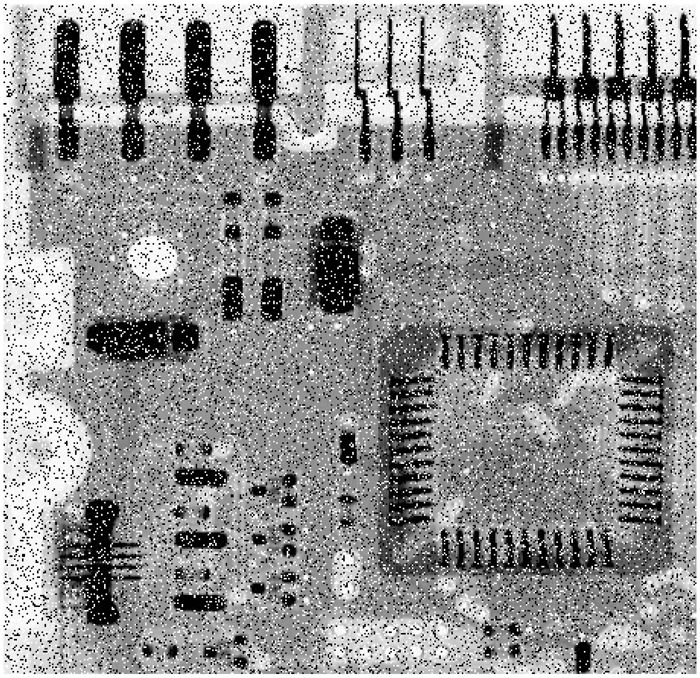
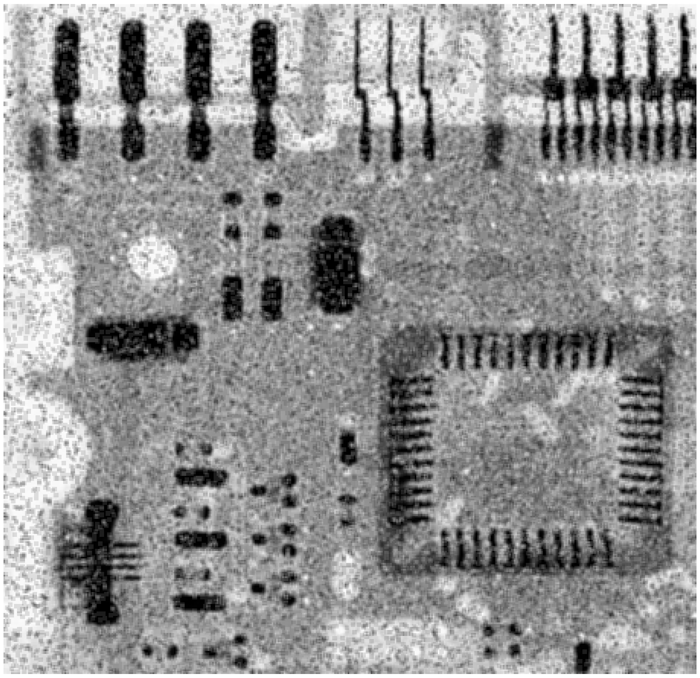
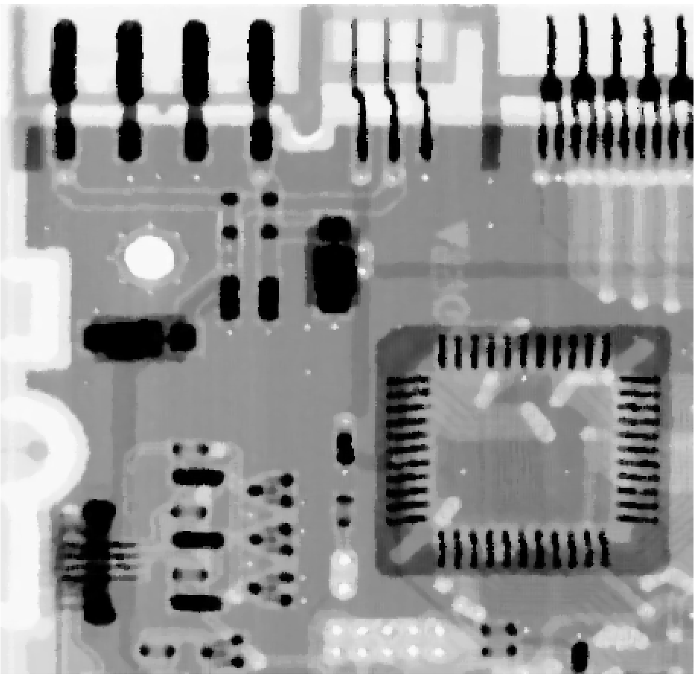
```
import cv2
from skimage import io, img_as_float
from skimage.filters import gaussian
from google.colab.patches import cv2_imshow # for image display
img = cv2.imread("/content/drive/My Drive/colab/noisysalterpepper.png", 0)
cv2_imshow(img)
```



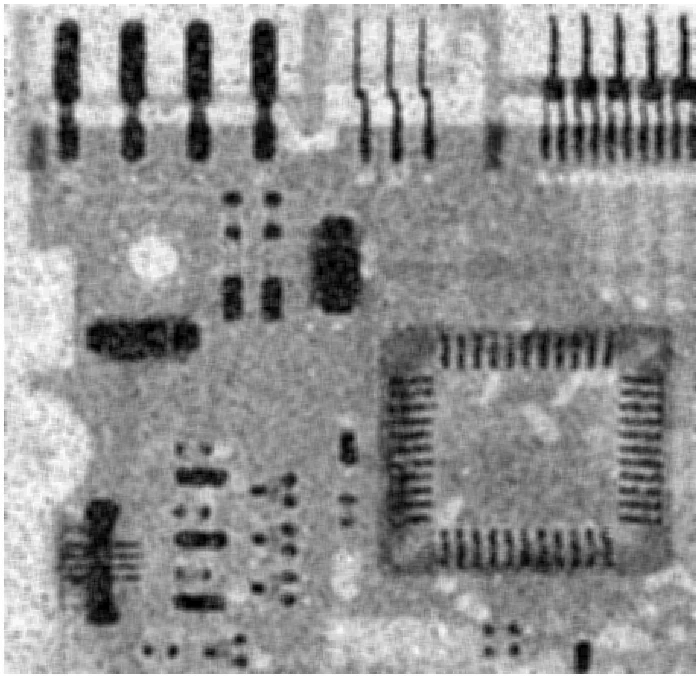
```
ksize = (3, 3)

# Using cv2.blur() method
mean = cv2.blur(img, ksize)
cv2_imshow(mean)
```



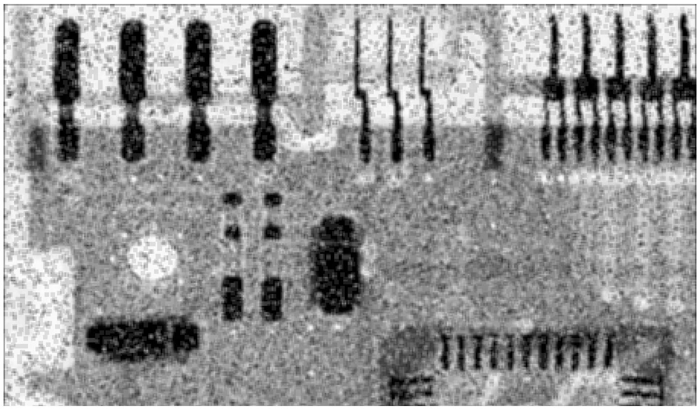
```
median = cv2.medianBlur(img,5)
cv2_imshow(median)
```



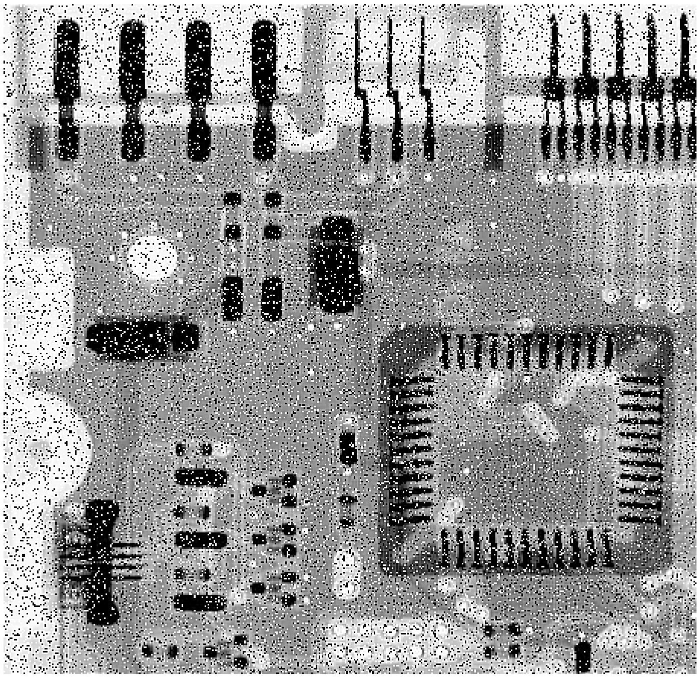
```
gaussian_using_cv2 = cv2.GaussianBlur(img, (5,5), 10,
                                      borderType=cv2.BORDER_REPLICATE)
cv2_imshow(gaussian_using_cv2)
```



```
gaussian_using_cv2 = cv2.GaussianBlur(img, (3,3), 10, borderType=cv2.BORDER_CONSTANT)
cv2_imshow(gaussian_using_cv2)
```



```
gauss = cv2.GaussianBlur(img, (3,3), 10)
# Apply Unsharp masking
unsharp_image = cv2.addWeighted(img, 2, gauss, -1, 0)
cv2_imshow(unsharp_image)
```



```
image = cv2.imread("/content/drive/My Drive/colab/cat.jpeg", 0)
sobelx = cv2.Sobel(src=image, ddepth=cv2.CV_64F, dx=1, dy=0, ksize=5) # Sobel Edge Detection on the X axis
sobely = cv2.Sobel(src=image, ddepth=cv2.CV_64F, dx=0, dy=1, ksize=5) # Sobel Edge Detection on the Y axis
sobelxy = cv2.Sobel(src=image, ddepth=cv2.CV_64F, dx=1, dy=1, ksize=5) # Combined X and Y Sobel Edge Detection
# Display Sobel Edge Detection Images
cv2_imshow(sobelx)
cv2_imshow(sobely)
cv2_imshow(sobelxy)
```