FACULTAD DE CIENCIAS EXACTAS, INGENIERIA Y AGRIMENSURA
UNIVERSIDAD NACIONAL DE ROSARIO

TECNICATURA UNIVERSITARIA EN INTELIGENCIA ARTIFICIAL

PROCESAMIENTO DE IMÁGENES

INFORME

TRABAJO PRÁCTICO N° 2

INTEGRANTES

-JUAN MORALES
M-7459/4
-GENARO CANCIANI
C-7449/7
-MALENA RUPPEN
R-4751/1

PROFESORES: GONZALO SAD, JUAN MANUEL CALLE, JOAQUÍN ALLIONE.

Ejercicio 1

El objetivo del **Problema 1** es desarrollar un algoritmo capaz de procesar la imagen monedas.jpg , la cual contiene **dados** y **monedas** sobre un fondo no uniforme. El algoritmo debe resolver tres tareas principales de forma automática:

- A. Segmentación: Separar las monedas y los dados del fondo.
- B. Clasificación y Conteo de Monedas: Distinguir los diferentes tipos de monedas y contarlas.
- C. Conteo y Determinación del Valor del Dado: Identificar cada dado y determinar el valor (número de puntos) de su cara superior.

Análisis y Técnicas de Resolución

1. Detección y Clasificación de Monedas

Problemas Enfrentados

La principal dificultad en la detección de monedas radica en el **fondo de intensidad no uniforme** y las **variaciones de brillo** y **sombras** que pueden afectar la detección de bordes y la forma circular.

Técnicas Utilizadas: Transformada de Hough para Círculos

La detección se realiza mediante la **Transformada de Hough para Círculos (cv2.HoughCircles)**. Esta técnica es robusta para encontrar formas circulares incluso con datos ruidosos o incompletos.

- **Preprocesamiento:** Se aplica un **Filtro de Mediana (cv2.medianBlur, kernel 9)** a la imagen en escala de grises. Este desenfoque es crucial para reducir el ruido sin afectar significativamente los bordes de los objetos circulares, lo que mejora el rendimiento de la transformada de Hough.
- **Parámetros de Hough:** Se ajustan los parámetros para optimizar la detección:
    - minDist=100: Asegura que los centros de los círculos estén separados por al menos 100 píxeles, evitando múltiples detecciones para una misma moneda.
    - param1=150, param2=40: Son umbrales que controlan la sensibilidad. param2 se ajustó para ser menos estricto (40), capturando la mayoría de las monedas.
    - minRadius=80, maxRadius=250: Limitan el rango de búsqueda a los tamaños esperados de las monedas.

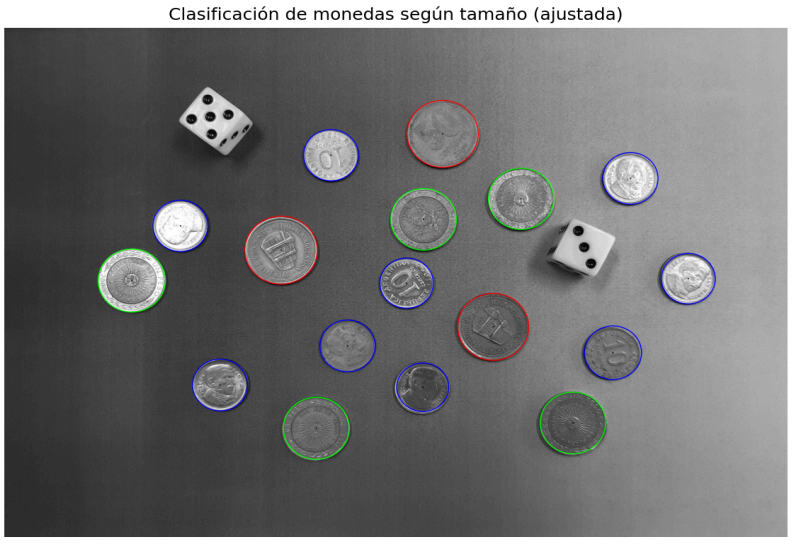
En la imagen de abajo se ve el resultado de aplicar el filtro de mediana a la imagen original.

Clasificación por Tamaño (Radio)

Una vez detectadas las monedas, la clasificación se realiza utilizando el **radio (r)** devuelto por la transformada de Hough. Se definen umbrales de radio fijos para agrupar las monedas en categorías:

- **Pequeñas ($0,10):** r < 145
- **Medianas ($1):** 145 <= r < 170
- **Grandes ($0,50):** r >= 170

La visualización final utiliza diferentes colores para cada clase (Azul: Pequeñas, Verde: Medianas, Rojo: Grandes), confirmando la segmentación y el conteo.



2. Detección y Conteo de Dados

Problemas Enfrentados

Los dados presentan varios desafíos:

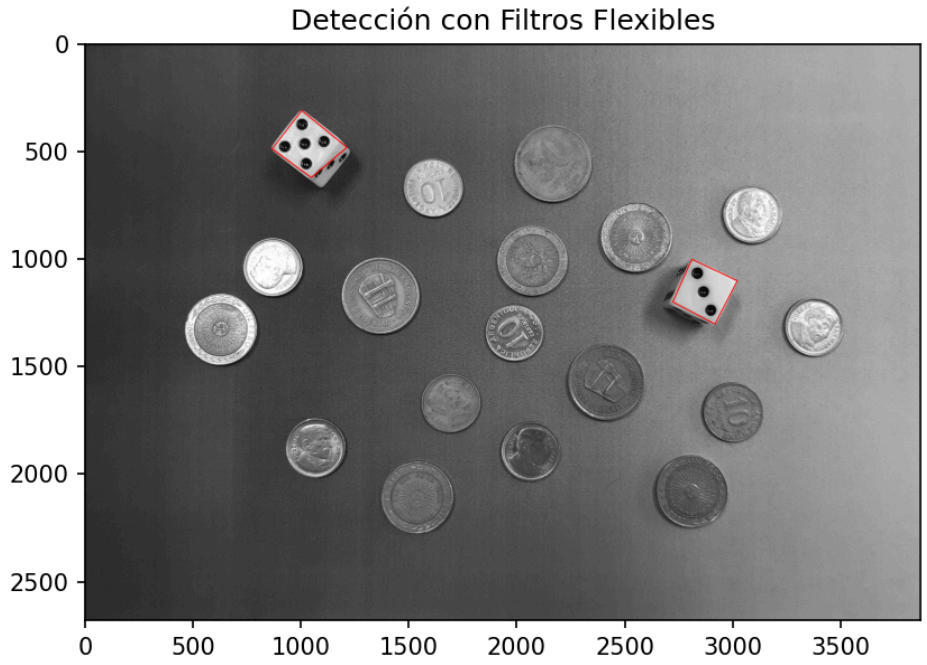
1. **Forma (Cuadrado/Cubo):** La forma es más variable debido a la perspectiva.
2. **Variación de Intensidad:** El cuerpo del dado es blanco, pero el fondo es gris no uniforme.
3. **Detección de Puntos (Valor):** Los puntos son pequeños y oscuros dentro del dado.

Técnicas de Detección (Segmentación)

La segmentación del cuerpo de los dados se realiza a través de un enfoque basado en **Umbralización y Contornos**:

1. **Umbralización (cv2.threshold):** Se aplica un umbral binario simple (thresh_value} = 170) a la imagen en escala de grises. Dado que los dados son objetos claros sobre un fondo oscuro-medio, este umbral aísla sus cuerpos blancos.

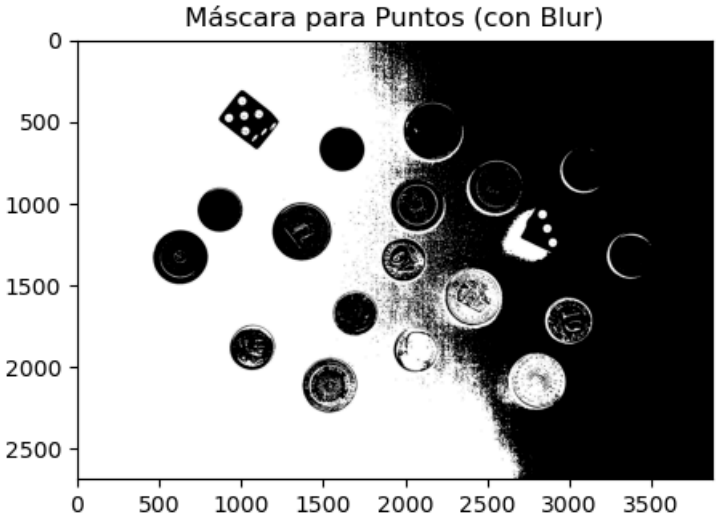
2. **Búsqueda de Contornos (cv2.findContours):** Se buscan los contornos externos (cv2.RETR_EXTERNAL) en la imagen umbralizada.
3. **Filtrado de Contornos (Clasificación por Forma):** Se aplican tres filtros geométricos para identificar únicamente los dados:
    - **Área Mínima:** Área > 35000 para eliminar ruido y objetos más pequeños que un dado.
    - **Aproximación Poligonal (cv2.approxPolyDP):** Se usa para simplificar el contorno y determinar el número de vértices. Se busca que el número de vértices sea **4** (cuadrilátero).
    - **Relación de Aspecto (Aspect Ratio):** Se calcula como w/h del *bounding box*. Un dado debe ser aproximadamente un cuadrado, por lo que se exige que 0.7 <= Aspect Ratio <= 1.3, lo que da flexibilidad a la distorsión por perspectiva.



Técnicas de Conteo de Puntos (Determinación del Valor)

Para determinar el valor de la cara superior de cada dado, se sigue un procedimiento enfocado en los puntos oscuros:

1. **Máscara de Puntos:** Se crea una nueva máscara usando **Umbralización Binaria Inversa** (thresh_value = 100) en la imagen de gris original. Esto aísla los puntos oscuros de los dados.
2. **Desenfoque:** Se aplica un pequeño **Filtro de Mediana (cv2.medianBlur, kernel 5)** a la máscara de puntos para mejorar la circularidad.



3. **Localización de Puntos (ROI):** Para cada contorno previamente identificado como dado, se define una **Región de Interés (ROI)**. Solo se procesa la sección de la máscara de puntos que corresponde a esa ROI.
4. **Detección de Círculos en el ROI:** Se aplica la **Transformada de Hough para Círculos** dentro de cada ROI de puntos. Se usan parámetros más estrictos (param2=15, minDist=10) y radios pequeños (minRadius=5, maxRadius=30) para detectar únicamente los puntos.
5. **Conteo:** El número de puntos detectados por cv2.HoughCircles en cada ROI corresponde al valor del dado.

Conclusiones

El algoritmo desarrollado demuestra una alta **robustez** al manejar los desafíos de la imagen, en particular el fondo no uniforme y las variaciones de iluminación.

- **Monedas:** La **Transformada de Hough para Círculos**, combinada con un preprocesamiento de desenfoque de mediana, fue altamente efectiva para la segmentación. La clasificación por umbralización del radio permitió distinguir las tres clases de monedas con éxito.
- **Dados:** La combinación de **Umbralización** (para aislar el cuerpo del dado) y el **Filtrado Geométrico de Contornos** (basado en área, número de vértices y relación de aspecto) segmentó los dados de manera confiable. El uso selectivo de la **Transformada de Hough** dentro de la ROI de cada dado fue clave para contar correctamente los puntos y determinar su valor, resolviendo la parte más compleja del ejercicio.

Ejercicio 2

El objetivo del Problema 2 es desarrollar un algoritmo capaz de procesar un conjunto de 12 imágenes (img01.png a img12.png) de vehículos, con el objetivo de localizar automáticamente la placa patente y segmentar sus caracteres. El algoritmo debe resolver dos tareas principales:

A. Detección y Segmentación de la Placa: Localizar la región de la imagen que contiene la patente.

B. Segmentación de Caracteres: Identificar y delimitar individualmente los caracteres alfanuméricos dentro de la placa.

Análisis y Técnicas de Resolución

Para resolver este problema, se implementó una estrategia de detección de abajo hacia arriba. En lugar de intentar detectar primero el recuadro de la patente (que puede tener algunos desperfectos), el algoritmo busca primero patrones que parecen caracteres y luego los agrupa para reconstruir la ubicación de la patente.

1. Preprocesamiento y Detección de Candidatos (Caracteres)

Problemas Enfrentados:

Las imágenes presentan variaciones significativas en la iluminación, el color del vehículo y la perspectiva. Además, existen elementos ruidosos como rejillas, luces y patrones en el suelo que pueden generar falsos positivos al buscar bordes.

Técnicas Utilizadas: Umbralización de Otsu y Análisis de Contornos

- **Binarización:** Se carga la imagen en escala de grises y se aplica una **Umbralización de Otsu** (cv2.threshold + cv2.THRESH_OTSU). Este método calcula automáticamente el umbral óptimo para separar el texto (oscuro) del fondo de la patente (claro).

- **Filtrado Inicial de Candidatos:** Se buscan los contornos (cv2.findContours) y se realiza una primera limpieza basada en propiedades geométricas individuales. Se

conservan sólo aquellos contornos que cumplen con las características físicas de un carácter de patente:

- **Relación de Aspecto (Aspect Ratio):** Calculada como w/h. Se filtran contornos que no sean verticales (1.5 < AR < 3.0).
- **Área:** Se eliminan contornos demasiado pequeños (ruido) o demasiado grandes (partes del auto), aceptando áreas en el rango 30 < área < 500.

2. Agrupación Lógica y Localización de la Patente

Problemas Enfrentados

El paso anterior detecta letras, pero también detecta rejillas, faros y ruido que pasaron el filtro de tamaño. El desafío es distinguir qué conjunto de contornos forma realmente una patente.

Ese fenómeno se observa en la siguiente imagen, donde se ve la detección de ruido incluso luego del filtrado:

Técnicas de Resolución: Algoritmo de Agrupación Heurística

Se desarrolló una función específica (filtrar_por_agrupacion) que ordena los candidatos espacialmente (de izquierda a derecha) y evalúa su relación con los vecinos. Para que un candidato se considere parte de un grupo de patente, debe cumplir tres condiciones heurísticas respecto al carácter anterior:

1. **Similitud de Altura:** Los caracteres deben tener tamaños similares.
2. **Alineación Vertical:** Los centros de los caracteres deben estar alineados en el eje Y.
3. **Proximidad Horizontal:** Los caracteres no pueden estar demasiado separados ni superpuestos excesivamente.

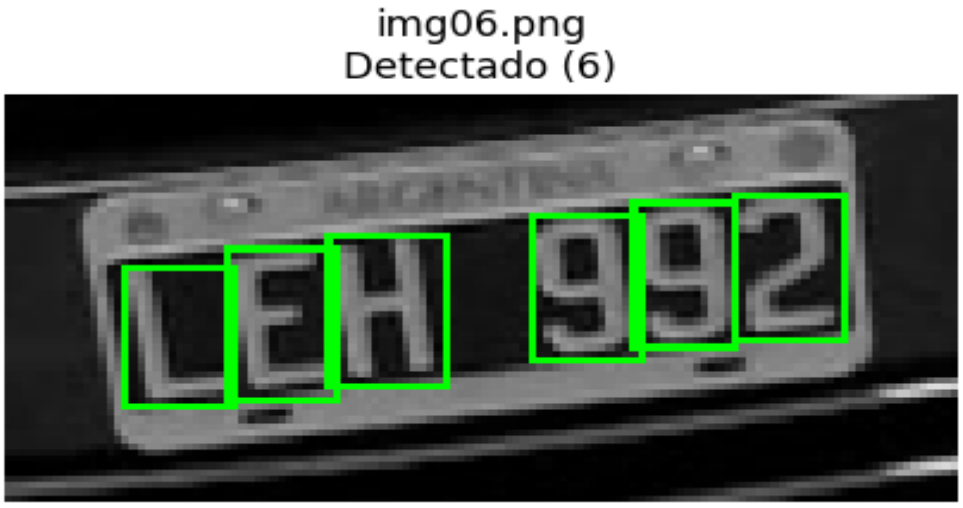
Selección del Grupo Ganador:

El algoritmo genera varios grupos potenciales y selecciona el "mejor grupo" basándose en la cantidad de elementos (se priorizan grupos de 6 o más caracteres) y el tamaño de los mismos.

3. Extracción y Visualización (Puntos A y B)
Una vez identificado el grupo de caracteres correcto, se resuelven simultáneamente las consignas A y B:

- **Segmentación de Caracteres (Punto B):** Las coordenadas de los contornos del grupo ganador constituyen la segmentación de los caracteres.
- **Detección de la Placa (Punto A):** Se calculan los límites extremos (min_x, min_y, max_x, max_y) de todo el grupo. Se añade un margen de seguridad (*padding* de 15 píxeles) y se realiza un recorte (*crop*) sobre la imagen original. Esto aísla la patente completa del resto del vehículo.

Ejemplo de imagen de placa extraída:

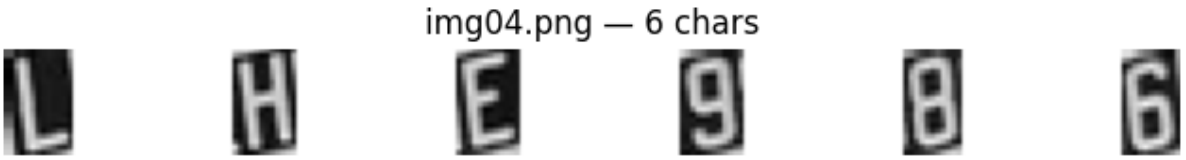


4. Procesamiento Individual y Visualización Final: Una vez aislado el recorte de la patente, se procede a separar sus elementos constitutivos para su análisis individual:

- **Ordenamiento Espacial (Secuencia Lógica):** Dado que los algoritmos de contornos no garantizan un orden específico, se reorganiza la lista de coordenadas basándose en el eje horizontal (X). Esto asegura que los caracteres se procesan y visualizan en el orden de lectura correcto (de izquierda a derecha), reconstruyendo la secuencia alfanumérica de la patente.
- **Extracción de ROI en Escala de Grises:** Se realiza un recorte (*slicing*) para cada carácter utilizando las coordenadas ordenadas. A diferencia de las etapas previas de detección, este recorte se aplica sobre la imagen en **escala de grises** (`recorte_gris`) y no sobre la máscara binaria. Esto permite preservar la información de intensidad, texturas y sombras originales de cada letra, descartando el ruido de binarización para la visualización final.
- **Visualización Desagregada:** Se genera una composición dinámica mediante *Matplotlib*, donde cada carácter segmentado se muestra en una sub-celda

independiente. Esto permite verificar visualmente la calidad de la segmentación individual, asegurando que cada recorte contenga un único carácter centrado y legible.

El resultado final en una de las patentes es el que se ve debajo, logrando segmentar los caracteres correctamente:



Conclusiones

La implementación basada en la agrupación de componentes demostró ser superior a la detección de bordes rectangulares para este dataset.

- La **Umbralización de Otsu** fue fundamental para manejar los cambios de luz sin necesidad de ajustar parámetros manuales imagen por imagen.
- El **Filtrado Heurístico** permitió descartar eficazmente el ruido estructurado (como bloques de piedra en el suelo) que suele tener geometrías similares a las letras, asegurando que solo se extraiga la región donde existe una alineación coherente de texto de las patentes (3 dígitos, un espacio, y 3 dígitos)
- El sistema logró detectar la mayoría de las patentes y segmentar caracteres incluso en imágenes con baja resolución o desenfoque, validando la robustez del enfoque algorítmico.