



Facultad de Ciencias Exactas, Ingeniería y Agrimensura  
Universidad Nacional de Rosario

Procesamiento de Imágenes  
TUIA  
Trabajo Práctico No. 2  
25/11/2024

Giuliano Crenna (C-7438/1)  
Bruno E. Pace (P-5295/7)  
Mariano Sancho (S-5778/9)

## Contenidos

<b>1 Abstract</b>	<b>2</b>
<b>2 Introducción</b>	<b>2</b>
<b>3 Problema 1: Identificación de monedas y datos</b>	<b>3</b>
3.1 Descripción . . . . .	3
3.2 Solución . . . . .	3
3.2.1 Carga de la imagen . . . . .	3
3.2.2 Preprocesamiento de la imagen . . . . .	4
3.2.3 Detección de monedas . . . . .	4
3.2.4 Clasificación de monedas . . . . .	5
3.2.5 Detección de datos . . . . .	5
3.2.6 Conteo de círculos en datos . . . . .	6
3.3 Resultados obtenidos . . . . .	7
3.4 Resultado Final . . . . .	7
<b>4 Problema 2: Detección de letras en patentes de automóviles</b>	<b>8</b>
4.1 Descripción . . . . .	8
4.2 Solución . . . . .	8
4.2.1 Carga de la imagen . . . . .	8
4.2.2 Preprocesamiento de la imagen . . . . .	8
4.2.3 Aplicación de transformación morfológica . . . . .	8
4.2.4 Umbralización . . . . .	9
4.2.5 Detección de componentes conectadas . . . . .	9
4.2.6 Filtrado de componentes . . . . .	10
4.2.7 Resultados finales . . . . .	11
4.3 Casos especiales . . . . .	12
4.3.1 Letras con separación insuficiente . . . . .	12
4.3.2 Letras unidas al borde de la patente . . . . .	12
<b>5 Conclusión Final</b>	<b>13</b>

## 1 Abstract

Este trabajo se centra en la implementación y análisis de técnicas de procesamiento de imágenes mediante herramientas computacionales avanzadas. Se abordan problemas prácticos que combinan conceptos teóricos y aplicaciones prácticas en Python, utilizando librerías especializadas. Los ejercicios están diseñados para reforzar habilidades en preprocesamiento, transformación y análisis de imágenes, con un enfoque en la resolución de desafíos reales. Los resultados obtenidos buscan evidenciar la efectividad de los métodos implementados, así como fomentar la comprensión de los conceptos fundamentales en el campo del procesamiento digital de imágenes.

## 2 Introducción

El procesamiento digital de imágenes es una disciplina esencial en la informática moderna, con aplicaciones que abarcan desde la medicina hasta la ingeniería y las ciencias sociales. Este trabajo práctico introduce a los estudiantes al manejo y análisis de imágenes digitales, utilizando herramientas y técnicas computacionales. A lo largo del documento, se desarrollan problemas que exploran temas como la transformación de imágenes, la mejora de la calidad visual y la extracción de características significativas. El objetivo principal es proporcionar una experiencia de aprendizaje práctica que permita a los estudiantes aplicar conceptos teóricos en un entorno computacional, mientras desarrollan habilidades clave en programación y análisis de datos visuales.

### 3 Problema 1: Identificación de monedas y dados

#### 3.1 Descripción

El objetivo principal del Problema 1 es procesar una imagen para identificar y clasificar dos tipos de objetos: monedas y dados. El problema se aborda mediante el uso de técnicas avanzadas de procesamiento de imágenes implementadas en Python utilizando la biblioteca OpenCV.

#### 3.2 Solución

A continuación, se describe el procedimiento detallado:

##### 3.2.1 Carga de la imagen

La imagen es cargada desde un directorio utilizando la función `img.reading`. La imagen original, mostrada en la Figura 1, se encuentra en formato BGR.

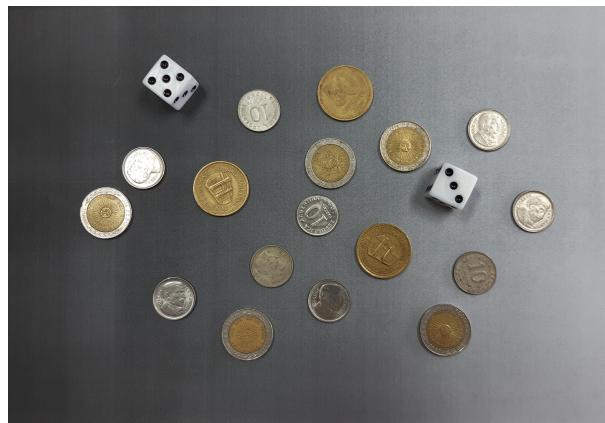
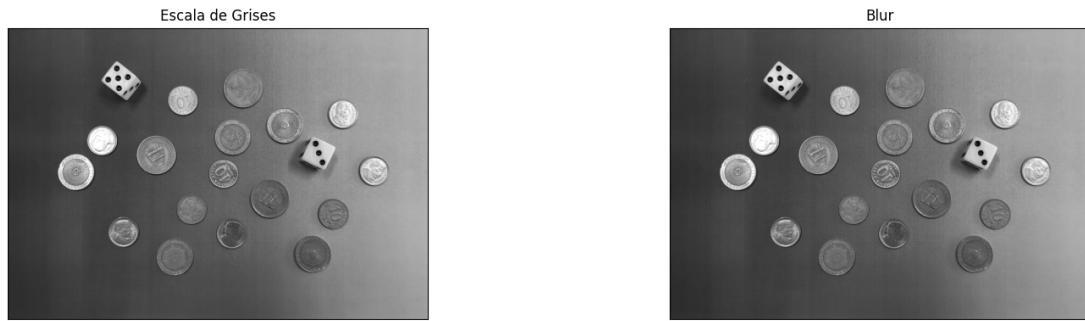


Figure 1: Imagen original.

### 3.2.2 Preprocesamiento de la imagen

La imagen se transforma a escala de grises mediante la función `img.preprocessing`, como se muestra en la Figura 2a. Luego, se aplica un filtro Blur Gaussiano (Figura 2b) para reducir el ruido y eliminar detalles irrelevantes.



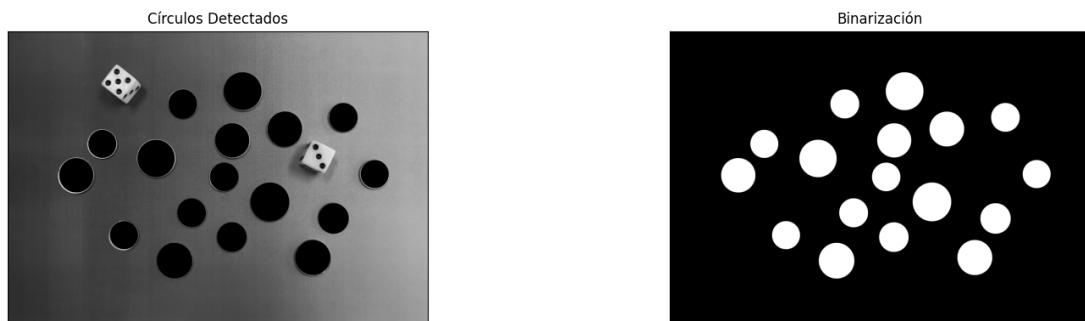
(a) Escala de grises.

(b) Blur aplicado.

Figure 2: Preprocesamiento de la imagen.

### 3.2.3 Detección de monedas

La función `detect_coins` utiliza HoughCircles para detectar círculos que representan las monedas. Los círculos detectados se muestran en la Figura 3a. Luego, se colorean de negro para facilitar la binarización (Figura 3b).



(a) Círculos detectados en la imagen.

(b) Imagen binarizada.

Figure 3: Detección y binarización de monedas.

### 3.2.4 Clasificación de monedas

Las monedas se clasifican según su área utilizando la función `coin_classification`. Las áreas clasificadas y sus valores están representadas con bounding boxes de colores en la Figura 4:

- Azul: \$0.50.
- Verde: \$0.10.
- Rojo: \$1.

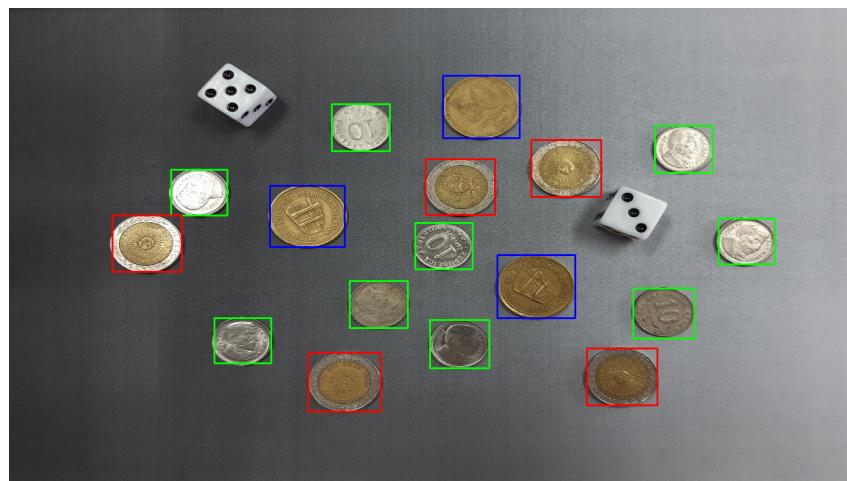


Figure 4: Monedas clasificadas con sus valores.

### 3.2.5 Detección de dados

La función `procesar_imagen_para_deteccion` transforma la imagen para detectar formas cuadradas (dados). A continuación, los dados se detectan basándose en el factor de forma, como se muestra en la Figura 5a y 5b.



(a) Detección de bordes con Canny.

(b) Contornos detectados.

Figure 5: Detección de dados.

### 3.2.6 Conteo de círculos en dados

Para cada dado detectado, se realizó un preprocesamiento adicional, que incluyó:

- Conversión de la región de interés (ROI) a escala de grises.
- Aplicación de un filtro Blur Gaussiano para suavizar detalles (Figura 6).
- Uso del algoritmo de HoughCircles para identificar círculos en las caras superiores.
  - Se utilizó un `dp=1.3` para ajustar la resolución de detección.
  - Los valores de `param1=10` y `param2=22` optimizaron la sensibilidad.
  - `minRadius=1` y `maxRadius=30` limitaron los tamaños detectados a círculos compatibles con los puntos de los dados.

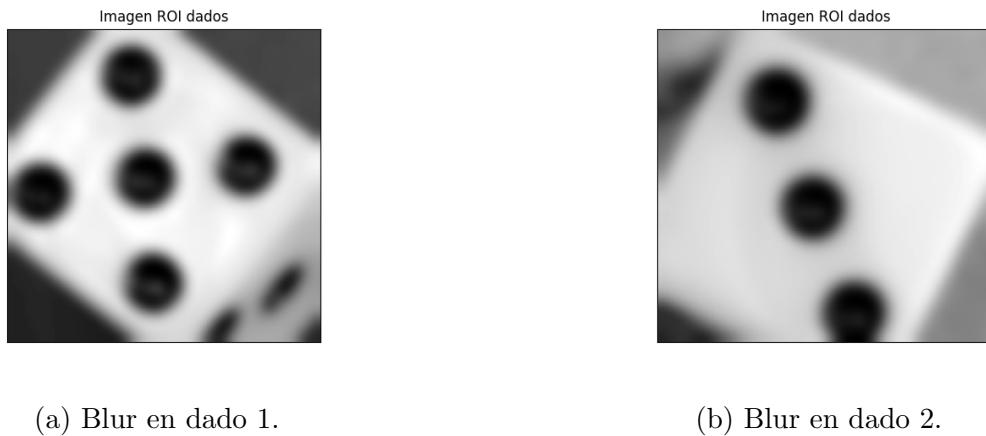


Figure 6: Imagen de dados preprocesados para conteo de círculos.

La Figura 7 muestra los círculos detectados en los dados tras la aplicación de Hough-Circles, donde cada círculo se representa con un contorno en verde.

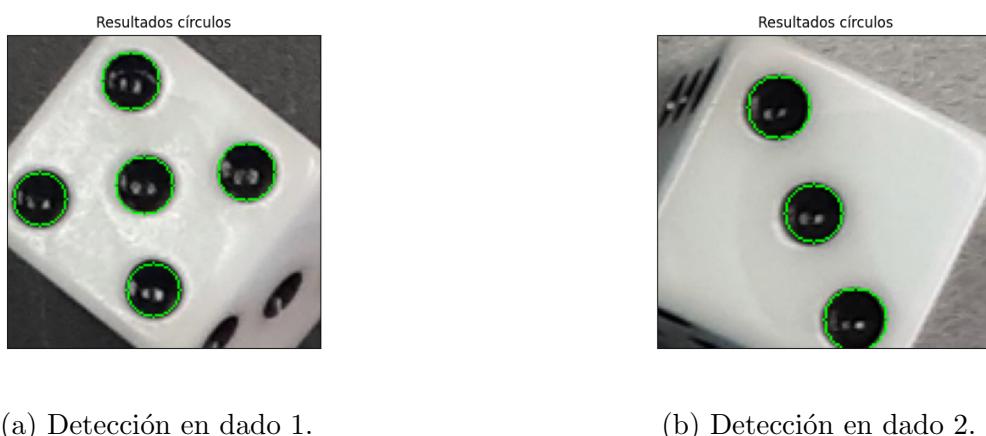


Figure 7: Círculos detectados en los dados.

Finalmente, los puntos detectados se sumaron para calcular la puntuación de cada dado, etiquetándolos directamente en la imagen final.

### 3.3 Resultados obtenidos

- Monedas:
  - Cantidad de monedas de \$1: 5.
  - Cantidad de monedas de \$0.50: 3.
  - Cantidad de monedas de \$0.10: 9.
- Dados:
  - Cantidad de dados detectados: 2.
  - Puntaje total: 8.

### 3.4 Resultado Final

Para finalizar la solución se creó una imagen con todos las detecciones. Tanto de dados como de monedas.



Figure 8: Imagen Final

## 4 Problema 2: Detección de letras en patentes de automóviles

### 4.1 Descripción

El objetivo del segundo problema es desarrollar un método para detectar y extraer las letras que forman parte de las patentes de automóviles en imágenes. Se cuenta con 12 imágenes de distintos automóviles, las cuales son procesadas mediante diversas técnicas de procesamiento de imágenes.

### 4.2 Solución

A continuación, se describen los pasos realizados para lograr la detección de las letras:

#### 4.2.1 Carga de la imagen

La imagen es cargada utilizando la función `img.reading`. Un ejemplo de una imagen original se muestra en la Figura 9.



Figure 9: Imagen original del automóvil.

#### 4.2.2 Preprocesamiento de la imagen

La imagen es transformada a escala de grises para simplificar el procesamiento. La Figura 10 muestra este resultado.

#### 4.2.3 Aplicación de transformación morfológica

Para resaltar las letras, se aplica una transformación morfológica Top-Hat utilizando un elemento estructural rectangular de  $7 \times 7$ . Este proceso elimina áreas oscuras amplias y resalta estructuras claras pequeñas, como las letras. La Figura 11 ilustra el resultado.



Figure 10: Imagen en escala de grises.



Figure 11: Resultado de la transformación Top-Hat.

#### 4.2.4 Umbralización

Posteriormente, la imagen es binarizada mediante un umbral fijo para obtener una imagen binaria. Este paso permite separar los objetos claros del fondo oscuro, como se muestra en la Figura 12.

#### 4.2.5 Detección de componentes conectadas

Para identificar todas las posibles letras, se emplea la función `connectedComponentsWithStats` de OpenCV. Este método analiza la imagen binarizada y encuentra regiones conectadas, generando estadísticas para cada componente. La Figura 13 muestra los objetos detectados inicialmente.



Figure 12: Imagen binarizada.



Figure 13: Componentes conectadas detectadas en la imagen binarizada.

#### 4.2.6 Filtrado de componentes

Se aplica un filtrado adicional para conservar únicamente las letras. Los criterios utilizados son:

- **Relación de aspecto:** Las letras tienen una relación de altura a ancho (*aspect ratio*) de aproximadamente 2:1.
- **Cercanía:** Las letras se encuentran alineadas, con coordenadas *y* similares y separaciones pequeñas en el eje *x*.
- **Área:** Se descartan componentes con áreas demasiado pequeñas o grandes.

#### 4.2.7 Resultados finales

Después del filtrado, las letras detectadas son resaltadas en la imagen final (Figura 14).



Figure 14: Resultado final: letras de la patente detectadas.



Figure 15: Resultado final: letras de la patente detectadas.

## 4.3 Casos especiales

Durante el procesamiento, se identificaron dos casos especiales que afectan la detección:

### 4.3.1 Letras con separación insuficiente

En algunas imágenes, la operación morfológica y la umbralización no separan adecuadamente las letras. Esto puede solucionarse ajustando el umbral o utilizando una operación morfológica de clausura (**MORPH\_CLOSE**) para llenar espacios entre las letras.

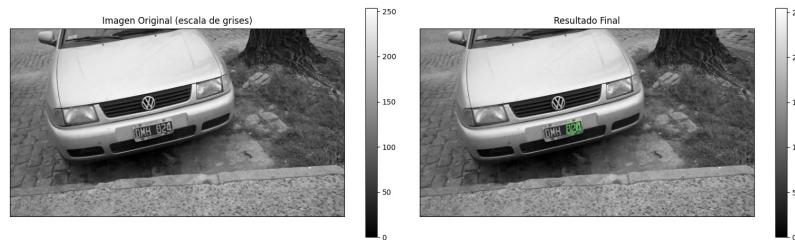


Figure 16: Caso 1.

### 4.3.2 Letras unidas al borde de la patente

En ciertos casos, las letras se fusionan con el borde de la patente, generando un objeto de mayor tamaño que no cumple con los criterios de filtrado. Esto puede corregirse aplicando una operación de apertura (**MORPH\_OPEN**) para separar las letras del borde.

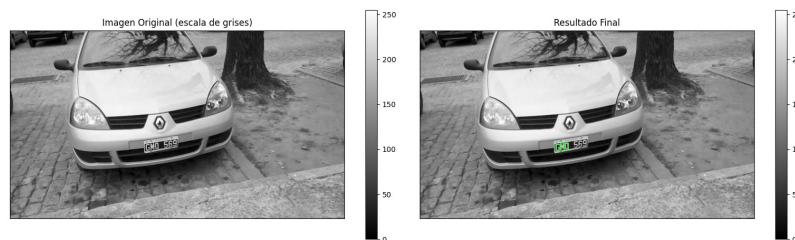


Figure 17: Caso 2.

## 5 Conclusión Final

El presente trabajo práctico ha demostrado la eficacia de las técnicas de procesamiento de imágenes para resolver problemas complejos y específicos, como la identificación de monedas y datos, y la detección de letras en patentes de automóviles.

En el **Problema 1**, se logró clasificar monedas según su valor y contabilizar los puntos de datos mediante una combinación de transformaciones geométricas, técnicas de segmentación (como HoughCircles) y análisis de componentes conectadas. La metodología implementada mostró robustez ante imágenes con características variadas, aunque algunos ajustes adicionales podrían ser necesarios para casos particulares, como cuando los objetos se encuentran superpuestos.

En el **Problema 2**, el uso de transformaciones morfológicas y el filtrado basado en relaciones de aspecto, área y cercanía permitió identificar de manera efectiva las letras presentes en las patentes. Sin embargo, se identificaron limitaciones en casos especiales, como letras fusionadas con bordes o con separación insuficiente, lo que plantea la necesidad de ajustes dinámicos en los parámetros o la incorporación de pasos adicionales en el pipeline de procesamiento.

A nivel general, se destacan los siguientes puntos clave:

- La importancia del preprocesamiento (como escalas de grises y operaciones morfológicas) para resaltar características de interés en las imágenes.
- El uso de técnicas de segmentación y análisis de componentes conectadas para clasificar y filtrar objetos de interés.
- La identificación de limitaciones y la propuesta de mejoras para enfrentar casos extremos, lo que refuerza la flexibilidad del enfoque adoptado.

En conclusión, este trabajo no solo permitió resolver los problemas planteados, sino que también evidenció la versatilidad y potencia de las herramientas de procesamiento digital de imágenes en aplicaciones prácticas. Los resultados obtenidos constituyen una base sólida para futuras implementaciones en contextos similares o más complejos.