original

noise

img with noise

img after gauss blur

Lab7-8

```python
import cv2
import matplotlib.pyplot as plt
import numpy as np
import math


img = cv2.imread('canny.png', 0)
blured_img = cv2.GaussianBlur(img, (5,5), 0)[0]
min = 80
max = 180
canny_img = cv2.Canny(blured_img, min, max)

plt.subplot(121)
plt.imshow(img)
plt.title('original')
plt.axis('off')
plt.subplot(122)
```

```
plt.imshow(canny_img)
plt.title('edge detection')
plt.axis('off')
plt.show()
```



```
src = cv2.imread('sudoku.png', cv2.IMREAD_GRAYSCALE)

dst = cv2.Canny(src, 50, 200, None, 3)
cdst = np.copy(cv2.cvtColor(dst, cv2.COLOR_GRAY2BGR))
lines = cv2.HoughLines(dst, 1, np.pi / 180, 150, None, 0, 0)

if lines is not None:
    for i in range(0, len(lines)):
        rho = lines[i][0][0]
        theta = lines[i][0][1]
        a = math.cos(theta)
        b = math.sin(theta)
        x0 = a * rho
        y0 = b * rho
        pt1 = (int(x0 + 1000*(-b)), int(y0 + 1000*(a)))
        pt2 = (int(x0 - 1000*(-b)), int(y0 - 1000*(a)))
        cv2.line(cdst, pt1, pt2, (255,0,255), 1, cv2.LINE_AA)
src = cv2.cvtColor(src, cv2.COLOR_GRAY2BGR)

plt.subplot(121)
plt.imshow(src)
plt.axis('off')
plt.title('original')
plt.subplot(122)
plt.imshow(cdst)
```

```
plt.axis('off')
plt.title('Detected lines by Houghlines Transformer')
plt.show()
```
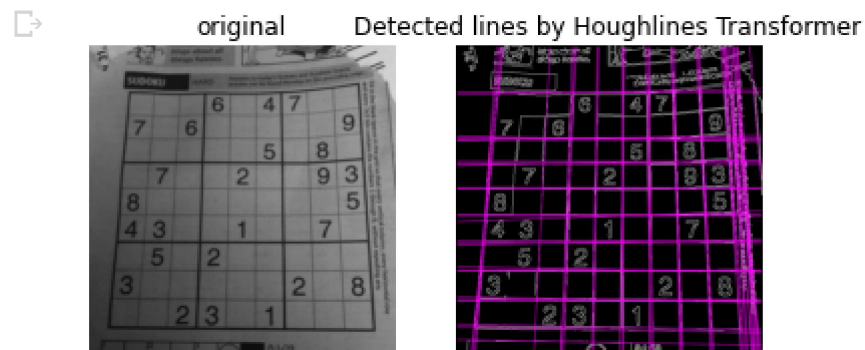

Lab11-12

```
import sys
import numpy as np
from imageio import imread, imwrite
from scipy.ndimage.filters import convolve


from tqdm import trange


!pip install seam-carving
```

```
    Collecting seam-carving
      Downloading seam_carving-1.0.0-py3-none-any.whl (10 kB)
    Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from seam-carving) (1.19.5)
    Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (from seam-carving) (1.4.1)
    Installing collected packages: seam-carving
    Successfully installed seam-carving-1.0.0
```