USN

Internal Assessment Test 2 – November 2024

Sub:	Digital Image Processing					Sub Code:	21EC722	Branch:	ECE
Date:	20-11-2024	Duration:	90 min's	Max Marks:	50	Sem / Sec:	7 – A, B, C, D		OBE

Answer any FIVE FULL Questions	MARKS	CO	RBT
1. Explain point operation,mask operation and global operation.	[10]	CO3	L1
2. Explain histogram equilization method of image enhancement.	[10]	CO3	L1
3. Explain the non linear gray level transfromations.	[10]	CO3	L2
4. Explain in brief types of smoothening filters.	[10]	CO4	L2
5. Discuss the following with respect to image enhancement I) High boost filtering II) Un sharp masking	[10]	CO4	L2
6. Expalin the method of Homomorphic filtering in image enhancement.	[10]	CO4	L2
7. Explain about different edge detection methods.	[10]	CO3	L3
8. What is LOG filter?How LOG is advantageous than laplacian filter?	[10]	CO3	L3

USN

Internal Assessment Test 2 – November 2024

Sub:	Network Security					Sub Code:	21EC722	Branch:	ECE
Date:	16-10-2024	Duration:	90 min's	Max Marks:	50	Sem / Sec:	7– A, B, C, D		OBE

Answer any FIVE FULL Questions	MARKS	CO	RBT
1. Explain point operation,mask operation and global operation.	[10]	CO3	L1
2. Explain histogram equilization method of image enhancemnt.	[10]	CO3	L1
3. Explain the non linear gray level transfromations.	[10]	CO3	L2
4. Explain in brief types of smoothening filters.	[10]	CO4	L2
5. Discuss the following with respect to image enhancement I)High boost filtering II)Un sharp masking	[10]	CO4	L2
6. Expalin the method of Homomorphic filtering in image enhancement.	[10]	CO4	L2
7. Explain about different edge detection methods.	[10]	CO3	L3
8. What is LOG filter?How LOG is advantageous than laplacian filter?	[10]	CO3	L3

1.Explain point operation,mask operation and global operation.

point operations are a common image processing technique that modifies the value of individual pixels without affecting neighboring pixels. They are often used to enhance or pre-process images.

Some examples of point operations include:

- **Contrast stretching**: A common point operation
- **Thresholding**: A common point operation
- **Gamma correction**: A common point operation
- **Noise reduction**: A common point operation
- **Automatic contrast adjustment**: A point operation that maps the highest and lowest intensities in an image

Point operations do not change the shape or spatial position of objects in an image. Instead, they increase, decrease, or leave unchanged the value or gray level of each pixel.

Masking is a photo editing technique that allows users to isolate or separate specific areas of an image for more precise editing. It's similar to placing a mask over the parts of an image that you want to hide or protect while exposing the other areas for editing.

Masking can be used for a variety of purposes, including:

- **Removing backgrounds**: Layer masking can be used to remove the background of an image, which can be useful for product photography.
- **Selective editing**: Layer masking can be used to selectively apply edits to specific parts of an image, such as changing the color of a product in an e-commerce photo.
- **Blending layers**: Masking can be used to blend layers.
- **Applying filters**: Masking can be used to apply filters to specific parts of an image.
  - In image processing, a global operation is a filter that applies the same basic process to every pixel in an image, regardless of their position. The output

value of any point in an image is dependent on all the pixels in the input image.

- Some examples of global operations include: Brightening and darkening, Contrast, Color corrections, Equalization, and Pseudocoloring.
- Global operations are more difficult to parallelize than local operations, which are easier to parallelize because the input data can be divided among processors. However, some global operations can be computed in parallel using a restricted form of divide and conquer called split and merge

2. Explain histogram equilization method of image enhancement.

A digital image is a two-dimensional matrix of two spatial coordinates, with each cell specifying the intensity level of the image at that point. So, we have an N x N matrix with integer values ranging from a minimum intensity level of 0 to a maximum level of L-1, where L denotes the number of intensity levels. Hence, the intensity levels of a pixel *r* can take on values from 0,1,2,3,…. (L-1). Generally, L = 2m, where m is the number of bits required to represent the intensity levels. Zero level intensity denotes complete black or dark, whereas L-1 level indicates complete white or absence of grayscale.

Intensity Transformation:

Intensity transformation is a basic digital image processing technique, where the pixel intensity levels of an image are transformed to new values using a mathematical transformation function, so as to get a new output image. In essence, intensity transformations is simply to implement the following function:

s=T(r)$s=T(r)$

where s is the new pixel intensity level and r is the original pixel intensity value of the given image and $r \geq 0$.

With different forms of the transformation function *T(r)*, we get different output images.

Common Intensity Transformation Functions:

1. Image negation: This reverses the grayscales of an image, making dark pixels whiter and white pixels darker. This is completely analogous to the photographic negative, hence the name.

s=L–1–r$s=L-1-r$

2. Log Transform: Here *c* is some constant. It is used for expanding the dark pixel values in an image.

s=clog(1+r)$s=clog(1+r)$

3. Power-law Transform: Here *c* and *γ* are some arbitrary constants. This transform can be used for a variety of purposes by varying the value of *γ*.

s=crγ $s=cr\gamma$

Histogram Equalization:

The histogram of a digital image, with intensity levels between 0 and (L-1), is a function $h(r_k) = n_k$, where r_k is the kth intensity level and n_k is the number of pixels in the image having that intensity level. We can also normalize the histogram by dividing it by the total number of pixels in the image. For an N x N image, we have the following definition of a normalized histogram function:

$$p(r_k)=n_k/N^2$$

This $p(r_k)$ function is the probability of the occurrence of a pixel with the intensity level r_k. Clearly,

$$\sum p(r_k)=1$$

The histogram of an image, as shown in the figure, consists of the x-axis representing the intensity levels r_k and the y-axis denoting the $h(r_k)$ or the $p(r_k)$ functions.

The histogram of an image gives important information about the grayscale and contrast of the image. If the entire histogram of an image is centered towards the left end of the x-axis, then it implies a dark image. If the histogram is more inclined towards the right end, it signifies a white or bright image. A narrow-width histogram plot at the center of the intensity axis shows a low-contrast image, as it has a few levels of grayscale. On the other hand, an evenly distributed histogram over the entire x-axis gives a high-contrast effect to the image.

In image processing, there frequently arises the need to improve the contrast of the image. In such cases, we use an intensity transformation technique known as histogram equalization. Histogram equalization is the process of uniformly distributing the image histogram over the entire intensity axis by choosing a proper intensity transformation function. Hence, histogram equalization is an intensity transformation process.

The choice of the ideal transformation function for uniform distribution of the image histogram is mathematically explained below.

Mathematical Derivation of Transformation Function for Histogram Equalization:

Let us consider that the intensity levels of the image *r* is continuous, unlike the discrete case in digital images. We limit the values that *r* can take between 0 and L-1, that is, $0 \leq r \leq L-1$. $r = 0$ represents black and $r = L-1$ represents white. Let us consider an arbitrary transformation function:

$$s=T(r)$$

where s denotes the intensity levels of the resultant image. We have certain constraints on $T(r)$.

- $T(r)$ must be a strictly increasing function. This makes it an injective function.

- *$0 \le T(r) \le L-1$*. This makes *T(r)* surjective.

The above two conditions make *T(r)* a bijective function. We know that such functions are invertible. So we can get back r values from s. We can have a function such that $r = T^{-1}(s)$

Let us now say that the probability density function (pdf) of *r* is *$p_r(x)$* and the cumulative distribution function (CDF) of *r* is *$F_r(x)$*. Now the CDF of *s* will be :

$$F_S(x)=P(s\le x)=P(T(r)\le x)=P(r\le T^{-1}(x))=F_r(T^{-1}(x)).$$

We put the first condition of *T(r)* precisely to make the above step hold true. The second condition is needed as *s* is the intensity value for the output image and so must be between o and (L-1).

So, a pdf of s can be obtained by differentiating *$F_S(x)$* with respect to x. We get the following relation:

$$p_s(s)=p_r(r)\frac{dr}{ds}$$

Now, if we define the transformation function as follows:

$$s=T(r)=(L-1)\int_0^r p_r(x)\,dx$$

Then using this function gives us a uniform pdf for *s*.

$$\frac{ds}{dr}=(L-1)\frac{d}{dr}\int_0^r p_r(x)\,dx=(L-1)p_r(r)$$

The above step used Leibnitz's integral rule. Using the above derivative, we get:

$$p_s(s)=p_r(r)\frac{dr}{ds}=p_r(r)\frac{1}{(L-1)p_r(r)}=\frac{1}{L-1}$$

So the pdf of *s* is uniform. This is what we want.

Now, we extend the above continuous case to the discrete case. The natural replacement of the integral sign is the summation. Hence, we are left with the following histogram equalization transformation function.

$$s_k=T(r_k)=(L-1)\sum_{j=0}^{k}p_r(r_j)=\frac{(L-1)}{N^2}\sum_{j=0}^{k}n_j$$

Since s must have integer values, any non-integer value obtained from the above function is rounded off to the nearest integer.

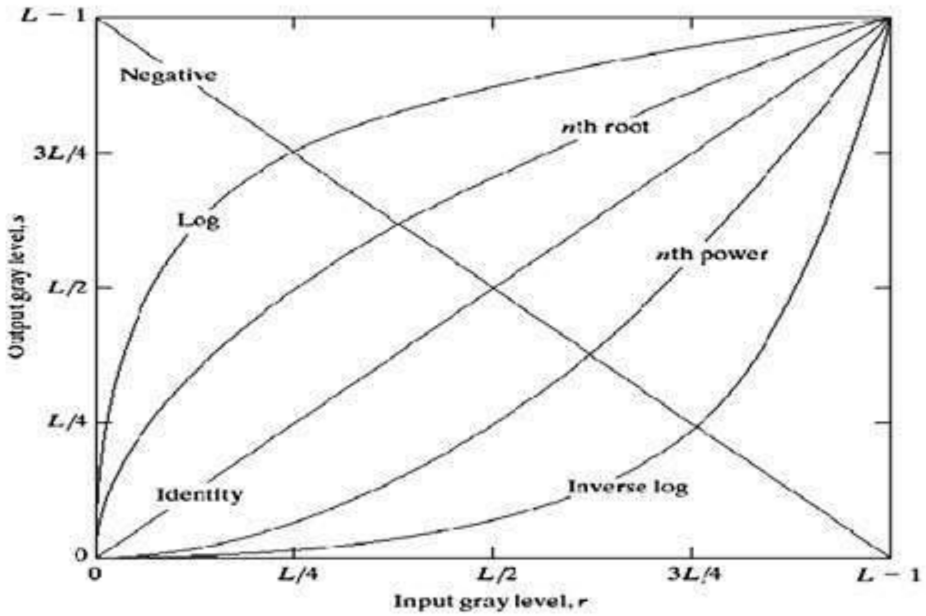
3. Explain the non linear gray level transfromations.

Gray level transformation

There are three basic gray level transformation.

- Linear
- Logarithmic
- Power – law

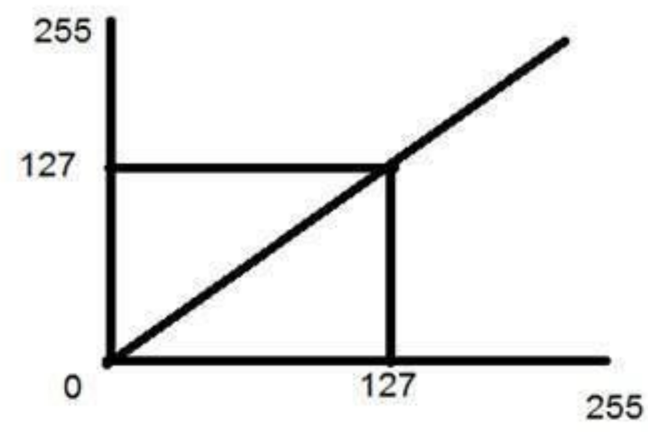
The overall graph of these transitions has been shown below.



Linear transformation

First we will look at the linear transformation. Linear transformation includes simple identity and negative transformation. Identity transformation has been discussed in our tutorial of image transformation, but a brief description of this transformation has been given here.

Identity transition is shown by a straight line. In this transition, each value of the input image is directly mapped to each other value of output image. That results in the same input image and output image. And hence is called identity transformation. It has been shown below:

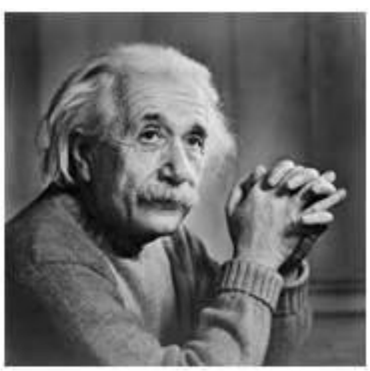


Negative transformation

The second linear transformation is negative transformation, which is invert of identity transformation. In negative transformation, each value of the input image is subtracted from the L-1 and mapped onto the output image.

The result is somewhat like this.

Input Image



Output Image

In this case the following transition has been done.

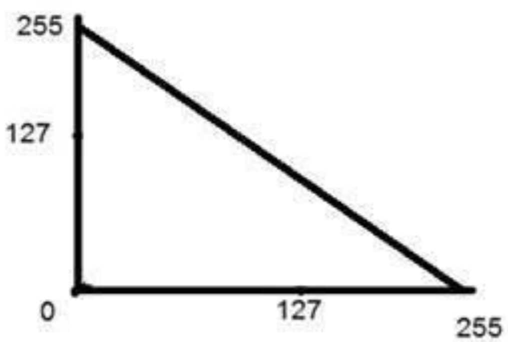
s = (L – 1) – r

since the input image of Einstein is an 8 bpp image, so the number of levels in this image are 256. Putting 256 in the equation, we get this

s = 255 – r

So each value is subtracted by 255 and the result image has been shown above. So what happens is that, the lighter pixels become dark and the darker picture becomes light. And it results in image negative.

It has been shown in the graph below.



Logarithmic transformations

Logarithmic transformation further contains two type of transformation. Log transformation and inverse log transformation.

Log transformation

The log transformations can be defined by this formula

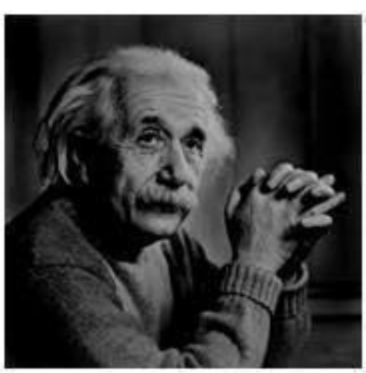
$s = c \log(r + 1)$.

Where s and r are the pixel values of the output and the input image and c is a constant. The value 1 is added to each of the pixel value of the input image because if there is a pixel intensity of 0 in the image, then log (0) is equal to infinity. So 1 is added, to make the minimum value at least 1.

During log transformation, the dark pixels in an image are expanded as compare to the higher pixel values. The higher pixel values are kind of compressed in log transformation. This result in following image enhancement.

The value of c in the log transform adjust the kind of enhancement you are looking for.

Input Image



Log Tranform Image

The inverse log transform is opposite to log transform.

Power – Law transformations

There are further two transformation is power law transformations, that include nth power and nth root transformation. These transformations can be given by the expression:

s=cr^γ

This symbol γ is called gamma, due to which this transformation is also known as gamma transformation.

Variation in the value of γ varies the enhancement of the images. Different display devices / monitors have their own gamma correction, that's why they display their image at different intensity.

This type of transformation is used for enhancing images for different type of display devices. The gamma of different display devices is different. For example Gamma of CRT lies in between of 1.8 to 2.5, that means the image displayed on CRT is dark.

Correcting gamma.

s=cr^γ

s=cr^(1/2.5)

The same image but with different gamma values has been shown here.

For example

Gamma = 10

Gamma = 8

Gamma = 6

4.Explain in brief types of smoothening filters.

Smoothing Spatial Filter

Smoothing filter is used for blurring and noise reduction in the image. Blurring is pre-processing steps for removal of small details and Noise Reduction is accomplished by blurring.

Types of Smoothing Spatial Filter

1. Linear Filter (Mean Filter)

2. Order Statistics (Non-linear) filter

These are explained as following below.

1. **Mean Filter:** Linear spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask. The idea is replacing the value of every pixel in an image by the average of the grey levels in the neighborhood define by the filter mask. Below are the types of mean filter:
   - **Averaging filter:** It is used in reduction of the detail in image. All coefficients are equal.
   - **Weighted averaging filter:** In this, pixels are multiplied by different coefficients. Center pixel is multiplied by a higher value than average filter.
2. **Order Statistics Filter:** It is based on the ordering the pixels contained in the image area encompassed by the filter. It replaces the value of the center pixel with the value determined by the ranking result. Edges are better preserved in this filtering. Below are the types of order statistics filter:
   - **Minimum filter:** 0th percentile filter is the minimum filter. The value of the center is replaced by the smallest value in the window.
   - **Maximum filter:** 100th percentile filter is the maximum filter. The value of the center is replaced by the largest value in the window.
   - **Median filter:** Each pixel in the image is considered. First neighboring pixels are sorted and original values of the pixel is replaced by the median of the list.

5. Discuss the following with respect to image enhancement

I) High boost filtering

II) Un sharp masking

High-boost filtering

- A high-boost filtered image, f_{hb} is defined at any point (x,y) as

$$f_{hb}(x,y) = Af(x,y) - \bar{f}(x,y) \quad where\ A \geq 1$$

$$f_{hb}(x,y) = (A-1)f(x,y) + f(x,y) - \bar{f}(x,y)$$

$$f_{hb}(x,y) = (A-1)f(x,y) + f_s(x,y)$$

This equation is applicable general and does not state explicity how the sharp image is obtained

6.Expalin the method of Homomorphic filtering in image enhancement.

Homomorphic filtering is an image enhancement technique that improves contrast and normalizes brightness across an image. It works by separating the illumination and reflectance components of an image, and then amplifying the reflectance components while reducing the illumination components:

1. **Convert to log domain**: Convert the input image to the log domain.
2. **Separate components**: The multiplicative components of the image are transformed to additive components by moving to the log domain.
3. **Filter**: Use a high-pass filter in the log domain to remove the low-frequency illumination component while preserving the high-frequency reflectance component

Homomorphic filtering can be used for improving the appearance of a grayscale image by simultaneous intensity range compression (illumination) and contrast enhancement (reflection).

Where,

m = image,

i = illumination,

r = reflectance

We have to transform the equation into frequency domain in order to apply high pass filter. However, it's very difficult to do calculation after applying Fourier transformation to this equation because it's not a product equation anymore. Therefore, we use 'log' to help solve this problem.

Then, applying Fourier transformation

Or

Next, applying high-pass filter to the image. To make the illumination of an image more even, the high-frequency components are increased and low-frequency components are decrease.

Where

H = any high-pass filter

N = filtered image in frequency domain

Afterward, returning frequency domain back to the spatial domain by using inverse Fourier transform.

Finally, using the exponential function to eliminate the log we used at the beginning to get the enhanced image

7. Explain about different edge detection methods.

Edges are significant local changes of intensity in a digital image. An edge can be defined as a set of connected pixels that forms a boundary between two disjoint regions. There are three types of edges:

- Horizontal edges
- Vertical edges
- Diagonal edges

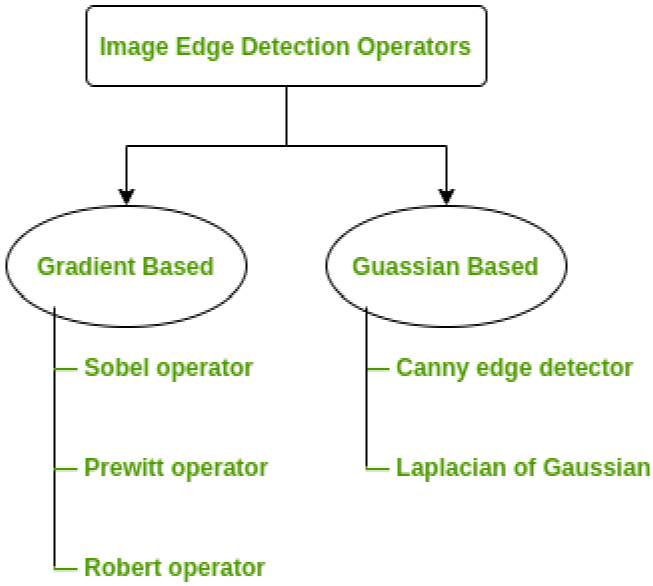
Edge Detection is a method of segmenting an image into regions of discontinuity. It is a widely used technique in digital image processing like

- pattern recognition
- image morphology
- feature extraction

Edge detection allows users to observe the features of an image for a significant change in the gray level. This texture indicating the end of one region in the image and the beginning of another. It reduces the amount of data in an image and preserves the structural properties of an image.

Edge Detection Operators are of two types:

- **Gradient –** based operator which computes first-order derivations in a digital image like, Sobel operator, Prewitt operator, Robert operator
- **Gaussian –** based operator which computes second-order derivations in a digital image like, Canny edge detector, Laplacian of Gaussian



Sobel Operator: It is a discrete differentiation operator. It computes the gradient approximation of image intensity function for image edge detection. At the pixels of an image, the Sobel operator produces either the normal to a vector or the corresponding gradient vector. It uses two 3 x 3 kernels or masks which are convolved with the input image to calculate the vertical and horizontal derivative approximations respectively –

Advantages:

1. Simple and time efficient computation
2. Very easy at searching for smooth edges

Limitations:

1. Diagonal direction points are not preserved always
2. Highly sensitive to noise
3. Not very accurate in edge detection
4. Detect with thick and rough edges does not give appropriate results

Prewitt Operator: This operator is almost similar to the sobel operator. It also detects vertical and horizontal edges of an image. It is one of the best ways to detect the orientation and magnitude of an image. It uses the kernels or masks –

Advantages:

1. Good performance on detecting vertical and horizontal edges
2. Best operator to detect the orientation of an image

Limitations:

1. The magnitude of coefficient is fixed and cannot be changed
2. Diagonal direction points are not preserved always

Robert Operator: This gradient-based operator computes the sum of squares of the differences between diagonally adjacent pixels in an image through discrete differentiation. Then the gradient approximation is made. It uses the following 2 x 2 kernels or masks –

Advantages:

1. Detection of edges and orientation are very easy
2. Diagonal direction points are preserved

Limitations:

1. Very sensitive to noise
2. Not very accurate in edge detection

Marr-Hildreth Operator or Laplacian of Gaussian (LoG): It is a gaussian-based operator which uses the Laplacian to take the second derivative of an image. This really works well when the transition of the grey level seems to be abrupt. It works on the zero-crossing method i.e when the second-order derivative crosses zero, then that particular location corresponds to a maximum level. It is called an edge location. Here the Gaussian operator reduces the noise and the Laplacian operator detects the sharp edges.
The Gaussian function is defined by the formula:

Where

is the standard deviation.
And the LoG operator is computed from

Advantages:

1. Easy to detect edges and their various orientations
2. There is fixed characteristics in all directions

Limitations:

1. Very sensitive to noise

2. The localization error may be severe at curved edges
3. It generates noisy responses that do not correspond to edges, so-called "false edges"

Canny Operator: It is a gaussian-based operator in detecting edges. This operator is not susceptible to noise. It extracts image features without affecting or altering the feature. Canny edge detector have advanced algorithm derived from the previous work of Laplacian of Gaussian operator. It is widely used an optimal edge detection technique. It detects edges based on three criteria:

1. Low error rate
2. Edge points must be accurately localized
3. There should be just one single edge response

Advantages:

1. It has good localization
2. It extract image features without altering the features
3. Less Sensitive to noise

Limitations:

1. There is false zero crossing
2. Complex computation and time consuming

Some Real-world Applications of Image Edge Detection:

- medical imaging, study of anatomical structure
- locate an object in satellite images
- automatic traffic controlling systems
- face recognition, and fingerprint recognition

8. What is LOG filter?How LOG is advantageous than laplacian filter?

A Laplacian filter is an image processing tool that detects edges in an image by calculating the second derivatives of an image's intensity:

The Laplacian filter measures how quickly the first derivatives change, which helps determine if a change in pixel values is an edge or a continuous progression.

The Laplacian filter highlights areas of rapid intensity change, producing a picture of all the edges in an image.

The Laplacian filter is often applied to an image that has first been smoothed with a Gaussian smoothing filter to reduce its sensitivity to noise.

The Laplacian filter kernel usually contains negative values in a cross pattern, centered within the array. The corners are either zero or positive values, and the center value can be either negative or positive.

Zero crossings in the Laplacian correspond to edges in the image

The Laplacian of Gaussian (LoG) filter is a useful tool for finding edges and blobs in images. It's advantageous over other filters because it can:

- **Reduce noise**: The LoG filter can reduce the impact of variations caused by noise.
- **Find edges**: The LoG filter can find edges by calculating the second spatial derivative of an image. The LoG response is positive on the darker side of an edge and negative on the lighter side.
- **Find blobs**: The LoG filter can also find blobs.

The LoG filter is defined as the Laplace operator applied to a Gaussian kernel. The Laplace operator is the sum of second order derivatives.