



TULA

PROCESAMIENTO DE IMÁGENES I

TRABAJO PRÁCTICO N° 2

GRUPO 14

Integrantes:

López, Eugenio
Peroni, Antonio
Pistelli, Pablo

Noviembre 2023

INTRODUCCIÓN

En este Trabajo Práctico se plantearon dos problemas para resolver relacionados al Procesamiento de Imágenes.

PROBLEMA 1 – Detección y clasificación de Monedas y Dados

La idea de este problema es que a partir de una imagen con monedas de diferentes tamaños y un par de dados, se debe lograr segmentar las monedas y los dados de manera automática. Luego, clasificar los distintos tipos de monedas y realizar un conteo de las mismas. Por último, determinar el número que presenta cada dado.

PROBLEMA 2 – Detección de patentes

A partir de imágenes de la vista anterior o posterior de diversos vehículos donde se visualizan las correspondientes patentes, se debe implementar un algoritmo que detecte automáticamente las patentes y segmente las mismas. Luego, se debe implementar un algoritmo de procesamiento que segmente los caracteres de la patente detectada en el punto anterior. Informar las distintas etapas de procesamiento y mostrar los resultados de cada etapa.

PROBLEMA 1

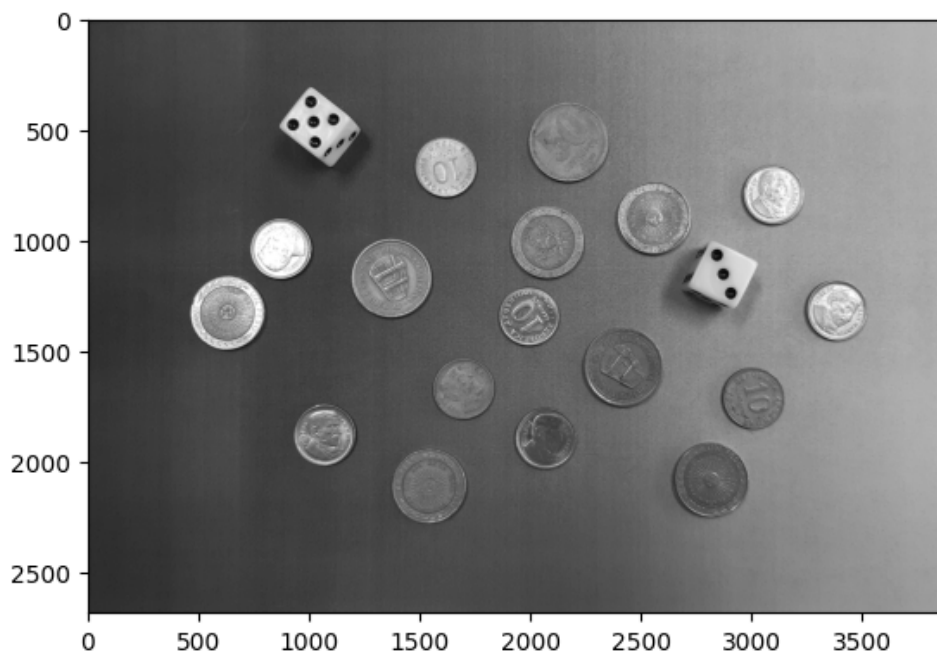
En este problema se deben segmentar los elementos de la imagen (dados y monedas) y obtener información específica de cada tipo.

Procedimiento para identificación de monedas:

a. Carga de imagen

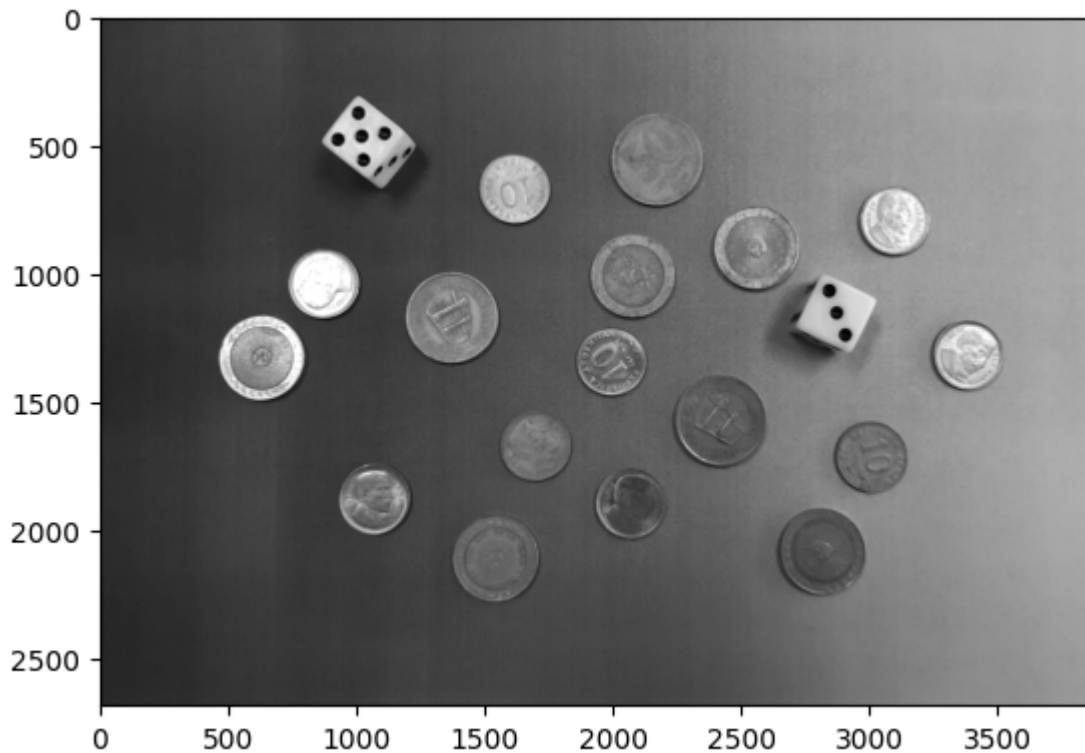


b. Paso a escala de grises



c. Homogeneización del fondo

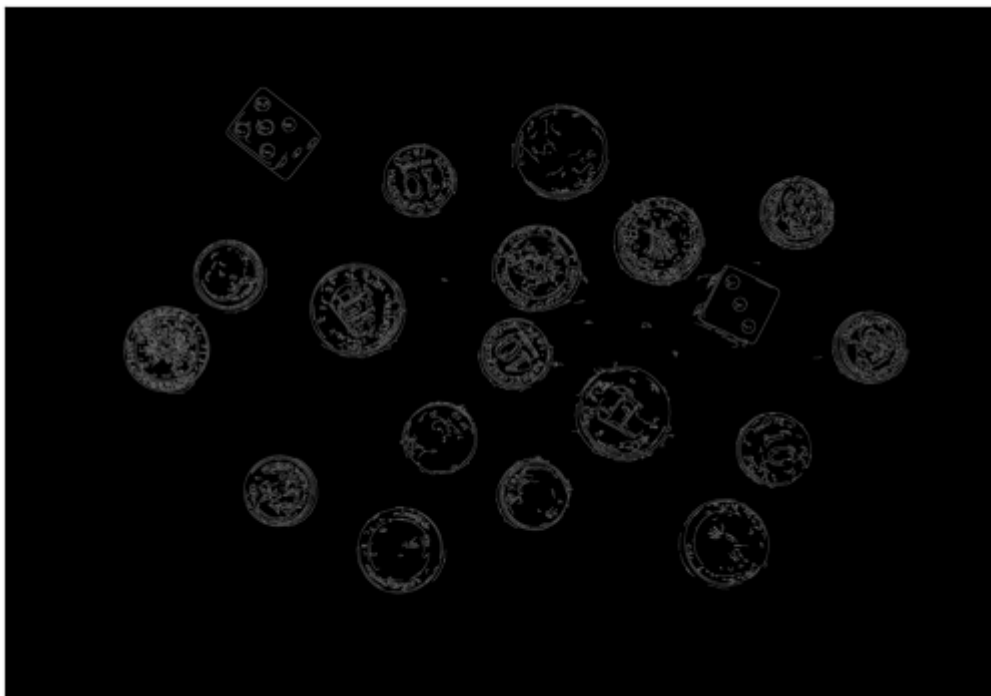
Para obtener un fondo uniforme y poder identificarlo completamente se aplica medianBlur. El **primer punto** de ajuste fue el tipo de blur y el tamaño del kernel.



d. Detección de bordes

Para detectar los bordes de los dados y monedas utilizamos Canny.

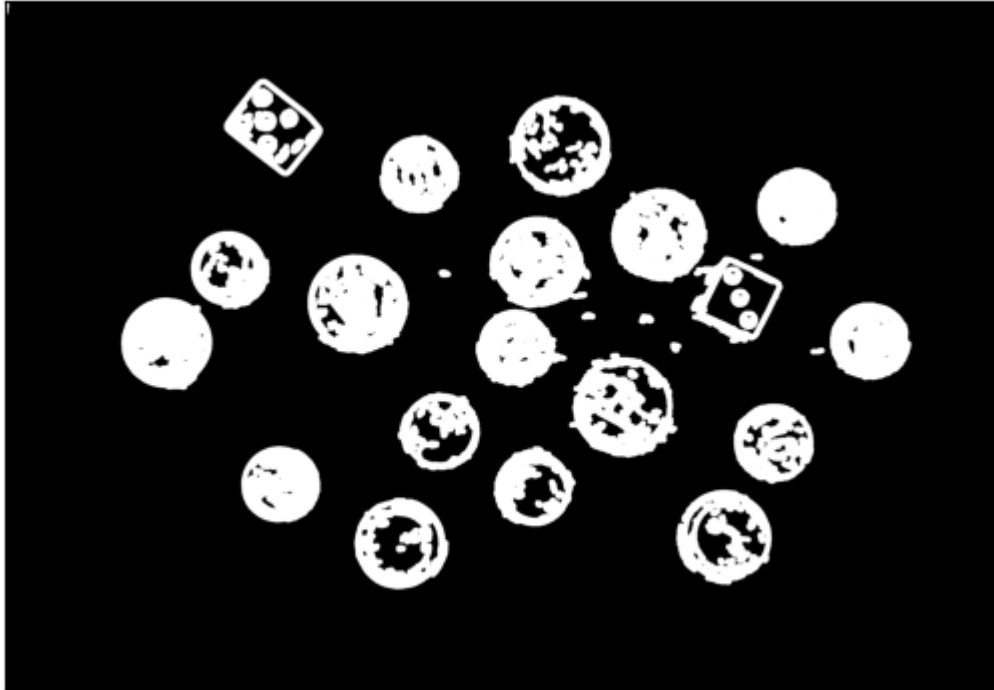
El **segundo punto** de ajuste fueron los parámetros de esta función (umbrales y apertura).



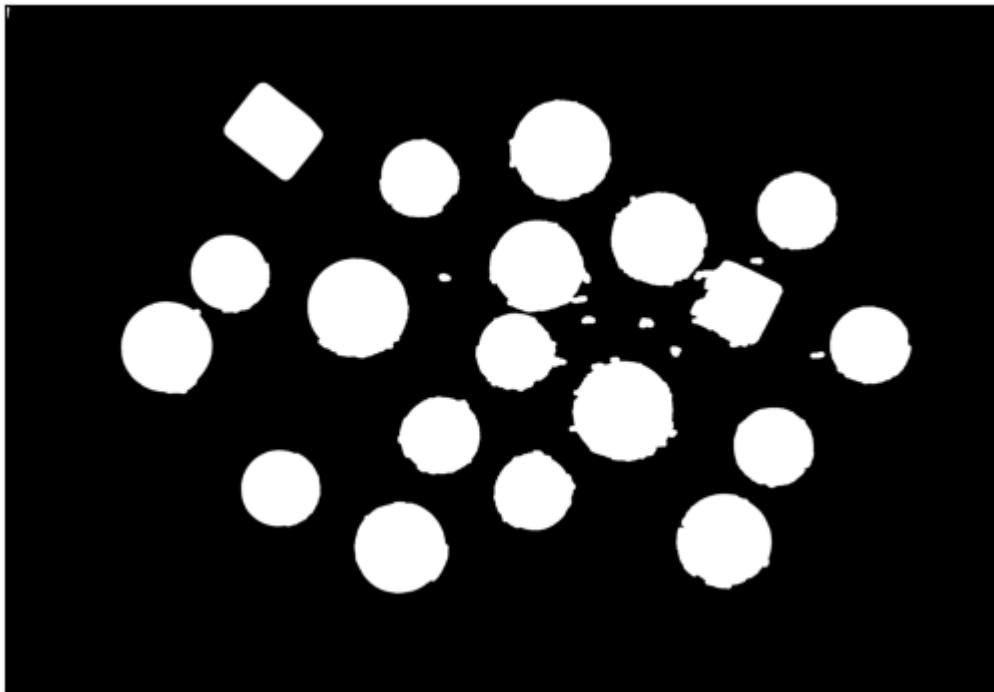
e. Homogeneización de los elementos

Una vez obtenidos los bordes, al solamente interesar los contornos externos (en principio) de los elementos de la imagen, se aplicaron transformaciones para conseguir imágenes planas y rellenas.

- En primer lugar aplicamos dilatación (**tercer punto** de ajuste):

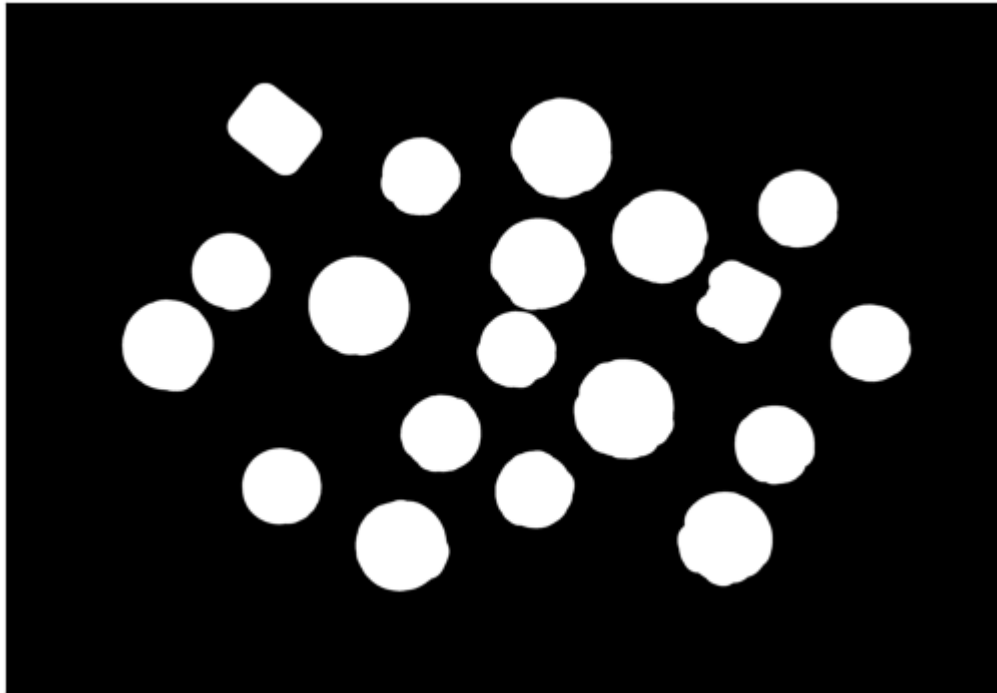


- Luego rellenamos los huecos (**cuarto punto** de ajuste):



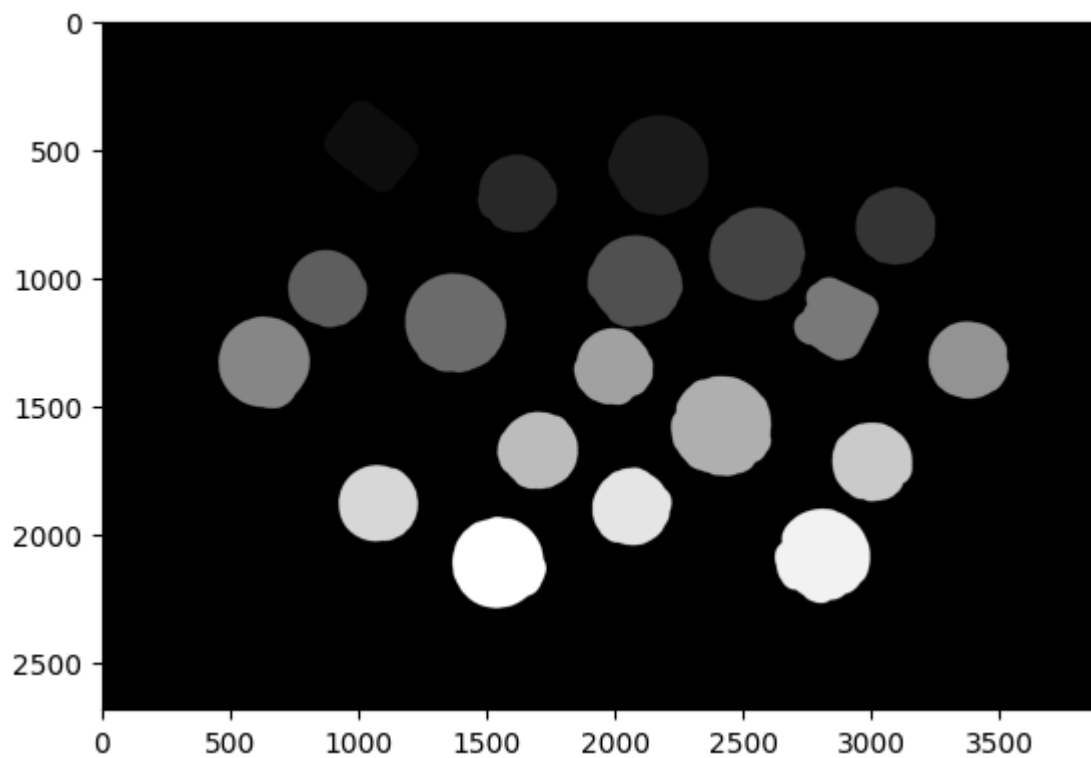
-

- Finalmente aplicamos apertura para eliminar el ruido en la imagen (**quinto punto de ajuste**):



f. Componentes conectadas

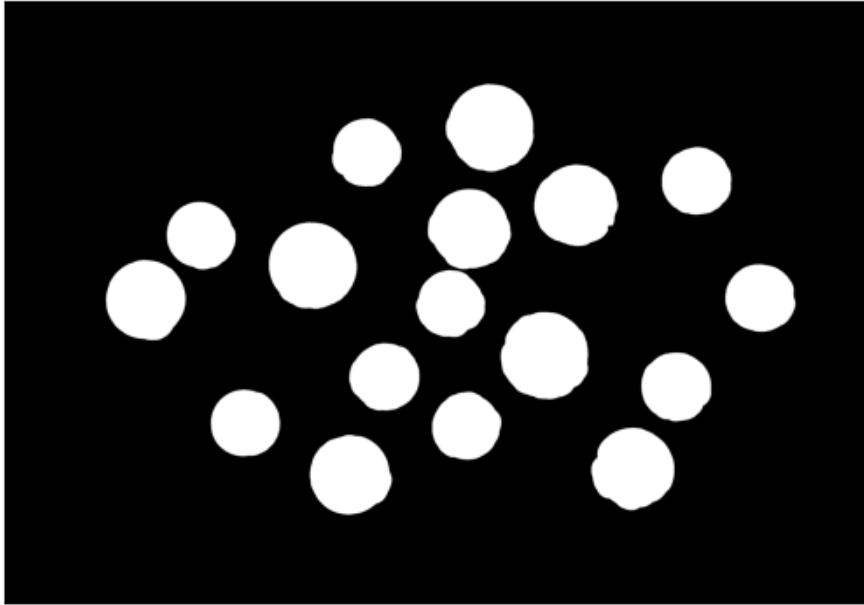
Identificamos las componentes conectadas en la imagen obtenida en el paso anterior.



g. Filtro por factor de forma

De las componentes obtenidas, en cada caso obtenemos el área y el perímetro y seleccionamos las que cumplen con un factor de forma acorde a un círculo. Este fue el **sexto punto** de ajuste, definido en $\rho > 0.66$.

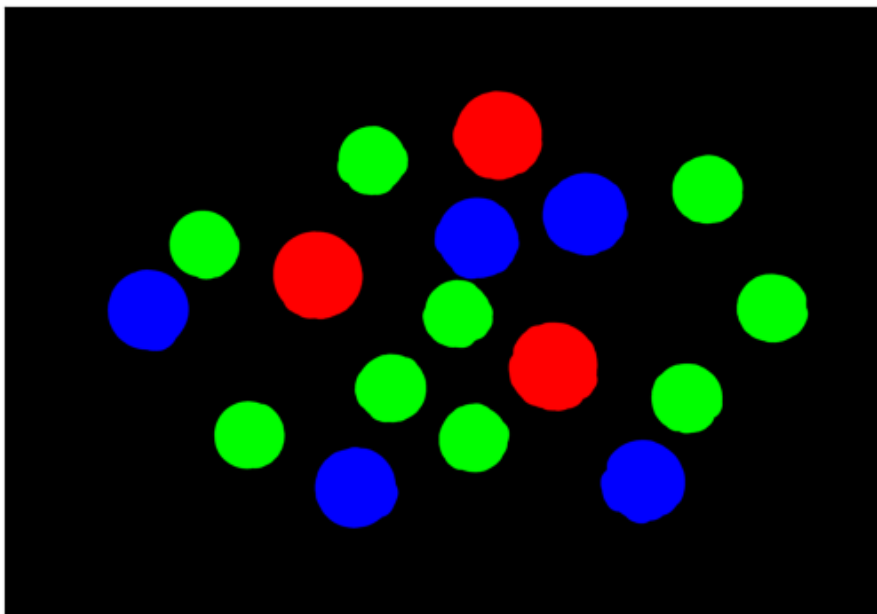
Imagen sin dados



h. Separación por tamaño

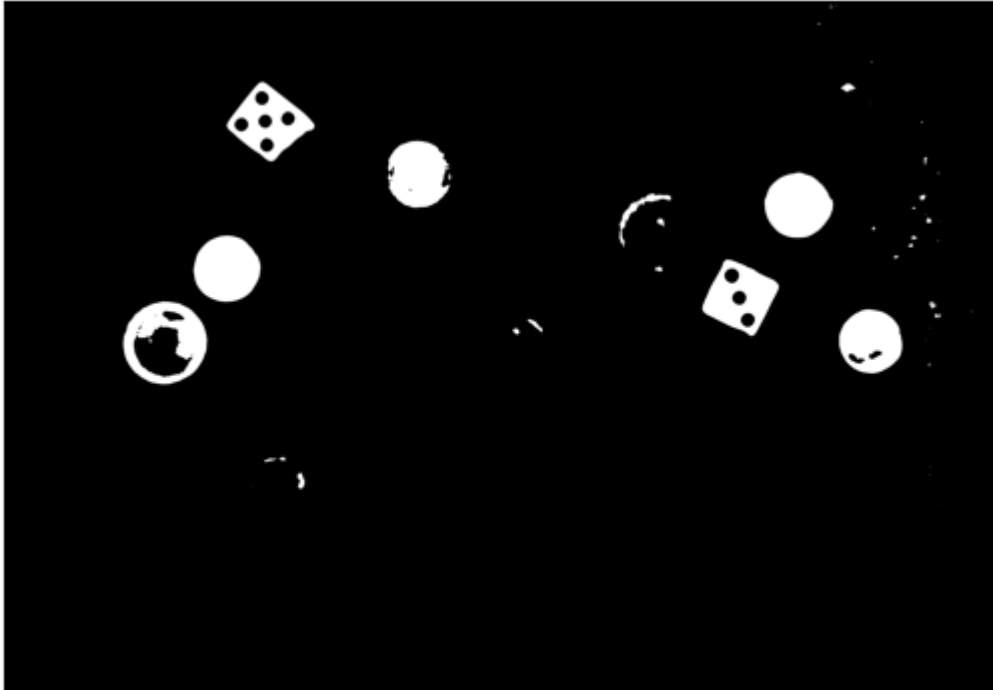
Dentro de la misma función de selección de círculos, aplicamos umbrales de área para separar las monedas por tamaño. Los rangos de área fueron el **séptimo punto** de ajuste de este algoritmo

Monedas clasificadas

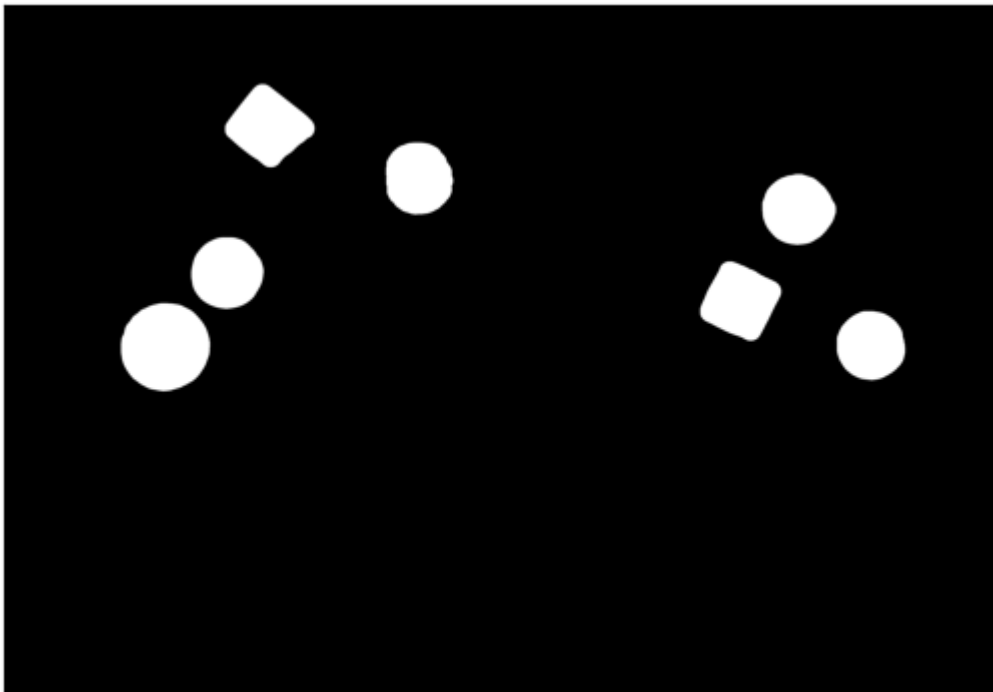


Para la detección de dados repetimos los mismos pasos. En este caso, en la selección de los parámetros de cada transformación buscamos resaltar los dados. El resultado obtenido fue el siguiente:

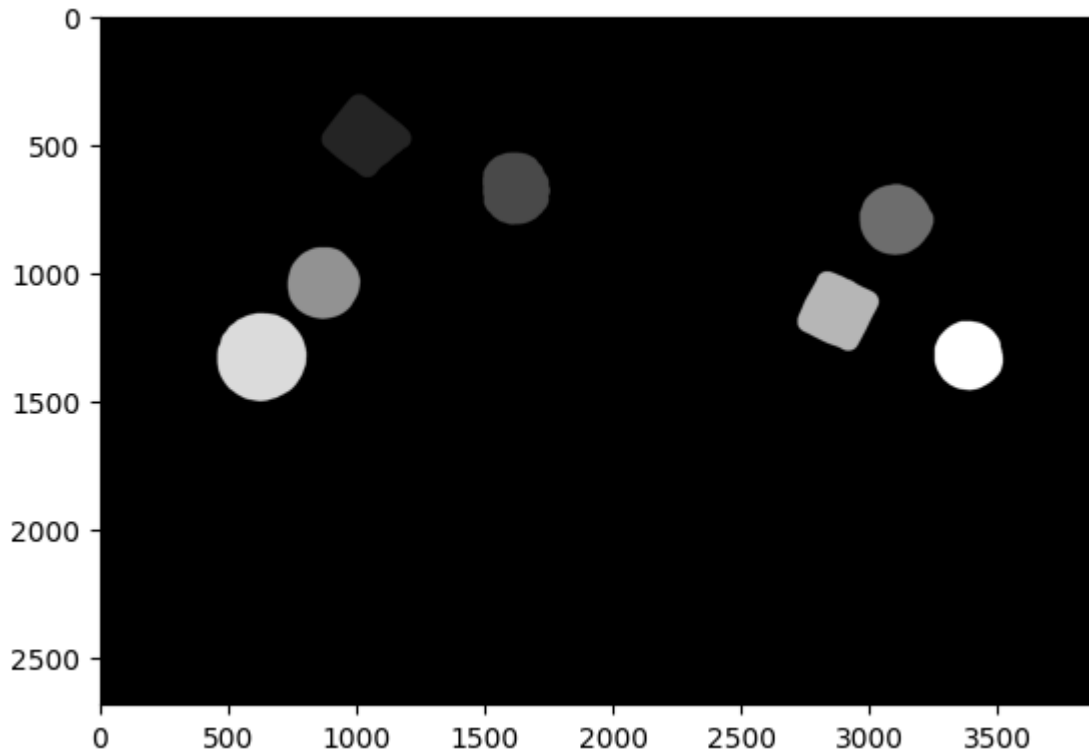
- Imagen umbralada sin bordes



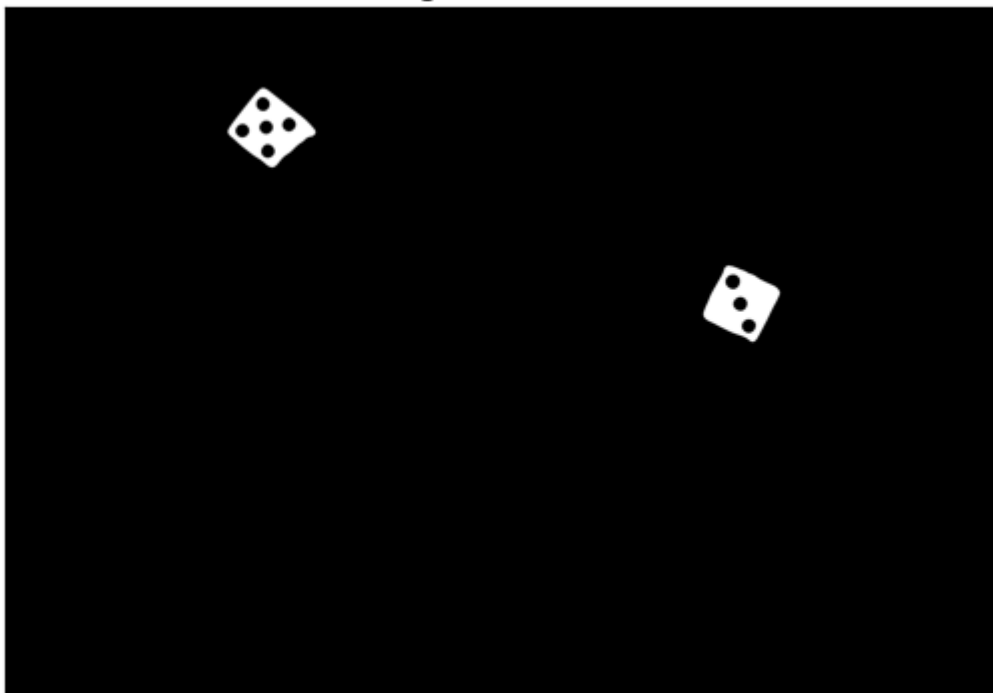
- Imagen dilatada, rellena, con apertura



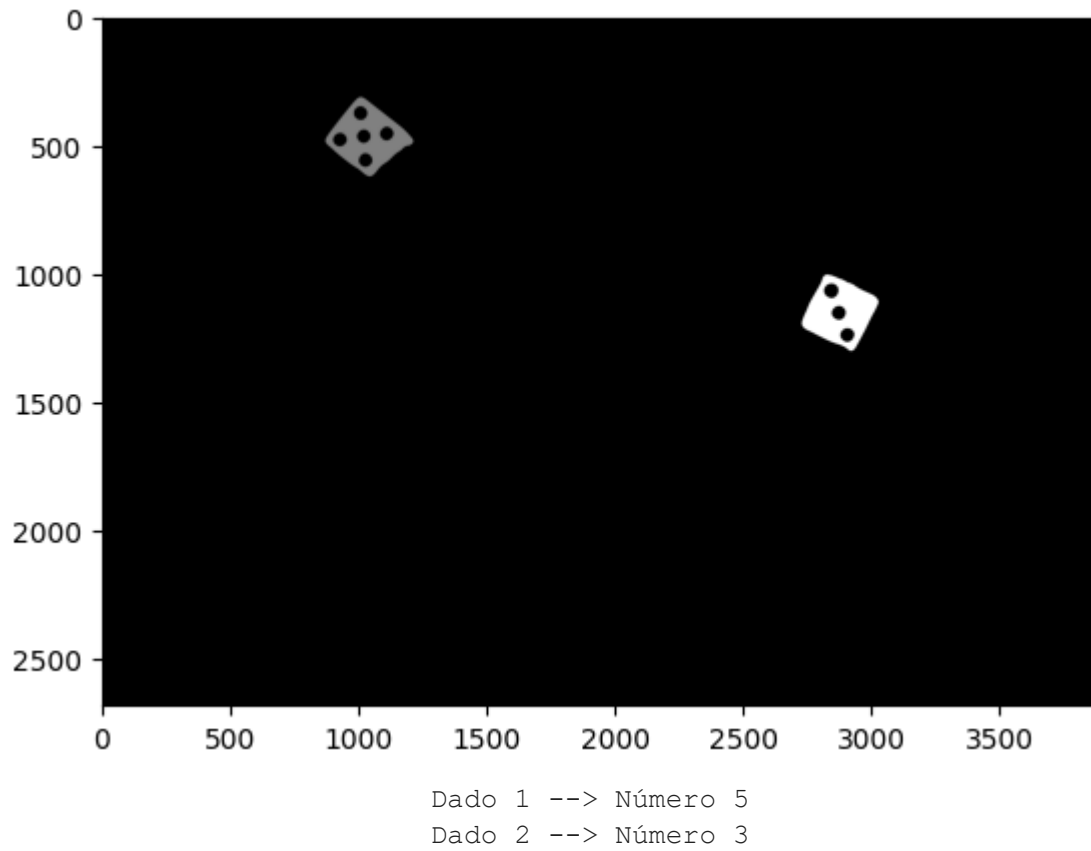
Aplicamos detección de componentes conectadas y esta vez utilizamos un factor de forma diferente para detectar cuadrados. El resultado fue $\rho < 0.65$.



Una vez detectados los dados, generamos una máscara y la aplicamos a la imagen umbralada sin bordes.



Finalmente, identificamos los contornos para cada dado y de esa manera podemos contar la cantidad de círculos internos de cada uno, obteniendo el número que representa la cara visible.

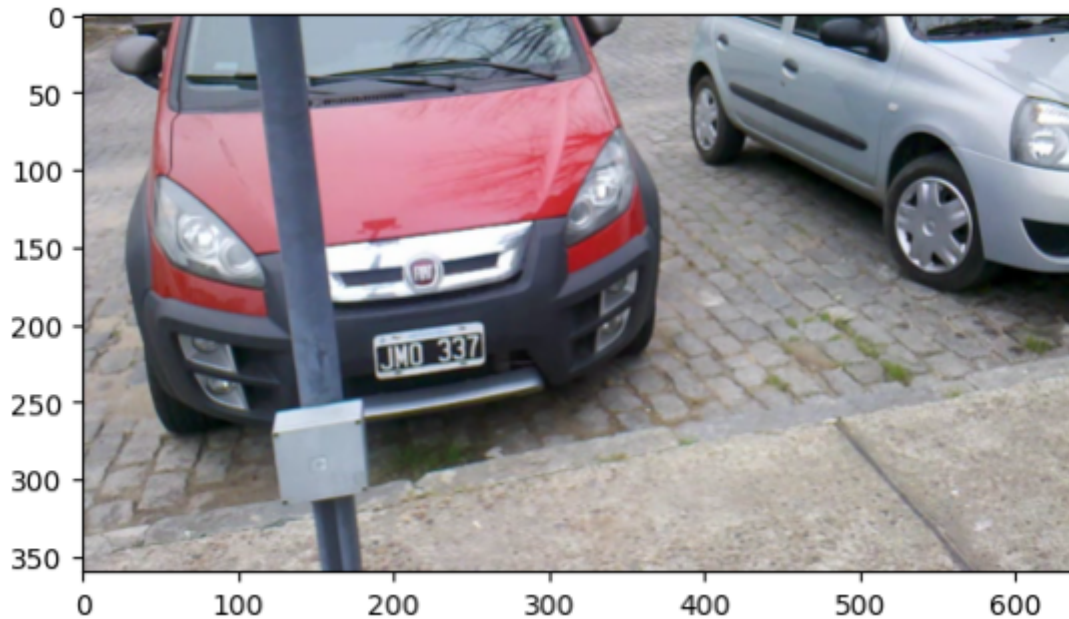


PROBLEMA 2

En el segundo ejercicio se deben identificar las patentes de los vehículos en las fotografías. En este caso fue necesario definir varios parámetros para las distintas funciones aplicadas, pero no se encontraron parámetros que devuelvan un resultado satisfactorio para todos los casos. Para los parámetros elegidos se obtuvo una eficacia de 8/12.

El procedimiento aplicado fue el siguiente:

a. Carga de la imagen



b. Paso a escala de grises



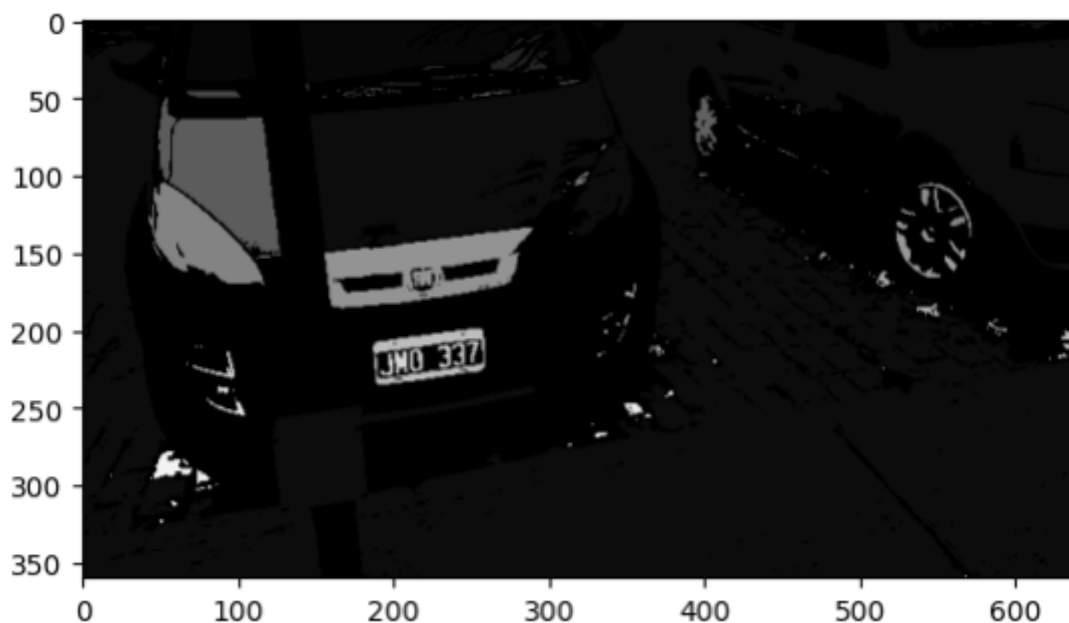
c. Umbralado binario

El **primer punto** de ajuste fue el valor del umbral. Se realizaron numerosas pruebas registrando cuáles eran los valores que mejor se aplicaban a la mayoría de las fotos. Los resultados más satisfactorios los encontramos entre 130 y 140.



d. Componentes conectadas

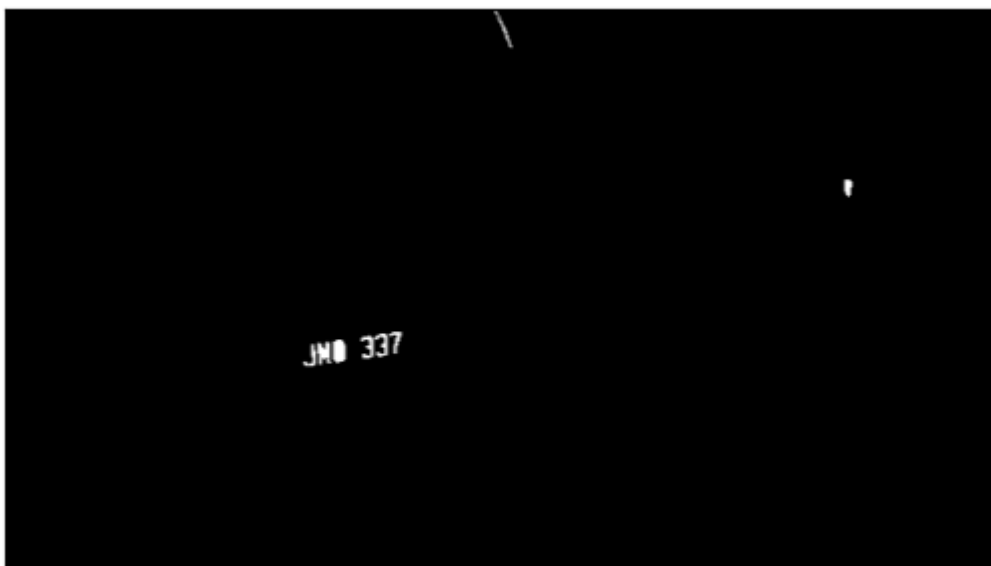
Luego se identificaron las componentes conectadas para poder independizar los caracteres que pertenecen a la patente. En algunos casos, algunos caracteres quedan conectados a los bordes de la patente o a otros caracteres. El umbralado se eligió para evitar lo más posible esas situaciones.



e. Selección de componentes conectadas según área

El **segundo punto** de ajuste fue el área de las componentes buscando eliminar los elementos que no pertenecen a la patente. Dependiendo del carácter, el área varía bastante. Finalmente encontramos las áreas deseadas entre 10 y 100 píxeles.

El **tercer punto** de ajuste también se aplicó en este paso y fue la relación de aspecto alto/ancho. Luego de las pruebas, el rango de relación de aspecto definido fue entre 1.4 y 2.5.

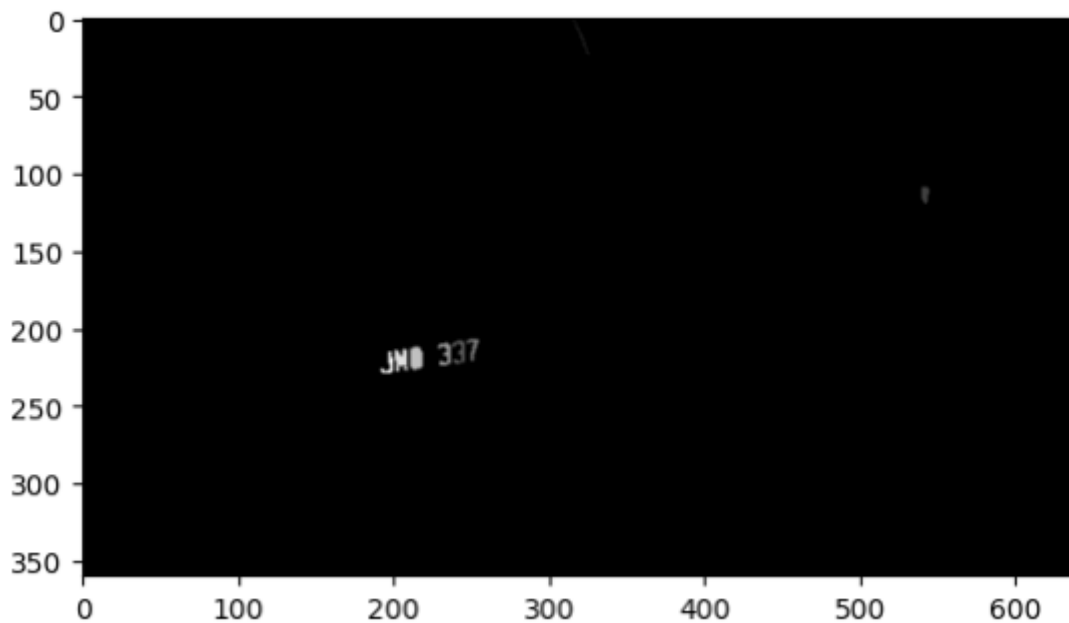


f. Componentes conectadas cercanas

Una vez obtenidas las componentes con área y relación de aspecto deseada se buscan las que están agrupadas. Para esto se definió una función *puntos_cercanos* que recibe como parámetros una lista o arreglo de puntos (x,y) y una distancia máxima entre ellos d . La función devuelve una lista de los puntos que se encuentren a distancias menores a la definida. Se utiliza esta función con la lista de centroides exceptuando el del fondo.

El **cuarto punto** de ajuste fue la distancia máxima entre ellos. Luego de varias pruebas se definió $d=20$.

Para los casos donde se pudieron independizar correctamente los 6 caracteres, con esta función se obtienen los 6 centroides de los mismos y se eliminan los elementos de la imagen que pueden haber quedado sin filtrar en los pasos anteriores. Sin embargo, en algunos casos en los que los parámetros anteriores generan la pérdida de algún carácter, esta función también es de utilidad si identifica 5 de los 6 componentes buscados. También consideramos válido este resultado para nuestro objetivo.



g. Cálculo de dimensiones de la patente

Para la lista final de componentes conectadas se elige el primer carácter para ubicar el vértice superior izquierdo del recorte que contendrá la patente.

Tanto la ubicación del vértice como el tamaño del recorte se calculan en función del tamaño y posición de los caracteres.

Vértice inicial:

```
[min(x_centroides) - max(width)*2 ; min(y_centroides) - max(heigh)]
```

Ancho:

```
max(width)*12
```

Alto:

```
max(heigh)*3
```

h. Recorte de la imagen original en el área seleccionada

De esta manera, obtenemos las ROI de cada imagen para su posterior análisis.

Patente



Como mencionamos al comienzo, el resultado fue satisfactorio para 8 de las 12 imágenes provistas. Una estrategia que podría utilizarse para mejorar el algoritmo anterior sería variar los puntos de ajuste para obtener las patentes de las imágenes faltantes.

Llegado al punto *f* de identificación de las componentes conectadas cercanas, si el resultado obtenido no contiene 6 elementos, podría reiniciarse el procedimiento con nuevos puntos de ajuste.

Para la segunda parte de este ejercicio, utilizaríamos los recortes obtenidos en el algoritmo anterior y aplicaríamos las mismas técnicas que en la detección de monedas para lograr segmentar los caracteres de cada patente. Este punto no llegó a ser resuelto.