

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.

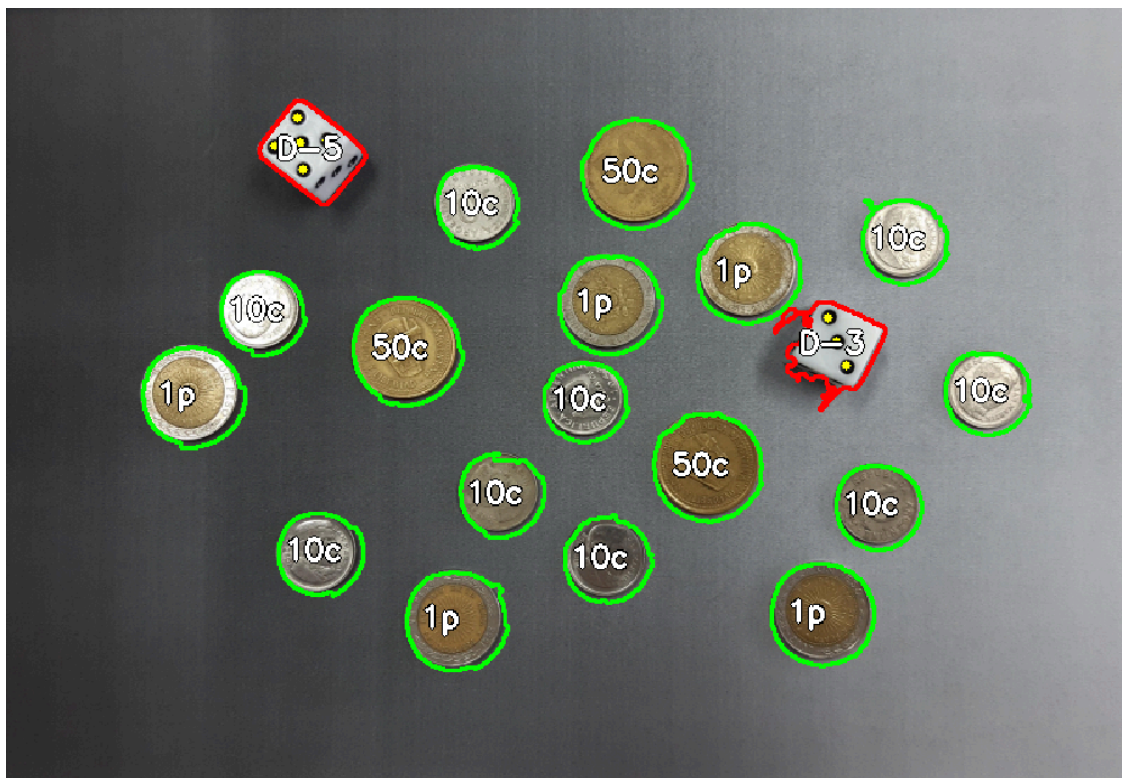
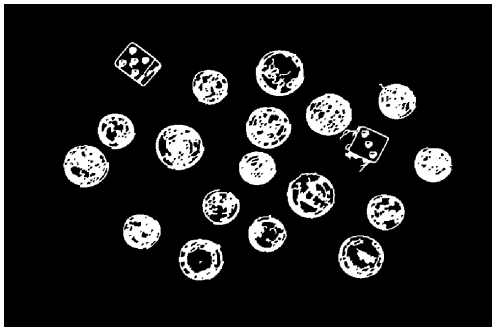
Una vez obtenidos los contornos, el script evalúa cada objeto mediante dos criterios principales:

1. **Factor de forma (Fp)** = $\text{área} / \text{perímetro}^2$, 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 preprocesado, 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.

Aquí tienes el informe adaptado para reflejar fielmente la lógica de tu código final, destacando especialmente la estrategia de **robustez iterativa** que implementaste en la segmentación.

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 algoritmo desarrollado divide el proceso en cuatro etapas secuenciales: preprocesamiento, definición morfológica, clasificación geométrica y una segmentación robusta con reintentos automáticos.

En la etapa de **Preprocesado**, la imagen original se convierte a escala de grises y se suaviza mediante un filtro *GaussianBlur* (5x5) para reducir el ruido. A continuación, se aplica un operador *Sobel en X* para resaltar los gradientes

verticales, característicos de los caracteres y los bordes laterales de la patente. El resultado se normaliza para maximizar el contraste y se somete a una umbralización de Otsu, generando una imagen binaria donde las texturas verticales se destacan del resto del vehículo.

Para la **Definición del objeto**, se utiliza morfología matemática con el fin de consolidar la región de la placa. Primero, se aplica una erosión vertical con un kernel rectangular estrecho (1x3) para eliminar ruido o líneas finas horizontales (como parrillas de radiadores). Posteriormente, se realiza una operación de cierre (*Closing*) con un kernel horizontal alargado (14x1), lo que permite fusionar los caracteres individuales en un único bloque sólido rectangular, facilitando su detección como un solo contorno.

En la **Clasificación de la patente**, se extraen los contornos de la imagen morfológica y se filtran mediante descriptores geométricos. Se evalúan la relación de aspecto (buscando proporciones rectangulares típicas de patentes), el área y la posición relativa en la imagen. El algoritmo incluye un criterio de selección que prioriza el candidato con mayor área que cumpla con los rangos establecidos, e incorpora una lógica específica para casos difíciles basada en la centralidad y puntuación geométrica del contorno.

Finalmente, la **Segmentación de caracteres** implementa una **estrategia de robustez iterativa**. El algoritmo intenta segmentar el recorte de la patente mediante una función núcleo que realiza los siguientes pasos:

1. Binarización (Otsu o umbral fijo) y erosión para separar los caracteres de los bordes blancos de la placa.
2. Refinamiento del recorte mediante componentes conectados para aislar la zona útil de texto.
3. Segunda binarización y erosión vertical para separar caracteres que pudieran estar "pegados".
4. Filtrado de los componentes finales basándose en el ancho y alto esperados de los caracteres.

Si la configuración inicial no detecta al menos 6 caracteres, el sistema reintenta automáticamente el proceso variando los parámetros (cambiando Otsu por umbral fijo o aumentando la intensidad de la erosión) hasta obtener el mejor resultado posible.

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.

