```
In [1]:  import cv2
         import matplotlib.pyplot as plt
         from ipywidgets import interactive
         import numpy as np
```

```
In [2]:  change = cv2.imread('./images/one-piece.jpg', cv2.IMREAD_GRAYSCALE)
```

```
In [3]:  def my_threshold(thresh):
             global dst
             global change
             th, dst = cv2.threshold(change, thresh, 255, cv2.THRESH_BINARY)
             fig, axes = plt.subplots(1, 2, figsize=(12,16))
             axes[0].imshow(change, 'gray')
             axes[1].imshow(dst, 'gray')

         interactive(my_threshold, thresh=(0, 255))
```

```
         interactive(children=(IntSlider(value=127, description='thresh', max=255), Output()), _d
         om_classes=('widget-in…
```

```
In [7]:  def my_blur(x, filtro):
             global dst
             global change
             match filtro:
                 case 'GaussianBlur':
                     dst = cv2.GaussianBlur(change,(x,y),cv2.BORDER_DEFAULT)

             fig, axes = plt.subplots(1, 2, figsize=(12,16))
             axes[0].imshow(change, 'gray')
             axes[1].imshow(dst, 'gray')
         filtros = ['GaussianBlur']
         interactive(my_blur, x=(1, 255, 2), filtro=filtros)
```

```
           File "C:\Users\joseg\AppData\Local\Temp\ipykernel_8960\3020035044.py", line 4
             match filtro:
                   ^
         SyntaxError: invalid syntax
```

```
In [37]:  def my_canny(x, y):
              global dst
              global change
              dst = cv2.Canny(change,x,y)
              fig, axes = plt.subplots(1, 2, figsize=(12,16))
              axes[0].imshow(change, 'gray')
              axes[1].imshow(dst, 'gray')

          interactive(my_canny, x=(0, 255), y=(0, 255))
```
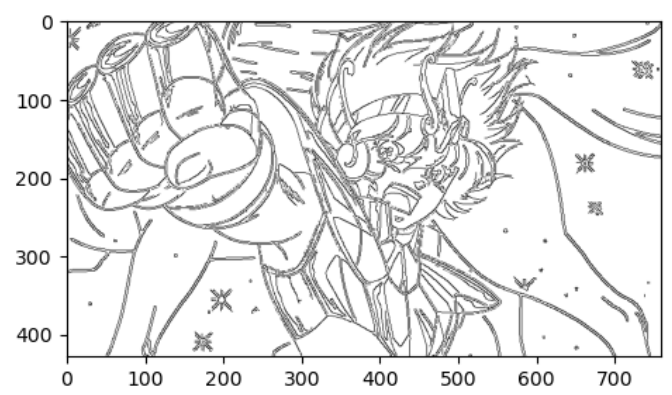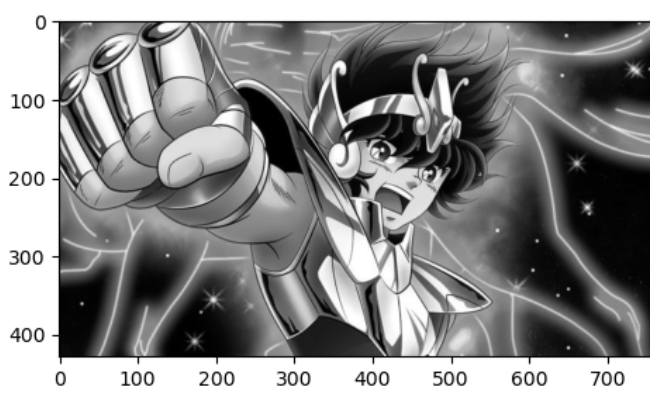
```
          interactive(children=(IntSlider(value=127, description='x', max=255), IntSlider(value=12
          7, description='y', ma…
```

```
In [5]:  #criando modelos
         myimage = cv2.imread('./images/v-c.jpg', cv2.IMREAD_GRAYSCALE)
         edge = cv2.Canny(myimage,100,200)
         th, dst = cv2.threshold(edge, 127, 255, cv2.THRESH_BINARY_INV+cv2.THRESH_OTSU)
         th, dst_bi = cv2.threshold(myimage, 127, 255, cv2.THRESH_BINARY)
         result = dst
         fig, axes = plt.subplots(1, 2, figsize=(12,16))
         axes[0].imshow(myimage, 'gray')
         axes[1].imshow(result, 'gray')
```

```
Out[5]:  <matplotlib.image.AxesImage at 0x199bac0bee0>
```