POTHOLE DETECTION USING DIGITAL IMAGE PROCESSING TECHNIQUES

Avantika Balaji, Charitha Uppalapati, Kausalyaa Sri
3rd year - V semester, B.Tech Computer Science and Engineering,
Amrita School of Computing, Coimbatore
Tamil Nadu, India

1. ABSTRACT

Roads serve as a platform for transportation and contribute significantly to the economy. One of the main issues with transportation infrastructure is potholes on the roadways. Numerous studies have suggested employing computer vision techniques, such as various image processing and object detection algorithms, to automate pothole detection. Monitoring and maintaining road surfaces is critical for driving comfort, transportation safety, and infrastructure integrity. Traditional road condition monitoring is done on a regular basis by specially designed instrumented vehicles, which takes time and money and only covers a small portion of the road network. Pothole detection must be automated with sufficient speed and accuracy, and the system must be simple to use and inexpensive to set up. For this study, we collected data on road conditions in a city using dedicated vehicles and image sensors. We observe noisy images in our dataset commonly associated with electronic noise which could be generated by the digital camera's image sensor and circuitry. A number of processing methods were applied to the collected data, and features from various spatial and frequency domains, as well as image processing methods such as filtering, de-noising, and segmentation, were extracted. Our proposed system is based on an image processing method using a camera sensor to detect potholes based on their characteristics such as colour intensity of the region - dark or light, shape - oval, round, or irregular, and texture - smooth or rugged. The results showed that frequency domain features outperformed other features in identifying potholes. The X filter outperformed the other filters tested in terms of pothole identification, with a precision of 88.5% and a recall of 75%. Finally, we tested the proposed method's universality and robustness using datasets generated from various road types.

Keywords: pothole, image processing, detection, segmentation, filtering, identification, noise

2. PROBLEM STATEMENT

Potholes are highly hazardous road surface conditions that prevent safe, secure, and reliable transportation and movement of people, goods, and services. Potholes on the road surface endanger the safety and comfort of most drivers and commuters. Poor road networks impede the smooth movement of goods and services, contributing to the economy's slow growth and development, whereas good road networks provide access to markets and enable fast and smooth transportation of goods and services from producers to consumers. Early detection and maintenance of potholes contributes to the creation of a reliable and safe road network that allows for the smooth movement of people, goods, and services.
Our goal here is to develop a pothole detection system using various image processing techniques that will serve as a framework to assist drivers avoid potholes on poorly constructed roads. Our dataset consists of a collection of low-angle images captured by a camera that serves as an image sensor and is mounted in front of the vehicle. Pothole detection system using image processing techniques is a solution to automate the surveillance of roads (particularly potholes) to help speed up the process of road assessment, which would lead to safer roads and faster road maintenance.

3. INTRODUCTION

Road defects, such as potholes and cracks, are becoming an increasingly serious issue for roads worldwide. They endanger all road users and cause significant vehicle damage. In India, road accidents caused by potholes led to the death of 5,626 people between 2018 and 2020, according to the latest government data **[1]** Maintaining high-quality road infrastructure is difficult for a variety of reasons, including harsh weather, unexpected road loads, and erratic wear and tear. Mumbai, India's commercial capital, has spent over Rs 21,000 crore in the two decades since 1998 just for pothole repairs.**[2]** This is expensive, and many local governments are currently facing significant budgetary constraints, resulting in less frequent inspections that can only cover a limited portion of road networks.

A real-time pothole detection system installed in vehicles can also warn drivers of the presence of potholes ahead of time. This would assist drivers in avoiding potholes and preventing serious damage to their vehicles. The real-time detection system based on digital image processing proposed in this paper outperforms other previously proposed systems in the research.

4. METHODOLOGY:

The goal of this research is to create a method for detecting potholes that uses digital image processing techniques. The overall workflow is divided into four stages: **(1) data acquisition, (2) data processing, (3) feature extraction, and (4) categorisation.** First, data acquisition is carried out in order to collect pothole image data on the road. Second, a series of data-processing steps are taken to process the raw data collected in the first stage. Third, a series of transformations, features from various domains are extracted from potential pothole windows. Finally, to distinguish road defects, filtering and morphological processes for segmentation are used for final detection.

4.1 DATA ACQUISITION

One of the ultimate objectives of image acquisition is to have a source of input that operates within such controlled and measured parameters that the same image can, if necessary, be almost flawlessly recreated under the same conditions, making it easier to find and remove anomalous variables.

4.2 DATA PROCESSING

(a) Image Resizing

In this process we resize or distort an image from one pixel grid to another. Using the original pothole image may result in a longer computational time. As a result, the image must be resized to a lower resolution for faster processing and improved segmentation.

(b) Gray Scale Conversion

The average pixel values (ranging from 0-255) of the primary colours red, green, and blue (commonly known as RGB) are combined. Each colour band's luminous intensity (24 bits) is combined into a reasonable approximated grayscale value (8 bits). Here, color information doesn't help us identify important edges or other features. Hence we can consider it noise and reject it which in turn helps reduce color complexity. It also helps enhance visualisation as it can differentiate between shadow details and highlights of the pothole as it is majorly in 2D

(c) Image Enhancement

This process helps improve the quality and information content of original data prior to processing. Common techniques that have been tested include contrast enhancement and spatial filtering. We observe that this helps emphasize and sharpen the pothole border to obtain a more visually pleasing, more detailed and less noisy output image.

Using power-law transformation:

A log transformation of an image means replacing all of the image's pixel values with logarithmic values. Log transformation is used for image enhancement because it expands the dark pixels of the image as compared to higher pixel values.

$$S = c * r^{gamma}$$

where,
R = input pixel value,
C = scaling constant and
S = output pixel value

Inference: Low intensity values in the input image are mapped to a wider range of output levels

(d) Thresholding

We change the pixels of an image to make the image easier to analyze. It is a way to create a binary image from a grayscale image or a full-colour image. The process of thresholding involves comparing each pixel value of the image (pixel intensity) to a specified threshold. Thresholding allows us to separate the foreground (pothole) from the background (road) of our image. Using two-step thresholding procedure we can extract potential undamaged road areas from the transformed disparity map. In this way there is a clear bifurcation of the undamaged road and the appearing pothole.

(e) Histogram Equalization

The objective of this technique is to give a linear trend to the cumulative probability function associated to the image. It accomplishes this by effectively spreading out the most frequent intensity values, i.e. stretching out the intensity range of the image. Thus, our pothole region image quality can be improved by modifying its histogram.

(f) Image Sharpening

This emphasises an image's edges and fine details. Image sharpening is accomplished by adding a signal proportional to a high-pass filtered version of the image to the original image.
This step involves the use of unsharp masking done in 2 steps:

1. A high-pass filter extracts the high-frequency components from the original image first.
2. The original image is then sharpened by adding a scaled version of the high-pass filter output to it.

4.3 FEATURE EXTRACTION

Frequency Domain Filtering

We are using frequency filters to process the images in our dataset. Our image is first Fourier transformed, then multiplied by the filter function before being re-transformed into spatial domain. We obsserve that high frequency attenuation results in a smoother image in the spatial domain, while low frequency attenuation enhances the edges.

$$\boxed{\begin{aligned} F(u) &= \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux}\, dx, \\ f(x) &= \int_{-\infty}^{\infty} F(u) e^{j2\pi ux}\, du \end{aligned}}$$

Fourier Transform

We implemented the following frequency domain filters and compared and contrasted them in order to choose the best one.

1. LOW PASS FILTERING - IMAGE SMOOTHING TECHNIQUES

It smoothes the image by reducing high frequency components while preserving low frequency components. The amount of strength reduced for each frequency depends on the design of the filter.

a) Ideal Low Pass Filter - It eliminates high-frequency noise from a digital image while retaining low-frequency components. The transfer function of ILPF of order n is defined as:

$$\mathrm{H(u,\ v)}=\begin{cases} 1 & D(u,v) \le D_0 \\ 0 & D(u,v) > D_0 \end{cases}$$

where

- D0 is a positive constant
- ILPF passes all the frequencies within a circle of radius D0 from the origin without attenuation and cuts off all the frequencies outside the circle.
- Because D0 is the transition point between H(u, v) = 1 and H(u, v) = 0, it is referred to as the cutoff frequency.
- D(u, v) is the Euclidean Distance from any point (u, v) to the frequency plane's origin:

$$\mathrm{D(u,\ v)}=\sqrt{(u^2+v^2)}$$

b) Butterworth Low Pass Filter - It removes high-frequency noise with very minimal loss of signal components. The transfer function of BLPF of order n is defined as:

$$H(u,v) = \frac{1}{1+[D(u,v)/D_0]^{2n}}$$

c) Gaussian Low Pass Filter - It removes high frequency (sharp) features oriented along the X or Y axis. The practical result is a loss of detail or "blurring" effect on the image. The transfer function of Gaussian Lowpass filters is:

$$\boxed{H(u,v) = e^{-D^2(u,v)/2D_0^2} \quad (8)}$$

2. HIGH PASS FILTERING - IMAGE SHARPENING TECHNIQUES

A high pass filter tends to retain high frequency information while reducing low frequency information in an image. The high pass filter kernel is intended to increase the brightness of the centre pixel in comparison to neighbouring pixels.

a) Ideal High Pass Filter - it enhance the fine details and highlight the edges of potholes in the image. The transfer function of the IHPF can be specified by the function:

$$\mathrm{H(u,\ v)}=\begin{cases} 0 & D(u,v) \le D_0 \\ 1 & D(u,v) > D_0 \end{cases}$$

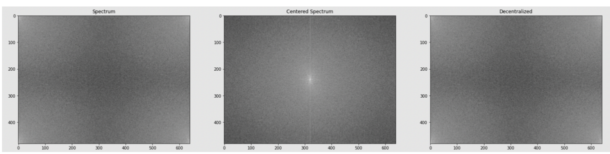
b) Butterworth High Pass Filter - It is the reverse operation of the Butterworth lowpass filter. The transfer function of BHPF of order n is defined as-

$$H(u,v) = \frac{1}{1+[D_0/D(u,v)]^{2n}}$$

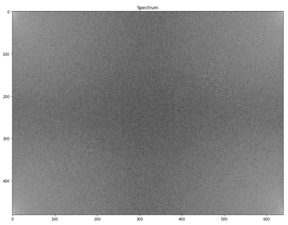
c) Gaussian High Pass Filter - It attenuates frequency components that are near to the image center (W/2, H/2). Transfer function of Gaussian High Pass Filter is given by:

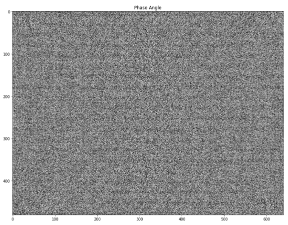
$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$$

The phase spectrum describes the phase of signal components as a function of frequency. Before passing filters and processing our image, we extract the phase spectrum from our original image and then get the centralised spectrum.



The phase angle values determine the shift in the image's sinusoid components. With zero phase, all of the sinusoids are centred at the same point, resulting in a symmetric image whose structure bears no resemblance to the original image at all.





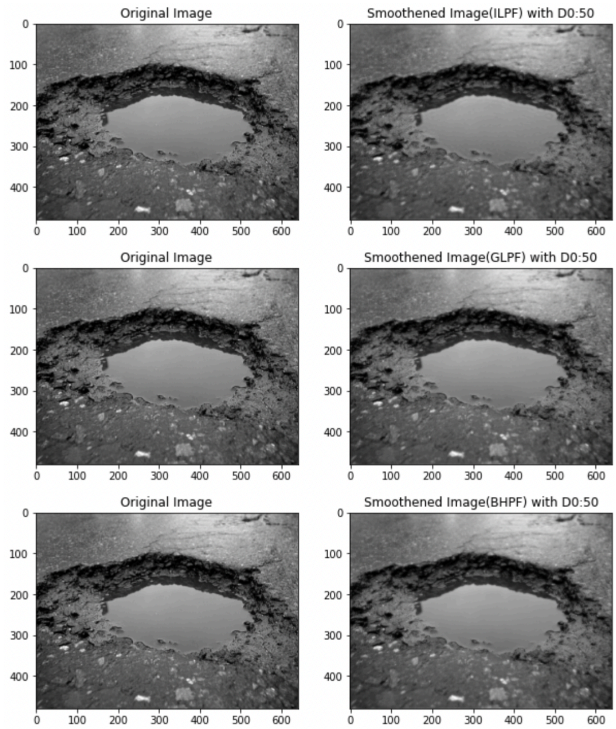
Following the implementation of each low pass and high pass filter with different threshold values (D0) ranging from sample multiples of 10: 10, 20, 30, 50, 100 for testing.

Threshold comparison:

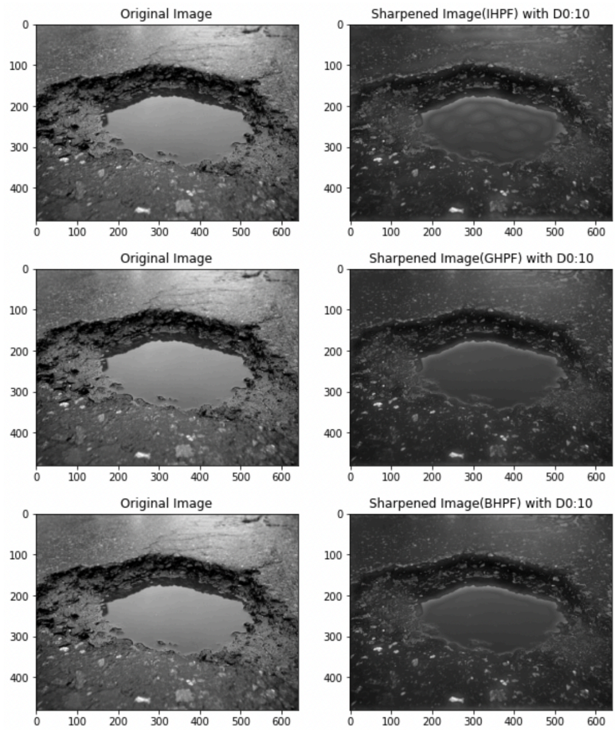
No	Filter applied	Best selected D0 value
1	Ideal Low Pass	50
2	Butterworth Low Pass	50
3	Gaussian Low Pass	50
4	Ideal High Pass	10
5	Butterworth High Pass	10
6	Gaussian High Pass	10

The table below summarises the best processed image obtained at the best fit threshold value for each filter used.

Output with Low Pass Filters - D0:50



Output with High Pass Filters - D0:10



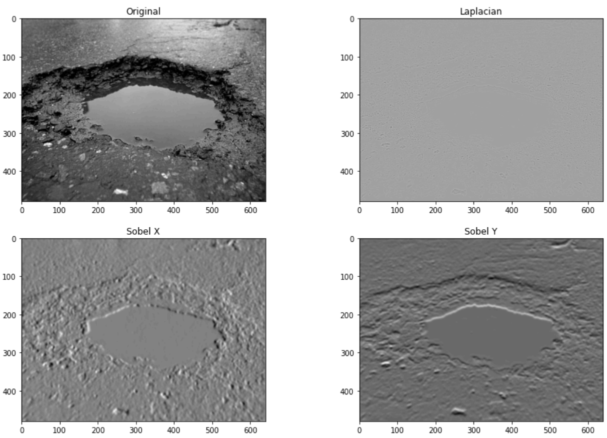
EDGE DETECTION

a) Laplacian Filter - It emphasises areas of rapid intensity change and is an example of a second order or second derivative enhancement method. It excels at detecting the fine details of the pothole's edges. Any feature with a sharp discontinuity will benefit from a Laplacian operator.
The Laplacian is a well-known linear differential operator that approximates the second derivative, which is given by:

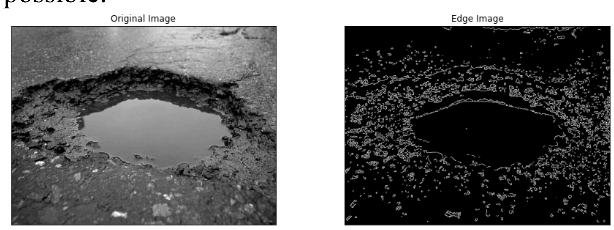
$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

where f denotes the image

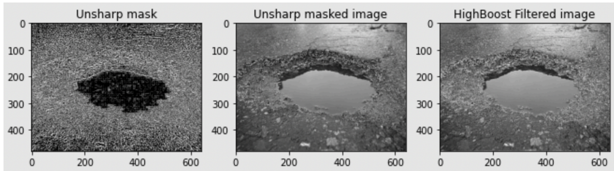
b) Sobel Edge Detection - When Sobel Edge Detection is used, the image is processed separately in the X and Y directions before being combined to form a new image that represents the sum of the X and Y edges of the image. We first convert the image from RGB to Grayscale. Then we use a technique known as kernel convolution. A kernel is a 3 x 3 matrix made up of symmetrically (or differently) weighted indexes. This will be the filter that we will implement for edge detection.



c) Canny Edge Detection - It uses a multi-stage algorithm to detect a wide range of edges in images. We get a good detection: the pothole can be easily distinguished from the background because the optimal detector eliminates the possibility of false positives and false negatives.
There is also the advantage of good localisation: the detected edges are as close to true edges as possible.

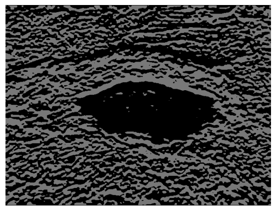


d) Unsharp Masking and High Boost Filtering - It enhances edges (and other high frequency components in an image) by subtracting an unsharpened, or smoothed, version of an image from the original image.



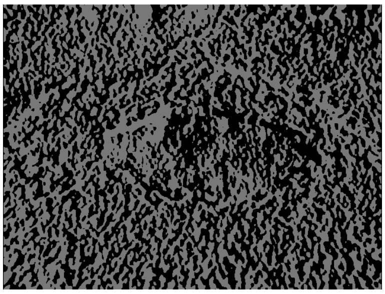
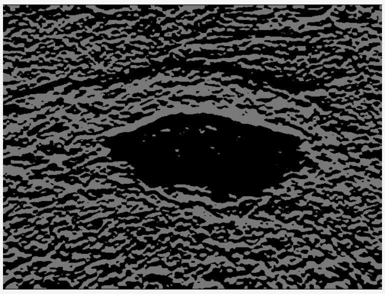
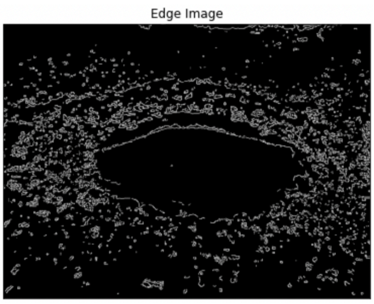
Laplacian vs Sobel:

Between Laplacian and Sobel, we see that Sobel produces a better result. Unlike the Sobel edge detector, the Laplacian edge detector uses only one kernel. The Laplacian filter detects sudden intensity transitions in the image and highlights the edges. However, its main disadvantage is its high sensitivity to noise, which causes edges to spread with smoothing. It amplifies the noise in the image, as we observe, hence it is not the best filter when we have noisy images.

	
LAPLACIAN	SOBEL

Here's a table that summarises the results of various edge detection filters.

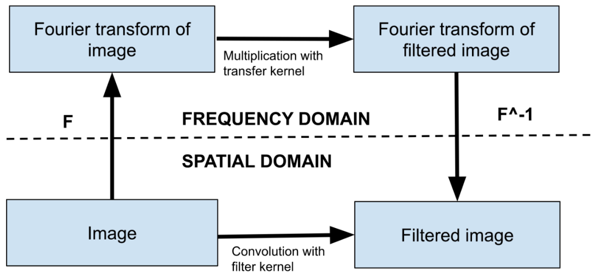
No	Filter	Edge Detection using filter

1	Laplacian	
2	Sobel - x	
3	Sobel - y	
4	Canny Edge	

(D) CATEGORISATION

4.3 <u>FREQUENCY DOMAIN VS SPATIAL DOMAIN FILTERING</u>

Rather than filtering the image in the spatial domain by convolving it with the filter kernel, the image can be transformed into frequency space using the FFT (indicated by the symbol F). The image's Fourier transform is then multiplied by the filter function (also known as the transfer function, i.e., the convolution kernel's Fourier transform), and the resulting frequency-space filtered image is subjected to the inverse Fourier transform (F^-1). Although this "detour" into the frequency domain appears to be extra work, the overall computational cost is significantly lower than that of a convolution with large kernel sizes.



This is especially true as the filter size increases. While spatial filters are commonly used for edge detection, frequency filters are used for high frequency emphasis. In this case, the filter does not completely block low frequencies, but rather magnifies high frequencies in comparison to low frequencies, thus projecting the potholes more clearly. We see that frequency domain filtering produces a much better filtered output than spatial domain filtering.

Frequency Domain Filtering	**Spatial Domain Filtering**
	

<u>NOISE</u>

Image noise is the random variation of brightness or colour information in captured images. It could be caused by external sources such as wind, rain, or dust settling on the lens of the camera mounted on the vehicle, which degrades the image signal. Noise is created in the image during the acquisition process or is added to the image during transmission over a wired or wireless medium.

We explored some of the following noise models:

1. Spatially independent noise models

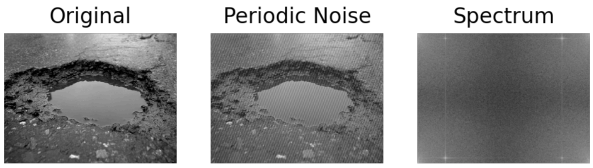
- Gaussian noise
- Speckle
- Uniform noise
- Impulse (salt-and-pepper) noise
- Rayleigh noise
- Erlang (Gamma) noise
- Exponential noise

2. Spatially dependent noise model

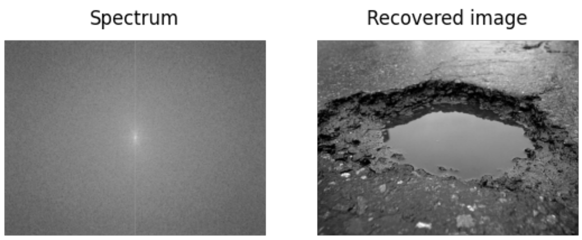
- Periodic noise

<u>NOISE MODELS</u>

1. **Periodic Noise -** We see that it creates strips in an order on an image in the spatial domain. It interferes with our original image with the pothole at a random frequency. We are using a notch filter to remove this noise. Since it is a band reject filter, it rejects the specific circular bands of frequencies around the frequency domain image's centre because frequency domain images are symmetric about opposite quadrants. It helps reject not only the central noisy peak but also the noisy frequency areas corresponding to noisy spikes. Our recovered image has removed the noise and is slightly better than the original.

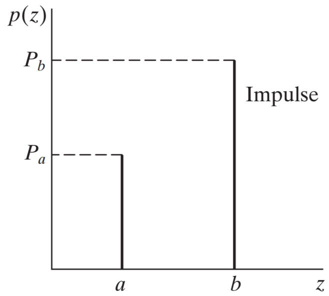


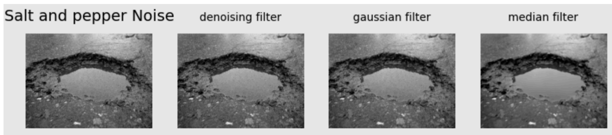
After applying Notch Filter



2. **Salt and Pepper Noise -** We add salt and pepper noise to an image by randomly scattering bright (255 pixel value) and dark (0 pixel value) pixels across it. Since it statistically drops the original data values, this model is also known as data drop noise. We apply denoising, gaussian and median filter to this noisy image and compare which gives us the best result.

$$p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases}$$

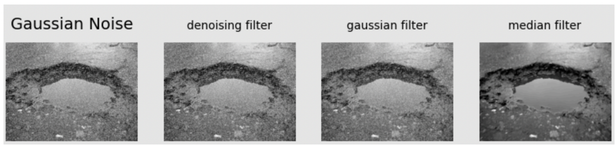




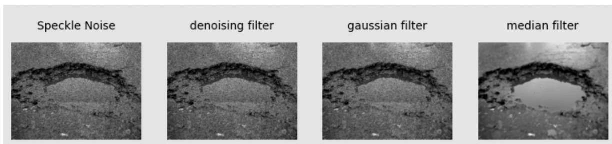
3. **Gaussian Noise -** We use the Gaussian probability distribution function to generate Gaussian noise on images. Our Gaussian probability distribution function is represented by the formula below.

$$P(y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y-\mu)^2}{2\sigma^2}}$$

Here's the result on applying the respective filters.



4. **Speckle Noise -** It is the granular noise seen in our images that exists inherently and degrades their quality. Speckle noise can be created by multiplying random pixel values by different image pixels.

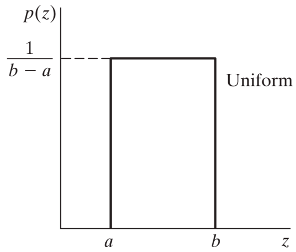


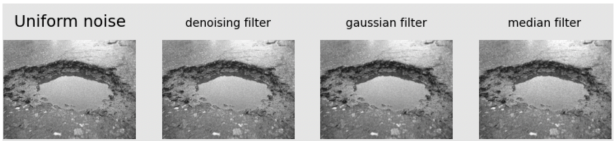
5. **Uniform Noise -** Quantization noise is the uniform noise caused by quantizing the pixels of an image to a number of distinct levels. The level of the grey values of the noise is uniformly distributed across a specified range in uniform noise. This noise is frequently used to degrade images in order to evaluate image restoration algorithms.

Permutation of a probability distribution function to distribute noise pixel intensity values:

$$p(z) = \begin{cases} \dfrac{1}{b-a} & \text{if } a \leq z \leq b \\ 0 & \text{otherwise} \end{cases}$$

Parameters a and b are essentially the intensity boundaries between which we distribute uniform noise PDF.





DE-NOISING METHODS

1. **Notch Filter -** It is a narrow band reject filter used to remove a single frequency or a narrow band of frequencies. Notch filters can be created from a combination of high-pass and low-pass filters. This filter rejects frequencies in specific neighbourhoods surrounding a centre frequency. The power spectrum densification of an image is used to detect noise spikes visually.

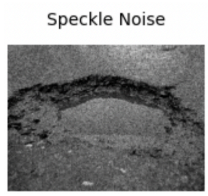
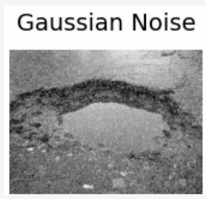
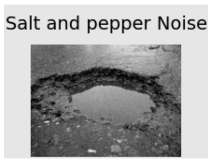
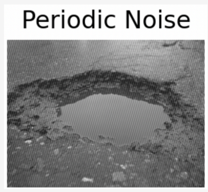
2. **Median Filter** - The median filter examines each pixel in the image individually and compares it to its neighbours to determine whether or not it is representative of its surroundings. It replaces the pixel value with the median of neighbouring pixel values rather than the mean of those values.

3. **Gaussian Filter -** Gaussian filtering, g, is used to blur images and remove noise and detail. In one dimension, the Gaussian function is:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

4. **Non Local Means Denoising Filter** - The non-local means algorithm replaces a pixel's value with an average of the values of a subset of other pixels: small patches centered on the other pixels are compared to the patch centered on the pixel of interest, and the average is performed only for pixels with patches close to the current patch. As a result, this algorithm can effectively restore textures that would otherwise be blurred by other denoising algorithms.

Conclusion of the best de-noising method used to remove the noise in each noise model

Noise Type	Noise Image	De-Noising Method
Uniform		
Speckle		
Gaussian		
Impulse (Salt and Pepper)		
Periodic (spatially dependent)		 Notch Filter

4.4.1 MORPHOLOGICAL IMAGE PROCESSING

Morphological image processing is a technique used to modify the shape and size of structures in an image. It is a non-linear image processing technique that is based on the shape of the objects in an image, rather than their intensity values. Morphological image processing is typically used to extract specific features or objects from an image, such as lines, edges, or text. It can also be used to fill in gaps or to remove noise or other undesirable features from an image.

Morphological techniques probe an image with a small shape or template called a **structuring element**. The structuring element is positioned at all possible locations in the image and it is compared with the corresponding neighbourhood of pixels. Some operations test whether the element "fits" within the neighbourhood, while others test whether it "hits" or intersects the neighbourhood:

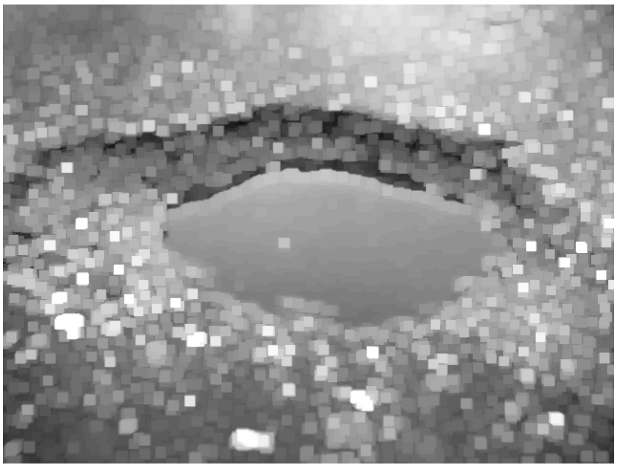
- The matrix dimensions specify the size of the structuring element.
- The pattern of ones and zeros specifies the shape of the structuring element.
- An origin of the structuring element is usually one of its pixels, although generally the origin can be outside the structuring element.

Structuring elements play in morphological image processing the same role as convolution kernels in linear image filtering.

There are several basic morphological operations that are commonly used in image processing, including **erosion, dilation, opening, and closing**. Erosion and dilation are used to modify the size of objects in an image, while opening and closing are used to remove small isolated pixels or fill in gaps between objects. Top-hat and black-hat transforms are operations used in morphology and digital image processing to extract small elements and details from given images.

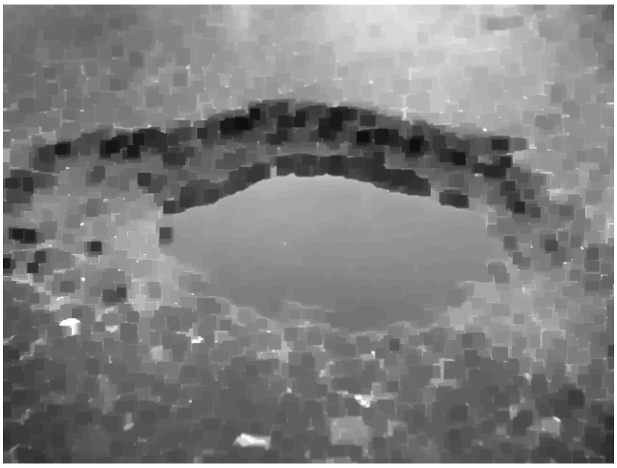
1. **Erosion**: Erosion is an operation that removes pixels from the boundaries of objects in an image. It is typically used to reduce the size of objects or to remove small, isolated pixels that may be present due to noise or other factors. Erosion is often used in combination with dilation to improve the performance of image analysis and object recognition algorithms.

2. **Dilation**: Dilation is the opposite of erosion and is used to add pixels to the boundaries of objects in an image. It is typically used to increase the size of objects or to fill in gaps between objects. Like erosion, dilation is often used in combination with other techniques to improve the performance of image analysis algorithms.



3. **Opening**: Opening removes small objects from the foreground (usually taken as the bright pixels) of an image, placing them in the background. Opening is similar to erosion as it tends to remove the bright foreground pixels from the edges of regions of foreground pixels. Opening operation is used for removing internal noise in an image. Opening is erosion operation followed by dilation operation.

4. **Closing:** On the other hand it removes small holes in the foreground, changing small islands of background into foreground. It is defined simply as a dilation followed by an erosion using the same structuring element used in the opening operation.



5. **Top Hat Filter:** It is defined as the difference between the input image and its opening by some structuring element. The top-hat filter is used to highlight bright objects against a dark background.

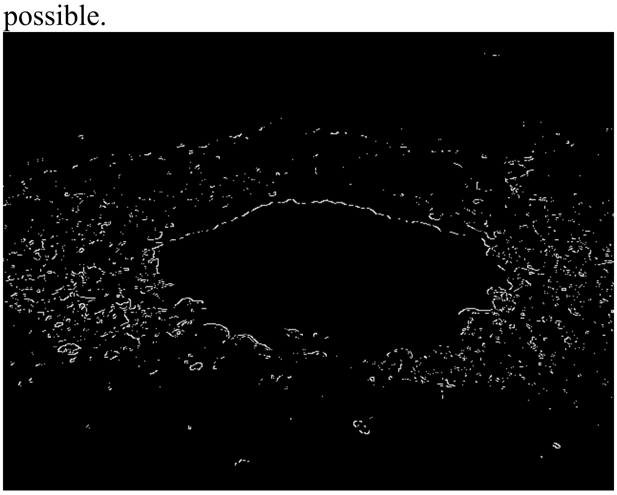
6. **Black Hat Filter**: It is defined as the difference between the closing and the input image. These transforms are used in a variety of image processing tasks, including feature extraction, background equalisation, image enhancement, and so on. The black-hat operation is used to enhance dark objects of interest against a bright background.

4.4.2 <u>EDGE DETECTORS</u>

1. **CANNY EDGE DETECTOR**

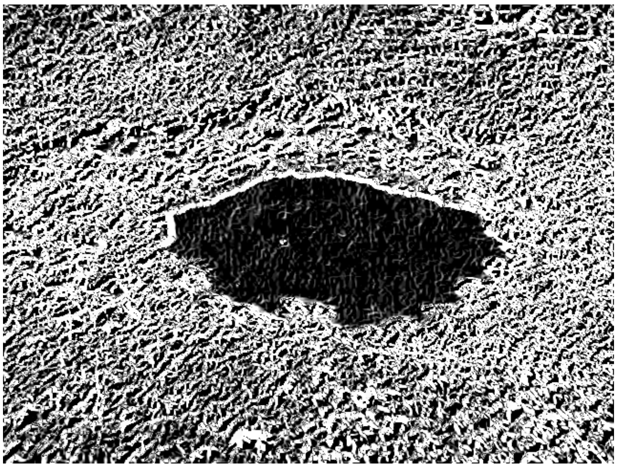
It is a multi-stage algorithm with steps involved:
1. Removal of noise in input image using a Gaussian filter.
2. To find intensity gradient of the image - Compute the derivative of Gaussian filter
3. Suppress the non-max edge contributor pixel points. This way we remove unwanted pixels which is not a part of the edge
4. Use Hysteresis Thresholding method to determine the true edges in the image.

The pothole can be easily distinguished from the background because the optimal detector eliminates the possibility of false positives and false negatives. There is also the advantage of good localisation: the detected edges are as close to true edges as possible.



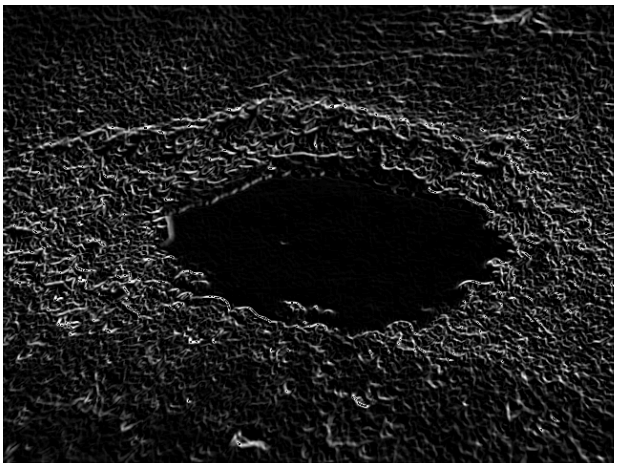
2. SOBEL EDGE DETECTOR

For edge detection, we use matrix to calculate areas of different intensities of an image. And wherever we find extreme differences in the intensities of the pixel, it means we have found the edge of an object. Here the image is processed separately in X and Y directions first, then the sum of the X and Y edges are taken which forms the combined output image. This works best when we first convert the image from RGB to standardised Grayscale since the Luminance plane of grayscale has far more important and distinct edge information. Then we do kernel convolution using a 3x3 matrix. So we basically run a 3x3 matrix kernel over all the pixels in the image. At every iteration, we measure the change in the gradient of the pixels that fall within this 3x3 kernel. The greater the change in pixel intensity, the more significant the edge there is.



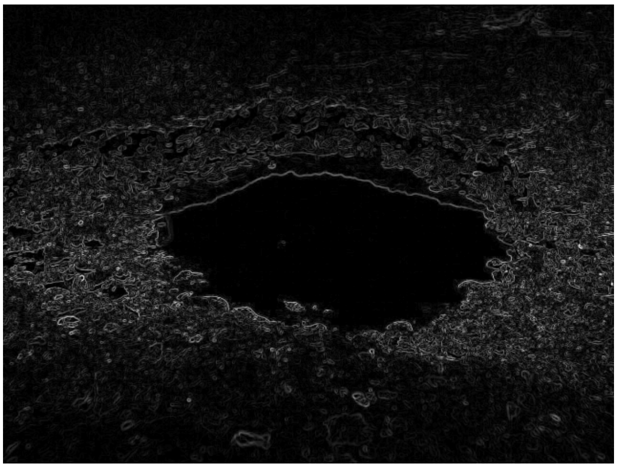
3. PREWITT EDGE DETECTOR

Prewitt mask is a first-order derivative mask. This also works along the X and Y-axis separately. Wherever there is a sudden change in pixel intensities, an edge is detected by the mask. This edge can be calculated by using differentiation. The edge is represented by the local maxima or local minima.



4. ROBERT EDGE DETECTOR

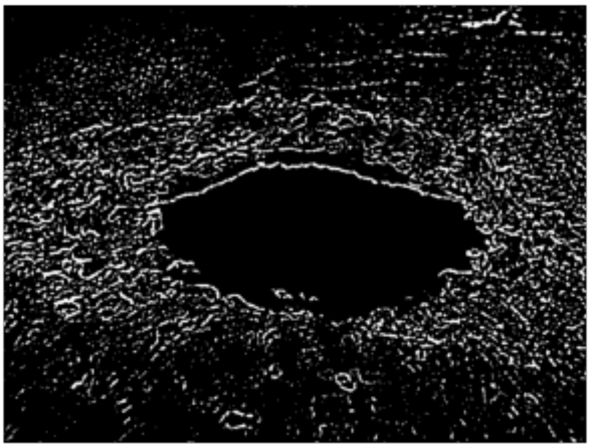
It thus highlights regions of high spatial frequency which often correspond to edges. As a differential operator, which approximates the gradient of an image through discrete differentiation which is achieved by computing the sum of the squares of the differences between diagonally adjacent pixels .



5. LAPLACIAN OF GAUSSIAN - MARR-HILDRETH EDGE DETECTOR

In Marr-Hildreth we first, we use a Gaussian filter on the noisy image to smoothen it instead of just simple averaging and then subsequently use the Laplacian filter for edge detection. The image is first filtered with the Laplacian of the Gaussian filter matrix, which is computed using the standard deviation value input for Marr-Hildreth edge detection. The standard deviation value determines the filter matrix width and controls the amount of smoothing produced by the Gaussian component. The Gaussian filtering's Laplacian will smooth the image and enhance the edges. Edge localization is performed after filtering by locating zero crossings at each pixel in all directions.

LOG



4.4.3 IMAGE SEGMENTATION

The process of dividing an image into multiple segments or regions, each of which corresponds to a different object or background in the image, is known as image segmentation. The goal of image segmentation is to simplify and/or change an image's representation into something more meaningful and understandable.

Image segmentation can be accomplished using a variety of techniques such as:

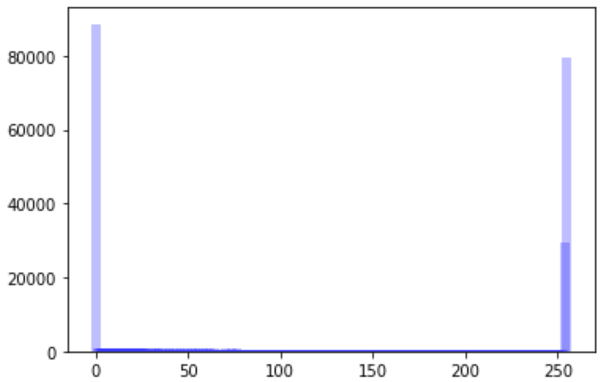
- Threshold Based Segmentation
- Edge Based Segmentation
- Region-Based Segmentation
- Clustering Based Segmentation
- Artificial Neural Network Based Segmentation

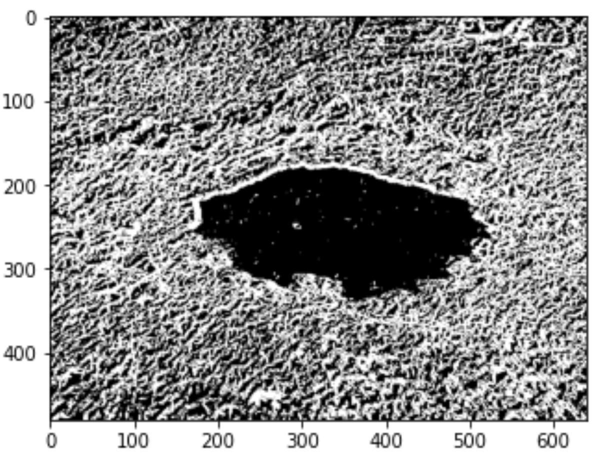
1. **Thresholding Techniques:**

There are several thresholding techniques:

Global thresholding: We use a bimodal image in this method. A bimodal image is one with two intensity peaks in the intensity distribution plot. One is for the object, and the other is for the background. Then we calculate the global threshold for the entire image and apply it to the entire image.

Optimal Thresholding: Optimal thresholding technique can be used to minimize the misclassification of pixels performed by segmentation.

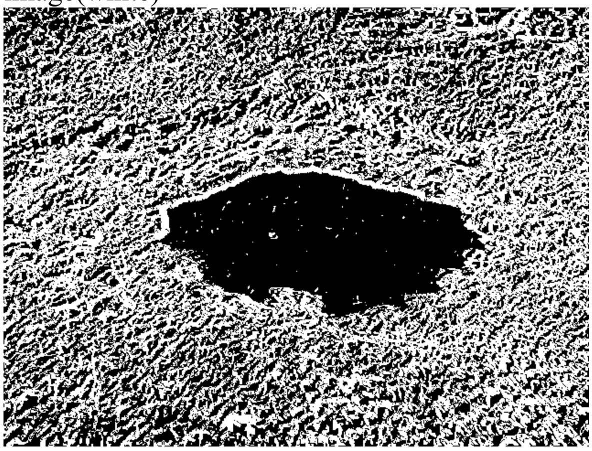




One of the ways to achieve an optimal threshold is Otsu's method. In this method, we find the spread of foreground and background of the pictures for all possible values of threshold. The threshold with the least spread is taken as the optimal threshold

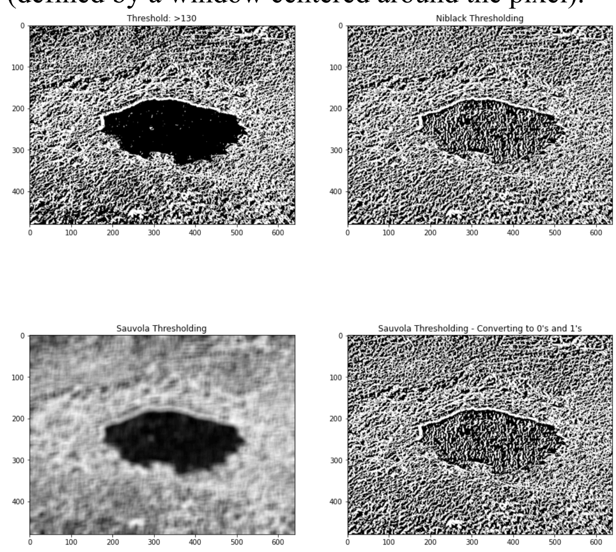
1. **Otsu's algorithm**

The algorithm iteratively searches for the threshold that minimizes the within-class variance, defined as a weighted sum of variances of the two classes (background and foreground). The colors in grayscale are usually between 0-255 (0-1 in case of float). So, If we choose a threshold of 100, then all the pixels with values less than 100(black) becomes the background and all pixels with values greater than or equal to 100 becomes the foreground of the image(white)



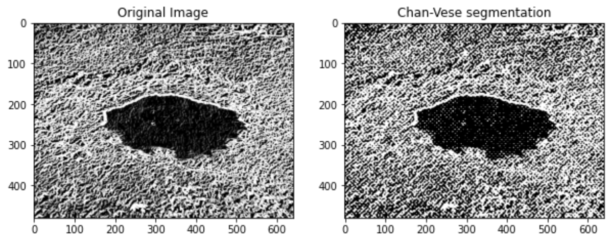
2. **Niblack and Sauvola thresholding**

Niblack and Sauvola thresholds are local thresholding techniques that are useful for images where the background is not uniform, especially for text recognition 1, 2. Instead of calculating a single global threshold for the entire image, several thresholds are calculated for every pixel by using specific formulae that take into account the mean and standard deviation of the local neighborhood (defined by a window centered around the pixel).



3. Chan-Vese Segmentation Algorithm

The Chan-Vese segmentation algorithm is designed to segment objects without clearly defined boundaries. This algorithm is based on level sets that are evolved iteratively to minimize an energy, which is defined by weighted values corresponding to the sum of differences intensity from the average value outside the segmented region, the sum of differences from the average value inside the segmented region, and a term which is dependent on the length of the boundary of the segmented region.



5. EVALUATION METRICS

We have evaluated our algorithms and filters using two of the following evaluation metrics:

1. **PSNR Ratio** - The PSNR block computes the peak signal-to-noise ratio, in decibels, between two images. This ratio is used as a quality measurement between the original and a compressed image. The higher the PSNR, the better the quality of the compressed, or reconstructed image.
2. **SSIM Index** - The Structural Similarity Index (SSIM) is a perceptual metric that quantifies image quality degradation caused by data compression or transmission losses. It is a full reference metric that necessitates the use of two images from the same image capture—a reference image and a processed image.

6. RESULTS AND CONCLUSION

Robert edge detector gives us a better similarity index of 0.0921 and a clear demarcation of the edges compared to the other edge detectors. However the PSNR of Canny edge detector is the best for filtering the edges with a value of 27.948. We observe that Sauvola thresholding produces a much better SSIM value of indicating a more structurally segmented image with a higher similarity to a pothole.

This can be combined with machine learning models used for classification such as decision tree, SVM, logistic regression in order to extract features from the frequency domain.

7. REFERENCES

[1] Ministry of Road Transport and Highways (MoRTH) data - https://www.ndtv.com/india-news/over-5-000-killed-in-road-accidents-caused-by-potholes-in-2018-20-transport-ministry-3276432

[2] The political economy of India's potholes, The New Indian Express, September 2022 https://www.newindianexpress.com/opinions/2022/sep/25/the-political-economy-of-indiaspotholes-2501582.html#:~:text=Such%20is%20the%20persistence%20of,was%20over%20Rs%201500%20crore

[3] An Automated Machine-Learning Approach for Road Pothole Detection Using Smartphone Sensor Data, Article by Chao Wu; Published: 28 September 2020

[4] https://towardsdatascience.com/image-segmentation-part-1-9f3db1ac1c50

[5] https://www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/topic4.htm