

# Universidad Nacional de Rosario

Facultad de Ciencias Exactas, Ingeniería y Agrimensura



---

## Color, Restauración y Morfología



## TUIA - IA 4.4 Procesamiento de Imágenes

Trabajo Práctico Nro. 2

---

**Fecha:** 20/11/2023

**Docentes:**

- Gonzalo Sad
- Facundo Reyes
- Julián Álvarez

**Grupo 9:**

- Revello Simon
- Giampaoli Fabio
- Ferrucci Constantino
- Arevalo Ezequiel

## ***Presentación***

Se presentan dos problemas recurrentes en el área de procesamiento de imágenes que requieren resolución utilizando como fuente el contenido teórico/práctico de la asignatura *Procesamiento de Imágenes Digitales 1*.

En este informe se pretende reflejar el estado inicial de los problemas, el procedimiento conceptual de resolución de cada uno por separado, y su presentación final junto con las conclusiones obtenidas.

## ***Problema 1***

Se presenta una fotografía tomada por un smartphone que contiene monedas de distintos valores, tamaños y colores distribuidas de forma aleatoria en la imagen. Además hay presentes dos dados, cada uno con su cara superior conteniendo la representación de dos números diferentes.

El objetivo es que mediante un algoritmo de procesamiento de imágenes, se puedan segmentar estos objetos de forma automática. Pero no solo sus formas, sino también sus tamaños, y en cierto modo, sus posiciones, debido a que en el caso de los dados, la cara superior es la que presenta el número de interés solamente.

Se requiere realizar un conteo de las monedas presentes en la imagen comprendiendo que cada tipo presenta un valor dado. Lo mismo ocurre con los dados, contabilizando sólo el número que aparece en la cara superior de los dados.



**Problema 2**

El segundo problema presentado trata del procesamiento de un conjunto de imágenes que son fotografías tomadas de autos, haciendo enfoque en sus patentes. Las patentes en cada caso tienen ángulos, iluminación, posiciones y características específicas del contexto de la imagen. En total contamos con 12 imágenes como esta en una carpeta.

El objetivo es la creación de un algoritmo de procesamiento de imágenes que automáticamente pueda procesar las imágenes para que pueda segmentar de cada imagen, primero la forma y posición de la patente como tal; y segundo, segmentar los caracteres de las patentes, pudiendo tomar una imagen dada y segmentar para reconocer fácilmente los caracteres correspondientes a la patente de cada imagen.

Esta es solo una muestra del conjunto de imágenes con el que se debe trabajar:

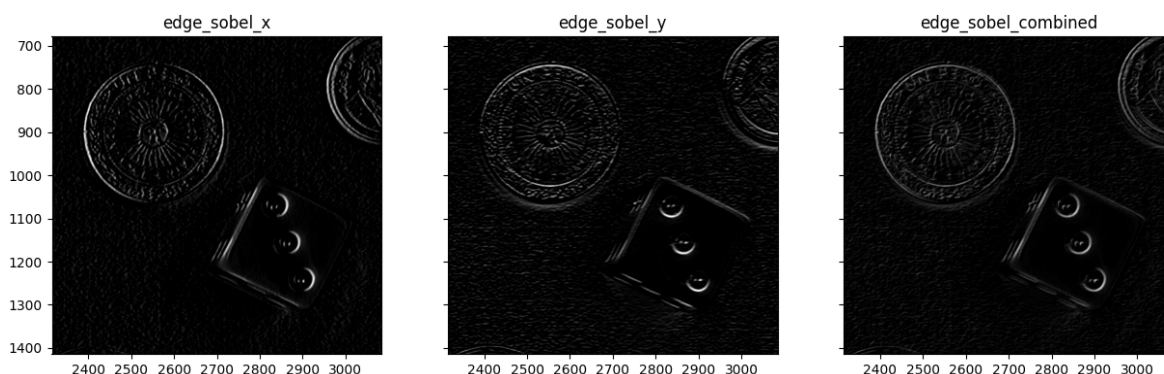


### Resolución - Problema 1

Como para el objetivo del problema no es de utilidad trabajar la imagen original con sus colores, inicialmente definimos la imagen en escala de grises para facilitar el procesamiento a una sola dimensión de profundidad. Además se reduce el ruido de la imagen con un filtro de blur para suavizar la información

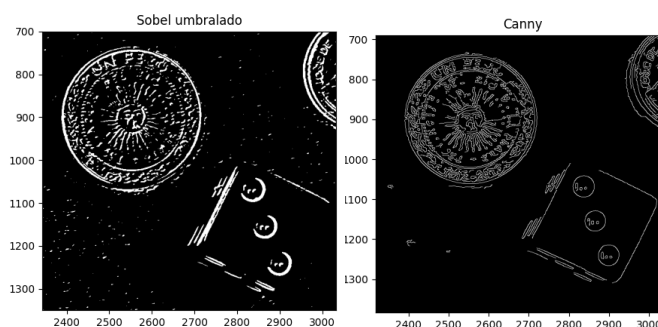


Durante el cursado de la asignatura hemos indagado en diferentes métodos de detección de bordes. Inicialmente se ha decidido utilizar Sobel como algoritmo de este tipo. Particularmente se han probado tres alternativas, aplicando Sobel en los distintos ejes por separado, y luego combinándolos, aplicando Sobel en ambas direcciones por separado y combinando manualmente los resultados individuales de cada eje.

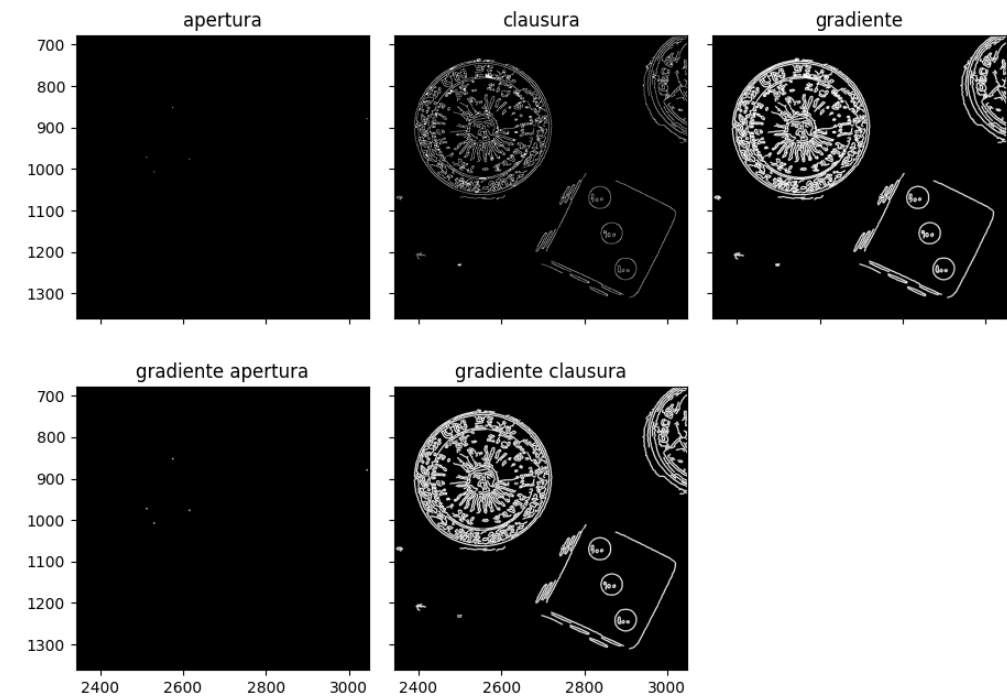


Como la imagen con los ejes combinados tiene resultados que combinan este filtro en ambas direcciones, se elige esta para realizar una muestra de umbralado de la misma

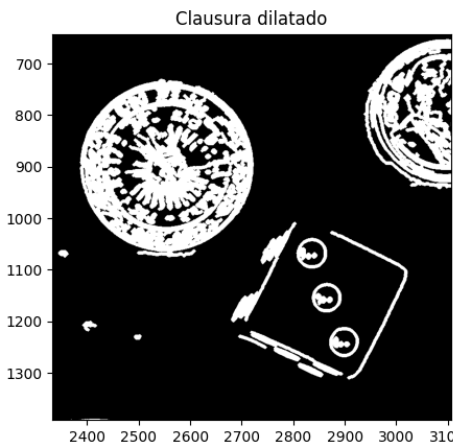
El umbralado sobre la imagen presenta mejores resultados. Pero posteriormente se nota que con el algoritmo de Canny para detección de bordes simplifica mucho la codificación, y el resultado logrado es más apropiado para continuar con el desarrollo:



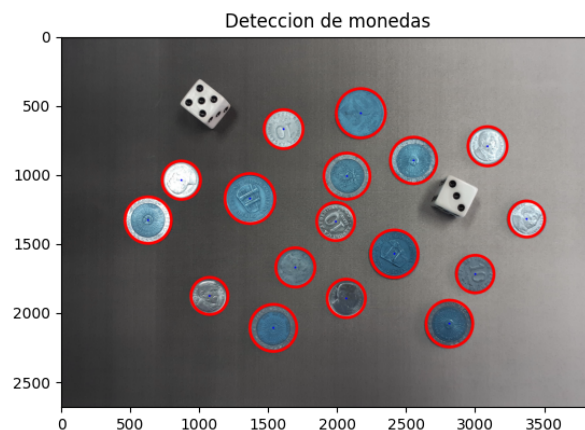
Sobre esta imagen, se aplican métodos de morfología con el fin de lograr que los bordes detectados por canny sean más precisos y visibles.



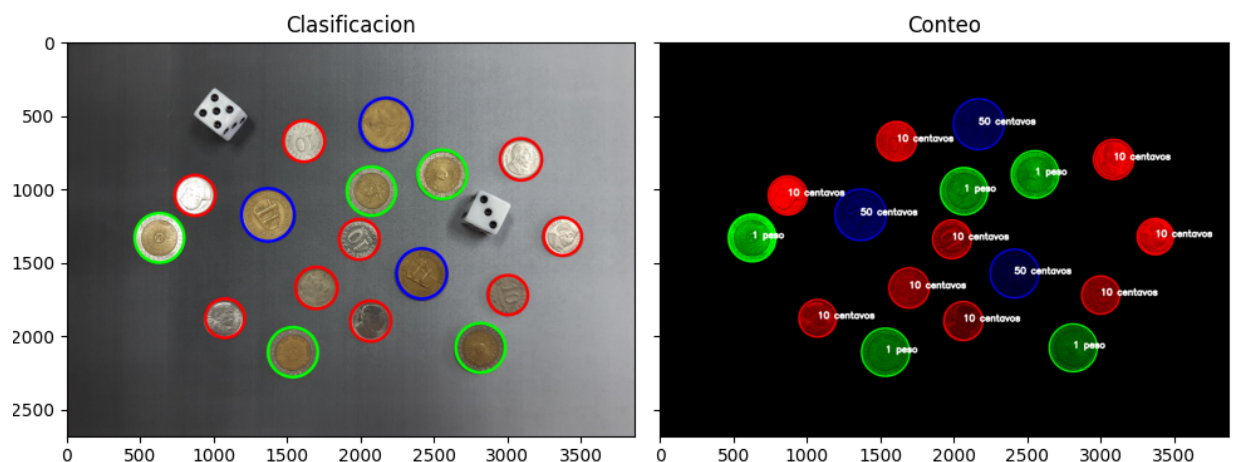
Notar como una operación de clausura (fc) sobre la imagen logra resultados suficientemente buenos sobre los bordes de la imagen procesada por Canny. Sobre esta imagen se elige dilatar para ensanchar los bordes para hacer evidente las formas de cada objeto.



Sobre esta imagen se crea una máscara. La imagen es tomada y utilizada por un detector de círculos *HoughCircles* con ciertos parámetros para que pueda encontrar cada moneda correctamente, y no confundir los puntos de los dados con monedas.



Una vez obtenida la circunferencia representada por cada moneda, se crea un mapeo de que valor y tipo de moneda que corresponde a cada circunferencia, encontrando manualmente el rango de valores en el que se mueven los radios de cada circunferencia. Estableciendo valores de radio máximo y mínimo para cada tipo de moneda, se llega a una clasificación como la siguiente:

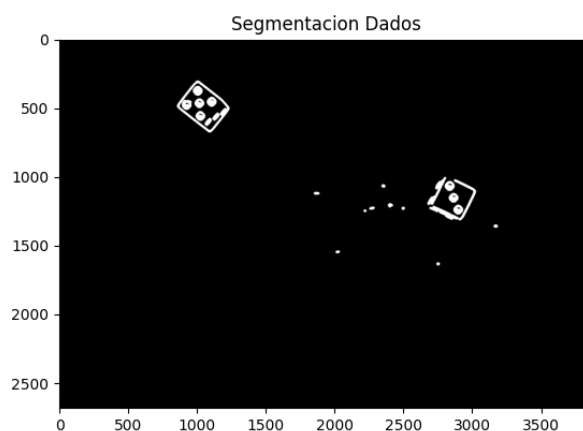


Mediante una función y el mismo mapa de valores de moneda que se han almacenado los recortes de cada circunferencia, se crea una función que contabiliza los valores de las monedas:

```
>>> suma_total = contar_y_sumar_valores(coins_map)
10c: 9 recortes, Valor individual: 0.9
50c: 3 recortes, Valor individual: 1.5
1p: 5 recortes, Valor individual: 5.0

>>> print(f"\Suma total de valores: {suma_total}")
Suma total de valores: 7.4
```

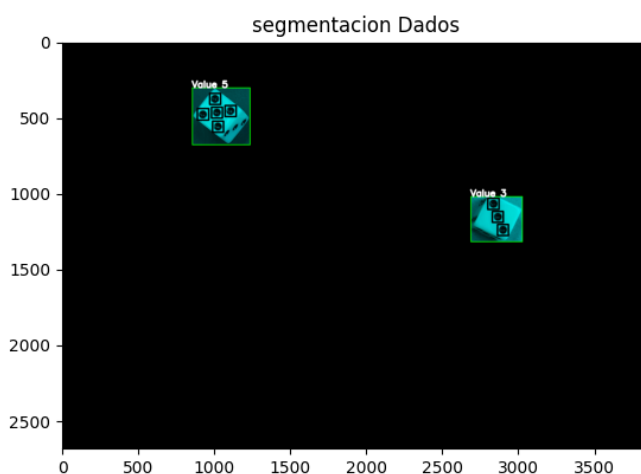
Para el reconocimiento de los dados se utiliza la misma máscara de las monedas. Pero su uso sera para eliminar las monedas de la máscara esta vez, quedándonos solamente con los objetos no circulares encontrados por la función previa, es decir, los dados:



En escencia, lo que se hizo fue 'quemar' a negro los círculos encontrados, desde el centro hasta pasando el mayor radio de circunferencia. Eliminado todas las circunferencias que cumplían el criterio anterior de la imagen. La misma fue dilatada para resaltar los bordes.

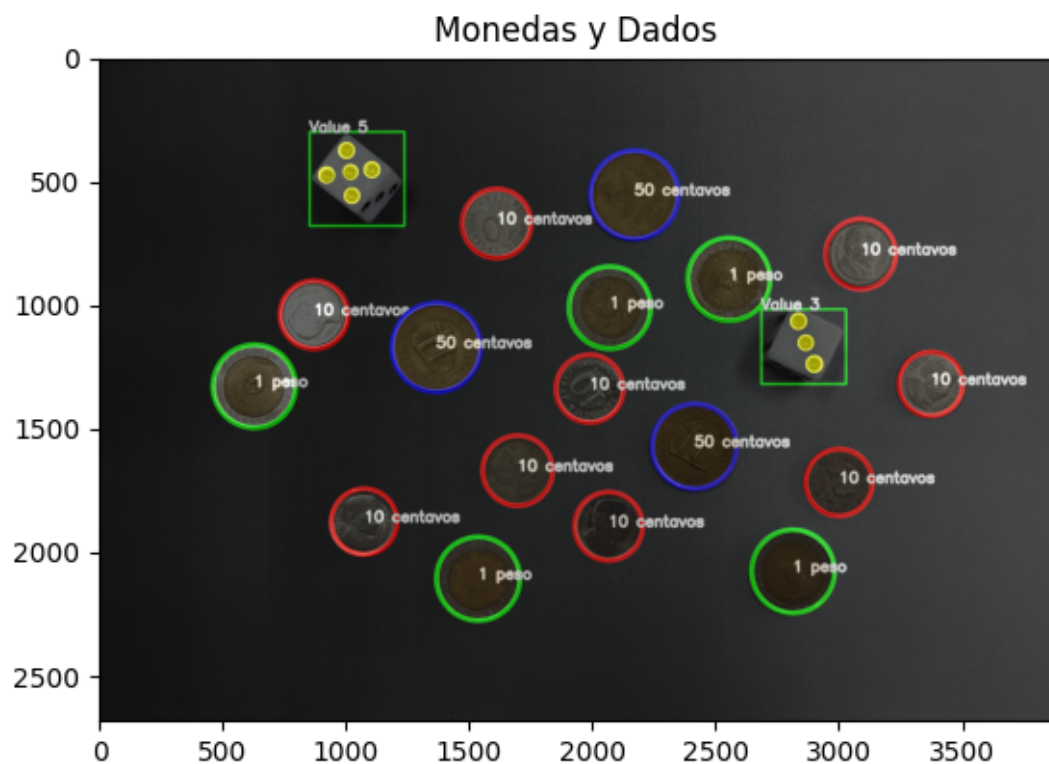
De esta imagen se obtienen los componentes conectados. Y esto es de utilidad debido a que sobre estos componentes podemos filtrar por aquellos que nos interesan de cada dado: La forma del dado; y los círculos de la cara superior.

La obtención de cada componente de interés se hace mediante filtros de área. Mediante ajuste manual de los umbrales de área, se encuentran aquellos que permiten deshacernos de los puntos laterales de ambos dados (área mayor a 3000). Luego, establecemos que la forma de los dados como tal tienen áreas mayores a 10000. Y las áreas entre 3700 y 4200 representan los círculos de la cara superior de cada dado.



La obtención del valor se obtiene aplicando HoughCircles sobre el área de cada dado y almacenando cada circunferencia en un diccionario. La longitud de los recortes del diccionario sería la cantidad de puntos de la cara superior del dado.

Como resultado final, de las dos estructuras de datos que almacenan cada objeto, valor y color para la segmentación, se consigue unificar todo lo aplicado en la imagen final satisfactoriamente:





## Resolución - Problema 2

Para la resolución del ejercicio se utiliza un enfoque más manual. La obtención de los caracteres se logra aplicando filtros y transformaciones a las imágenes para filtrar aquellas de las imágenes que cumplen con criterios encontrados manualmente de qué características tienen los caracteres.

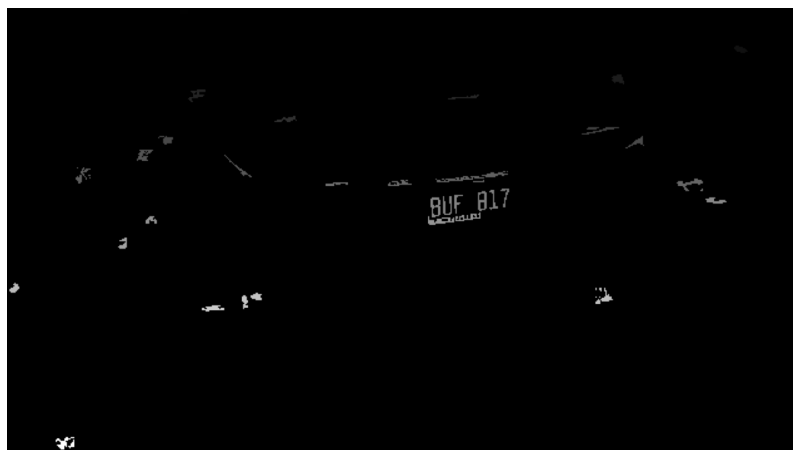
El primer interés es que de una fotografía de una patente de auto, podamos hacer transformaciones tal que nos quede una máscara con solo los caracteres de la patente de la iteración actual.

Para ello, como no interesa la información que puede proveer una escala de colores de tres dimensiones, y es mucho más simple trabajar con una versión binaria de la imagen, se escala a grises y se umbrala con un valor específico para la imagen que permita la clara visibilidad de los caracteres de la patente:



De esta imagen se obtienen luego los componentes conectados presentes en la imagen. Como el umbral seleccionado fue específico para la imagen particular para maximizar la visibilidad e individualidad de cada caracter, al tratar con componentes conectados sabemos de antemano que entre todos los encontrados están los caracteres.

Para encontrar los caracteres se realizan filtros sobre los componentes encontrados en la imagen. Cómo podemos obtener la información del rectángulo que encierra el área de un componente conectado, podemos realizar filtros por área mínima y máxima de los componentes. Se consiguen los umbrales de área mínima y máxima que logran encontrar entre otros elementos, los caracteres.

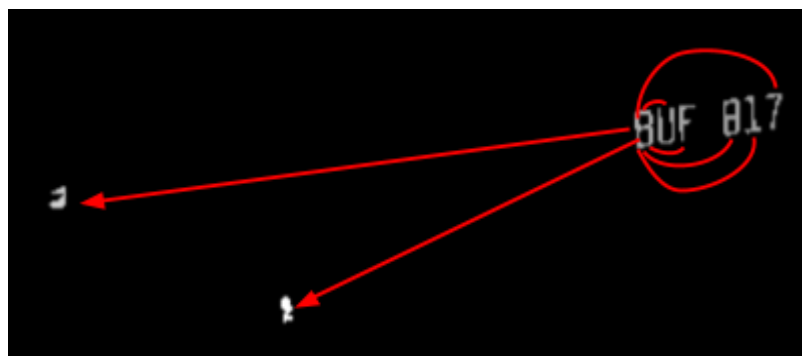


Pero como aun así siguen apareciendo elementos no deseados en la imagen que cumplen el criterio de área, se agrega un segundo criterio. El segundo criterio de filtro es la relación de aspecto. Definimos la relación de aspecto como el cociente entre el alto y el ancho de una imagen. Encontramos manualmente el rango de valores de relación de aspecto de los caracteres específicos de cada imagen. De esta forma, nos quedamos solo con algunos pocos elementos que cumplen los criterios:



Como notamos de forma general para todas las imágenes que en este punto, los elementos sobrantes a los caracteres se encuentran relativamente lejos de los caracteres, se utiliza un tercer filtro sobre esta última imagen.

El tercer filtro implica medir la cercanía de los componentes conectados de la imagen hasta este punto. Para ello se establece una iteración entre todos los componentes encontrados, almacenando la distancia euclidiana entre un componente y todos los demás.



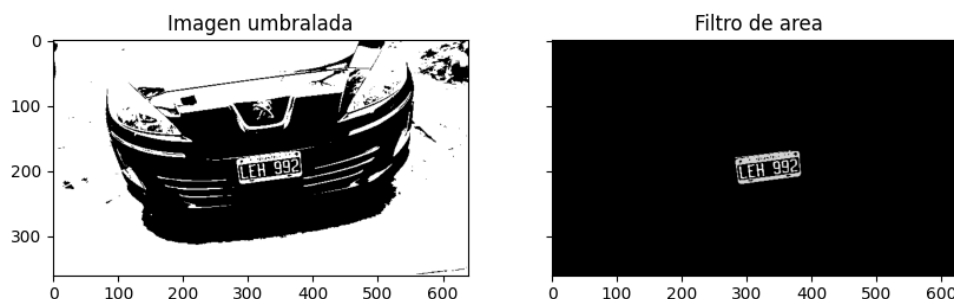
Cuando obtenemos almacenados todas las distancias entre elementos, almacenamos en una copia los elementos que cumplen con el criterio de que la distancia entre el elemento a incluir y los demás sea menor a un umbral establecido manualmente para cada imagen.

Y de esta manera, solo nos quedamos con los 6 caracteres de cada patente en un mascara:

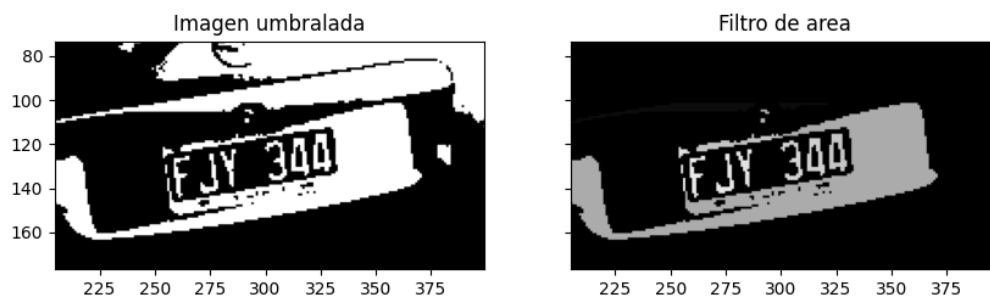


Para la segmentación de la forma de la patente como tal, el enfoque es el mismo, pero cambiando los umbrales de cada criterio. Se umbrales la imagen con un valor específico para cada imagen que intente visualizar la forma de la patente como tal. Para ello, nuevamente se trabajan filtro por área y relación de aspecto sobre los componentes conectados de cada imagen.

En varias ocasiones, se puede segmentar la patente completa del auto con facilidad:



En otras, es más complejo de segmentar la patente de la imagen ya que no se encuentran umbrales que permitan separar a la figura de la patente respecto del fondo del auto:



Esto se debe principalmente a que las condiciones de luminosidad, ángulos y formas de la patente y auto tienen escala de valores de grises tan similares que no permite encontrar un umbral que los separe perfectamente, sin perder otra información útil en la imagen.

En otras ocasiones, de hecho no se puede segmentar la patente de ninguna manera con este método de umbrales de valores. Por ejemplo, la la imagen 12 tiene una patente que posee un color demasiado similar al fondo del auto, por lo que no se ha podido segmentar la patente, aunque sí sus caracteres.



Para la segmentación de las patentes este método de umbrales no es eficiente para resolver la totalidad del problema. Lo mejor sería probar metodologías y herramientas más avanzadas con transformaciones que permitan relevar y segmentar los detalles de la imagen.

## **Conclusiones Generales**

La implementación para resolver el **Problema 1** demuestra un enfoque efectivo en la segmentación y clasificación de monedas y dados en una imagen. Los aspectos fueron trabajados incluyen:

### **Detección de Monedas:**

- Utilizamos transformadas de Hough para identificar círculos en la imagen.
- Clasificamos monedas según sus radios y coloración adecuada en la imagen original.
- Se contó y etiquetó precisamente las monedas detectadas.

### **Detección de Dados:**

- Se utilizaron máscaras morfológicas para destacar áreas relevantes para los dados.
- Aplicamos transformadas de Hough para la detección de círculos en la máscara.
- Clasificamos eficientemente los dados según su área y marcado adecuado en la imagen.

### **Visualización y marcados de Resultados:**

- Generamos visualizaciones claras para mostrar las áreas de interés, círculos detectados y resultados finales con información clasificada.
- Incorporamos etiquetas informativas sobre el valor de las monedas y dados.

Como conclusión, se logró con éxito cumplir con la consigna, inclusive se logró forjar un camino para llegar al objetivo entre tantas herramientas, un detalle a mejorar podría depender de la calibración de parámetros (como por ejemplo umbrales con Canny y la transformada de Hough) para así aumentar la precisión de detección de los objetos dentro de la imagen, o incluso la automatización de los mismos.

La implementación para resolver el **Problema 2** se centra en el procesamiento de imágenes de patentes de automóviles. A continuación se presenta un resumen de los aspectos trabajados en la solución de la consigna:

#### **Configuración Paramétrica:**

- Se proporciona una configuración paramétrica para cada imagen dentro de la clase, en la que incluimos umbrales, área, aspecto relativo y distancia, lo que nos permite adaptarnos a variaciones en las imágenes.

#### **Umbralización y Filtrado:**

- Se utiliza la umbralización para resaltar áreas de interés en las imágenes. El filtrado basado en área y aspecto relativo se aplica para seleccionar regiones en las cuales puedan ser potencialmente las patentes.

#### **Conectividad y Centroides:**

- La función `connectedComponentsWithStats` la empleamos para etiquetar y obtener regiones conectadas.
- Se identifican centroides válidos según criterios de área y aspecto relativo.

#### **Filtrado por Distancia:**

- Se implementa un filtro adicional para quedarnos con los centroides cercanos, ayudando y enriqueciendo la precisión en la detección.

Al igual que en el punto 1 se concluye que luego de varios intentos en el que se modificaron varios parámetros y herramientas de OpenCV utilizadas, se logró cumplir la consigna solicitada para la detección de caracteres, pero la solución fue parcial para la detección de las patentes.

La resolución de este problema ha sido de selección manual que requiere trabajo imagen por imagen para la selección de parámetros mediante prueba y error. La resolución de este ejercicio pudo haber sido desarrollada con otro enfoque que intenta automatizar el proceso de escalado, o trabajar con umbrales globales en principio, para luego intinar detalles en cada imagen.

Como se observa en las anteriores conclusiones se tocaron varios aspectos dados en los capítulos de la materia en los cuales se utilizaron herramientas y algoritmos dados como filtrados, detección de bordes, detección de componentes, morfología, segmentación y demás haciendo uso de librerías como OpenCV y Numpy para realizar la mayor parte de este trabajo.