

## **INFORME TP2**

**PROCESAMIENTO DE IMÁGENES, 2025**



**FACULTAD DE CIENCIAS EXACTAS, INGENIERÍA Y AGRIMENSURA**

Tecnicatura Universitaria en Inteligencia Artificial

*Bollini Lorenzo - B-6729/6*

*Speranza Emanuel - S-5882/3*

## Problema 1 – Detección y clasificación de monedas y dados

Para resolver este problema se desarrolló un script con el objetivo de detectar objetos en una escena y clasificarlos como *monedas* o *dados*, para luego distinguir entre los distintos tipos de monedas y determinar la cantidad de puntos de cada dado.

El procesamiento comienza cargando la imagen y escalándola. Luego se convierte a escala de grises y se aplica un suavizado mediante **GaussianBlur**, con el fin de reducir ruido antes de detectar bordes. Para corregir variaciones de iluminación en el fondo, se utiliza un **Top-Hat morfológico**, técnica basada en una operación de apertura que destaca objetos brillantes sobre fondos irregulares.

En la etapa de detección de bordes, se aplica **Canny**, seguido por una secuencia de **dilatación**, **clausura** y **erosión**, con el objetivo de cerrar contornos incompletos sin unir objetos adyacentes. Esto permite obtener bordes definidos y correctamente separados, adecuados para la detección de contornos externos.

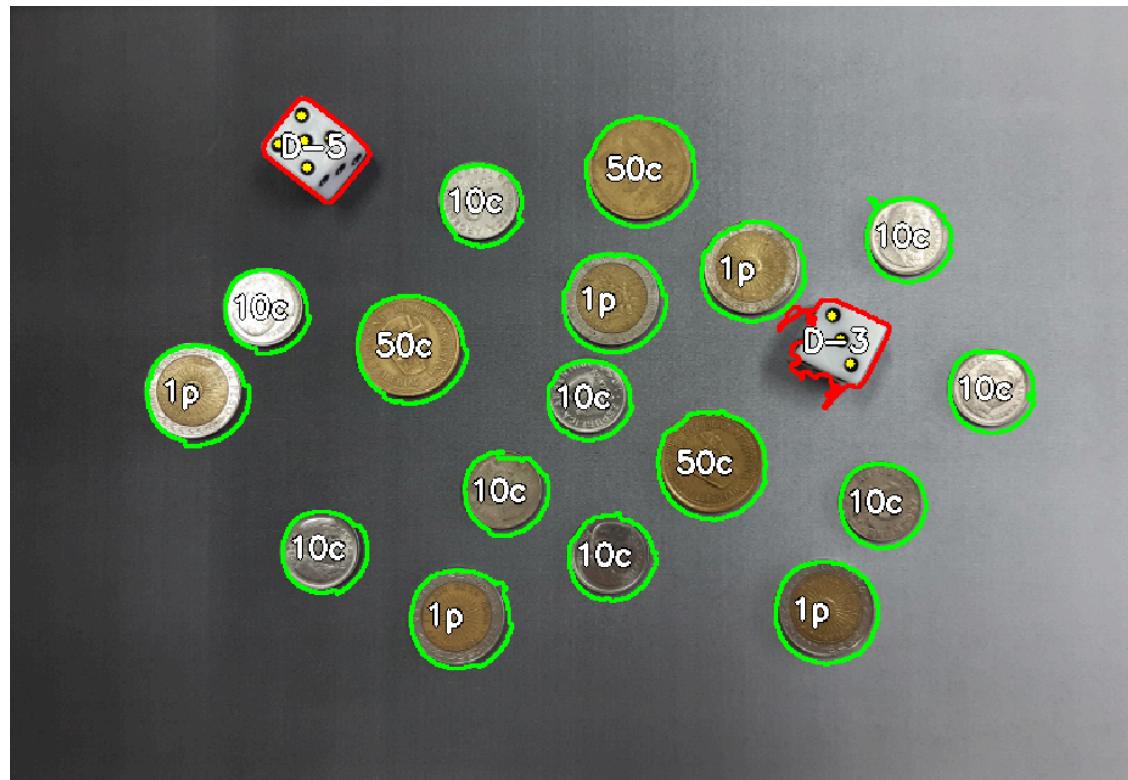
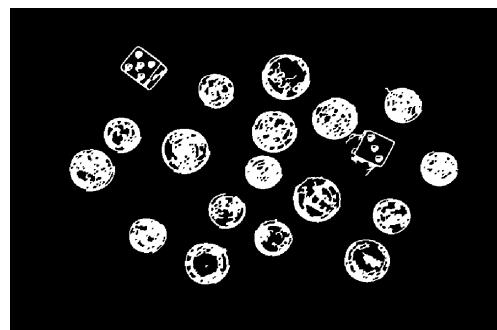
Una vez obtenidos los contornos, el script evalúa cada objeto mediante dos criterios principales:

1. **Factor de forma (Fp)** = área / perímetro<sup>2</sup>, que diferencia objetos redondeados (monedas) de objetos con forma cuadrada (dados).
2. **Cantidad de vértices**, obtenidos con approxPolyDP, para reforzar la clasificación cuando el factor de forma no es concluyente.

Si un contorno se clasifica como moneda, se estima su denominación según su área proyectada, distinguiendo entre **10 cent**, **1 peso** y **50 cent**.

Para los contornos clasificados como dados, el sistema recorta el interior del objeto y aplica umbralización adaptativa para detectar los puntos. Luego calcula la circularidad de cada componente conectado para asegurarse de contar únicamente las marcas válidas, ignorando ruido o bordes rotos. Finalmente, se determina el valor del dado según la cantidad de puntos detectados.

El script dibuja las etiquetas sobre la imagen original y muestra un resumen final, indicando la cantidad de monedas clasificadas por denominación y los valores encontrados en los dados. Esto cumple con los requerimientos del problema integrando técnicas de preprocessado, morfología matemática, detección de contornos y análisis geométrico.



## **Problema 2 – Detección de patentes y segmentación de caracteres**

El segundo problema aborda la detección automática de patentes vehiculares y la segmentación de sus caracteres. El proceso se divide en cuatro etapas principales, utilizando herramientas de preprocesamiento, morfología y descriptores geométricos.

En el **Preprocesado** la imagen se convierte a escala de grises, se suaviza mediante GaussianBlur y se aplican gradientes verticales con **Sobel en X**, con el objetivo de resaltar los bordes característicos de la placa. Luego se normaliza el resultado para mejorar el contraste y se aplica **umbralización de Otsu**, obteniendo así una imagen binaria donde la zona de la patente se destaca del resto del vehículo.

**En la Definición del objeto** utiliza morfología matemática para refinar el bloque correspondiente a la placa. Primero se realiza una erosión vertical con un kernel estrecho para eliminar detalles finos como las rejillas de la parrilla al frente del auto. Después se aplica una **clausura horizontal**, uniendo los elementos que componen la placa y consolidando su estructura rectangular.

En la **Clasificación de la patente** se detectan los contornos de la imagen morfológica y se calcula para cada uno su relación de aspecto, área y posición en la imagen. Estos descriptores funcionan como filtros que permiten descartar objetos que no cumplen con la geometría típica de una patente. El algoritmo selecciona como candidato final el contorno con mayor área que cumple todos los criterios, devolviendo sus coordenadas.

Con la patente localizada, la **Segmentación de caracteres** trabaja sobre el recorte correspondiente.

Finalmente, el programa muestra el recorte de la patente detectada, la segmentación obtenida y la cantidad total de caracteres hallados, cumpliendo así con los objetivos planteados para el problema.

