

# Pregunta 6

November 21, 2019

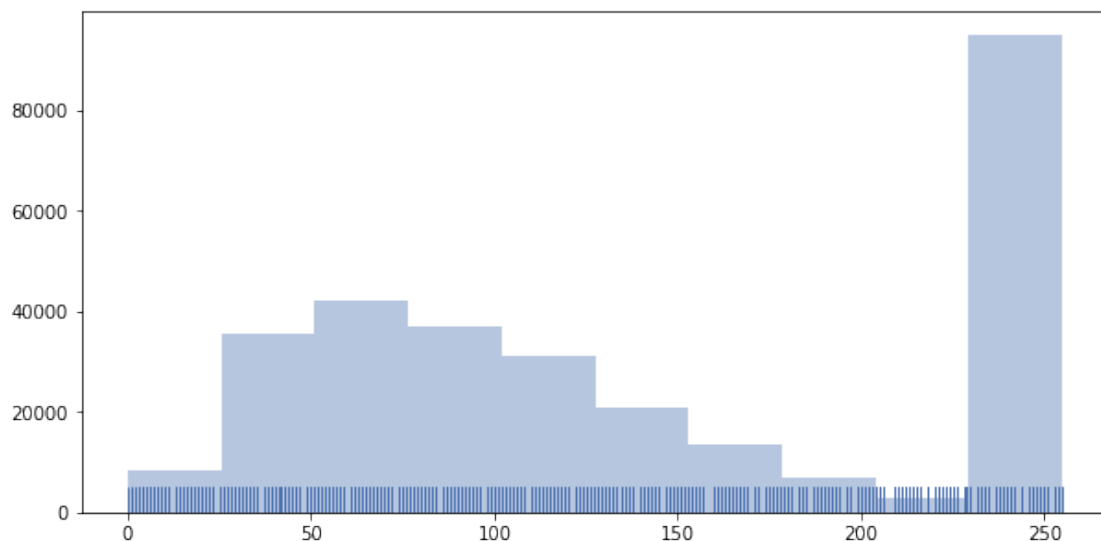
## 1 Pregunta 6

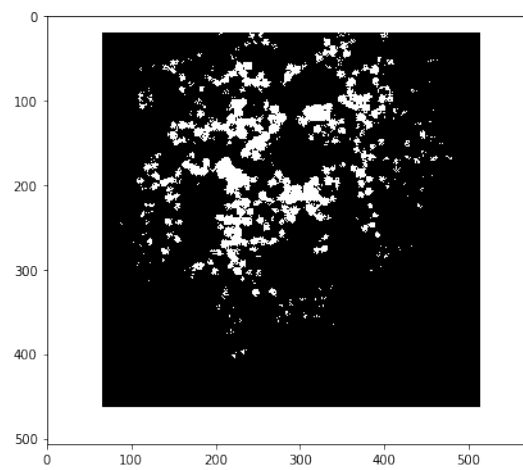
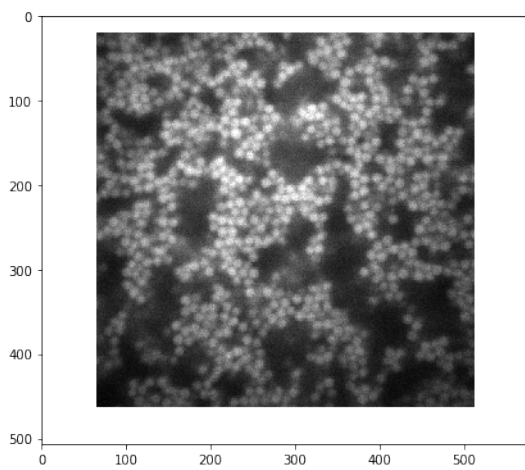
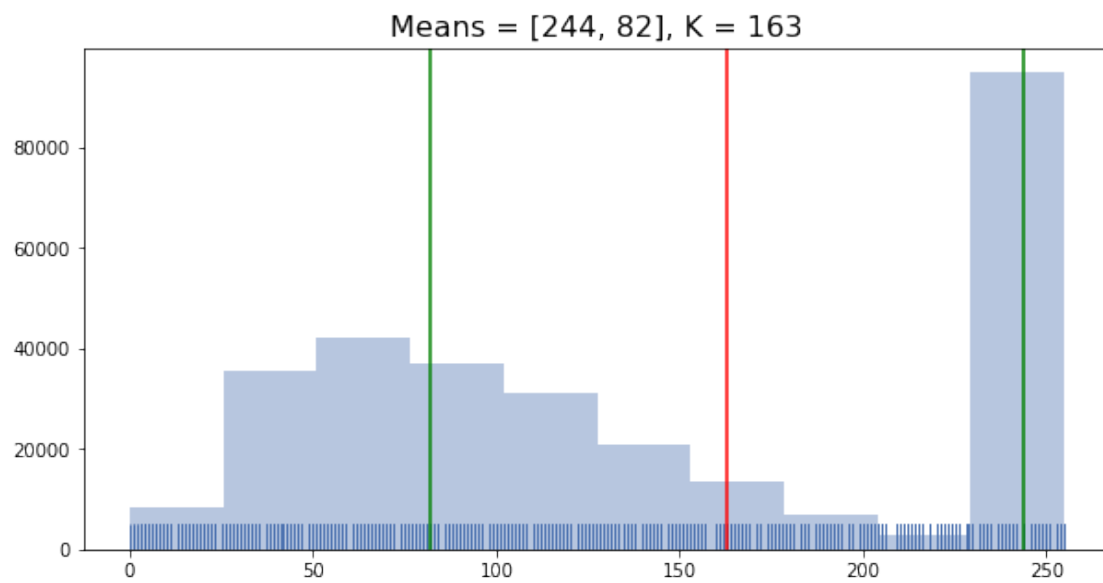
(2 puntos) La imagen Ex3Preg6(a).tif muestra una imagen tomada con un microscopio de cultivo de bacterias identificadas por los círculos intensos:

- (0.5 puntos) Usando una técnica de umbralización global, segmente la imagen y muestre el resultado de la segmentación.
- (0.5 puntos) A la imagen original se le aplicó una umbralización con valores locales y al resultado se le realizó una apertura morfológica obteniendo la imagen Ex3Preg6(b).tif. Usando esta imagen, cuente y etiquete cuantos objetos de la segmentación pueden considerarse células independientes.
- (1 punto) Continuando con la imagen anterior. Cuente y etiquete cuantos objetos de la segmentación pueden considerarse 2 células agrupadas, y cuantos y cuales más de 2 células.

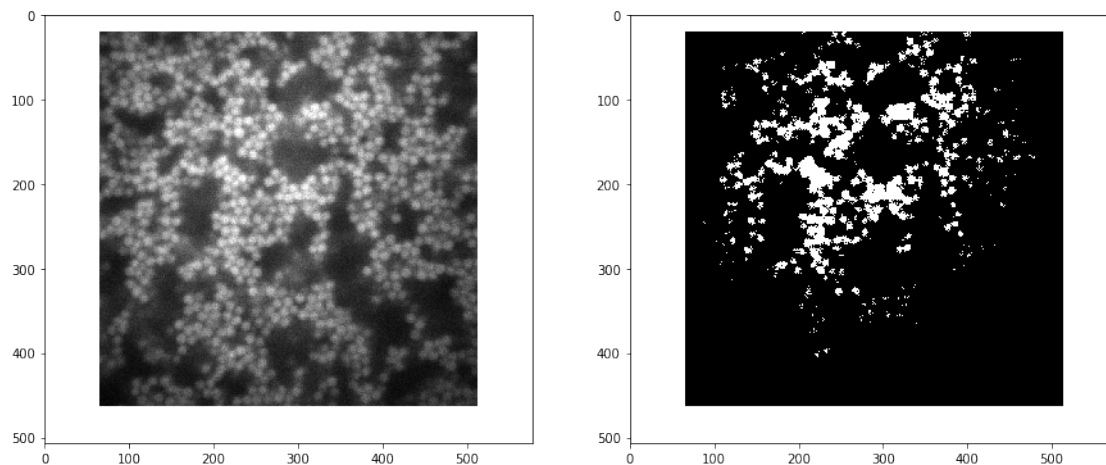
**1.1 a. (0.5 puntos) Usando una técnica de umbralización global, segmente la imagen y muestre el resultado de la segmentación.**

[10]: <matplotlib.axes.\_subplots.AxesSubplot at 0x1c1f08cb50>

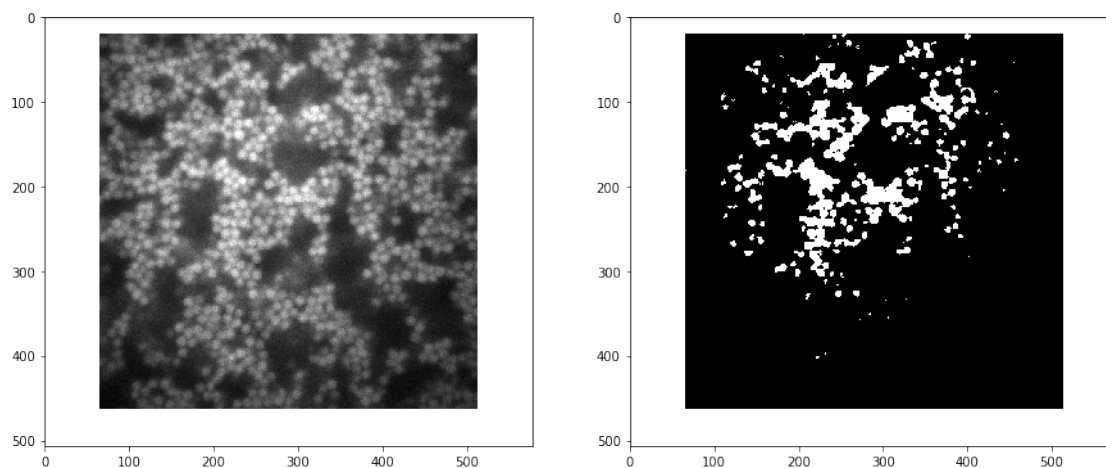




Como podemos ver, una técnica de umbralización estándar como k-medias móviles, con dos medias, da resultados muy pobres.

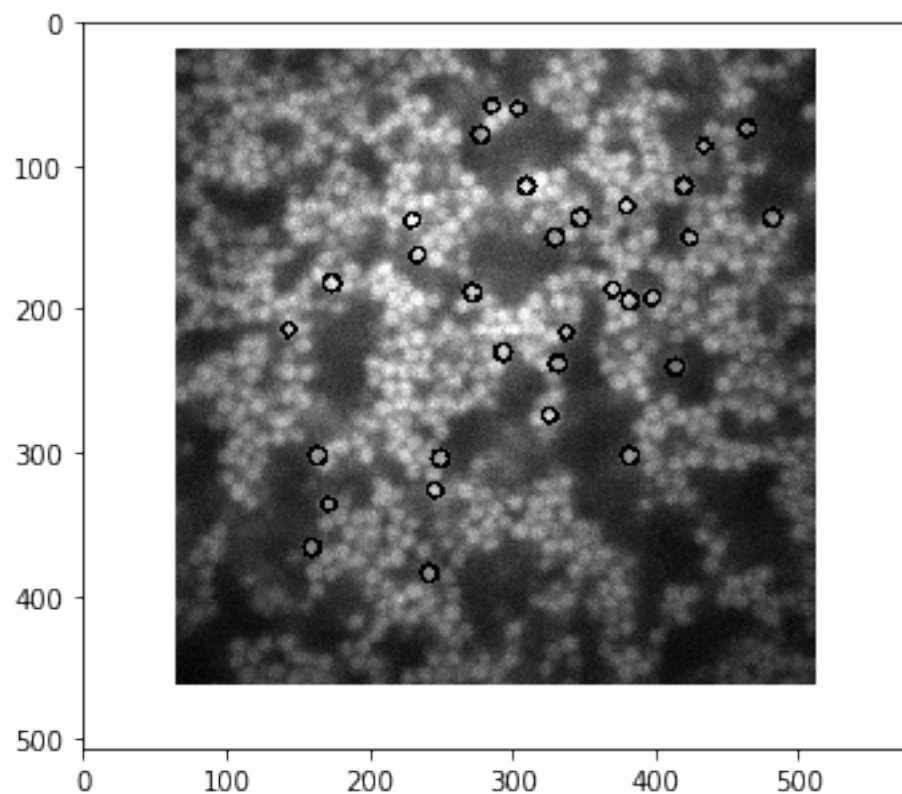


El algoritmo de Otsu no logra mejorar mucho la segmentación (esto era de esperarse dado que el histograma original era claramente bimodal).



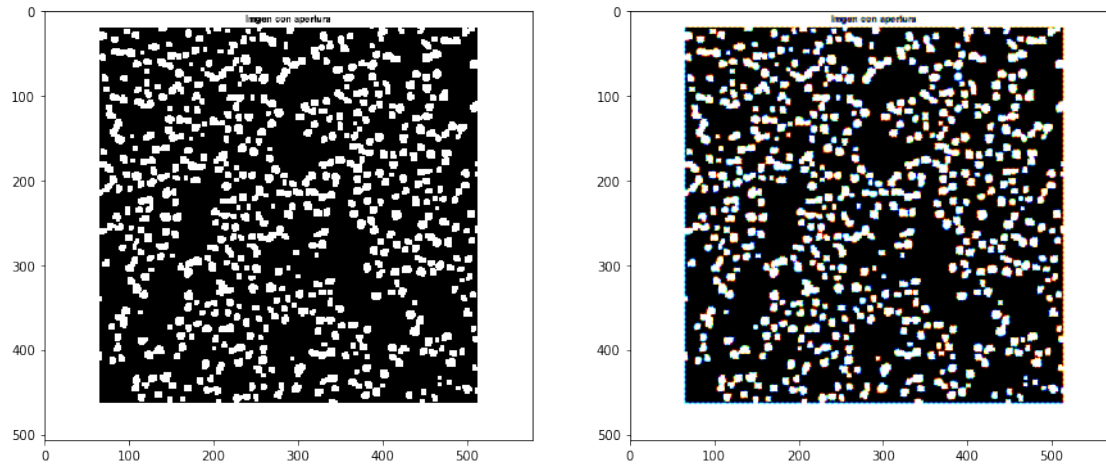
El algoritmo de Otsu no logra mejorar mucho la segmentación aún en combinación con un suavizado Gaussiano.

[22]: <matplotlib.image.AxesImage at 0x1c28687290>



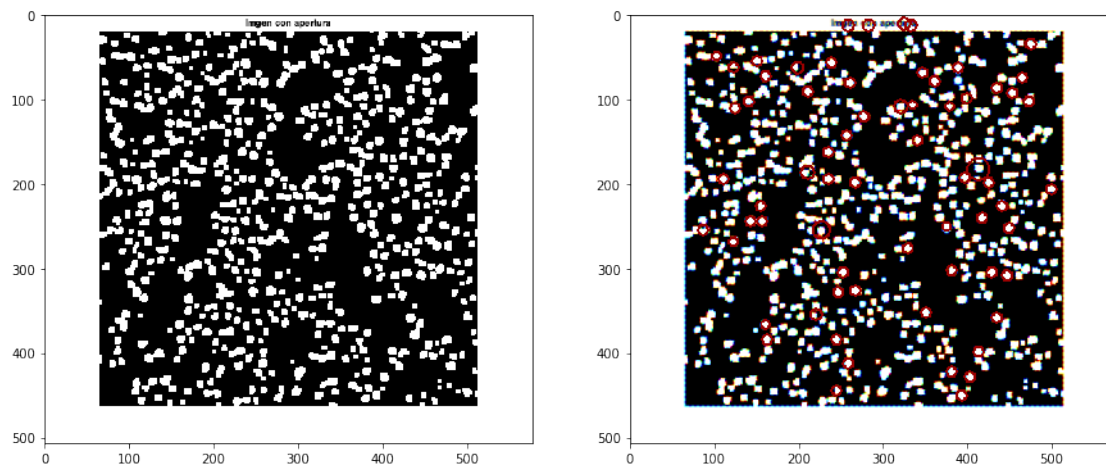
Al ver que las células tenían una apariencia más o menos circular, parecía una buena idea usar una transformada de Hough para buscar los círculos de la imagen, pero esto entregó resultados muy pobres. Tal vez esto podría funcionar con la imagen binaria.

- 1.2 b. (0.5 puntos) A la imagen original se le aplicó una umbralización con valores locales y al resultado se le realizó una apertura morfológica obteniendo la imagen Ex3Preg6(b).tif. Usando esta imagen, cuente y etiquete cuantos objetos de la segmentación pueden considerarse células independientes.



### 1.2.1 Primera aproximación :

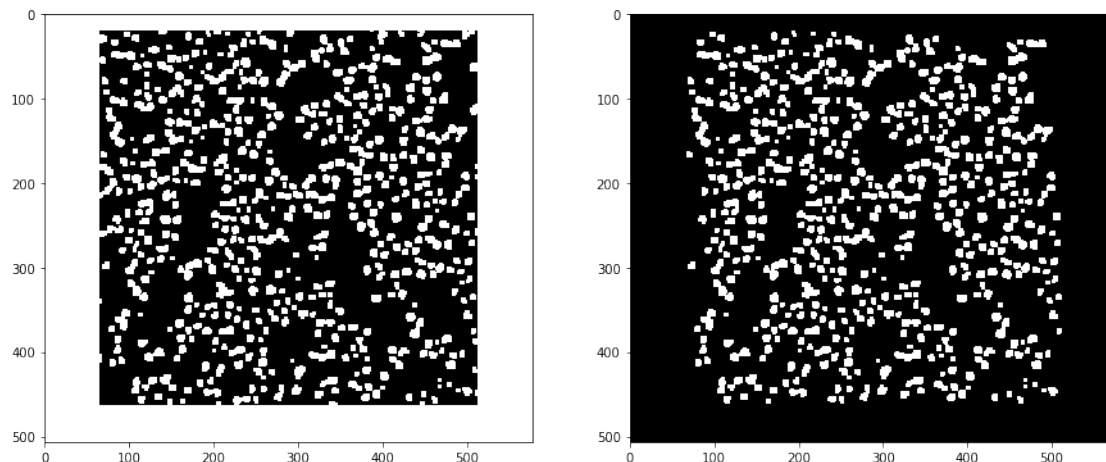
El uso de la transformada de Hough para círculos, no para líneas. Al ver la imagen, uno podría pensar que un círculo es una buena aproximación de la forma de una célula, por lo tanto los círculos encontrados por una transformada de Hough serían las células que buscamos identificar, caracterizar y contabilizar



Aquí podemos ver que aunque la imagen principal carece de falsos positivos (es decir todos los círculos dibujados dentro de la región útil de la imagen contienen una célula) el **número de falsos**

**negativos es altísimo** : sólo una pequeña parte de las células observadas fueron identificadas por `cv.HoughCircles()`.

Esto nos indica que tal vez las células no se asemejan tanto a un círculo. Por esta razón, no se explorará más a fondo esta vía de acción. Cabe mencionar que la transformada encuentra círculos en el texto de encabezado : **Imagen con apertura**. Por esta razón, en adelante se trabajará con otra imagen recortada a mano para excluir este texto que podría causar problemas en la segmentación más adelante.



[30]: <matplotlib.axes.\_subplots.AxesSubplot at 0x1c27389190>

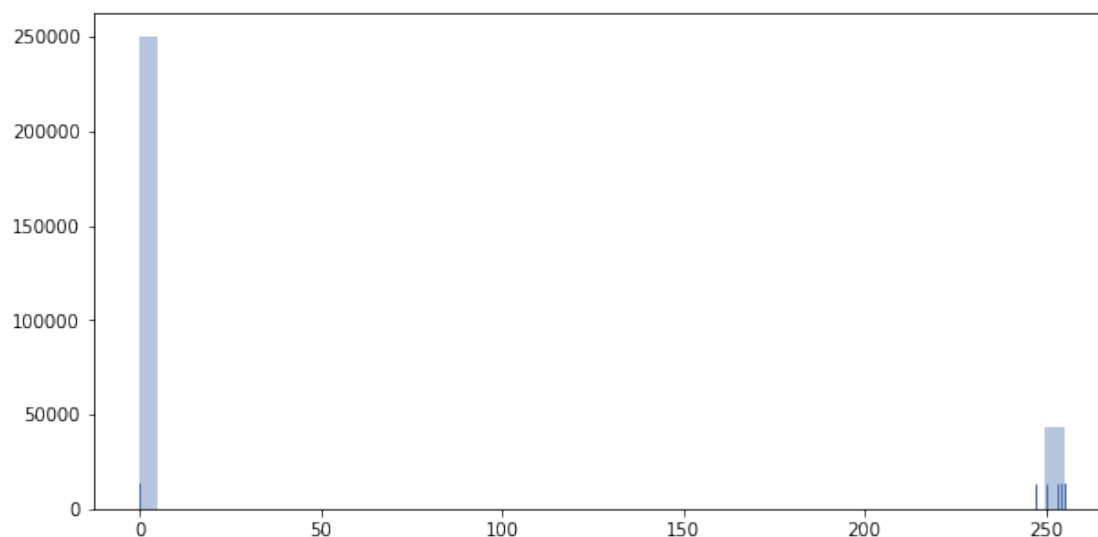
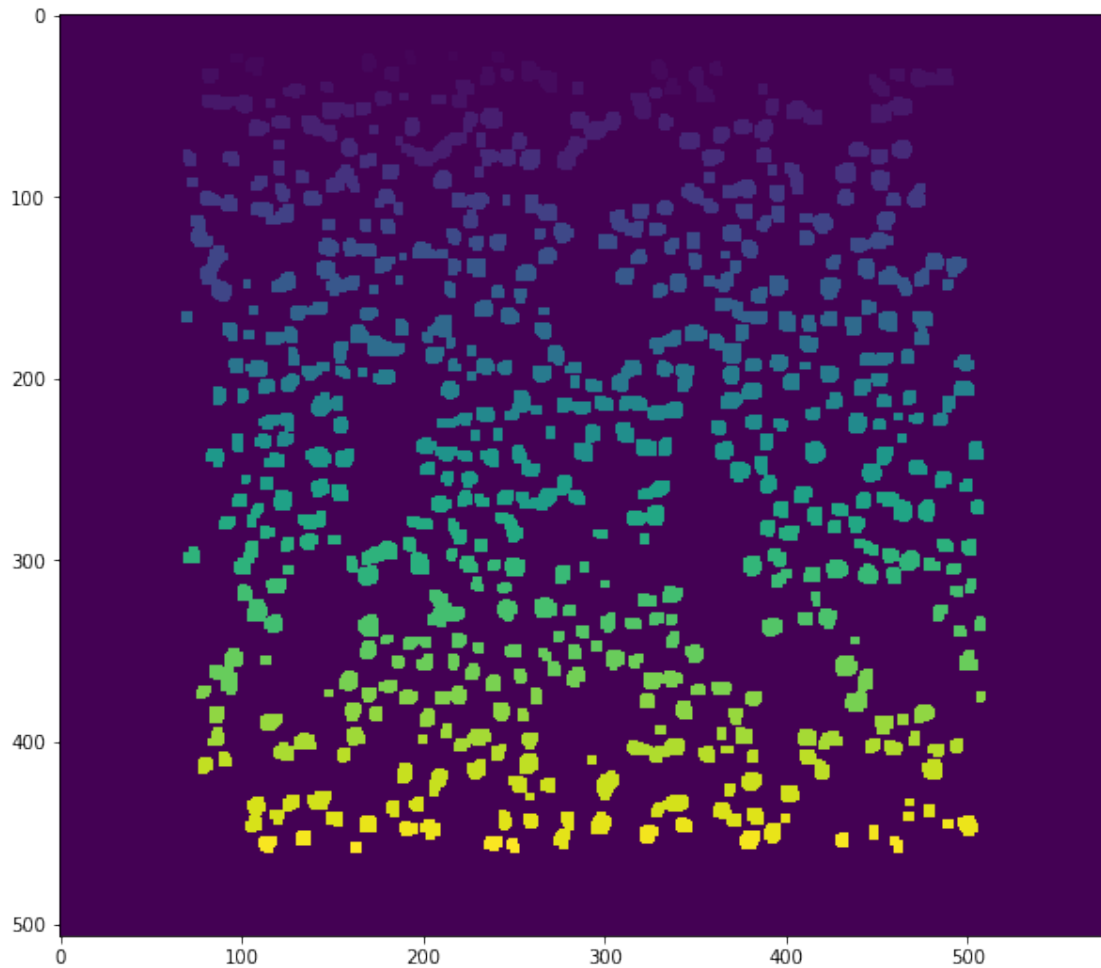


Imagen claramente binaria.

474



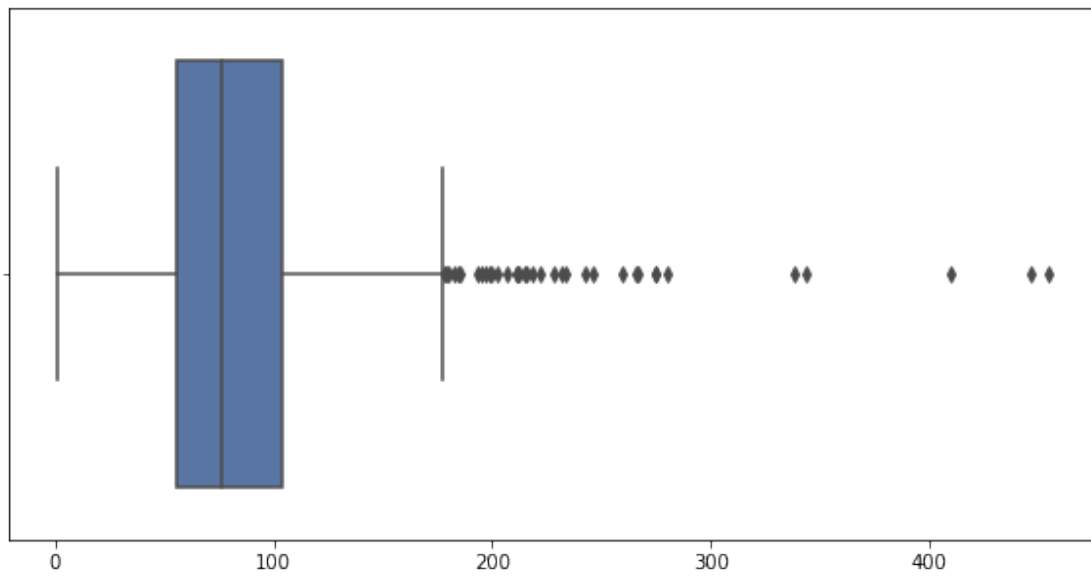
Aquí podemos ver cómo `skimage.label()` es efectivo para identificar objetos en una imagen binaria.

Bajo las hipótesis : 1. El área observable de las bacterias sigue una distribución normal razonablemente estrecha. 2. El área de la máscara de segmentación generada a través de una umbralización con valores locales y una apertura morfológica es una buena aproximación del área de una célula.

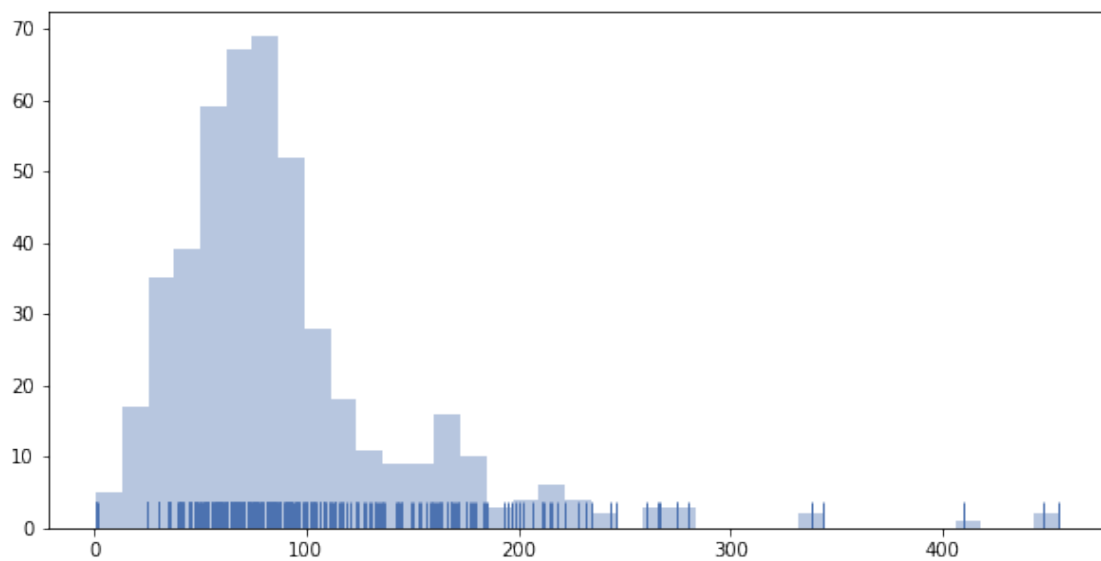
La conclusión lógica sería que donde se tenían dos o más células y que la apertura unió las regiones de segmentación, el valor del área aumentará consecuentemente.

```
[33]: (
      area
count  474.000000
mean    90.411392
std     59.468520
min      1.000000
25%     55.250000
50%     76.000000
75%    104.000000
```

```
max      455.000000, <matplotlib.axes._subplots.AxesSubplot at 0x1c32f91f10>)
```



```
[34]: <matplotlib.axes._subplots.AxesSubplot at 0x1c330202d0>
```



Analizando la distribución de las áreas, se encuentran algunos puntos cercanos a cero, éstos serán inspeccionados a continuación.

```
[35]: (0      1
      2      1
```



```

3      1
4      1
1      2
241    25
226    25
29     25
199    25
33     25
Name: area, dtype: int64, 362      266
166    267
416    275
329    275
219    280
306    338
263    344
39     410
98     447
40     455
Name: area, dtype: int64)

```

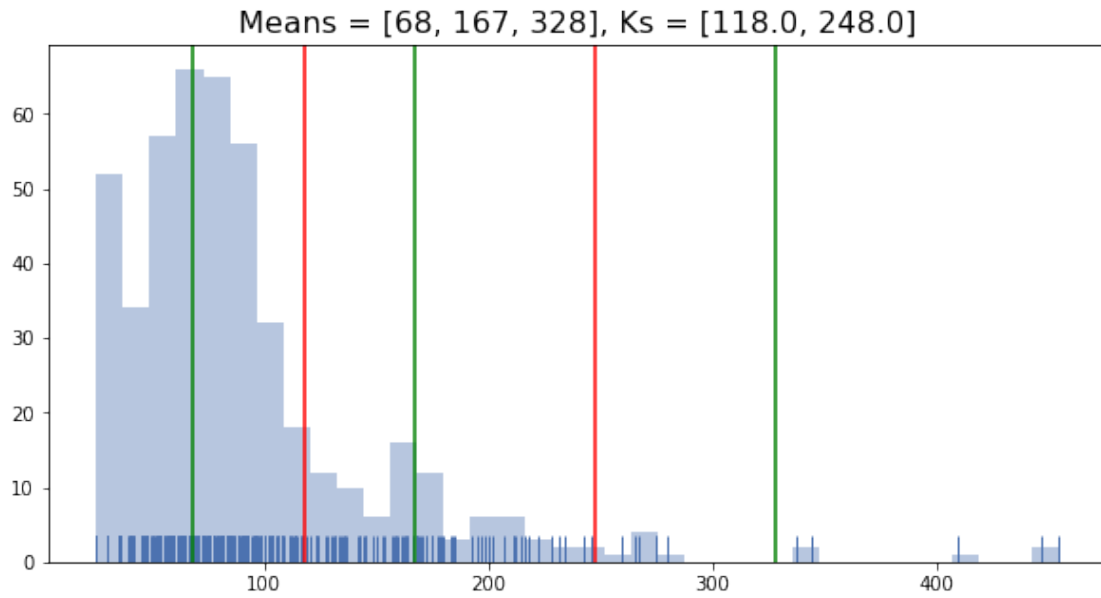
Algunos de los primeros valores de las áreas corresponden claramente a falsos positivos, porque no tenemos células de uno o dos pixeles. Estos pixeles blancos no se observaban en la imagen original, de cualquier manera se retirarán manualmente para no sesgar el análisis posterior.

Se propone clasificar los cúmulos de 1, 2 y 3 o más bacterias en función del área. Se utilizará el algoritmo de las medias móviles con tres grupos, es decir dos umbrales.

```

      means      k
0  68.559367    NaN
1 167.696203 118.127785
2 328.818182 248.257192
      means      k
0     68     NaN
1    167    118.0
2    328    248.0

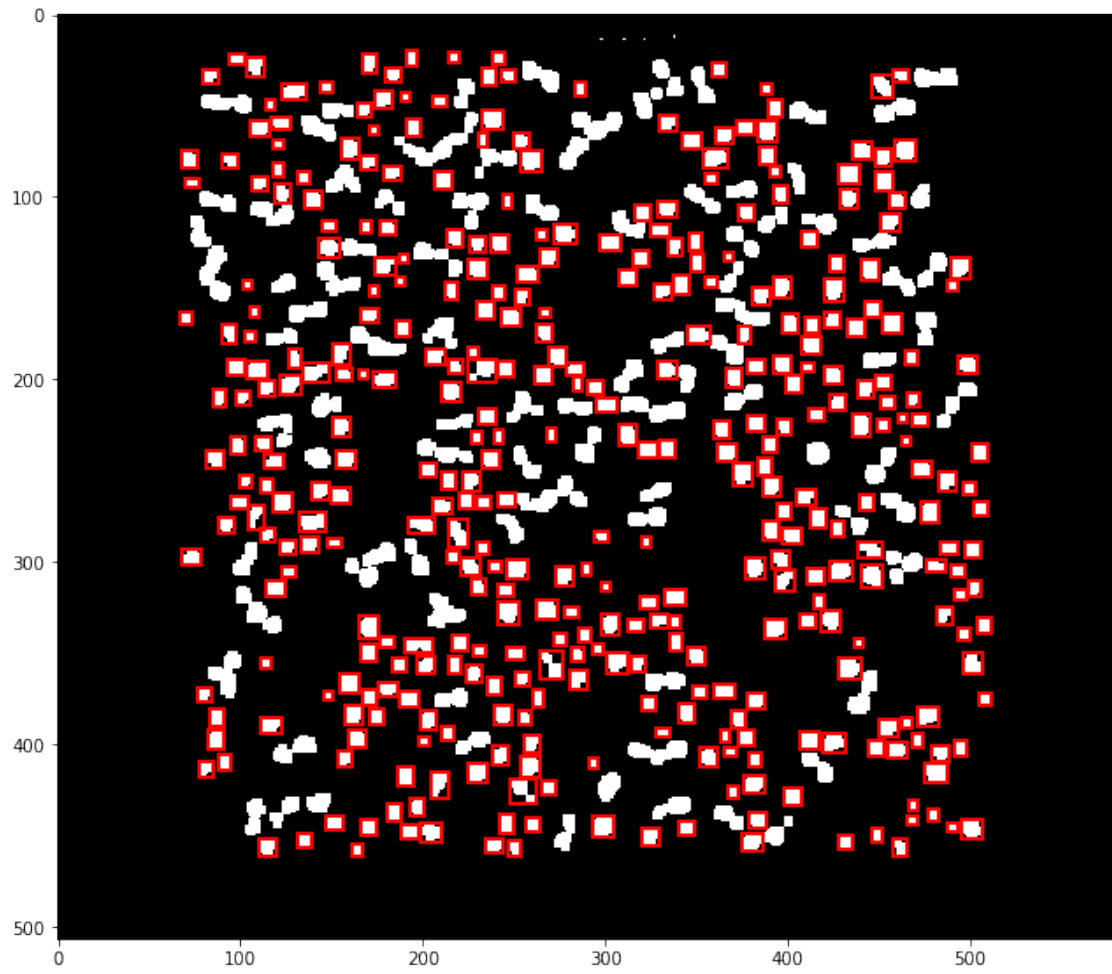
```

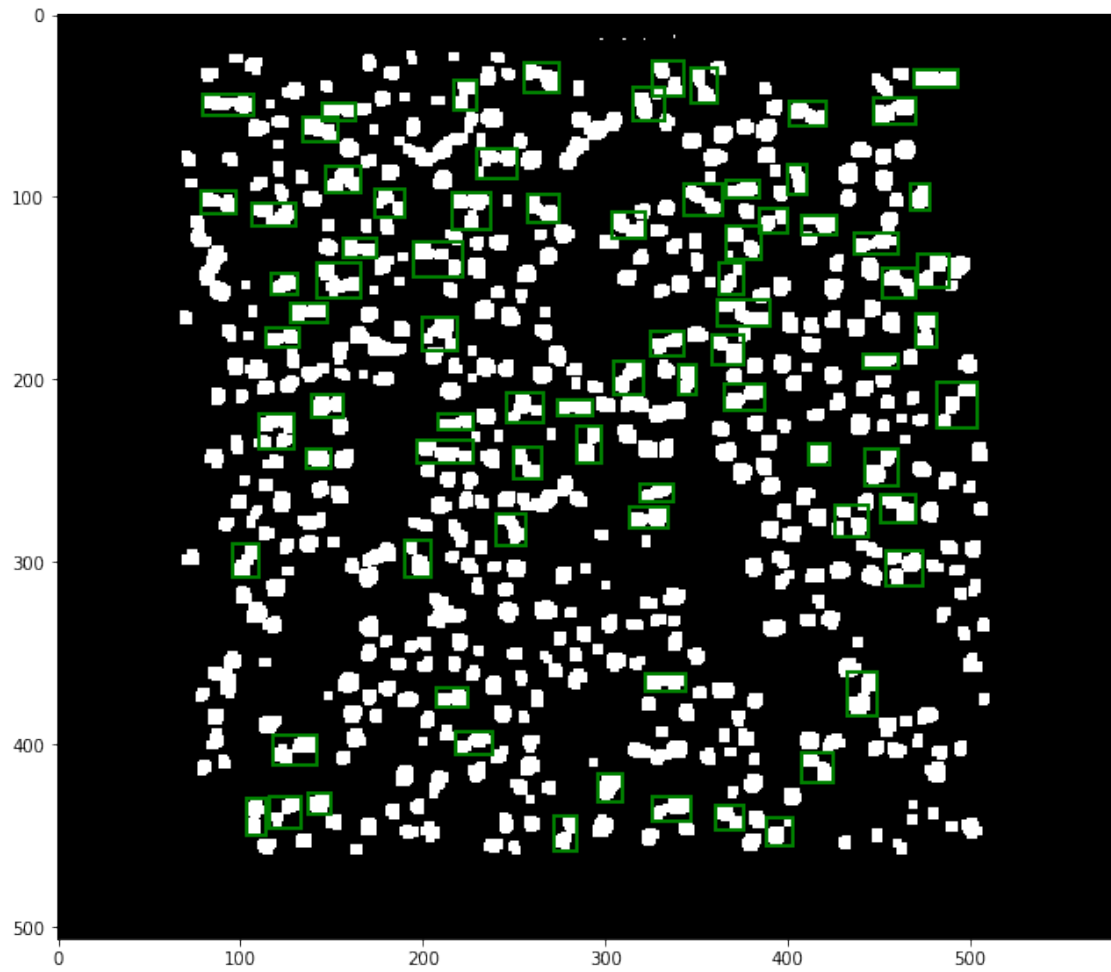


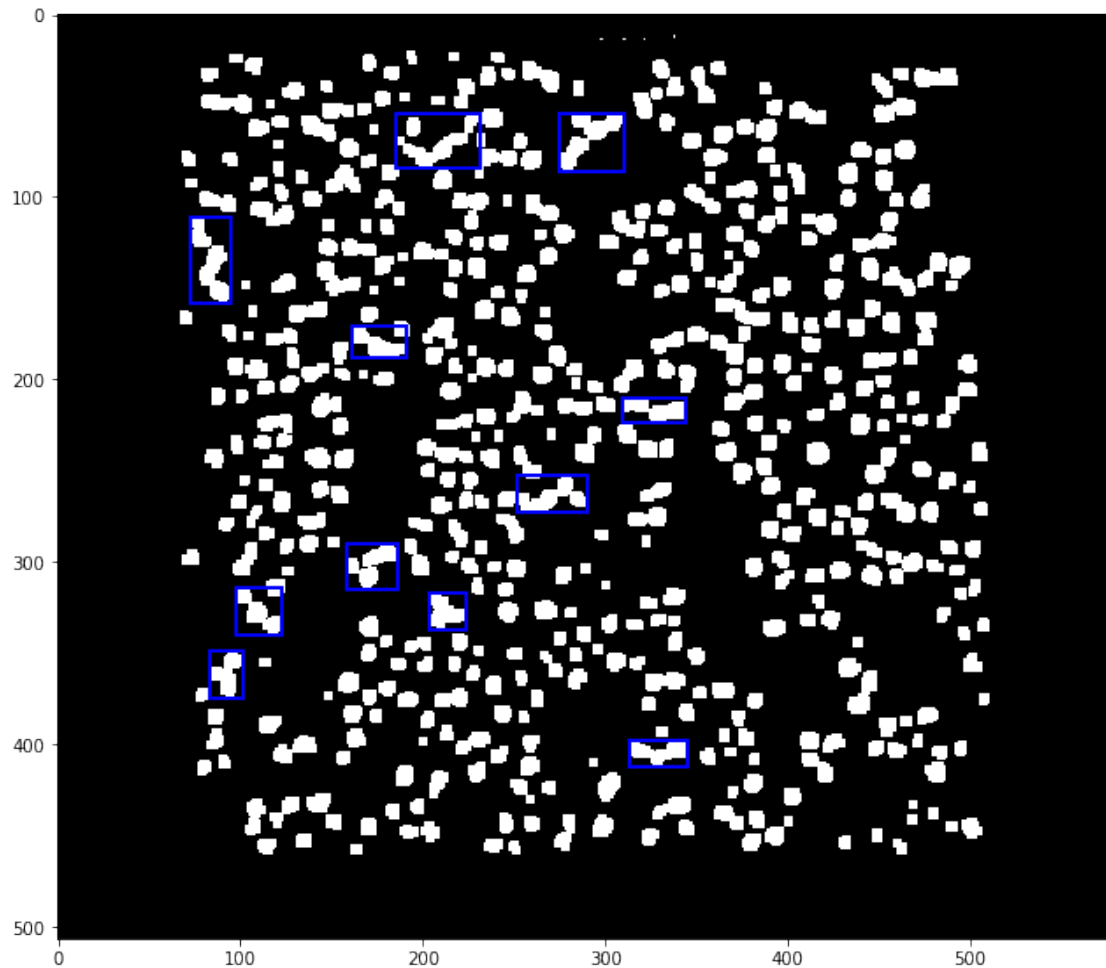
Creamos una lista de listas en la cual están contenidos tres grupos de bacterias (cúmulos de 1, 2 o más) identificados por `skimage.label()`, separados en función de los umbrales encontrados promediando los promedios encontrados por `sklearn.KMeans()`.

A continuación se observan las máscaras identificando los grupos respectivos. Nótese que el conjunto con la mayor cantidad de errores (sobre todo falsos positivos) es el de dos células.

Grupos observados : 1. En rojo : una célula. 2. En verde : dos células. 3. En azul : más de dos células.







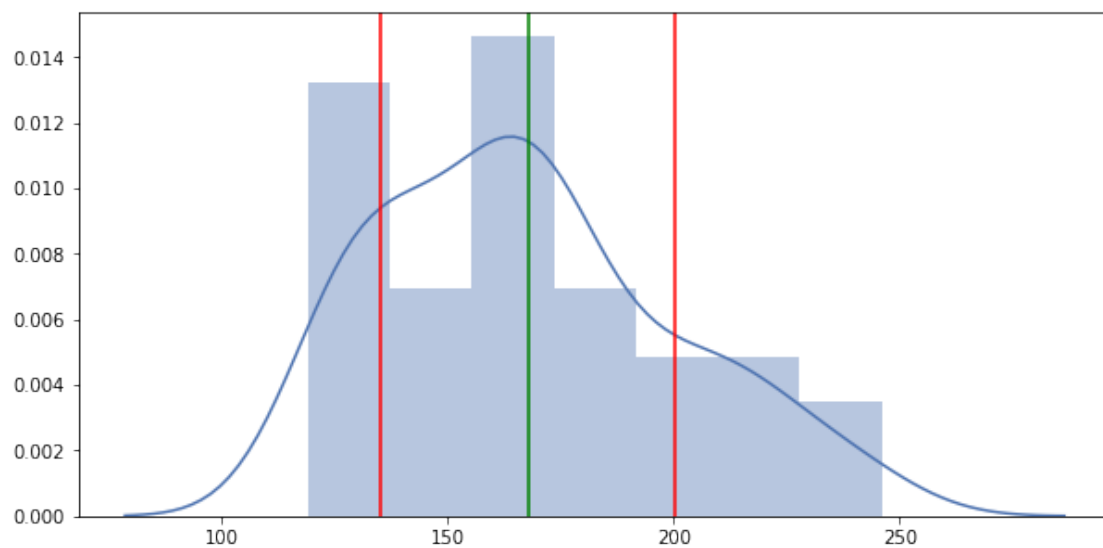
```
[41]: {'Objetos de 1 células': 379,
      'Objetos de 2 células': 79,
      'Objetos de 3 células': 11}
```

Aquí está el conteo del número de bacterias por objeto en la máscara de segmentación (imagen binaria). De entre todas las clases, debemos confiar menos en la segunda categoría, dado que la mera separación en función del área no basta para distinguir adecuadamente entre cúmulos de una célula grande, dos y tres de las cuales una o dos pueden ser pequeñas.

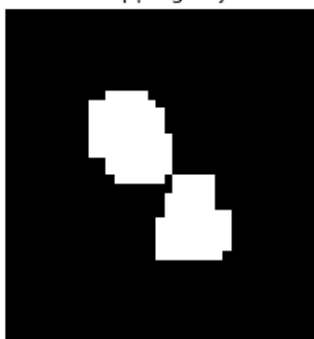
Reporte de errores observados en el conjunto de **dos células**.

1. Falsos positivos : 10
2. Falsos negativos : 1

```
[47]: <matplotlib.lines.Line2D at 0x1c32448f10>
```



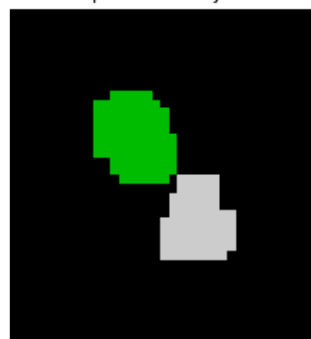
Overlapping objects



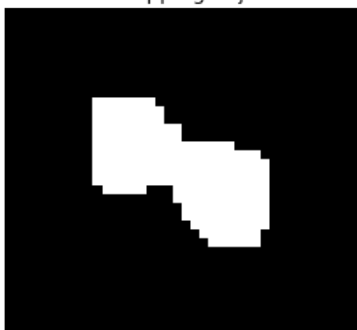
Distances



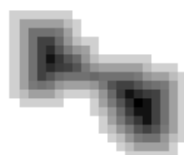
Separated objects



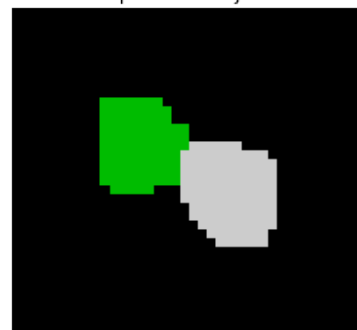
Overlapping objects



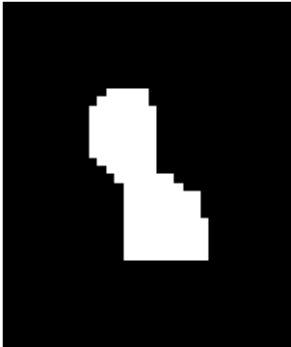
Distances



Separated objects



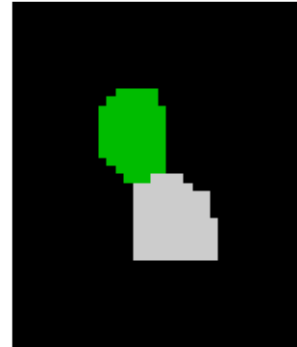
Overlapping objects



Distances



Separated objects



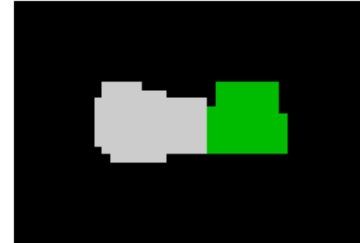
Overlapping objects



Distances



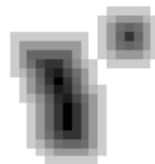
Separated objects



Overlapping objects



Distances



Separated objects



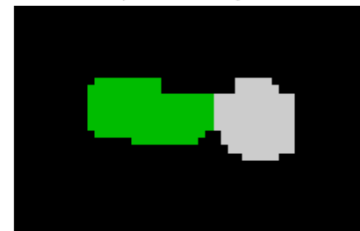
Overlapping objects



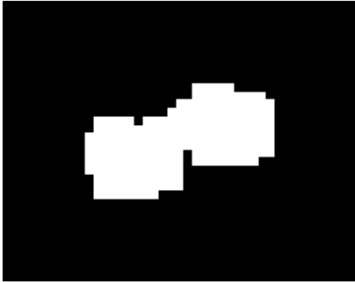
Distances



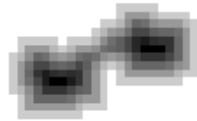
Separated objects



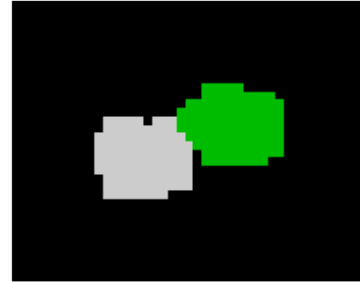
Overlapping objects



Distances



Separated objects



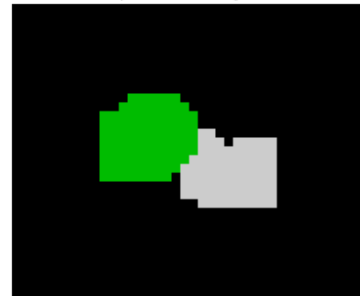
Overlapping objects



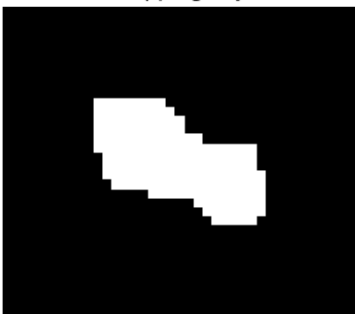
Distances



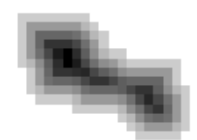
Separated objects



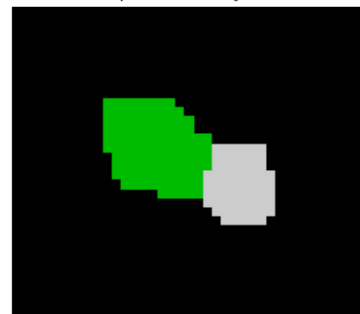
Overlapping objects



Distances

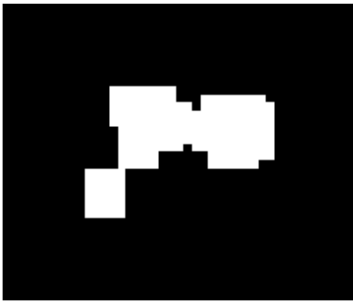


Separated objects

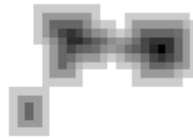




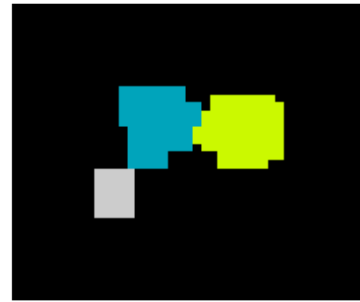
Overlapping objects



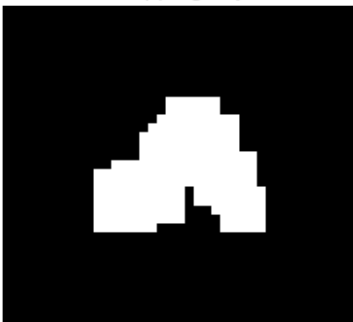
Distances



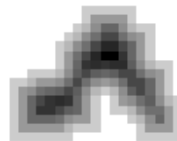
Separated objects



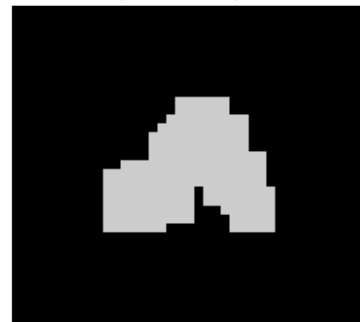
Overlapping objects



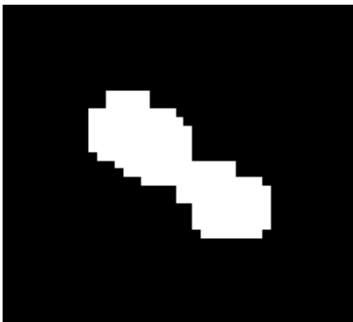
Distances



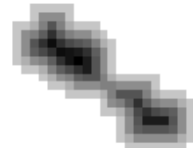
Separated objects



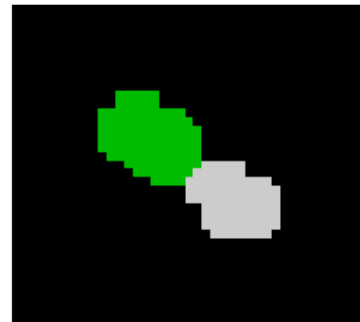
Overlapping objects



Distances



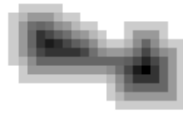
Separated objects



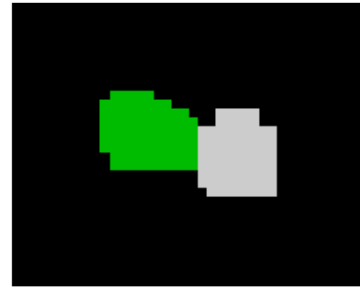
Overlapping objects



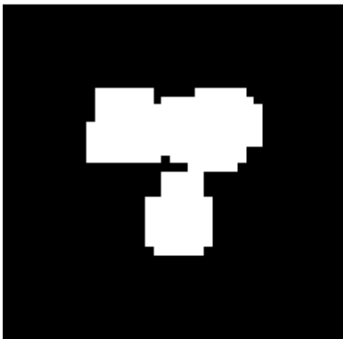
Distances



Separated objects



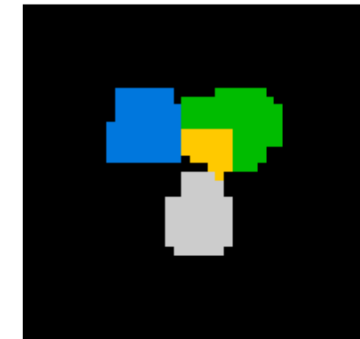
Overlapping objects



Distances



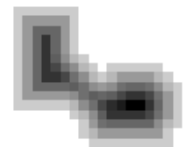
Separated objects



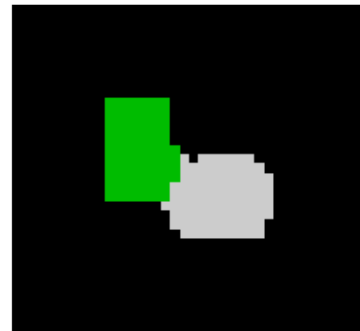
Overlapping objects



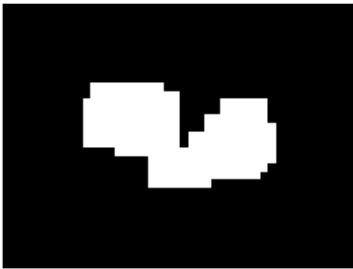
Distances



Separated objects



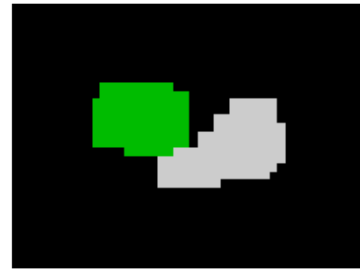
Overlapping objects



Distances



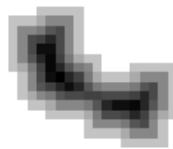
Separated objects



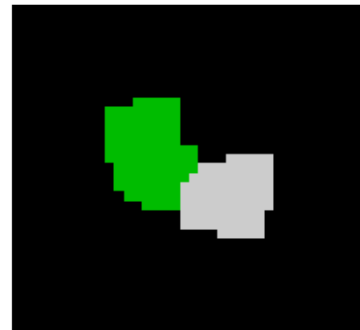
Overlapping objects



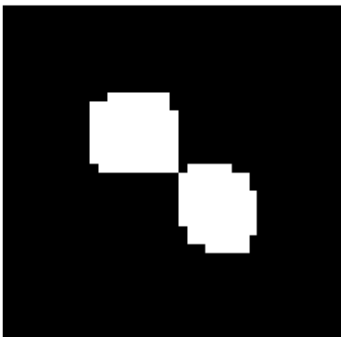
Distances



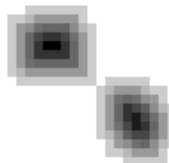
Separated objects



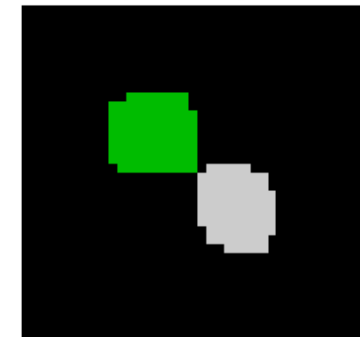
Overlapping objects



Distances



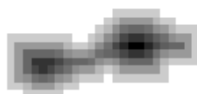
Separated objects



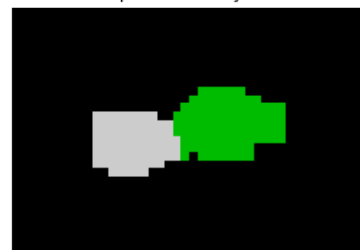
Overlapping objects



Distances



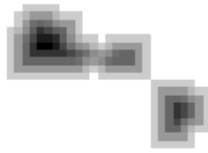
Separated objects



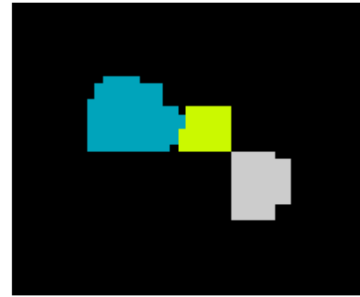
Overlapping objects



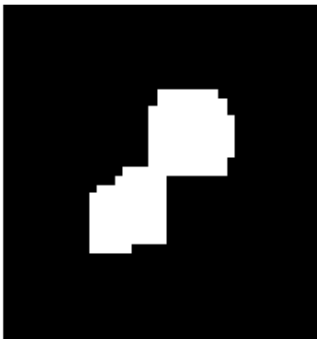
Distances



Separated objects



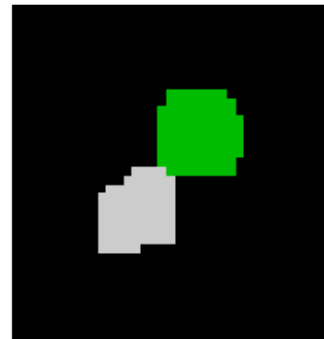
Overlapping objects



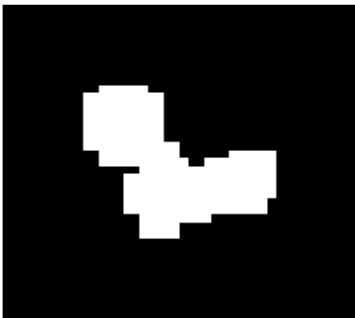
Distances



Separated objects



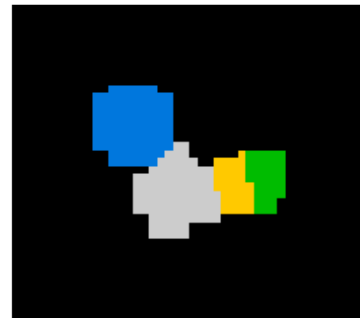
Overlapping objects

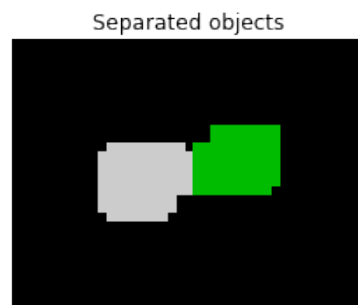
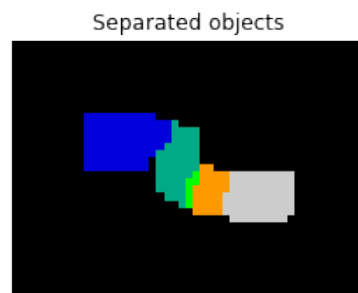
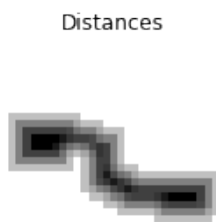
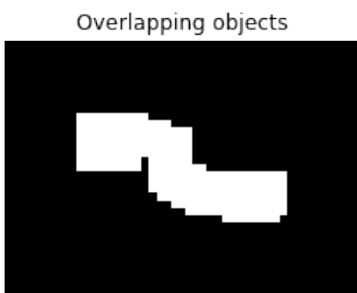
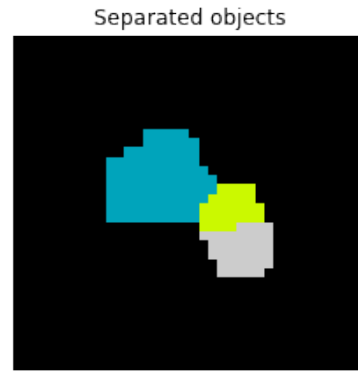
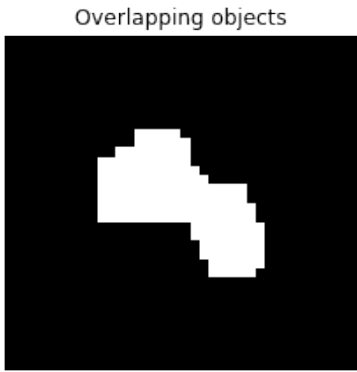


Distances

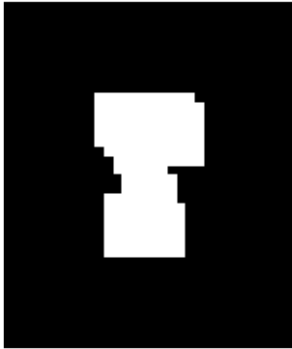


Separated objects





Overlapping objects



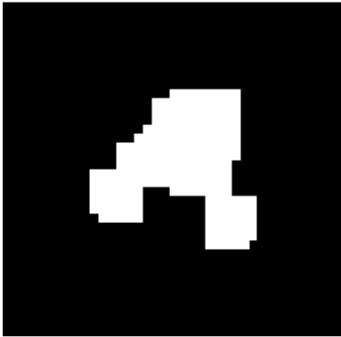
Distances



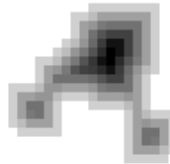
Separated objects



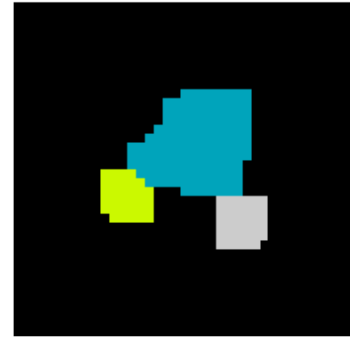
Overlapping objects



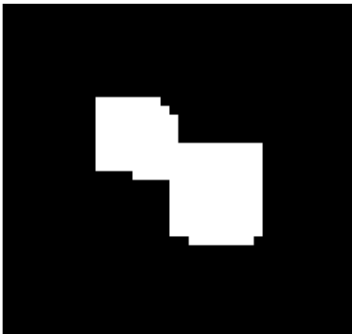
Distances



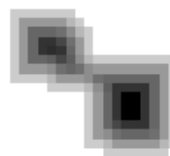
Separated objects



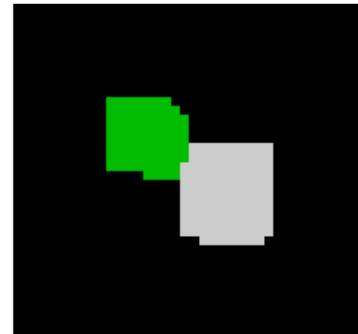
Overlapping objects



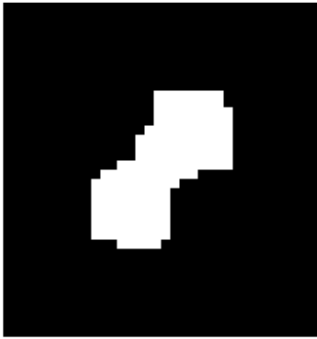
Distances



Separated objects



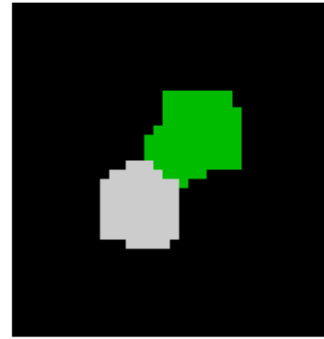
Overlapping objects



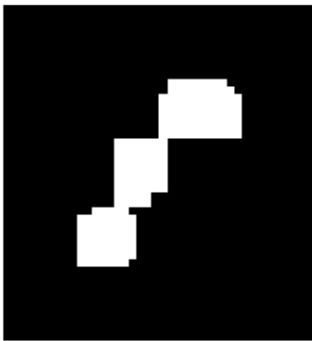
Distances



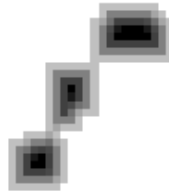
Separated objects



Overlapping objects



Distances



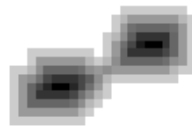
Separated objects



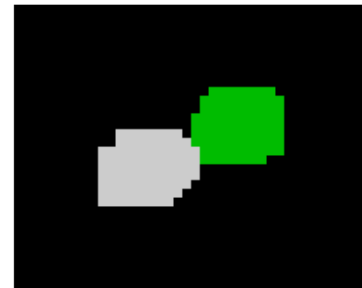
Overlapping objects



Distances



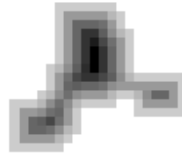
Separated objects



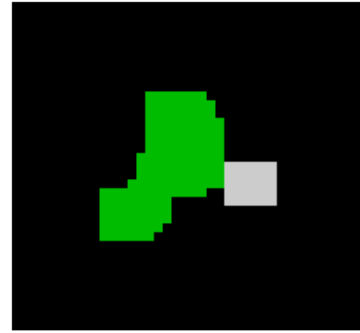
Overlapping objects



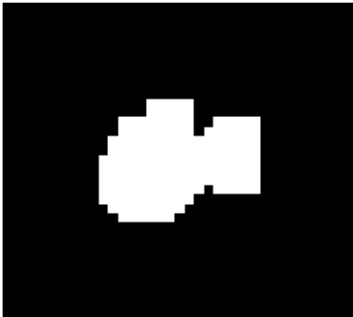
Distances



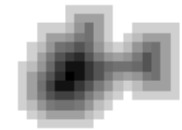
Separated objects



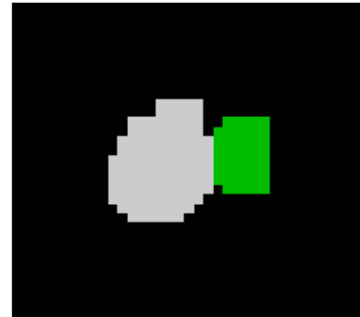
Overlapping objects



Distances



Separated objects



Overlapping objects



Distances

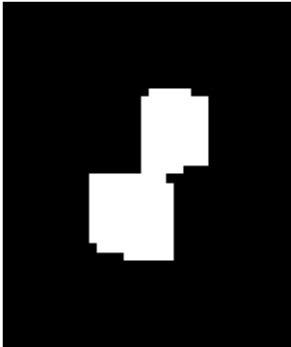


Separated objects





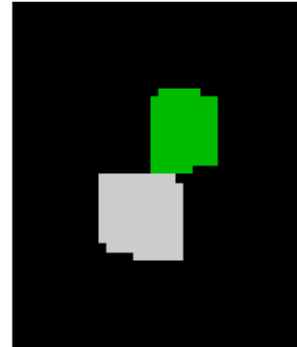
Overlapping objects



Distances



Separated objects



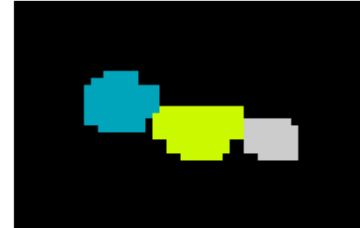
Overlapping objects



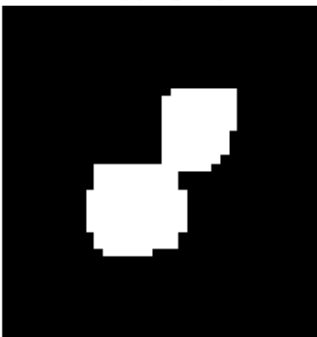
Distances



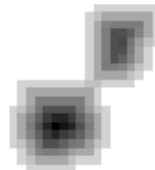
Separated objects



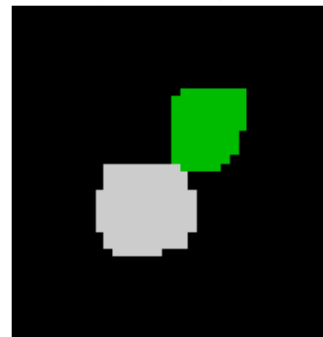
Overlapping objects



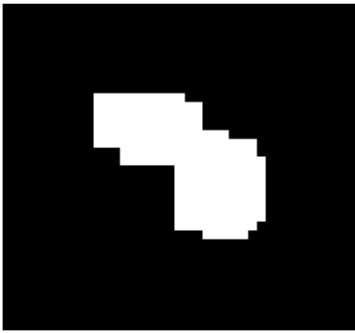
Distances



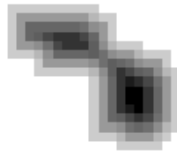
Separated objects



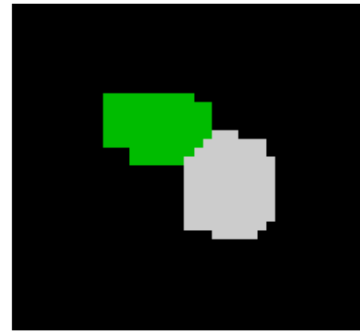
Overlapping objects



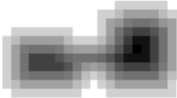
Distances



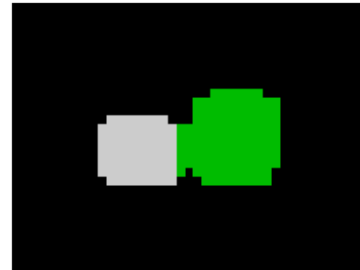
Separated objects



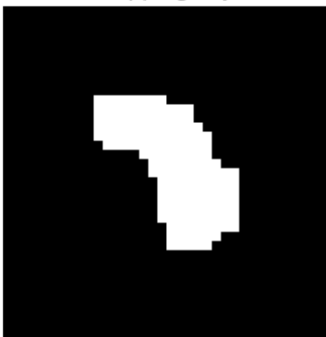
Distances



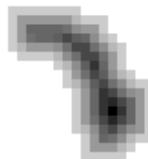
Separated objects



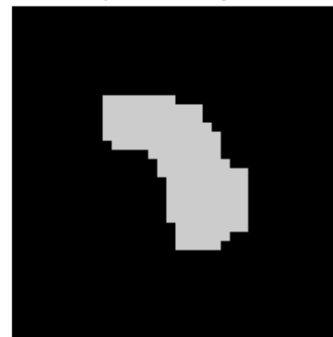
Overlapping objects



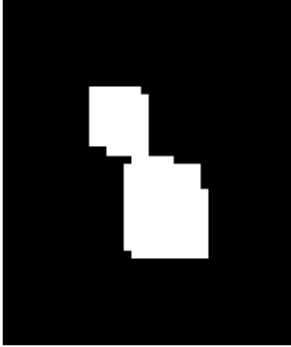
Distances



Separated objects



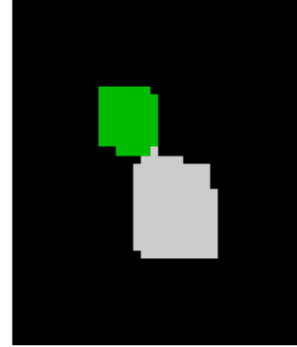
Overlapping objects



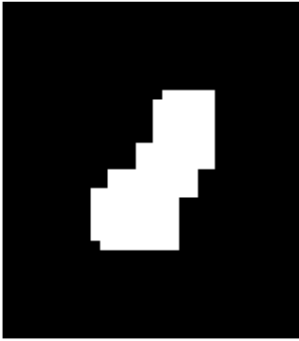
Distances



Separated objects



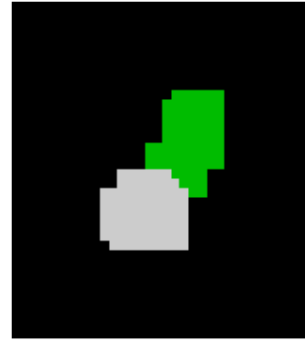
Overlapping objects



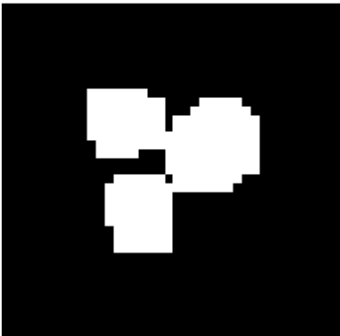
Distances



Separated objects



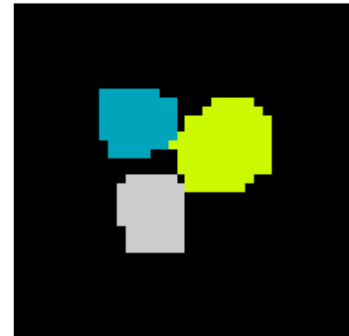
Overlapping objects



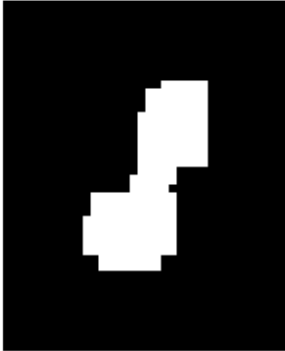
Distances



Separated objects



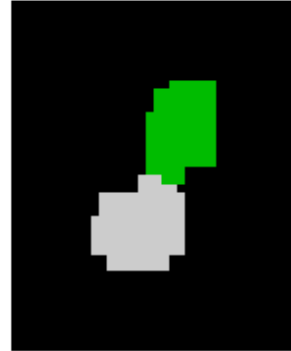
Overlapping objects



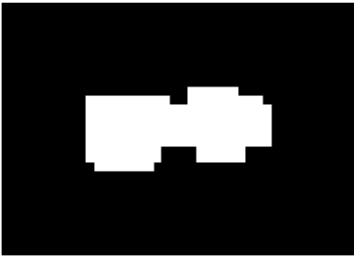
Distances



Separated objects



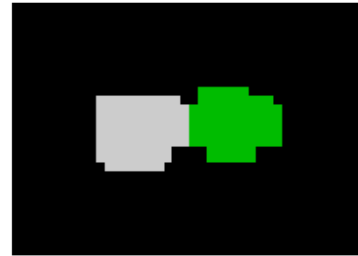
Overlapping objects



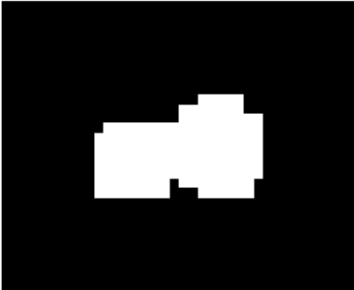
Distances



Separated objects



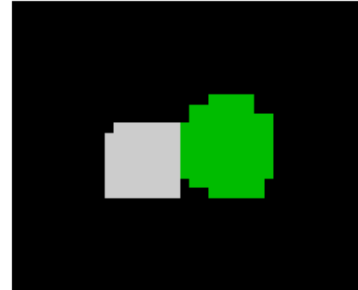
Overlapping objects



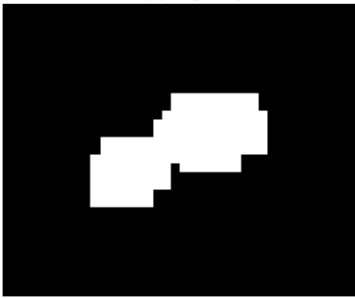
Distances



Separated objects



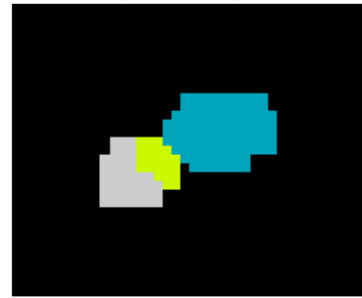
Overlapping objects



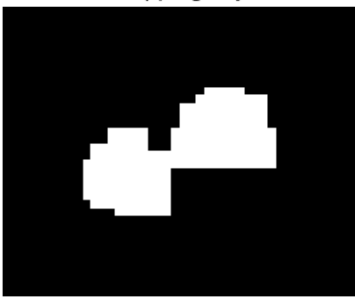
Distances



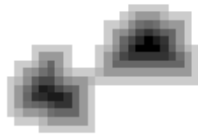
Separated objects



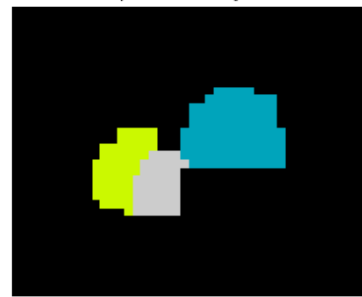
Overlapping objects



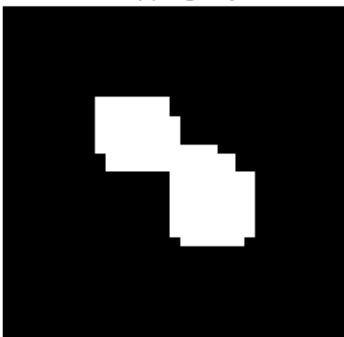
Distances



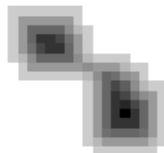
Separated objects



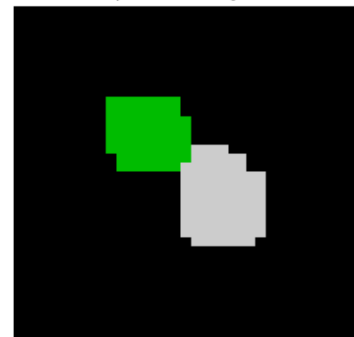
Overlapping objects

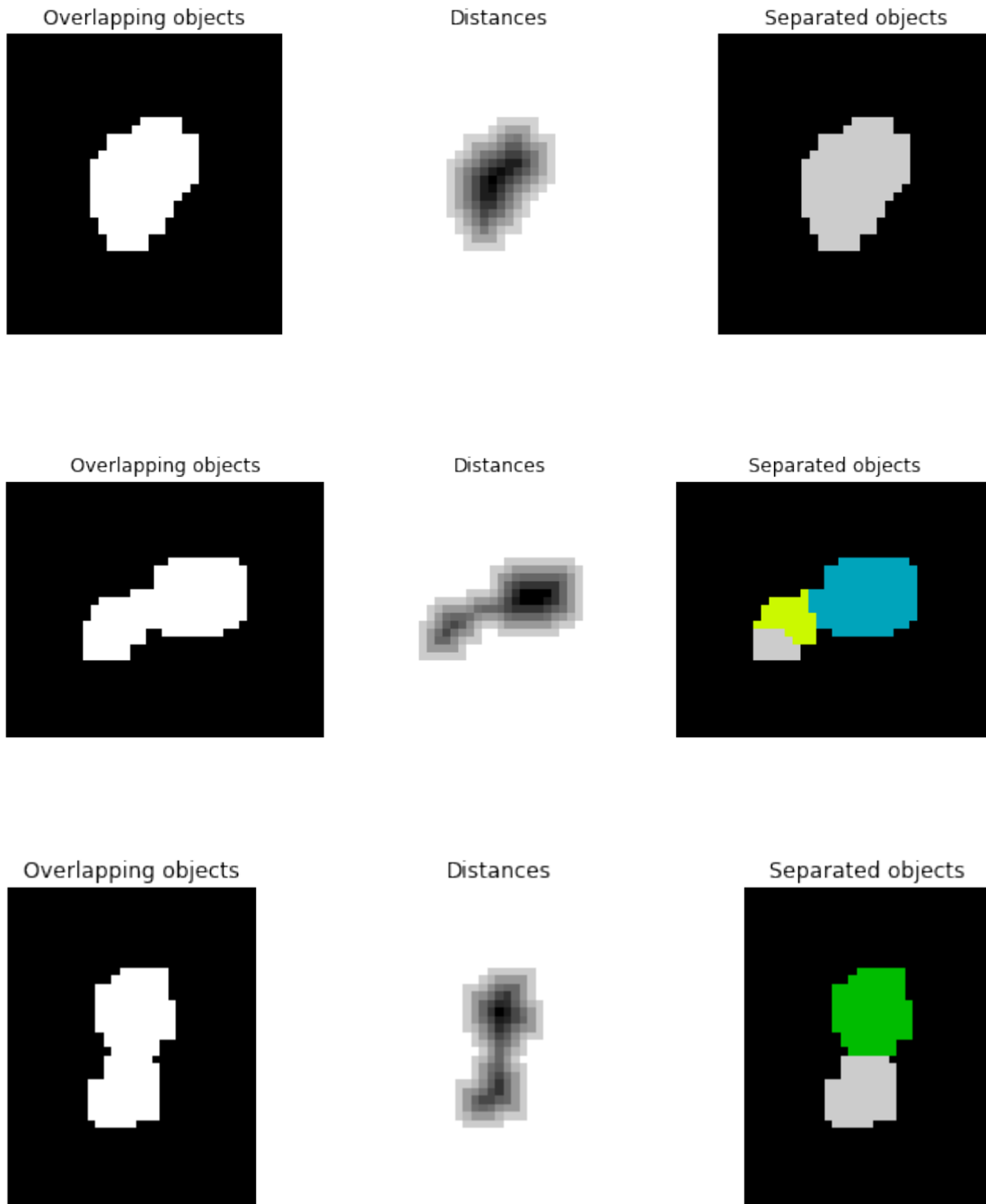


Distances

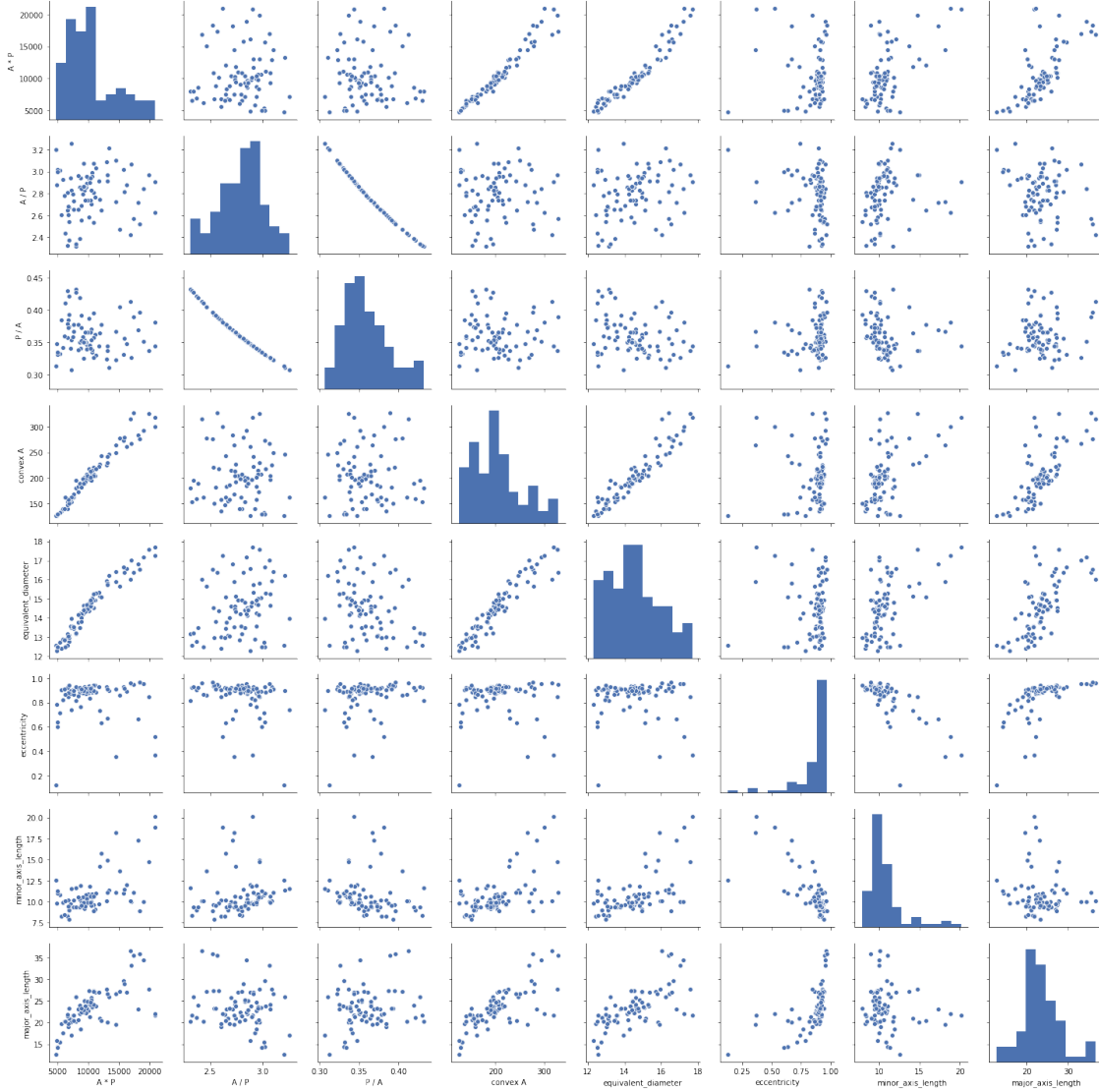


Separated objects





[78]: <seaborn.axisgrid.PairGrid at 0x1c2b14e750>



### 1.2.2 Perspectivas de mejora del conteo celular :

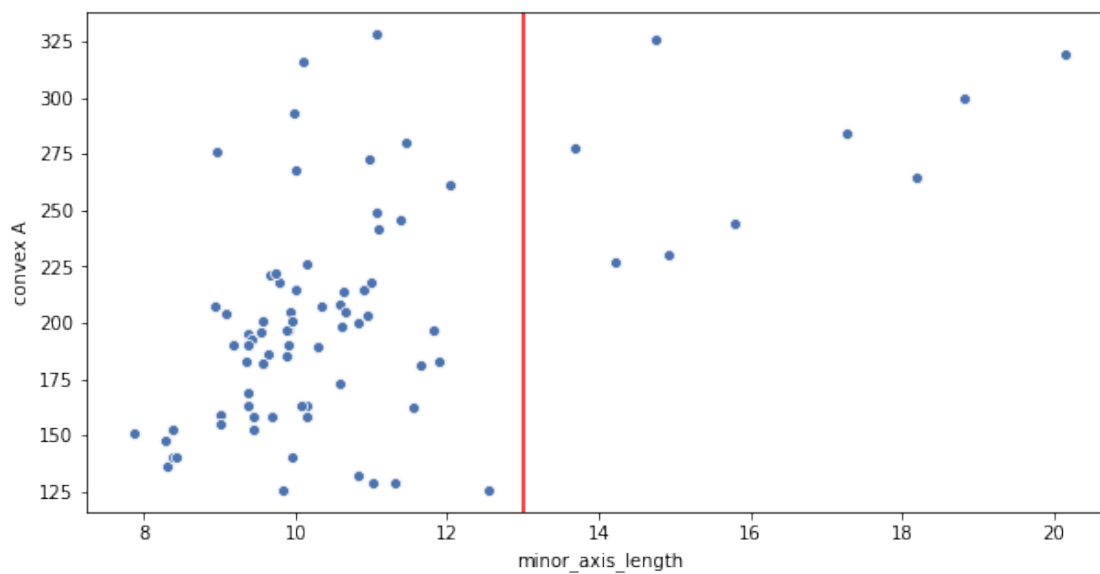
Aquí podemos observar que las propiedades que probablemente sean más útiles para mejorar la clasificación se encuentran en las primeras cuatro filas y en las últimas tres columnas. Ya que en los *scatterplots* generados hay una separación considerable a lo largo de los ejes horizontales entre los que creeríamos son **verdaderos positivos** (cúmulo de puntos mayor) y los que son **falsos positivos** (puntos dispersos entre sí y alejados del cúmulo principal).

Observando a qué regiones pertenecen y etiquetándolos, se podría entrenar un modelo de inteligencia artificial sea una red neuronal o una máquina de soporte vectorial para poder clasificarlos eficientemente.

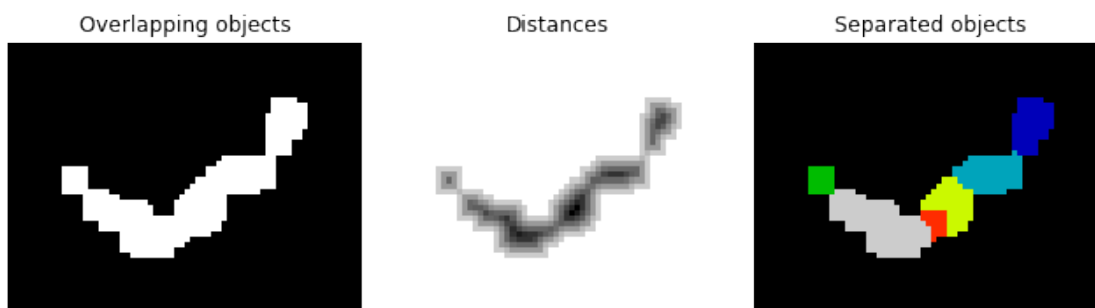
La máquina de soporte vectorial encontraría (idealmente) el hiperplano que mejor separase los

cúmulos. Un ejemplo artificial (construido a mano) se muestra a continuación.

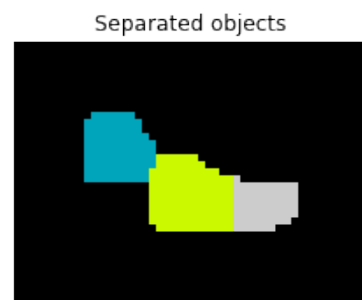
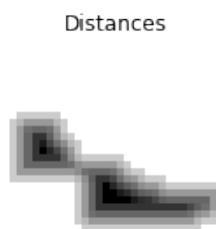
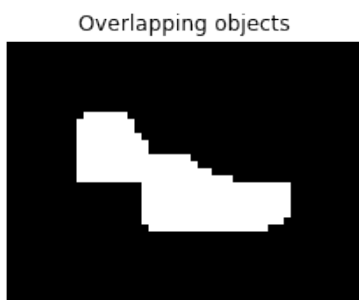
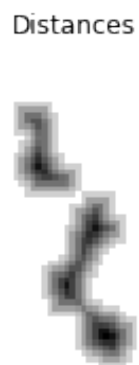
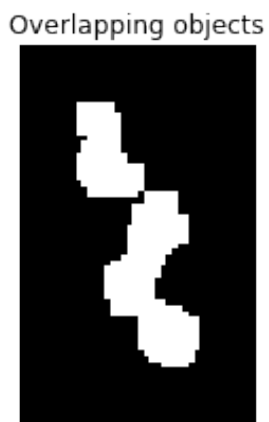
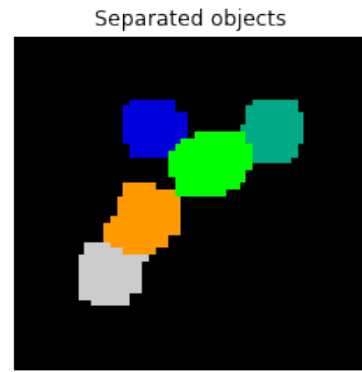
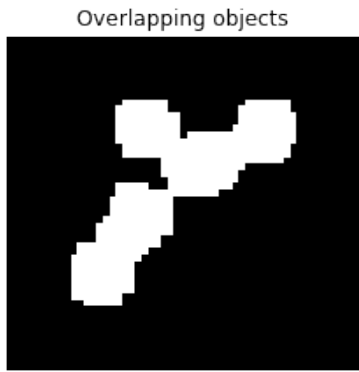
[84]: <matplotlib.lines.Line2D at 0x1c30b27d50>



### 1.2.3 Extra : Visualización de subsegmentaciones gracias al algoritmo Watershed.







Overlapping objects



Distances



Separated objects



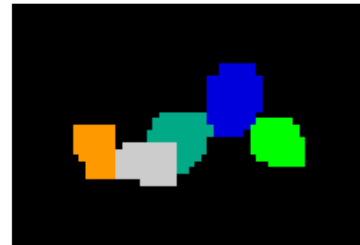
Overlapping objects



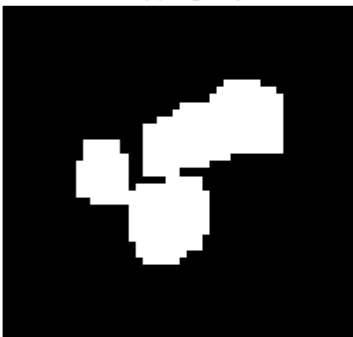
Distances



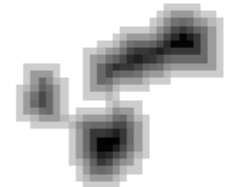
Separated objects



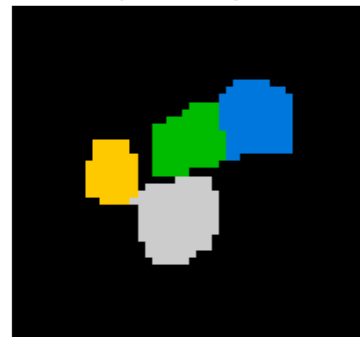
Overlapping objects



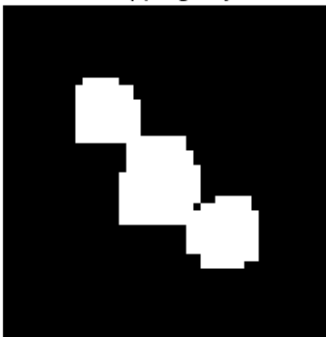
Distances



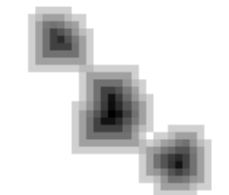
Separated objects



Overlapping objects



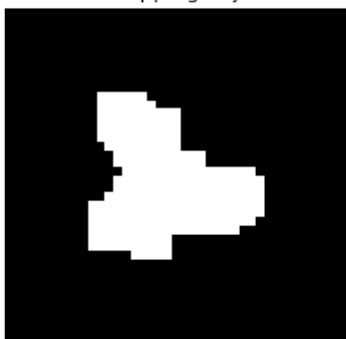
Distances



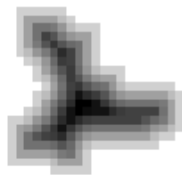
Separated objects



Overlapping objects



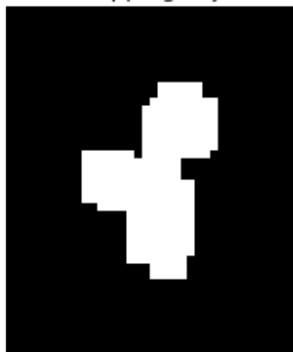
Distances



Separated objects



Overlapping objects



Distances



Separated objects



Overlapping objects



Distances



Separated objects

