**EXPERIMENT:**

**AIM:** performing convolution on an image and masking

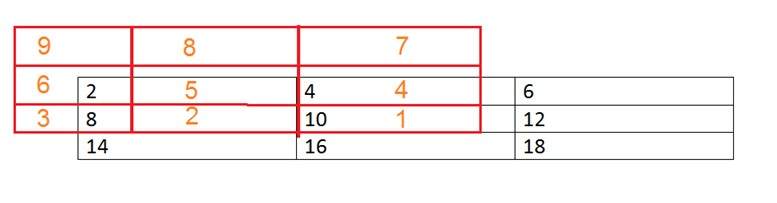
**SOFTWARE:** Python(spyder)

**THEORY:**

**Convolution**

* Convolution is a general purpose filter effect for images.
* Is a matrix applied to an image and a mathematical operation comprised of integers
* It works by determining the value of a central pixel by adding the weighted values of all its neighbors together
* The output is a new modified filtered image

Convolving mask over image. It is done in this way. Place the center of the mask at each element of an image. Multiply the corresponding elements and then add them , and paste the result onto the element of the image on which you place the center of mask.



The box in red color is the mask, and the values in the orange are the values of the mask. The black color box and values belong to the image. Now for the first pixel of the image, the value will be calculated as

First pixel = (5\*2) + (4\*4) + (2\*8) + (1\*10)

= 10 + 16 + 16 + 10

= 52

Place 52 in the original image at the first index and repeat this procedure for each pixel of the image.

**Why Convolution**

Convolution can achieve something, that the previous two methods of manipulating images can’t achieve. Those include the blurring, sharpening, edge detection, noise reduction e.t.c.

**Mask**

* Mask is also a signal. It can be represented by a two dimensional matrix. The mask is usually of the order of 1x1, 3x3, 5x5, 7x7 . A mask should always be in odd number, because other wise you cannot find the mid of the mask. Why do we need to find the mid of the mask.

A mask is a filter. Concept of masking is also known as spatial filtering. Masking is also known as filtering. In this concept we just deal with the filtering operation that is performed directly on the image.

### A sample mask has been shown below

|  |  |  |
| --- | --- | --- |
| -1 | 0 | 1 |
| -1 | 0 | 1 |
| -1 | 0 | 1 |

## What is filtering

The process of filtering is also known as convolving a mask with an image. As this process is same of convolution so filter masks are also known as convolution masks.

### **How it is done**

The general process of filtering and applying masks is consists of moving the filter mask from point to point in an image. At each point (x,y) of the original image, the response of a filter is calculated by a pre defined relationship. All the filters values are pre defined and are a standard.

### **Types of filters**

Generally there are two types of filters. One is called as linear filters or smoothing filters and others are called as frequency domain filters.

### **Why filters are used?**

Filters are applied on image for multiple purposes. The two most common uses are as following:

* Filters are used for Blurring and noise reduction
* Filters are used or edge detection and sharpness

### **Blurring and noise reduction**

Filters are most commonly used for blurring and for noise reduction. Blurring is used in pre processing steps, such as removal of small details from an image prior to large object extraction.

### **Masks for blurring**

The common masks for blurring are.

* Box filter
* Weighted average filter

In the process of blurring we reduce the edge content in an image and try to make the transitions between different pixel intensities as smooth as possible.

Noise reduction is also possible with the help of blurring.

### **Edge Detection and sharpness**

Masks or filters can also be used for edge detection in an image and to increase sharpness of an image.

## What are edges

We can also say that sudden changes of discontinuities in an image are called as edges. Significant transitions in an image are called as edges.A picture with edges is shown below.

### Cocept of Mask **Original picture Same picture with edges**

### Concept of Mask

**PROGRAM:**

import numpy as np

import cv2

from matplotlib import pyplot as plt

# Load an image

image = cv2.imread(r"C:\Users\HP\Pictures\Picture7.jpg", cv2.IMREAD\_GRAYSCALE)

# Define a convolution kernel

kernel = np.array([[1, 1, 1],

[1, -8, 1],

[1, 1, 1]])

# Perform convolution

convolved\_image = cv2.filter2D(image, -1, kernel)

# Resize the mask to match the shape of the convolved image using NumPy

mask = np.array([[0, 1, 0],

[1, 1, 1],

[0, 1, 0]])

# Resize the mask using NumPy

mask = cv2.resize(mask, (convolved\_image.shape[1], convolved\_image.shape[0]), interpolation=cv2.INTER\_NEAREST)

# Apply masking

masked\_image = convolved\_image \* mask

# Display the original, convolved, and masked images

plt.figure(figsize=(10, 5))

plt.subplot(131)

plt.imshow(image, cmap='gray')

plt.title('Original Image')

plt.subplot(132)

plt.imshow(convolved\_image, cmap='gray')

plt.title('Convolved Image')

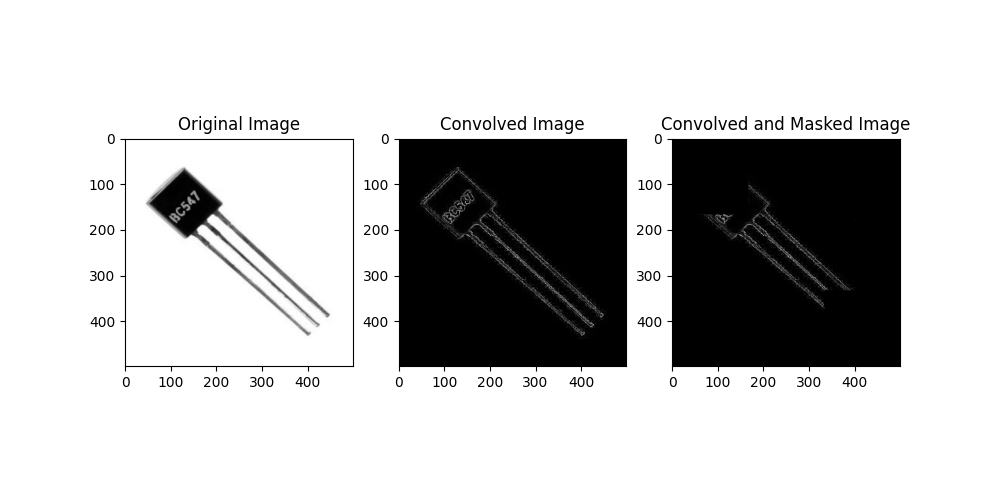
plt.subplot(133)

plt.imshow(masked\_image, cmap='gray')

plt.title('Convolved and Masked Image')

plt.show()

**output:**

****

**Conclusion:**

The convolution operation helps in extracting features from the image, and masking allows for selective modification based on a specific pattern.