

Facultad de Ciencias Exactas, Ingeniería y Agrimensura

Tecnicatura Universitaria en Inteligencia Artificial



## PROCESAMIENTO DE IMÁGENES

**IA 4.4**

## **Informe N°2**

**Año:** 2025

**Grupo:** 1

### **Integrantes:**

- Calabozo, Nicolas - C-7551/5
- Perez, Sebastián - P-5334/1
- Lapolla, Martín Facundo - L-3372/3

**Fecha de presentación:** 01/12/2025

### **Docentes:**

- Ing. Gonzalo Sad
- Ing. Juan Manuel Calle
- Ing. Joaquín Allione

# Introducción

El presente informe corresponde a la resolución del Trabajo Práctico N°2, cuyo propósito es aplicar conceptos de segmentación y detección de objetos mediante el desarrollo de dos problemas enfocados en la extracción automática de información en imágenes.

En el Problema 1, se aborda el desafío de clasificar y cuantificar objetos sobre un fondo de intensidad no uniforme. El objetivo es elaborar un algoritmo capaz de segmentar automáticamente monedas y dados, permitiendo no solo clasificar los tipos de monedas presentes, sino también interpretar y sumar los valores numéricos de las caras de los dados detectados.

Por otro lado, el Problema 2 se plantea como una aplicación de reconocimiento vehicular sobre un conjunto de imágenes de automóviles. En este caso, se combinan técnicas de localización y procesamiento morfológico para detectar la región de la placa patente y, posteriormente, lograr la segmentación individual de los caracteres que la componen.

Ambos desarrollos se implementaron utilizando la biblioteca OpenCV, junto con NumPy y Matplotlib, asegurando la visualización de los resultados en cada etapa del procesamiento.

## Problema 1 - Segmentación y Clasificación de Monedas y Dados

Para dar solución al problema de detección y clasificación de monedas y dados en una imagen con fondo de intensidad no uniforme, se diseñó un algoritmo secuencial estructurado en seis etapas de procesamiento:

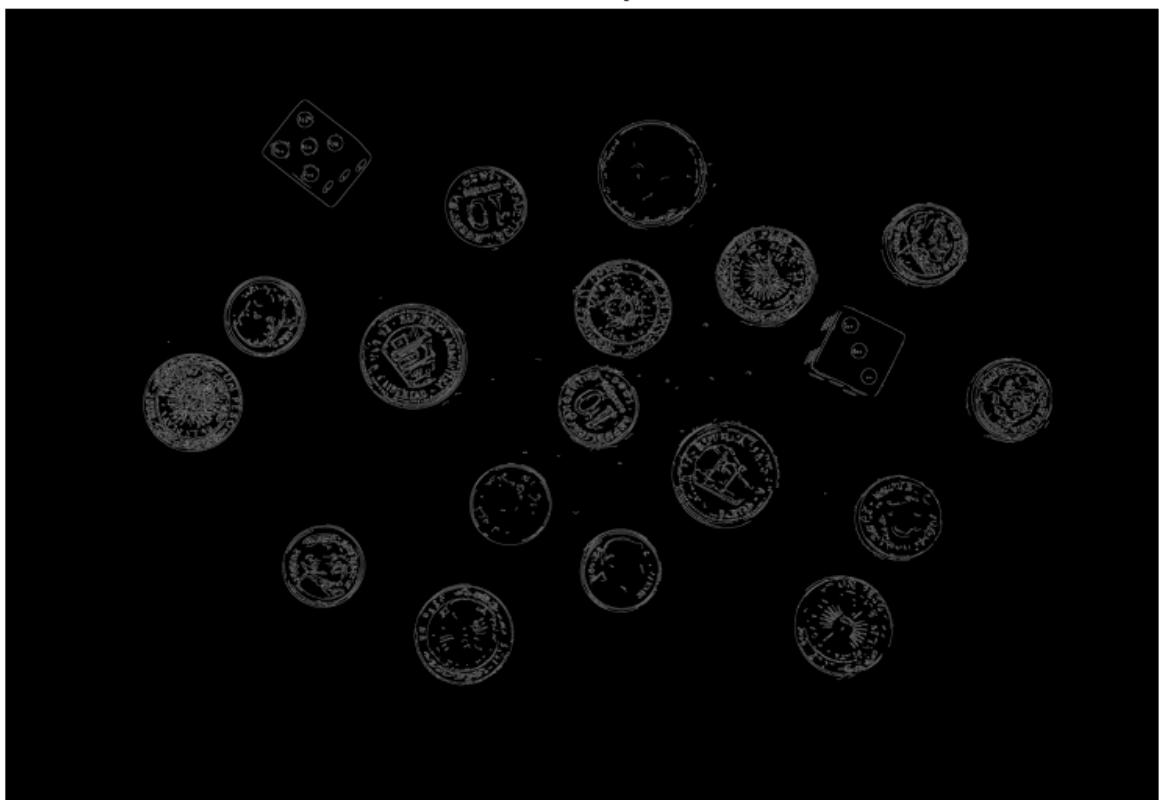
### Etapa 1: Preprocesamiento

Dada la naturaleza del fondo, se optó por trabajar en el dominio de los bordes. Inicialmente, se convirtió la imagen a escala de grises y se aplicó un filtrado para suavizar el ruido.



Posteriormente, se utilizó el algoritmo de detección de bordes Canny, el cual es robusto frente a los cambios graduales de iluminación del fondo, permitiendo resaltar únicamente las transiciones de alta frecuencia correspondientes a los límites entre los objetos y el fondo.

1. Canny

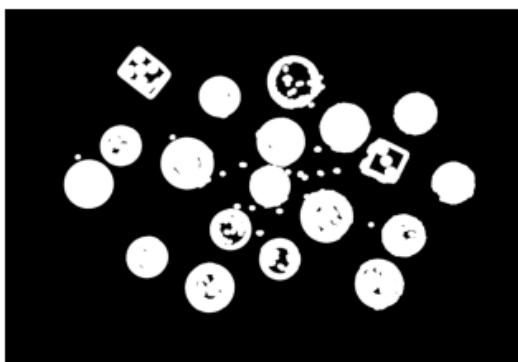


## Etapa 2: Morfología - Tratamiento de imagen para segmentar contornos

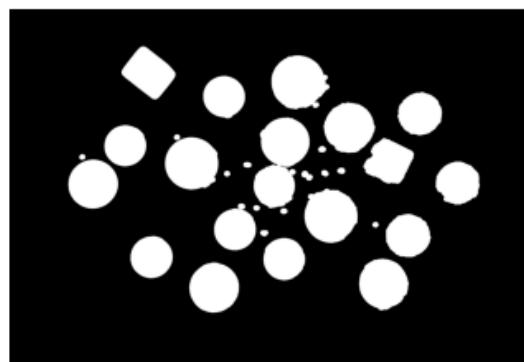
La salida del detector Canny suele presentar discontinuidades o líneas finas que no forman regiones cerradas. Para solucionar esto, se aplicaron operaciones morfológicas (dilatación, relleno y erosión) con el fin de conectar los trazos y consolidar los bordes. Esto nos da un primer acercamiento para que, al buscar contornos, se obtengan figuras cerradas que representan lo más fielmente posible la geometría de los objetos presentes.

Comenzando por la dilatación, se tuvo que utilizar un kernel lo suficientemente grande de 21x21, seleccionado de manera experimental, para que los bordes detectados por Canny engloben la totalidad de la geometría de los objetos. Una vez dilatada la imagen y asegurado el cierre de los contornos, se aplicó una función para el relleno de agujeros, obteniendo así una máscara binaria sólida representativa de cada objeto.

2. Dilate (21x21)

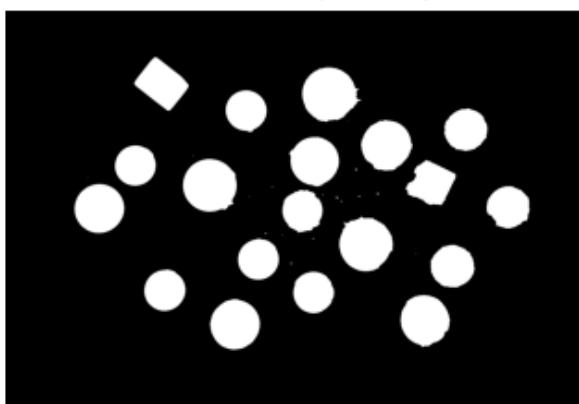


3. Relleno

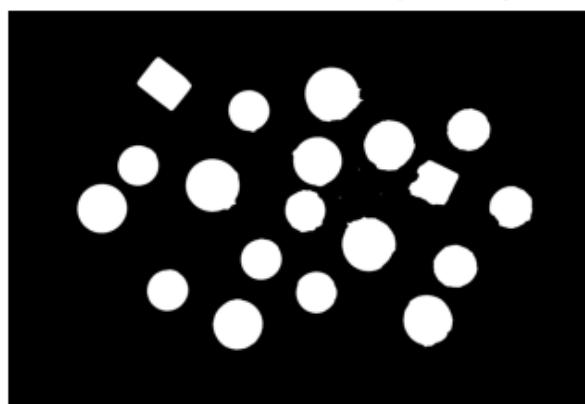


Posteriormente, se realizó una erosión con el mismo tamaño de kernel (21x21) utilizado en la dilatación. El propósito de esta operación inversa es contrarrestar el aumento de área provocado por la dilatación inicial, ajustando así los bordes de las máscaras lo más cerca posible al tamaño real de los objetos representados.

4. Erosion (21x21)



5. Filtro de Mediana (13x13)



Finalmente, se utilizó un filtro de mediana con un kernel de 13x13. Esta etapa final tiene una doble función: eliminar pequeñas zonas de ruido (sal y pimienta) generados por las operaciones morfológicas y, simultáneamente, suavizar los bordes de las máscaras resultantes para facilitar la posterior clasificación por forma.

### **Etapa 3: Clasificación Detección de Dados y Monedas**

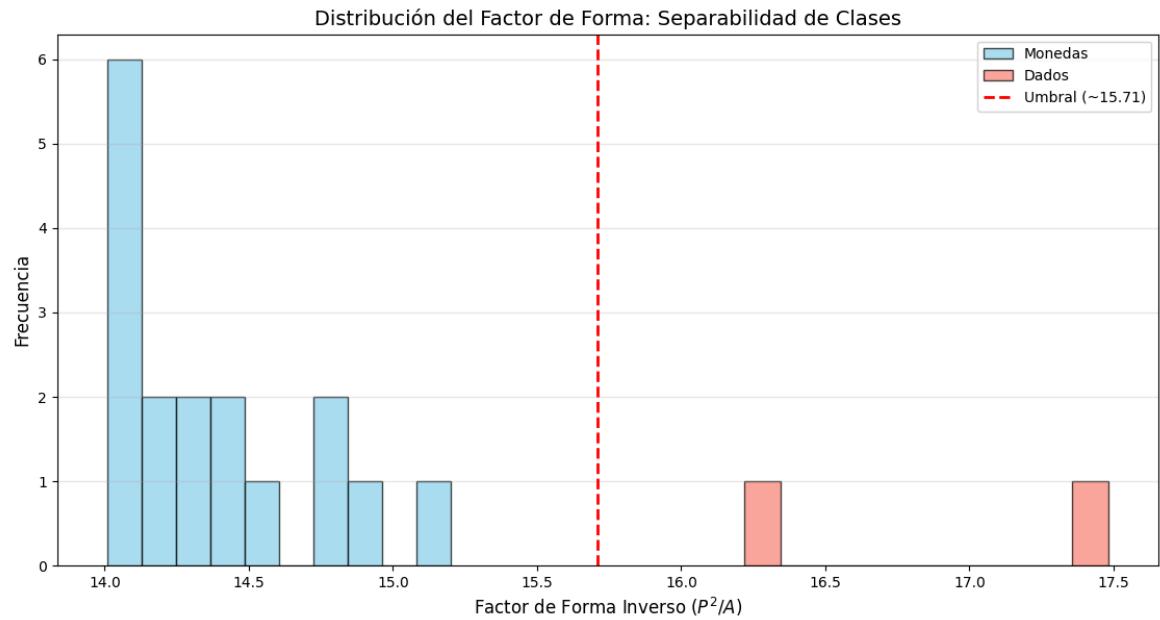
Utilizando la máscara obtenida en la Etapa 2, encontramos los contornos de cada objeto en la imagen. Una vez obtenidos los contornos candidatos, se procedió a diferenciar entre "dados" y "monedas". Para ello se utilizó el Factor de Forma Inverso, definido mediante la relación entre el área y el perímetro del objeto. Este descriptor es invariante a la escala y rotación, permitiendo separar los objetos en dos grupos: aquellos con geometría cuadrada (dados) y aquellos con geometría circular (monedas).

ETAPA 3: Detección - 17 Monedas - 2 Dados - Factor de Forma: 15.5



La imagen obtenida por nuestro procesamiento hasta el momento presenta bordes, alrededor de los objetos, que no son regulares. Si bien la forma general es correcta, estas irregularidades impactan significativamente el cálculo del factor de forma inverso ya que aumentan artificialmente el perímetro de cada objeto observado. Nuestra máscara obtenida de la etapa dos, sin embargo, es lo suficientemente regular como para poder discriminar entre dados y monedas.

Para ello decidimos establecer un umbral a partir del cual podremos realizar esta clasificación binaria.



Un umbral de decisión posible puede ser el promedio entre el factor de forma máximo de las monedas y el mínimo de los dados.

## Etapa 4: Normalización de Contornos de Monedas

Debido a la presencia de bordes irregulares en las monedas, el área calculada directamente del contorno puede tener variaciones. Como nuestro objetivo es discriminar el tipo de moneda por tamaño, en esta etapa se normalizan las monedas detectadas, ajustando su forma a una elipse de área mínima a partir de los contornos. Esto permite obtener una medida de radio y área más consistente a la hora de clasificar los tipos de moneda. Podemos visualizar el resultado de dicho ajuste en la siguiente imagen.

ETAPA 4 - Normalización - Elipses Ajustadas al Contorno Original



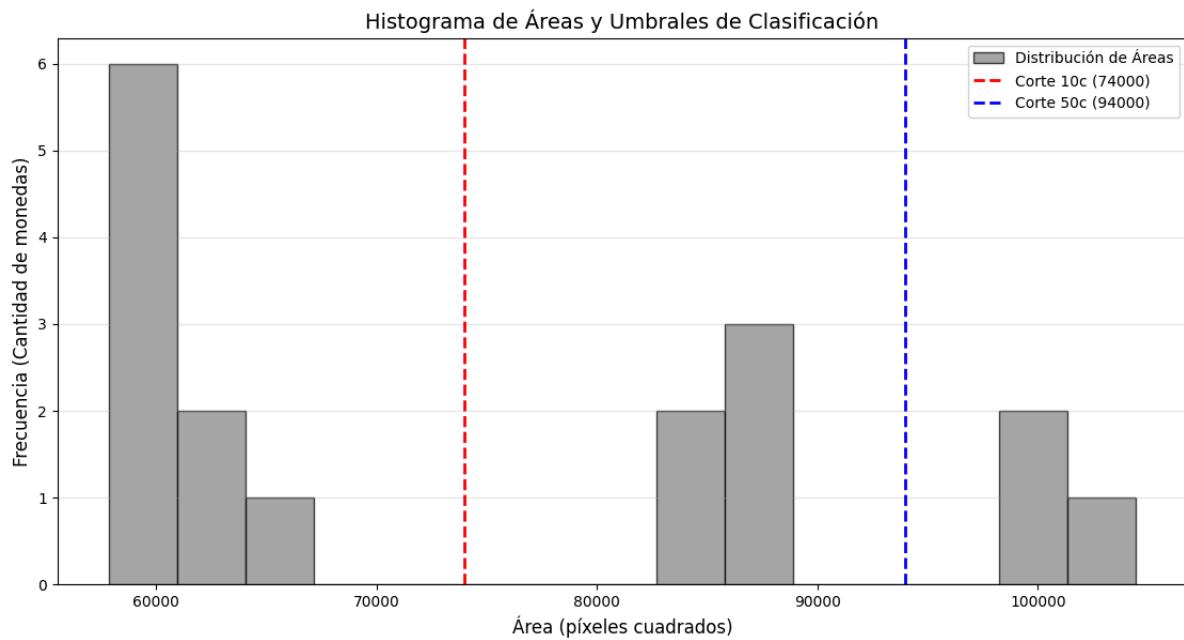
## Etapa 5: Clasificación de Monedas por Área

Con los contornos normalizados, se realiza la clasificación del valor de la moneda. Dado que el tamaño físico de las monedas es el discriminante principal, se establecieron umbrales de área basados en el radio calculado, agrupando las detecciones en sus respectivas denominaciones según su tamaño relativo en la imagen.

ETAPA 5 - Clasificación por Tamaño (Area de la Elipse)



Para visualizar el criterio de corte, utilizamos el siguiente histograma



Podemos observar claramente en la distribución de área de las monedas, los criterios de corte para la clasificación. Tomamos un área máxima de 74000 píxeles cuadrados para clasificar las monedas de diez centavos, un área mínima de 94000 píxeles cuadrados para las monedas de cincuenta centavos y, por descarte, obtuvimos las monedas de un peso.

## **Etapa 6: Clasificación de Dados por Análisis de Componentes Internos**

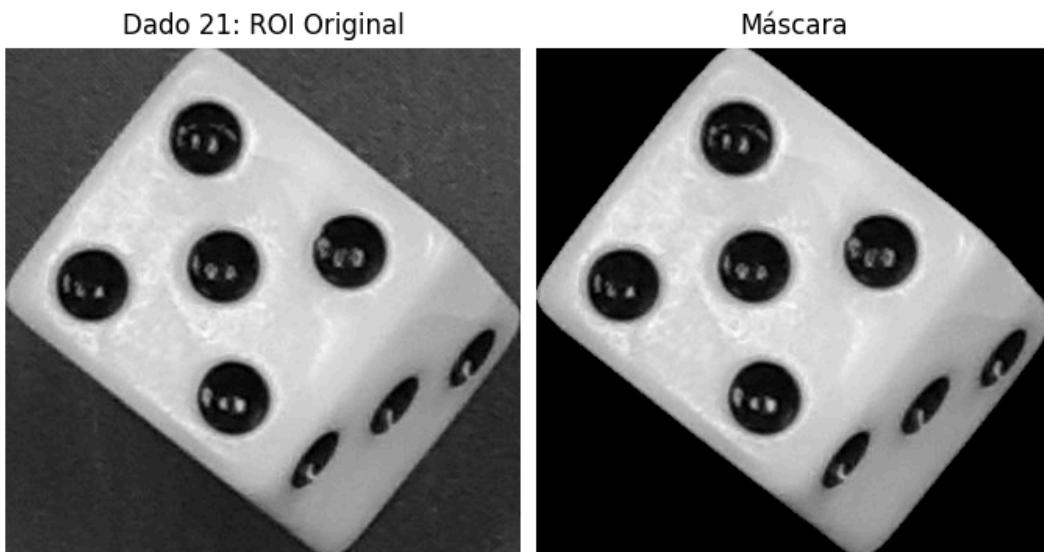
Finalmente, para determinar el valor de los dados, se analizaron los componentes internos de los contornos clasificados como cuadrados. Se utilizó nuevamente un criterio de forma para validar que los elementos internos correspondan a los puntos del dado y no a ruido o reflejos. Una vez validados los puntos mediante su circularidad, se procedió al conteo de los mismos para asignar la etiqueta numérica correspondiente a cada dado.

### **Extracción de ROI y Enmascaramiento**

Para aislar el problema y evitar interferencias del entorno, se extrajo la Región de Interés correspondiente al bounding box del dado detectado. Sobre esta imagen se aplicó la máscara binaria generada en la Etapa 2. Esto permite apagar el fondo, asegurando que el análisis de intensidad se realice exclusivamente sobre la superficie del dado

### **Segmentación de Candidatos**

Se realizó una umbralización sobre la ROI para detectar las regiones oscuras que contrastan con la superficie clara del dado. En esta fase, el algoritmo detecta todos los componentes conexos oscuros, lo cual incluye tanto los puntos legítimos como no deseados, tales como sombras proyectadas por el dado sobre el fondo

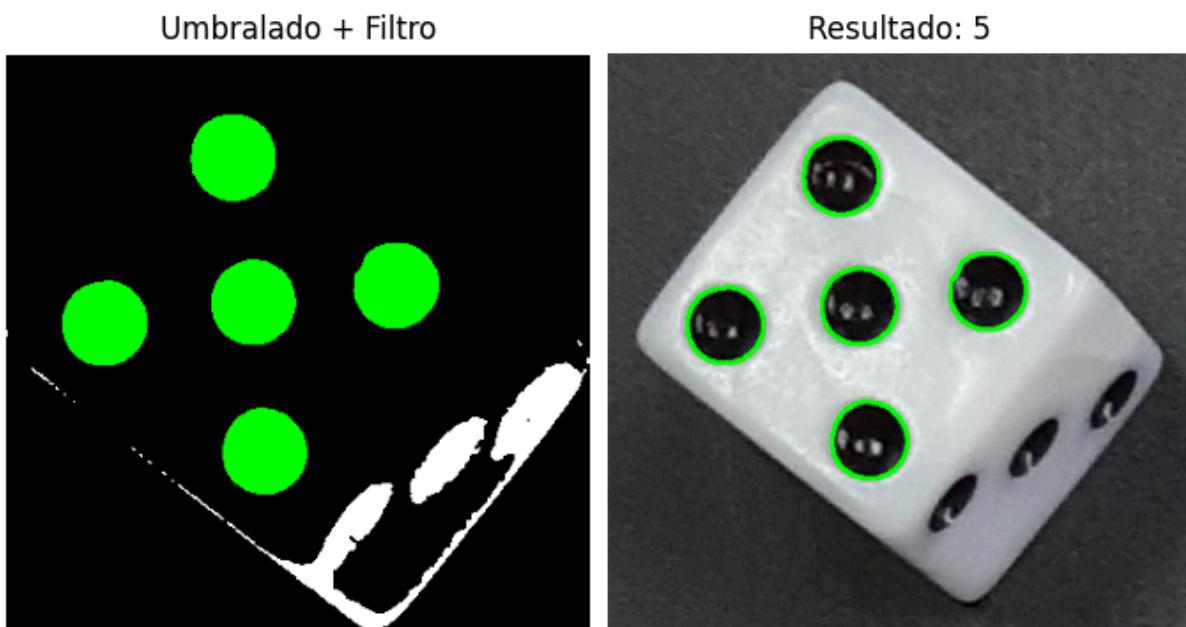


## Filtrado por Factor de Forma

Para discriminar entre los puntos del dado y el ruido, se reutilizó el descriptor de Factor de Forma Inverso.

Dado que los puntos de los dados son circulares por diseño, podemos utilizar la misma técnica para contabilizar los puntos reales, eliminando así los elementos no deseados que no se conformen al factor de forma circular.

Se aplicó un filtro que valida únicamente aquellos contornos cuya circularidad se encuentra dentro de un rango de tolerancia elegido. Esta vez, el umbral se seleccionó de manera informada.



## Conteo y Asignación de Etiqueta

Finalmente, el algoritmo contabiliza la cantidad de componentes que superaron la etapa de filtrado. Este número entero se asigna directamente como la etiqueta de clase del objeto. En el ejemplo ilustrado, se validaron 5 componentes, clasificando la cara con un valor de '5'.

## ETAPA 6 - Clasificación de Dados por Factor de Forma

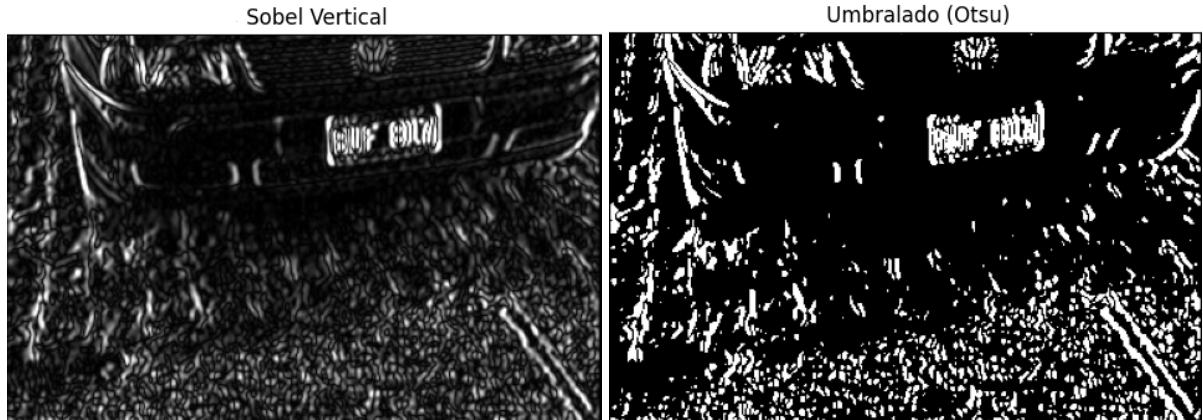


En esta imagen podemos observar la correcta clasificación de ambas caras de los dados, con los contornos resaltados y los valores asociados a sus caras.

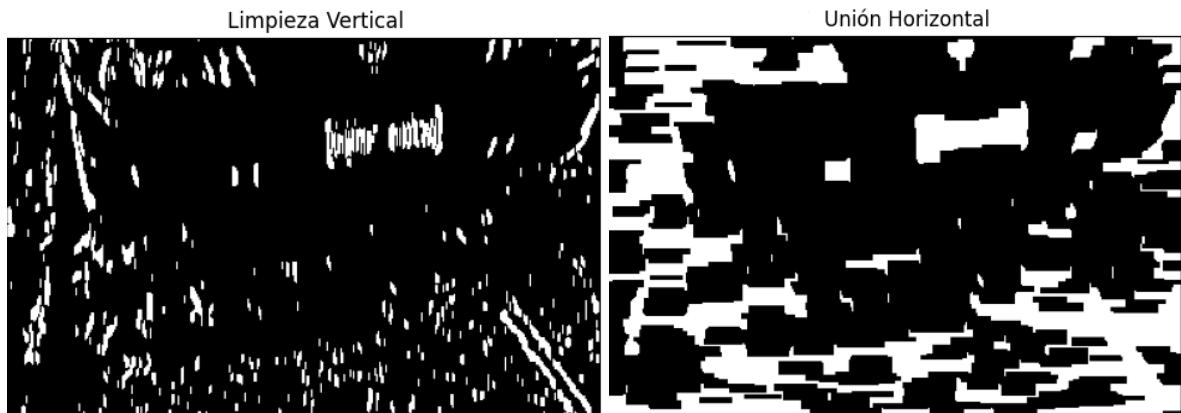
## Problema 2 - Detección de patentes

El objetivo de este problema consiste en el desarrollo de un algoritmo capaz de detectar y segmentar automáticamente la placa patente en una serie de imágenes de autos y luego segmentar los caracteres.

Para optimizar el procesamiento y reducir la aparición de falsos positivos en áreas irrelevantes (como el cielo, árboles o parabrisas), definimos un recorte automático de la imagen (ROI). Descartamos el 25% superior, el 5% inferior y un 20% en los márgenes laterales, centrando el análisis exclusivamente en la zona donde tiende a ubicarse la patente. Luego de esto convertimos la región de interés a escala de grises y aplicamos una mejora de contraste local (CLAHE) para resaltar los caracteres incluso en condiciones de sombra. Posteriormente, suavizamos la imagen con un filtro Gaussiano y aplicamos Sobel vertical para detectar los bordes, debido a que los caracteres de la patente presentan una componente vertical que contrasta con el fondo de la chapa. El resultado se binarizó mediante Otsu.



Para distinguir los caracteres de la patente de otros elementos como el suelo o parrillas, diseñamos una secuencia de operaciones morfológicas a llevar a cabo. Primero aplicamos una operación de apertura con un elemento estructural rectangular vertical (1x5), esto actuó como un filtro de tamaño y forma, eliminando el ruido granular pequeño (como la textura del piso) pero conservando los trazos verticales de los caracteres. A continuación, utilizamos una clausura con un kernel horizontal (13x3), esto permitió fusionar los caracteres individuales en un único objeto sólido sin conectar la patente con elementos superiores o inferiores como el paragolpes.



Sobre los objetos resultantes de la morfología, extrajimos los contornos y aplicamos un filtrado geométrico estricto. Primero se descartaron objetos verticales y ruidos pequeños. Luego utilizamos un rectángulo rotado, ‘*minAreaRect*’, para calcular con precisión las dimensiones y orientación, conservando sólo aquellos contornos con una relación de aspecto entre 2.0 y 4.5, y descartando aquellos con una inclinación mayor a 45°. También incorporamos un filtro de solidez (*extent*) para descartar contornos con mucho espacio vacío en su interior.

Finalmente, para asegurar que el candidato seleccionado fuera efectivamente una patente y no un bloque sólido (como una óptica o un espejo), implementamos una validación de textura basada en Canny. Analizamos la densidad de bordes en tres líneas de corte horizontales a distintas alturas del candidato, una patente válida debe presentar una cantidad consistente de transiciones (cortes) en estas líneas debido a la alternancia de los caracteres. Como resultado de esto el algoritmo logró aislar exitosamente la región de la patente en la totalidad de las 12 imágenes.

5. Detección



6. Recorte Final



Una vez obtenida la región de la patente en el apartado anterior, el siguiente paso fue aislar individualmente cada uno de los caracteres (letras y números) que la componen.

Para lograr la segmentación, implementamos una función que procesa el recorte de la patente basándonos en la premisa de que los caracteres son regiones oscuras (o claras, dependiendo de la patente) con una geometría y disposición espacial específica.

Inicialmente, descartamos un pequeño margen perimetral de la imagen recortada (10% superior/inferior y 2% lateral) para eliminar bordes o restos del marco de la patente. Luego, convertimos la imagen a escala de grises y aplicamos un filtro bilateral, el cual suaviza el ruido manteniendo los bordes de los caracteres definidos.

1. Imagen Gris



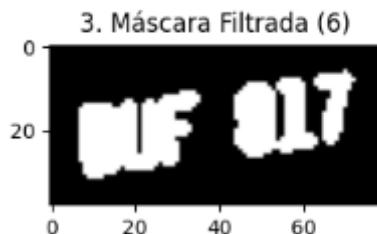
Para separar los caracteres del fondo, utilizamos un binarización adaptativa (Gaussian). A diferencia de un umbral global, este método calcula el umbral para cada píxel basándose en sus vecinos, lo que resulta muy robusto frente a las variaciones de iluminación (sombras o brillos) típicas en las patentes vehiculares.



Sobre la imagen binarizada, extrajimos todos los contornos presentes. Sin embargo, muchos de estos contornos corresponden a ruido. Para identificar cuáles corresponden a caracteres, aplicamos filtros basados en propiedades geométricas esperadas:

- Área mínima: Se descartaron contornos muy pequeños (ruido).
- Altura relativa: Un carácter debe ocupar una porción significativa de la altura de la patente (al menos el 20%).
- Relación de aspecto: Se filtraron contornos demasiado anchos o demasiado delgados que no corresponden a la proporción de un carácter alfanumérico.

A pesar del filtrado individual, algunos elementos no deseados podían persistir. Para resolver esto, implementamos otra función, ‘filtrar\_por\_agrupacion’. Esta asume que los caracteres de una patente están alineados horizontalmente y tienen alturas similares. Calculamos la mediana de la altura y de la posición vertical de todos los candidatos. Luego, descartamos aquellos que se desviaban significativamente de estos valores medios, logrando así conservar únicamente los caracteres presentes en la placa patente.



Como resultado el algoritmo demostró ser eficaz para aislar los caracteres en buena parte de las imágenes, generando recortes individuales de cada letra y número.

img01.png			B	U	F	8	1	7
img02.png			G	Q	7	9		
img03.png			M	H	0	1		
img04.png			L	H	E	9	6	
img05.png			F	J	Y	3		
img06.png			L	E	H	9	4	
img07.png			L	O	D	1	2	
img08.png			J	M	O	3		
img09.png			F	F	D	7	6	5
img10.png			D	O	O	3	22	
img11.png			G	M	D			
img12.png			B	X		8		

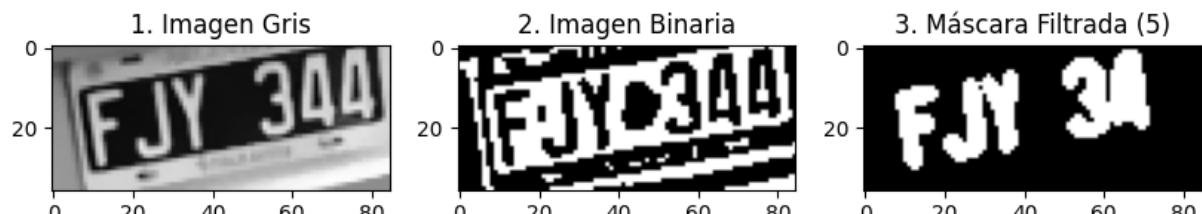
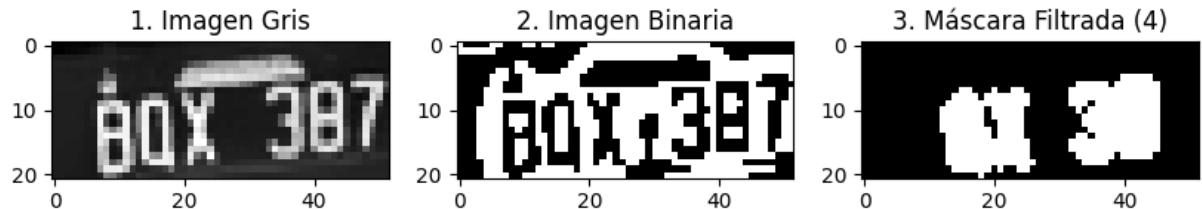
## Conclusión

En el Problema 1, la principal dificultad fue la no uniformidad del fondo, lo que nos hizo descartar el uso de un umbralado global simple. Comprobamos que trabajar en el dominio de los bordes (Canny) seguido de una etapa morfológica robusta (dilatación y relleno de agujeros) fue la estrategia más eficaz que encontramos para obtener máscaras sólidas. Un punto importante de este desarrollo fue la validación práctica de los descriptores geométricos: el Factor de Forma Inverso demostró ser una herramienta discriminante muy buena para separar geometrías cuadradas (dados) de circulares (monedas) de manera invariante a la rotación, permitiendo posteriormente una clasificación basada puramente en áreas normalizadas y conteo de componentes internos.

Por otro lado, el Problema 2 presentó el mayor desafío técnico al enfrentarnos a imágenes del mundo real con iluminación variable y ruido. Nuestro enfoque inicial, que intentaba rectificaciones geométricas complejas para rotar las patentes, resultó ser propenso a errores de recorte y computacionalmente costoso. Esto nos llevó a replantear la estrategia hacia otro preprocesamiento: la implementación de CLAHE (mejora de contraste local) resultó vital para resaltar los caracteres en patentes oscurecidas por sombras .

La segmentación de caracteres individuales exigió una combinación creativa de morfología y heurística. Aprendimos que el uso de una erosión con kernel vertical es una técnica muy útil para separar caracteres que se tocan lateralmente sin perder su estructura . Además, la incorporación de conocimiento sobre la relación de aspecto nos permitió implementar una lógica de "división forzada" (*splitting*), corrigiendo automáticamente los casos donde la binarización fusionaba dos letras en un solo bloque.

Hay un amplio rango de mejora para el algoritmo y la implementación de nuestra segmentación de caracteres ya que hay máscaras binarias que, a simple vista, sería sencillo encontrar sus contornos.



El problema de optimizar al mínimo detalle los filtros que nos permiten obtener estas máscaras junto a la detección de caracteres, es que si modificamos algún valor, por ejemplo, de área mínima o tolerancia para determinar los espacios entre letras es que ganamos caracteres para algunas patentes y, a su vez, generamos ruido para otras. A modo de propuesta para la mejora, podría utilizarse otro enfoque en vez de la segmentación por filtrado de posibles candidatos (detecciones de posibles caracteres), como un análisis de componentes conectados.

En definitiva, este trabajo nos permitió integrar las etapas de segmentación con la extracción de características , evidenciando que una buena elección de descriptores geométricos simplifica drásticamente la lógica de clasificación posterior.