Lab 8



Spatial Filtering Basics-1

**Submitted By:**

Name:Fatima Seemab

CMS ID:291310

BSCS 9B

**Task #1: Effect of averaging and the size of averaging filters**

Consider the following image. Apply averaging with a filter size of 3\*3, 5\*5, 15\*15, and 35\*35. A sample 3\*3 averaging filter is given below.

| 1 | 1 | 1 |
| --- | --- | --- |
| 1 | 1 | 1 |
| 1 | 1 | 1 |



What do you observe when increasing the size of the filter and why?

Apply different weighted averaging filters on the same image and note down the effect they have on the input image. One weighted averaging filter is given below.

| 1 | 2 | 1 |
| --- | --- | --- |
| 2 | 4 | 2 |
| 1 | 2 | 1 |



**Code:**

**Averaging:**

def smooth(img,filter):

from PIL import Image,ImageOps

import numpy

import math

import cv2

import IPython.display as display

im=cv2.imread(img)

x=im.shape[0]

y=im.shape[1]

print(x,y)

im = cv2.cvtColor(im, cv2.COLOR\_BGR2GRAY)

filter\_one\_side=math.floor(filter/2)

img\_pad = cv2.copyMakeBorder(im, filter\_one\_side, filter\_one\_side, filter\_one\_side, filter\_one\_side, cv2.BORDER\_CONSTANT, (0,0,0))

arr\_size=filter\*filter

arr=numpy.zeros(arr\_size)

for i in range(filter\_one\_side,x-filter\_one\_side):

for j in range(filter\_one\_side,x-filter\_one\_side):

f=-math.floor(filter/2)

s=-math.floor(filter/2)

m=0

for k in range (filter):

for l in range (filter):

arr[m]=img\_pad[i+f][j+s]

s=s+1

m=m+1

s=-math.floor(filter/2)

f=f+1

value=int(numpy.mean(arr))

img\_pad[i][j]=(value)

cv2.imwrite("Lab\_7.png", img\_pad)

cv2.imshow("Lab\_7.png", img\_pad)

cv2.waitKey(0)

smooth('Lab7.png',3)

**3 \* 3 Outputs:**Text

Description automatically generated with low confidence

**5 \* 5 Outputs:**

A picture containing text

Description automatically generated

**15 \* 15 Outputs:** A picture containing text

Description automatically generated

A picture containing text

Description automatically generated

**35 \* 35 Outputs:**

Graphical user interface

Description automatically generated

**Weighted Averaging:**

**def weightedavg(img,filter,mask):**

from PIL import Image,ImageOps

import numpy

import math

import cv2

import IPython.display as display

im=cv2.imread(img)

x=im.shape[0]

y=im.shape[1]

print(x,y)

im = cv2.cvtColor(im, cv2.COLOR\_BGR2GRAY)

filter\_one\_side=math.floor(filter/2)

img\_pad = cv2.copyMakeBorder(im, filter\_one\_side, filter\_one\_side, filter\_one\_side, filter\_one\_side, cv2.BORDER\_CONSTANT, (0,0,0))

img\_pad\_output=numpy.zeros((x,y))

arr\_size=filter\*filter

arr=numpy.zeros(arr\_size)

for i in range(filter\_one\_side,x-filter\_one\_side):

for j in range(filter\_one\_side,x-filter\_one\_side):

f=-math.floor(filter/2)

s=-math.floor(filter/2)

m=0

for k in range (filter):

for l in range (filter):

arr[m]=img\_pad[i+f][j+s]\*mask[m]

s=s+1

m=m+1

s=-math.floor(filter/2)

f=f+1

value=int(sum(arr)/sum(mask))

img\_pad\_output[i][j]=(value)

cv2.imwrite("Lab\_7weight.png", img\_pad\_output)

cv2.imshow("Lab\_7weight.png", img\_pad\_output)

cv2.waitKey(0)

arr=[1,2,1,2,4,2,1,2,1]

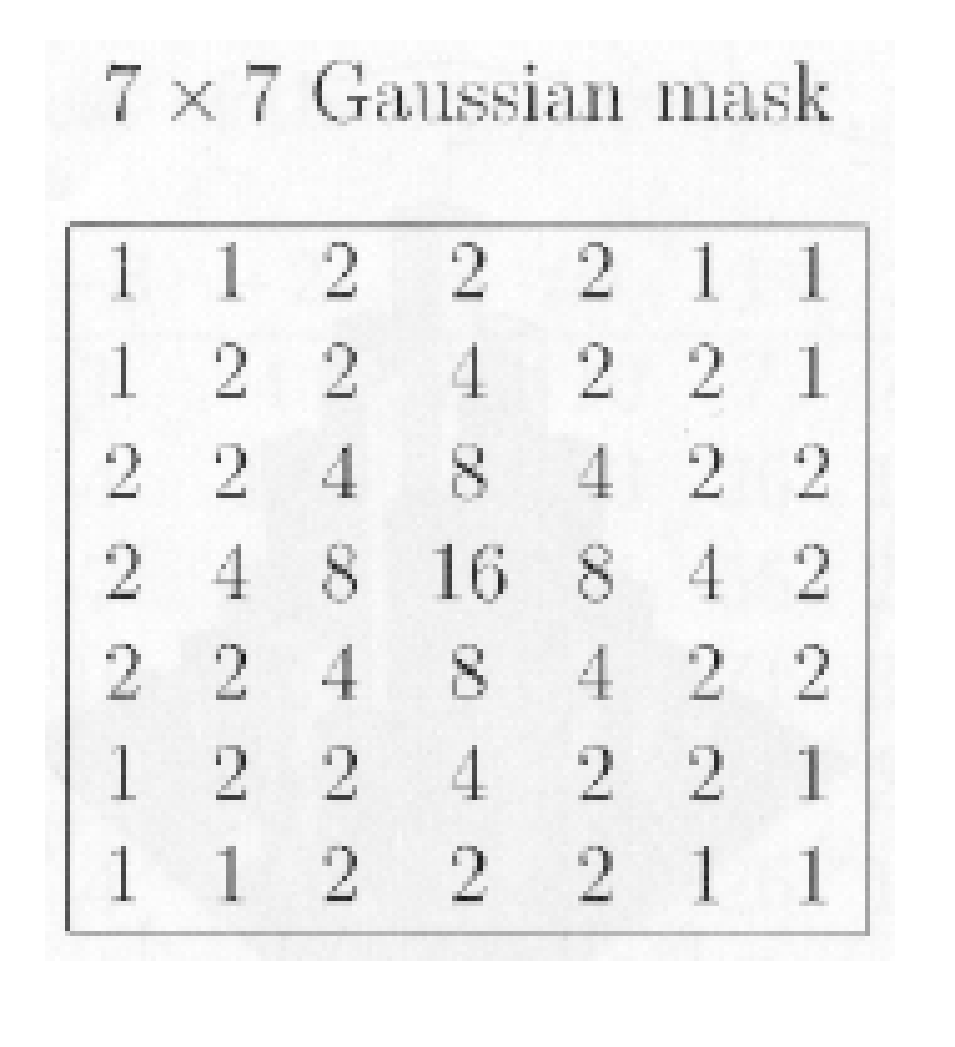
weightedavg('Lab\_7.png',3,arr)

**OUTPUT:**A picture containing text

Description automatically generated

**Task #2: Gaussian smoothing**

Apply the following Gaussian filter to the image given above. Here, the σ = 1.4. What impact do you think happen when the value of σ is increased? Don’t forget the normalizing factor while applying the given Gaussian filter.



**CODE:**

**def gaussian(img,filter,mask):**

from PIL import Image,ImageOps

import numpy

import math

import cv2

import IPython.display as display

im=cv2.imread(img)

x=im.shape[0]

y=im.shape[1]

print(x,y)

im = cv2.cvtColor(im, cv2.COLOR\_BGR2GRAY)

filter\_one\_side=math.floor(filter/2)

img\_pad = cv2.copyMakeBorder(im, filter\_one\_side, filter\_one\_side, filter\_one\_side, filter\_one\_side, cv2.BORDER\_CONSTANT, (0,0,0))

img\_pad\_output=numpy.zeros((x,y))

arr\_size=filter\*filter

arr=numpy.zeros(arr\_size)

for i in range(filter\_one\_side,x-filter\_one\_side):

for j in range(filter\_one\_side,x-filter\_one\_side):

f=-math.floor(filter/2)

s=-math.floor(filter/2)

m=0

for k in range (filter):

for l in range (filter):

arr[m]=img\_pad[i+f][j+s]\*mask[k][l]

s=s+1

m=m+1

s=-math.floor(filter/2)

f=f+1

value=int(sum(arr)/numpy.sum(mask))

img\_pad\_output[i][j]=(value)

cv2.imwrite("Lab\_7weight.png", img\_pad\_output)

cv2.imshow("Lab\_7weight.png", img\_pad\_output)

cv2.waitKey(0)

arr=[

[1,1,2,2,2,1,1],

[1,2,2,4,2,2,1],

[2,2,4,8,4,2,2],

[2,4,8,16,8,4,2],

[2,2,4,8,4,2,2],

[1,2,2,4,2,2,1],

[1,1,2,2,2,1,1]

]

gaussian('Lab7.png',7,arr)

**OUTPUT:**

A picture containing text

Description automatically generated