**Image Processing Homework 3**

**Group Partner: Abdullah Rihawi 2211011093**

**Submitted by: Abdul Malik 2211011098**

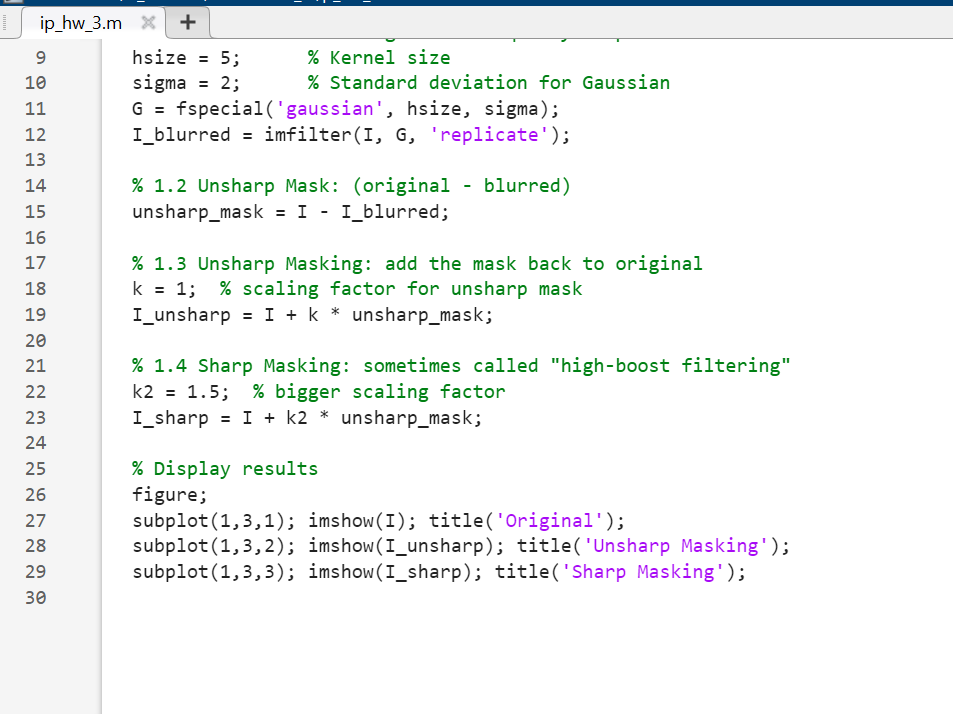
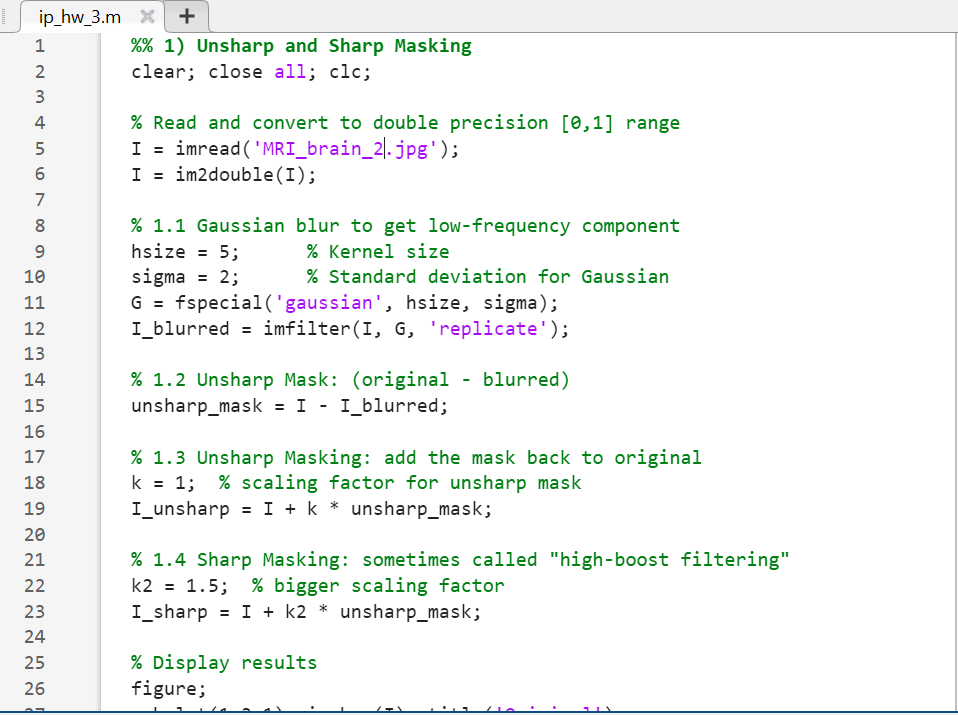
**1. Unsharp and Sharp Masking**

**Definition:**Unsharp masking is a technique that sharpens an image by subtracting a blurred version of it from the original. This emphasizes the edges by enhancing high-frequency components. Sharp masking, on the other hand, is a more aggressive version where the high-frequency component (often obtained via high-pass filtering) is added back to the image with a stronger weight.

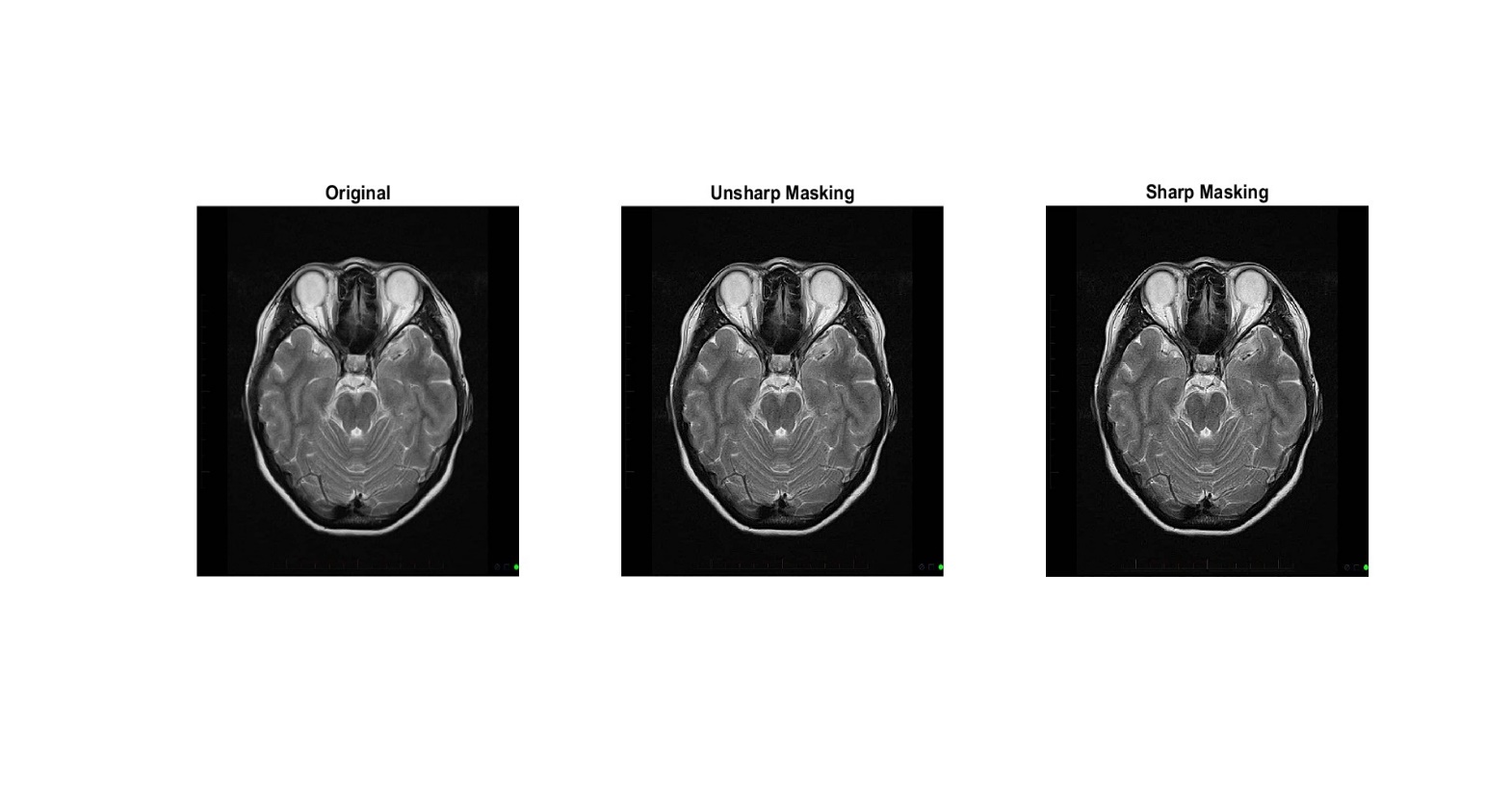
**Differences in Effects:**Unsharp masking subtly enhances image details and is commonly used for improving clarity without making the image look artificial. Sharp masking increases edge contrast more strongly, which can lead to more visually striking results but may also introduce noise or halos around edges.

**Example:**

Below is the MATLAB code *(Figure 1)* for applying Sharp and Unsharp masking on an MRI image of a human’s brain. Sharp and Unsharp masking has been applied to the original image and the results are shown *(Figure 2)*.



*Figure 1: MATLAB Code for Sharp and Unsharp Masking*

**

*Figure 2: Results after applying Sharp and Unsharp masking*

**Observations:**

As we can see in the figure above *(Figure 2)*, comparing both sharp and unsharp masking to the original image, in the Unsharp masking the details and edges are more prominent while in Sharp masking the details are highlighted at the cost of noise.

**Advantages & Limitations:**

* *Unsharp masking* is controlled and suitable for enhancing fine details with minimal noise amplification. However, its effect can be mild.
* *Sharp masking* provides stronger enhancement, but at the cost of increased noise and possible artifacts.

**2. Laplacian of Gaussian (LoG) and Difference of Gaussian (DoG) Filters**

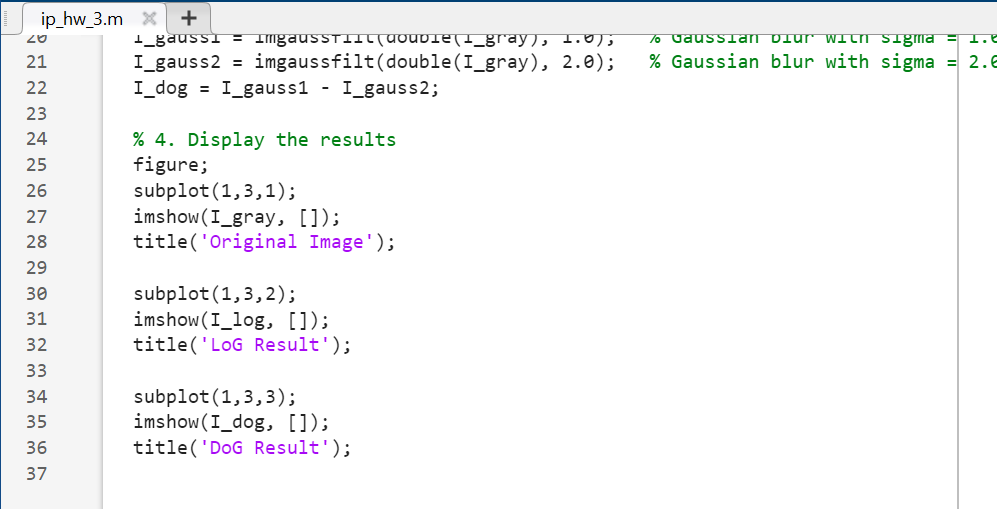
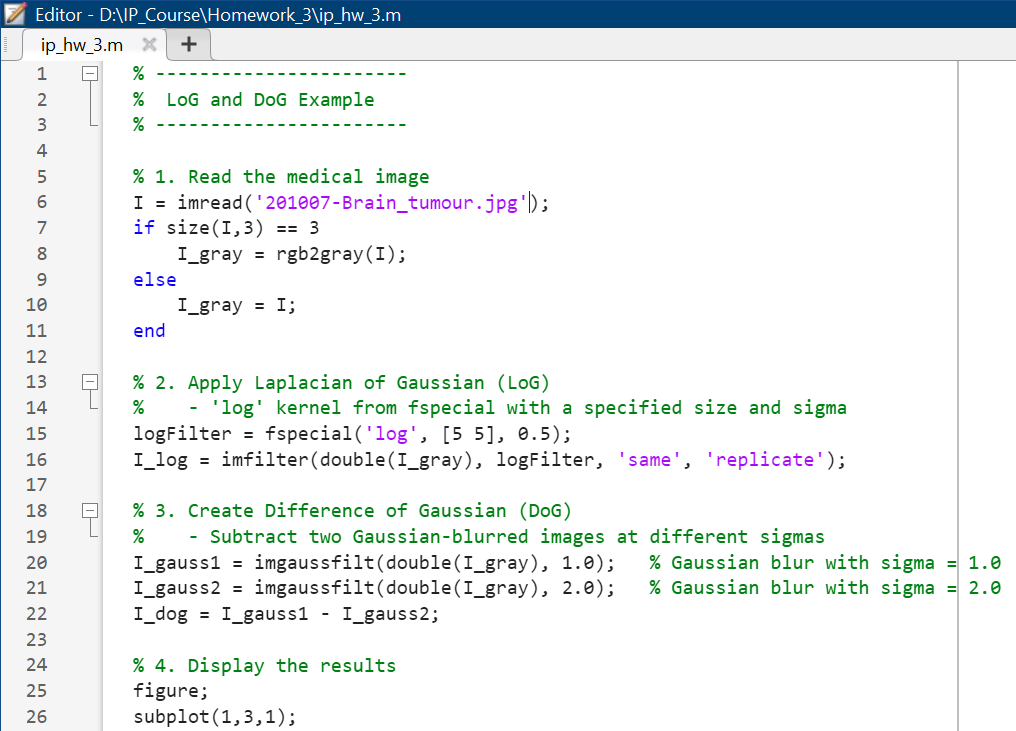
**Concepts:**

* The **Laplacian of Gaussian (LoG)** first smooths the image using a Gaussian filter to reduce noise, then applies the Laplacian operator to detect edges.
* The **Difference of Gaussians (DoG)** approximates LoG by subtracting two images blurred with different Gaussian kernels.

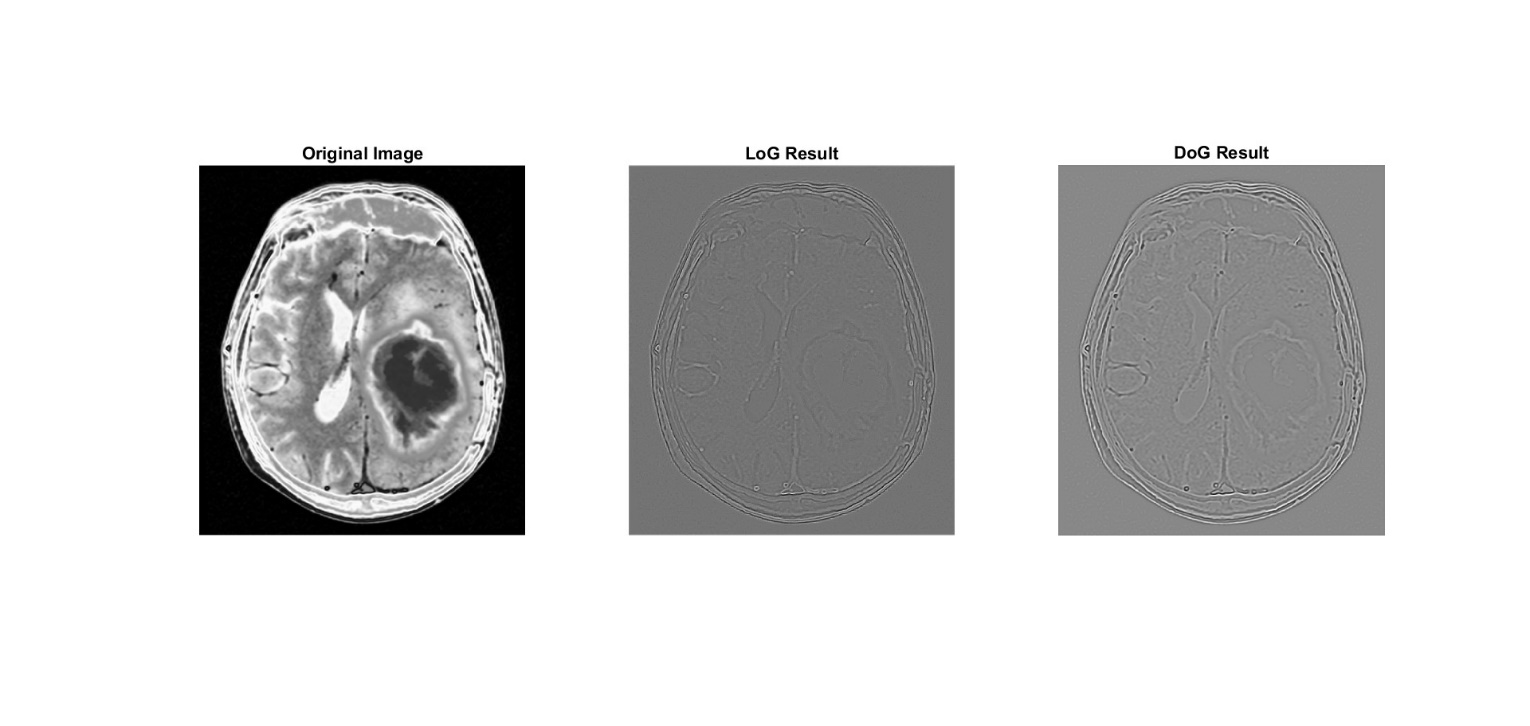
**Comparison of Characteristics:**

* LoG is more precise and better at detecting fine edge details but computationally more intensive.
* DoG is faster, simpler to compute, and works well for general-purpose edge detection, although it might miss some finer structures.

**Example:**  
When applied to an image, LoG might help detect subtle fractures, while DoG can rapidly highlight major bone outlines or organ boundaries. Below is the MATLAB code *(Figure 3)* in which a colored Brain Tumor image of a human’s brain is taken, converted to gray level image and then LoG and DoG filters were applied on it.



*Figure 3: MATLAB Code for LoG and DoG*

**

*Figure 4: Results after applying LoG and DoG*

**Observations:**

In the image *(Figure 4)* above, we can observe after comparing each filtered image to our original image, that the LoG gives noticeable edges which are more essential in terms of featuring extraction for further process while DoG gives overall structure of the image with edges but we are losing some useful information like sharp edges and structure.

**Use Cases:**

* Choose **LoG** when precise edge localization is crucial, such as in tumor margin detection.
* Choose **DoG** in real-time systems where speed is essential, like live ultrasound imaging.

**3. Mean Filters**

**Definitions and Formulas:**

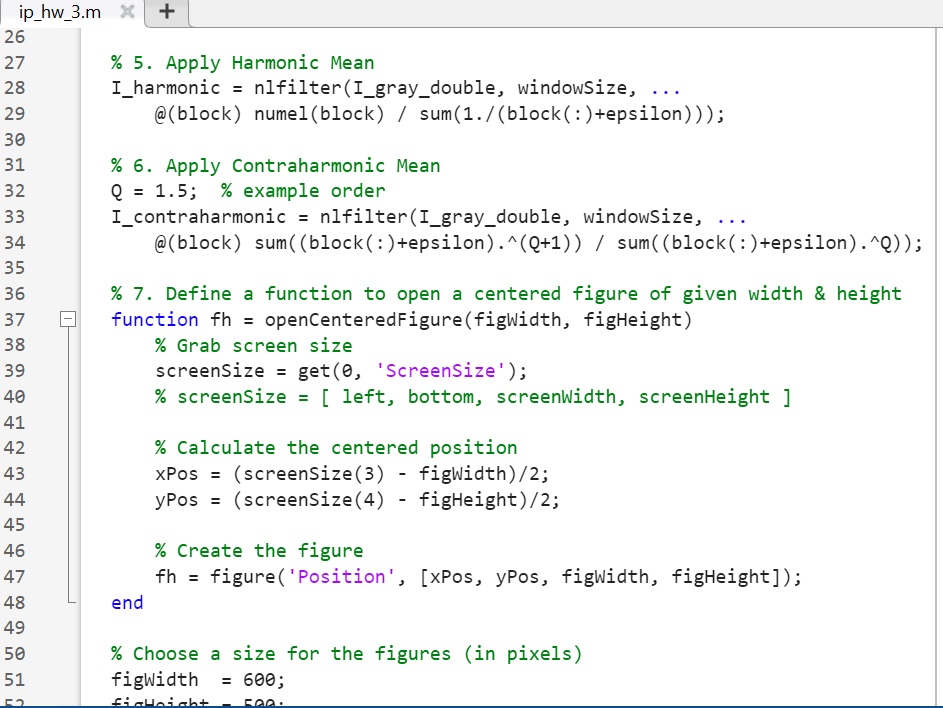
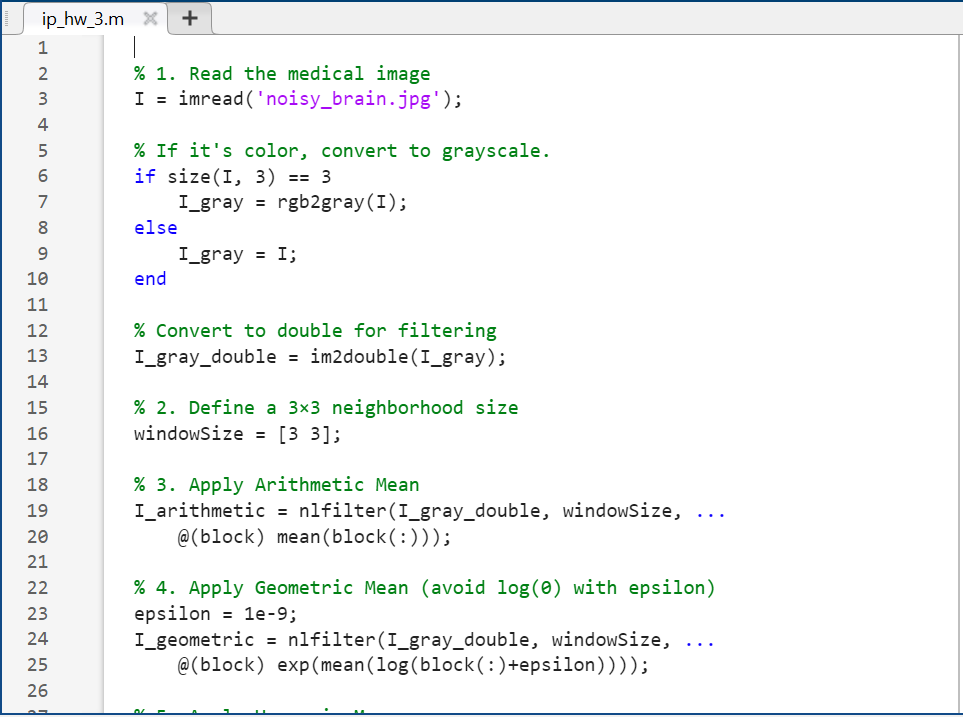
* Arithmetic Mean Filter: Averages the pixel values within a neighborhood:
* Geometric Mean Filter: Takes the nth root of the product of pixel values:
* Harmonic Mean Filter: Inverse of the average of inverses:
* Contraharmonic Mean Filter:

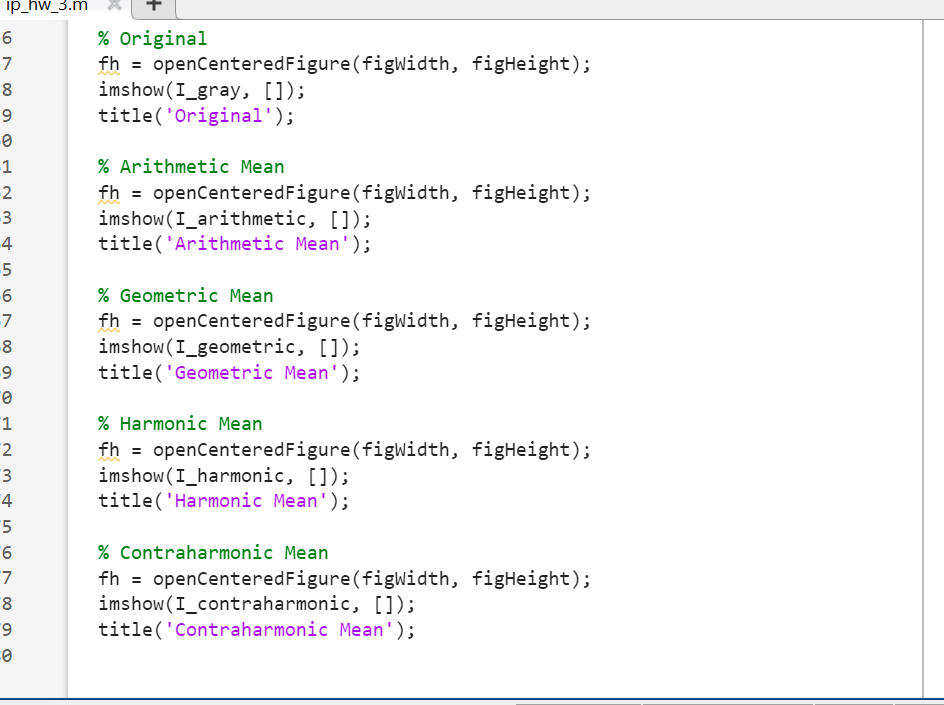
where Q determines whether salt (Q < 0) or pepper (Q > 0) noise is removed.

**Examples:**

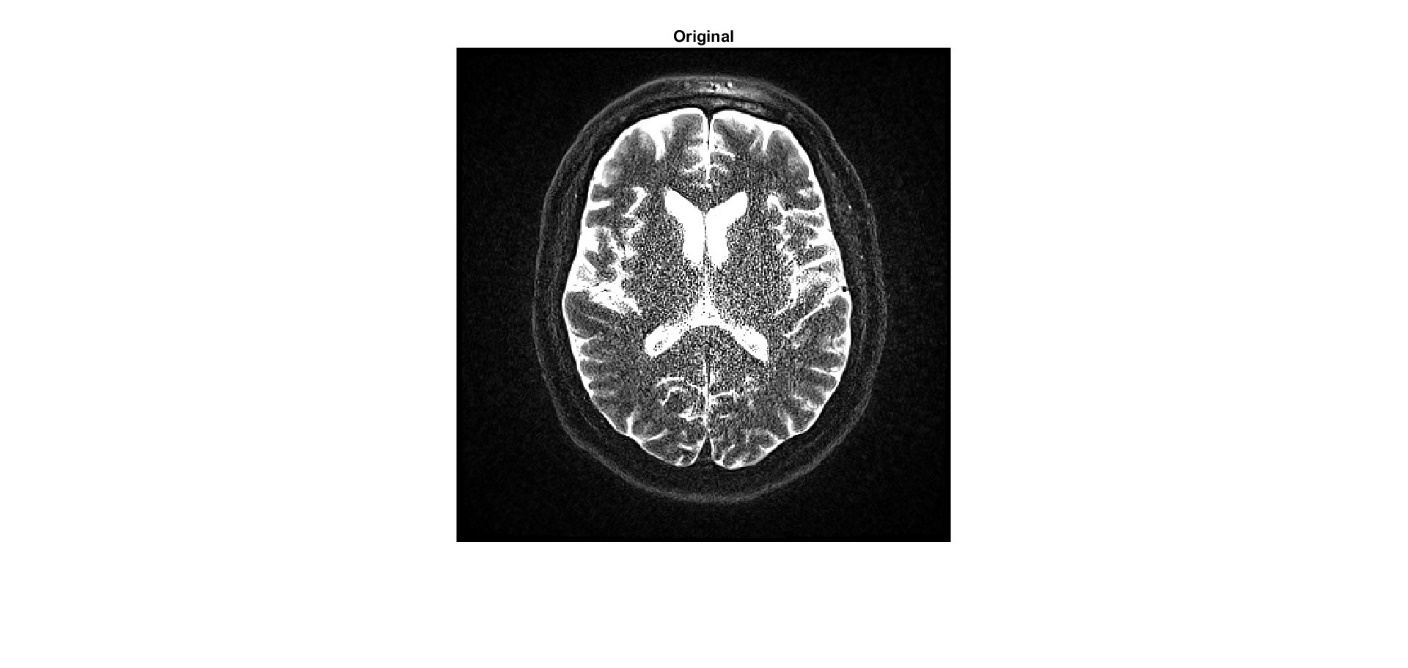
* Arithmetic Mean can smooth a noisy MRI but may blur edges.
* Geometric Mean is better for multiplicative noise, e.g., speckle noise in ultrasound.
* Harmonic Mean is great for reducing salt noise.
* Contraharmonic is adjustable based on the noise type.

Below is the MATLAB code *(Figure 5)* in which each four of these filters are applied to a noisy brain MRI image.

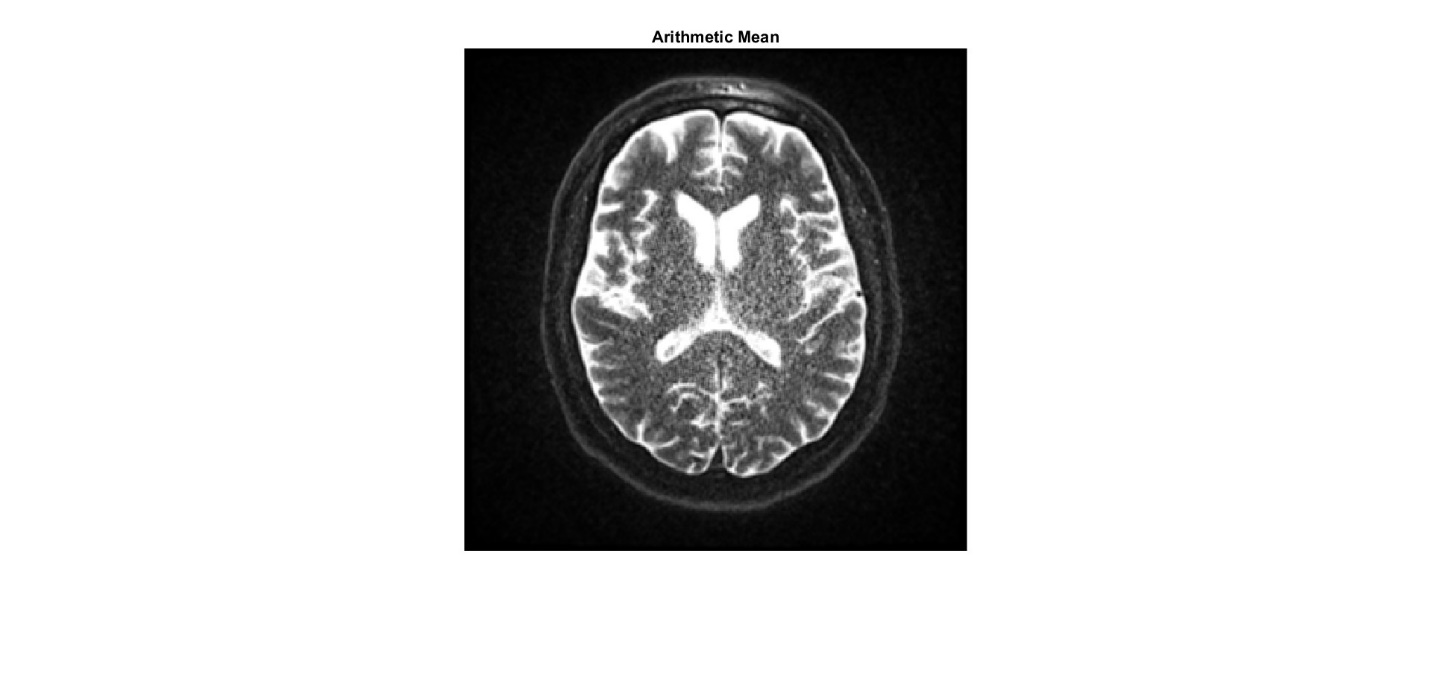




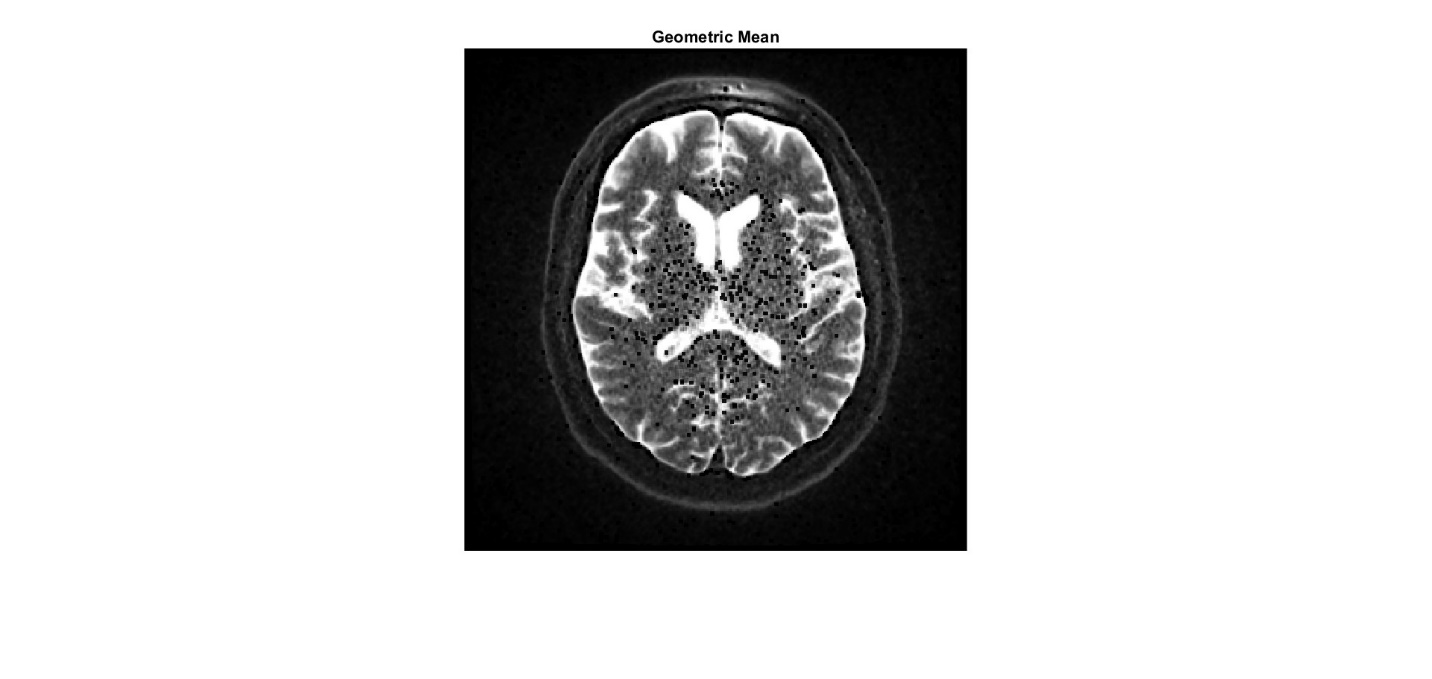
*Figure 5: MATLAB code for Arithmetic Mean, Geometric Mean, Harmonic Mean and Contraharmonic filters*

**

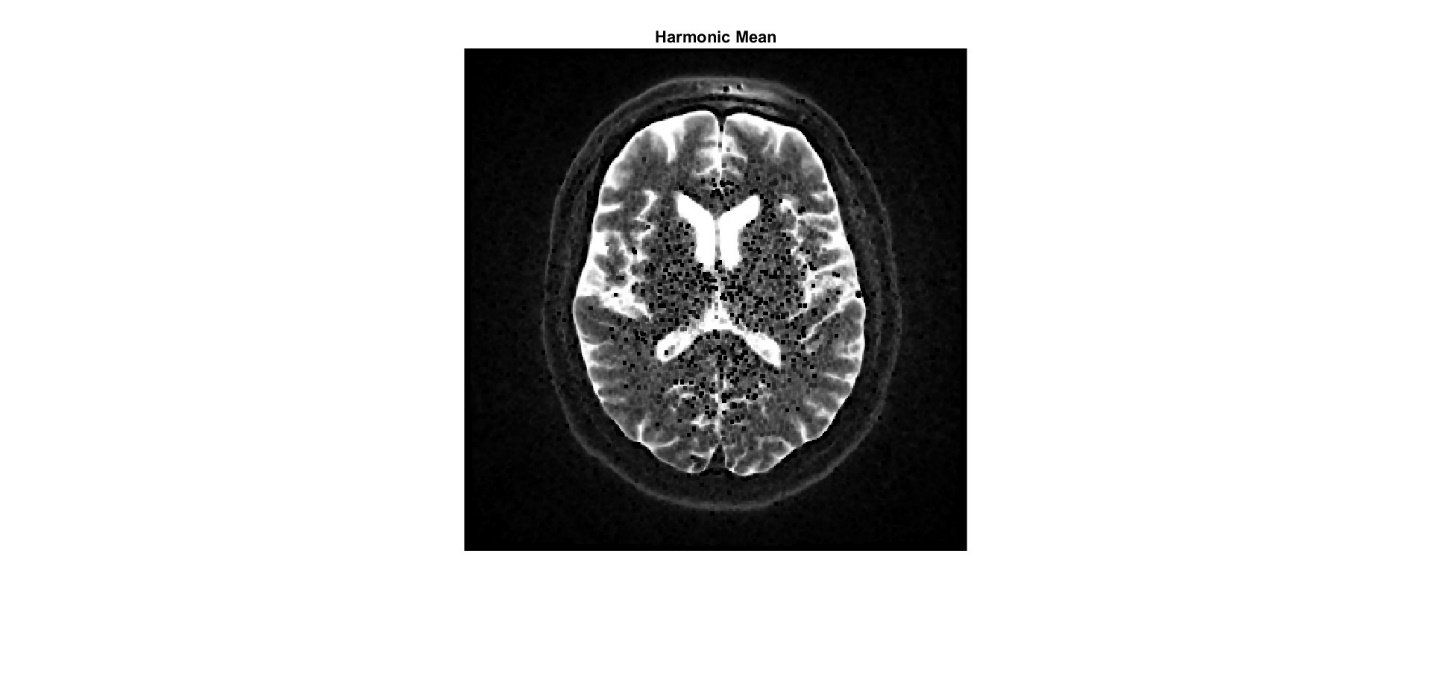
*Figure 6: Original Image*

**

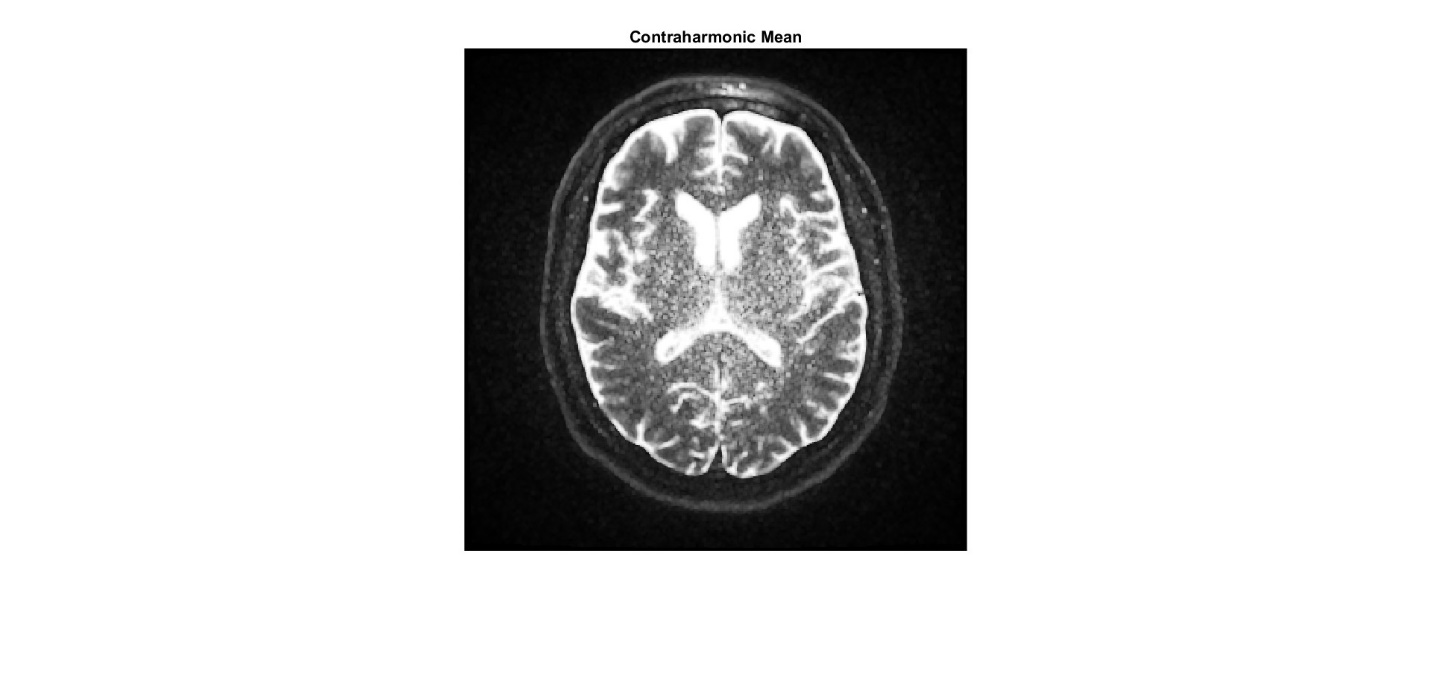
*Figure 7: Arithmetic Mean Filter Image*

**

*Figure 8: Geometric Mean Filter Image*

**

*Figure 9: Harmonic Mean Filter Image*

**

*Figure 10: Contraharmonic Filter Image*

**Observations:**

Above shown *(Figure 7 – Figure 10)* images are the results after applying Arithmetic Mean, Geometric Mean, Harmonic Mean and Contraharmonic filters respectively to our original noisy brain image *(Figure 6).* We can observe that Arithmetic mean filter smooths the noise, geometric mean filter is keeping the details same while keeping the smoothness and so does the harmonic mean filter. On the other hand, Contraharmonic filter is making the image more undesirable and distorted because in our application we selected the value of Q a bit less suitable. Tuning the Q value can give us desirable results.

**Performance Comparison:**

* Arithmetic mean is good for light smoothing.
* Geometric and harmonic filters preserve more structure while filtering.
* Contraharmonic is powerful when tuned well but can distort images if Q is poorly chosen.

**4. Order Statistic Filters**

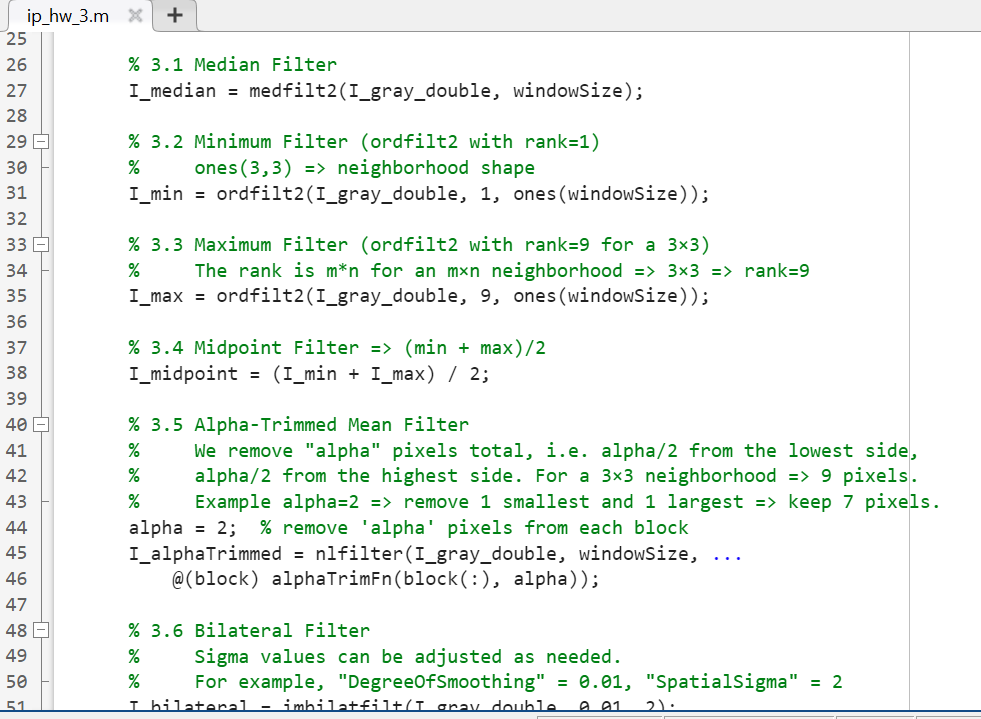
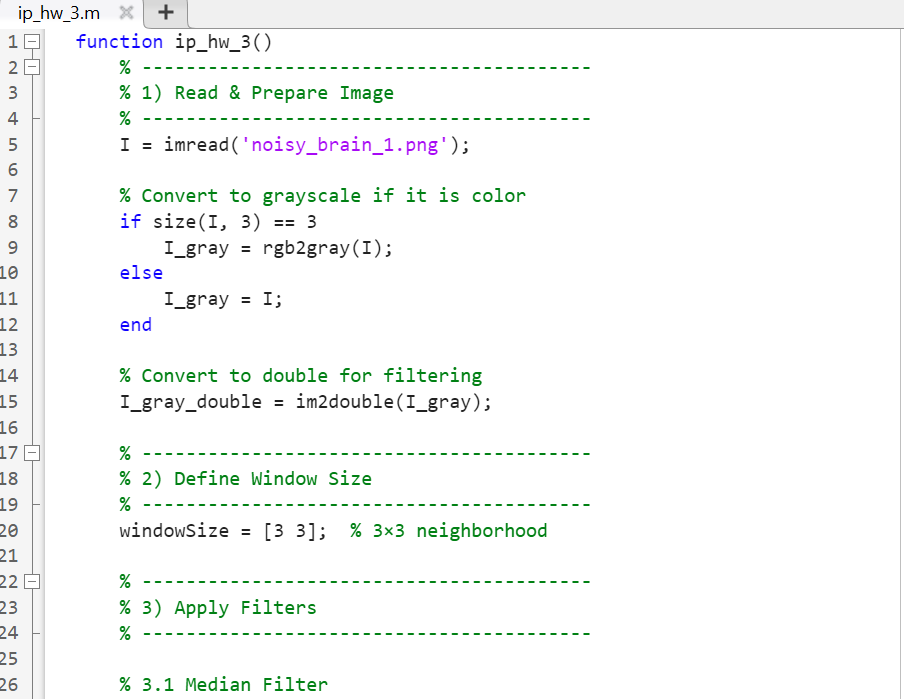
**Definitions:**

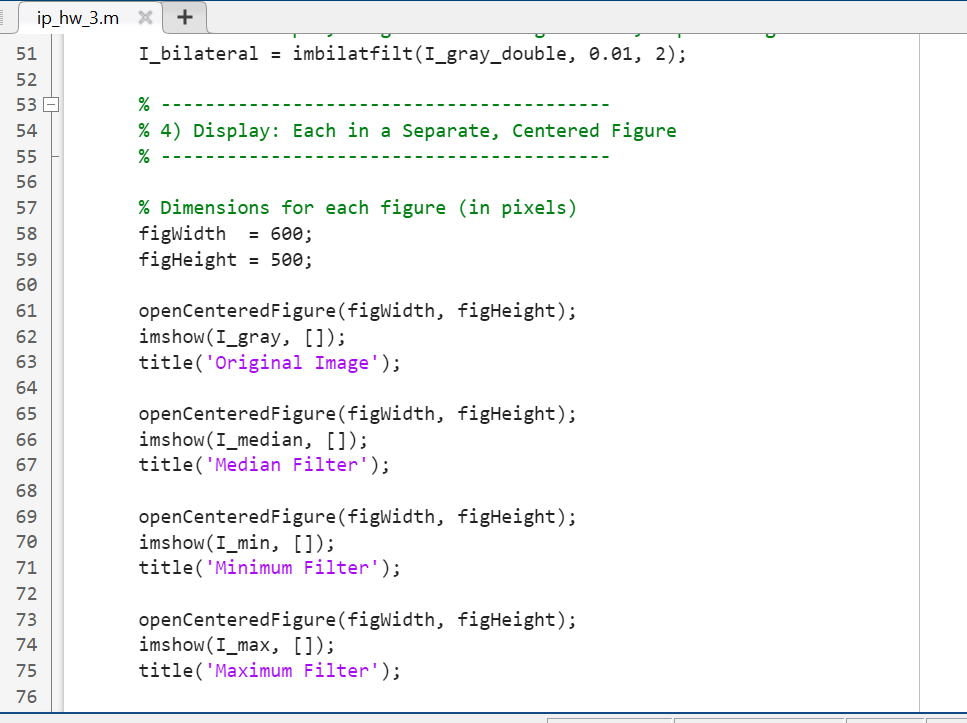
* Median Filter: Replaces a pixel with the median value in its neighborhood.
* Minimum Filter: Takes the smallest pixel value.
* Maximum Filter: Takes the largest pixel value.
* Midpoint Filter: Averages the minimum and maximum:
* Alpha-Trimmed Mean Filter: Removes a percentage of highest and lowest values and averages the rest.
* Bilateral Filter: Weights pixels based on both spatial distance and intensity similarity to preserve edges.

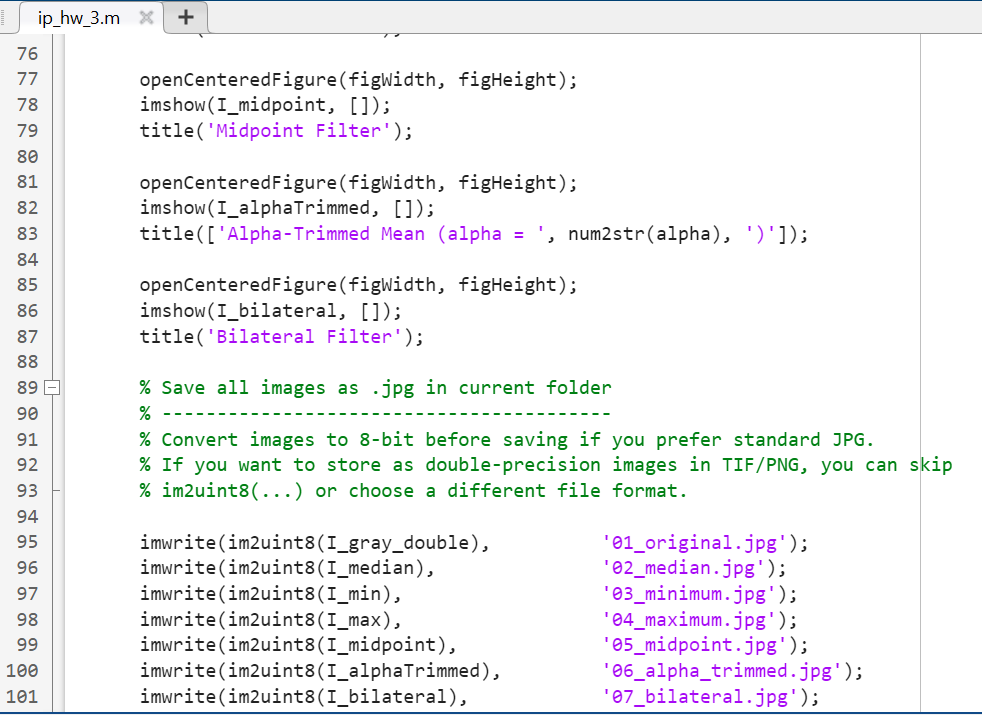
**Concepts:**These filters are non-linear and good at removing impulsive noise (like salt-and-pepper), unlike mean filters which can blur details.

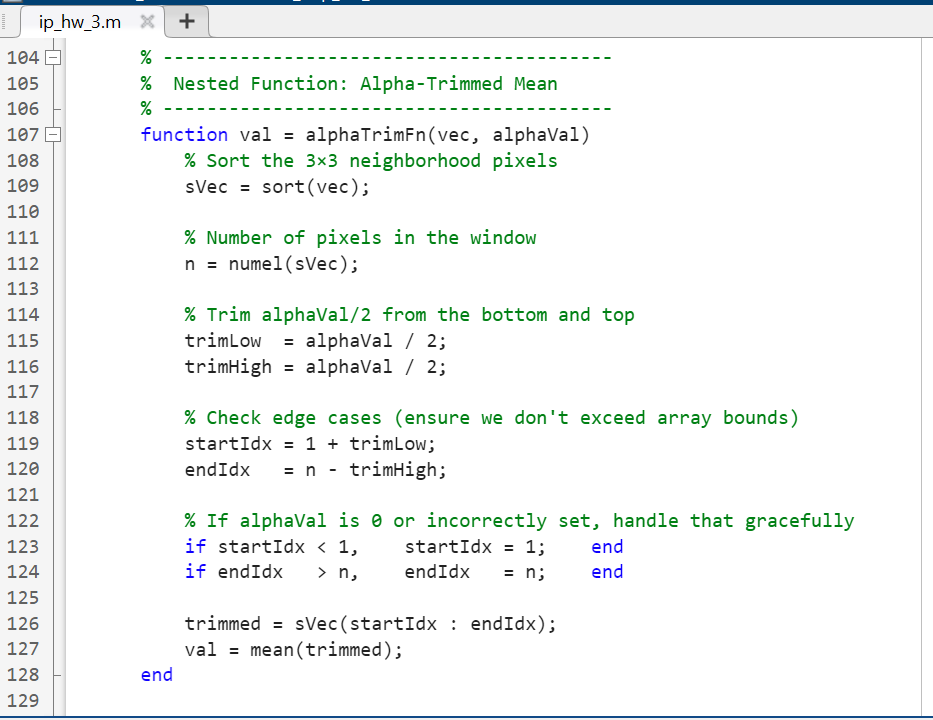
**Examples:**

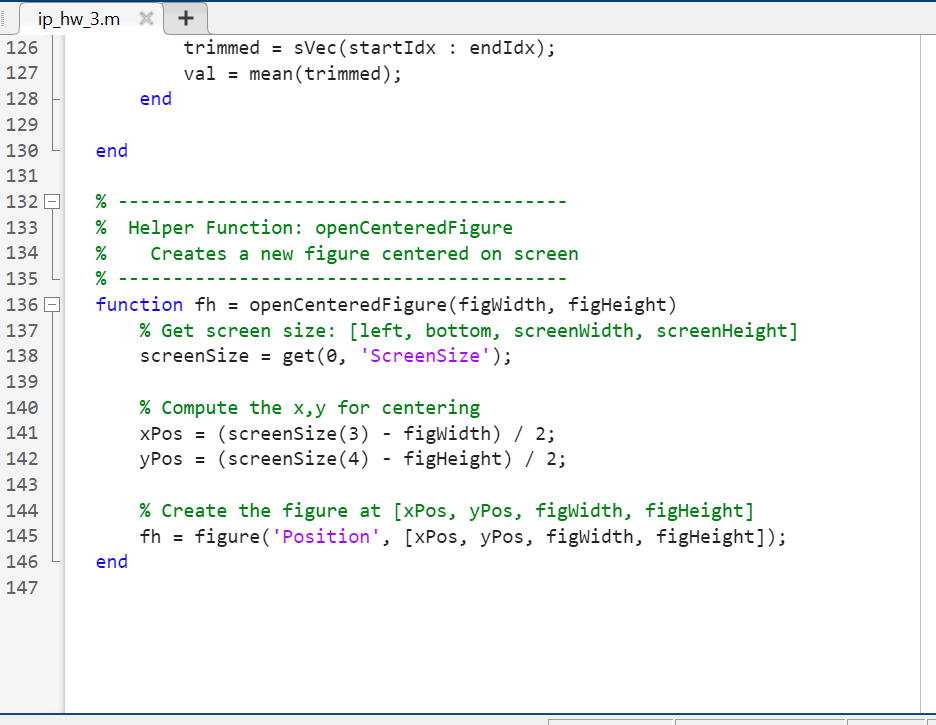
Below is the MATLAB code to perform the operations on an MRI image of a brain and making filtered images using the order statistics filters.



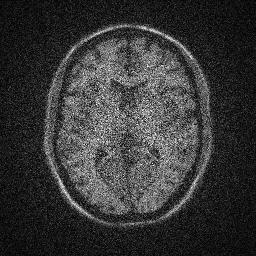




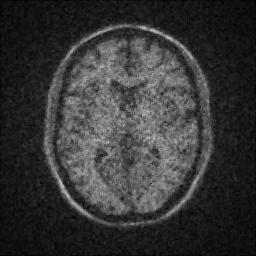




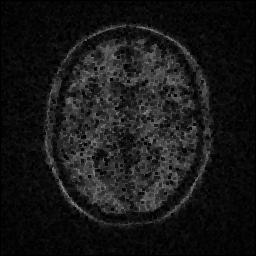
*Figure 11: MATLAB code to apply Order Statistics Filters to an image*

**

*Figure 12: Original Image*

**

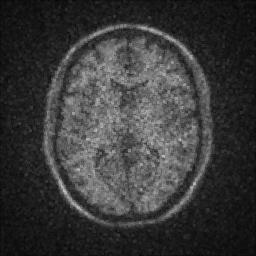
*Figure 13: Image after median filter*

**

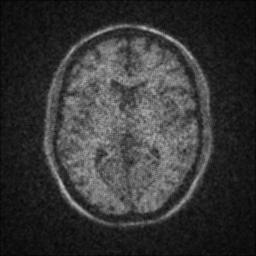
*Figure 14: Image after minimum filter*

**

*Figure 15: Image after maximum filter*

**

*Figure 16: Image after midpoint filter*

**

*Figure 17: Image after alpha trimmed filter*

**

*Figure 18: Image after bilateral filter*

**Observations:**

From the visualization of the images above *(Figure 13 – Figure 18)* which are the filtered versions of an original noisy MRI brain image *(Figure 12)* we can deduce what each filter is doing. Each filter firstly performs neighborhood operation and makes the image smoother by reducing noise. Median filter performs neighborhood operation based on the median value of the pixel and thus gives a smoother image but with less edge visibility. Minimum filter is reducing the intensity of each pixel (making them darker). Maximum filter doing the opposite of minimum filter i.e. increasing the intensity of pixels (making them brighter). Midpoint filter does neighborhood operation on pixels on the basis of thus equally distributes the minimum and maximum intensity among the pixels. Alpha trimmed filter is making the image smoother but hiding a lot of useful information particularly the edges of the image which contain the information about the structure of the image. Finally bilateral filter works with distributing the pixel values based on spatial distance and similar intensity and preserves the edges as we can observe that Image after applying bilateral filter *(Figure 18)* is the closest to the original image among all the other order statistic filters.

* Bilateral Filter is ideal for enhancing MRI images while keeping anatomical boundaries sharp.

**Advantages & Limitations:**

* Median is robust for impulse noise but can remove fine detail.
* Min/Max are useful for specific effects (e.g., morphological operations).
* Alpha-trimmed is a balance between mean and median.
* Bilateral offers the best detail preservation but is computationally expensive.

**5. Comparison and Recommendations**

When working with different types of filters in image processing, especially in biomedical imaging, it’s important to understand not just how they work, but when to use them and what trade-offs they come with. Each filter we discussed—mean filters, order statistic filters, and edge detectors like LoG and DoG—has its strengths and weaknesses, and the best one really depends on the task at hand.

Let’s start with noise reduction. If our image is affected by random bright or dark pixels (like salt-and-pepper noise), median filters or contraharmonic mean filters are excellent choices. The median filter works by replacing each pixel with the median value in its neighborhood, which is really effective for removing these sudden spikes of noise without blurring the image too much. The contraharmonic mean filter goes a step further we can actually tune a parameter “Q” depending on whether the noise is mostly light (salt) or dark (pepper), making it quite versatile.

On the other hand, if we just want to smooth out general background noise (like Gaussian noise), the arithmetic mean filter might do the job, though it tends to blur edges. If we want better edge preservation while still reducing noise, the bilateral filter is a fantastic choice. It smooths similar pixels while keeping edges sharp, something very useful in medical images where preserving boundaries between tissues is crucial. The only downside is that it’s a bit slow compared to others.

Edge detection is another major area where filters like LoG (Laplacian of Gaussian) and DoG (Difference of Gaussians) come into play. These filters don’t just reduce noise, they help us spot edges and transitions in the image, like where one tissue ends and another begins. LoG is more accurate but slower, while DoG is faster and good enough for most cases. So, for precision tasks like locating tumor boundaries, LoG might be better, but for real-time processing or simpler applications, DoG could be the smarter choice.

When it comes to preserving image details, filters like the alpha-trimmed mean offer a nice balance. They throw away a few of the most extreme pixel values (which are likely noise) and average the rest. It’s like a smarter version of the mean filter, less likely to blur important features.

Now, let’s talk speed vs. quality. Simple filters like the arithmetic mean or median filter are quick and easy to implement, which makes them suitable for large datasets or real-time applications. More advanced filters like bilateral or alpha-trimmed mean take longer to compute but give you better quality, especially when we need to keep fine details presrved.