

Experiment 3 : Smoothing and Sharpening

Code :

```
from PIL import Image
import numpy
import cv2

class smooth():
    def __init__(self,wd,ht,im ):
        self.w = wd
        self.h = ht
        self.img = im
        self.new_img = Image.new("RGB",(wd,ht),"white")
        self.pixels = self.new_img.load()

    def createImg(self):
        k,l = 1,1
        mat = numpy.ones((self.h+2,self.w+2))
        print(mat.shape)
        for i in range(self.h):
            for j in range(self.w):
                pix = self.img.getpixel((j,i))
                if pix>255:
                    print(pix)
                # mat[k+i][l+j] = pix[0]
                mat[k+i][l+j] = pix
        for l1 in range(self.w):
            if l1==0:
                mat[0][0] = mat[1][1]
                mat[0][self.w+1] = mat[1][self.w]
                mat[self.h+1][self.w+1] = mat[self.h][self.w]
                mat[self.h+1][0] = mat[self.h][1]
            mat[0][l1+1] = mat[1][l1+1]
            mat[self.h+1][l1+1] = mat[self.h][l1+1]
        for l2 in range(self.h+2):
            if l2>0 and l2<(self.w+1):
                mat[l2][0] = mat[l2][1]
                mat[l2][self.w+1] = mat[l2][self.w]
        return mat
```

```

def smoothen(self):
    sum = 0
    im = self.createImg()
    for p in range(self.h-1):
        for m in range(self.w-1):
            for p1 in range(p,p+3):
                for p2 in range(m,m+3):
                    sum = sum+im[p1][p2]
            act = sum/9
            sum = 0
            self.pixels[m,p] = (int(round(act)),int(round(act)),int(round(act)))
    self.new_img.show()

```

```

class sharp:
    def sharpen(self):
        image = cv2.imread('img.jpg',0)
        # cv2.startWindowThread()
        # cv2.namedWindow("Original")
        cv2.imshow('Original',image)
        # cv2.waitKey(0)
        # cv2.destroyAllWindows()
        kernel_sharpening = numpy.array([[[-1,-1,-1],
                                           [-1,9,-1],
                                           [-1,-1,-1]]])
        sharpened = cv2.filter2D(image,-1,kernel_sharpening)
        # cv2.startWindowThread()
        # cv2.namedWindow("Sharpened")
        cv2.imshow("Sharpened",sharpened)
        cv2.waitKey(0)
        cv2.destroyAllWindows()

```

```

def main():
    img = Image.open("img3.jpg")
    pixel = img.getpixel((50,50))
    print(pixel)
    w,h = img.size
    print("W-> ",w)
    print("H -> ",h)
    smth = smooth(w,h,img)
    smth.smoothen()
    shrp = sharp()

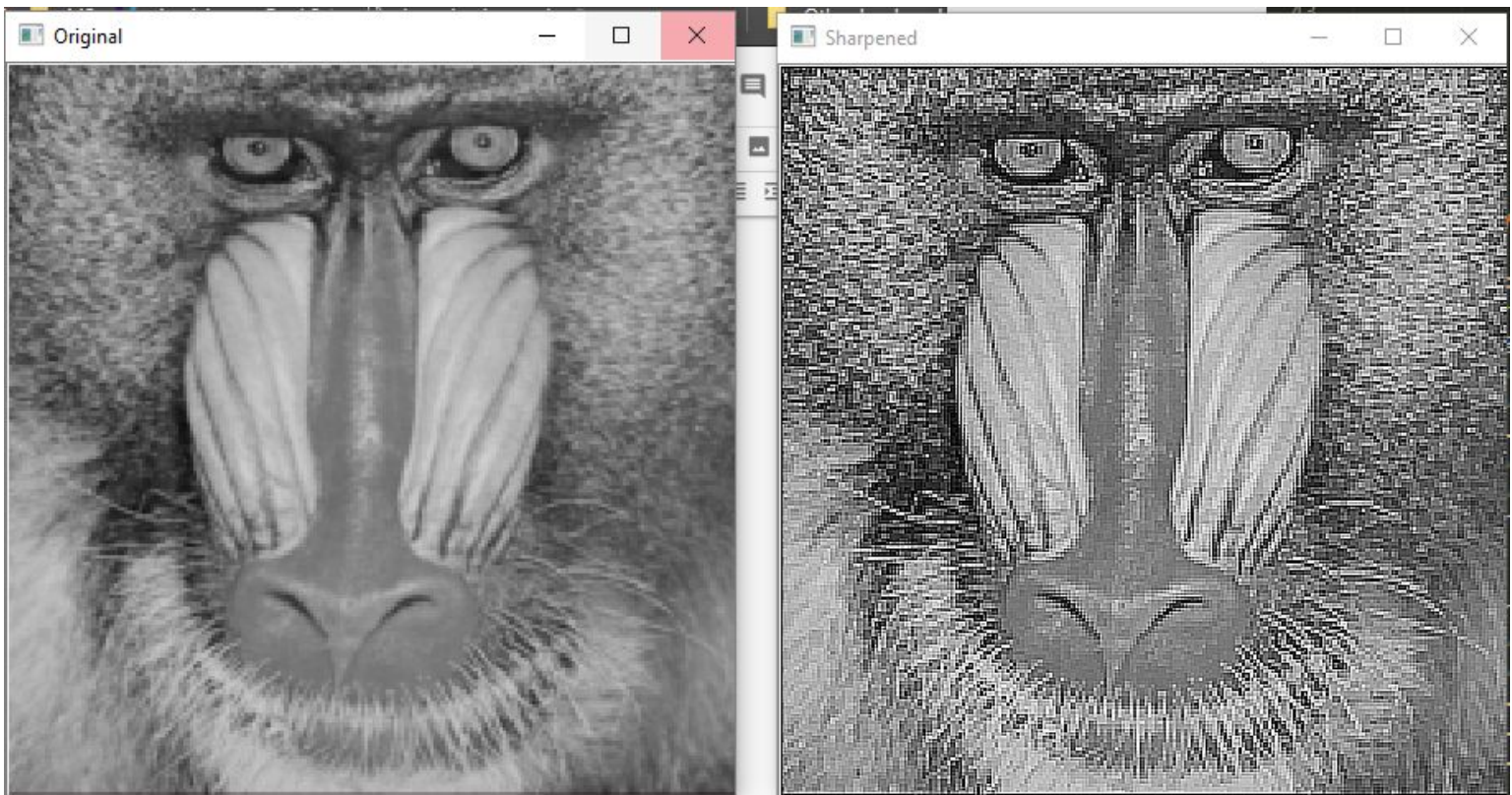
```

```
shrp.sharpen()
```

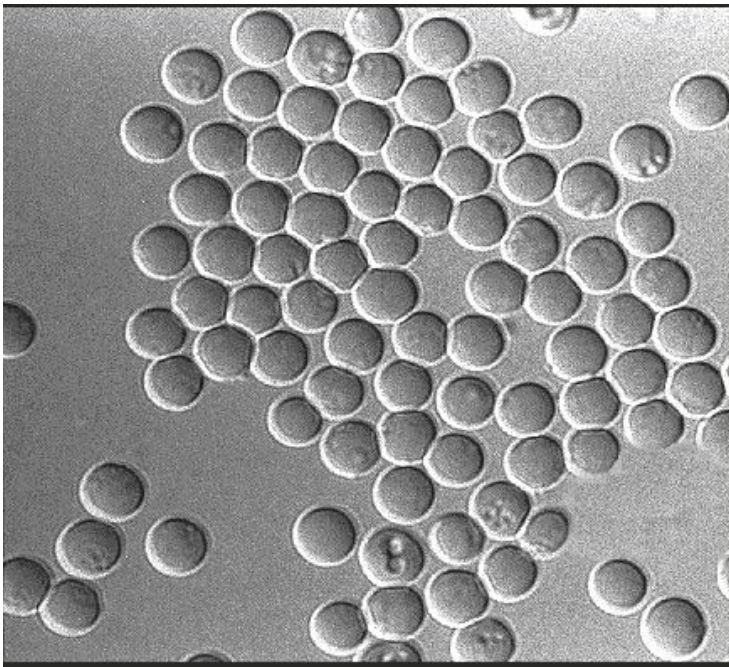
```
if __name__ == '__main__':  
    main()
```

Output :

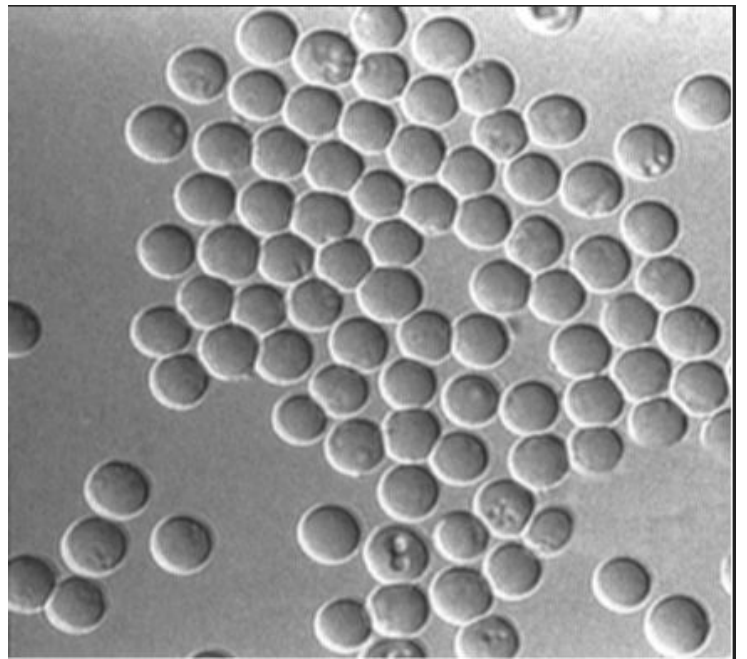
Sharpening :



Smoothing :



Original



Smoothened