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

READING IMAGE

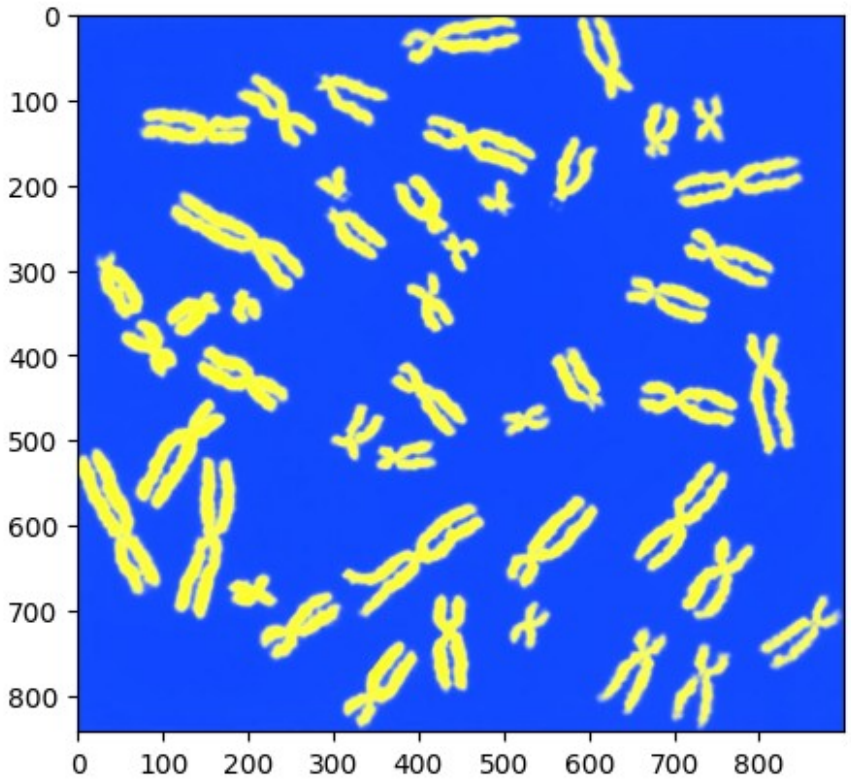
```
img=mpimg.imread('chromosomes.jpg')
print(img)
print(img.shape)
```

```
[[[ 20  75 254]
  [ 20  75 254]
  [ 20  75 254]
  ...
  [ 21  77 252]
  [ 22  78 253]
  [ 22  78 253]]

 [[ 20  75 254]
  [ 20  75 254]
  [ 20  75 254]
  ...
  [ 21  76 254]
  [ 21  76 254]
  [ 22  77 255]]

 [[ 20  75 254]
  [ 20  75 254]
  [ 20  75 254]
  ...
  [ 20  75 253]
  [ 21  76 254]
  [ 21  76 254]]

 ...

 [[ 24  79 255]
  [ 22  77 255]
  [ 20  75 253]
  ...
  [ 23  78 255]
  [ 23  78 255]
  [ 23  78 255]]

 [[ 29  81 253]
  [ 27  79 251]
  [ 25  77 251]
  ...
```

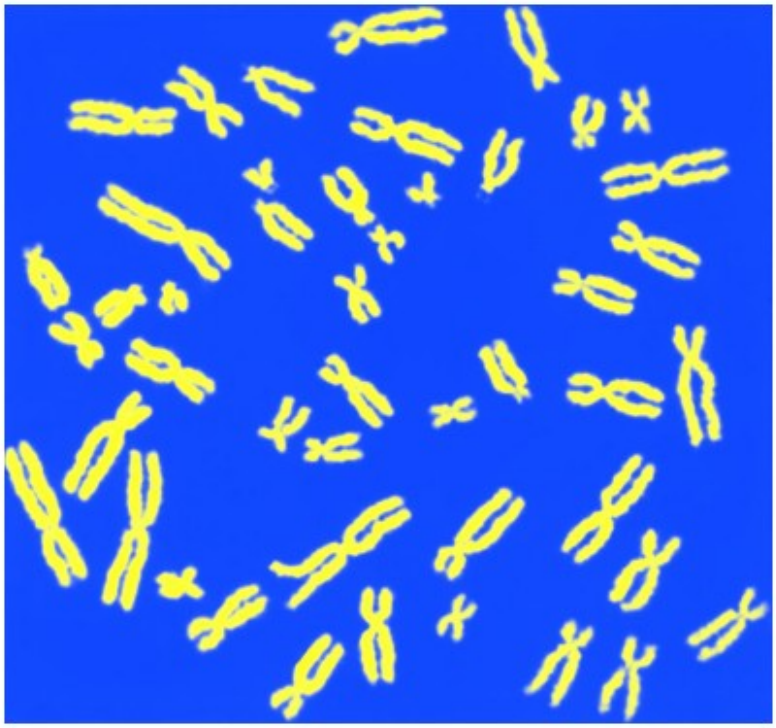
```
  [ 26  77 254]
  [ 26  77 254]
  [ 26  77 254]]

 [[ 29  81 253]
  [ 27  79 251]
  [ 25  77 251]
  ...
  [ 26  77 254]
  [ 26  77 254]
  [ 26  77 254]]]
(842, 900, 3)
```

PLOTTING IMAGE

```
imgplot=plt.imshow(img)
```

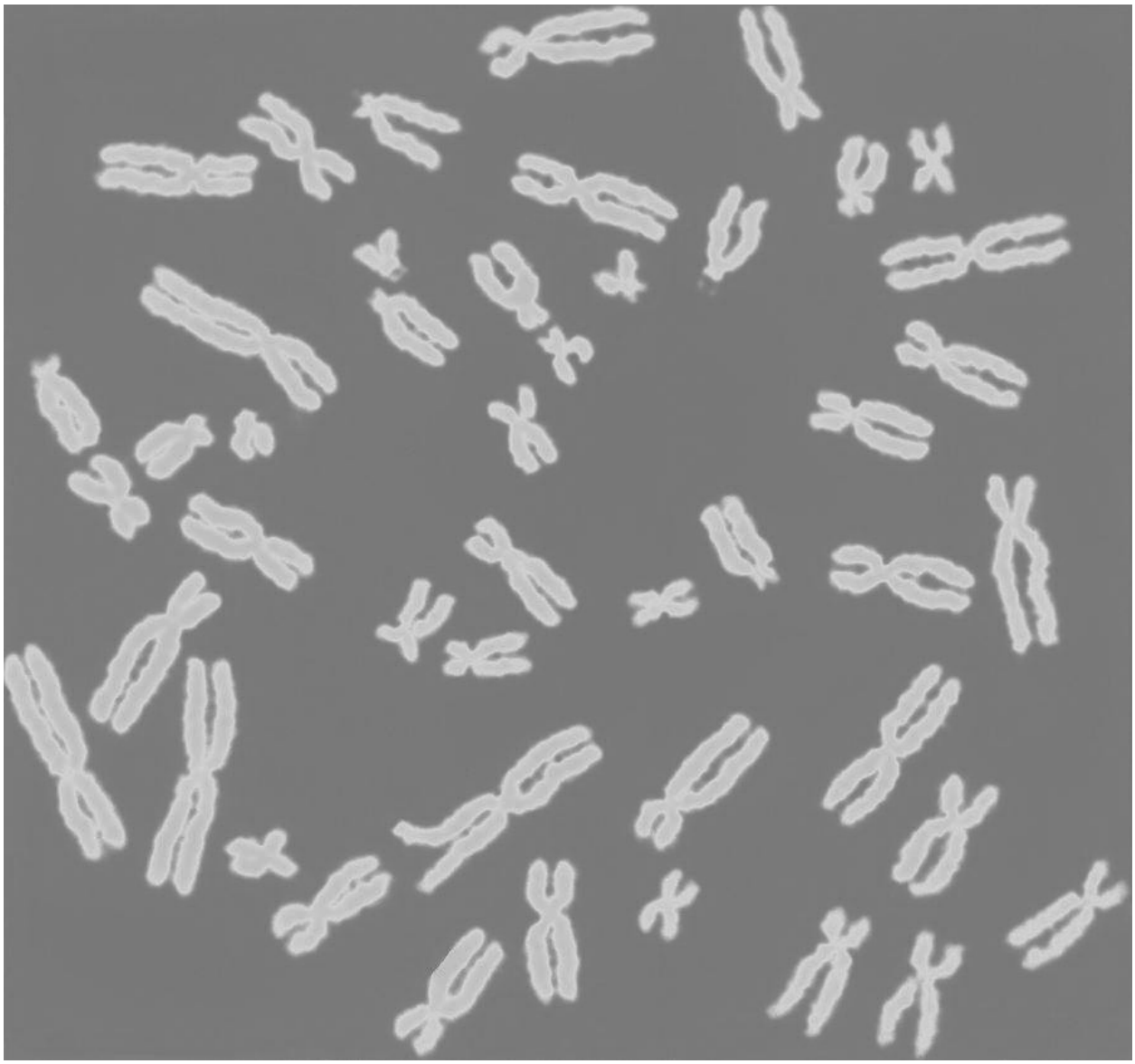



```
#REMOVING AXES
plt.axis('off')
plt.imshow(img)

<matplotlib.image.AxesImage at 0x7bc04f163790>
```



CONVERTING TO GRAYSCALE IMAGE

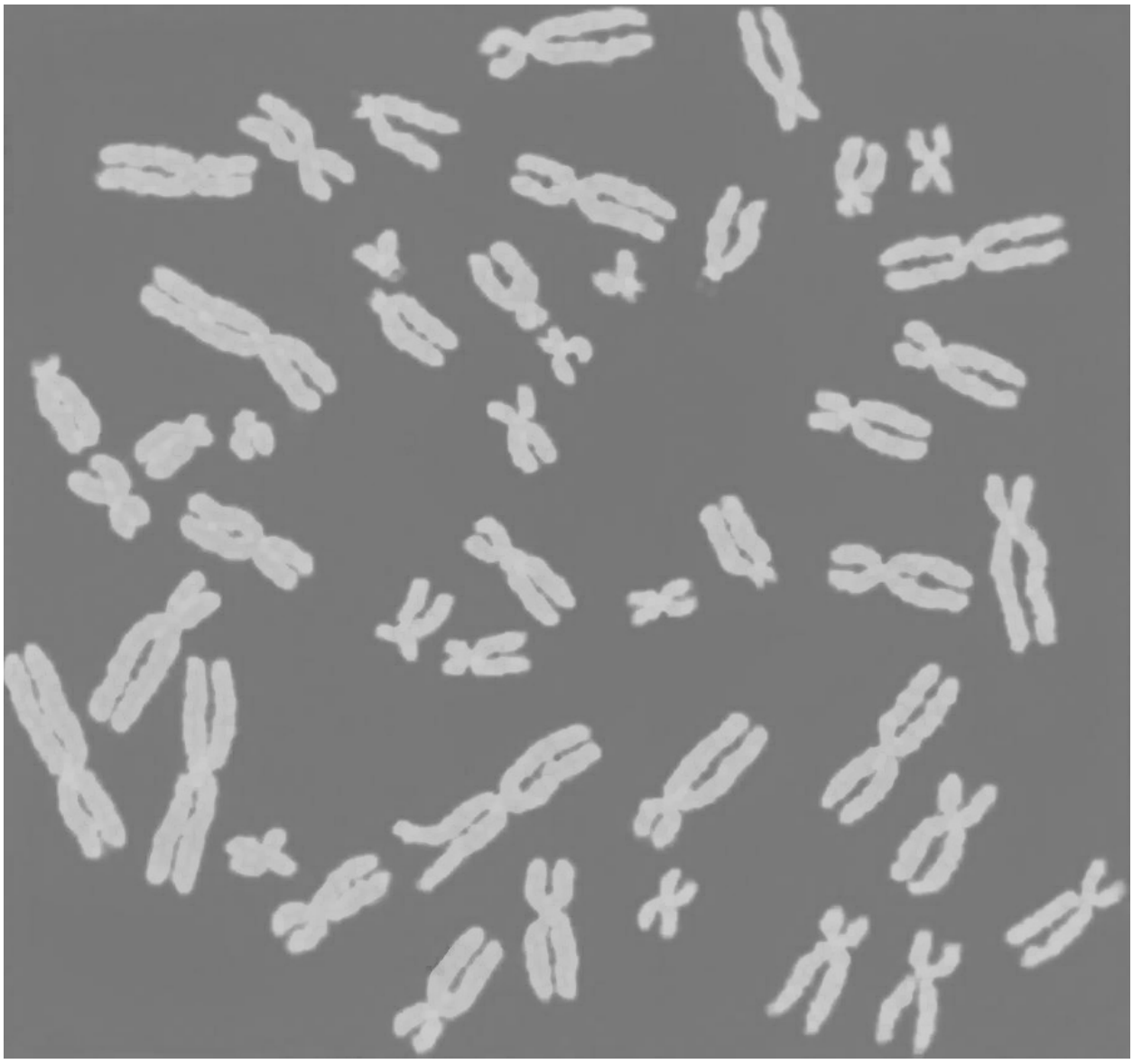
```
gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
cv2_imshow(gray_img)
```

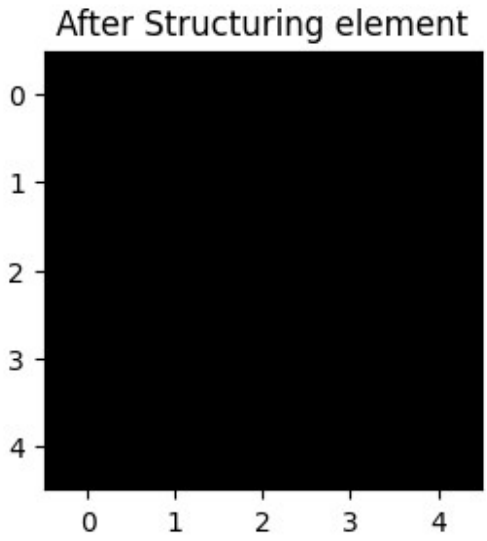


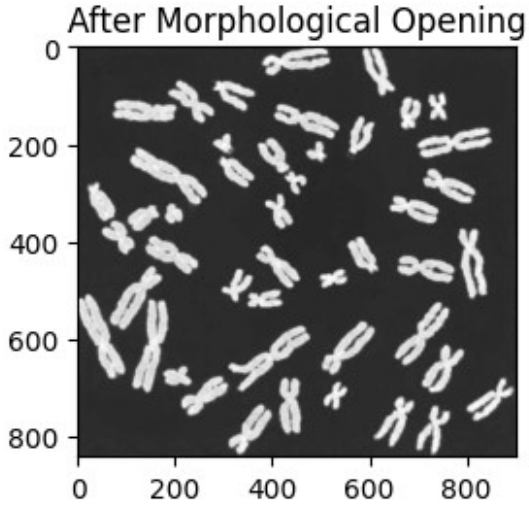
Applying morphological opening for background removal

```
morph_rect = cv2.getStructuringElement(shape=cv2.MORPH_RECT, ksize=(5,
5))
img_open = cv2.morphologyEx(gray_img, cv2.MORPH_OPEN, morph_rect)
cv2_imshow(img_open)
cv2.waitKey(0)
cv2.destroyAllWindows()


plt.subplot(1,2,1)
plt.imshow(morph_rect, cmap='gray')
plt.title('After Structuring element')
plt.show()
```

```
plt.subplot(1,2,2)
plt.imshow(img_open, cmap='gray')
plt.title('After Morphological Opening')
plt.show()
```

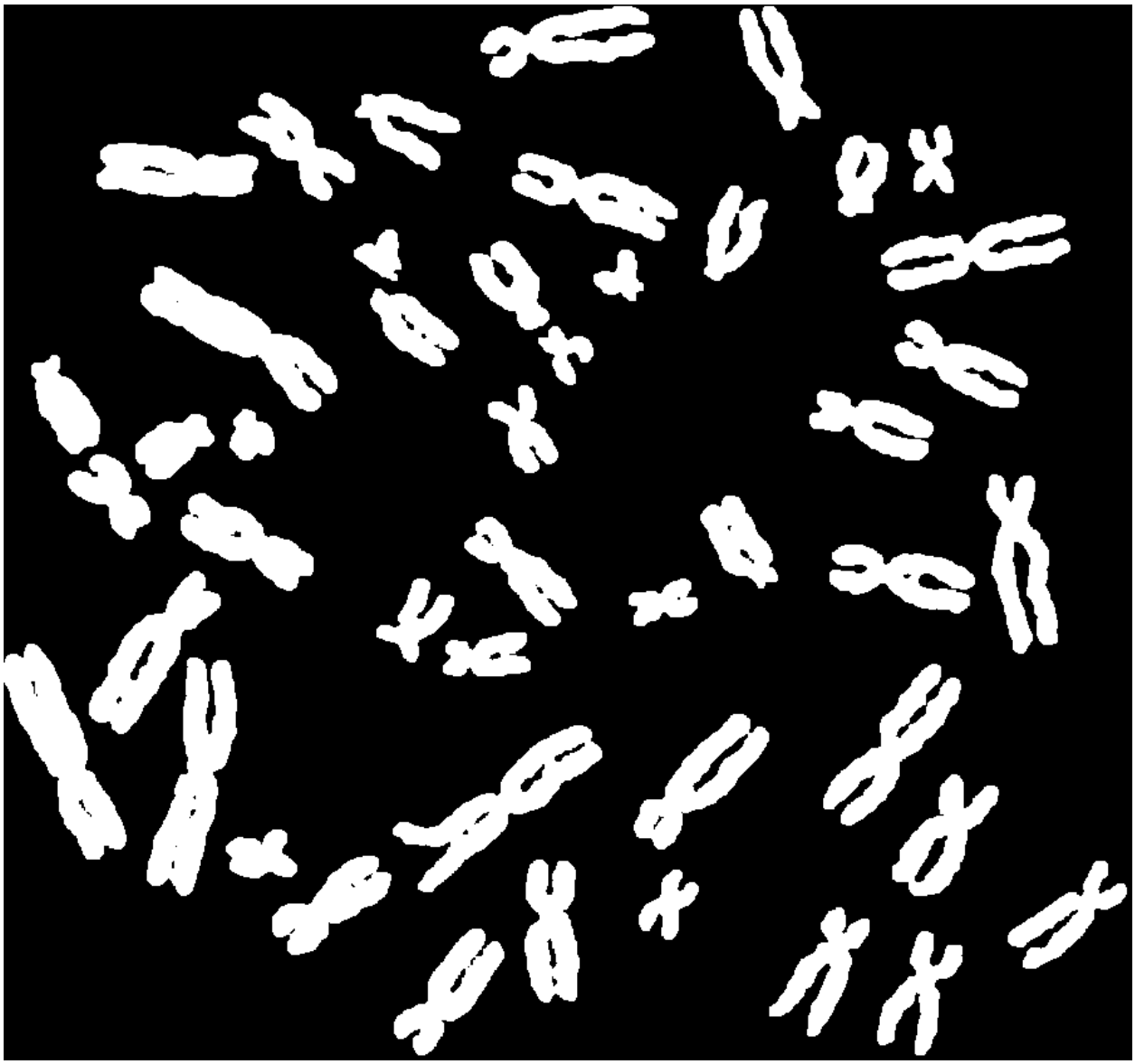






Binarization

```
ret,thresh = cv2.threshold(img_open, 0, 255, cv2.THRESH_BINARY +
cv2.THRESH_OTSU)
cv2_imshow(thresh)
```



FIND CONTOURS

```
contours, hierarchy = cv2.findContours(thresh, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
print(len(contours))
```

```
46
```

FEATURES

```
import pandas as pd
df={'X':[],'Y':[],'Width':[],'Height':[],'shape_info':[],'Area':
[],'Perimeter':[],'Circularity':[]}
for i in contours:
  img_area=cv2.contourArea(i)
```

```
  img_perimeter=cv2.arcLength(i,True)
  img_circularity=(4 * np.pi * img_area) / (img_perimeter ** 2)
  x,y,w,h=cv2.boundingRect(i)
  df['X'].append(x)
  df['Y'].append(y)
  df['shape_info'].append(f"{w} x {h}")
  df['Height'].append(h)
  df['Width'].append(w)
  df['Area'].append(img_area)
  df['Perimeter'].append(img_perimeter)
  df['Circularity'].append(img_circularity)
table=pd.DataFrame(df)
table
```

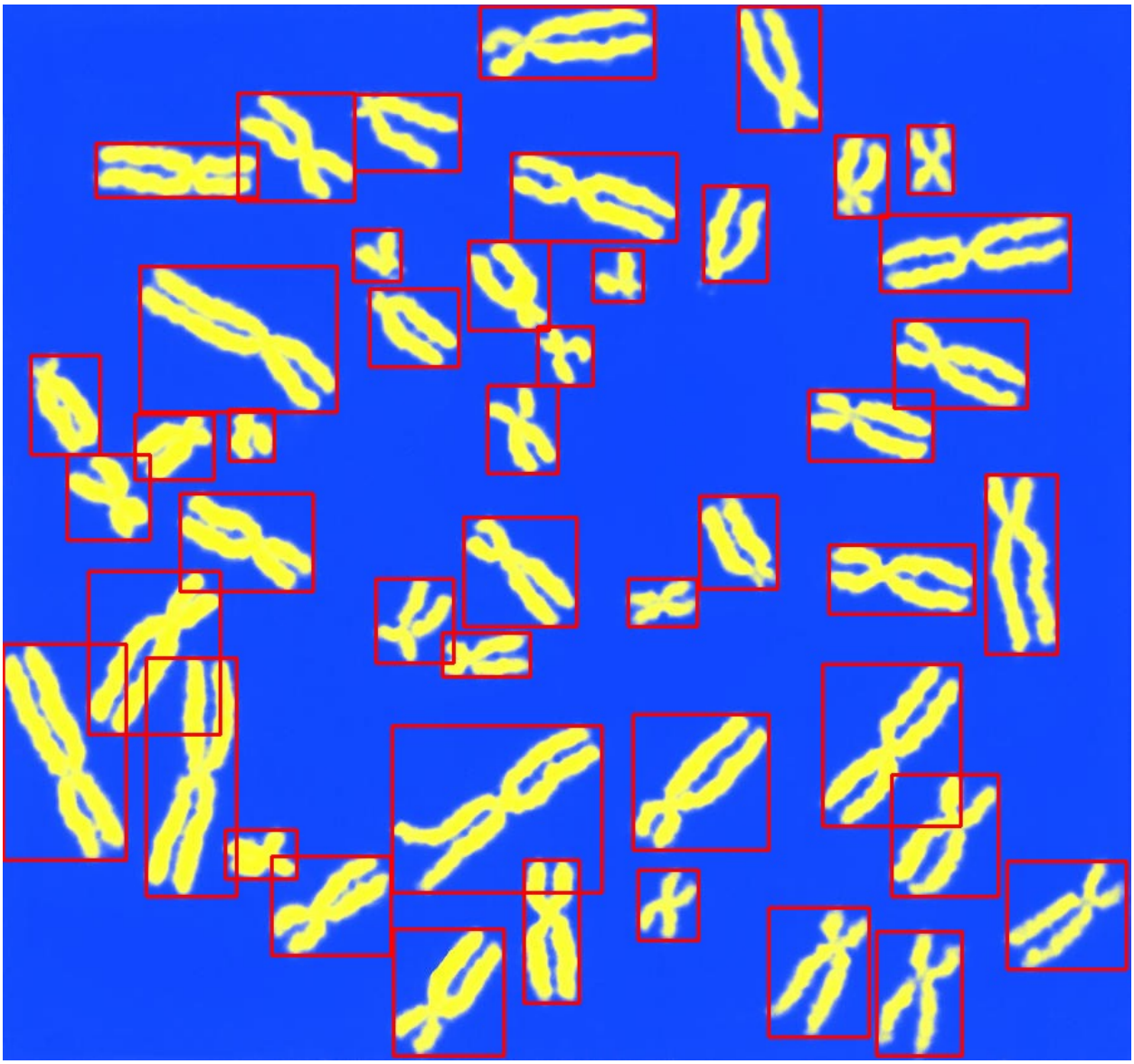
```
      X    Y  Width  Height shape_info    Area   Perimeter
Circularity
0   696  739     68      99    68 x 99  2528.0  415.705624
0.183829
1   311  737     88     101   88 x 101  3550.5  367.747254
0.329914
2   610  720     80     103   80 x 103  2854.5  438.859952
0.186246
3   506  690     48      56    48 x 56  1177.5  214.409161
0.321873
4   800  683     95      86    95 x 86  2626.0  446.901582
0.165227
5   415  682     44     114   44 x 114  3700.0  370.592927
0.338546
6   214  679     95      79    95 x 79  3316.5  303.178713
0.453411
7   177  658     57      39    57 x 39  1446.5  183.095453
0.542217
8   708  614     85      97    85 x 97  3664.0  341.906634
0.393868
9   310  575    167     133  167 x 133  5970.5  547.712764
0.250101
10  502  566    108     108  108 x 108  4409.0  357.989894
0.432323
11  653  526    110     129  110 x 129  4293.0  605.553384
0.147118
12  114  521     72     190   72 x 190  6303.0  605.646749
0.215933
13    0  510     98     172   98 x 172  6365.5  464.374671
0.370941
14  350  501     70      35    70 x 35  1730.5  236.267025
0.389561
15  498  458     55      38    55 x 38  1168.5  178.124890
0.462796
16  297  458     62      67    62 x 67  1645.0  268.936072
```

```
0.285810
17   68  452    105     130  105 x 130  5011.0  385.303603
0.424158
18  658  431    116      55   116 x 55  3371.0  469.989895
0.191775
19  367  409     90      87    90 x 87  2864.5  317.605119
0.356848
20  555  392     62      74    62 x 74  2395.0  229.965510
0.569102
21  141  390    106      78   106 x 78  3647.0  310.735061
0.474641
22  783  375     57     143   57 x 143  4034.5  589.203098
0.146039
23   51  359     66      68    66 x 68  2140.5  245.722869
0.445486
24  105  327     63      52    63 x 52  1908.0  189.338093
0.668825
25  180  323     36      41    36 x 41   971.5  131.639610
0.704498
26  642  308     99      56    99 x 56  2757.5  382.291410
0.237103
27  386  304     56      70    56 x 70  1725.0  271.279218
0.294555
28   22  280     55      79    55 x 79  2343.5  223.722870
0.588375
29  426  257     44      47    44 x 47   923.5  182.267025
0.349326
30  710  252    106      70   106 x 70  3092.0  413.161468
0.227620
31  292  227     71      62    71 x 62  2259.0  240.107646
0.492396
32  109  209    157     116  157 x 116  5730.0  496.357425
0.292264
33  470  196     40      41    40 x 41   840.0  143.740114
0.510897
34  371  189     64      71    64 x 71  2399.0  229.137083
0.574182
35  279  180     38      41    38 x 41   855.0  139.053823
0.555661
36  699  168    151      61   151 x 61  4046.5  636.315796
0.125587
37  558  145     51      76    51 x 76  2227.5  233.722869
0.512419
38  405  119    132      70   132 x 70  4002.0  441.989895
0.257432
39   74  111    129      43   129 x 43  4146.0  332.308655
0.471798
40  663  105     42      65    42 x 65  1899.0  189.539104
0.664259
```

```
41  721   97     36      54    36 x 54  1200.0  224.994947
0.297883
42  280   72     84      61    84 x 61  2240.0  339.906635
0.243634
43  187   71     93      86    93 x 86  2980.5  342.291410
0.319674
44  586    2     65      99    65 x 99  2711.0  403.019332
0.209743
45  380    2    139      57   139 x 57  3888.0  569.303603
0.150747
```

Bounding box

```
for c in contours:
  x,y,w,h=cv2.boundingRect(c)
  cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
# plt.imshow(cv2.cvtColor(img,cv2.COLOR_BGR2RGB))
# plt.title('Bounding Box')
# plt.show()
cv2_imshow(cv2.cvtColor(img,cv2.COLOR_BGR2RGB))
```

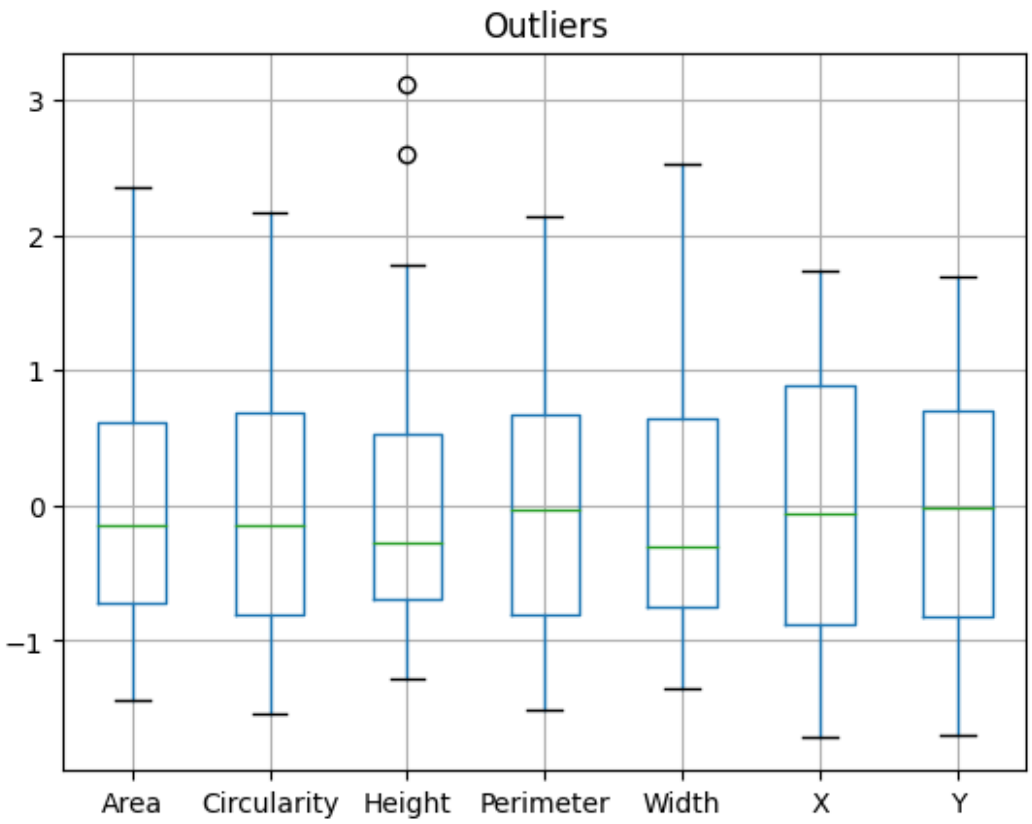


NORMALISING

```
n_df = (table - table.mean()) / table.std()
Q1 = n_df.quantile(0.25)
Q3 = n_df.quantile(0.75)
IQR = Q3 - Q1
lower = Q1 - 1.5 * IQR
upper = Q3 + 1.5 * IQR
outliers = ((n_df < lower) | (n_df > upper)).sum().sum()
outliers
```

```
<ipython-input-54-23f3017b18ad>:1: FutureWarning: The default value of
numeric_only in DataFrame.mean is deprecated. In a future version, it
will default to False. In addition, specifying 'numeric_only=None' is
deprecated. Select only valid columns or specify the value of
```

```
numeric_only to silence this warning.
  n_df = (table - table.mean()) / table.std()
<ipython-input-54-23f3017b18ad>:1: FutureWarning: The default value of
numeric_only in DataFrame.std is deprecated. In a future version, it
will default to False. In addition, specifying 'numeric_only=None' is
deprecated. Select only valid columns or specify the value of
numeric_only to silence this warning.
  n_df = (table - table.mean()) / table.std()
<ipython-input-54-23f3017b18ad>:2: FutureWarning: The default value of
numeric_only in DataFrame.quantile is deprecated. In a future version,
it will default to False. Select only valid columns or specify the
value of numeric_only to silence this warning.
  Q1 = n_df.quantile(0.25)
<ipython-input-54-23f3017b18ad>:3: FutureWarning: The default value of
numeric_only in DataFrame.quantile is deprecated. In a future version,
it will default to False. Select only valid columns or specify the
value of numeric_only to silence this warning.
  Q3 = n_df.quantile(0.75)
<ipython-input-54-23f3017b18ad>:7: FutureWarning: Automatic reindexing
on DataFrame vs Series comparisons is deprecated and will raise
ValueError in a future version. Do `left, right = left.align(right,
axis=1, copy=False)` before e.g. `left == right`
  outliers = ((n_df < lower) | (n_df > upper)).sum().sum()
<ipython-input-54-23f3017b18ad>:7: FutureWarning: Automatic reindexing
on DataFrame vs Series comparisons is deprecated and will raise
ValueError in a future version. Do `left, right = left.align(right,
axis=1, copy=False)` before e.g. `left == right`
  outliers = ((n_df < lower) | (n_df > upper)).sum().sum()

2

n_df.boxplot()
plt.title('Outliers')
plt.show()
```


Outliers
3
2
1
0
−1
Area
Circularity
Height
Perimeter
Width
X
Y