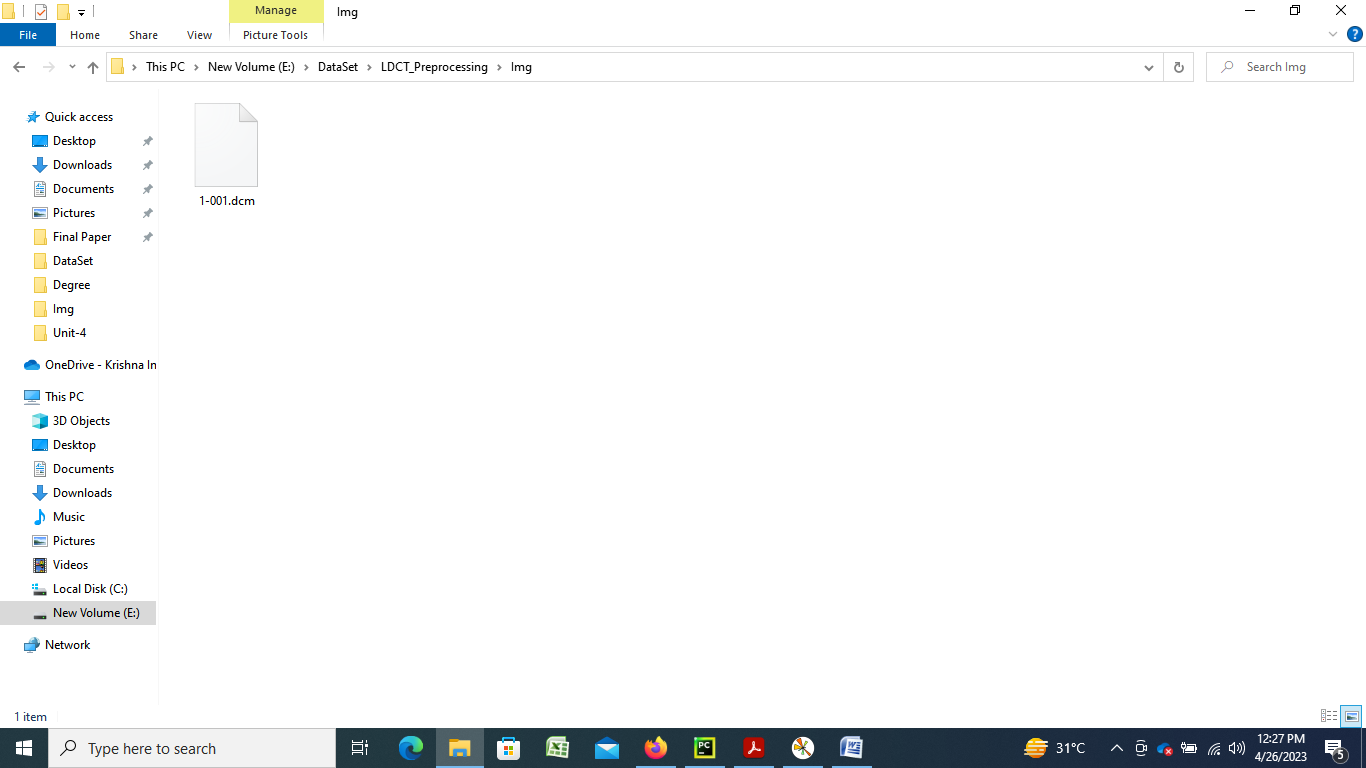
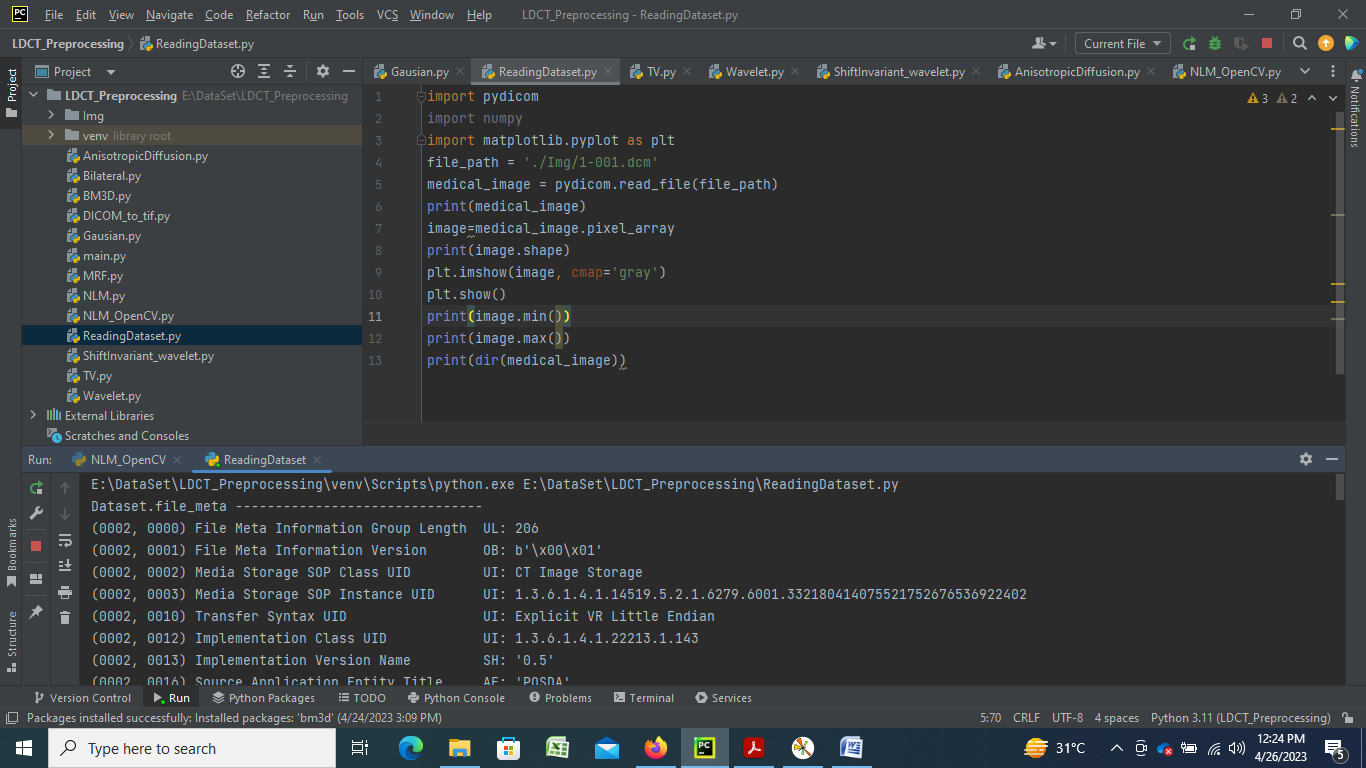
**Reading Dataset:**

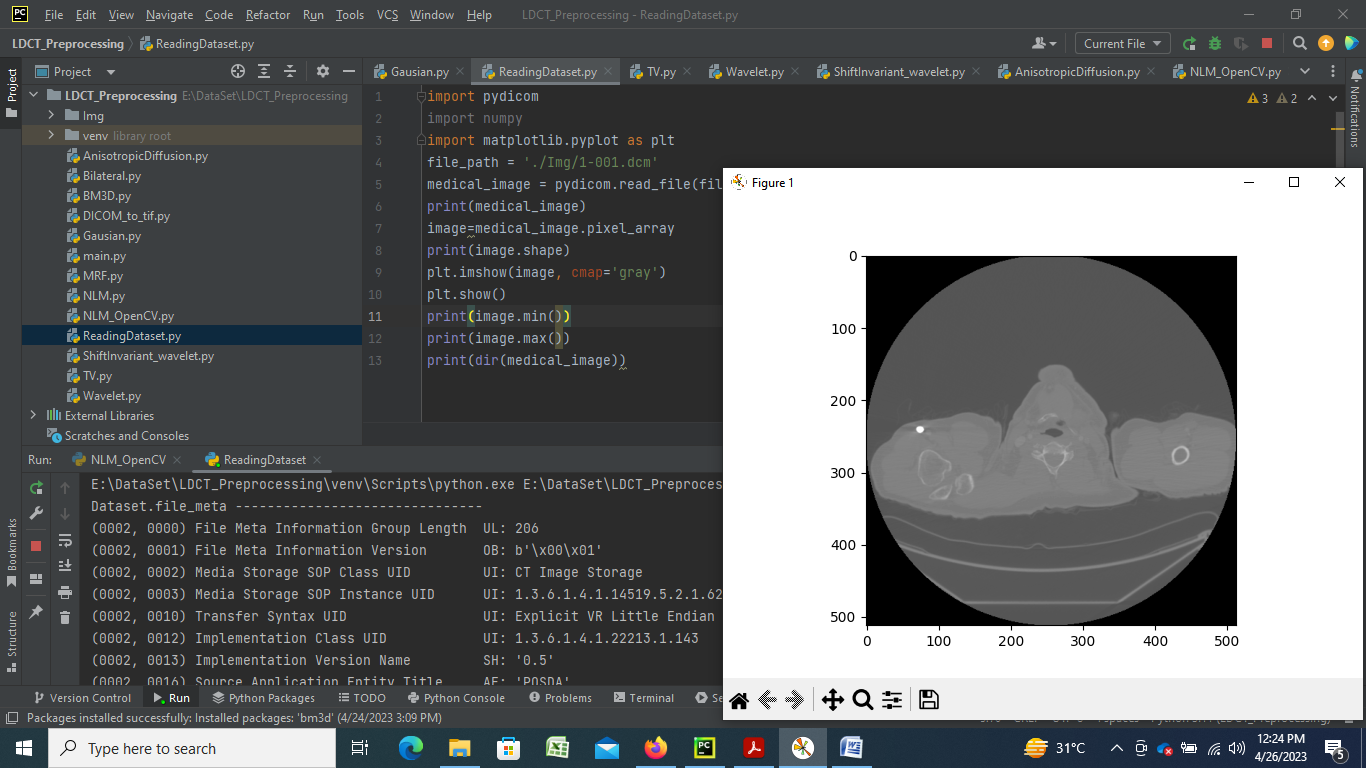
**Initially we have 16 bits dcm Images.**



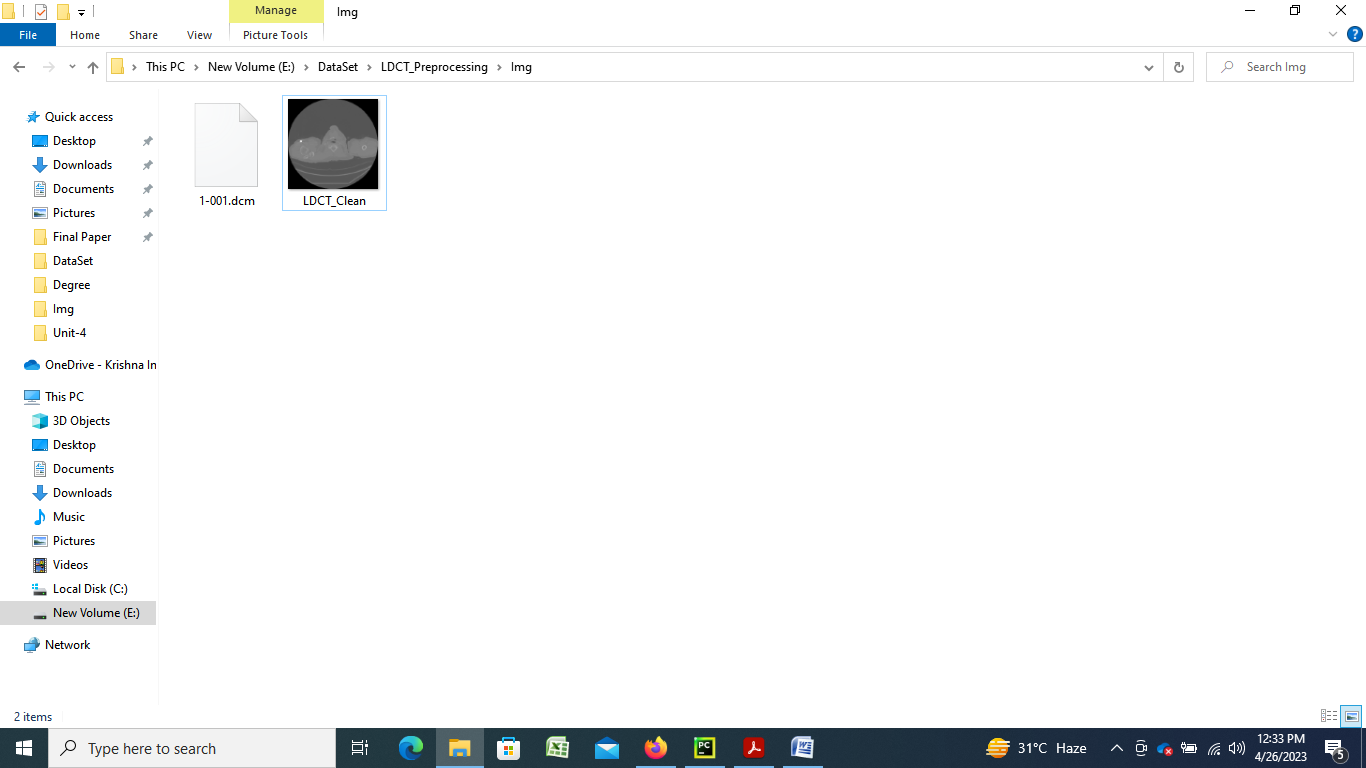
**Python Code to read DICOM images:**

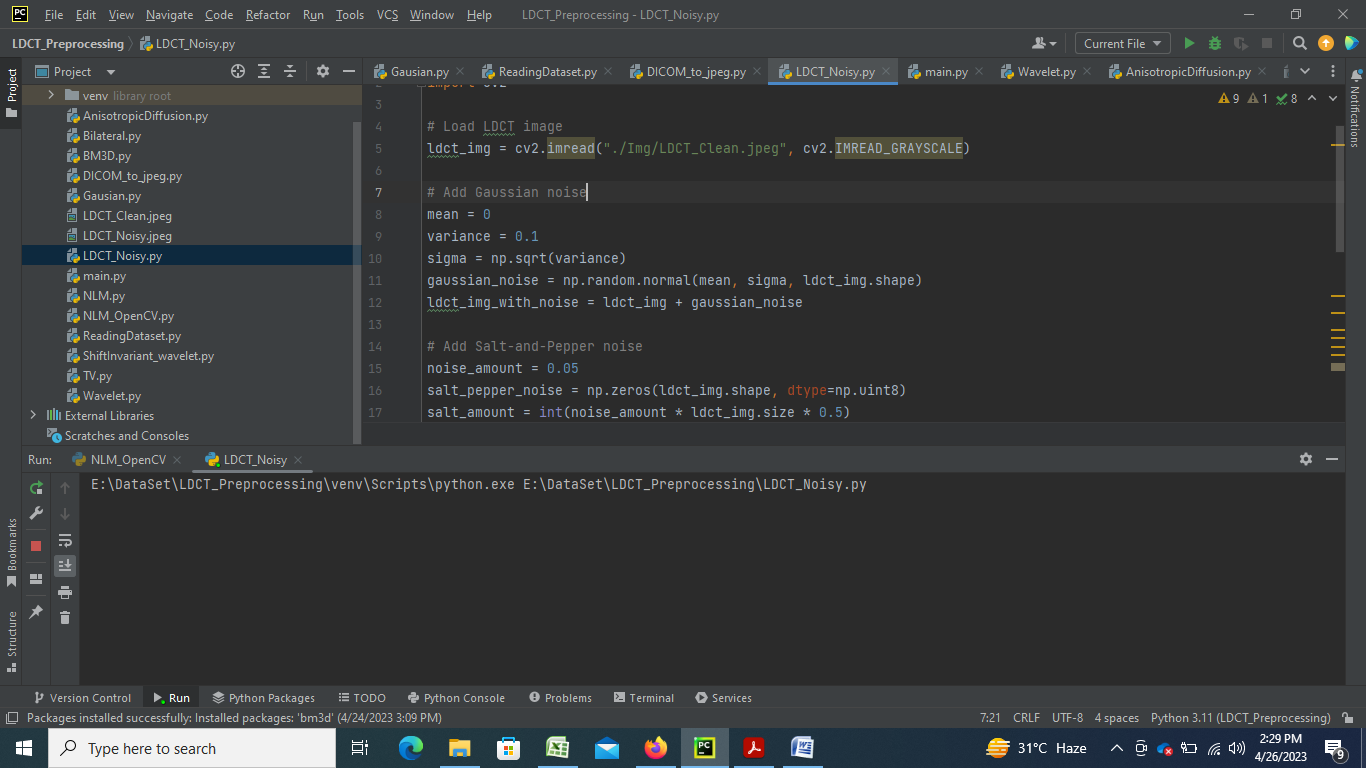


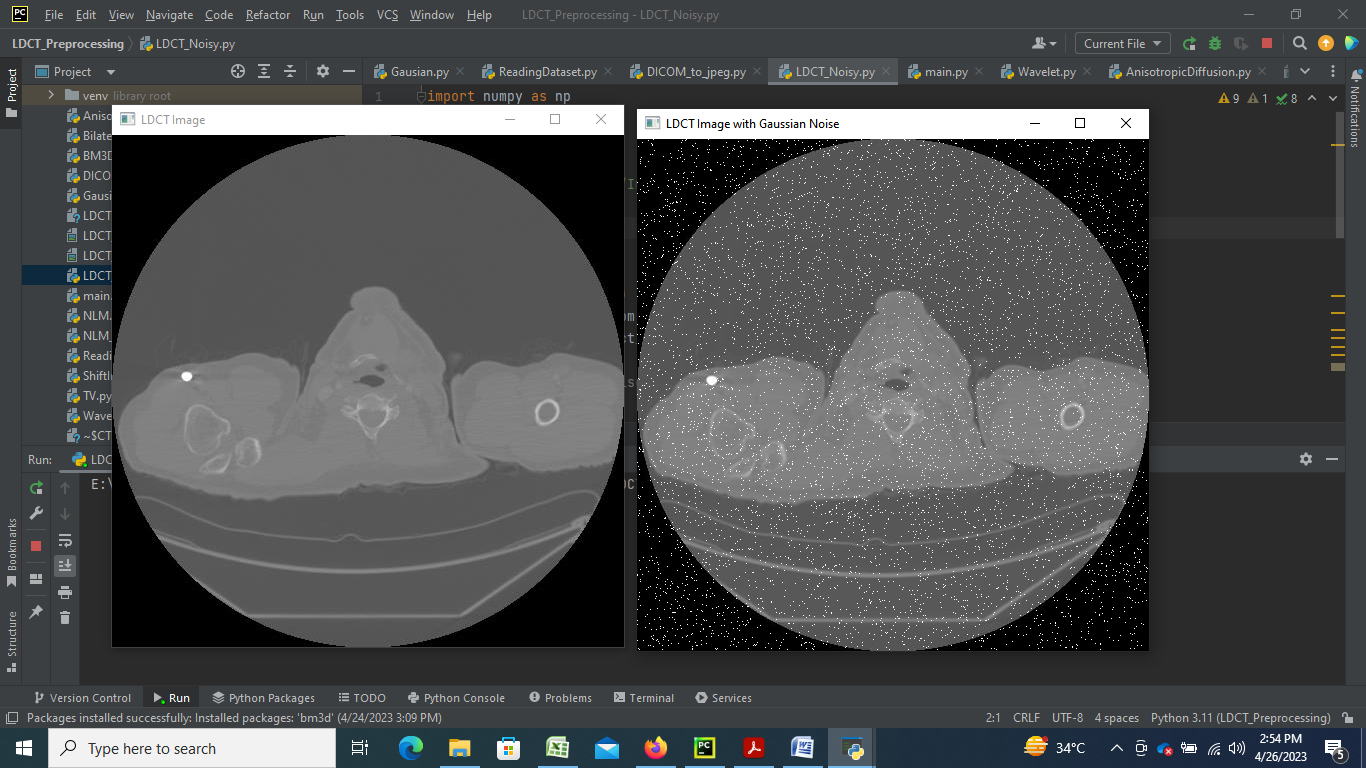
**Python Output DICOM images:**

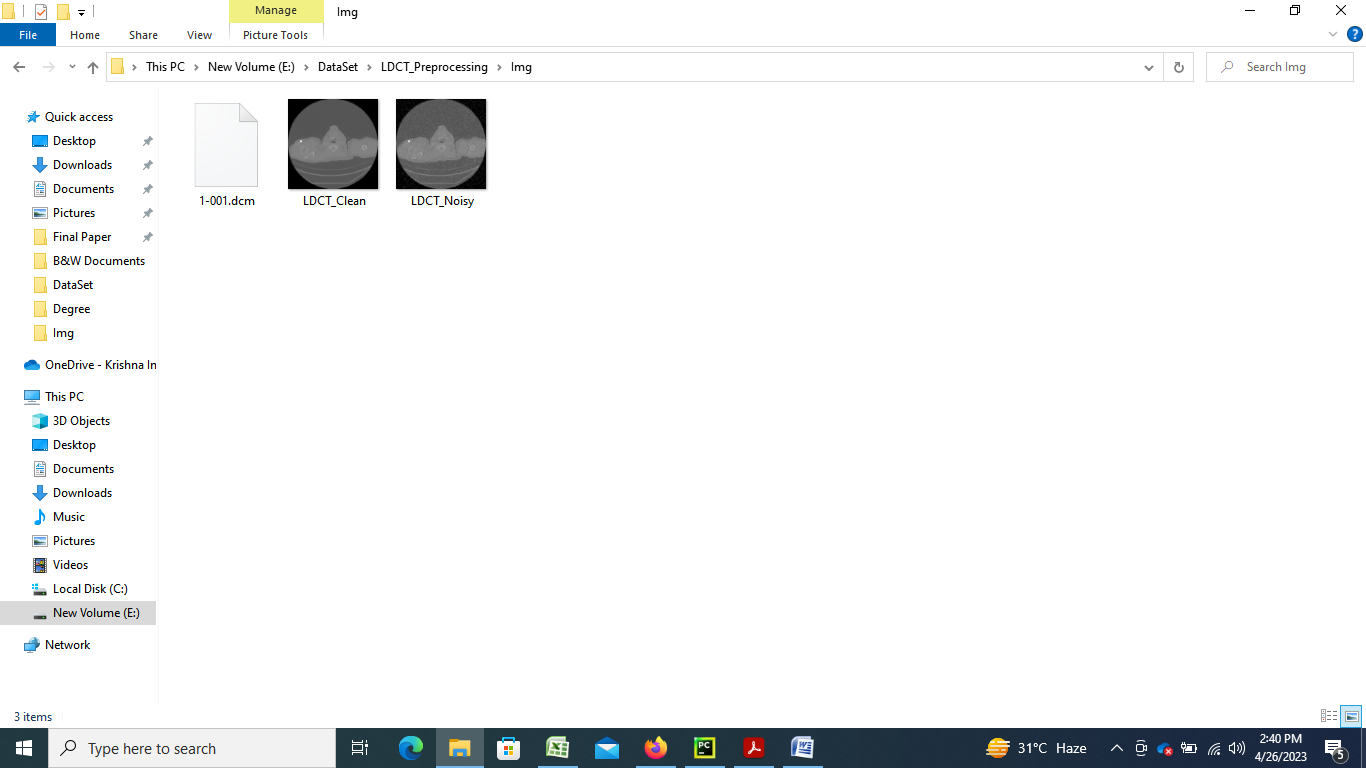


**Conversion of DICOM image into jpeg Image:**

****

**Creating LDCT Noisy Image: **

****

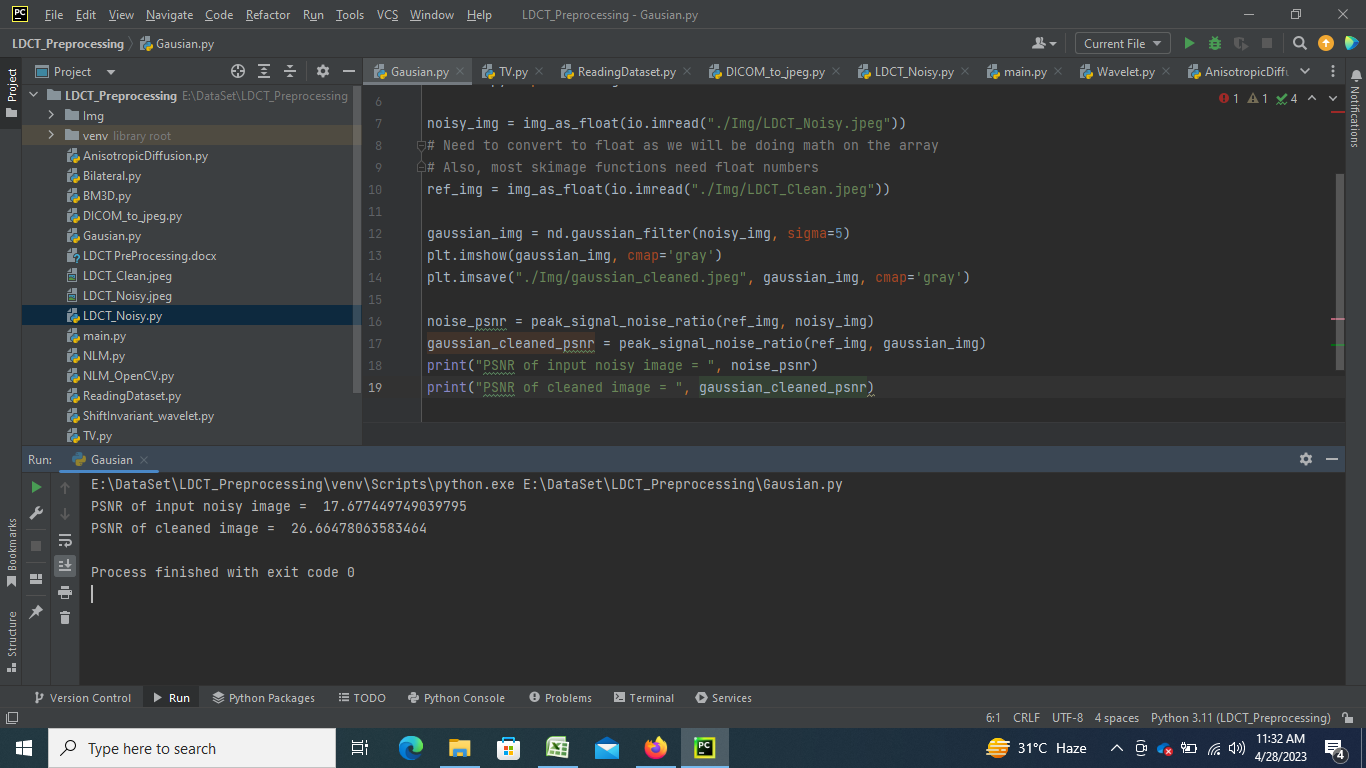
****

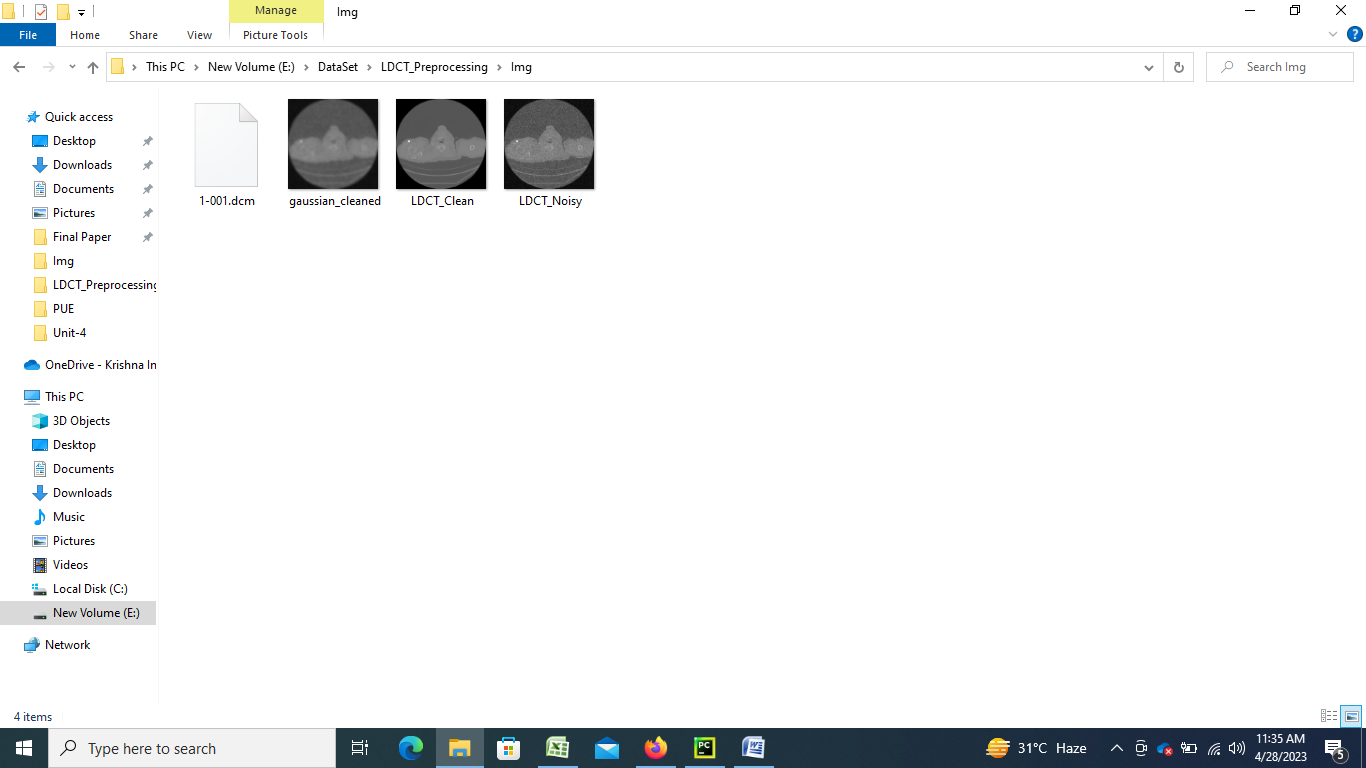
**Pre-Processing using Gaussian filter:**

A Gaussian filter is a type of digital filter used for image processing and computer vision applications. It is commonly used for image denoising, which involves removing noise from an image while preserving its sharpness and details.

The Gaussian filter works by convolving the image with a Gaussian kernel, which is a matrix of weights that represent the shape of a Gaussian distribution. The Gaussian kernel assigns higher weights to the pixels that are closer to the center of the kernel, and lower weights to the pixels that are further away. This has the effect of smoothing out the image, while preserving its overall structure.

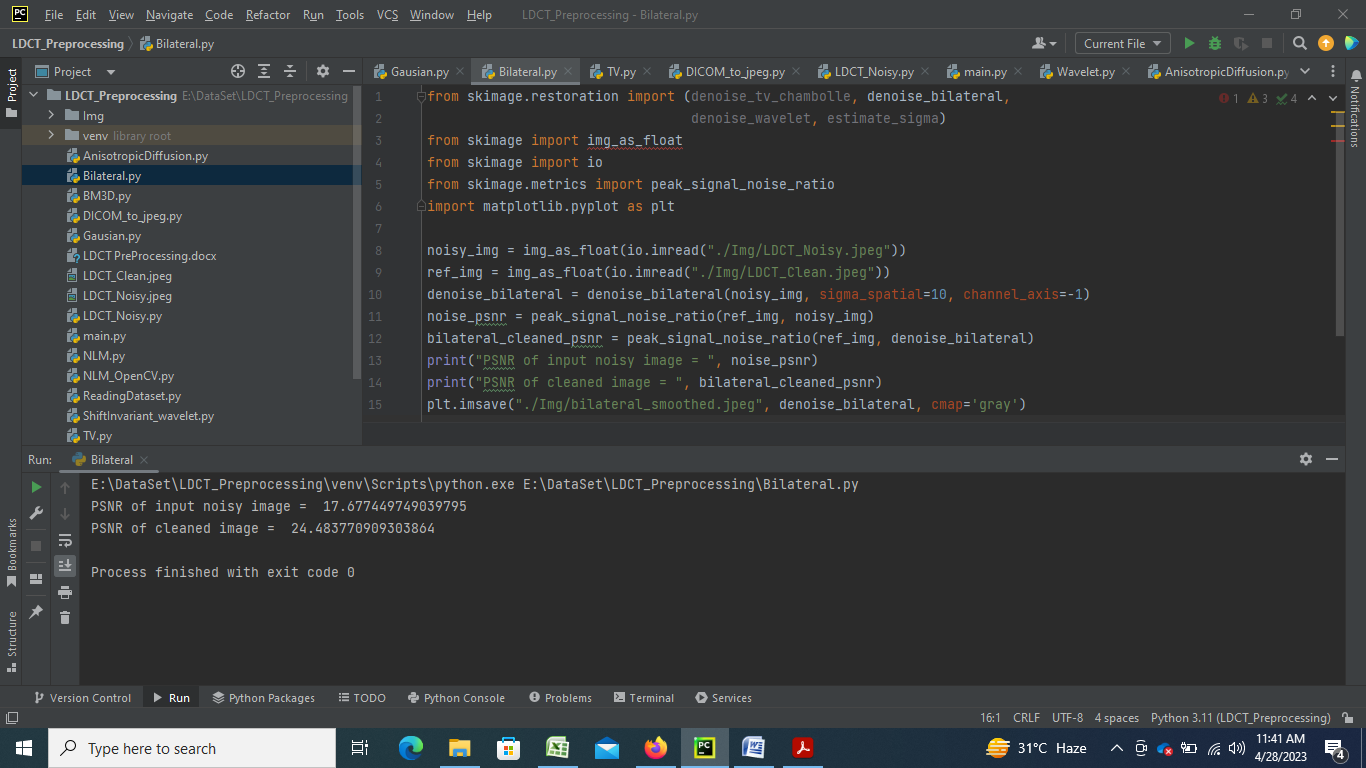
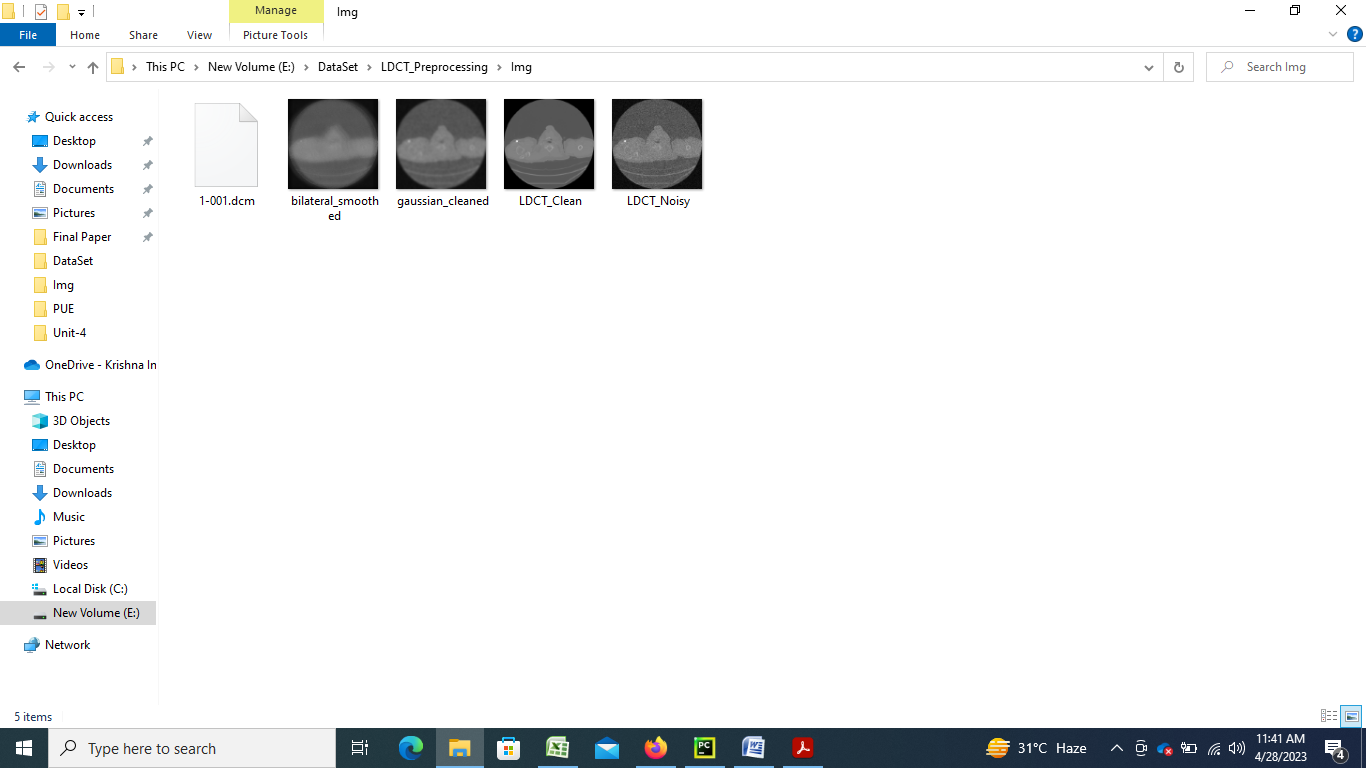
To use a Gaussian filter for denoising, you can apply the filter to the image using a convolution operation. The size of the filter kernel and the strength of the Gaussian distribution can be adjusted to control the amount of smoothing and the degree of noise reduction.

****



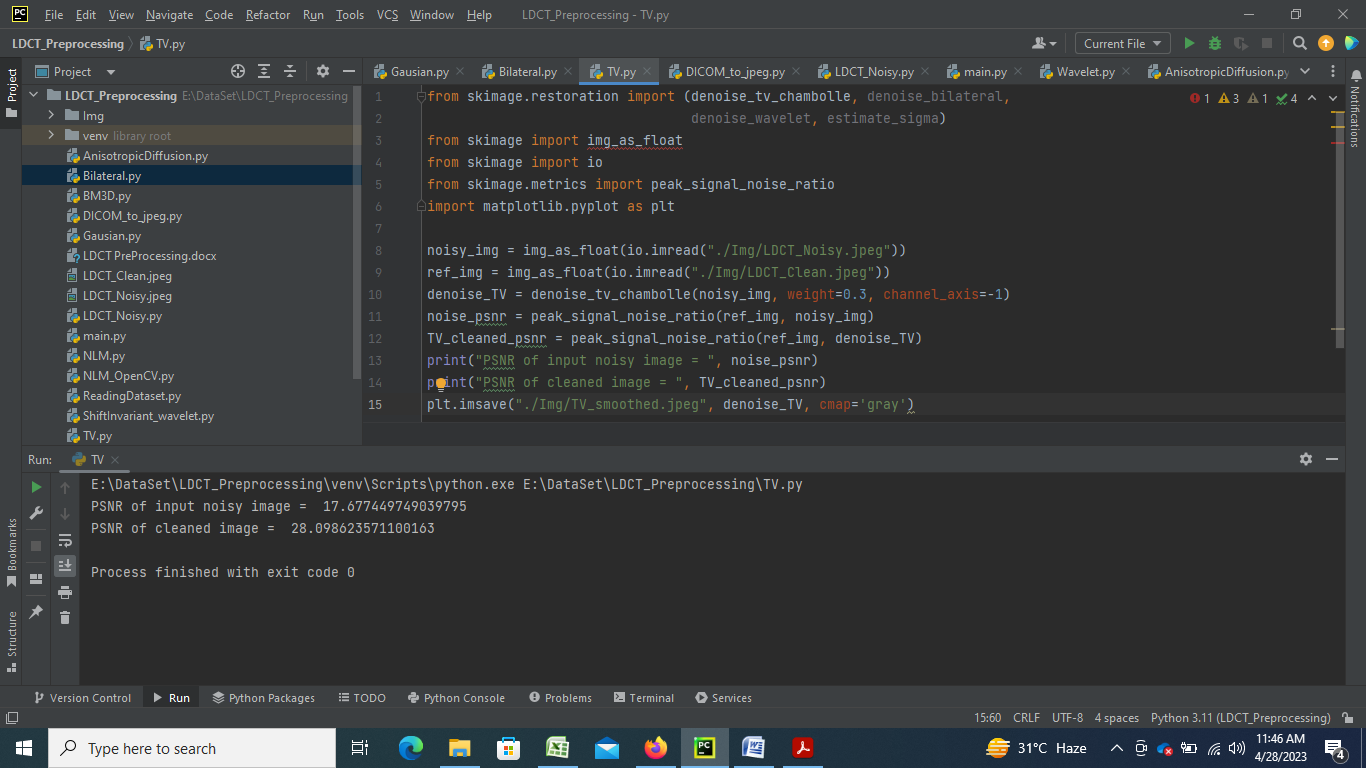
**Pre-Processing using Bilateral Method:**

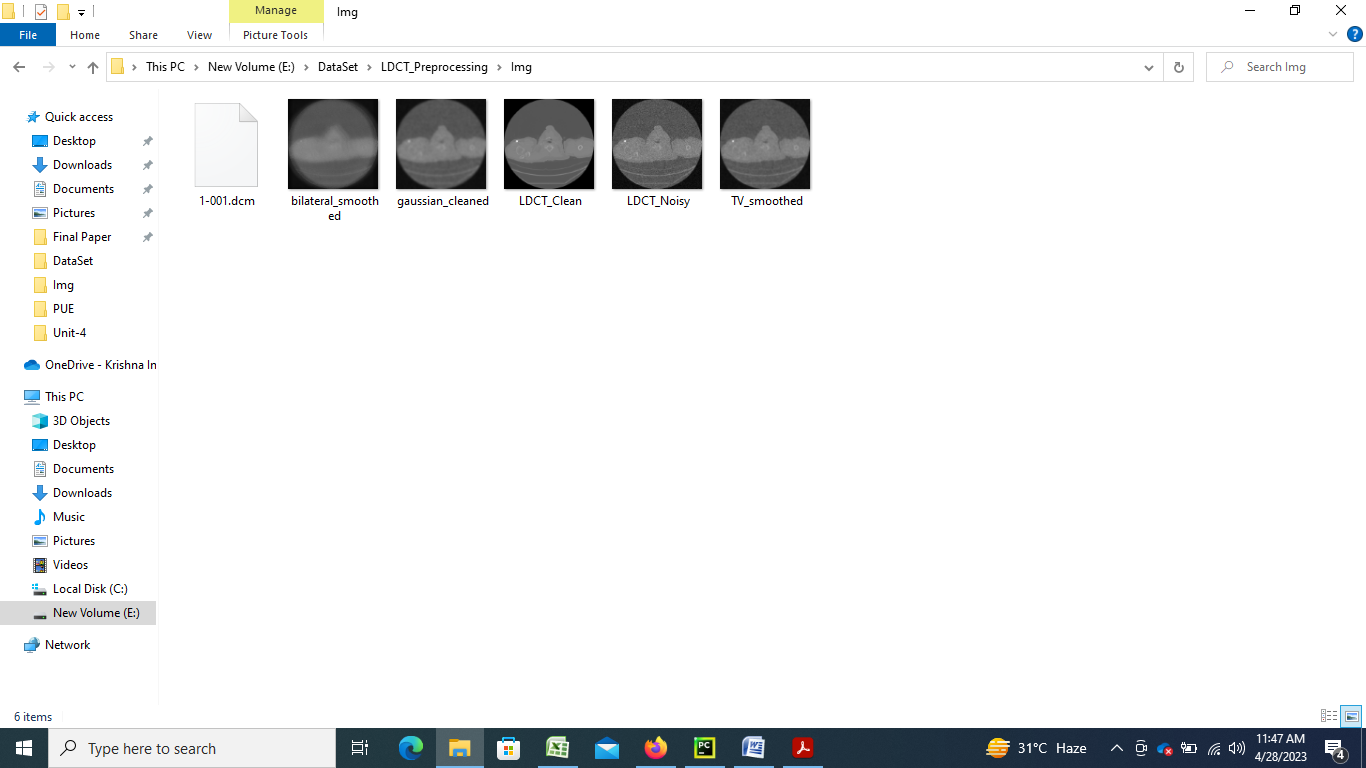
Bilateral filtering is a popular pre-processing technique used in image processing and computer vision applications. It is used to smooth out an image while preserving its edges and details. The technique uses a weighted average of the pixel values in a local neighborhood, where the weights are determined by both the spatial distance between pixels and the difference in intensity values.

** **

**Pre-Processing using TV Method:**

TV (Total Variation) denoising is a popular pre-processing technique used in image processing and computer vision applications. It is used to remove noise from an image while preserving its edges and details. The technique is based on minimizing the total variation of the image, which measures the amount of change in pixel values across neighboring pixels.

****

****

**Pre-Processing using Wavelet Method:**

Wavelet transform is a mathematical tool that is commonly used in signal processing and data analysis. It can be used for various purposes, such as signal denoising, feature extraction, and compression. The wavelet transform is based on the concept of decomposing a signal into different frequency components, which can provide valuable information about the signal.

The wavelet transform is performed using a wavelet function, which is a mathematical function that is localized in both time and frequency domains. The wavelet function is used to extract information about the signal at different scales, which allows for a more efficient and accurate analysis of the signal.

In order to use the wavelet transform for pre-processing data, the following steps can be taken:

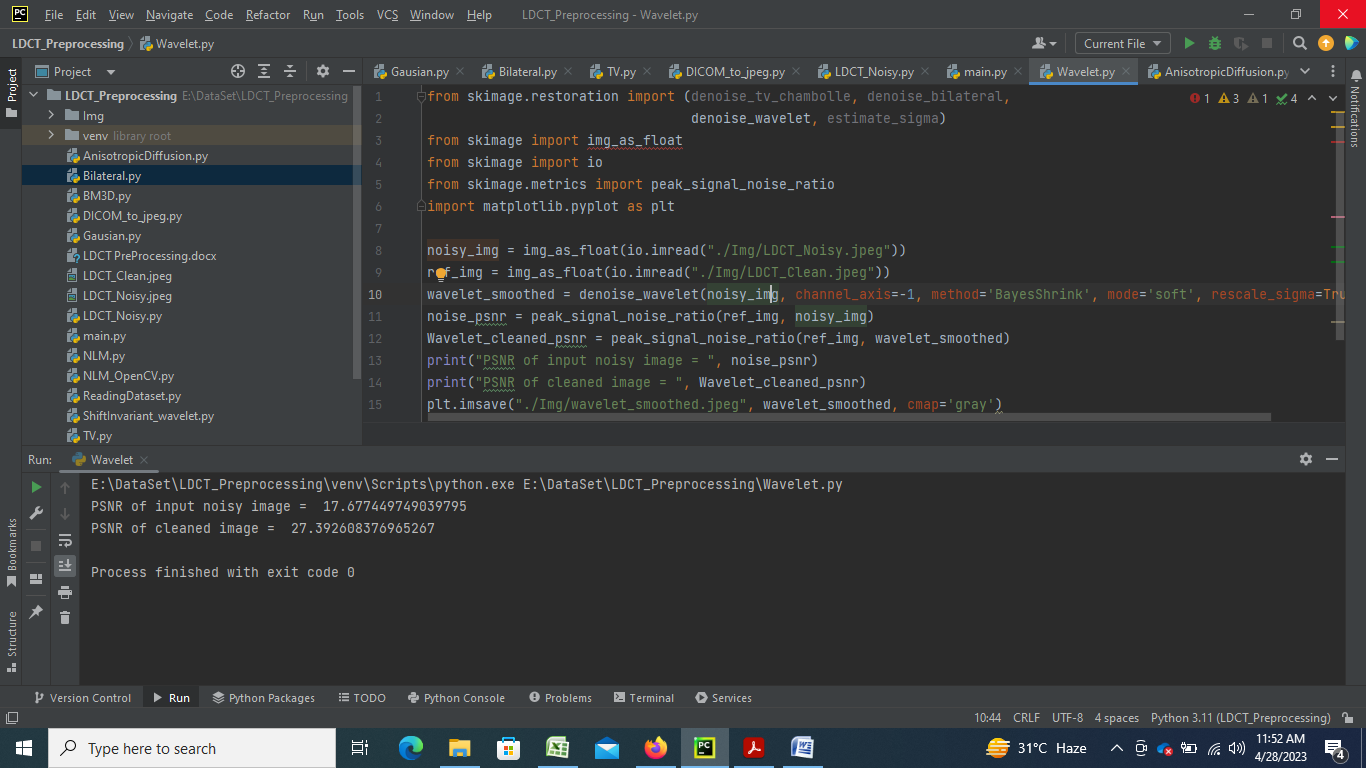
1. Choose a wavelet function: There are many different types of wavelet functions, each with its own strengths and weaknesses. The choice of wavelet function depends on the specific application and the characteristics of the signal being analyzed.
2. Apply the wavelet transform: The wavelet transform is applied to the signal using a specific algorithm. The result of the wavelet transform is a set of coefficients that represent the signal at different scales and frequencies.
3. Thresholding: In order to remove noise from the signal, a thresholding step can be applied to the wavelet coefficients. The thresholding step involves setting all coefficients below a certain threshold to zero, which effectively removes the noise from the signal.
4. Inverse wavelet transform: The inverse wavelet transform is then applied to the thresholded coefficients to reconstruct the denoised signal.

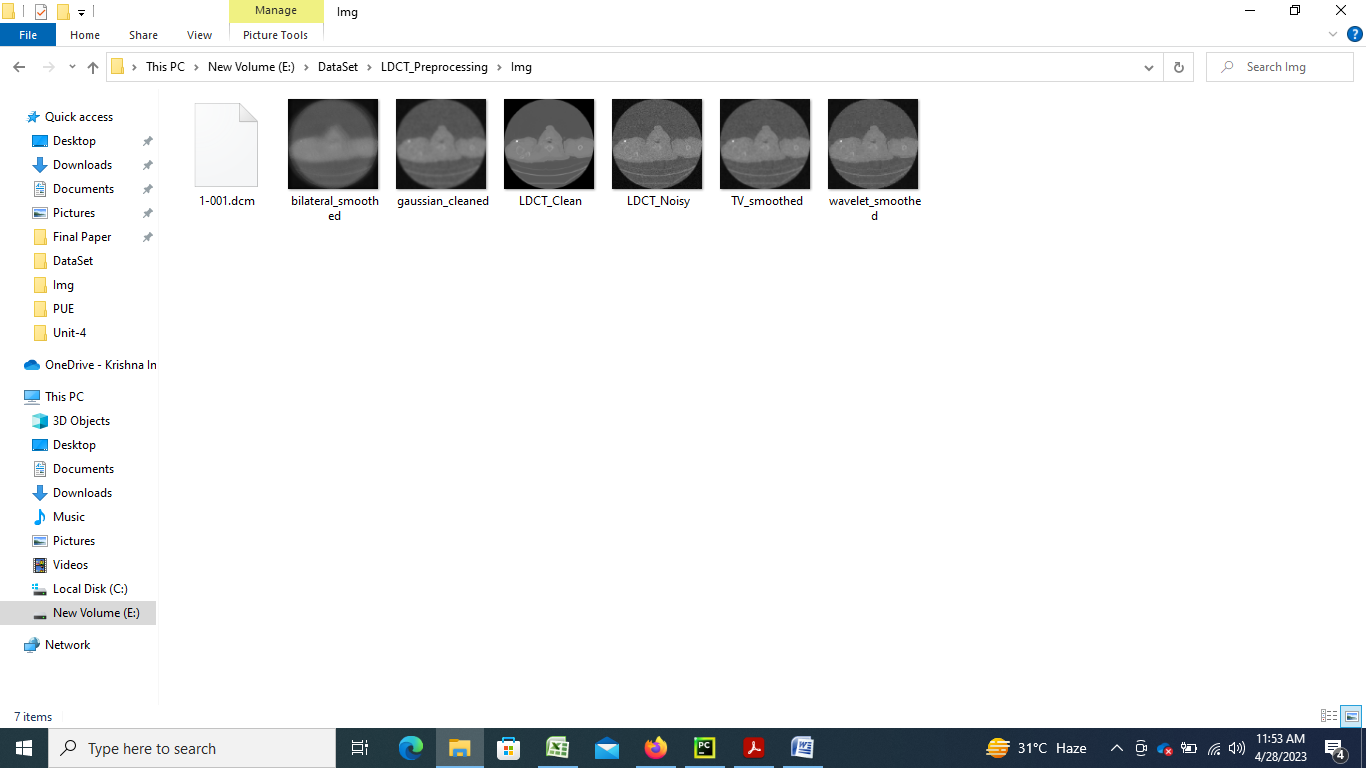
Overall, the wavelet transform can be a powerful tool for pre-processing data, especially when dealing with signals that contain noise or other unwanted artifacts. By decomposing the signal into different frequency components and selectively removing noise using thresholding, the wavelet transform can provide a more accurate and efficient analysis of the data.

Top of Form

Bottom of Form

Top of Form





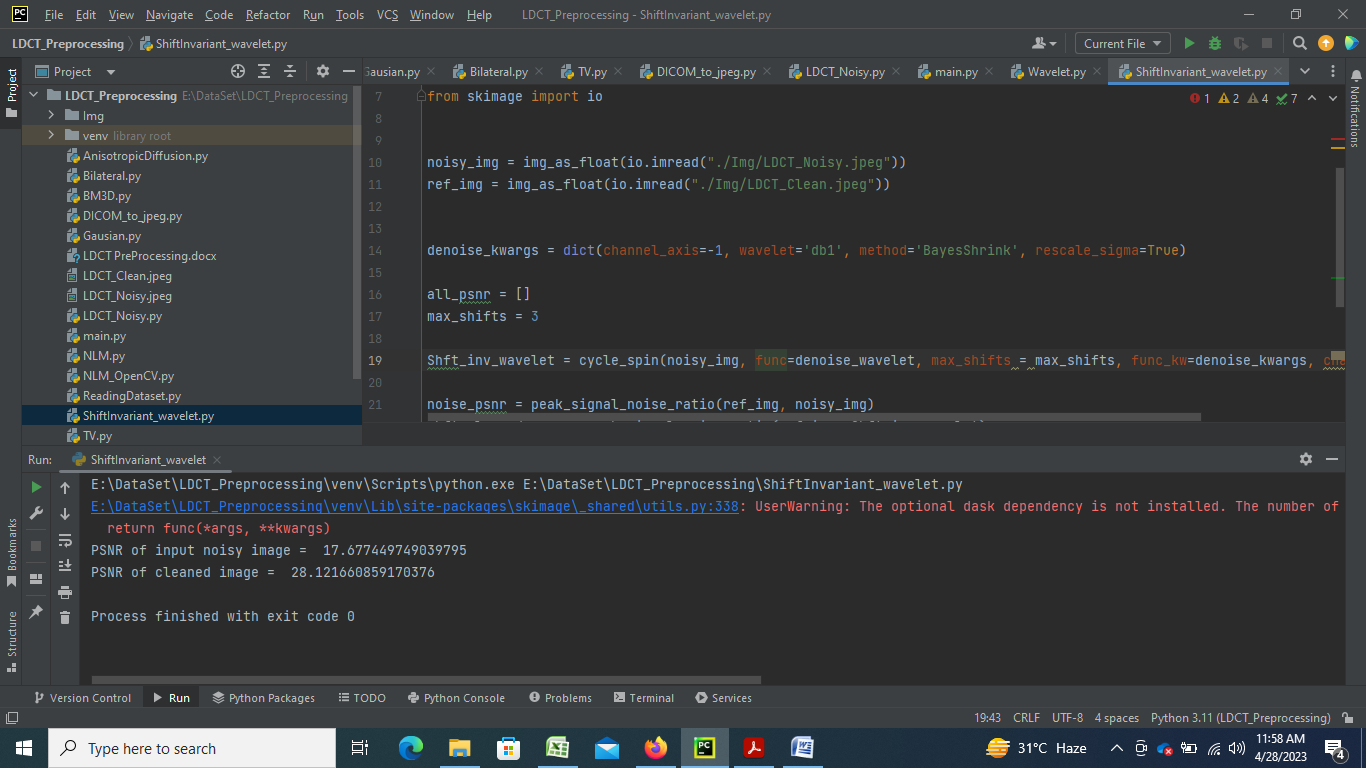
**Pre-Processing using Shift invariant wavelet denoising:**

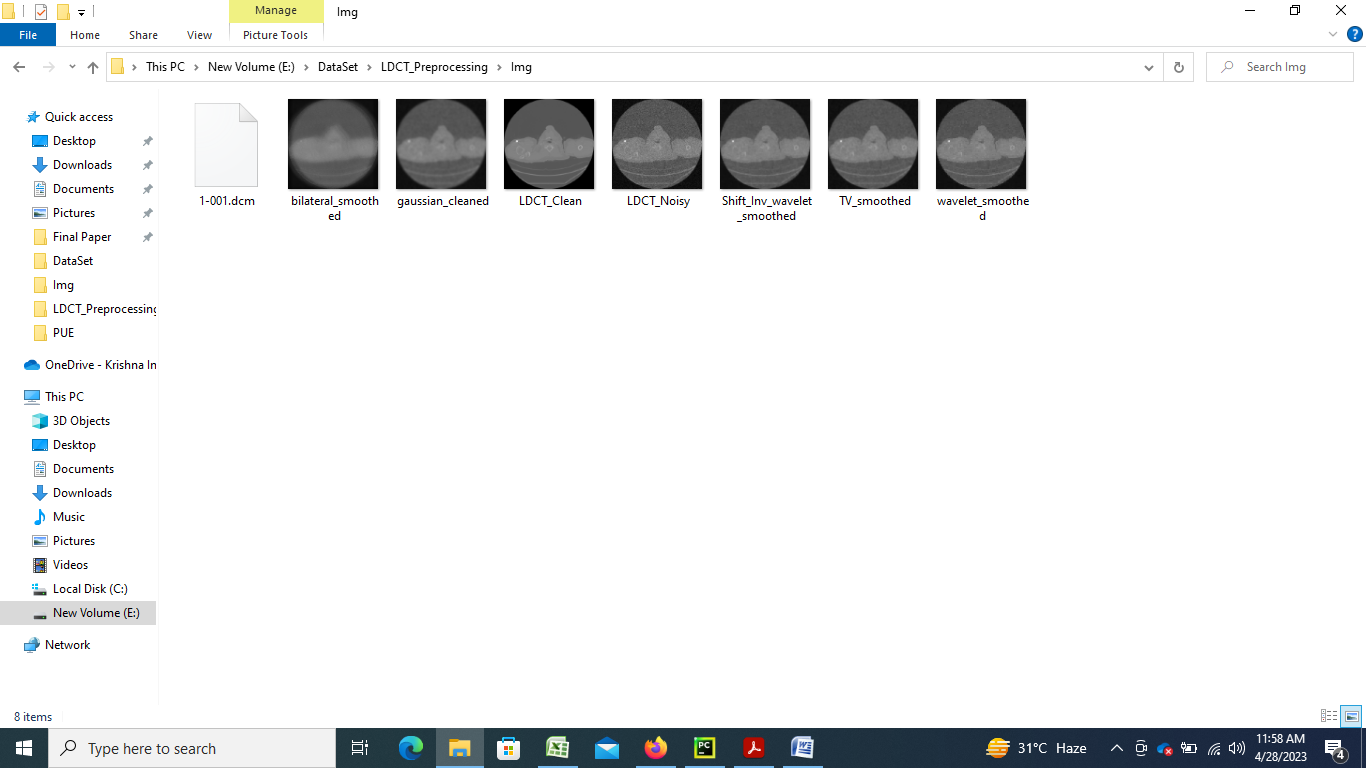
Shift invariant wavelet denoising is a type of wavelet denoising that is particularly useful for processing signals that are shift-invariant, meaning that their statistical properties remain the same even when they are shifted in time or space. This method of pre-processing data can be used to remove noise from signals while preserving important features, such as edges or other structural information.

The process of shift invariant wavelet denoising can be broken down into the following steps:

1. Choose a wavelet basis: Select a wavelet basis that is shift invariant, such as the discrete wavelet packet transform (DWPT), the dual-tree complex wavelet transform (DT-CWT), or the stationary wavelet transform (SWT).
2. Decompose the signal: Use the selected wavelet basis to decompose the signal into its wavelet coefficients at different scales and orientations.
3. Estimate the noise level: Estimate the level of noise in the signal using statistical techniques, such as the median absolute deviation (MAD) or the scale-invariant mean absolute deviation (sMAD).
4. Thresholding: Apply a threshold to the wavelet coefficients to remove coefficients that are below the estimated noise level. There are different types of thresholding methods that can be used, such as soft or hard thresholding, or adaptive thresholding.
5. Reconstruct the signal: Use the thresholded wavelet coefficients to reconstruct the denoised signal using the inverse wavelet transform.

Shift invariant wavelet denoising is a powerful pre-processing technique that can be used for a variety of applications, such as image or audio processing. By using a shift invariant wavelet basis and estimating the noise level, this method can effectively remove noise from signals while preserving their important features.





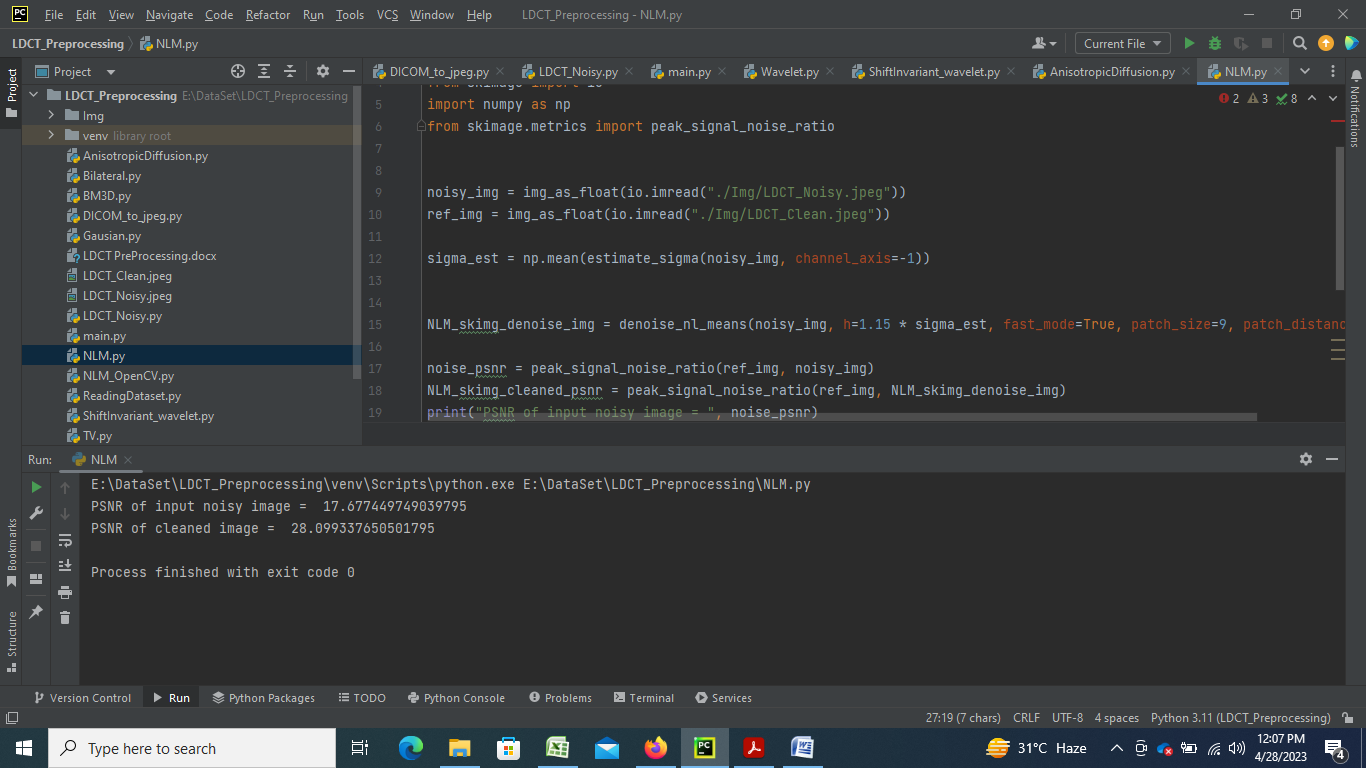
**Pre-Processing using NLM from SKIMAGE:**

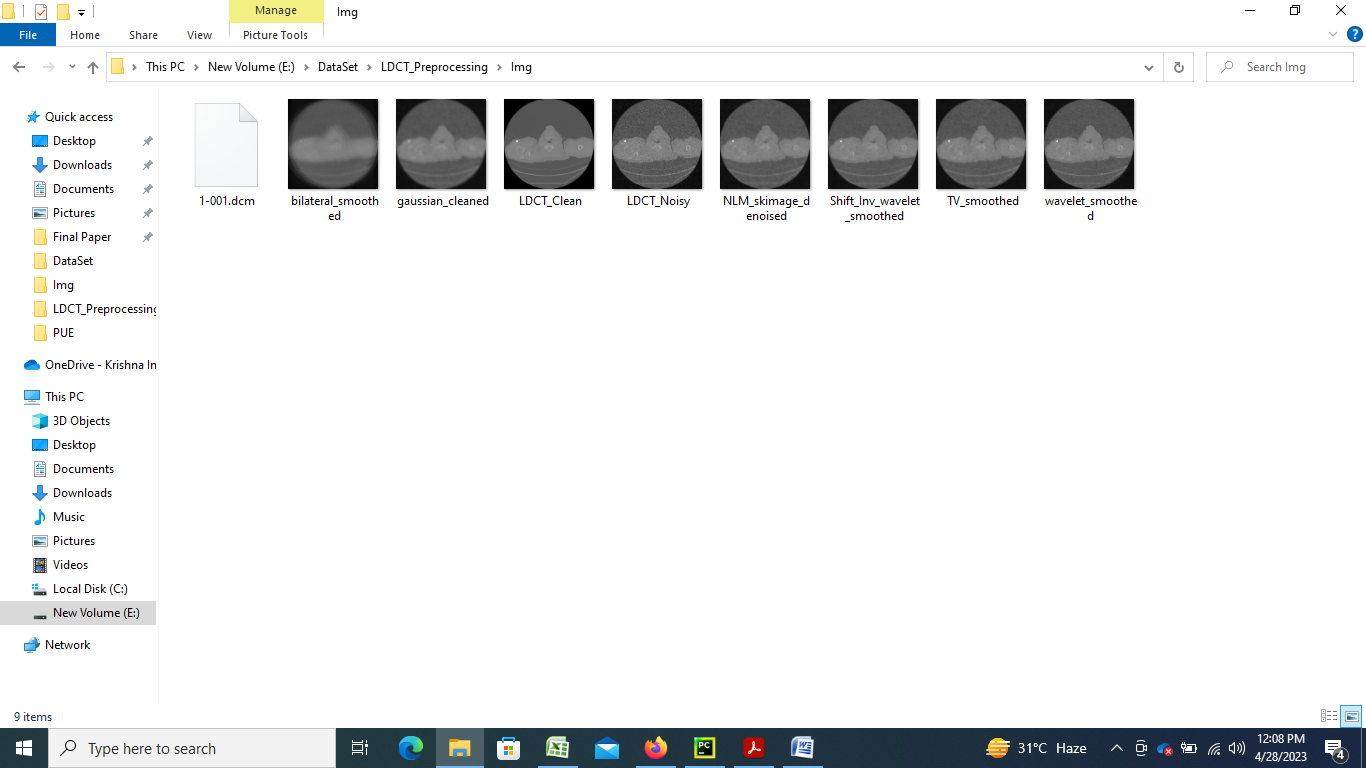
NLM (Non-Local Means) is a denoising algorithm that can be used for pre-processing images or other types of signals. It works by comparing small patches of the signal with other similar patches, regardless of their location in the signal. This approach allows for more effective noise reduction, as it takes into account the overall structure and features of the signal.

To perform pre-processing using NLM, the following steps can be taken:

1. Import the necessary libraries: Depending on the programming language being used, different libraries may be required to use the NLM algorithm. For example, in Python, libraries such as NumPy and OpenCV may be useful.
2. Load the signal: Load the signal that needs to be pre-processed.
3. Convert the signal to an appropriate format: Convert the signal to an appropriate format for processing by the NLM algorithm. For example, in Python, the signal can be converted to a NumPy array.
4. Add noise to the signal: Add noise to the signal to simulate a noisy signal.
5. Apply the NLM algorithm: Apply the NLM algorithm to the noisy signal. This can be done using various techniques, such as iterating over the signal and comparing each patch with other similar patches, or using pre-built functions from libraries such as OpenCV.
6. Evaluate the results: Evaluate the results of the NLM algorithm by comparing the denoised signal to the original signal using metrics such as the peak signal-to-noise ratio (PSNR).
7. Visualize the results: Visualize the original signal, the noisy signal, and the denoised signal using appropriate visualization techniques.

By following these steps, the NLM algorithm can be used to pre-process noisy signals and obtain a denoised version of the signal.



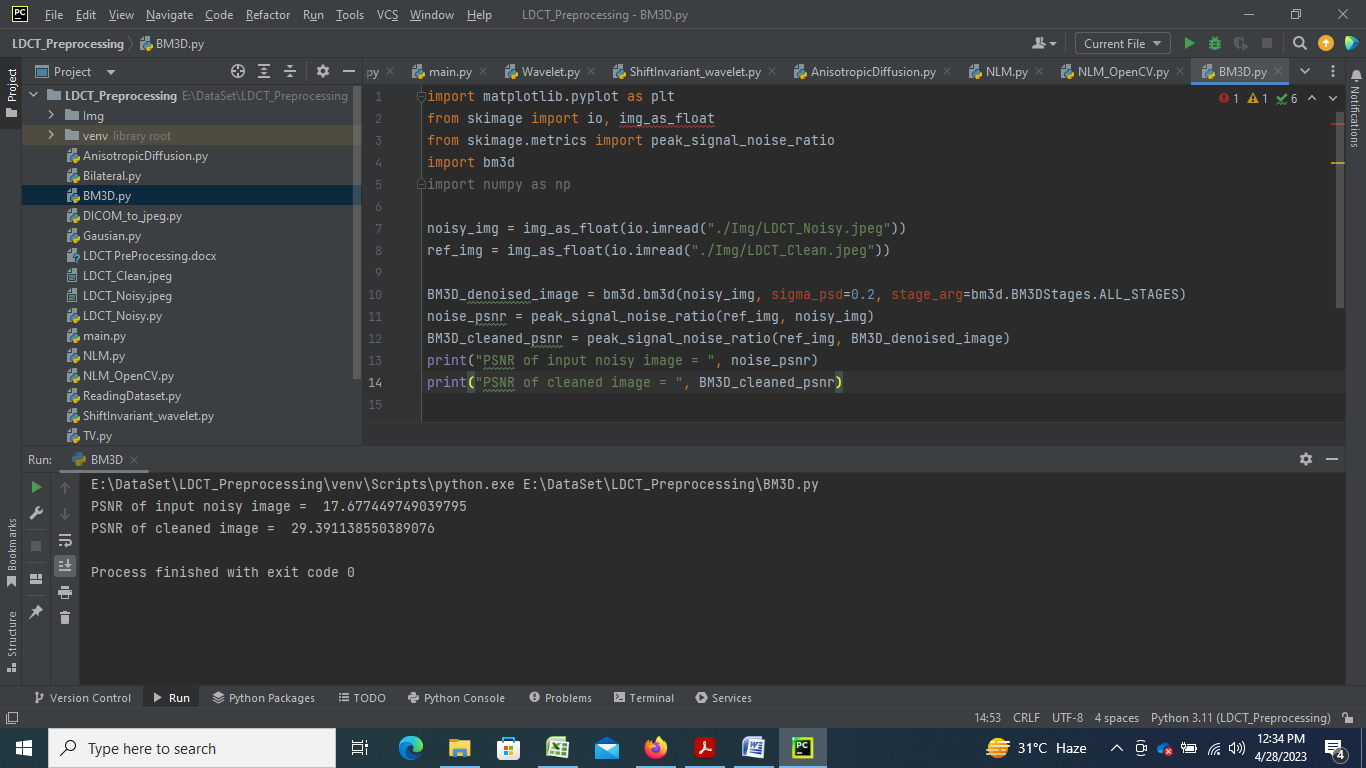


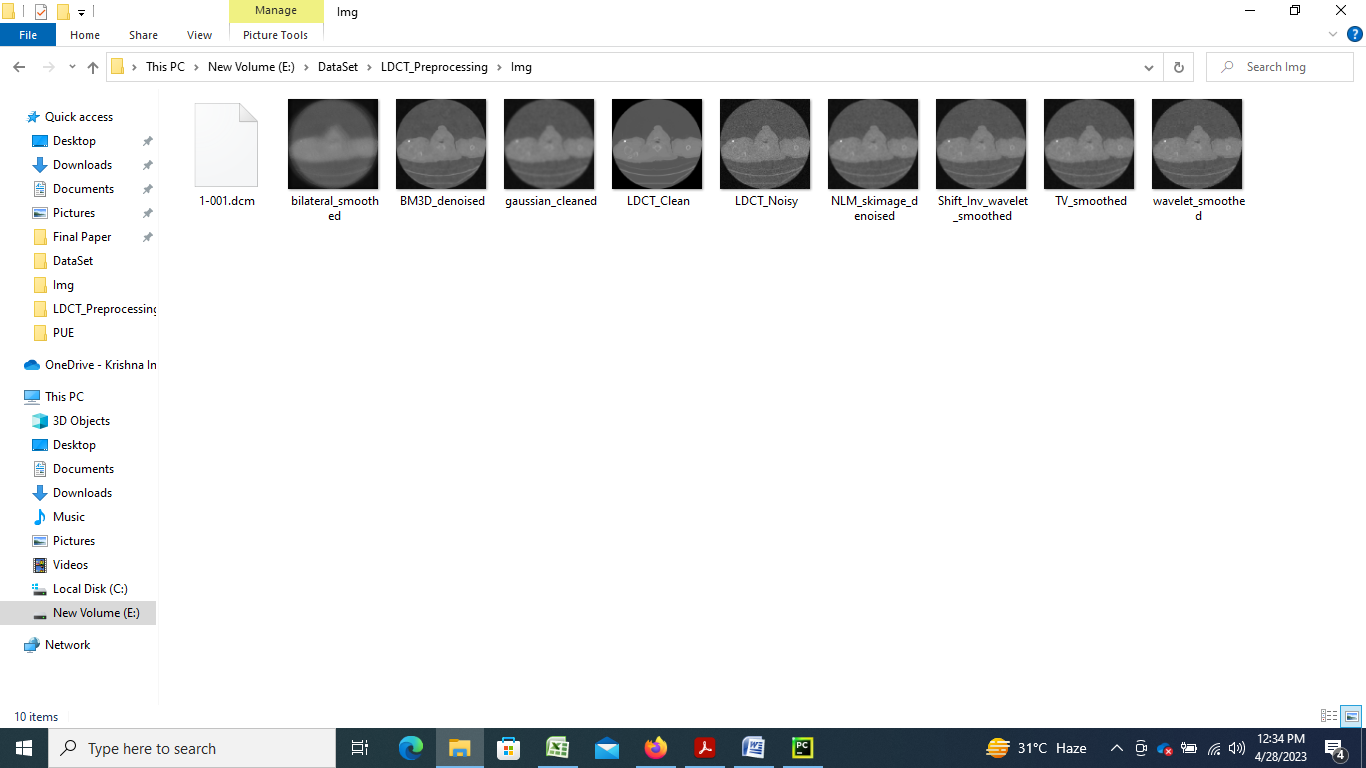
**Pre-Processing using BM3D Block-matching and 3D filtering:**

BM3D (Block-matching and 3D filtering) is a well-known image denoising algorithm that has been widely used in image processing. The basic idea of BM3D is to group similar image patches using a block-matching algorithm and then apply a 3D filtering process on these patches to remove noise.

The pre-processing steps using BM3D typically involve the following:

1. Block Matching: Divide the input image into small non-overlapping blocks and search for similar blocks within a search window. This is typically done using the sum of absolute differences (SAD) or normalized cross-correlation (NCC) metric.
2. Grouping: Group the similar blocks found in step 1 into clusters or groups based on their similarity. This grouping step is important as it helps in preserving the structural information of the image.
3. 3D Filtering: For each group, perform 3D filtering by stacking the similar blocks along a third dimension and applying a 3D transform such as the discrete cosine transform (DCT) or the discrete wavelet transform (DWT) to the stacked block. The transform coefficients are then thresholded to remove the noise and the inverse transform is applied to obtain the denoised image.
4. Aggregation: Finally, the denoised blocks are aggregated back into the original image using a weighted averaging technique.

****

****