**SVKM’s NMIMS**

**Mukesh Patel School of Technology Management & Engineering**

**Department of Electronics and Telecommunication Engineering**

**Subject: Image and Video Processing Program: B.Tech**

**Sem: VII ACAY: 2020-21**

**EXPERIMENT NO. 2**

**Aim:**

To write a program in Python to implement spatial resolution (down-sampling and up- sampling) and observe its effect.

**Software:**  PYTHON.

**Prerequisite:**

|  |  |
| --- | --- |
| Sr. No | Concepts |
| 1. | Spatial resolution and grey level resolution |

**Outcome:**

After successful completion of this experiment students will be able to:

1. Understand the effect of varying spatial resolution

**Theory:**

**Spatial Resolution:**

The term spatial resolution corresponds to the total number of pixels in the given image. If the number of pixels is more, then the resolution of the image is more.

***Down-sampling:***

In down-sampling technique, number of pixels in the given image is reduced depending on the sampling frequency. Due to this, resolution and size of the image decreases.

***Up-sampling:***

The number of pixels in the down-sampled image can be increased by using up-sampling interpolation techniques. Up-sampling technique increases the resolution as well as the size of the image.

Some commonly used up-sampling techniques are

* Nearest neighbor interpolation
* Bilinear interpolation
* Cubic interpolation

**Algorithm:**

*Spatial Resolution: Down-sampling*

* Read the original image
* Select a down sampling rate say, *fs*
* New image will consist of pixels as follows:
  + After the pixel in the first column, select every pixel after *fs* column
  + Similarly, after the first row, select every pixel after *fs* row

*Spatial Resolution: Up-sampling*

* Read the above down-sampled image
* Select the up-sampling rate say, *fs*
* Create a new image of zeros having size of original image
* Fill this new image with pixel values of down-sampled image by skipping rows and columns at rate *fs*
* The remaining values in the new image can be filled by using neighborhood averaging technique.

|  |
| --- |
| Name of the Experiment: To implement spatial resolution variations on the given image |
| Roll No. C018 Name: Dhruvit Jain |
| Program: B.Tech ExTC Semester : VII |
| Date of Performance:17/07/2020 Date of Submission: 17/07/2020 |

**Code:**

**Down sampling:**

from skimage import io

import matplotlib.pyplot as plt

image = io.imread ("C:/Users/dhruv/Desktop/face.png")

io.imshow (image)

img\_down = image.copy ()

sh = image.shape

print (sh)

ds = 5

rowd = int(sh[0]/ds) +1

cold = int(sh[1]/ds) +1

img\_down = img\_down[0:rowd,0:cold]

sh = image.shape

print (sh)

rw=0

cl=0

for r in range (0,sh[0],ds):

cl=0

for c in range (0,sh[1],ds):

img\_down [rw][cl]= image[r][c]

cl += 1

rw += 1

sh1 = img\_down.shape

print (sh1)

plt.figure()

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

plt.subplot(1,2,1)

io.imshow (image)

plt.title ('Original Image')

plt.subplot(1,2,2)

io.imshow (img\_down)

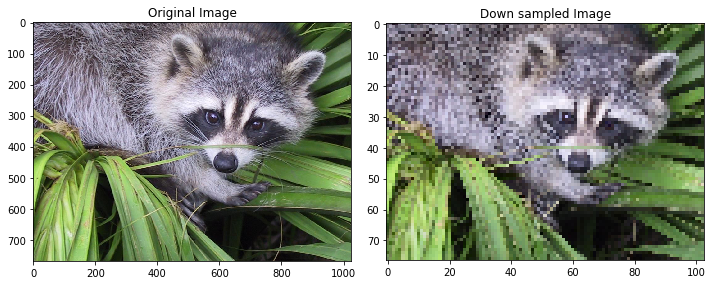
plt.title ('Down sampled Image')

**Output for down sampling:**

ds=5



ds=10



ds=25



**Up sampling:**

up = 5

img\_up = image.copy()

sh = img\_down.shape

r=0

c=0

for row in range (0,sh[0]):

c=0

for col in range (0,sh[1]):

img\_up[r][c]=img\_down[row][col]

img\_up [r,c+1:c+up] = 0

c+=up

img\_up [r+1 : r+up, :] = 0

r+=up

shu = img\_up.shape

#use average method of interpolation

#for columns

for row in range (0,shu[0],up):

for col in range (0,shu[1]-up,up):

temp1 = img\_up[row,col]/2

temp2 = img\_up[row,col+up]/2

temp = temp1 + temp2

img\_up [row,col+1 : col+up] = [int(temp[0]),int(temp[1]),int(temp[2])] #for RGB

#for rows

for row in range (0,shu[0]-up,up):

for col in range (0,shu[1]):

temp1 = img\_up[row,col]/2

temp2 = img\_up[row+up,col]/2

temp = temp1 + temp2

img\_up [row:row+up,col] = [int(temp[0]),int(temp[1]),int(temp[2])]

plt.figure()

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

plt.subplot(1,3,1)

io.imshow (image)

plt.title ('Original Image')

plt.subplot(1,3,2)

io.imshow (img\_down)

plt.title ('Down sampled Image')

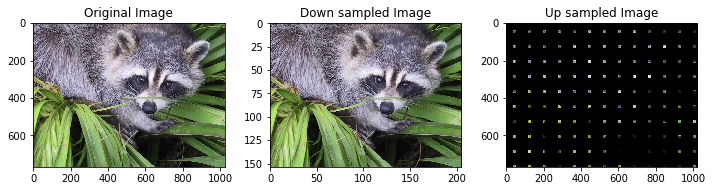
plt.subplot(1,3,3)

io.imshow (img\_up)

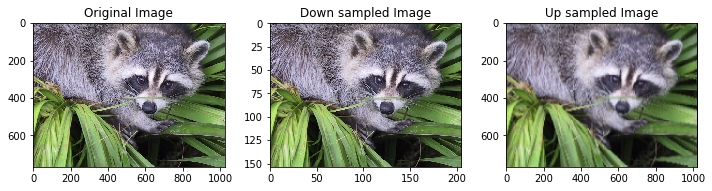
plt.title ('Up sampled Image')

**Output for up sampling:**

Before Average Interpolation:

****

After Average Interpolation:



**Conclusion**

* Down Sampling reduces the size of the image in terms of columns and rows.
* As we increase the rate of down sampling, the quality of the image deteriorates.
* For up sampling, if additional rows and columns are replaced by 0, then the quality of image worsens.
* If additional rows and columns are replaced by the average of neighboring pixels, then the quality of image improves.
* Quality of up sampled image can be further improved by using bilinear or cubic interpolation.