Name: Drumil Kotecha

Roll no. C052

Aim: To detect edges of the given image using DOG

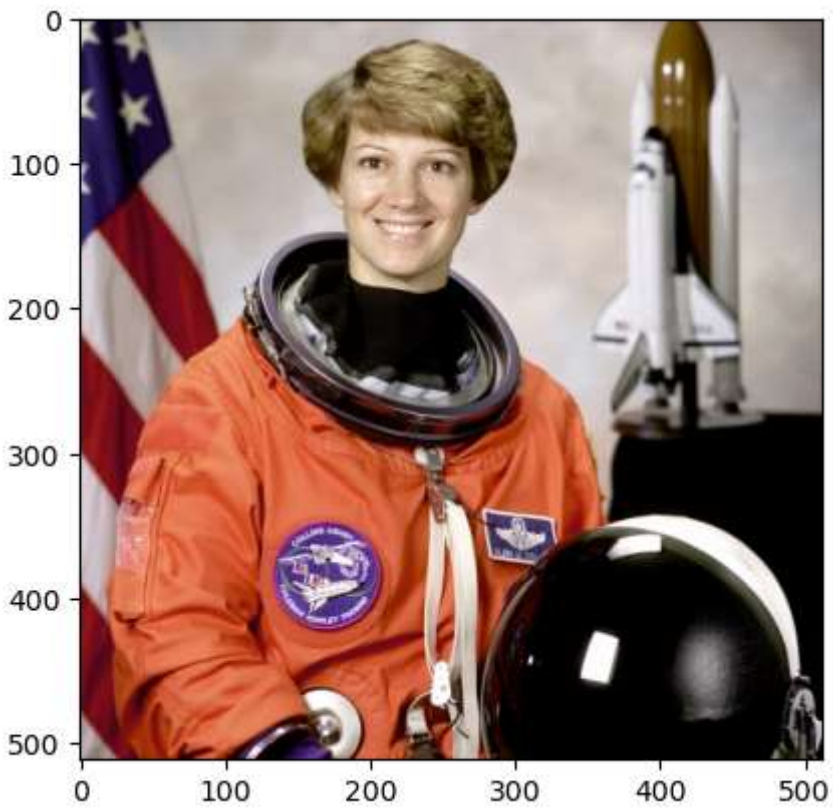
In [4]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as pl
import cv2
from skimage import data
from skimage.color import rgb2gray
```

In [10]:
```
img = data.astronaut()
img.shape
```

Out[10]:
```
(512, 512, 3)
```

In [15]:
```
plt.imshow(img)
```

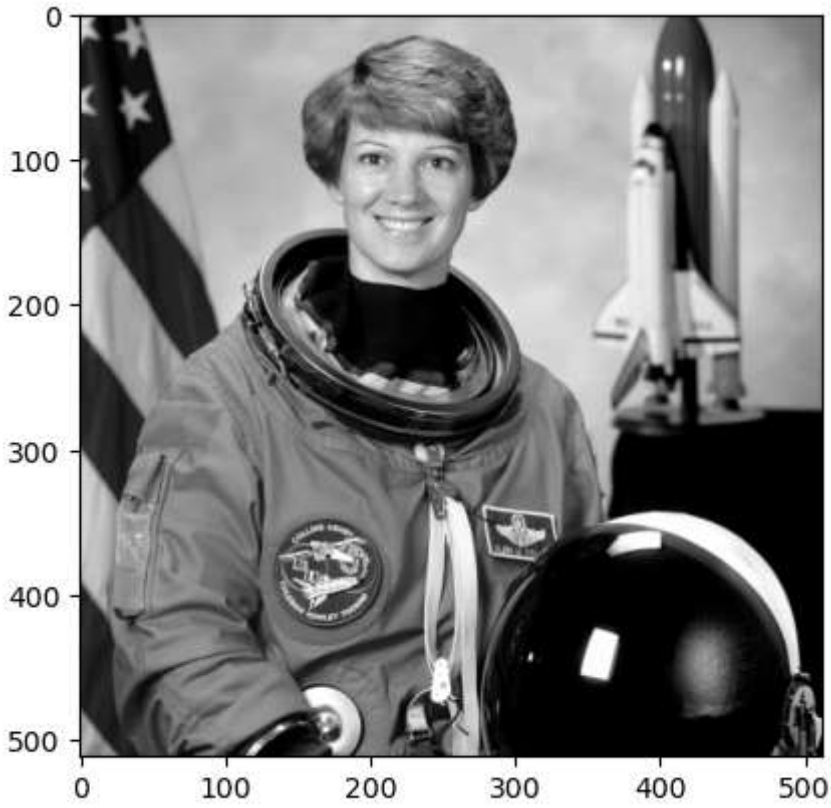
Out[15]:
```
<matplotlib.image.AxesImage at 0x1e311cba5d0>
```



In [16]:
```
image_gray = rgb2gray(img)
```

In [13]:
```
plt.imshow(image_gray, cmap='gray')
```

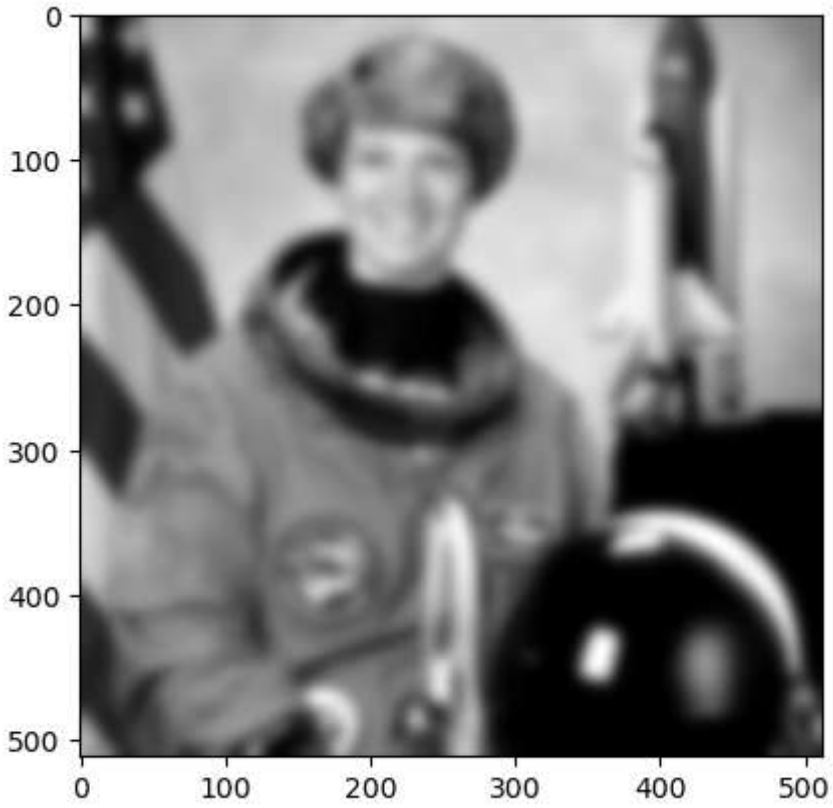
Out[13]:
```
<matplotlib.image.AxesImage at 0x1e311e3c390>
```



In [23]:
```
image_blur_low1= cv2.GaussianBlur(image_gray,(31,31),1)
image_blur_low2= cv2.GaussianBlur(image_gray,(31,31),2)
image_blur_low3= cv2.GaussianBlur(image_gray,(31,31),3)
image_blur_low4= cv2.GaussianBlur(image_gray,(31,31),4)
image_blur_low5= cv2.GaussianBlur(image_gray,(31,31),5)
```

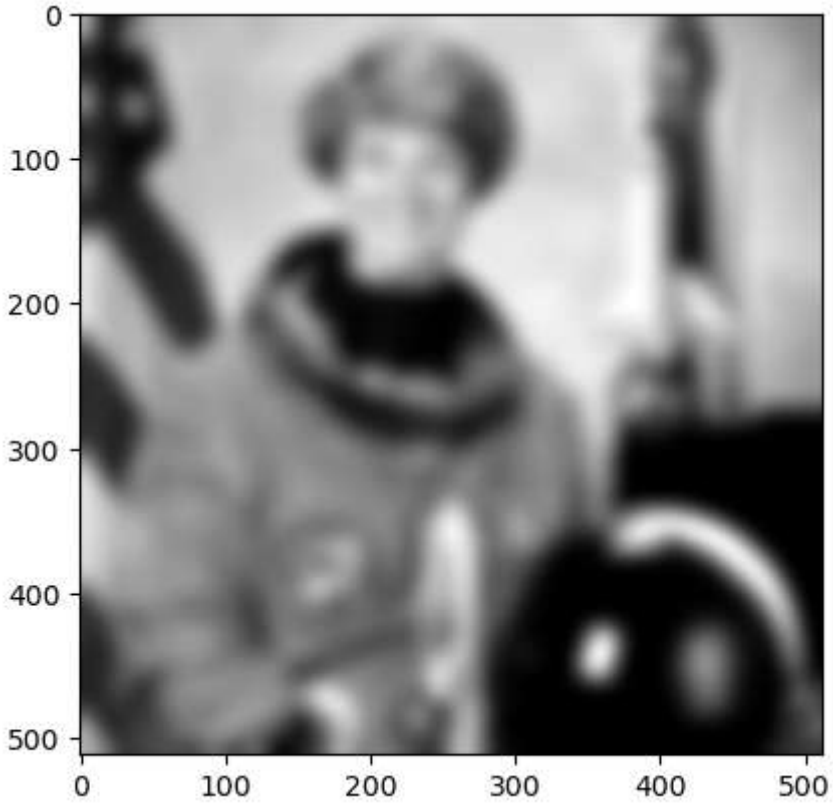
In [24]:
```
plt.imshow(image_blur_low5, cmap='gray')
```

Out[24]:
```
<matplotlib.image.AxesImage at 0x1e314fcbd10>
```



In [26]:
```
image_blur_high8= cv2.GaussianBlur(image_gray,(31,31),8)
plt.imshow(image_blur_high8, cmap='gray')
```

Out[26]:
```
<matplotlib.image.AxesImage at 0x1e311e5ce50>
```

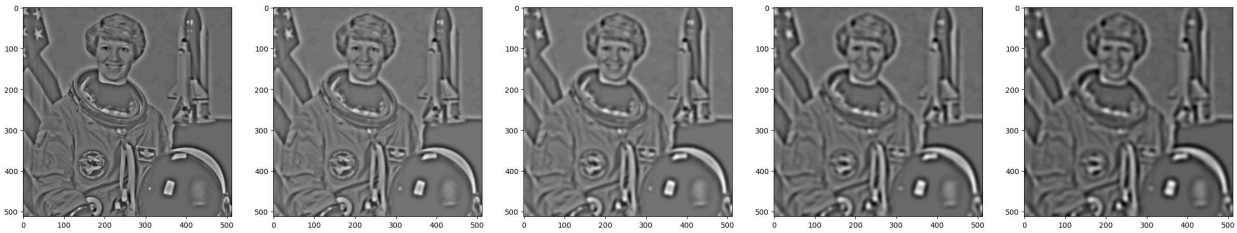


In [27]:
```
dog1=image_blur_low1-image_blur_high8
dog2=image_blur_low2-image_blur_high8
dog3=image_blur_low3-image_blur_high8
dog4=image_blur_low4-image_blur_high8
dog5=image_blur_low5-image_blur_high8
```

In [29]:
```
fig, axs = plt.subplots(1, 5, figsize=(30, 30))

axs[0].imshow(dog1, cmap='gray')
axs[1].imshow(dog2, cmap='gray')
axs[2].imshow(dog3, cmap='gray')
axs[3].imshow(dog4, cmap='gray')
axs[4].imshow(dog5, cmap='gray')
```

Out[29]:
```
<matplotlib.image.AxesImage at 0x1e31aed9ad0>
```

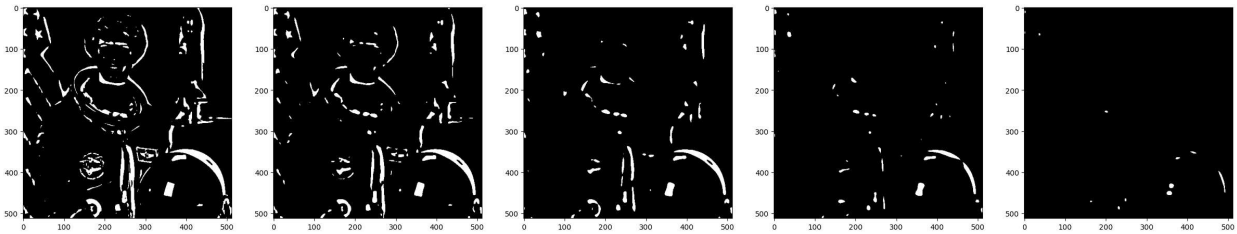


In [32]:
```
mx=np.max(dog1)
th=0.2*mx
dog1_th = np.where(dog1>th,255,dog1)
dog2_th = np.where(dog2>th,255,dog2)
dog3_th = np.where(dog3>th,255,dog3)
dog4_th = np.where(dog4>th,255,dog4)
dog5_th = np.where(dog5>th,255,dog5)

fig, axs = plt.subplots(1, 5, figsize=(30, 30))

axs[0].imshow(dog1_th, cmap='gray')
axs[1].imshow(dog2_th, cmap='gray')
axs[2].imshow(dog3_th, cmap='gray')
axs[3].imshow(dog4_th, cmap='gray')
axs[4].imshow(dog5_th, cmap='gray')
```

Out[32]:
```
<matplotlib.image.AxesImage at 0x1e31adeb510>
```



Conclusion: Difference of Gaussian (DOG) is applied to identify the edges of the given image for the following parts of standard deviation. It shows different number of edge pixels.

SD = 1,8

SD = 2,8

SD = 3,8

SD = 4,8

SD = 5,8

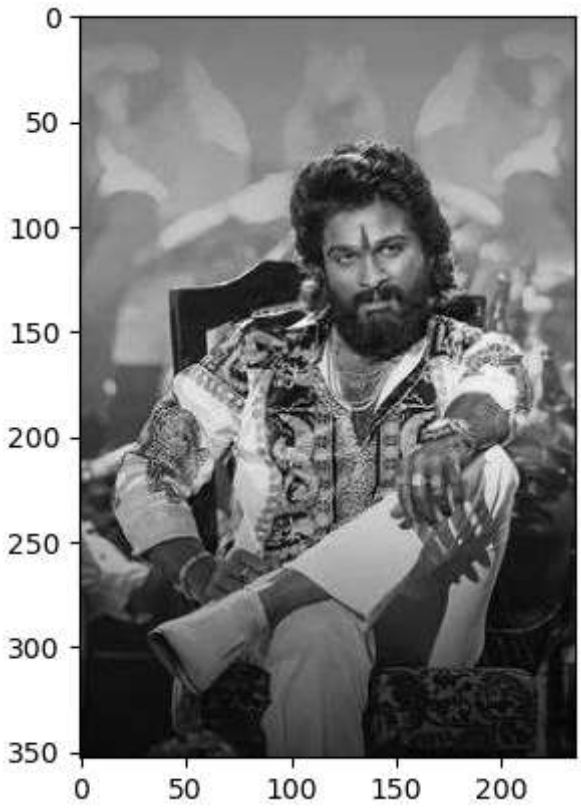
--> For the first pair i.e. (1,8), the smaller edges are noticable

--> For the first pair i.e. (5,8), only larger pixels are retained

Small edge pixels can be retained by using low sigma values (1) for the given image Only large edge pixels can be identified (5).

In [99]:
```
leo = rgb2gray(cv2.imread('pushpa.jpg'))
plt.imshow(leo, cmap='gray')
```

Out[99]:
```
<matplotlib.image.AxesImage at 0x1e3482aa450>
```



In [100…
```
leo.shape
```

Out[100]:
```
(353, 236)
```

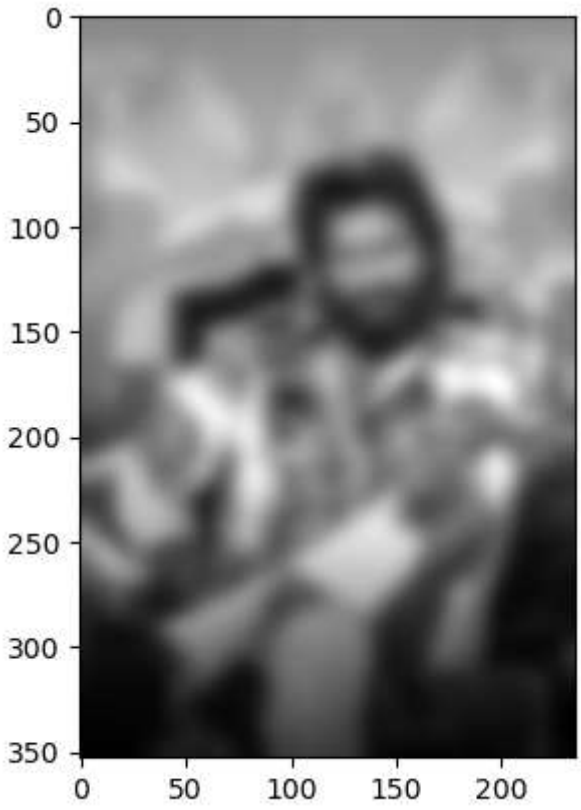
In [101…
```
leo.size
```

Out[101]:
```
83308
```

In [102…
```
image_blur_low1_leo= cv2.GaussianBlur(leo,(31,31),1)
image_blur_low2_leo= cv2.GaussianBlur(leo,(31,31),2)
image_blur_low3_leo= cv2.GaussianBlur(leo,(31,31),3)
image_blur_low4_leo= cv2.GaussianBlur(leo,(31,31),4)
image_blur_low5_leo= cv2.GaussianBlur(leo,(31,31),5)
```

In [103…
```
plt.imshow(image_blur_low5_leo, cmap='gray')
```

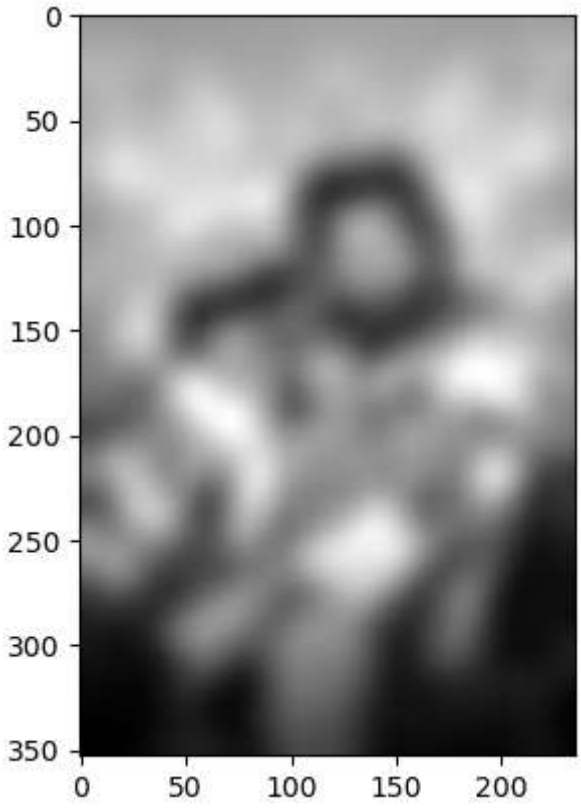
Out[103]: <matplotlib.image.AxesImage at 0x1e348331250>



In [104…

```
image_blur_high9_leo= cv2.GaussianBlur(leo,(31,31),9)
plt.imshow(image_blur_high9_leo, cmap='gray')
```

Out[104]: <matplotlib.image.AxesImage at 0x1e3483a2250>

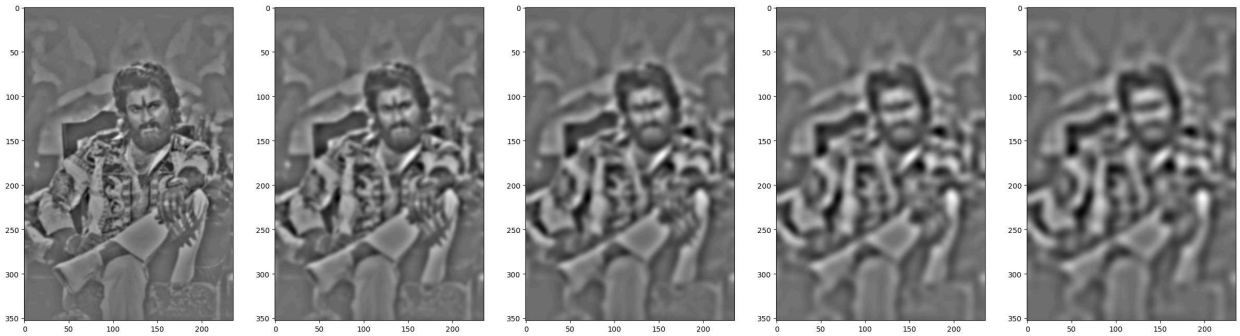


In [105…
```
dog1_leo=image_blur_low1_leo-image_blur_high9_leo
dog2_leo=image_blur_low2_leo-image_blur_high9_leo
dog3_leo=image_blur_low3_leo-image_blur_high9_leo
dog4_leo=image_blur_low4_leo-image_blur_high9_leo
dog5_leo=image_blur_low5_leo-image_blur_high9_leo
```

In [106…
```
fig, axs = plt.subplots(1, 5, figsize=(30, 30))

axs[0].imshow(dog1_leo, cmap='gray')
axs[1].imshow(dog2_leo, cmap='gray')
axs[2].imshow(dog3_leo, cmap='gray')
axs[3].imshow(dog4_leo, cmap='gray')
axs[4].imshow(dog5_leo, cmap='gray')
```

Out[106]:
```
<matplotlib.image.AxesImage at 0x1e348393510>
```

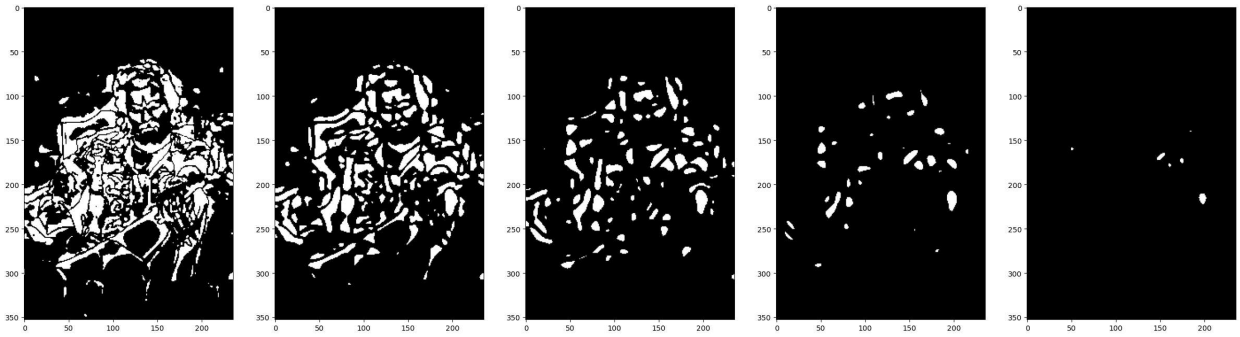


In [110…
```
mx=np.max(dog1_leo)
th=0.3*mx
dog1_th_leo = np.where(dog1_leo>th,255,dog1_leo)
dog2_th_leo = np.where(dog2_leo>th,255,dog2_leo)
dog3_th_leo = np.where(dog3_leo>th,255,dog3_leo)
dog4_th_leo = np.where(dog4_leo>th,255,dog4_leo)
dog5_th_leo = np.where(dog5_leo>th,255,dog5_leo)

fig, axs = plt.subplots(1, 5, figsize=(30, 30))

axs[0].imshow(dog1_th_leo, cmap='gray')
axs[1].imshow(dog2_th_leo, cmap='gray')
axs[2].imshow(dog3_th_leo, cmap='gray')
axs[3].imshow(dog4_th_leo, cmap='gray')
axs[4].imshow(dog5_th_leo, cmap='gray')
```

Out[110]:
```
<matplotlib.image.AxesImage at 0x1e34c3c6c10>
```



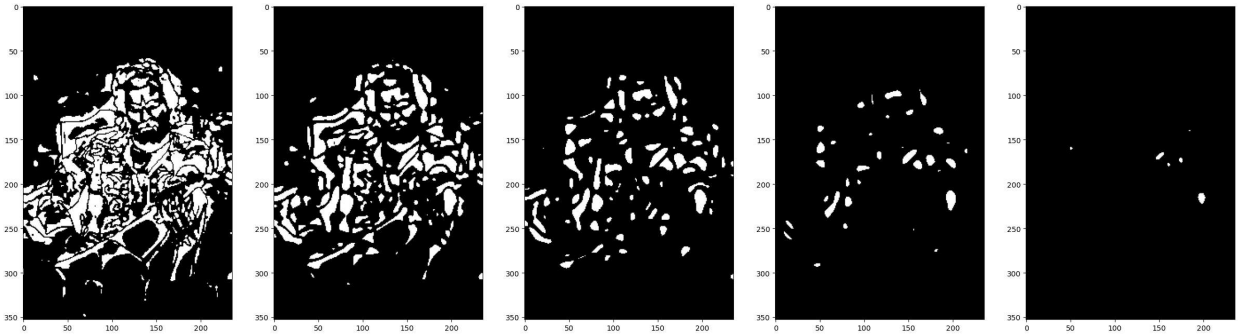
In [108…
```
mx=np.max(dog1_leo)
th=0.2*mx
dog1_leo[np.abs(dog1_leo) > th] = 255
```

```
dog2_leo[np.abs(dog2_leo) > th] = 255
dog3_leo[np.abs(dog3_leo) > th] = 255
dog4_leo[np.abs(dog4_leo) > th] = 255
dog5_leo[np.abs(dog5_leo) > th] = 255

fig, axs = plt.subplots(1, 5, figsize=(30, 30))

axs[0].imshow(dog1_leo, cmap='gray')
axs[1].imshow(dog2_leo, cmap='gray')
axs[2].imshow(dog3_leo, cmap='gray')
axs[3].imshow(dog4_leo, cmap='gray')
axs[4].imshow(dog5_leo, cmap='gray')
```

Out[108]:
```
<matplotlib.image.AxesImage at 0x1e349051ad0>
```

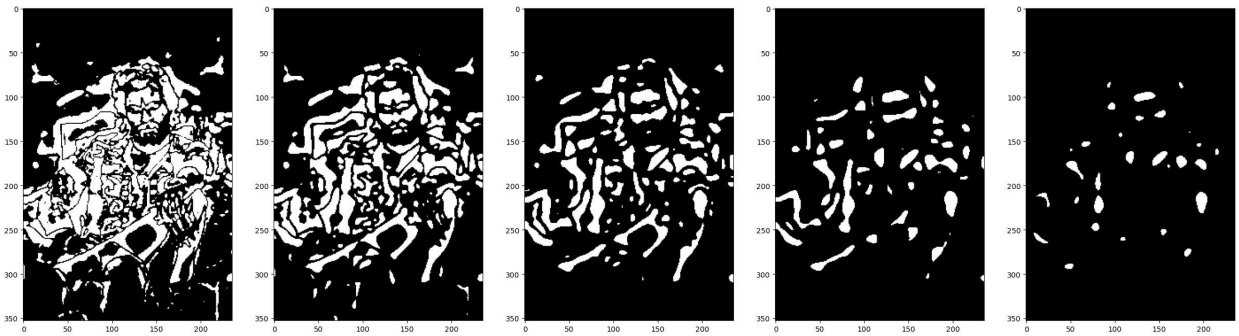


In [109…
```
mx=np.max(dog1)
th=0.1*mx
dog1_th_leo = np.where(dog1_leo>th,255,dog1_leo)
dog2_th_leo = np.where(dog2_leo>th,255,dog2_leo)
dog3_th_leo = np.where(dog3_leo>th,255,dog3_leo)
dog4_th_leo = np.where(dog4_leo>th,255,dog4_leo)
dog5_th_leo = np.where(dog5_leo>th,255,dog5_leo)

fig, axs = plt.subplots(1, 5, figsize=(30, 30))

axs[0].imshow(dog1_th_leo, cmap='gray')
axs[1].imshow(dog2_th_leo, cmap='gray')
axs[2].imshow(dog3_th_leo, cmap='gray')
axs[3].imshow(dog4_th_leo, cmap='gray')
axs[4].imshow(dog5_th_leo, cmap='gray')
```

Out[109]:
```
<matplotlib.image.AxesImage at 0x1e34ae3a890>
```



If threshold 20% -> 10%, edges appear thicker and additional smaller edge pixels are retained

If threshold 20% -> 30%, smaller edge pixels disappear even for low values of (\sigma)