الجامعة الإسلامية العالمية شيتاغونغ
International Islamic University Chittagong

# LAB REPORT

| | |
|---|---|
| **DEPARTMENT** | Computer Science and Engineering |
| **COURSE-CODE** | CSE-4875 |
| **COURSE-TITLE** | Pattern Recognition and Image Processing sessional |
| **TOPIC NAME** | Implementing Some image processing problems , Increase and decrease the brightness of the image |

## SUBMITTED TO

**Mr. Mohammad Mahadi Hassan**

Associate Professor,

Dept. of CSE , IIUC.

## SUBMITTED BY

| | |
|---|---|
| **ID No.** | C173205 |
| **NAME** | Abeda Asma |
| **PROGRAM** | BSc. In CSE |
| **SEMESTER** | 8th |
| **SECTION** | 8AF |
| **SESSION** | Spring-2021 |
| **DATE OF SUBMISSION** | 09-08-2021 |

# Introduction

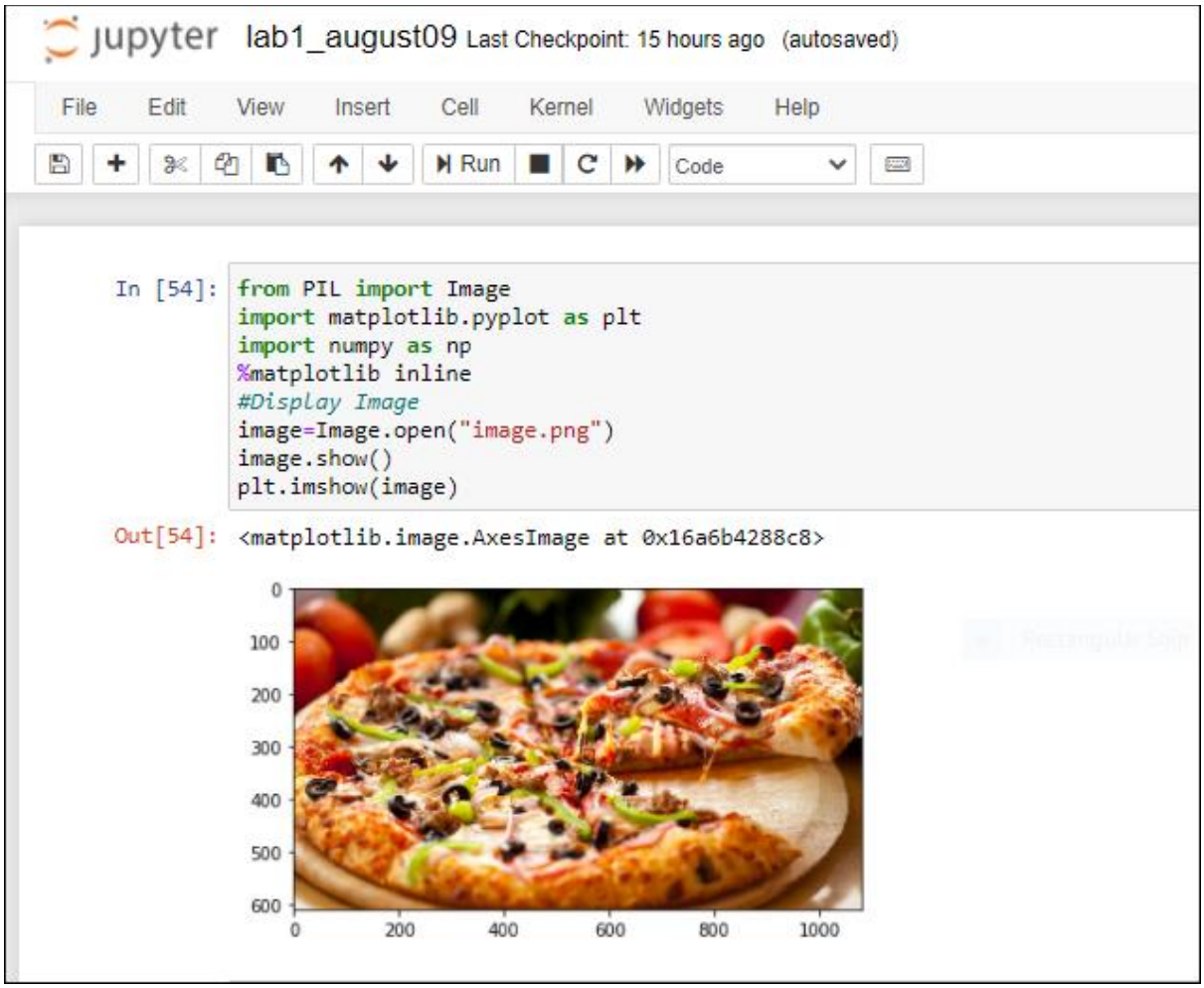
Image processing is a method that enhances image quality through the analysis and manipulation of a digitised image. It is processed with mathematical algorithms where the output of the image may either be improved or finding important data. This can include eliminating noise and artefacts, improving contrast, and the illumination of small structures.

The aim of this report is to implement some sample image processing problems and increase-decrease the brightness of the image

In order to simplify developing, debugging, and running the code, we have used the standard tools of the Python programming language together with the PIL  library. PIL (Python Imaging Library) is a free Python programming language library that adds support for opening, managing, and storing multiple image file formats. We have implemented our code using the jupyter notebook environment .

# Some sample image processing Code

- **Display image**

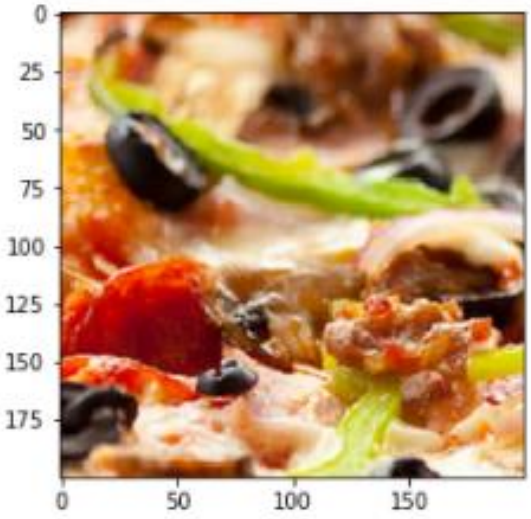


❖ **Image size, Format , mode , save and Copy**

```
In [34]: #image size ,format ,mode
         print(image.size)
         print(image.format)
         print(image.mode)

         (1080, 608)
         PNG
         RGB

In [35]: #Save
         image.save("newimage.jpeg")

In [36]: #Copy
         copied_image=image.copy()
         plt.imshow(copied_image)

Out[36]: <matplotlib.image.AxesImage at 0x16a624d1248>
```


❖ **Crop Image**

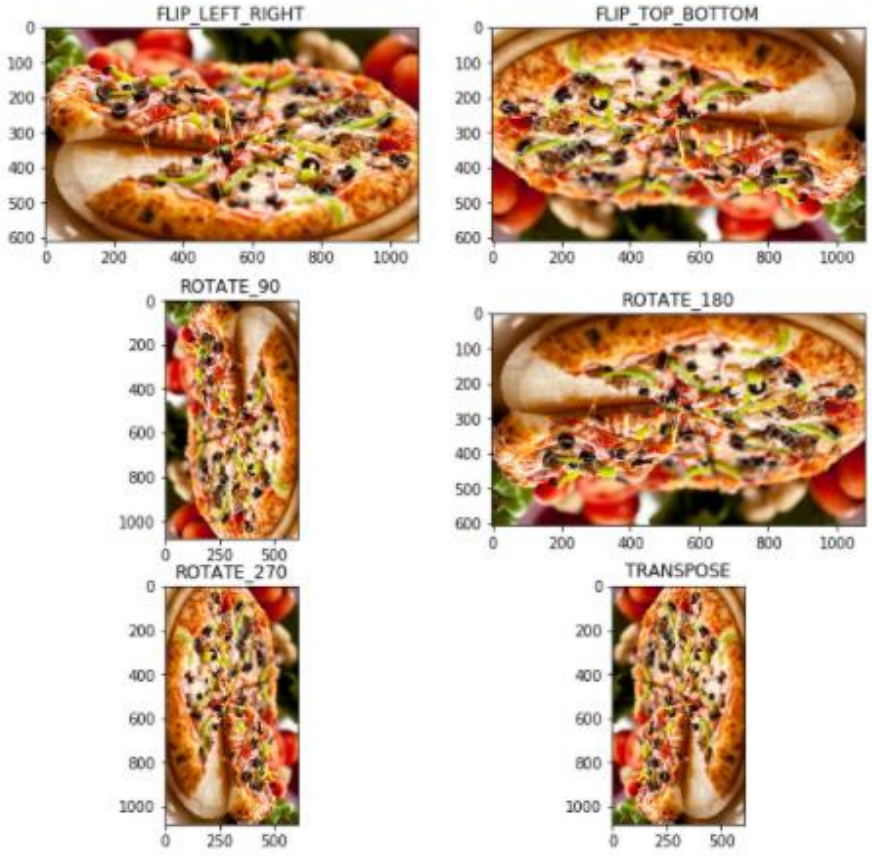
```
In [37]: #Crop
         left=50
         top=200
         right=250
         bottom=400
         crop_image=image.crop((left,top,right,bottom))
         plt.imshow(crop_image)

Out[37]: <matplotlib.image.AxesImage at 0x16a62533f48>
```



❖ **Image Transpose, Flip , Rotate**

```
In [38]: transpose_image1=image.transpose(Image.FLIP_LEFT_RIGHT)
         transpose_image2=image.transpose(Image.FLIP_TOP_BOTTOM)
         transpose_image3=image.transpose(Image.ROTATE_90)
         transpose_image4=image.transpose(Image.ROTATE_180)
         transpose_image5=image.transpose(Image.ROTATE_270)
         transpose_image6=image.transpose(Image.TRANSPOSE)
         plt.figure(figsize=(10,10))
         plt.subplot(3,2,1)
         plt.imshow(transpose_image1)
         plt.title("FLIP_LEFT_RIGHT")
         plt.subplot(3,2,2)
         plt.imshow(transpose_image2)
         plt.title("FLIP_TOP_BOTTOM")
         plt.subplot(3,2,3)
         plt.imshow(transpose_image3)
         plt.title("ROTATE_90")
         plt.subplot(3,2,4)
         plt.imshow(transpose_image4)
         plt.title("ROTATE_180")
         plt.subplot(3,2,5)
         plt.imshow(transpose_image5)
         plt.title("ROTATE_270")
         plt.subplot(3,2,6)
         plt.imshow(transpose_image6)
         plt.title("TRANSPOSE")
```

```
Out[38]: Text(0.5, 1.0, 'TRANSPOSE')
```



❖ **Resize Image**

```
In [39]: newsize=(300,300)
         plt.figure(figsize=(10,10))
         resized_image1=image.resize(newsize,Image.BILINEAR)
         resized_image2=image.resize(newsize,Image.NEAREST)
         resized_image3=image.resize(newsize,Image.BOX)
         resized_image4=image.resize(newsize,Image.HAMMING)
         resized_image5=image.resize(newsize,Image.BICUBIC)
         resized_image6=image.resize(newsize,Image.LANCZOS)
         plt.subplot(3,2,1)
         plt.imshow(resized_image1)
         plt.title("BILINEAR")
         plt.subplot(3,2,2)
         plt.imshow(resized_image2)
         plt.title("NEAREST")
         plt.subplot(3,2,3)
         plt.imshow(resized_image3)
         plt.title("BOX")
         plt.subplot(3,2,4)
         plt.imshow(resized_image4)
         plt.title("HAMMING")
         plt.subplot(3,2,5)
         plt.imshow(resized_image5)
         plt.title("BICUBIC")
         plt.subplot(3,2,6)
         plt.imshow(resized_image6)
         plt.title("LANCZOS")
```

```
Out[39]: Text(0.5, 1.0, 'LANCZOS')
```

## Brightness and Contrast Adjustment

An image must have the proper brightness and contrast for easy viewing. Brightness refers to the overall lightness or darkness of the image. Contrast is the difference in brightness between objects or regions.

Figure a  shows the effect of increasing the contrast. The details are now very visible; however, almost everything else in the image is saturated. For example ,if we  look at the noise around the border of the image.

Figures b shows the effect of changing the brightness. Increasing the brightness makes every pixel in the image appear lighter. Conversely, decreasing the brightness makes every pixel in the image appear darker. These changes can improve the viewability of excessively dark or light areas in the image, but will saturate the image if taken too far

The brightness control centers the zoomed section on the pixel values of interest. The contrast adjustment is a way of zooming in on a smaller range of pixel values.  Most digital imaging systems allow the brightness and contrast to be adjusted in just this manner, and often provide a graphical display of the output transform .

- **Image Filter Contrast Enhance**

```
In [1]: from PIL import Image, ImageFilter
        img = Image.open('image.png')
        enc_img = img.filter(ImageFilter.DETAIL)
        enc_img.show()

In [2]: from PIL import Image, ImageEnhance
        img = Image.open('image.png')
        img_con = ImageEnhance.Contrast(img)
        img_con.enhance(1.3).show("90% more contrast")
```

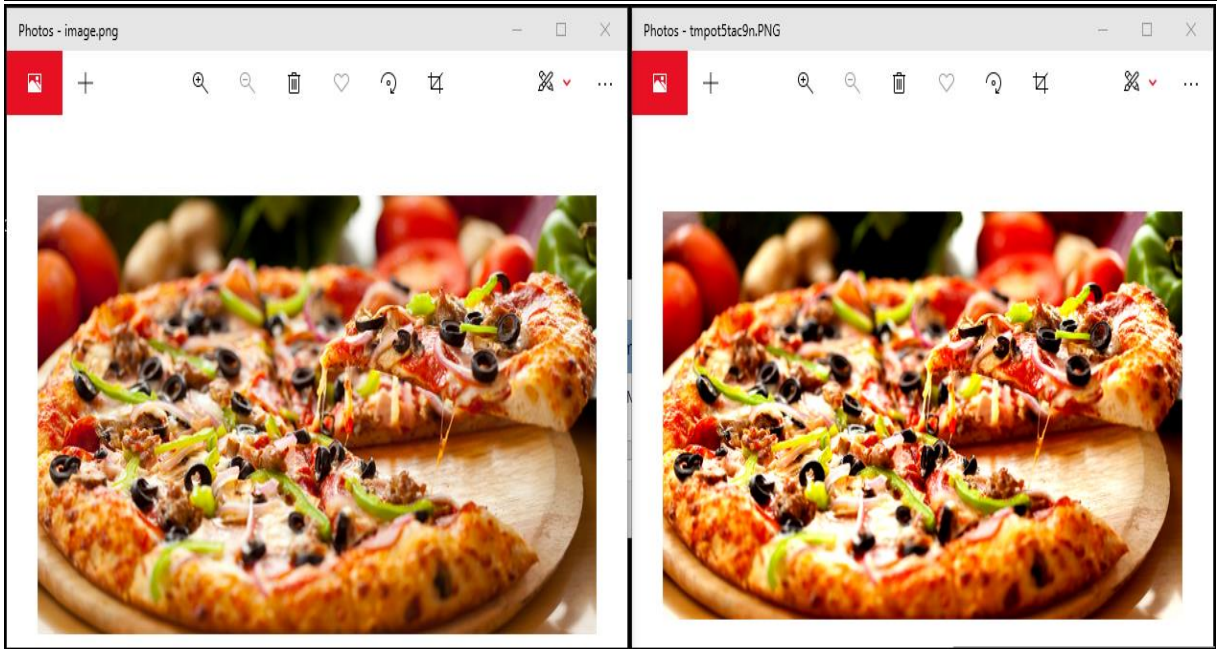
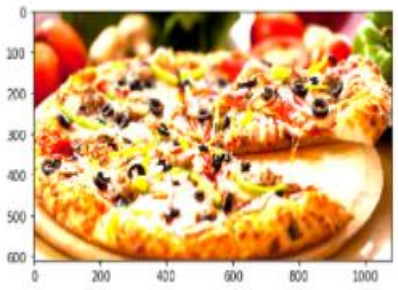



Figure a : Before and after enhancing contrast

❖ **Increase-decrease the brightness of the image**

```
In [52]: #image brightness enhancer
         from PIL import Image, ImageEnhance
         img = Image.open("image.png")
         enhancer = ImageEnhance.Brightness(img)
         enhanced_img=enhancer.enhance(1.5)
         plt.imshow(enhanced_img)
Out[52]: <matplotlib.image.AxesImage at 0x16a6b359e48>
```




```
In [53]: #image brightness decrease
         from PIL import Image, ImageEnhance
         img = Image.open("image.png")
         factor=0.5 #Decrease brightness
         dark_img=enhancer.enhance(factor)
         dark_img.save('darkened.png')
         plt.imshow(dark_img)
Out[53]: <matplotlib.image.AxesImage at 0x16a6b3c3288>
```


Figure b : Image Brightness increase and decrease

## Discussion:

In this Experiment, implemented some simple image processing problem such as – display an image, format ,copy ,crop ,flip-rotate-transpose, resize and Changing the Brightness and Contrast level of an image.These are the most basic thing with an image. When the brightness is adjusted, the entire range of tones within the image is raised or lowered accordingly. When the contrast adjustment is raised, the middle tones are eliminated. The image will have a higher percentage of darks or blacks and whites or highlights with minimal mid-tone.