Project 2

Kshitija Suresh Jadhav

1. A short introduction and a brief description of the theory and the algorithms you are implementing. (i.e., the histogram equalization, log transformation, rotation, etc.).

For this project I implemented 5 key image processing techniques:

1. Histogram Equalization: A common technique for increasing contrast in an image is histogram equalization (or histogram flattening). The goal is to redistribute gray levels among pixels so that the resulting histogram is flat.
   This redistribution causes the gray level range to be compressed around the histogram's peak, which is where most of the pixels are, and enlarged near the tails. More detail is seen since gray levels are extended for most pixels.
   The process entails adjusting the image's intensity levels so that the resultant image's histogram is roughly flat, indicating a uniform distribution of pixel intensities. When working with excessively bright or dark photos, this technique can greatly improve the contrast of the image.
   My code: The algorithm I used calculates the histogram of the input image and uses it to distribute the intensity values evenly across the histogram. This was achieved by mapping the cumulative distribution function (CDF) of the original image to a linear CDF that spans the entire range of intensity levels.
2. Log Transformation: When a picture has a large range of intensity values, the log transformation is employed for contrast enhancement and dynamic range compression. Details in darker areas of a picture are made more visible by compressing and stretching the bright pixels using a logarithmic function applied to each pixel value.
   My code: here I implemented a logarithmic function to the intensity values of the pixels. The formula used is $s=c\log(1+r)$, where r is the input pixel value, s is the output pixel value, and c is a constant. The transformation compresses the range of high-intensity values while expanding the range of low-intensity values.
3. Rotation: By rotating a picture around a central point, rotation modifies the image while maintaining its dimensions. This is helpful for tasks that require the image to be viewed from various angles or for reorienting images.

My code: When a picture is rotated by a specified angle, all of the pixels' new positions are determined. The new locations might not line up perfectly with the pixel grid, therefore interpolation is typically required. Basic trigonometry is the basis for the rotation formula, which revolves points around the origin and then translates them back to their correct positions inside the image.

4. Gaussian Blur: Gaussian blur is a smoothing technique used to reduce image noise and detail using a Gaussian filter. It is widely used in image processing to create a 'blurring' effect.
   Code: The gaussian blur operation involves convolution of the image with a Gaussian function. Closer pixels are given a higher weight based on the Gaussian distribution, and

the surrounding pixels are weighted to average their values in this procedure. As a result, the image appears blurry and smooth, with less discernible noise and edges.

5. Median filter: A non-linear digital filtering method called the median filter is frequently used to eliminate noise from signals or pictures. The primary concept involves iterating through the image pixel by pixel, substituting the values with the median values of adjacent pixels for each one. The "window," or pattern of neighbors, moves pixel by pixel over the whole image.
   Code: In order to apply the median filter, all of the nearby neighborhood's pixel values are sorted into numerical order, and the pixel under consideration is then replaced with the list's median value. 'Salt and pepper' noise can be effectively removed from an image using this technique while maintaining the image's edges.

2. The results of running your implementations on the images given above. You are welcome to test other images as well.

I implemented 3 files:

One to perform all the 5 above techniques on the mentioned images:

Taken parameters:

Theta = 45 degrees

Standard deviation = 1

Size of the kernel = 3 (Since it’s given that kernel size has to be taken as 3 x 3 matrix)

1. Auto.pnm results:

Original Image - auto.pnm

Median Filtered Image - auto.pnm

Rotated Image - auto.pnm

Log Transformed Image - auto.pnm

Blurred Image - auto.pnm

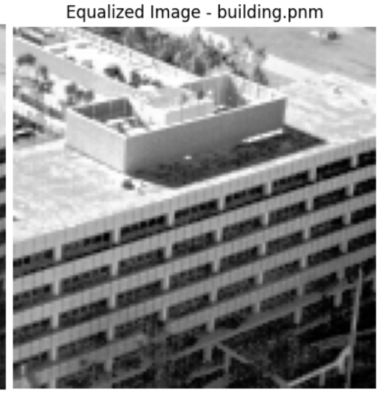
Equalized Image - auto.pnm

2. Building.pnm

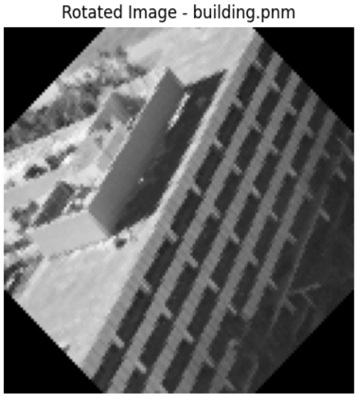
Original Image - building.pnm

Log Transformed Image - building.pnm

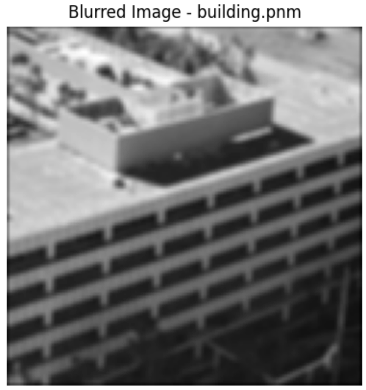
Equalized Image - building.pnm



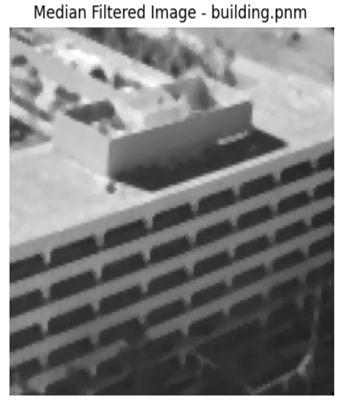
Rotated Image - building.pnm



Blurred Image - building.pnm

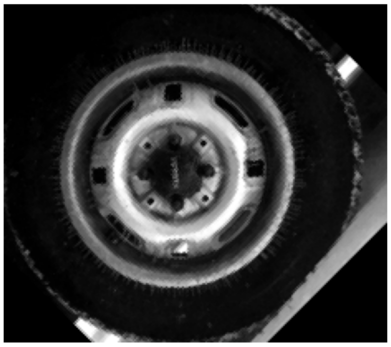


Median Filtered Image - building.pnm



3. Tire.pnm

Rotated Image - tire.pnm



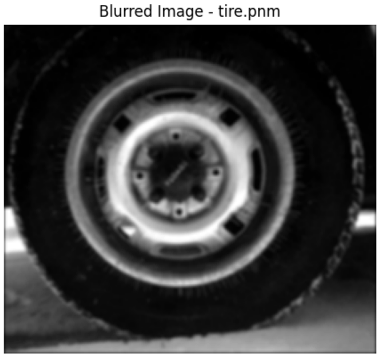
Original Image - tire.pnm

Median Filtered Image - tire.pnm

Log Transformed Image - tire.pnm

Equalized Image - tire.pnm

Blurred Image - tire.pnm



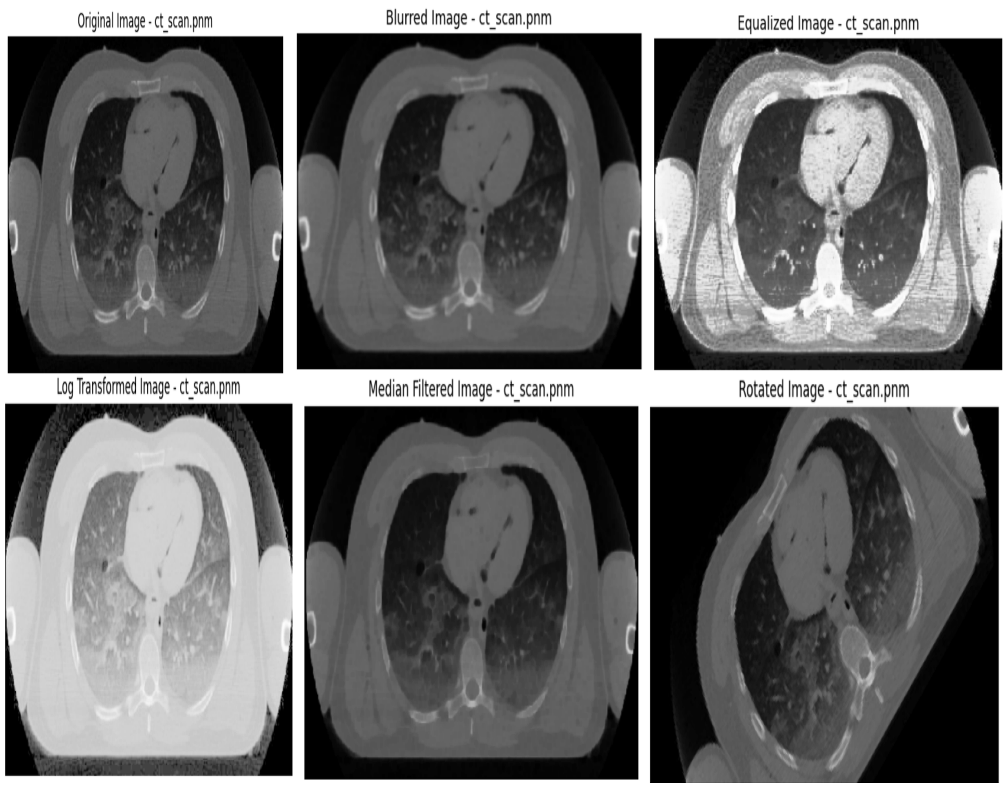
4. child.pnm

Original Image - child.pnm

Blurred Image - child.pnm

Equalized Image - child.pnm

5. ct_scan.pnm



3. An analysis/interpretation of your results:
a. What does each function do?

1. Histogram equalization: modifies an image's histogram to make it roughly consistent, hence adjusting the contrast of the picture. By redistributing the intensity levels of the image, it improves overall contrast by making the light areas somewhat darker and the dark areas lighter.

2. Log Conversion: Gives the image's intensity values a logarithmic scale. By compressing the high-intensity values and increasing the values of the dark pixels, this technique is especially good at bringing out the features in the darker regions.
3. Rotate Image: This function allows you to rotate an image about its center by a predetermined angle. Gaps that the rotation would have otherwise caused are essentially filled in by setting all pixels to black that don't directly match onto the original image.
4. Gaussian Blur: Function: Uses a Gaussian filter to smooth out the image. By averaging the values of the pixels inside the range specified by a Gaussian kernel, this process lowers image noise and detail.
5. The median filter
   Functionality: The median value of the intensities in the neighborhood that the kernel defines is used to replace each pixel's value in this filter. It works especially well at keeping edges intact while eliminating noise like salt and pepper.

b. How does it affect the image?

1. Histogram Equalization: Enhances the global contrast of the image, making details in both bright and dark areas more discernible.
2. Log Transformation: Dark regions become lighter, allowing for enhanced visibility of details in shadows without significantly affecting the bright areas.
3. Rotate Image: Changes the orientation of the image. This can introduce black spaces in the corners or edges depending on the angle of rotation.
4. Gaussian Blur: Reduces sharpness and detail, resulting in a smoother appearance. This can help in reducing noise or the effect of minor imperfections.
5. Median Filter: Reduces noise without blurring edges significantly. This filter is particularly effective in preserving sharpness while removing speckle noise.

c. Where it may be useful and where not?

1. Histogram equalization: helps with improvement in visibility of images with poor contrast. Not useful in images where the contrast is already well balanced, as it can lead to unnatural effects.
2. Log Transformation: In situations when it's important to highlight dark areas without oversaturating brighter ones, like in imaging for medical purposes, less helpful in photos with a low dynamic range or consistent brightness.
3. Rotate Image: Practical for adjusting an image's orientation. Not helpful when maintaining picture integrity (e.g., corner details) or when the original orientation is preferred.
4. Gaussian Blur: Good for eliminating noise or producing special effects (like background blur) during pre-processing stages. Not helpful for tasks like text recognition that need to preserve a high level of information.
5. Median Filtering: In digital photography or when pre-processing an image for edge detection, the median filter is helpful for reducing noise in the image while maintaining the edges. Less effective when noise is dispersed evenly or when blurring in general.