

PCD 11

J0303201065 Rahma Fairuz Rania

1. Skeletonisasi mengurangi objek biner

In [1]:

```
from skimage.morphology import skeletonize
from skimage import data
import matplotlib.pyplot as plt
from skimage.util import invert

# Invert the horse image
image = invert(data.horse())

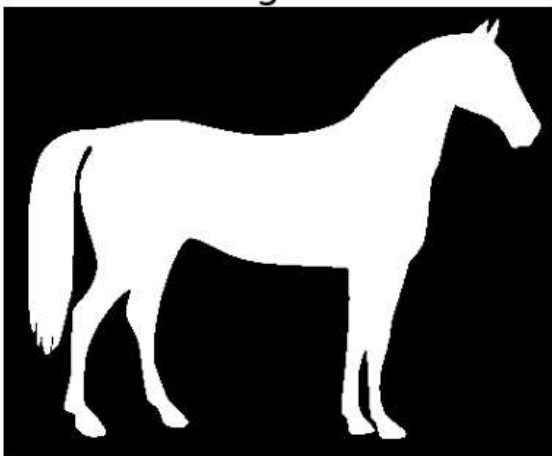
# perform skeletonization
skeleton = skeletonize(image)

# display results
fig, axes = plt.subplots(nrows=1, ncols=2,
figsize=(8, 4), sharex=True, sharey=True)
ax = axes.ravel()

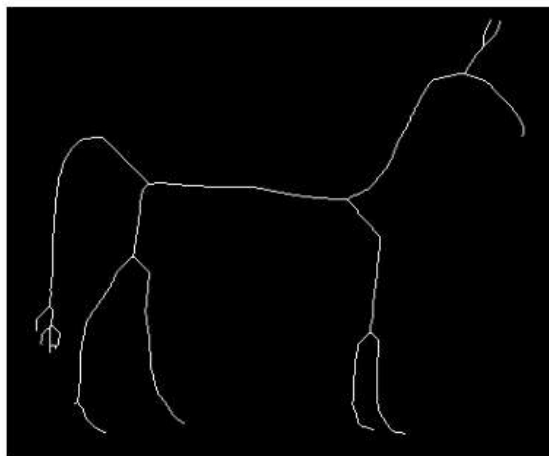
ax[0].imshow(image, cmap=plt.cm.gray)
ax[0].axis('off')
ax[0].set_title('original', fontsize=20)
ax[1].imshow(skeleton, cmap=plt.cm.gray)
ax[1].axis('off')
ax[1].set_title('skeleton', fontsize=20)
fig.tight_layout()

plt.show()
```

original



skeleton



2. Active contour

In [2]:

```

import numpy as np
import matplotlib.pyplot as plt
from skimage.color import rgb2gray
from skimage import data
from skimage.filters import gaussian
from skimage.segmentation import active_contour

img = data.astronaut()

# Data for circular boundary
s = np.linspace(0, 2*np.pi, 400)
x = 220 + 100*np.cos(s)
y = 100 + 100*np.sin(s)
init = np.array([x, y]).T

# formation of the active contour
cntr = active_contour(gaussian(img, 3), init, alpha=0.015, beta=10, gamma=0.001)
fig, ax = plt.subplots(1, 2, figsize=(7, 7))
ax[0].imshow(img, cmap=plt.cm.gray)
ax[0].set_title("Original Image")
ax[1].imshow(img, cmap=plt.cm.gray)

# circular boundary
ax[1].plot(init[:, 0], init[:, 1], '--r', lw=3)
ax[1].plot(cntr[:, 0], cntr[:, 1], '-b', lw=3)
ax[1].set_title("Active Contour Image")

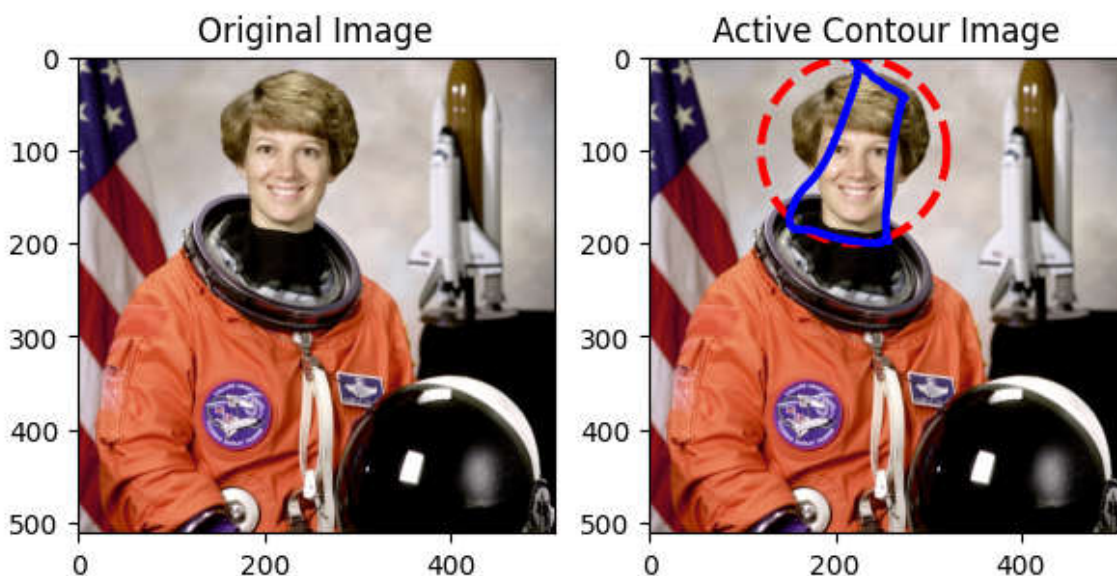
```

C:\Users\Asus\AppData\Local\Temp\ipykernel_12948\203992475.py:17: RuntimeWarning: Images with dimensions (M, N, 3) are interpreted as 2D+RGB by default. Use `multichannel=False` to interpret as 3D image with last dimension of length 3.

```
cntr = active_contour(gaussian(img, 3), init, alpha=0.015, beta=10, gamma=0.001)
```

Out[2]:

```
Text(0.5, 1.0, 'Active Contour Image')
```



3. Deteksi wajah

In [3]:

```
import io
import zipfile
import requests
import numpy as np
import cv2
import matplotlib.pyplot as plt
%matplotlib inline

img = cv2.imread('p.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
path = 'haarcascade_frontalface_default.xml'
face_cascade = cv2.CascadeClassifier(path)

for x, y, w, h in face_cascade.detectMultiScale(gray, 1.3):
    cv2.rectangle(gray, (x, y), (x + w, y + h), (255, 0, 0), 2)

fig, ax = plt.subplots(1, 1, figsize=(8, 6))
ax.imshow(gray, cmap=plt.cm.gray)
ax.set_axis_off()
```

