



FACULTAD DE  
CIENCIAS EXACTAS,  
INGENIERIA Y AGRIMENSURA



---

## Tecnicatura Universitaria en Inteligencia Artificial

### Procesamiento de Imágenes

---

# Informe Trabajo Práctico 2

Integrantes:

- Herrera Morena - (H-1187/8)
- María Victoria Oviedo - (O-1820/1)
- Zorzolo Rubio Juana - (Z-1217/3)

2024

## ÍNDICE

Resumen.....	2
Metodología.....	3
Problema 1.....	3
Problema 2.....	3
Desarrollo/Implementación.....	4
Problema 1 - Detección y clasificación de monedas y datos.....	4
Problema 2 - Detección de patentes.....	5
Desafíos Técnicos/Problemas a los que nos enfrentamos.....	5
Problema 1.....	5
Problema 2.....	5
Resultados.....	6
Problema 1.....	6
Problema 2.....	6
Pruebas y Ejemplos de Ejecución.....	7
Problema 1.....	7
Problema 2.....	11
Conclusiones.....	13
Anexos.....	14

## Resumen

En este trabajo se desarrollaron algoritmos para la segmentación, clasificación y conteo de objetos en imágenes utilizando técnicas de procesamiento de imágenes. El objetivo principal del **Problema 1** fue identificar y clasificar monedas y datos presentes en una imagen con fondo de intensidad no uniforme. Para ello, se implementaron procesos de preprocesamiento, detección de bordes, operaciones morfológicas y clustering. Además, se calcularon los valores de las caras de los dados y las denominaciones de las monedas. El **Problema 2**, que abarca la detección de patentes vehiculares, está en desarrollo.

## Introducción

Nos enfrentamos a dos problemas:

- Segmentar y clasificar monedas y datos.
- Detectar y segmentar patentes vehiculares y sus caracteres.

El objetivo principal fue diseñar algoritmos eficientes que puedan procesar imágenes con condiciones de iluminación variadas y realizar análisis detallado de los objetos detectados.

## Metodología

### **Problema 1**

Se diseñó una serie de pasos de procesamiento compuesto por las siguientes etapas:

1. **Preprocesamiento:** Conversión de la imagen a escala de grises, reducción de ruido mediante desenfoque gaussiano y detección de bordes con el operador de Canny.
2. **Operaciones morfológicas:** Dilatación y erosión para limpiar bordes y mejorar la detección de contornos.
3. **Segmentación y clasificación:** Identificación de monedas y datos utilizando contornos y un análisis del factor de forma.
4. **Clustering:** Uso del algoritmo K-Means para clasificar las monedas por denominaciones.

### **Problema 2**

1. Se inicia con una imagen de entrada, la cual es transformada a escala de grises y opcionalmente ecualizada. Este paso asegura la uniformidad en la iluminación y facilita la segmentación posterior.
2. Se utiliza el algoritmo de Canny para detectar bordes en la imagen, ajustando los umbrales de manera iterativa ([ajustar\\_canny](#)).
3. El mejor ajuste se selecciona en función del número de contornos encontrados que cumplen con restricciones de área mínima y máxima.
4. Los contornos detectados son evaluados según:
  - **Área del contorno:** Sólo se seleccionan contornos dentro de un rango establecido (1500-35000 píxeles).
  - **Relación de aspecto (ancho/alto):** Los bounding boxes con proporciones similares a caracteres son considerados válidos.
5. Los bounding boxes seleccionados se procesan para:
  - Combinar regiones superpuestas mediante la métrica de Intersección sobre Unión (IoU).
  - Filtrar cajas cercanas y alineadas para garantizar que solo se incluyan caracteres relevantes.
6. Cada bounding box se recorta, se redimensiona y se pega en una nueva imagen con fondo blanco. Las letras se organizan horizontalmente con un espacio definido para simular una representación de la patente completa.

## Desarrollo/Implementación

### Problema 1 - Detección y clasificación de monedas y datos

1. **Segmentación de la imagen**
  - Se utilizó el método de Canny para detectar bordes ajustando parámetros [t1](#) y [t2](#).
  - Las operaciones morfológicas eliminaron ruido y cerraron regiones de bordes incompletos.
2. **Clasificación de objetos**
  - Los objetos fueron diferenciados entre monedas y datos basándose en su área y factor de forma.

- Se realizó un conteo automático de cada tipo de objeto, clasificándolos y marcándolos en la imagen.

### 3. Determinación de denominaciones y valores

- Para las monedas, se empleó K-Means con 3 clusters para clasificar las áreas en categorías de valor.
- En el caso de los dados, se implementó un algoritmo para contar los puntos de la cara superior.

## Problema 2 - Detección de patentes

El código se organiza en varias funciones principales para cada etapa del proceso:

1. `ajustar_canny`: Identifica los mejores umbrales para el algoritmo de Canny.
2. `procesar_patente`: Filtra contornos y extrae bounding boxes válidos.
3. `detectar_caracteres`: Detecta componentes conectados y bounding boxes con características geométricas específicas.
4. `combinar_bounding_boxes`: Combina cajas superpuestas según su IoU.
5. `seleccionar_cajas_cercanas`: Filtra cajas basándose en proximidad y alineación.
6. `extraer_y_mostrar_bboxes`: Recorta y guarda imágenes individuales de los caracteres detectados.
7. `pegar_y_guardar_patente`: Combina los caracteres recortados en una única imagen de salida.

## Desafíos Técnicos/Problemas a los que nos enfrentamos

### Problema 1

En este problema, el mayor desafío fue distinguir de manera eficiente las monedas de los dados. Esto se debe a que ambos tipos de objetos comparten características similares en términos de tamaño, lo que dificulta su segmentación y clasificación. Además, el fondo no uniforme de la imagen `monedas.jpg` añadió complejidad al proceso, ya que requería técnicas de preprocessamiento robustas para eliminar el ruido y resaltar los bordes relevantes. Fue crucial ajustar cuidadosamente los parámetros de detección de bordes mediante el operador Canny y realizar operaciones morfológicas para obtener contornos limpios.

### Problema 2

El principal desafío en este problema fue lograr una segmentación precisa de las patentes en imágenes de vehículos con características diversas y condiciones de iluminación variables. Las patentes no siempre están centradas, y pueden variar en iluminación y orientación. Implementar operaciones morfológicas robustas y ajustar el algoritmo de detección de bordes (Canny) fueron pasos clave para localizar las patentes.

## Resultados

### Problema 1

- **Conteo de objetos:**
  - Monedas detectadas: 17
  - Dados detectados: 2
- **Clasificación de monedas:**
  - Total estimado de dinero: 7.40
- **Datos:**
  - Valor total de las caras superiores: 8

La siguiente imagen muestra el resultado final del procesamiento:



### Problema 2

**Detección de patentes:** Las regiones que contienen las letras de la patente son correctamente detectadas en el 83% de los casos (con problemas en las imágenes 3 y 9).

**Extracción de Caracteres:** Los caracteres individuales son correctamente detectados en el 87,5% de los casos (con problemas en las imágenes 3 y 9, donde también tuvimos inconvenientes con las patentes).

**Imagen Final Consolidada:** La imagen final contiene los caracteres de la patente organizados horizontalmente con un espaciado uniforme.

## Pruebas y Ejemplos de Ejecución

### Problema 1

Se realizaron pruebas con el archivo [monedas.jpg](#), logrando segmentar y clasificar los objetos con alta precisión. Se muestran ejemplos de ejecución en las siguientes imágenes:

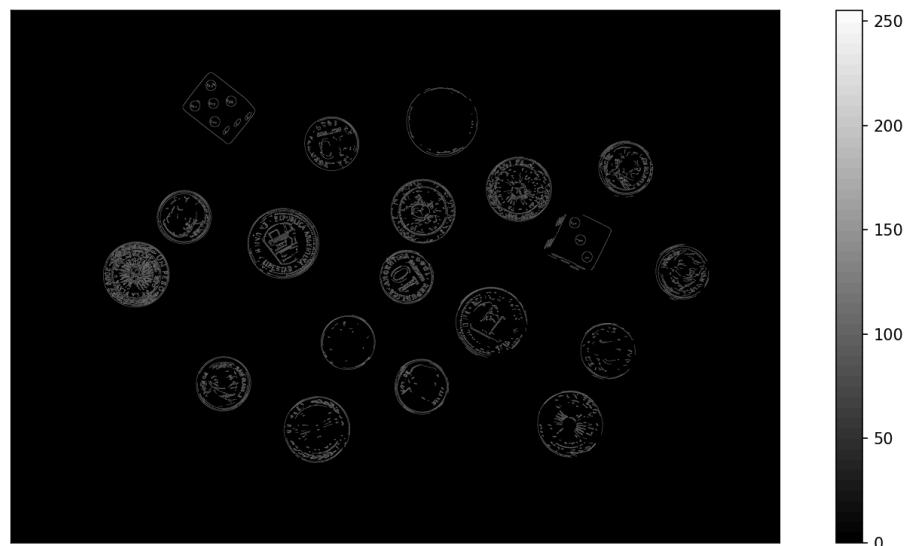
#### Imagen original:



1. Imagen en escala de grises (función [procesar\\_imagen\(\)](#)):



2. Desenfoque Gaussiano (función `procesar_imagen()`):



3. Identificación de dados y monedas (función `clasificar_tipo_objeto()`):

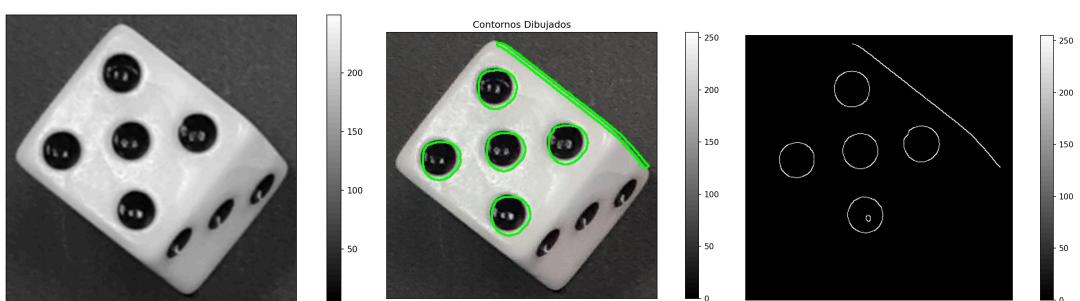


#### 4. Identificación del dado (función `procesar_imagen_dado()`):

*Para el dado con valor 3:*



*Para el dado con valor 5:*



*valor de los dados:*

```
>>> img_dado1 = cv2.imread('dados/DADO_10.jpg')
>>> nro_dado1 = procesar_imagen_dado(img_dado1, umbral_canny_1=100, umbral_canny_2=452)
Número del dado detectado: 3
>>> img_dado2 = cv2.imread('dados/DADO_18.jpg')
>>> nro_dado2 = procesar_imagen_dado(img_dado2, umbral_canny_1=280, umbral_canny_2=460)
Número del dado detectado: 5
>>> total = nro_dado1 + nro_dado2
>>> print("Total de los dados: ", total)
Total de los dados:  8
```

5. Identificación de monedas (función `detectar_tipo_monedas()`):

```
>>> img_resultado, total_dinero = detectar_tipo_monedas(img, areas_monedas, coordenadas_monedas)
>>> print(f"Total de dinero: {total_dinero} pesos")
Total de dinero: 7.399999999999995 pesos
```

6. Imagen final luego de correr el programa principal:



```
>>> print(f"Total monedas: {conteo_monedas}")
Total monedas: 17
>>> print(f"Total dados: {conteo_dados}")
Total dados: 2
```

## Problema 2

Se realizaron pruebas utilizando las imágenes ([img01-12.png](#)) contenidas en la carpeta patentes, logrando detectar la patente y sus caracteres y generar una imagen consolidada con los mismos. A continuación, se muestran los pasos del procesamiento y los resultados obtenidos para la imagen 8 ([img\\_o8.png](#)):

### Imagen original:



### 1. Imagen en escala de grises (función `procesar_patente()`):



2. Detección de bordes con Canny (función `ajustar_canny()`):



3. Detección de contornos (función `procesar_patente()`):



4. Recorte de la patente (función `encontrar_rectangulo_contenedor()`):



5. Extracción de caracteres (función `seleccionar_cajas_cercanas()`):



#### 6. Imagen final consolidada (función `pegar_y_guardar_patente()`):



### Conclusiones

El procesamiento de imágenes demostró ser una herramienta efectiva para abordar los problemas planteados en este trabajo. En el **Problema 1**, se implementaron técnicas como la detección de bordes con Canny, operaciones morfológicas y clustering, logrando clasificar y segmentar monedas y datos con alta precisión. Esto permitió calcular correctamente el valor total de las monedas y los datos, demostrando la utilidad de los algoritmos diseñados incluso en imágenes con fondos complejos y variabilidad en iluminación.

Para el **Problema 2**, se enfrentaron mayores desafíos debido a las condiciones variables de las imágenes, como iluminación no uniforme, orientación de las patentes y la presencia de ruido. A pesar de estas dificultades, los algoritmos desarrollados lograron una tasa de éxito del 83% en la detección de las patentes y del 87,5% en la extracción de caracteres. Esto evidencia la robustez del enfoque utilizado, aunque también destaca la necesidad de optimizar ciertos aspectos, **como la segmentación de caracteres en imágenes con condiciones adversas.**

En general, el trabajo realizado permitió cumplir con los objetivos planteados, resaltando la importancia de ajustar cuidadosamente los parámetros y de iterar en las implementaciones para obtener resultados confiables.

## Anexos

Repositorio de github:

<https://github.com/juanazorzolo/TP2-PDI-TUIA-Herrera-Oviedo-Zorzolo>