

## Aufgabe 3 - Räumliche Operationen

Punktoperation, eine Nachbarschaftsoperation, eine geometrische Transformation oder eine globale Operation?

```
a) def a(img):  
    return img - np.median(img)
```

Diese Funktion zieht von jedem Pixel den Median des Arrays ab und wird somit einzeln auf jeden Pixel angewendet. Das Bild wird abgedunkelt. Es handelt sich um eine Punktoperation.

```
b) def b(img):  
    return (np.abs(img) - img) / 2
```

Auch diese Funktion wird auf jeden Pixel einzeln angewendet und ist eine Punktoperation.

```
c) def c(img):  
    return np.minimum(img, 255 - img)
```

Diese Funktion erzeugt ein Bild, welches nur Werte von 0 bis 128 enthält. Alle helleren Flächen werden durch ihre Negative ersetzt. Es handelt sich um eine Punktoperation.

```
d) def d(img):  
    return img - img @ img
```

Das @ steht hier für die Matrixmultiplikation und für diese werden komplette Spalten und Zeilen betrachtet, weswegen es sich um eine Nachbarschaftsoperation handelt.

```
e) def e(img):  
    return np.transpose(img)
```

Die Funktion gibt die transponierte Matrix zurück. Das Bild wird um 90 Grad gegen den Uhrzeigersinn gedreht. Die Transponierte ist die Matrix gespiegelt an ihrer Diagonalen. Somit wird die Position der Pixel berücksichtigt und es handelt sich um eine geometrische Operation.

```
f) def f(img):  
    return img[1:-1:,1:-1]+ img[:-2,:-2] + img[2:,2:]
```

Globale Operation?

```
g) def g(img):  
    result = np.zeros_like(img, dtype=np.int)  
    for x in range(img.shape[0]):  
        for y in range(img.shape[1]):  
            result[x,y] = np.sum(img[:,x,:y])
```

```

result[:, :] = result[0,0]
return result

```

Diese Funktion überschreibt doch in der zweitletzten Zeile alle zuvor festgelegten Werte mit dem Wert des oberen linken Pixels? Also ist die Rechnung davor unnötig? Ich würde sagen, dass es sich um eine Nachbarschaftsoperation handelt, da sich immer die bestimmten Zeilen und Spalten angeguckt werden, aber nicht das komplette Bild mit einbezogen wird.

```

h) def h(img):
    result = np.copy(img)
    for x in range(img.shape[0]-1):
        for y in range(img.shape[1]-1):
            a = img[x,y]
            b = img[x,y+1]+img[x+1,y+1]+img[x+1,y]
            if a>b and False:
                result[x,y]=a
            else:
                result[x,y]=b
    return result

```

Wenn der Pixel an der betrachteten Stelle x,y größer ist als die Summe seiner 3 benachbarten Pixel, dann wird der Wert an x,y beibehalten. Somit wird nur die nahegelegenen Pixel betrachtet und es handelt sich um eine Nachbarschaftsoperation.