Experiment 1 : Point Processing Operations

Aim : To implement point processing operations.

Code :

```
from PIL import Image
class imgProcess:
    def __init__(self,im,wid,height,c):
        self.img = im
        self.w = wid
        self.h = height
        self.ch = c
        self.new_img = Image.new("RGB",(wid,height),"white")
        self.pixels = self.new_img.load()
    def digital_negative(self):
        for wd in range(self.w):
            for ht in range(self.h):
                pix = self.img.getpixel((wd,ht))
                r = 255-pix[0]
                g = 255-pix[1]
                b = 255-pix[2]
                self.pixels[wd,ht] = (r,g,b)
        print(self.new_img)
        self.new_img.show()

    def thresholding(self):
        thr = int(input("Threshold : "))
        for wd in range(self.w):
            for ht in range(self.h):
                pix = self.img.getpixel((wd,ht))
                if pix[0]>thr:
                    r = 255
                else:
                    r = 0
                if pix[1]>thr:
                    g = 255
```

```
            else:
                g = 0
            if pix[2]>thr:
                b = 255
            else:
                b = 0
            self.pixels[wd,ht] = (r,g,b)
    print(self.new_img)
    self.new_img.show()
def grey_slice_without(self):
    nthr = int(input("Minimum Threshold : "))
    xthr = int(input("Maximum Threshold : "))
    for wd in range(self.w):
        for ht in range(self.h):
            pix = self.img.getpixel((wd,ht))
            if pix[0]<xthr and pix[0]>nthr:
                r = 255
            else:
                r = 0
            if pix[1]<xthr and pix[1]>nthr:
                g = 255
            else:
                g = 0
            if pix[2]<xthr and pix[2]>nthr:
                b = 255
            else:
                b = 0
            self.pixels[wd,ht] = (r,g,b)
    print(self.new_img)
    self.new_img.show()

def grey_slice_with(self):
    nthr = int(input("Minimum Threshold : "))
    xthr = int(input("Maximum Threshold : "))
    for wd in range(self.w):
        for ht in range(self.h):
            pix = self.img.getpixel((wd,ht))
```

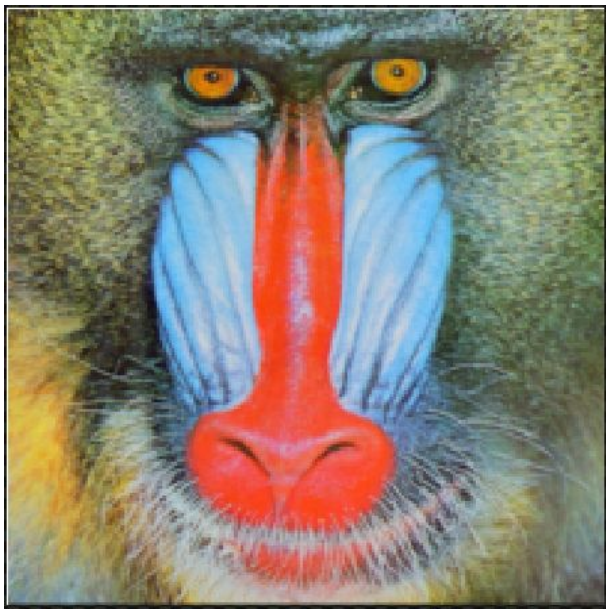
```
                if pix[0]>xthr and pix[0]<nthr:
                    r = 255
                else:
                    r = pix[0]
                if pix[1]>xthr and pix[1]<nthr:
                    g = 255
                else:
                    g = pix[1]
                if pix[2]>xthr and pix[2]<nthr:
                    b = 255
                else:
                    b = pix[2]
                self.pixels[wd,ht] = (r,g,b)
        print(self.new_img)
        self.new_img.show()

def main():
    img = Image.open("img.jpg")
    pixel = img.getpixel((50,50))
    print(pixel)
    # ------------------------------------------------------------
    w,h = img.size
    ch = int(input("1. Digital Negative\n2. Thresholding\n3. Grey level
slicing without background\n4. Grey level slicing with background\nEnter :
"))
    ip = imgProcess(img,w,h,ch)
    if ch == 1:
        ip.digital_negative()
    elif ch == 2:
        ip.thresholding()
    elif ch == 3:
        ip.grey_slice_without()
    else:
        ip.grey_slice_with()
```

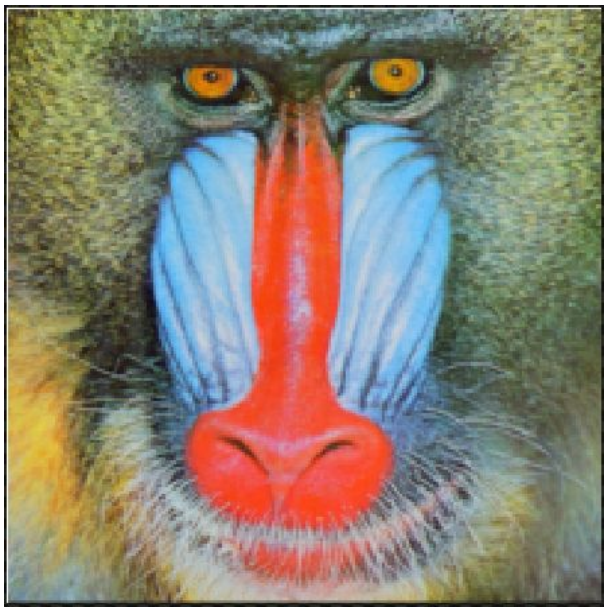
```
if __name__ == '__main__':
    main()
```

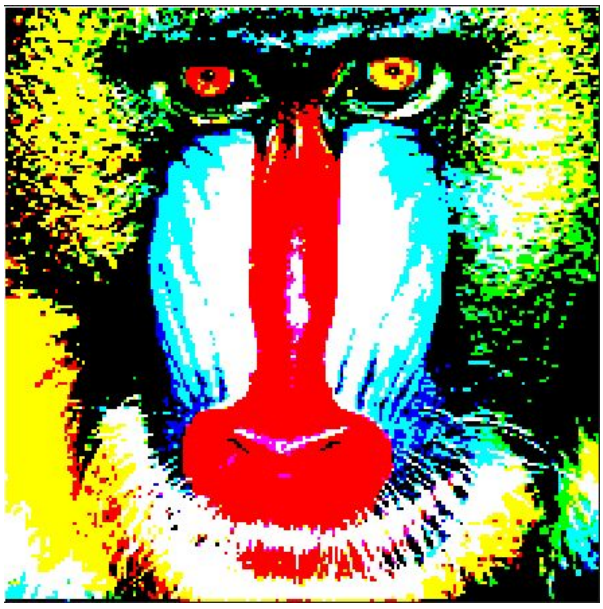
Output :

Digital Negative

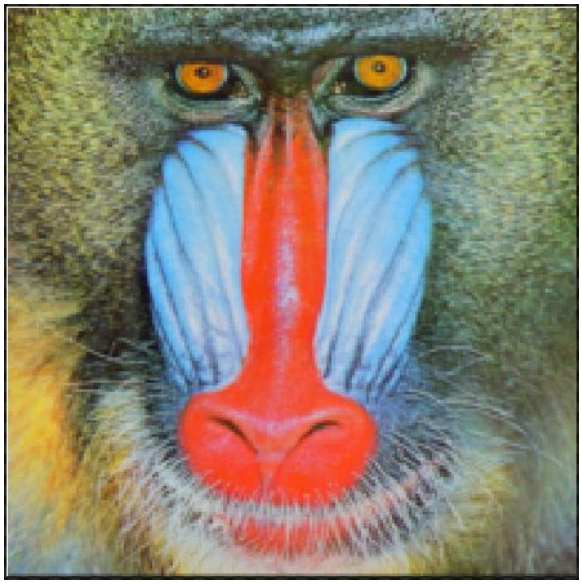


Thresholding





Grey Slice without background



Grey Slice with background