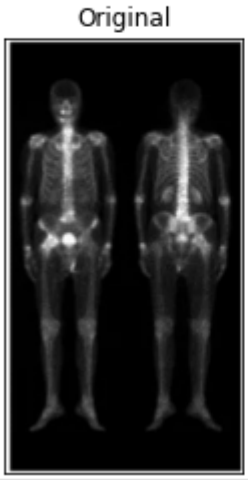
Combining Spatial Enhancement

```
import numpy as np
from google.colab.patches import cv2_imshow
import cv2
import matplotlib.pyplot as plt

%matplotlib inline
img = cv2.imread('bonescan.jpg', 0)

laplacian = cv2.Laplacian(img, cv2.CV_64F)

plt.subplot, plt.imshow(img, cmap = 'gray')
plt.title('Original'), plt.xticks([]), plt.yticks([])
#A
```


```
import numpy as np
from google.colab.patches import cv2_imshow
import cv2
import matplotlib.pyplot as plt

%matplotlib inline
img = cv2.imread('bonescan.jpg', 0)

laplacian = cv2.Laplacian(img, cv2.CV_64F)

plt.subplot, plt.imshow(img, cmap = 'gray')
plt.title('Original'), plt.xticks([]), plt.yticks([])
#A
```

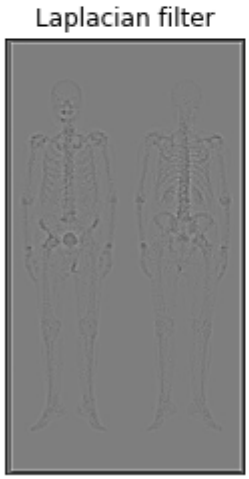

```
(Text(0.5, 1.0, 'Original'),
 ([], <a list of 0 Text major ticklabel objects>),
 ([], <a list of 0 Text major ticklabel objects>))
```



```
%matplotlib inline
img = cv2.imread('bonescan.jpg',0)

laplacian = cv2.Laplacian(img, cv2.CV_64F)

plt.subplot, plt.imshow(laplacian, cmap = 'gray')
plt.title('Laplacian filter'), plt.xticks([]), plt.yticks([])
```

```
%matplotlib inline
img = cv2.imread('bonescan.jpg',0)

laplacian = cv2.Laplacian(img, cv2.CV_64F)

plt.subplot, plt.imshow(laplacian, cmap = 'gray')
plt.title('Laplacian filter'), plt.xticks([]), plt.yticks([])
```

```
(Text(0.5, 1.0, 'Laplacian filter'),
 ([], <a list of 0 Text major ticklabel objects>),
 ([], <a list of 0 Text major ticklabel objects>))
```



```
print(img.shape)


img=cv2.imread('bonescan.jpg',0)

laplacian=cv2.Laplacian(img, cv2.CV_64F)

sub = cv2.subtract(img,laplacian,dtype=cv2.CV_64F)
cv2_imshow(sub)
```

```
[ ] print(img.shape)
```

```
(334, 184)
```

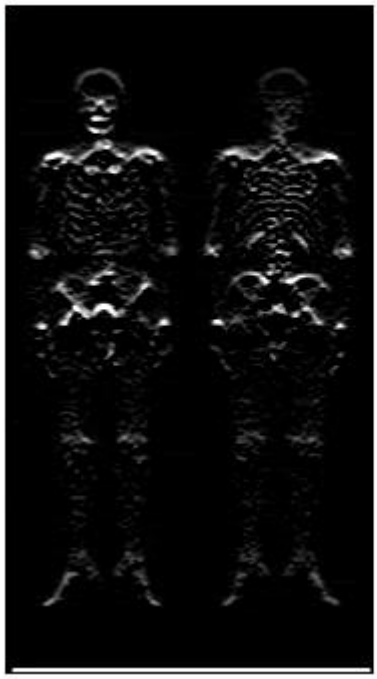
```
img=cv2.imread('bonescan.jpg',0)

laplacian=cv2.Laplacian(img, cv2.CV_64F)

sub = cv2.subtract(img,laplacian,dtype=cv2.CV_64F)
cv2_imshow(sub)
```

```
#Sobel filter of bonescan image

# sobel =cv2.Laplacian(img, cv2.CV_64F)

img = cv2.imread('bonescan.jpg')
sobelx = cv2.Sobel(img, cv2.CV_64F, 1, 0)
cv2_imshow(sobelx)
```


```
#Sobel filter of bonescan image

# sobel =cv2.Laplacian(img, cv2.CV_64F)

img = cv2.imread('bonescan.jpg')
sobelx = cv2.Sobel(img, cv2.CV_64F, 1, 0)
cv2_imshow(sobelx)
```

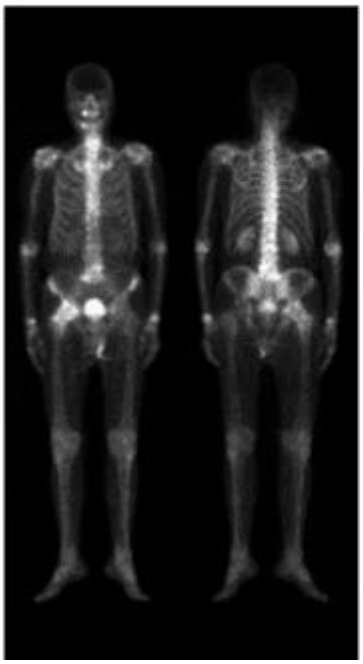

```
sobely = cv2.Sobel(img, cv2.CV_64F, 0,1)
cv2_imshow(sobely)
```

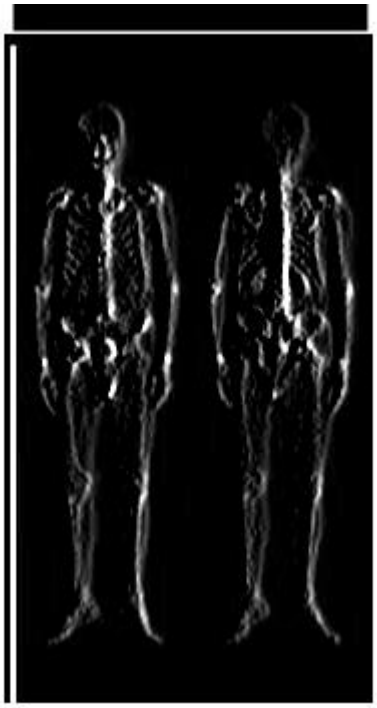
```
sobely = cv2.Sobel(img, cv2.CV_64F, 0,1)
cv2_imshow(sobely)
```



```
cv2_imshow(img)
mask = np.ones([5,5])/25
blurred = cv2.filter2D(sobelx,cv2.CV_64F,-1, mask)
cv2_imshow(blurred)
```

```
cv2_imshow(img)
mask = np.ones([5,5])/25
blurred = cv2.filter2D(sobelx,cv2.CV_64F,-1, mask)
cv2_imshow(blurred)
```





```
print(sub.shape)
print(blurred.shape)


bitand = cv2.bitwise_and(sub, laplacian)
```

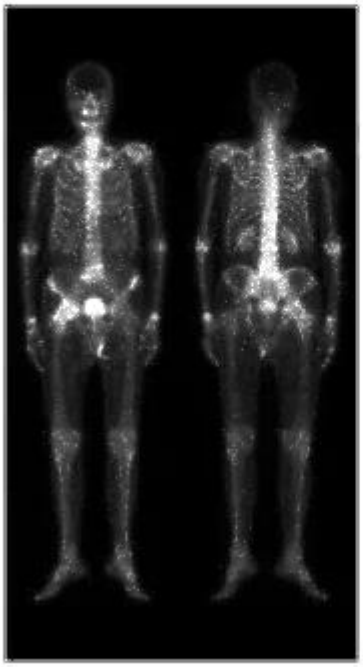
```
[ ]  print(sub.shape)
     print(blurred.shape)

     (334, 184)
     (334, 184, 3)

[ ]  bitand = cv2.bitwise_and(sub, laplacian)

     cv2_imshow(bitand)
```

```
imgn = cv2.imread('bonescan.jpg', 0)


# add = cv2.add(img,laplacian,dtype=cv2.CV_64F)
add = cv2.add(imgn, bitand, dtype = cv2.CV_64F)


cv2_imshow(add)# image g
```

```
[ ]  imgn = cv2.imread('bonescan.jpg', 0)
```

```
[ ]  # add = cv2.add(img,laplacian,dtype=cv2.CV_64F)
     add = cv2.add(imgn, bitand, dtype = cv2.CV_64F)
```

```
     cv2_imshow(add)# image g
```



```
#power law transformation
powerlaw = np.array(255 * (add/255) ** 0.6, dtype = 'uint8')
```

```
[ ]  #power law transformation
     powerlaw = np.array(255 * (add/255) ** 0.6, dtype = 'uint8')
```

```
cv2_imshow(powerlaw)
```

```
#Comparing the original and final image

cv2_imshow(img)
cv2_imshow(powerlaw)
```

```
#Comparing the original and final image

cv2_imshow(img)
cv2_imshow(powerlaw)
```