Image Processing and Computer Vision

Sara Mukherjee

July 8, 2024

Contents

Chapter 1

Image Alignment and Warping

1.1 Introduction

A digital image is composed of elements called pixels. Each pixel is associated with discrete and finite quantities for numeric representation of its intensity or gray level which is obtained as a function of its spatial co-ordinates (x and y co-ordinates). The intensity values typically range from 0-255 (8 bit integers).Images can be classified into varius colortypes like :

- **Gray Images :** Gray Images can be thought of black and white images. The intensity values of the pixels lie in the range 0-255, basically as 8 bit integers.
- **RGB or Truecolor Images :** In these type of images, each pixel is associated with an intensity value corresponding to Red, Blue and Green color. Each image can be associated with three-dimensional planes. The first plane in the third dimension represents the red pixel intensities, the second plane represents the green pixel intensities, and the third plane represents the blue pixel intensities.
- **Binary Images :** As the name suggests, the pixels making up the images are associated with intensity values either represented as 1 or 0. So these images have only two colors - black and white.

1.2 Image Alignment

Image alignment or image registration is the process of aligning two images. Two images are said to be aligned if corresponding co-ordinates of the two images give information about the same physical points. So basically we need to transform two images into one one co-ordinate system. Now with respect to one image, another image can some amount of rotation about any point, translation, or it may also be sheared to some extent. So we need to use some motion models to bring them to the same co-ordinate system.

1.2.1 Motion Models

- **Translation** Let $\forall(x_1, y_1)$ in Image 1, co-ordinates of Image 2 can be represented as $(x_2, y_2) = (x_1 + t_x, y_1 + t_y)$

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$$

- **Rotation** Rotation about point (x_c, y_c) anti-clockwise through angle θ.
  $x_2 = (x_1 - x_c)cos\theta - (y_1 - y_c)sin\theta + x_c$
  $y_2 = (x_1 - x_c)sin\theta + (y_1 - y_c)cos\theta + y_c$

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \begin{pmatrix} cos\theta & -sin\theta & x_c \\ sin\theta & cos\theta & y_c \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 - x_c \\ y_1 - y_c \\ 1 \end{pmatrix}$$

- **Rotation and Translation**
  $x_2 = (x_1 - x_c)cos\theta - (y_1 - y_c)sin\theta + t_x$

$y_2 = (x_1 - x_c)sin\theta + (y_1 - y_c)cos\theta + t_y$

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \begin{pmatrix} cos\theta & -sin\theta & t_x \\ sin\theta & cos\theta & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 - x_c \\ y_1 - y_c \\ 1 \end{pmatrix}$$

- **Affine Transformation** Affine transformations include translation, rotation, shearing, scaling etc. and their compositions in any order or sequence. Affine transformations preserve colinearity. It preserves lines, points, planes, but not necessarily the origin of original system.
- **Scaling About Origin**

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \begin{pmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$$

- **Shearing**
  - **Vertical Shearing**

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$$

  - **Horizontal Shearing**

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$$

Composition of multiple motions can be represented as multiplication of corresponding transformation matrices associated with those motions.

1.2.2 Control Point Registration

Let's assume we have two mis-aligned images (images with relative motion w.r.t one another). Now, one way to align them is to keep one of the images fixed, and applying various tranformation to the second one. For the sake of simplicity, let's assume only translation and rotation was there as relative motion between the two images, so we keep applying rotation+transformation matrix to the moving image in which we vary θ, t_x and t_y in a given range e.g -90° to 90° for θ, -100 to 100 for t_x and t_y. And we choose the model, in which mean squared error is minimum.

$$MSSD = \frac{1}{N}\Sigma_{x,y\in\Omega}(I_1(x,y) - I_2(x,y))^2$$

1.3 Image Warping

Literally, warping means to getting out of shape by being bent or twisted. Image Warping means distorting the shape of an image by manipulating it digitally.

1.3.1 Forward Warping

In this kind of warping, transform is applied directly to the original image.This can lead to some holes in transformed image and also multiple pixels can have the same transformed coordinates.

1.3.2 Reverse Warping

Reverse warping can be done if injective transform was applied to the original image. One problem in this case is to decide the value of original pixel if upon calculation we get non-integer co-ordinates as pixel value for the original pixel. We can use any of the methods below to solve that problem -

- **Nearest Neighbour Method** Assign it to an integer pixel value which is closest to the calculated non-integer co-ordinates.

- **Bilinear Method** Here we approaximate image intensity using a bilinear function - $f(x,y) = a_0 + a_1x + a_2y + a_3xy$, the coefficients can be found out by the following equation -

$$\begin{pmatrix} 1 & x_1 & y_1 & x_1y_1 \\ 1 & x_2 & y_2 & x_2y_2 \\ 1 & x_3 & y_3 & x_3y_3 \\ 1 & x_4 & y_4 & x_4y_4 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} f(x_1,y_1) \\ f(x_1,y_2) \\ f(x_2,y_1) \\ f(x_2,y_2) \end{pmatrix}$$

1.4 Field Of View Issues

If two images are mis-aligned, while calculating MSSD, we should consider pixels present only in the common region of overlap of the two images.

1.5 Intensity Changes in Images

- Suppose a function exists between the corresponding intensities of two images and we know that function, let -

$$I_1(x1, y1) = g(I_2(x2, y2)), \forall (x_1, y_1), (x_2, y_2) \in \Omega$$

, then-

$$transformed - MSSD = \frac{1}{N} \Sigma_{x,y \in \Omega} (g(I_1(x,y)) - I_2(x,y))^2$$

- Suppose a linear relation exists between the intensities of corresponding points of two images, but the complete equation is not known -

$$I_1(x,y) = aI_2(x,y) + b, \forall (x_1, y_1), (x_2, y_2) \in \Omega$$

, then -

$$NCC = \frac{\Sigma_{x,y \in \Omega} (I_1 - \overline{I_1})(I_2 - \overline{I_2})}{\sqrt{(\Sigma_{x,y \in \Omega} (I_1 - \overline{I_1})^2)(\Sigma_{x,y \in \Omega} (I_2 - \overline{I_2})^2)}}$$

NCC is called Normalized Cross-Correlation, or Correlation Coefficient.

1.5.1 Image Histogram

Image Histogram represents the relative frequency of occurance of verious intensity levels in an image.The intensity axis(normally the x-axis is numbered from 0 to $L-1$), whereas the y-axis represents $p(r_k) = \frac{n_k}{n}$. Basically, image histograms are used when there does exist a functional relationship between the intensities of two images, but that relationship is not known to us, we can use joint image histograms then.

1.5.2 Joint Image Histogram

Joint Image Histogram is basically constructed by plotting the intensities at pixels of image1 against the intensities at corresponding pixels of image2. Measures of dispersion of a joint histogram -

- **Entropy** $S(x) = -\Sigma_{x \in DX} p(x = X) \log_2 p(x = X)$ Entropy is calculated from the normalized distribution of X, and not from the actual values.
- **Joint Entropy** $S(x) = -\Sigma_{x \in DX} \Sigma_{y \in DY} p(X = x, y = Y) \log_2 p(x = X, y = Y)$
  Minimizing joint entropy of the joint intensity distribution of two images is the method of aligning two images with different intensity profiles i.e., maximizing the correlation-coefficient.

1.6 Steps In An Image Alignment Algorithms

1. Choosing an optimizing metric, which is either MSSD, or Joint Entropy.
2. Choosing a motion model e.g, affine, rotation+translation etc.
3. Choosing an interpolation algorithm for image warping
4. Choosing an optimization algorithm.

1.7 Applications Of Image Alignment

- **Template Matching** It is the process of looking for the presence of template, (a smaller image) within a bigger image.Here, we move over each pixel of the bigger image and calculate the MSSD of the smaller image and the area sorrounding the pixel we picked from the bigger image with dimensions equal to the smaller image(that area can also be rotated by varying angles to get more accurate results, we'll take the angle at which minimum MSSD is observed.). The point at which we get the least value of MSSD will be the point of maximum match, now the value of the least MSSD can be used to analyze whether or not that template is present in the bigger image.
- **Image Panoramas** This is the method of stitching various images taken from different view points, to get the total image.This method is known as homography.

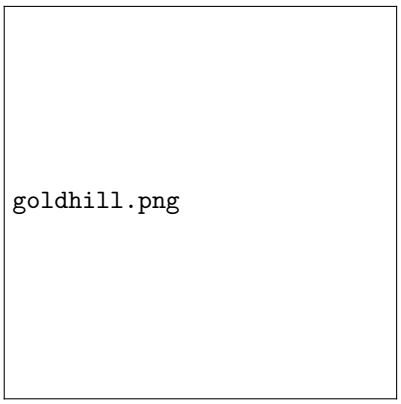
Chapter 2

Image Intensity Transformations and Image Enhancement

Image enhancement is the process of manipulating an image to make it more visually suitable for some specific purposes.Below are some important and popular image transformations for enhancement .

2.1 Image Negatives

This is just like the photographic negative, in which intensity levels get reversed.



(a) Image 2.1

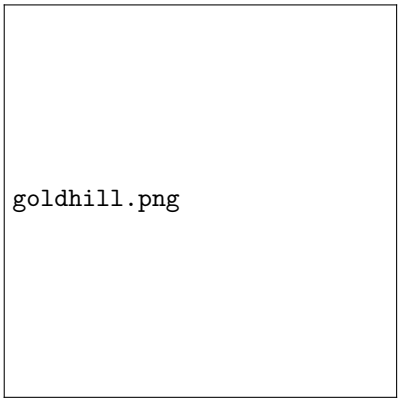
(b) Negative Image of Image 2.1

2.2 Logarithmic Transformations

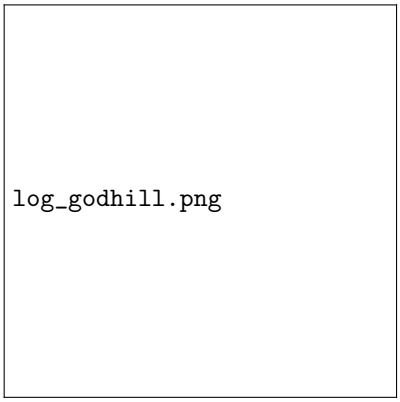
Logarithmic transformations decreases contrast in high intensity regions and tranforms a low intensity region of an image into wider range of output intensities.It allows perception of great detail.

$$s = c \log 1 + r$$

where, s = output intensity, r = input intensity, c = constant.



(a) Image 2.1

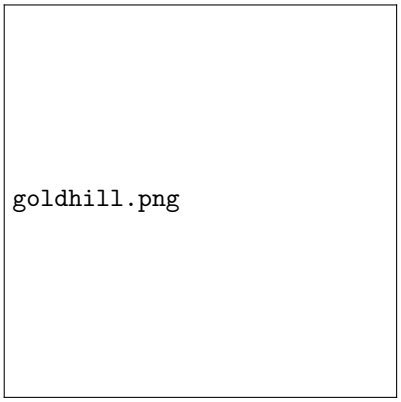


(b) Logarithmically Tansformed with c=1.

2.3 Power Law Tranformstion(Gamma Tranformation)

This leads to higher intensities being mapped to a narrower range, and lower intensities being mapped to a greater range.

$$s(r) = cr^{\gamma}$$



(a) Image 2.1

(b) Power Tansformed with c=1, $\gamma = 0.6$.

(a) Image 2.1

(b) Power Tansformed with c=1, $\gamma = 3$.

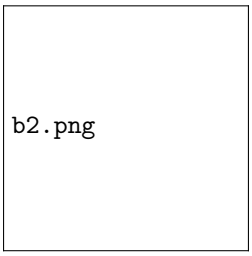
2.4 Bit-plane Slicing

Each gray-scale image has the intensities of its pixels as 8-bit integers. So each gray scale image can be thought to be made up of 8 binary images.

$$J(x, y) = \Sigma_{k=q'}^{q} 2^{k-1} I(x, y, k)$$

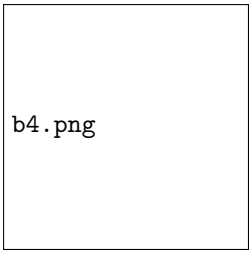
Below are the 8 binary planes extracted from Image 2.1 -

(a) Binary Plane 1



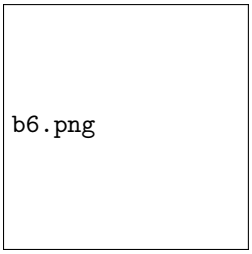
(b) Binary Plane 2

(c) Binary Plane 3

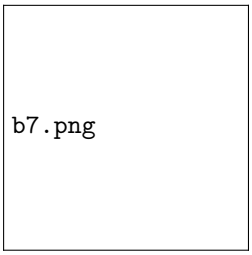


(d) Binary Plane 4

(e) Binary Plane 5



(f) Binary Plane 6



(g) Binary Plane 7

(h) Binary Plane 8

2.5 Histogram Equalization

Histogram Equalization is the process of uniformly distributing the image histogram over the total intensity range of the image. Thus it is an intensity transformation process.Let the intensity values of image range from 0 to $L-1$ and we're applying a transformation function $T(r)$ to the image. The conditions on $T(r)$ are -

- It should be strictly increasing, i.e., it must be an injective function.
- $T(r)$ ranges from 0 to L-1 , i.e., it is an surjective function.

From the above two properties, we can say that T^{-1} exists. Now let $p_r(x)$ be the probability density function (PDF) of r, and $F_r(x)$ be the Cumulative Distributive Function (CDF) of r.Therefore, the CDF of s can be expressed as -

$$F_s(x) = P(s \leq x) = P(T(r) \leq x) = P(r \leq T^{-1}(x) = F_r(T^{-1}(x)))$$

Now,

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

If we define our transformation function like -

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

Now differentiating the above equation, we get -

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

Therefore,

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

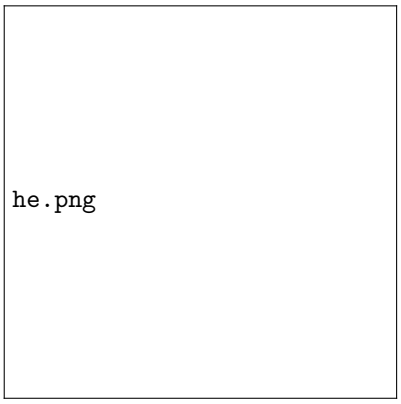
And thus, we end up getting a uniform pdf for s. Now, we're transforming integrals to summations, to extend our function for discrete values of r.

$$p_s = (L-1)\Sigma_{j=0}^{k} p_r(r_j)$$

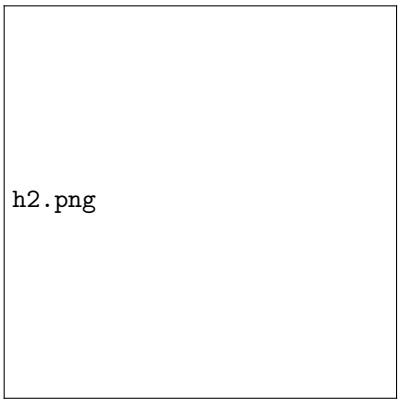
After applying Histogram Equalization to thw above image -

(a) Image 2.1

(b) Image Histogram of Image 2.1



(a) Image 2.2:Histogram Equalized form of Image 2.1



(b) Image Histogram of Image 2.2

2.6 Histogram Specification

Histogram Specification, or Histogram Matching is the process of transforming the intensity profile of one image such that its image histogram matches with a specified image histogram. Let $p_r(r)$ be the PDF of r (our original image), and $p_z(z)$ be the desired PDF after transforming it. Now the CDFs are given by -

$$F_r(r_k) = \Sigma_{j=0}^{k} p_r(r_j)$$

$$F_s(s_k) = \Sigma_{j=0}^{k} p_s(s_j)$$

Now we need to map each r value of original image to the z value that has the same probability in the desired pdf. i.e., $S(r_j) = G(z_i)$ or $z = G^{-1}(S(r))$.

2.7 Spatial Filters

2.7.1 Local Spatial Filters

Pixel value of a point is changed based on some weighted combination of pixels from a small (often rectangular) neighborhood around the pixel.

$$g(x,y) = \Sigma_{i=-a}^{a}\Sigma_{j=-a}^{a} w(i,j)f(x+i,y+j)$$

where $g(x,y)$ constitute the output pixel values, and $w(i,j)$ are the weights, also referred to as the mask of filter.

2.7.2 Correlation and Convolution

Correlation :- $(w \otimes f)(x,y) = \Sigma_{i=-a}^{a}\Sigma_{j=-a}^{a} w(i,j)f(x+i,y+j)$

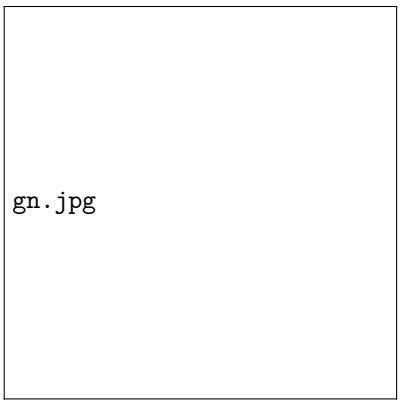
Convolution :- $(w * f)(x,y) = \Sigma_{i=-a}^{a}\Sigma_{j=-a}^{a} w(i,j)f(x-i,y-j)$ Convolution is commutative and associative, while correlation is neither commutative nor associative.

It turns out that the response of a linear time-invariant (LTI) system to any input signal x is the convolution of the signal with the impulse response of the LTI system.

2.7.3 Gaussian Noise

Gaussian noise is a kind of signal noise that has a probability density function equal to that of the normal distribution.So, basically the perturbations are random numbers from the Gaussian distribution. The formula for Gaussian distribution of x with mean μ and standard deviation σ is -

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



(a) Image 2.3

(b) Image 2.3 with gaussian noise, $\sigma = 0.25$

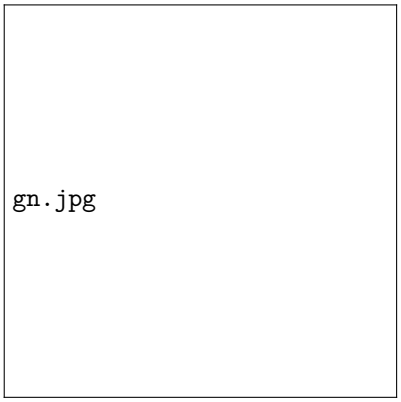
2.7.4 Impulse Noise

Random large magnitude perturbations at a few pixels. Gaussian noise affects a large number of pixels but to a smaller extent, whereas impulse affects a few pixels but to a large extent.

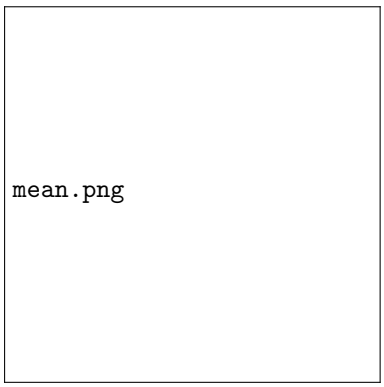
2.7.5 Mean Filter

Mean filering is based on convolution. It calculates the mean of the sorrounding pixel values and the pixel value of the point and replaces the pixel value of the point with the mean value.

$$g(x,y) = \frac{1}{(2a+1)^2}\Sigma_{i=-a}^{a}\Sigma_{j=-a}^{a} f(x+j,y+i)$$



(a) Image 2.3



(b) Image 2.3 with mean filter

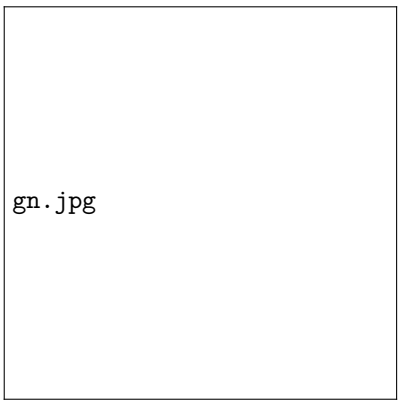
2.7.6 Weighted Mean Filter

This is a modified form of mean filter in which more weight is given to pixels nearer to a point, and less weight is given to pixels far from the point. This is also known as Gaussian Weighted Filter.

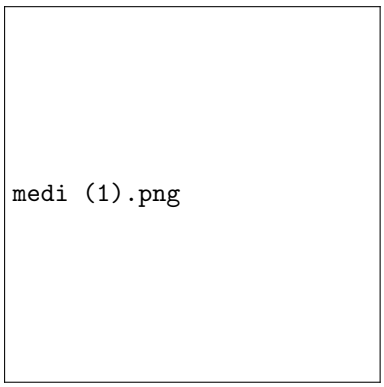
$$g(x,y) = \frac{\Sigma_{i=-a}^{a}\Sigma_{j=-a}^{a} f(x+j,y+i)e^{\frac{-(i^2+j^2)}{2\sigma^2}}}{\Sigma_{i=-a}^{a}\Sigma_{j=-a}^{a} e^{\frac{-(i^2+j^2)}{2\sigma^2}}}$$

2.7.7 Median Filter

Median Filters are generally used when there are large outliers in the image. Median filters give better preservation of features under impulse noise than mean filter. The median filter cannot be implemented using a convolution as it is nonlinear, unlike a mean filter which is linear and is implemented by convolution.



(a) Image 2.3



(b) Image 2.3 with median filter

2.7.8 Sharpening Filters

Smoothening filters generally work by calculating local average (integration), sharpening filters work by calculating local intensity derivatives, as it enhances local distinctive features. So basically, local derivative magnitudes are added back to the low-contrast image.

2.7.9 Digital Derivative Operator(For 1-D Images)

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) - 2f(x) + f(x-1)$$

Generally, second derivative is preferable for image sharpening beacuse many edges in images are ramp-like, in which case first derivative will give thick edges (undesirable), whereas second derivative gives thin edges, as it is zero along a smooth ramp.

2.7.10 Laplacian Of An Image

If an image in two-dimensional, we are interested in rotationally invariant second derivative operator. A filter is said to be rotationally invariant if rotating the image and then applying the filter to the rotated image gives the same result as applying the filter to the image and then rotating the result.

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

$$\implies \Delta^2 f = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

The two masks which are used in image sharpening are -

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} and \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

It is interesting to note that, unlike image smoothening, where weights sum up to 1, in sharpening, the weights sum up to 0.

2.7.11 Laplacian For Image Sharpening

The Laplacian operators/masks do not emphasize on regions where the intensity varies rather slowly, it emphasizes the regions where there is change in intensity, thus leading to sharpening of image features. Sharpening is performed by subtracting the Laplacian from the image -

$$g(x,y) = f(x,y) - c\Delta^2 f(x,y), c > 0$$

The Laplacian is subtracted because it leads to enhanced intensity on regions where the intensity slope was decreasing, i.e., the difference in intensity was getting less observable, similarly it decreases intensity over regions where slope was originally increasing, thus adding more sharpness to distict features of an image. **Therefore, the convolution mask for sharpening can be represented as -**

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} - c \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 0 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & -c & 0 \\ -c & 1+4c & -c \\ 0 & -c & 0 \end{pmatrix}$$