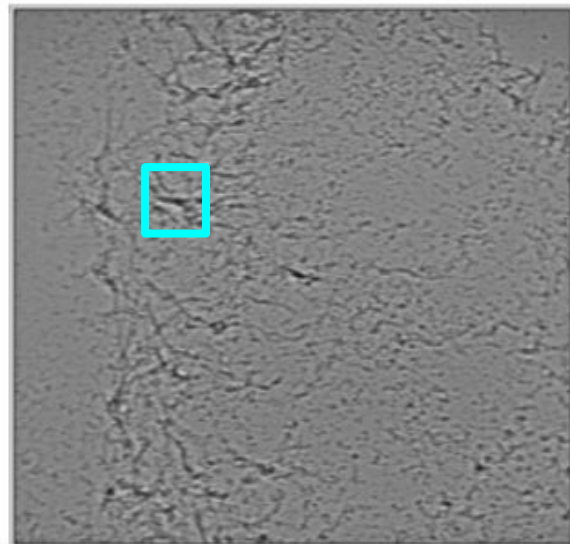
Image Filtering: Gabor



Gabor Filter(12 scales, 16 orientations)



Original Image



Filtered Image showing Maximum response

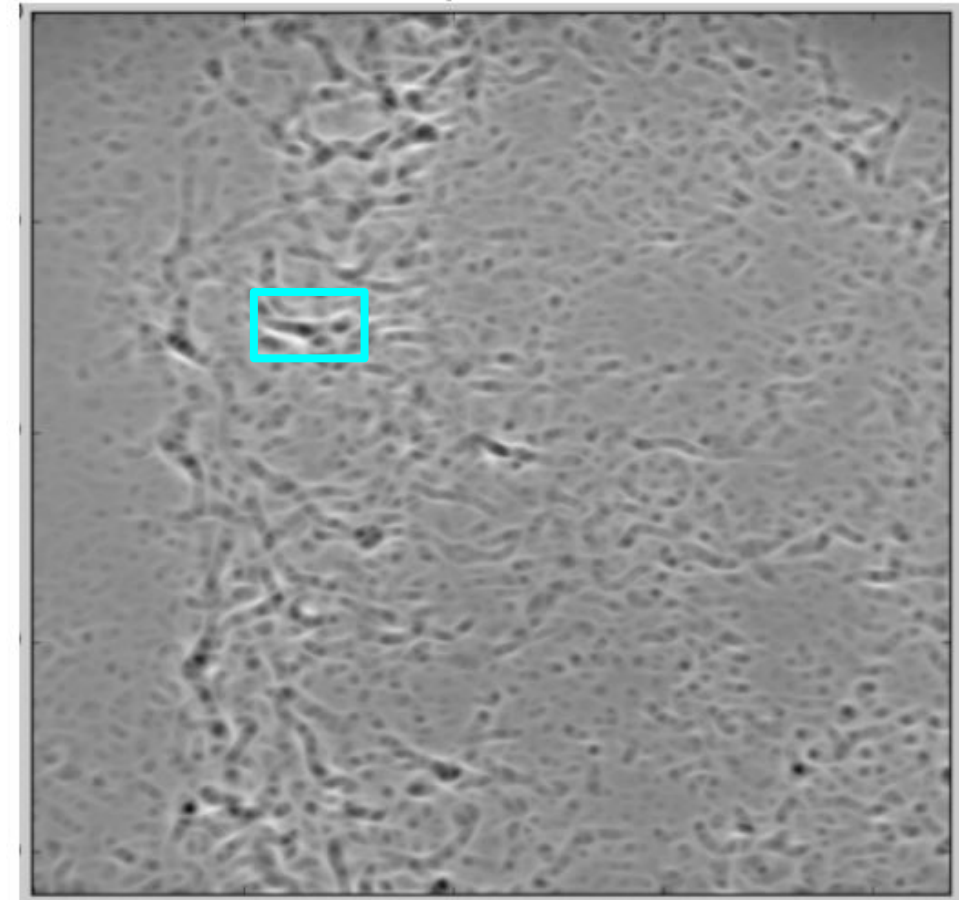
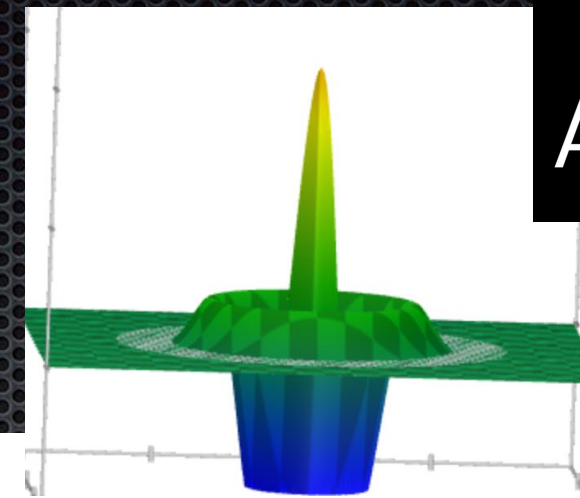
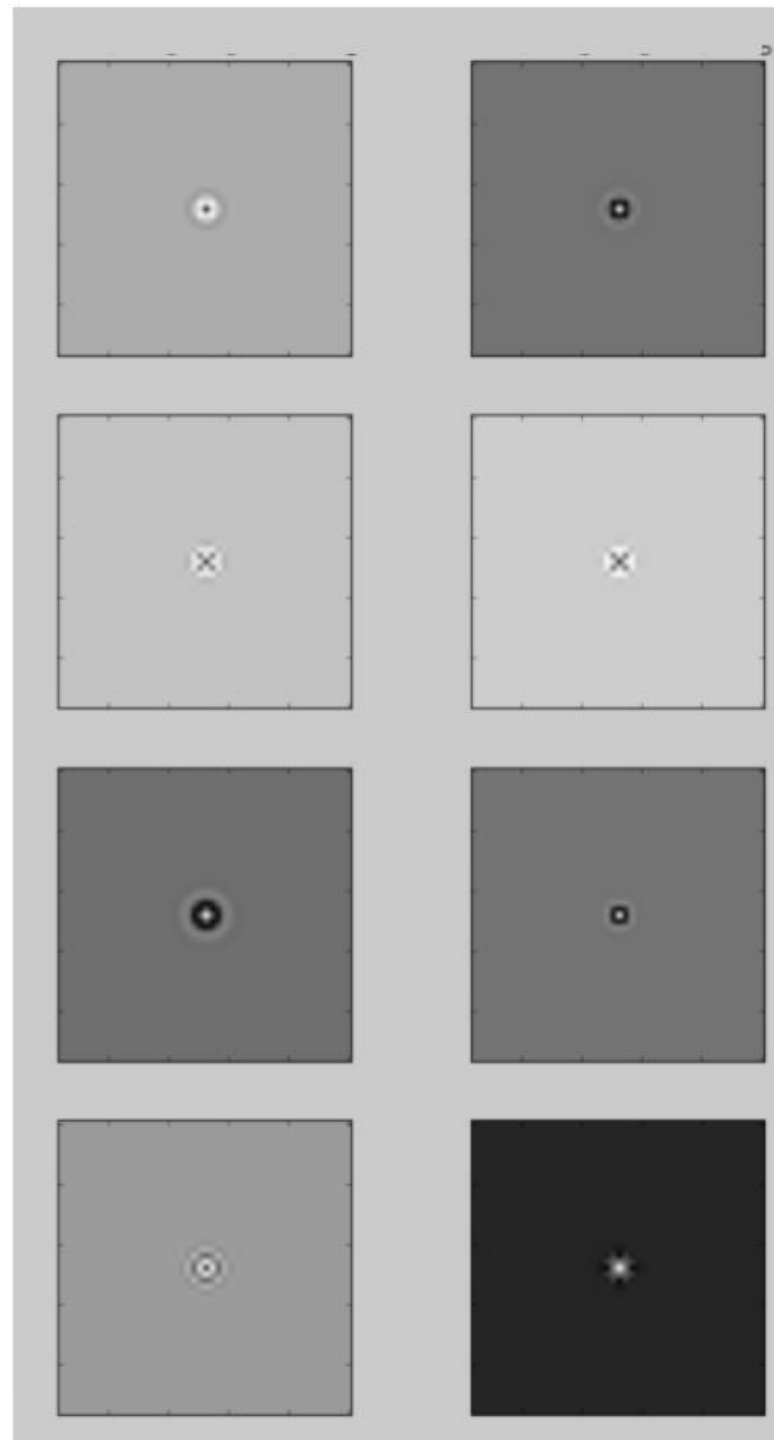
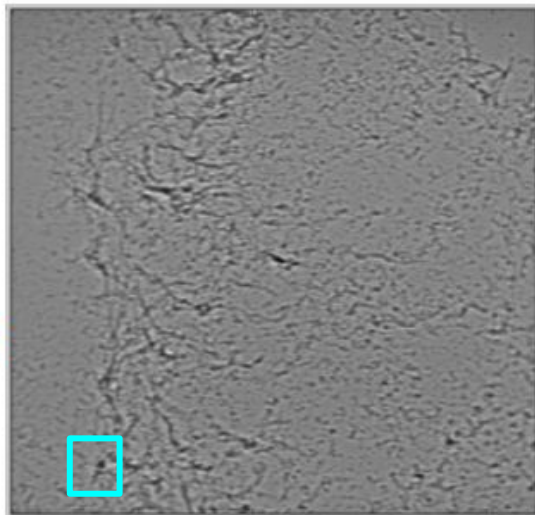


Image Filtering: Radial



Radial Filter(4 scales, $\tau = 2$)

Original Image



Filtered Image showing Maximum response

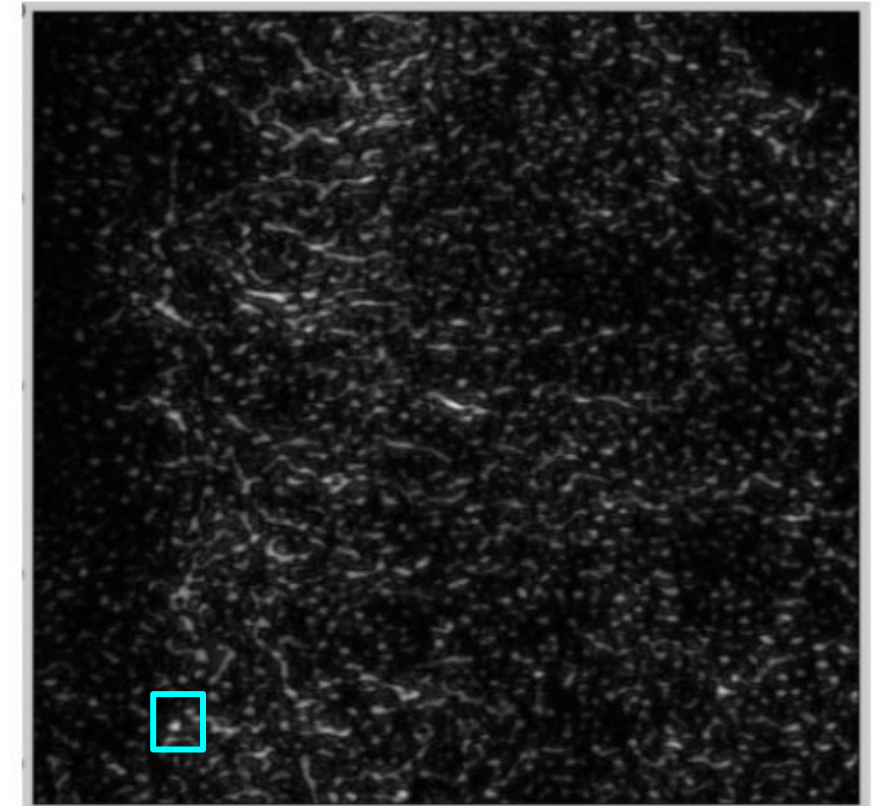
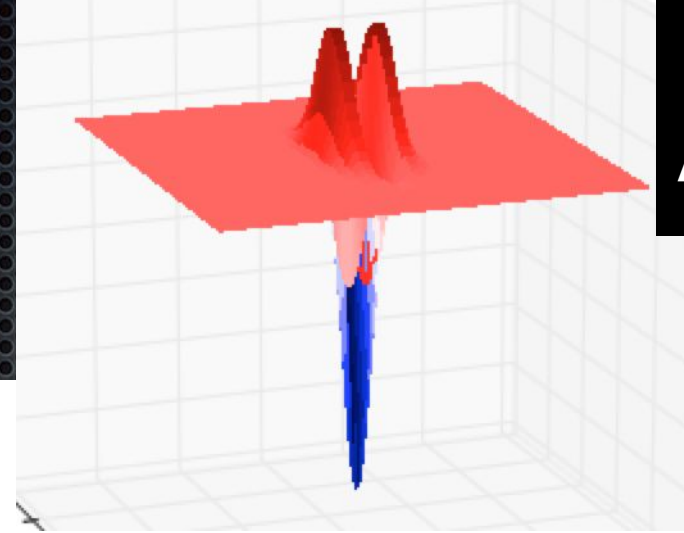
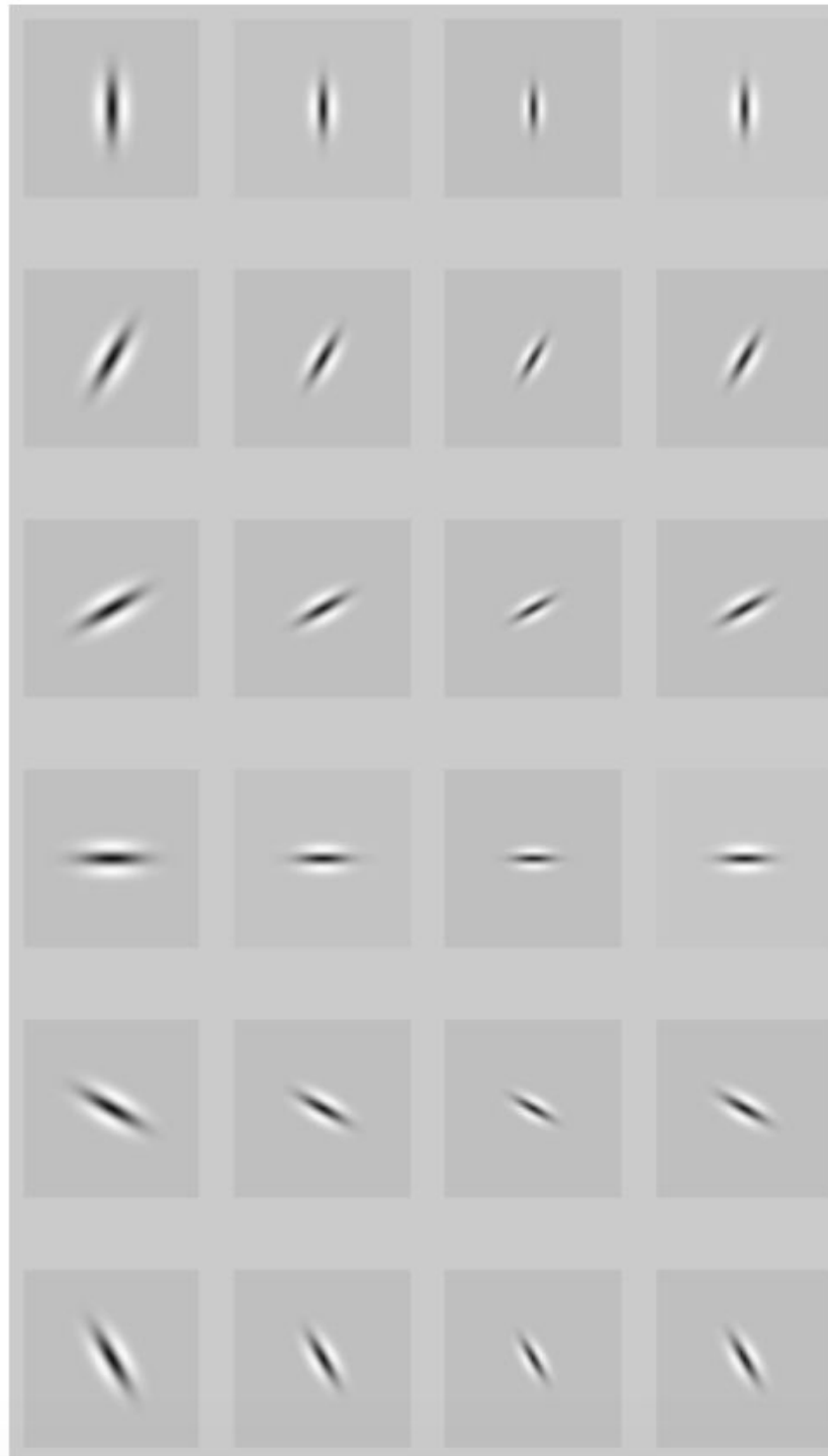


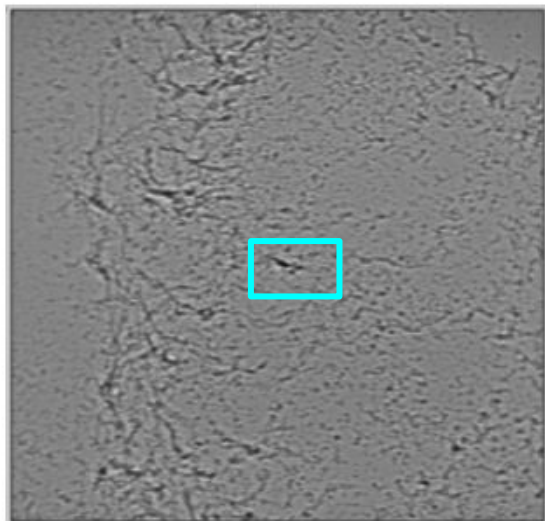
Image Filtering: Bar



Bar Filter(4 scales, 6 orientations)



Original Image



Filtered Image showing Maximum response

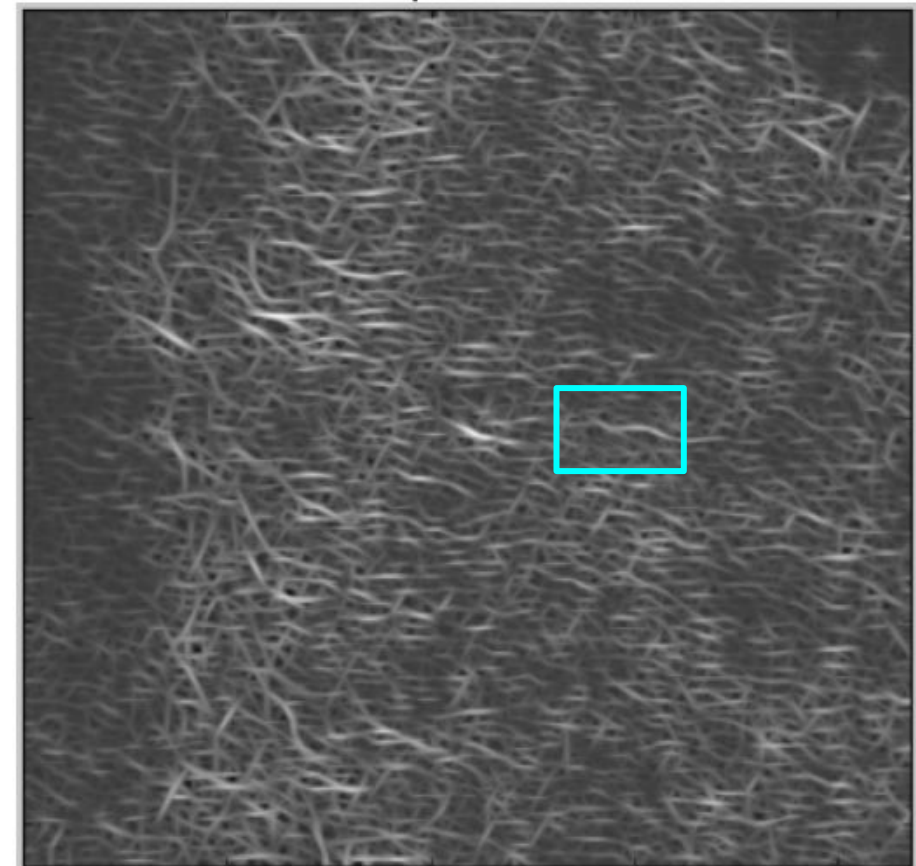
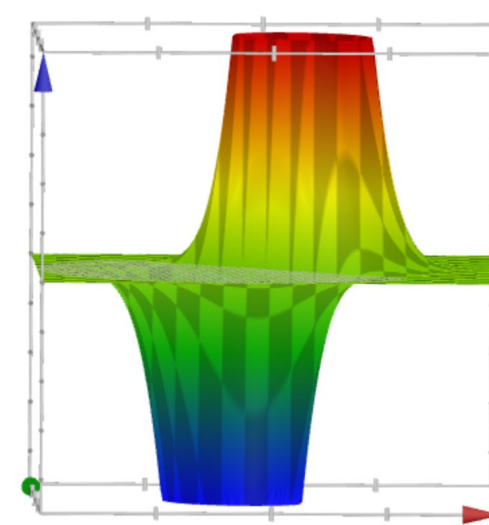
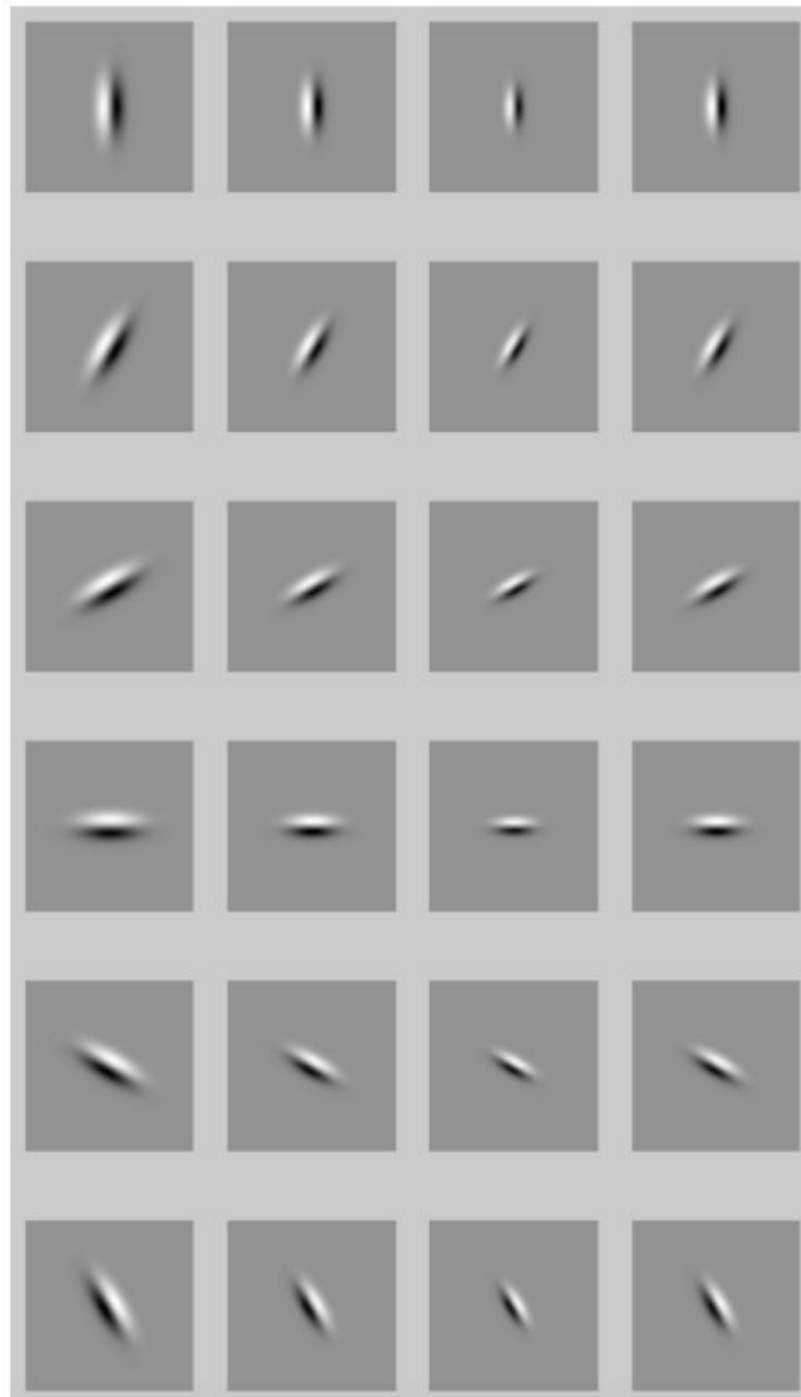


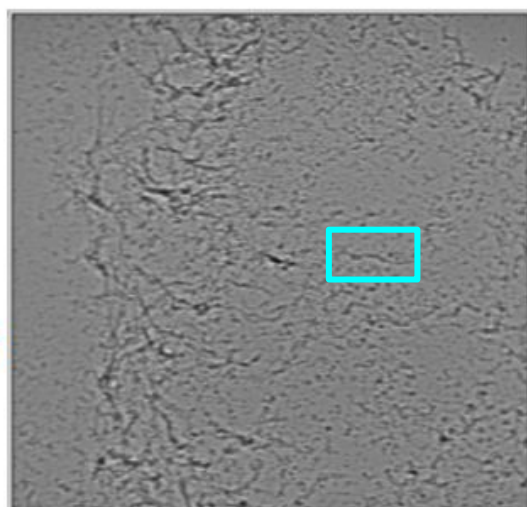
Image Filtering: Edge



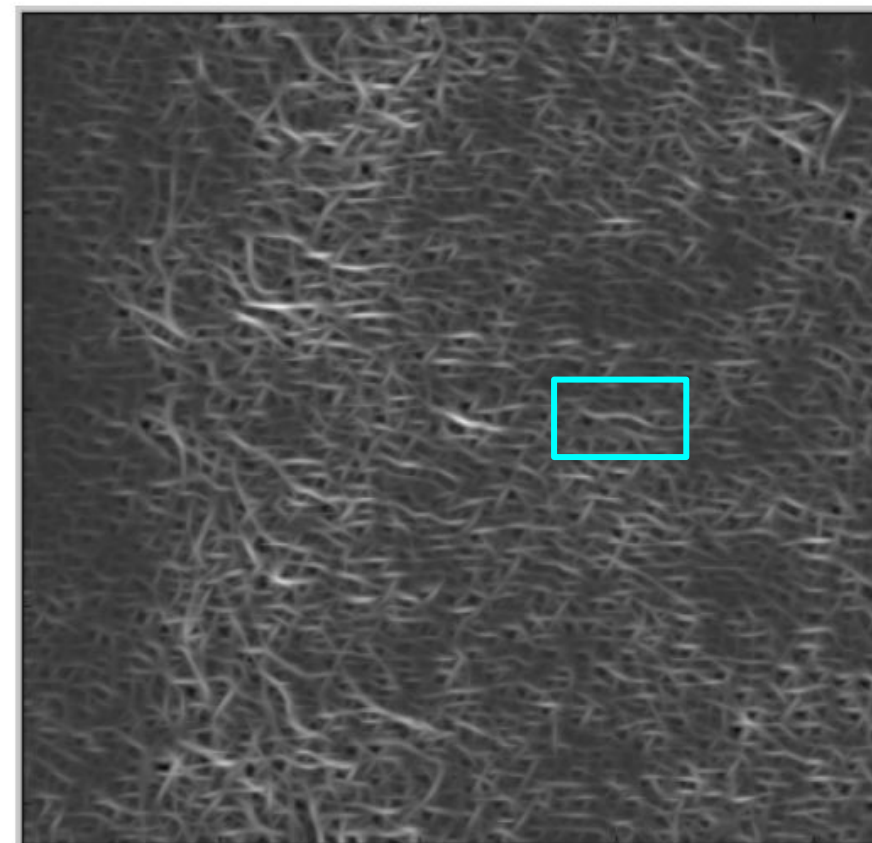
Edge Filter(4 scales, 6 orientations)



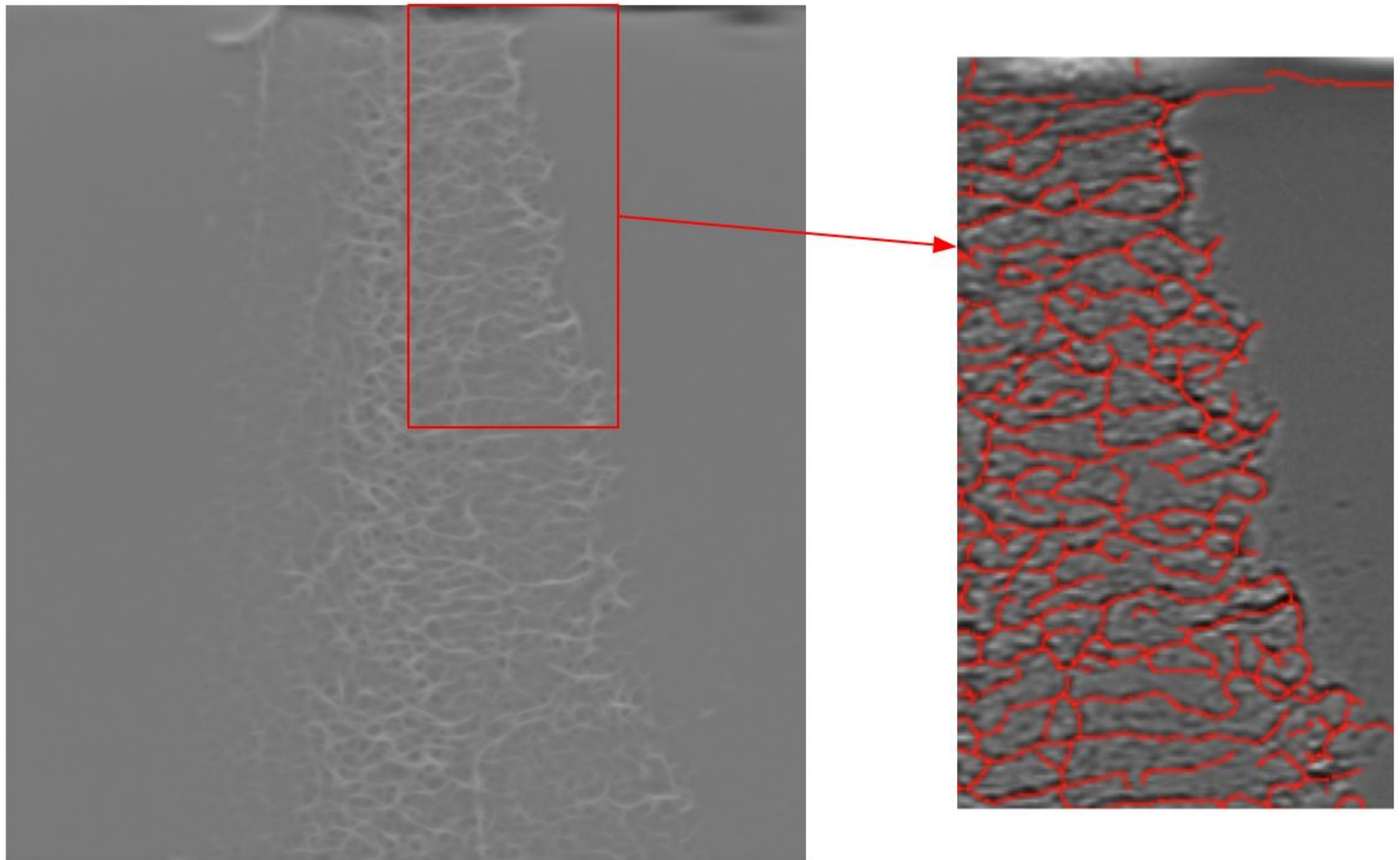
Original Image



Filtered Image showing Maximum response

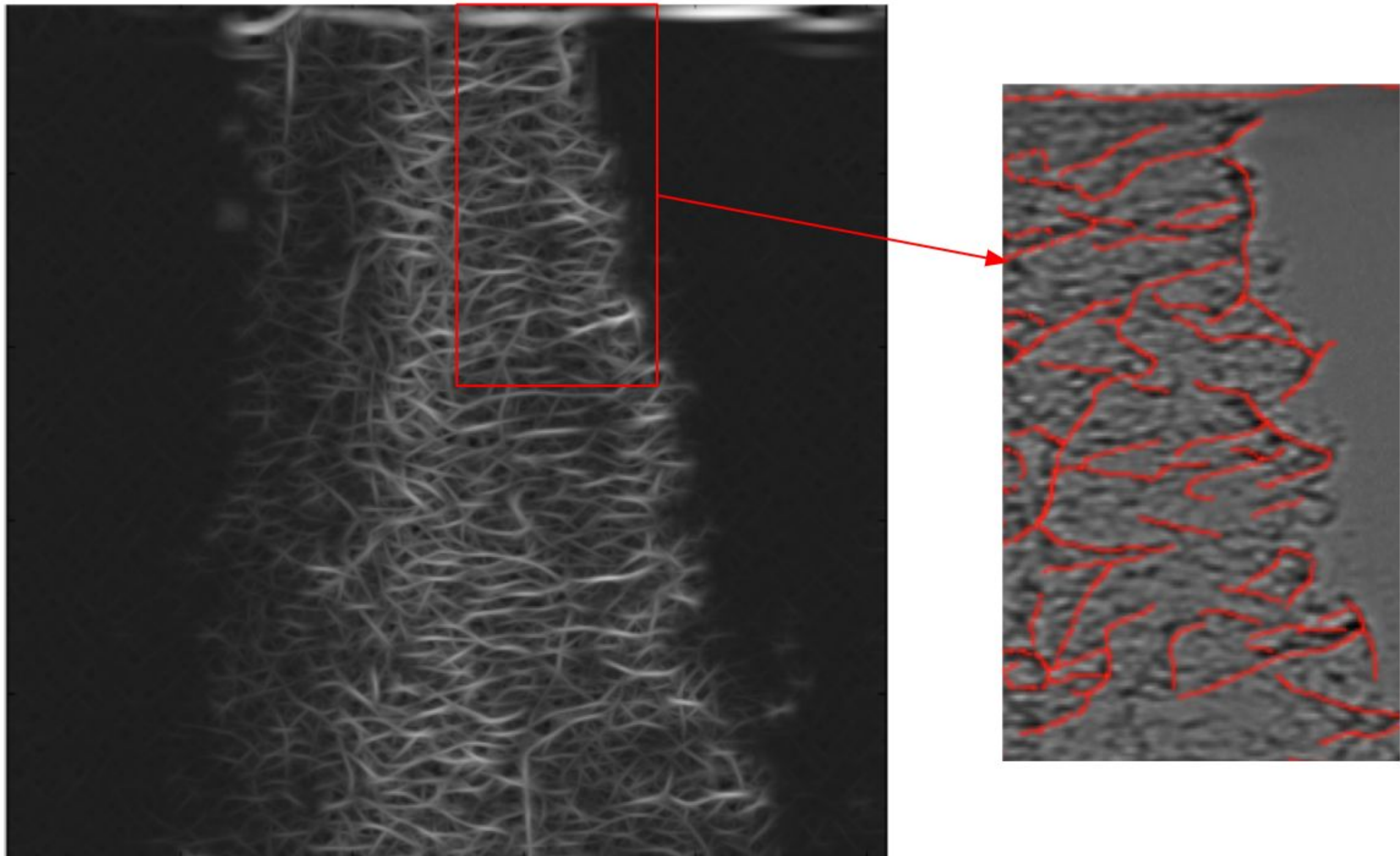


Bank of Gabor Filters and Ridge Detection



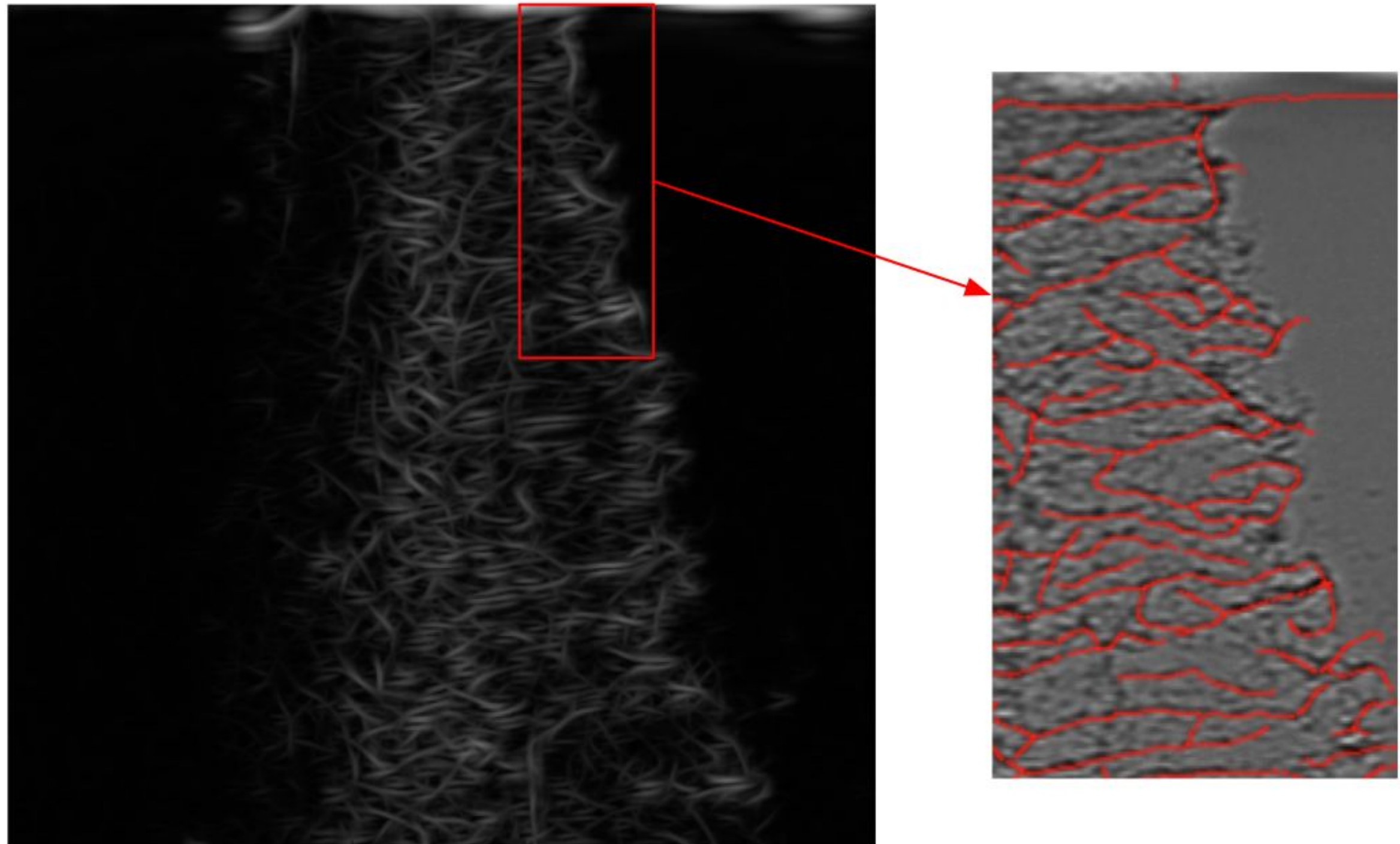
- Gabor Filter parameters include 4 scales and 35 orientations per scale 140 filters with the sigma (which corresponds to the flow thickness) =1.51

Bank of Bar Filters and Ridge Detection



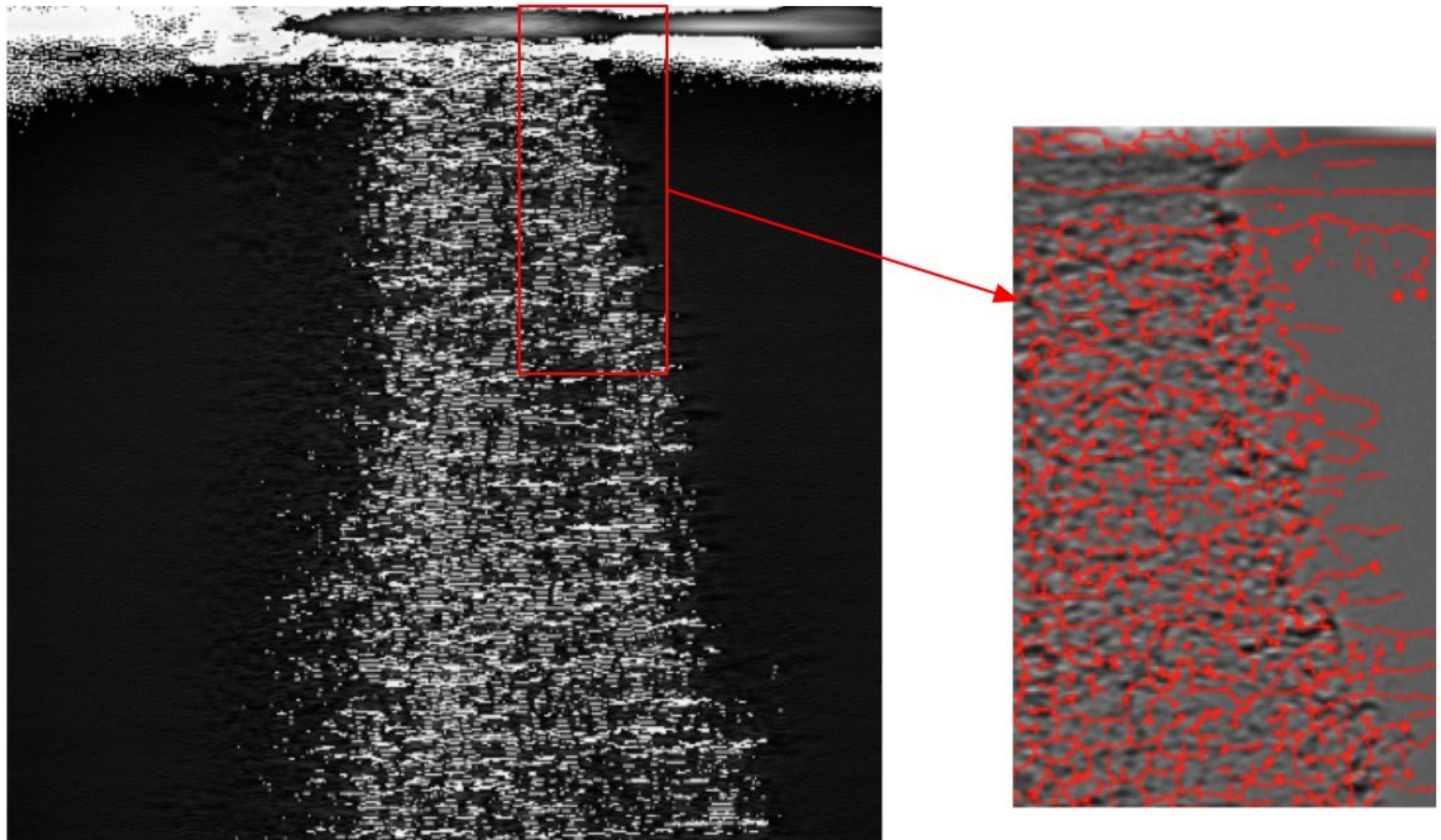
- Bar Filter parameters include 6 scales and 35 orientations per scale 210 filters kernel size set to 21×21

Bank of Edge Filters and Ridge Detection



Edge Filter parameters include 6 scales and 35 orientations per scale 210 filters kernel size set to 21×21

Bank of Radial Filters and Ridge Detection



- Radial Filter Bank includes 25 rotationally invariant filters

Ridge detector results for the following parameters set

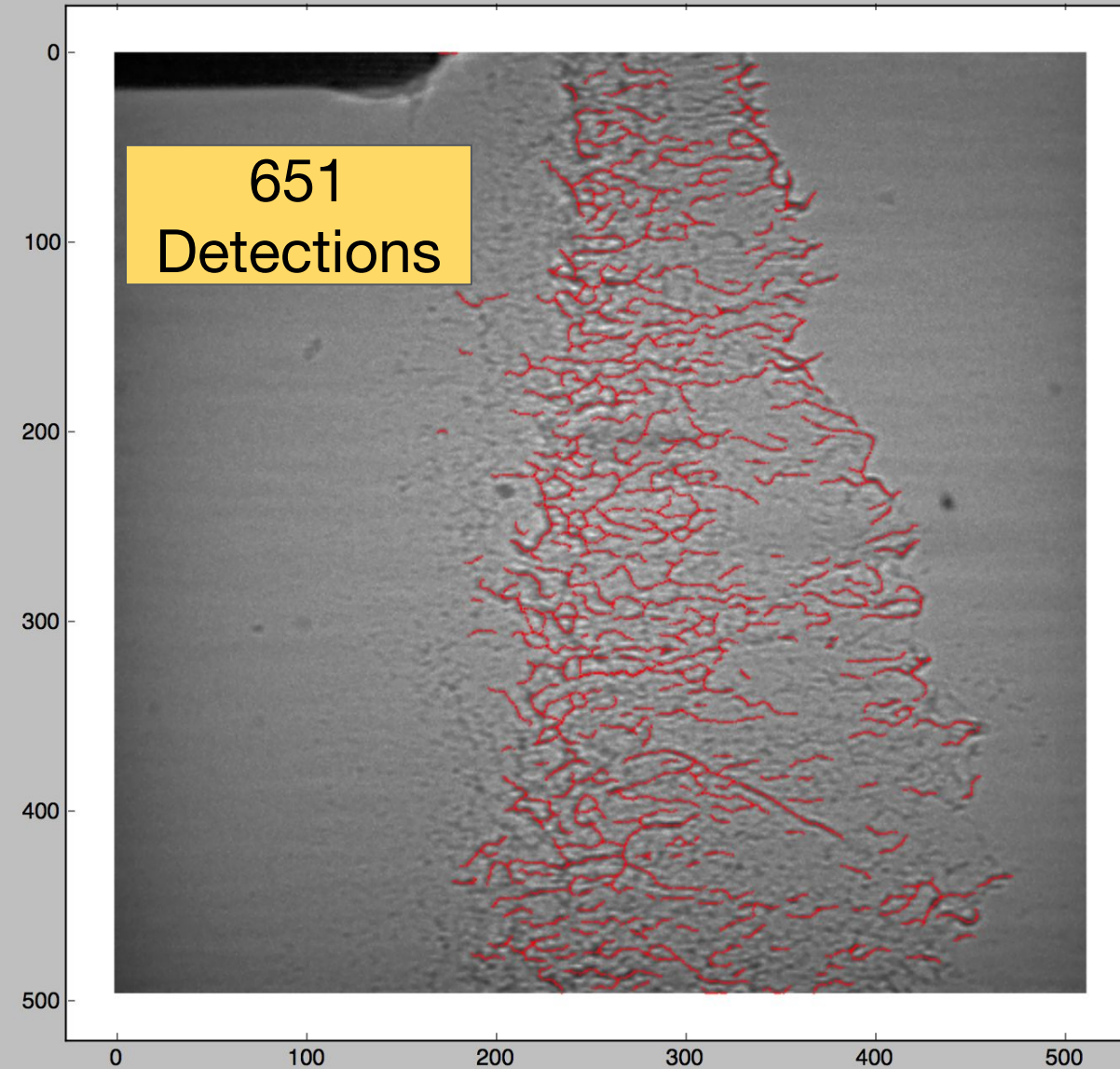
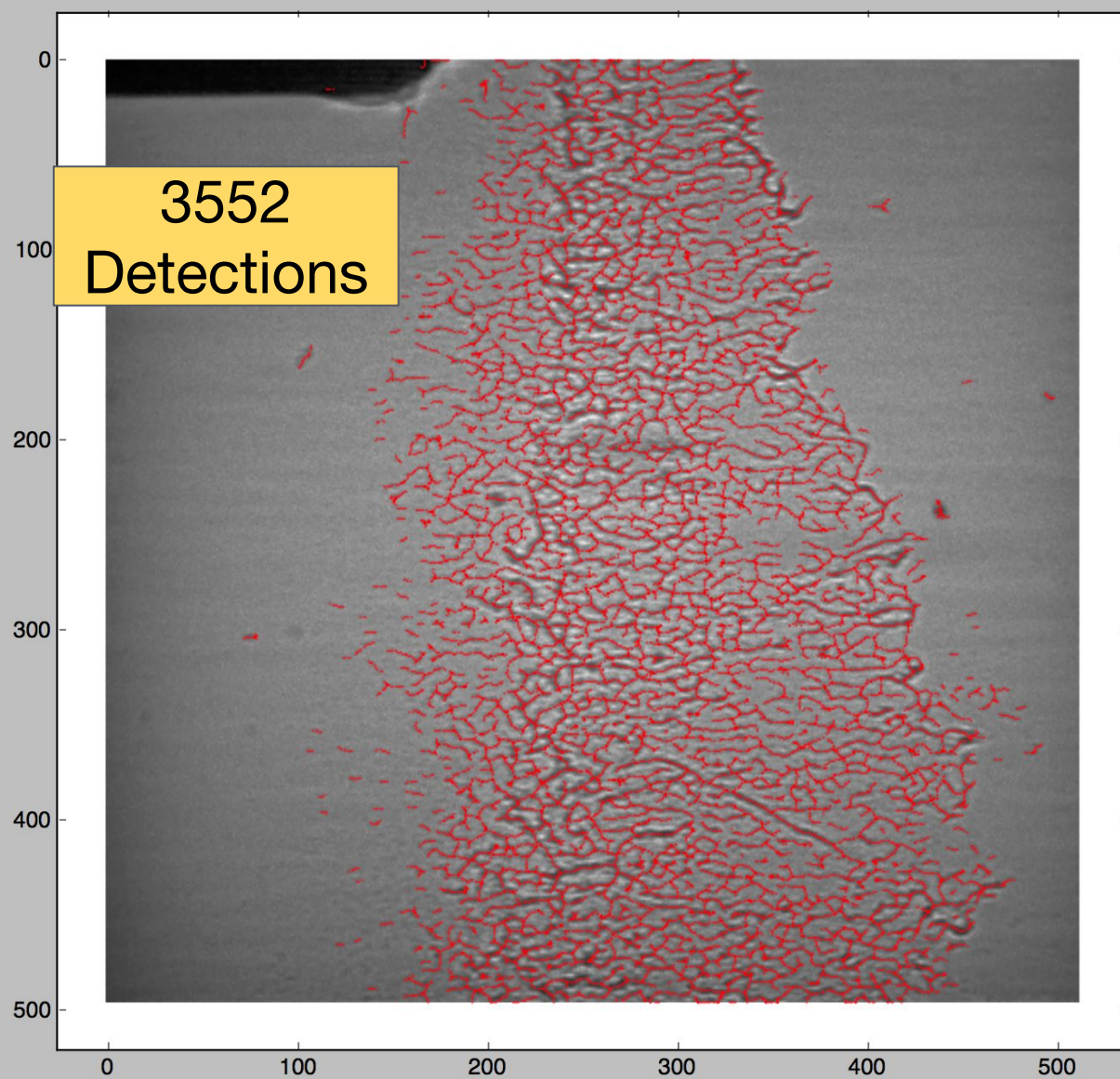
Parameter Set 1:

- . Upper Threshold = 2.0
- . Lower Threshold = 1.51
- . Sigma = 1.51 (thickness)
- . DarkLine = False (Identifies darker ridges in the image)

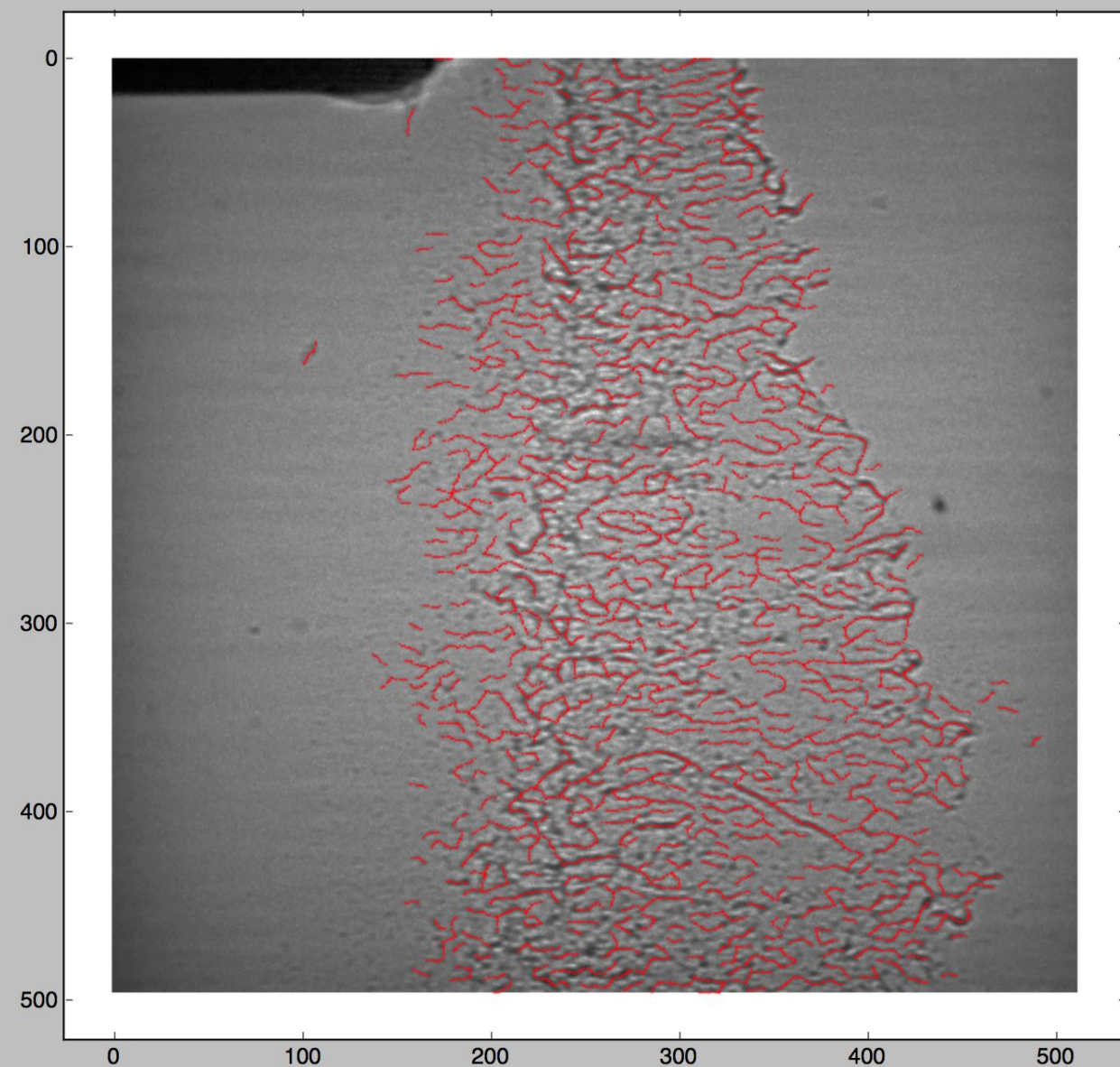
Parameter Set 2:

- . Upper Threshold = 7.0
- . Lower Threshold = 3.06
- . Sigma = 1.51 (thickness)
- . DarkLine = False (Identifies darker ridges in the image)

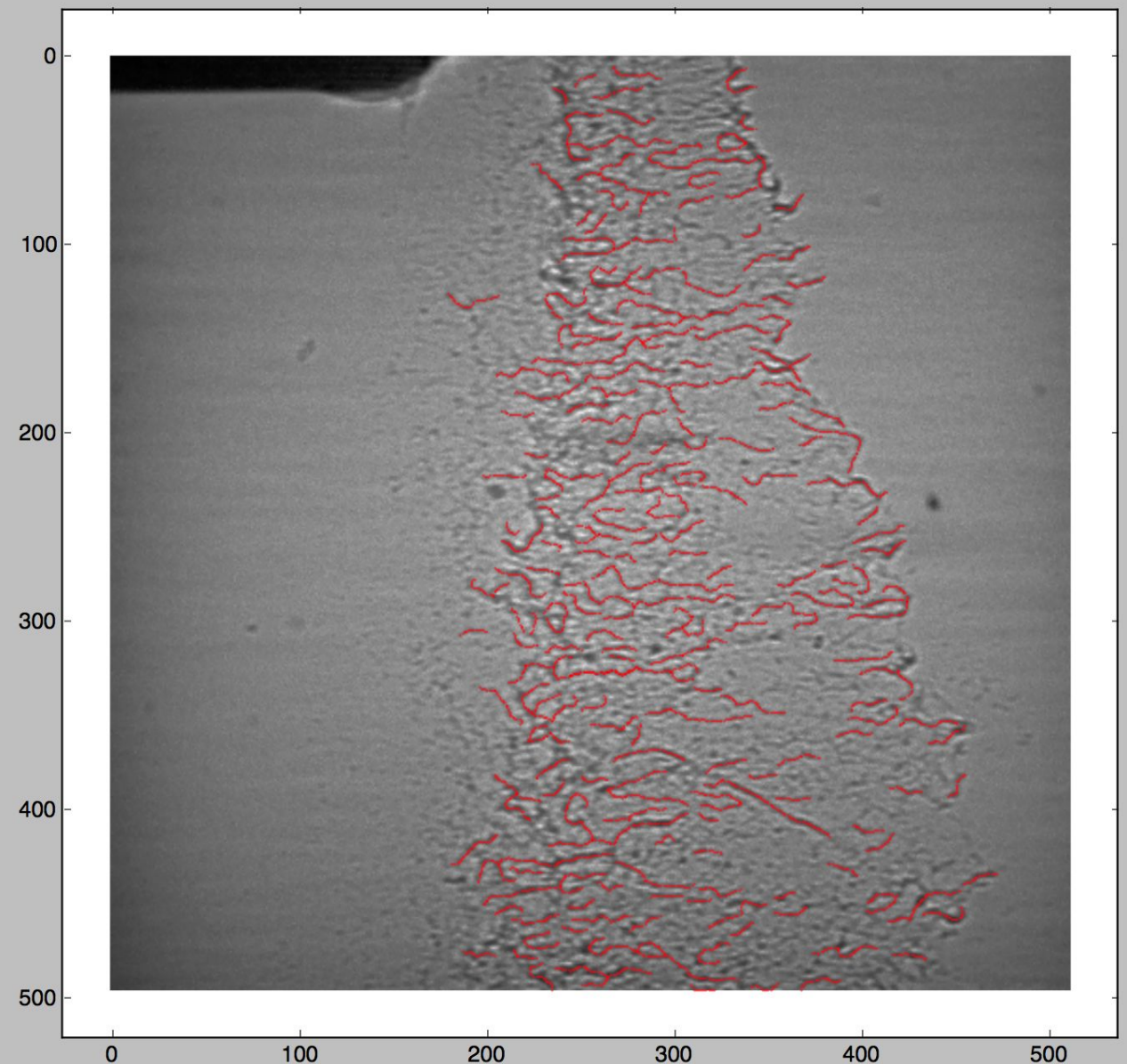
All Detected Ridges



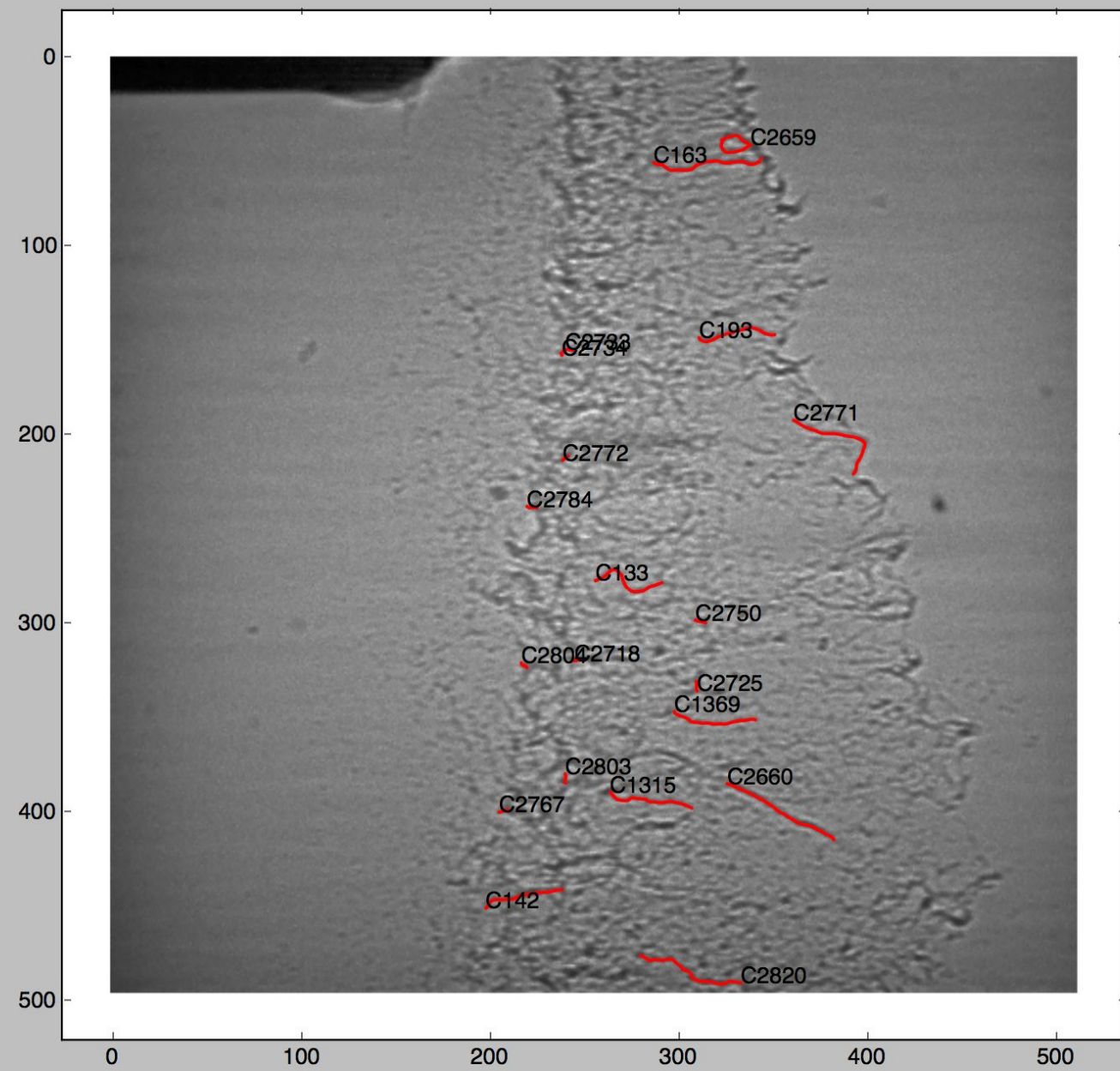
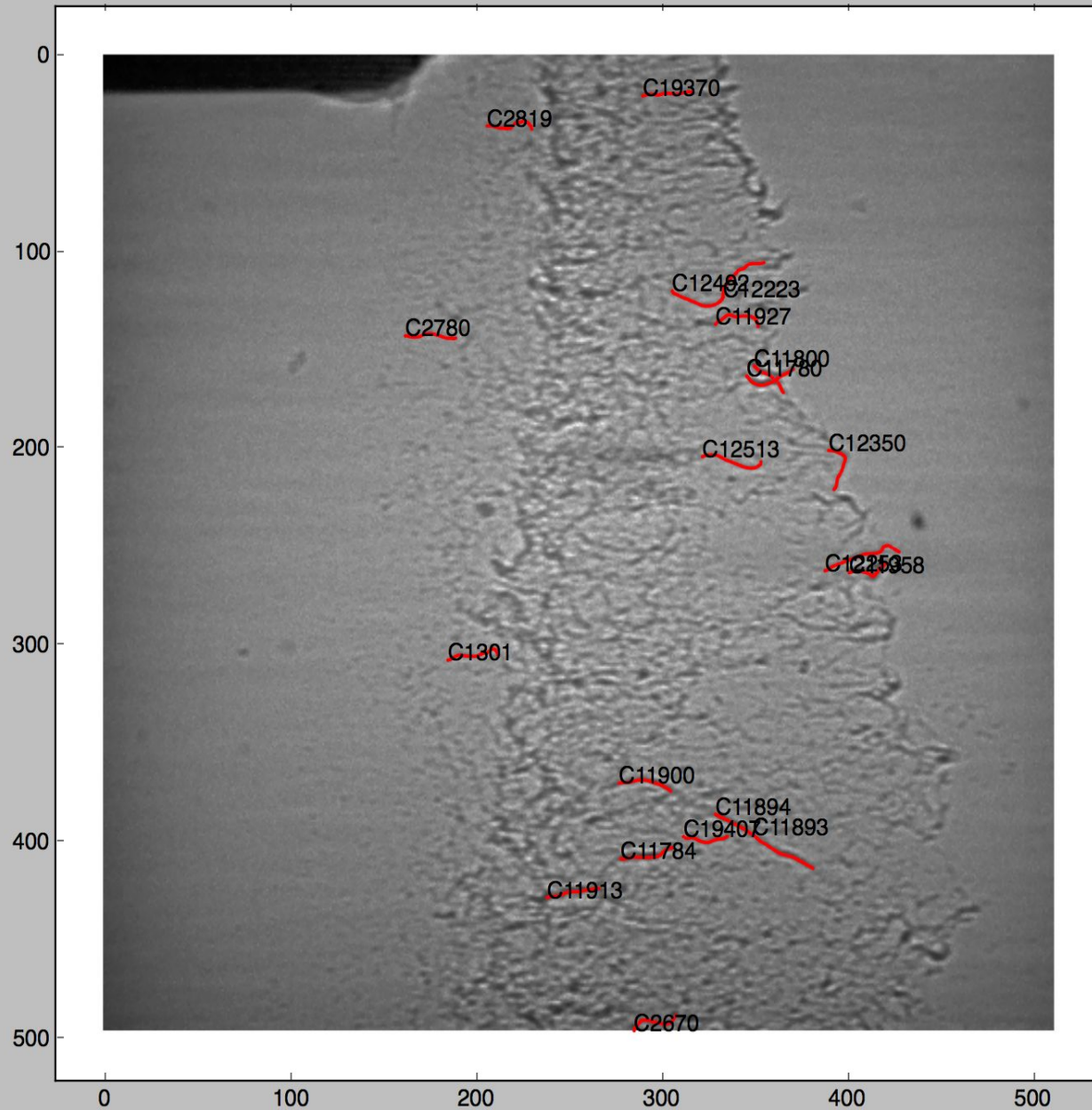
Longest 2000 ridges



Longest 300 ridges

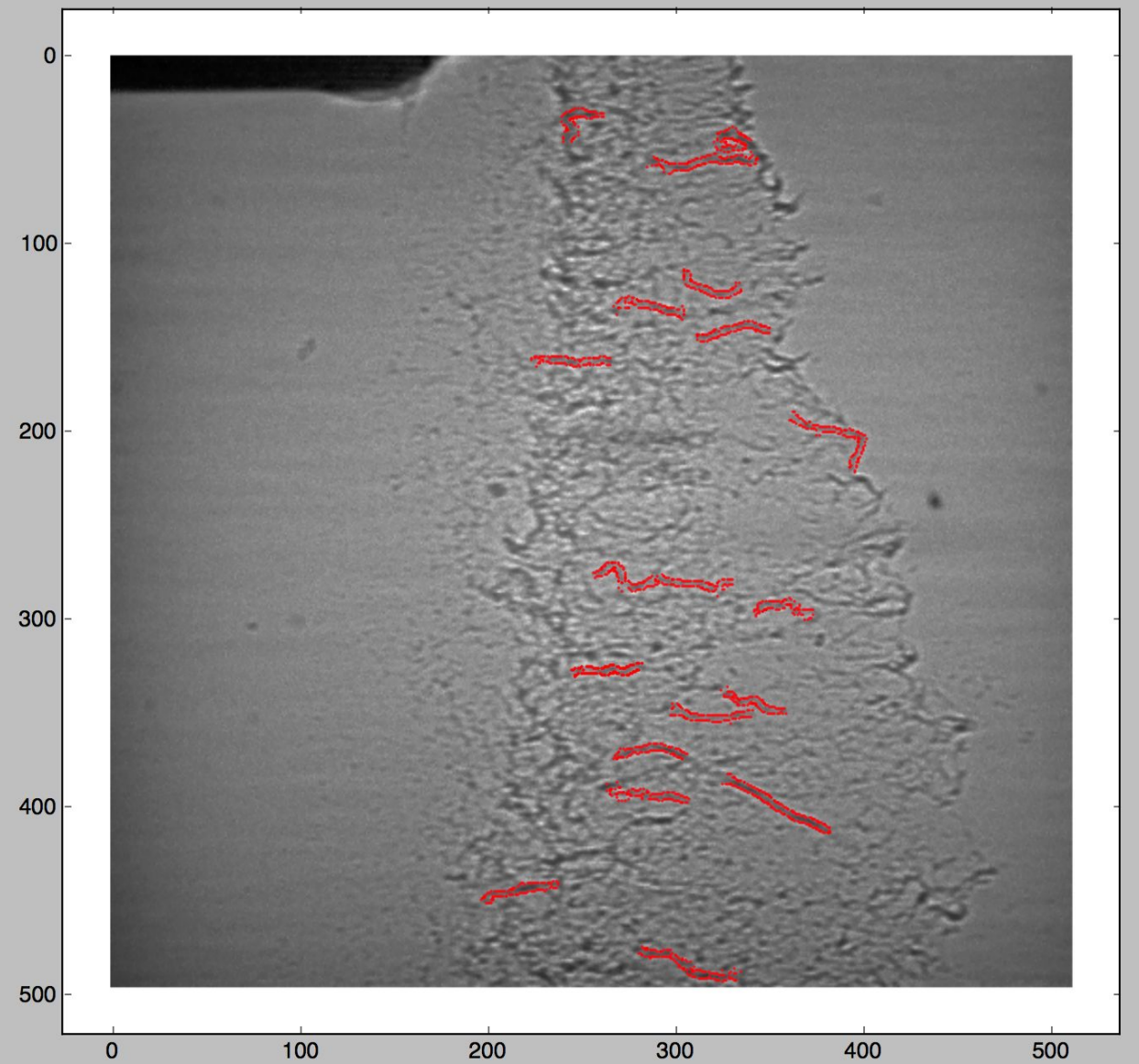
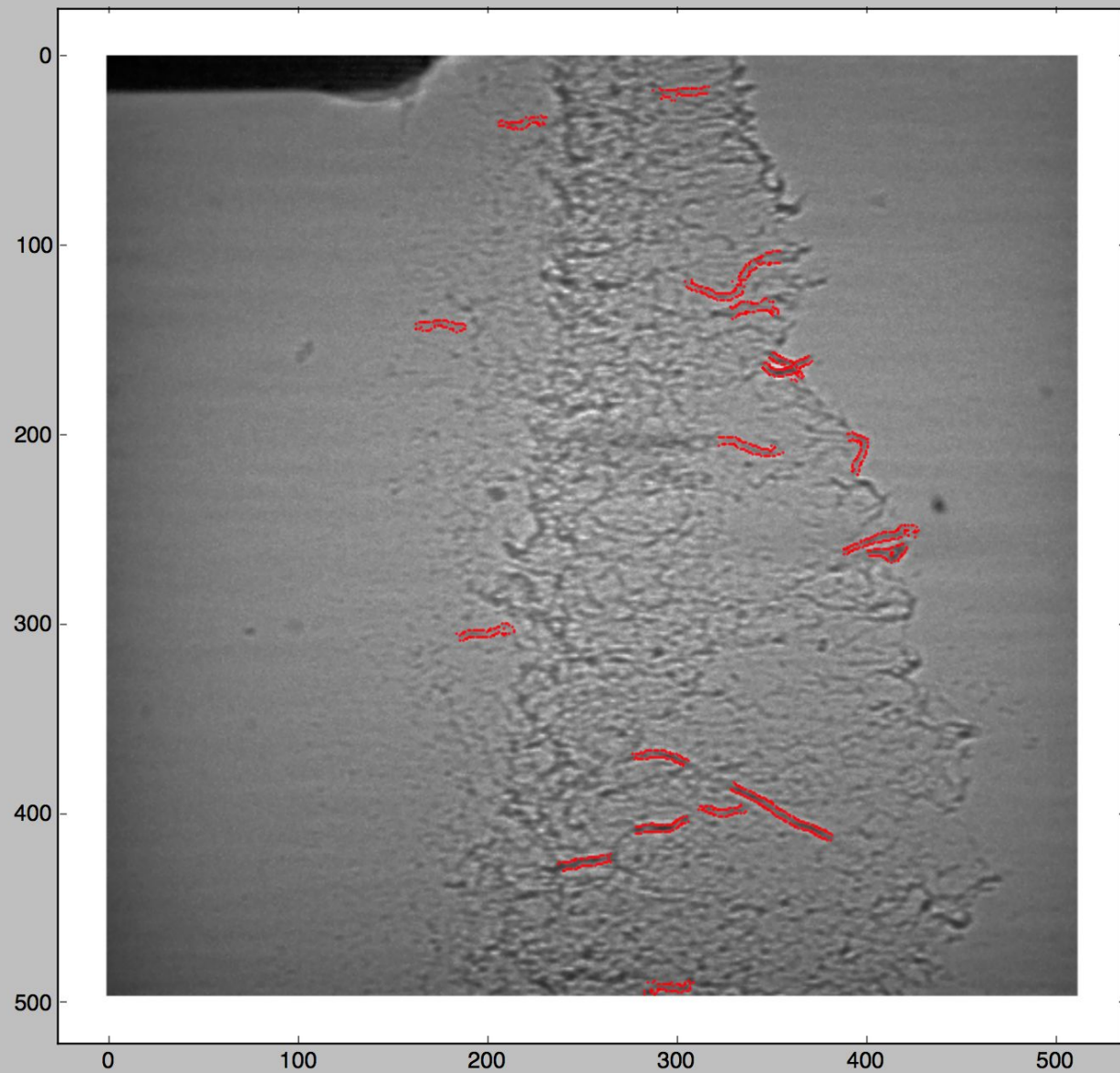


20 Ridges with Annotation



- The resulting csv files generated assign a unique id to each ridge and the characteristics of a ridge like the Ridge ID, Centroid (x and y), length of the ridge, start point of a ridge, end point of a ridge, average ridge width, minimum width, maximum width, standard deviation.

Edge Results for 20 Ridges

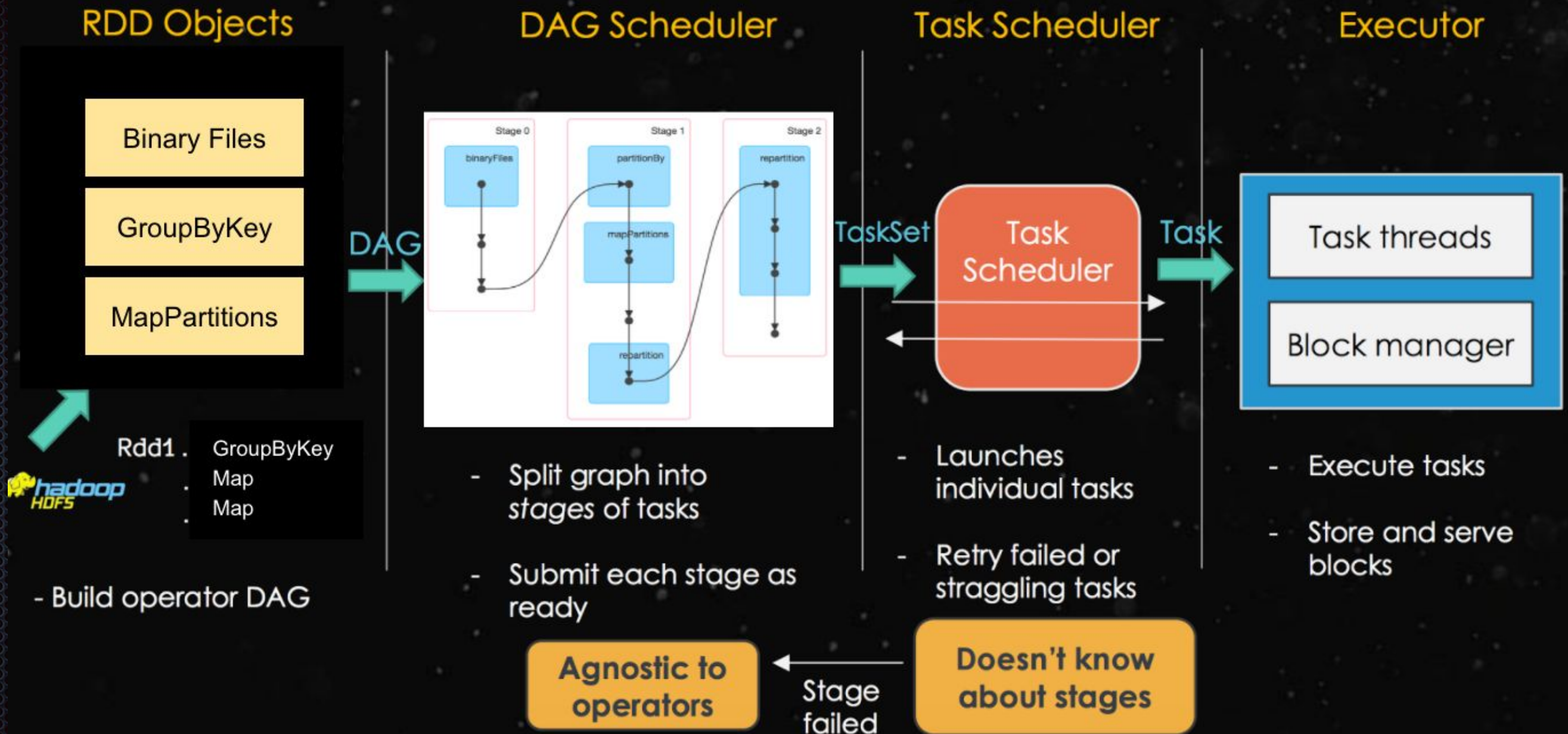


Cray Urika-GX Sage System and Cooley System

- We evaluate the performance of the bank of filters on the Sage Urika-GX system and the Cooley cluster at the Argonne National Laboratory.
- Sage system consists of 32 nodes, consisting of 25 compute nodes and 7 service nodes
 - Each node has two 16-core Intel processors
 - 256 gigabytes of RAM
 - 800 gigabytes of SSD
 - 4 terabytes of hard-disk.
 - Shared HDFS-based distributed file system
- Cooley system has 126 compute nodes
 - Each node has 2 CPU with 6 cores each per CPU
 - 384GB RAM per node
 - 345GB local scratch space
 - Shared GPFS file system

Apache Spark Implementation

SCHEDULING PROCESS



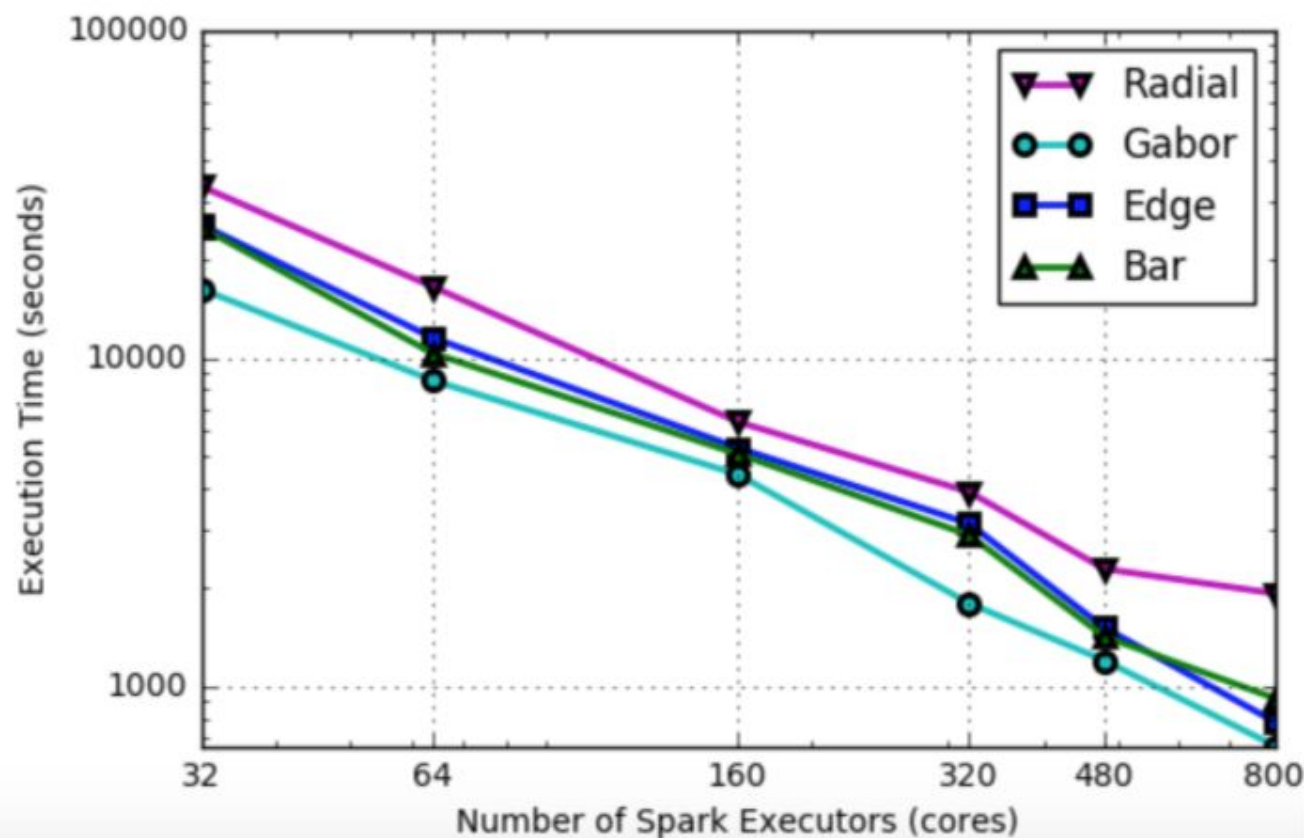
Initial Setup on Cray Urika-GX System

- Initial tests use Apache Spark(1.6.3) applications integrated with Ipython/Jupyter Notebooks
Local Anaconda and Spark (1.6.3) installations were used
- Configuration for Spark (1.6.3) is similar to the default configuration available for Spark 1.5 on the Sage System
- We have now moved on to Spark 2.0.2 integrated with Ipython/Jupyter Notebooks
- Notebooks are accessed via the socks proxy server setup using ssh dynamic port forwarding via the jlse login nodes
<http://sagellogin1.ftm.alcf.anl.gov:8889/tree>

Strong Scaling Results on the Sage Urika-GX System and Cooley System

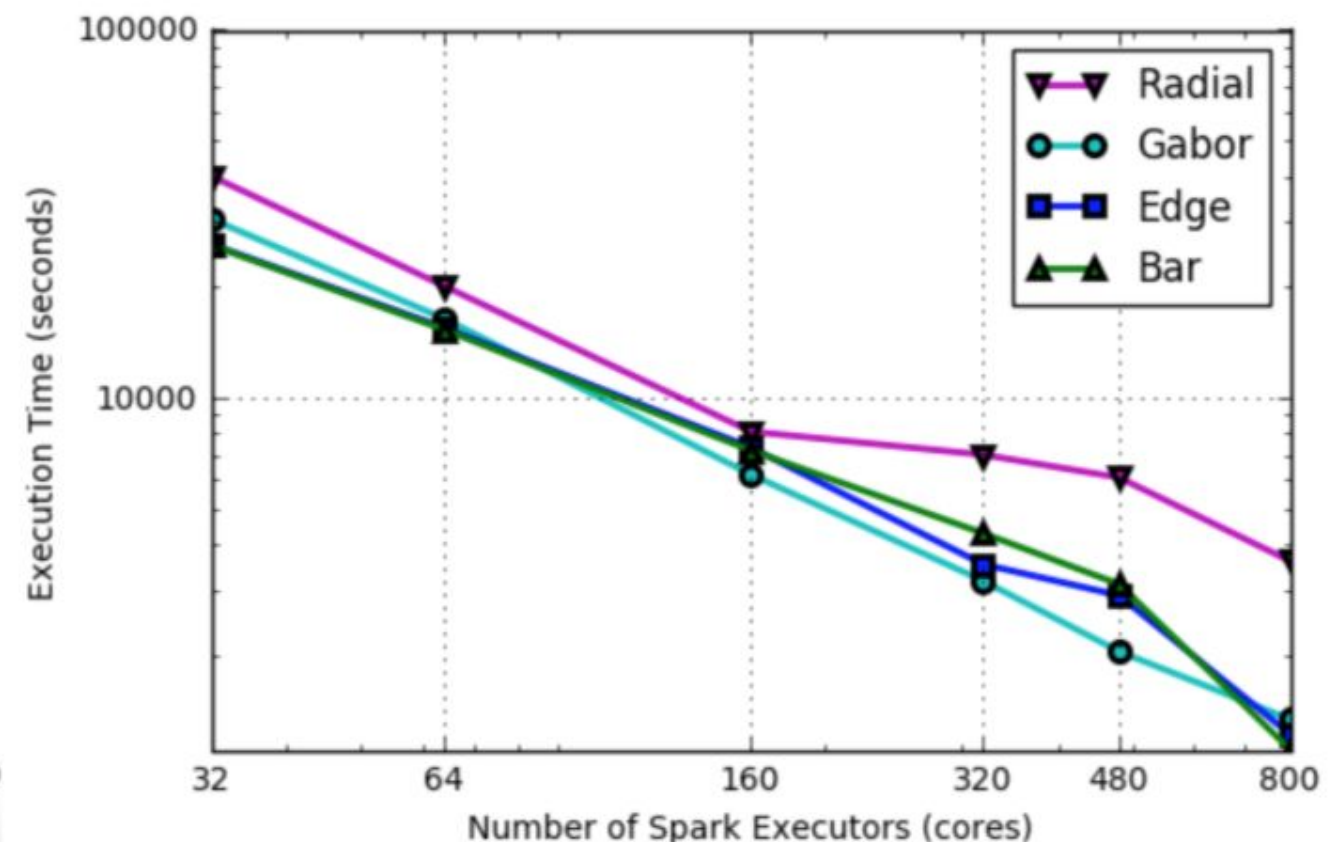
Sage Urika-GX System

Strong Scaling: 4800 Images
(~2.4GB)



Cooley System

Strong Scaling: 4800 Images
(~2.4GB)

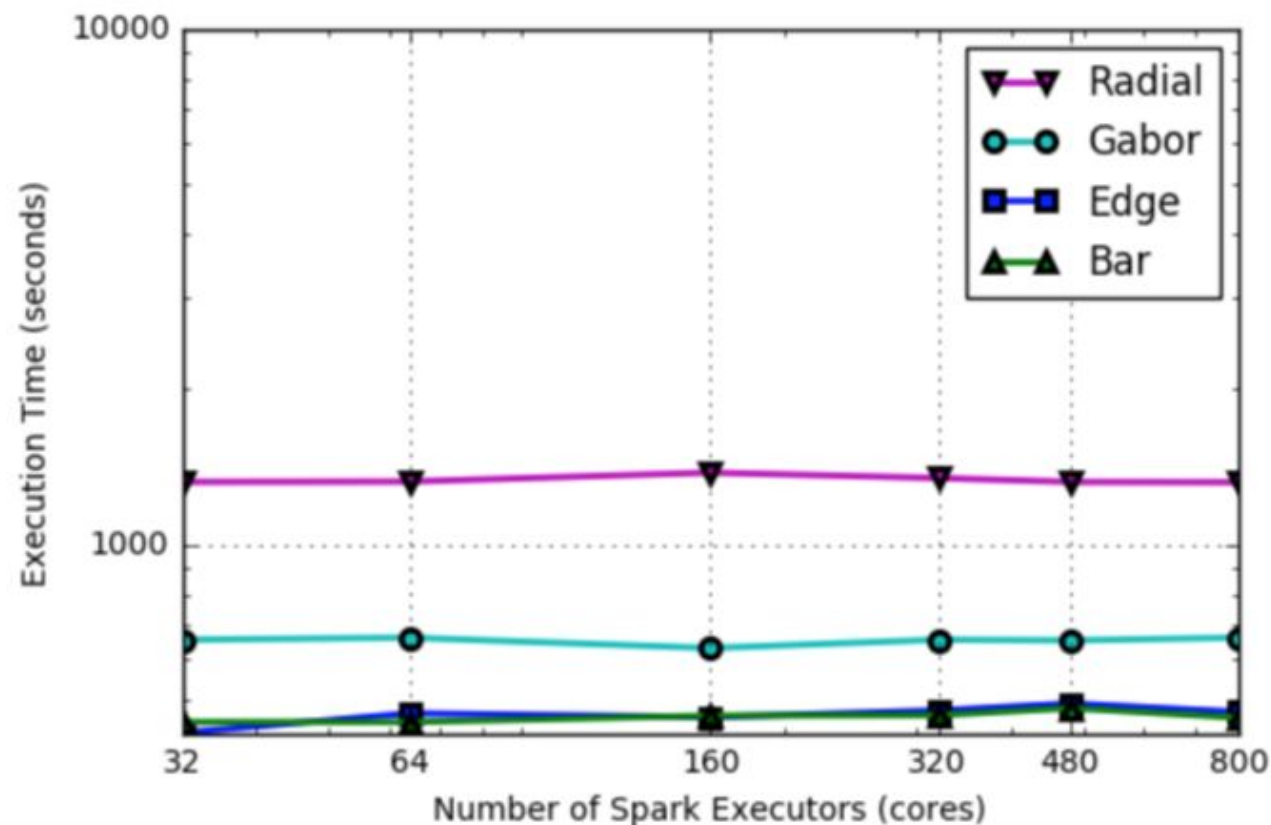


- Strong scaling performance for scales from 32 executors (cores) to 800 executors (cores) with Spark on 4800 images.
- For Gabor filters implementation on the Sage system, as we strong scale from 32 cores to 800 cores, our execution time decreases from 270 minutes to 11 minutes and we achieve a 98% improvement in performance
- On the Cooley System, as we strong scale from 32 cores to 800 cores, our execution time decreases from 510 minutes to 22 minutes and we achieve a 91% improvement in performance
- At 800 cores, Gabor filters execution time is 3 times faster than the Radial filters.

Weak Scaling Results on the for a modified implementation

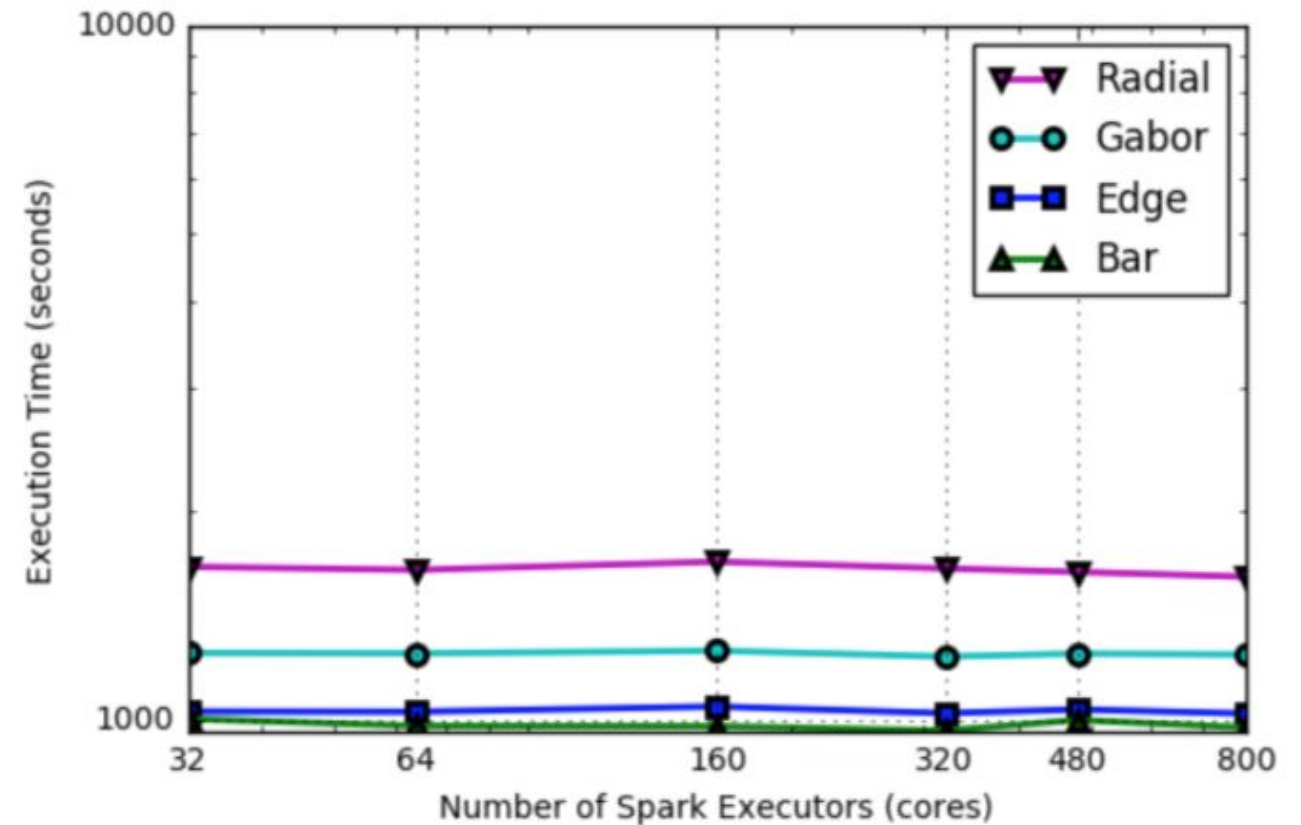
Sage Urika-GX System

Weak Scaling: 6 images per core
(~2.4GB)



Cooley System

Weak Scaling: 6 images per core
(~2.4GB)

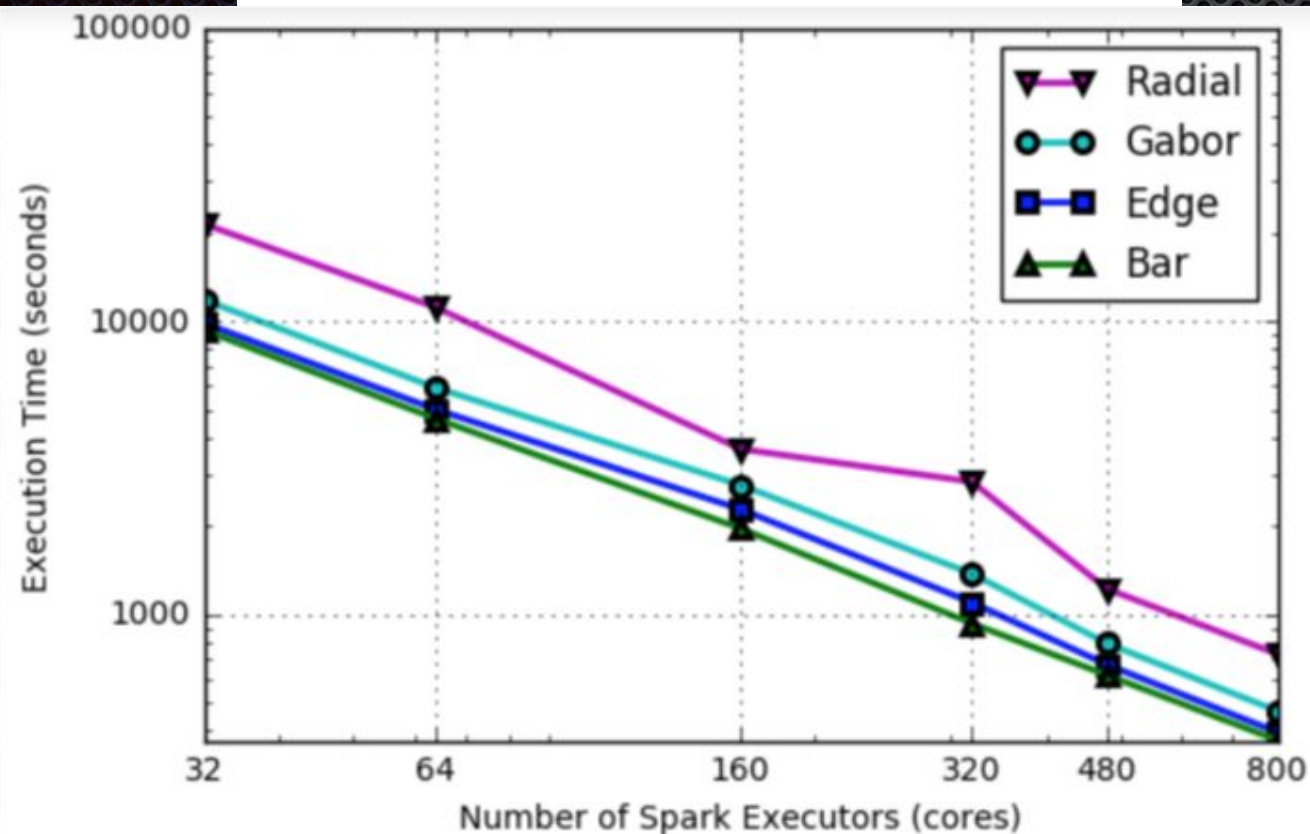


- Weak Scaling results are presented for roughly 6 images per core

Strong and Weak Scaling Results on the for a modified implementation

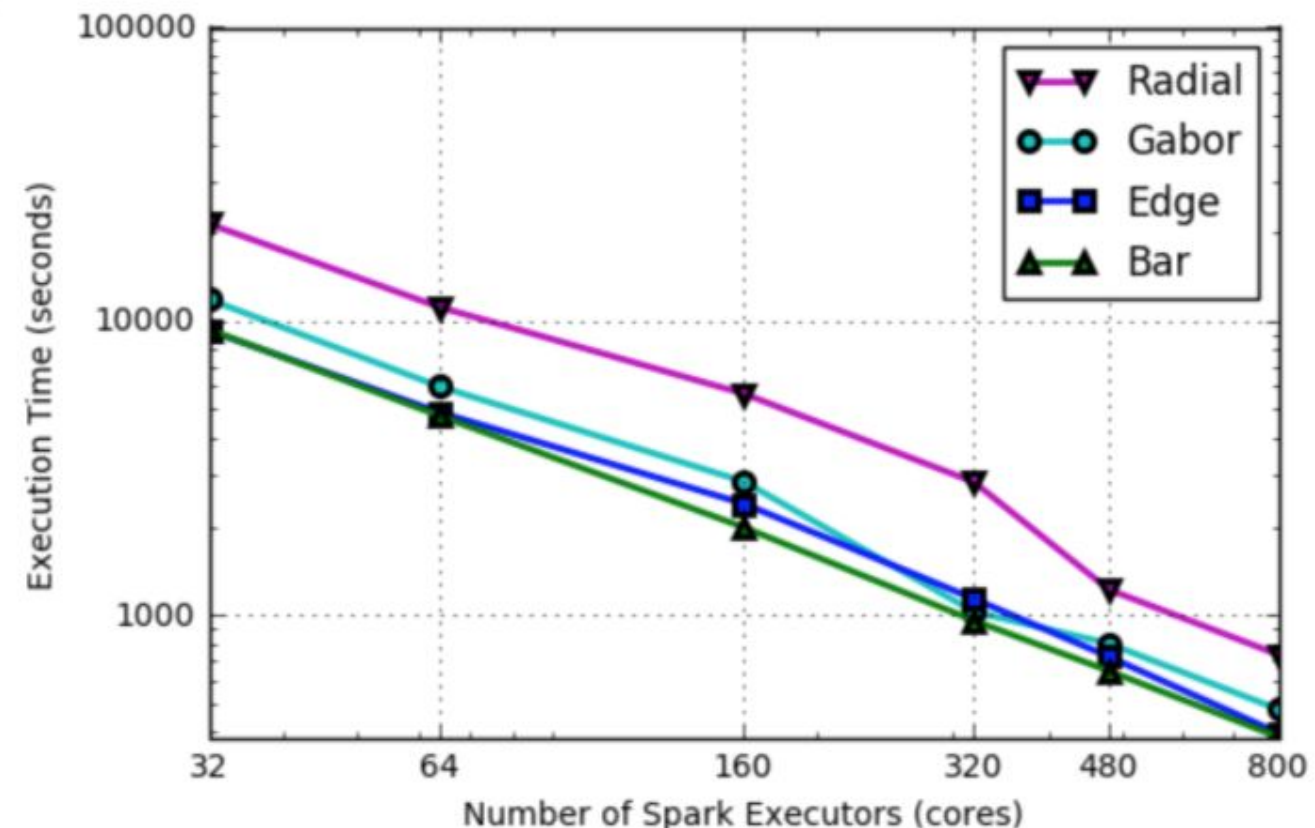
Sage Urika-GX System

Strong Scaling: 4800 Images (~2.4GB)



Cooley System

Strong Scaling: 4800 Images (~2.4GB)

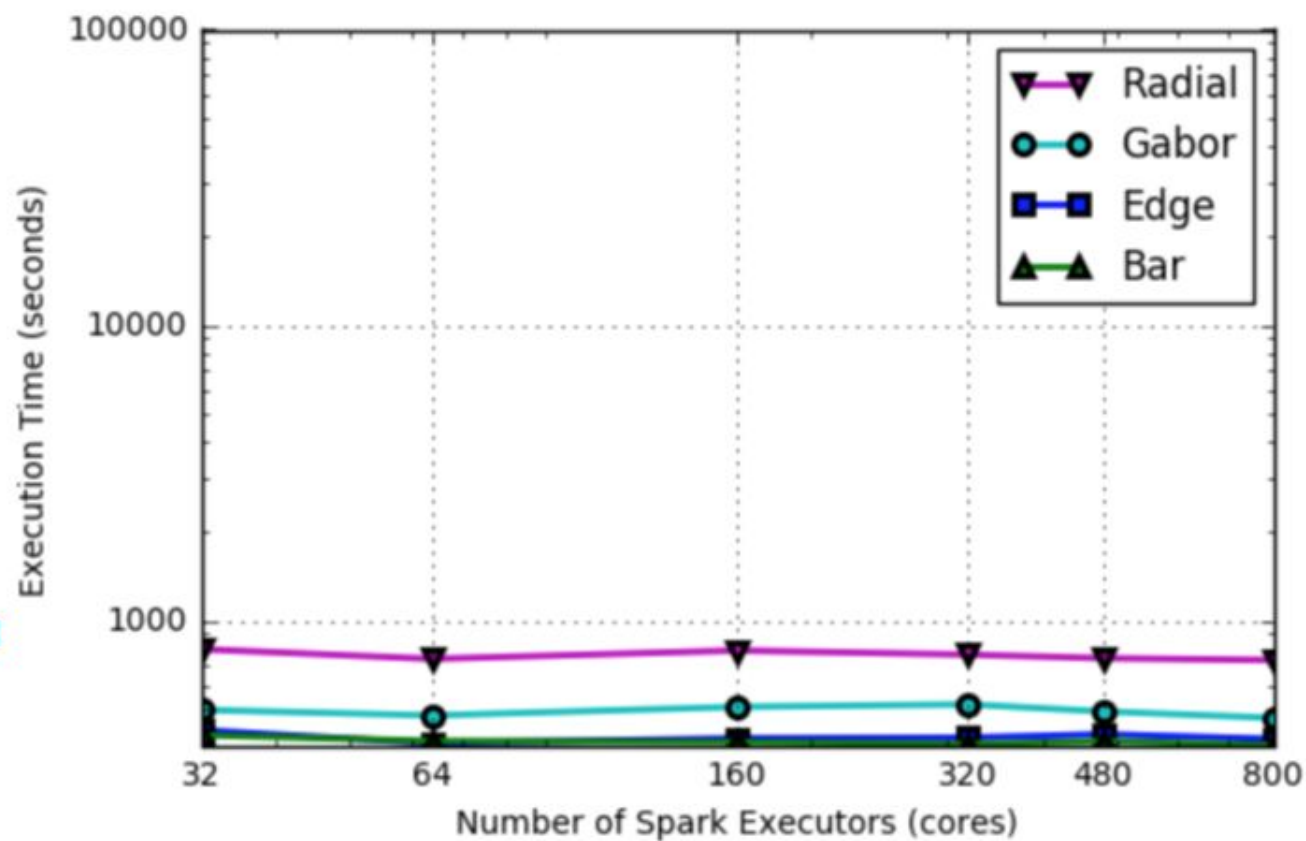


- Pre-processing and post processing of images are merged to form a single stage in the Spark DAG. This eliminates the Shuffle between stages and we get a 40% improvement in performance for Gabor filter bank
- Gabor filters implementation on the Sage system, as we strong scale from 32 cores to 800 cores, our execution time decreases from 196 minutes to 7.8 minutes and we achieve a 100% improvement in performance
- On the Cooley System, as we strong scale from 32 cores to 800 cores, our execution time decreases from 197 minutes to 8.1 minutes and we achieve a 97% improvement in performance
- At 800 cores, Gabor filters achieve a 49% improvement in execution time over the Radial filters.

Weak Scaling Results on the for a modified implementation

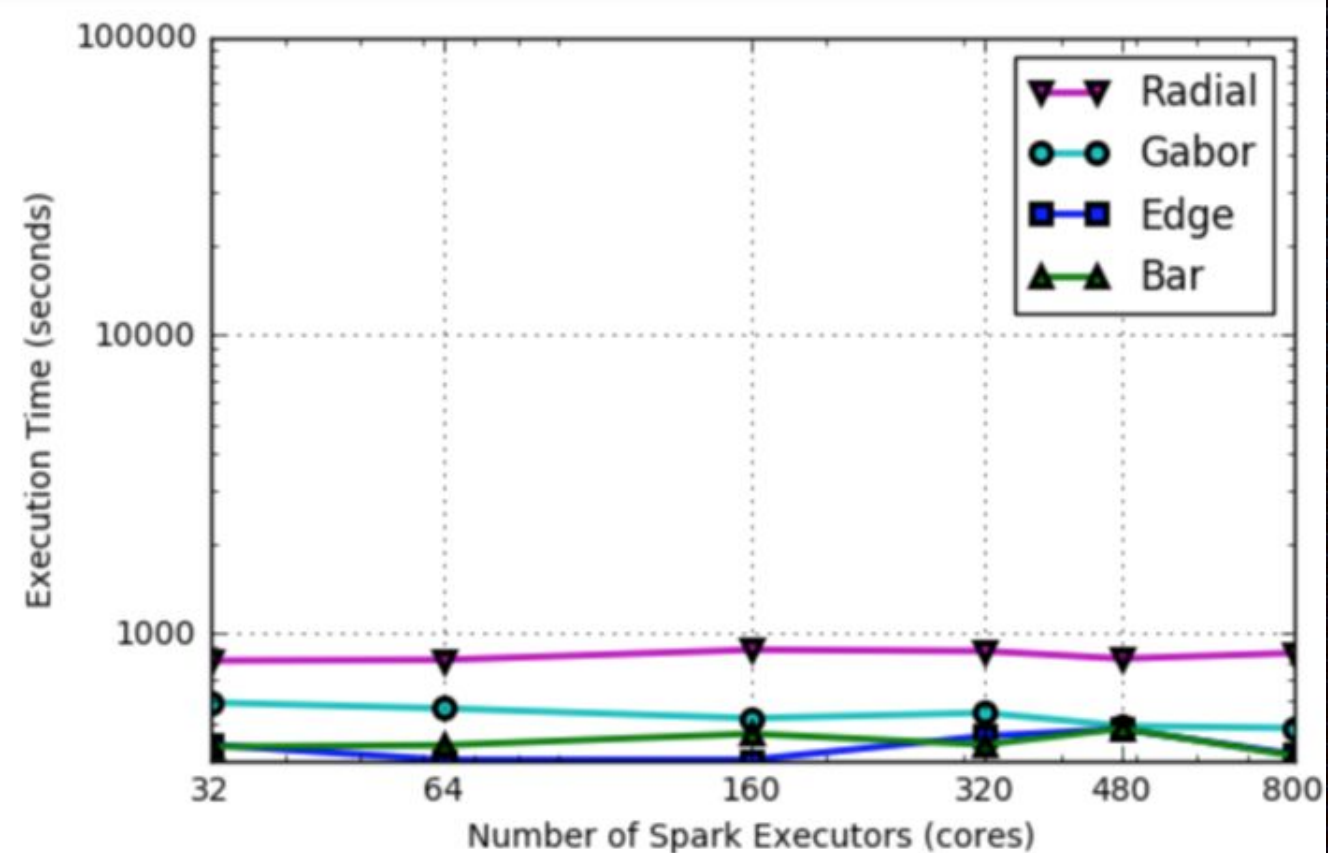
Sage Urika-GX System

Weak Scaling: 6 images per core
(~2.4GB)



Cooley System

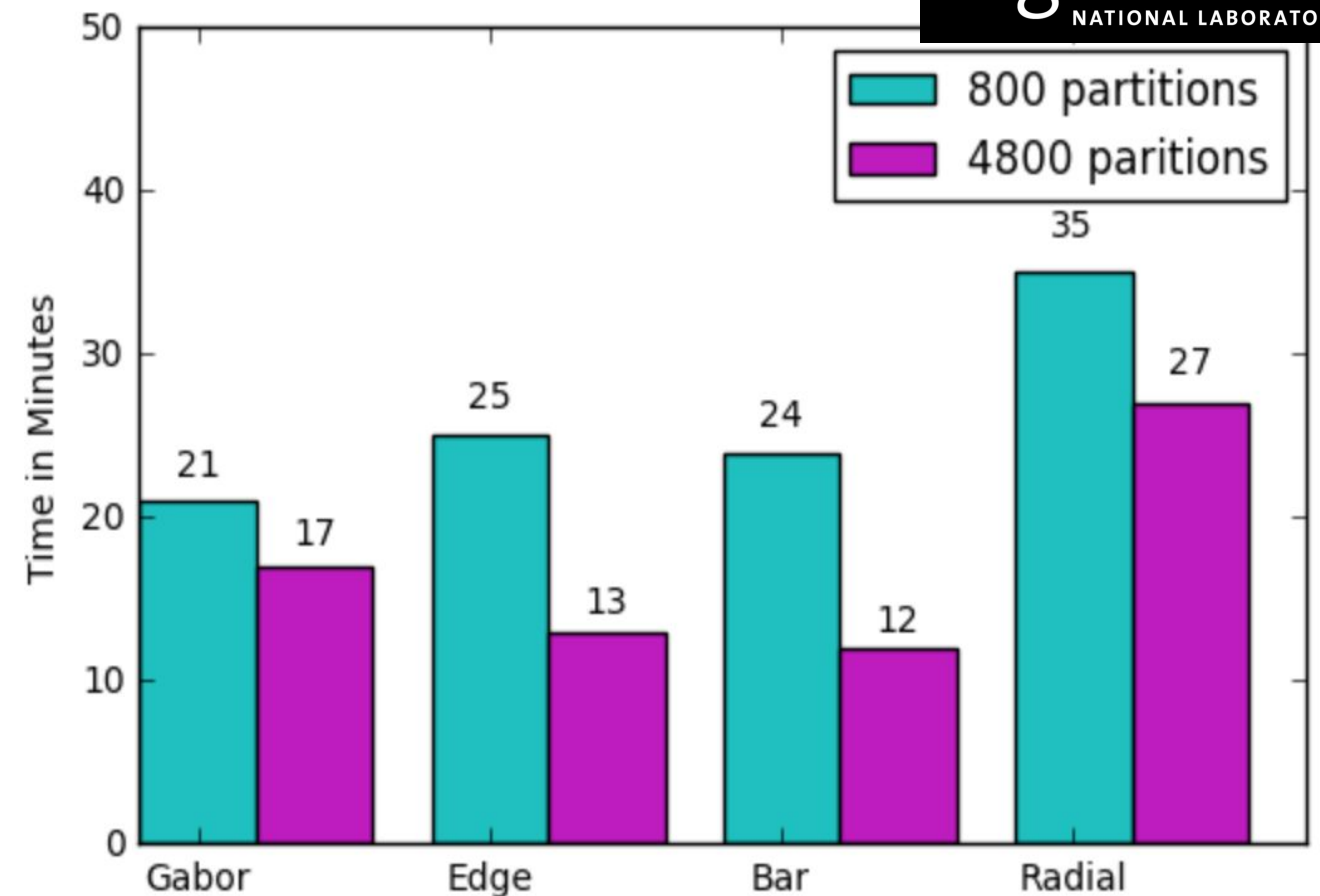
Weak Scaling: 6 images per core
(~2.4GB)



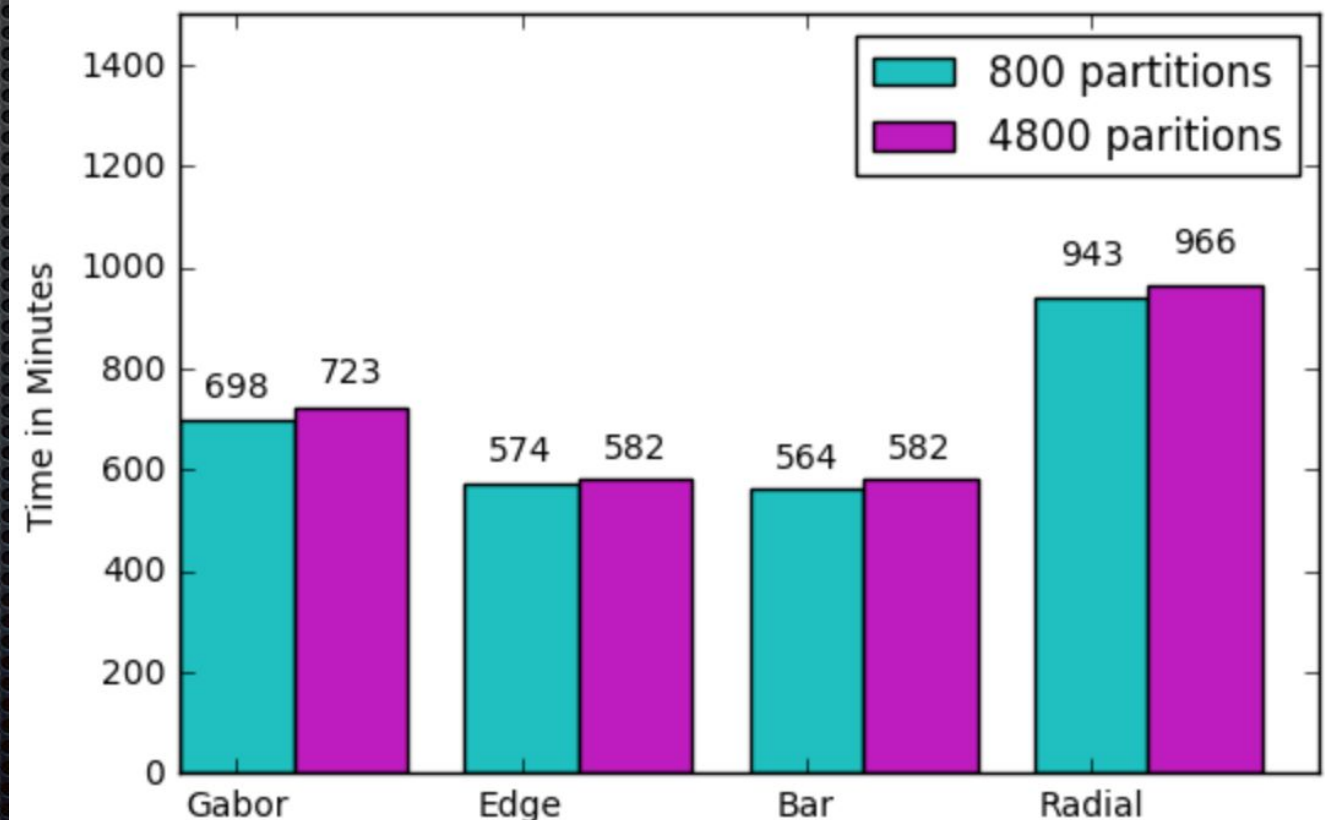
Choice of Data Partition

- Figures 1 and 2, show results of processing 12000 and 210,000 images on the Sage System which uses the mesos scheduler.
- The number of partitions here correlates to the number of tasks assigned by Spark that runs on each core of the executor.
- Processing 12,000 images on 800 partitions, i.e. 15 images per partition took more time when compared to processing 12,000 images with 4800 partitions i.e. 2 images per partitions.
- Tasks that are compute heavy take longer to start and hence we can achieve a performance improvement by using higher partition count for when the task size is smaller.
- Processing 210,000 images, degrades the performance for higher partition count since all tasks are heavier and take significantly longer to start.

12000 images (~2.5 GB)

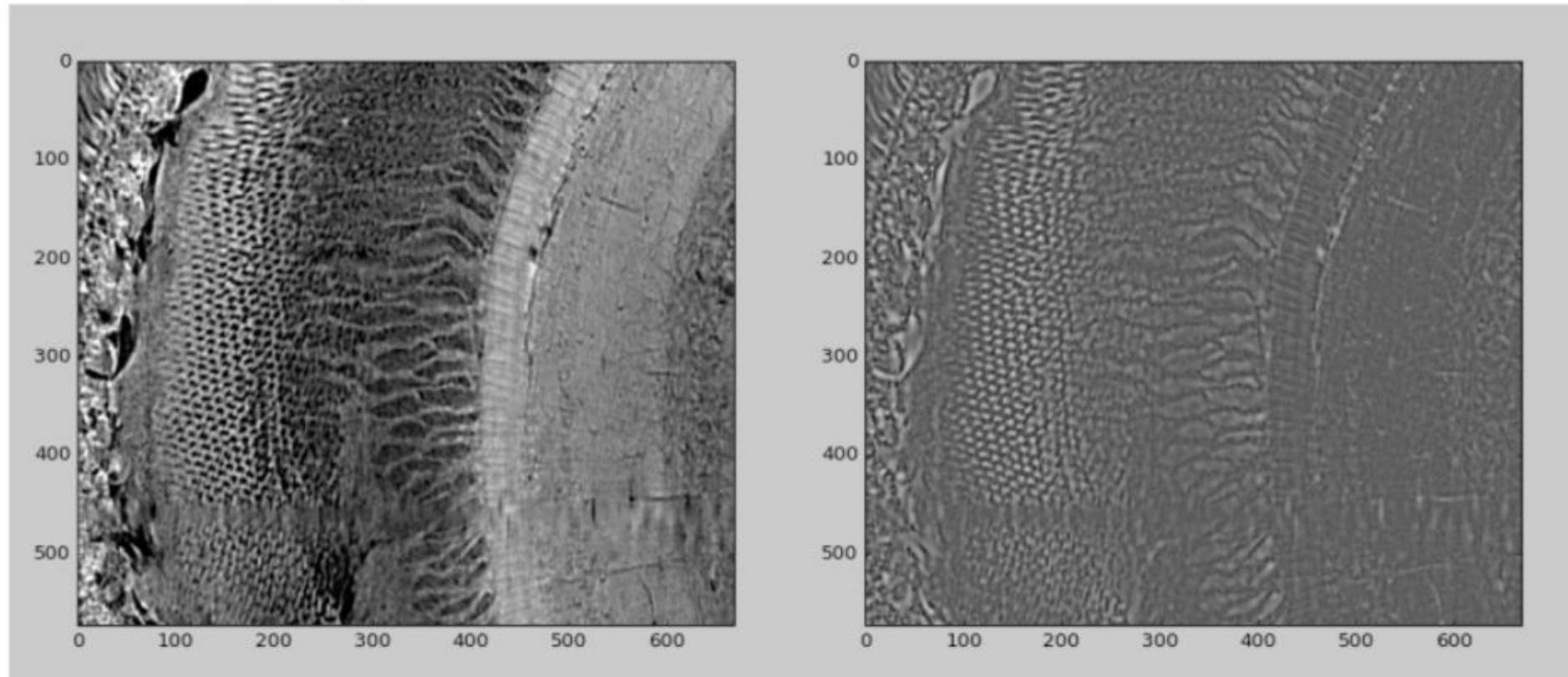


210,000 images (~100 GB)

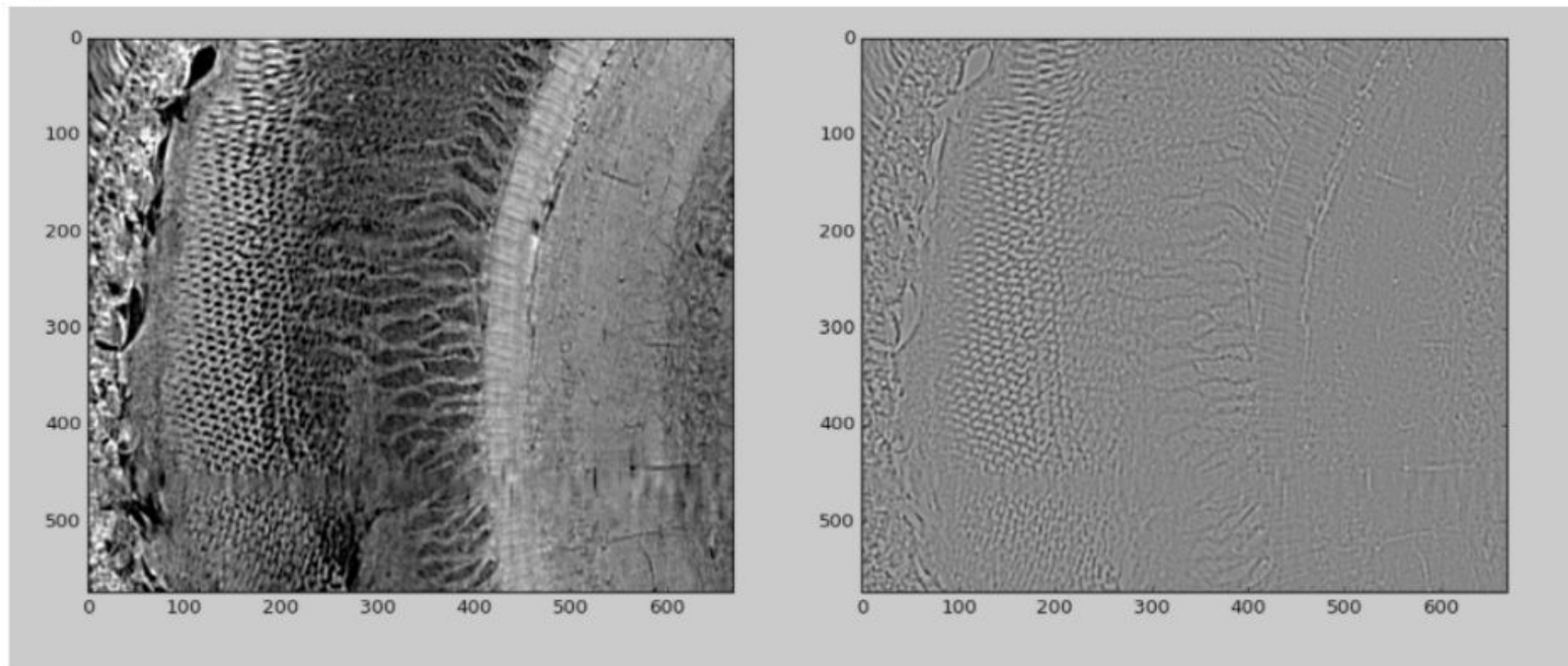


Filtering of Butterfly Brain Image

LoG on Butterfly image:



Sigma : 1.415



Sigma: 2.5

Future Work

- Extend our bank of filters to include more filters.
- This will provide the flexibility to use a filter based on the image dataset being processed.
- A combination of different types of filters on a single image to extract a combination of relevant features by targeting a region of interest.
- The filter responses can be used as input for various machine learning processing pipelines.

Conclusion

- We implemented a Bank of Filters for image analysis and evaluation of the Fuel Spray Data imaged at APS.
- A similar approach is applied for extracting features of interest on the brain datasets.
- We have scaled the analysis on to **800 cores of both Sage and Cooley systems**.
- Processing of 210,000 (~100GB) images took ~3 hours to process with ~54 seconds to process 1 image.
- This reduces the computation time by **700 times** when compared to processing the images sequentially as done in APS

Thank You