



Ain Shams University
Faculty of Computer and Information Science
Scientific Computing department

Project Title
Parallel Median Filter

By Team 31 SC

Name	ID	Section
Nour Mohamed Hussein Kamaly	20191700701	5
Nourhan Abdel-Karim Khalaf Abdel-Hafez	20191700716	5
Mohammed Nour-Elden Abbas Ismael	20191700583	4
Abdul-Rahman Sayed Ali Mohammed	20191700339	3

Under the supervision of

Dr. Heba Khaled
Scientific Computing Department,
Faculty of computer and Information Science
Ain Shams University

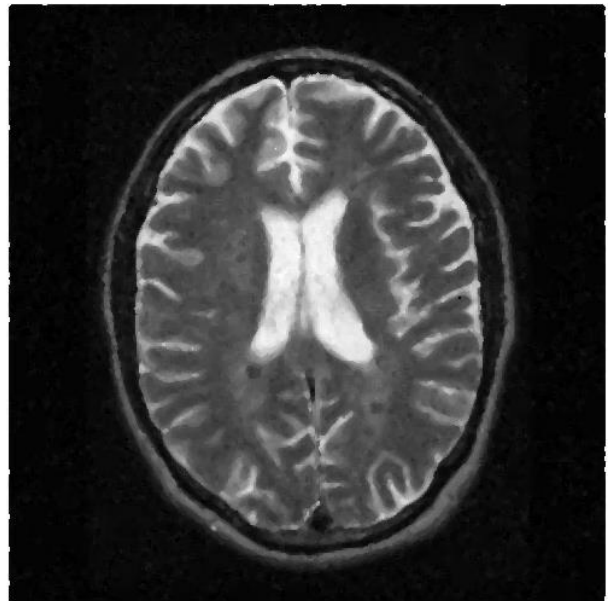
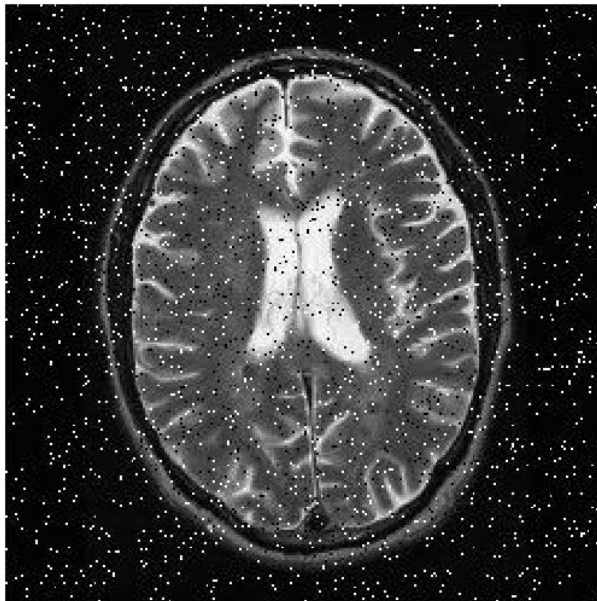
What is salt and pepper noise?

Salt-and-pepper noise, also known as impulse noise, is a form of noise sometimes seen on digital images. This noise can be caused by sharp and sudden disturbances in the image signal. It presents itself as sparsely occurring white and black pixels.

An effective noise reduction method for this type of noise is the median filter

Median filter

The median filter is a non-linear digital filtering technique, often used to remove noise from an image or signal. Such noise reduction is a typical pre-processing step to improve the results of later processing (for example, edge detection on an image). Median filtering is very widely used in digital image processing because, under certain conditions, it preserves edges while removing noise (but see the discussion below), also having applications in signal processing.



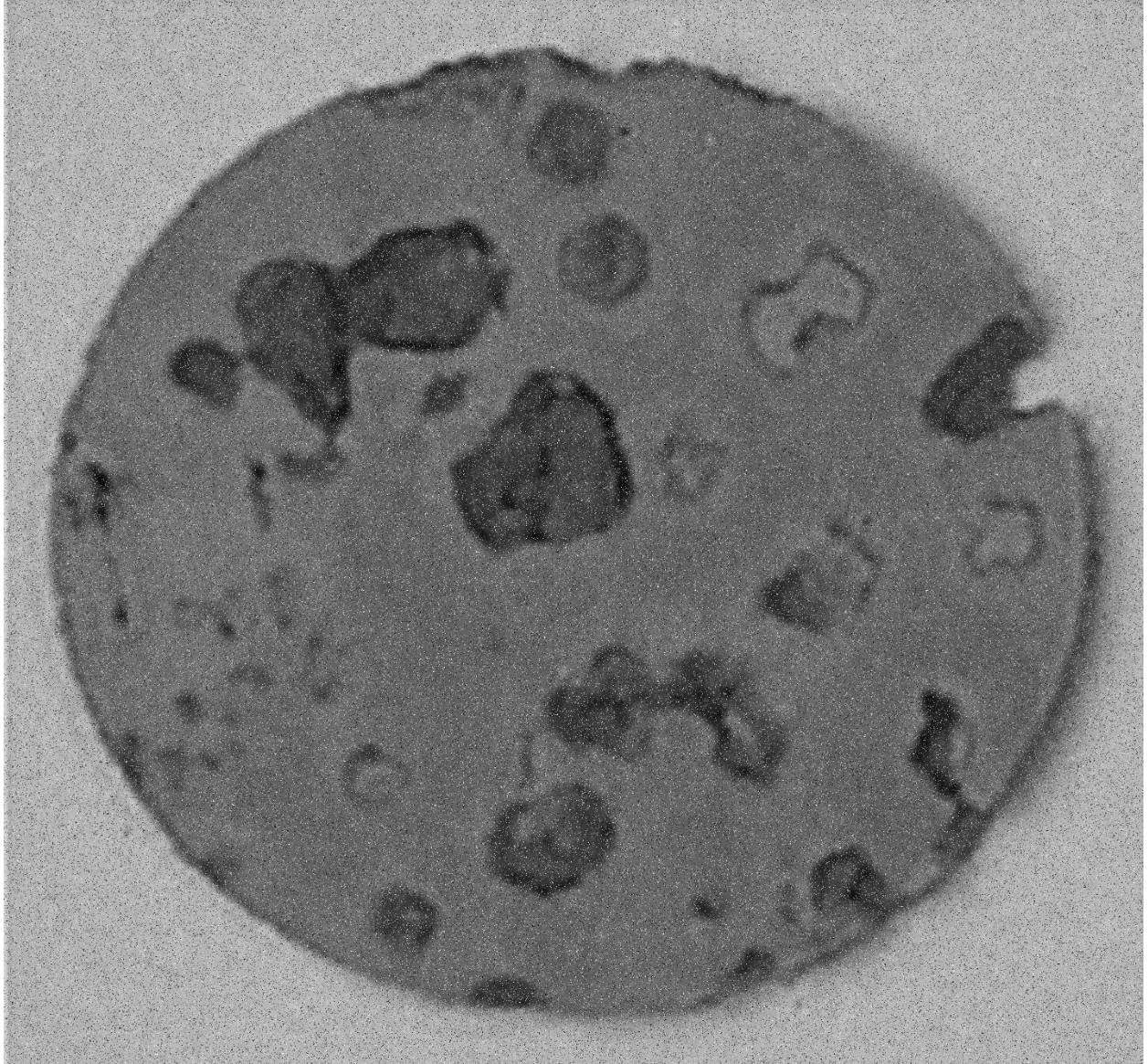
Enhancements and findings

- The time complexity of the median filter in sequential code is $O(\text{Width} * \text{Height} * \text{filterSize}^2)$. Width * Height to filter each pixel in the image and filterSize^2 as each pixel is filtered according to its (filterSize^2) neighbors.
- By using MPI (message passing interface), the image was distributed across N number of processors, each executing the code of the median filter, and the results were gathered at a specific core, to reconstruct the filtered image.
- As we increased the number of cores in parallel programming, the time taken used to decrease, until a threshold then it started to increase afterwards, this is due to the overhead on the processor to distribute the data and the tasks.
- The optimal number of processors to use is 8.

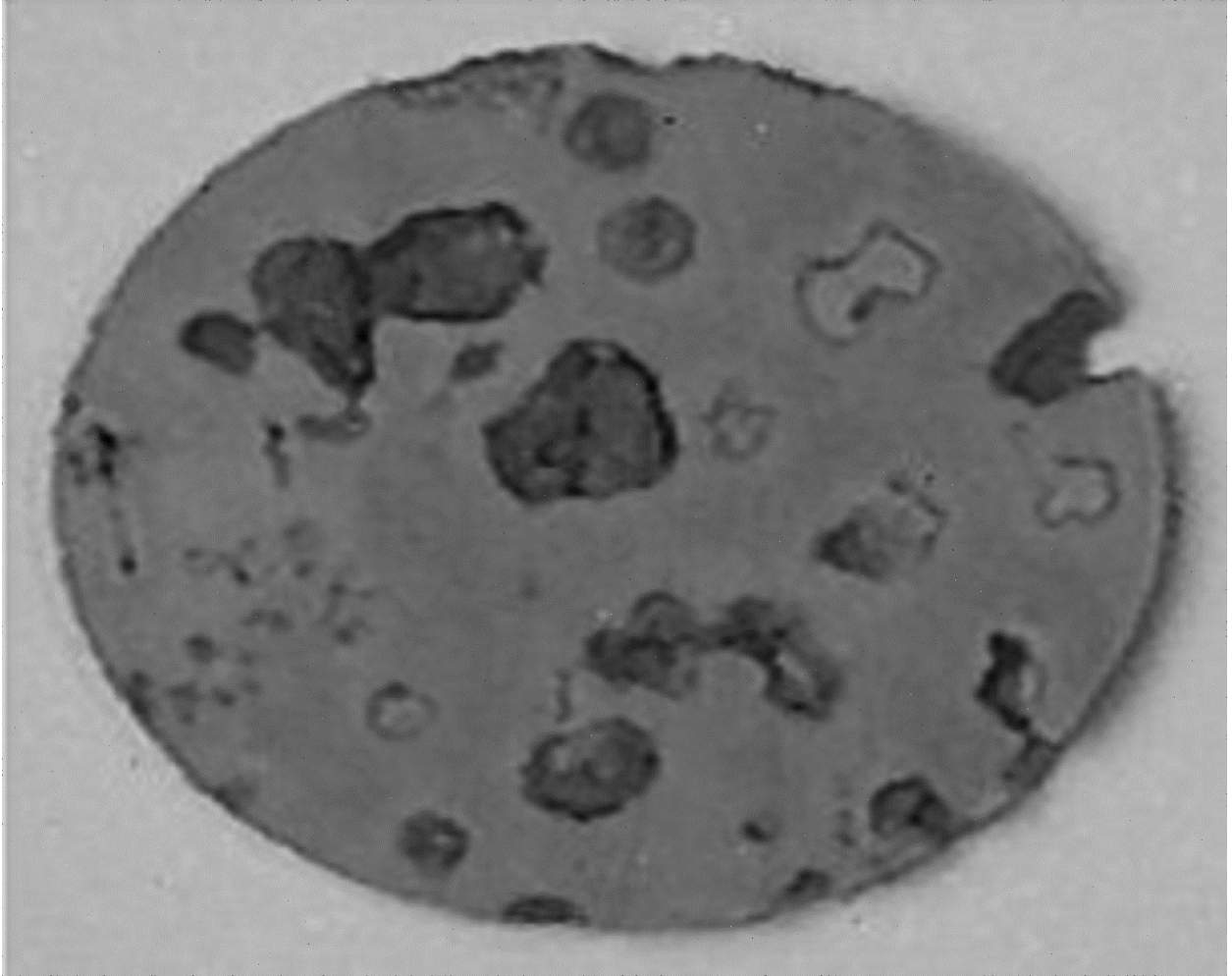
Results

	N*N	2N*2N	5N*5N	10N*10N
Sequential	61	197	1230	5133
MPI solution	23	78	323	1313

Unfiltered 10N*10N picture



Sequential filtering



Parallel filtering

