Facultad de
Ciencias Exactas,
Ingeniería y Agrimensura

Universidad Nacional de Rosario

Facultad de Ciencias Exactas, Ingeniería y Agrimensura

Procesamiento de Imágenes 1 - TUIA

TRABAJO PRACTICO Nº 02

25/11/2024

Alumnos:

- Bravi Eugenio (B-6600/1)
- Nemeth Ulises (N-1249/1)

Docentes:

- Sad Gonzalo
- Alvarez Julián
- Calle Juan Manuel

Problema 1 – Detección y clasificación de monedas y dados

El problema 1 consiste de una imagen con monedas y dados en la que hay que segmentar los elementos de la misma y clasificarlos según su valor en el caso de las monedas y según su puntaje en el caso de los dados.

Figura 1: Monedas y dados

Resolución

Para el preprocesamiento de la imagen lo único que se hizo fue pasar la imagen a escala de grises para la posterior detección de círculos.

Figura 1.1: Escala de grises

Para la detección de los círculos se utilizo la “Transformada de Hough” a través del método HoughCircles de Opencv. El gran beneficio de este método es que si se ajustan correctamente los parámetros ofrece una gran robustez ante el ruido que pueda tener la imagen y permite detectar círculos de distintos tamaños.
Para la detección de las monedas se utilizó los siguientes parámetros:
HoughCircles(minDist=100, param1=90, param2=190, minRadius=100, maxRadius=200)

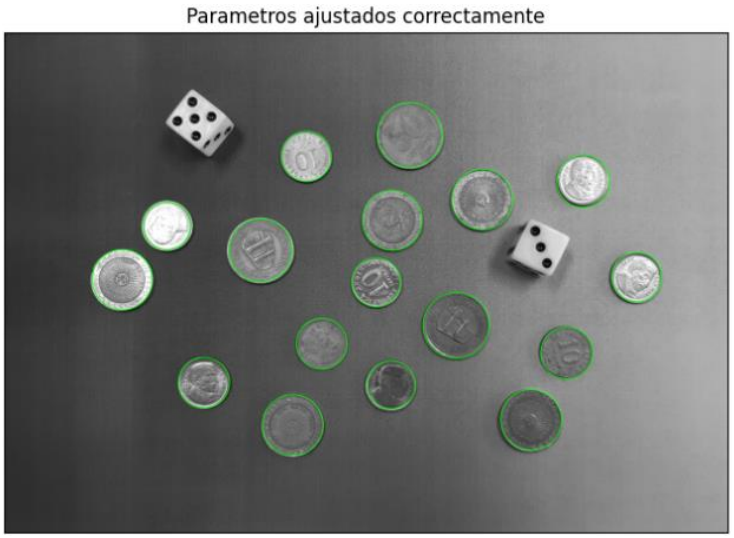

Figura 1.2: Parámetros ajustados correctamente

El parámetro **minDist** es la distancia mínima entre los centros de los círculos, si el parámetro tiene un valor bajo puede detectar varias veces el mismo circulo o si es muy alto puede que no detecte los círculos de monedas que están cerca entre sí.

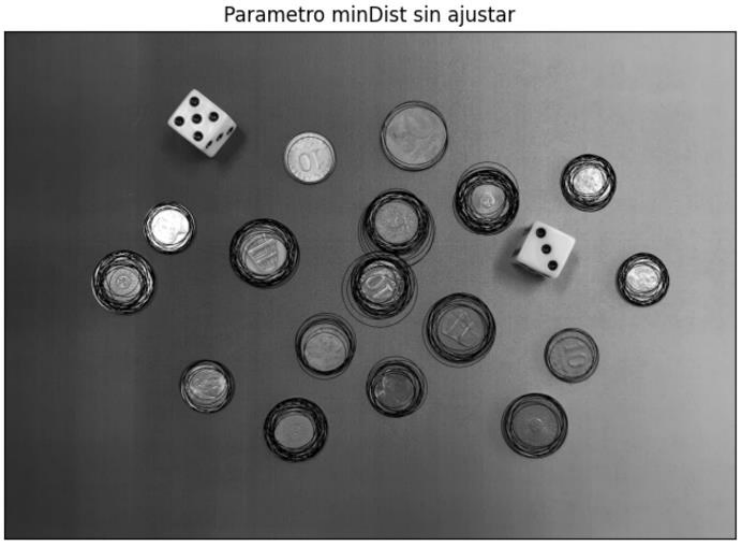

Figura 1.3: minDist sin ajustar

Los parámetros param1 y param2 regulan que tan estricta es la detección de los círculos, si los valores son muy bajos va a detectar falsos positivos(ruido) y si los valores son muy altos se vuelve muy estricto y puede que no detecte correctamente a las monedas.

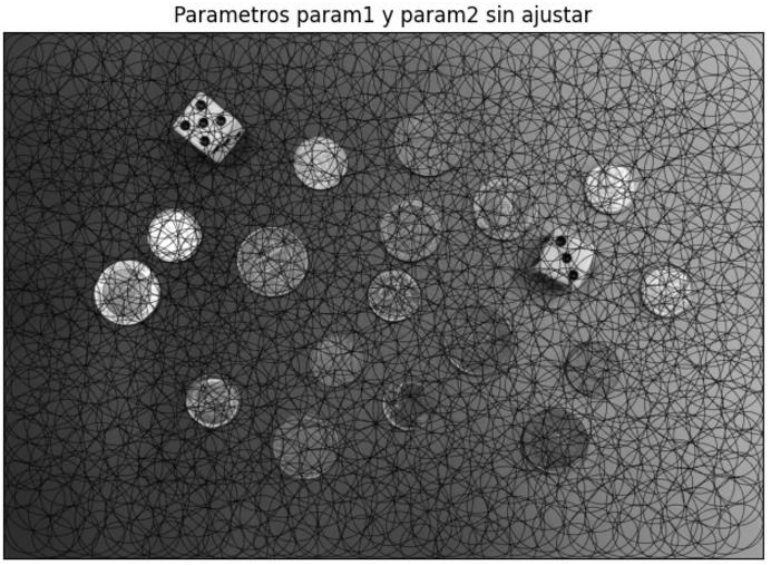

Figura 1.4: param 1 y 2 sin ajustar

Los parámetros minRadius y maxRadius establecen un umbral en los tamaños de círculos a detectar.

Figura 1.5: min y max Radius sin ajustar

Para la detección de los dados se utilizó el mismo método pero con diferentes parámetros HoughCircles(minDist=10, param1=50, param2=50, minRadius=5, maxRadius=30)

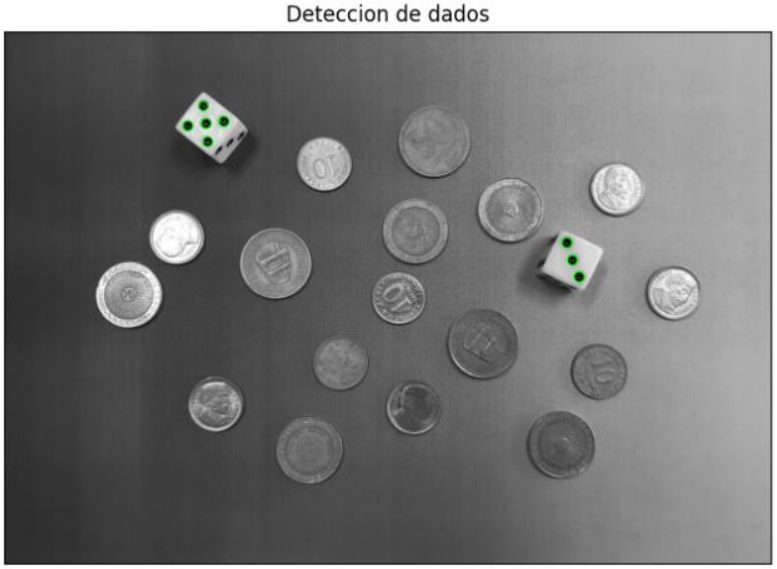

Figura 1.6: detección de dados

Para la clasificación a las monedas con un radio mayor o igual a 167 se las clasifico como monedas de **50 centavos (azul)**, a las monedas con un radio entre 159 y 166 como monedas de **1 peso (rojo)** y al resto como monedas de **10 centavos (verde)**.
Para los dados se los clasifico como de **izquierda (turquesa)** o **derecha (rosa)** según si estaban en la primera mitad de la imagen o en la segunda, y al puntaje se lo calculo contando los círculos de la cara del dado.

Figura 1.6: Clasificación de monedas y dados

Problema 2 – Detección de patentes

El problema 2 consiste en detectar automáticamente las patentes de 12 autos y segmentar los caracteres de la patente, en este problema nos encontramos con autos de distintos colores y donde todas llevan el mismo formato de patente (XXX XXX) pero puede variar el color de la chapa en algunas patentes, ej: centro negro y bordes blancos; centro negro y bordes negros. También nos enfrentamos a patentes con suciedad o rayones que generan ruido al momento de procesarlas.

Resolución

Para la resolución de este problema se decidió buscar las letras de las patentes y con las coordenadas de las letras ubicar la patente, se eligió este método ya que las letras de las patentes son más fáciles de detectar que la placa de la patente, esto se debe a que las letras son siempre blancas sobre un fondo negro, también tienen una proporción en las que son más altas que largas y también tienen aproximadamente el mismo alto y el mismo largo. El inconveniente de detectar directamente las placas de las patentes es que hay casos en los que hay patentes tienen la chapa completamente negra y el auto también es negro lo que dificulta su detección. De esta forma detectar las letras de las patentes generan resultados más consistentes.

Para el preprocesamiento de las imágenes lo primero que se hace es pasar la imagen a escala de grises.

Figura 2.1: Imagen original

Figura 2.2: Imagen en escalas de grises

Luego se obtiene la imagen binaria de la imagen en escala de grises.

Figura 2.3: Clasificación de monedas y dados

Luego a la imagen en escalas de grises se le aplica la transformación Top Hat y se binariza para poder generar un mayor contraste en las letras de la patente.

Figura 2.4: Top Hat

Figura 2.5: Top Hat binarizado

Luego se obtiene la intersección entre la imagen binaria de la imagen de escalas de grises y la del Top Hat binario, esto sirve para reducir levemente el ruido que puede haber dentro de la patente.

Figura 2.6: Intersección

Una vez terminado el preprocesamiento se buscan los contornos de las letras de las patentes, filtrando los contornos obtenidos con cv2.findContours según la proporción y tamaño de los mismos.

Figura 2.7: Detección de letras

Para quedarnos solamente con las letras, agrupamos los contornos por cercanía teniendo en cuenta que la variable ‘y’ en las letras varían muy poco porque están siempre en la misma altura. Luego nos quedamos con el grupo con mayor cantidad de elementos.

Una vez obtenido el grupo con los contornos buscamos la menor y la mayor coordenada de ‘x’, ‘w’ y de ‘y’ para poder extrapolar las coordenadas de la patente.

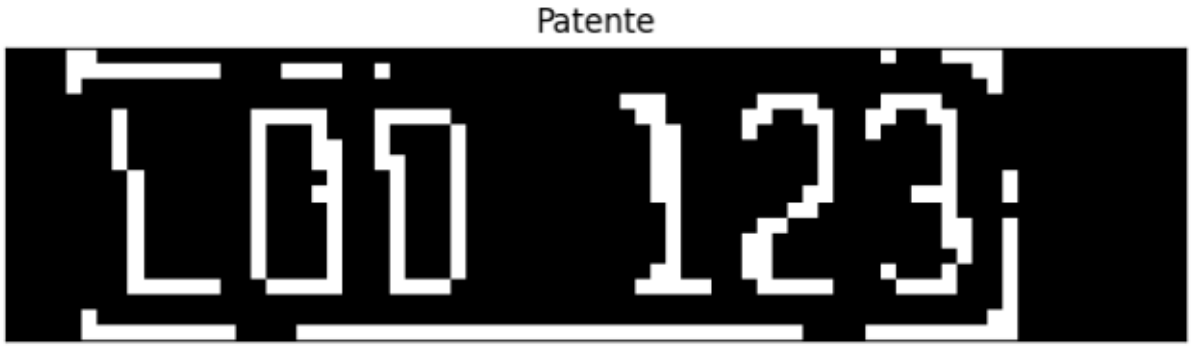

Figura 2.8: Patente

Luego de obtener la patente se le segmentan los caracteres de la misma forma que se hizo previamente.

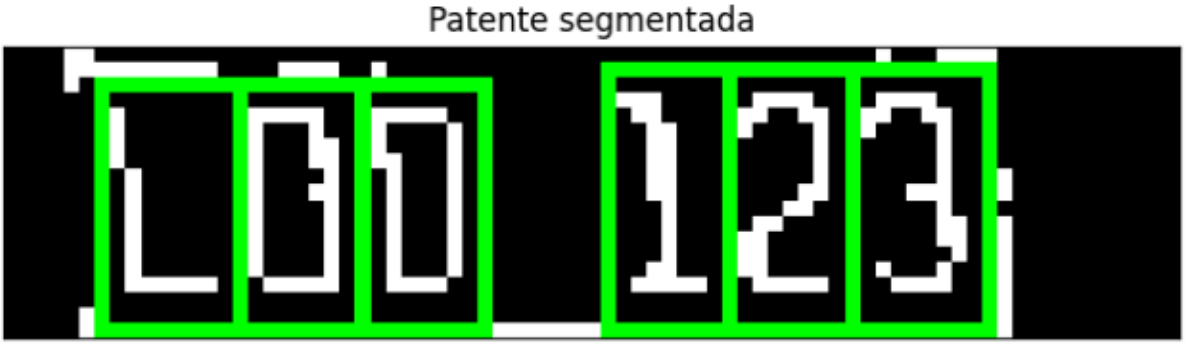

Figura 2.9: Patente segmentada

En caso de que no se encuentren los 6 caracteres de la patente debido al ruido que hay en la misma se procede a procesar la patente en la imagen de escala de grises y aplicarle el umbral THRESH_TOZERO con el objetivo de eliminar el ruido en la imagen.

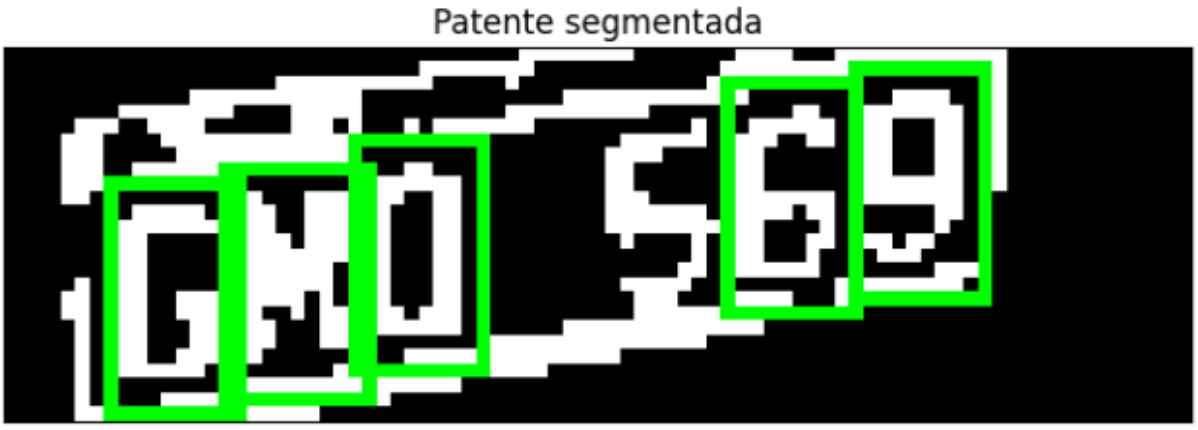

Figura 2.10: Patente segmentada incorrectamente

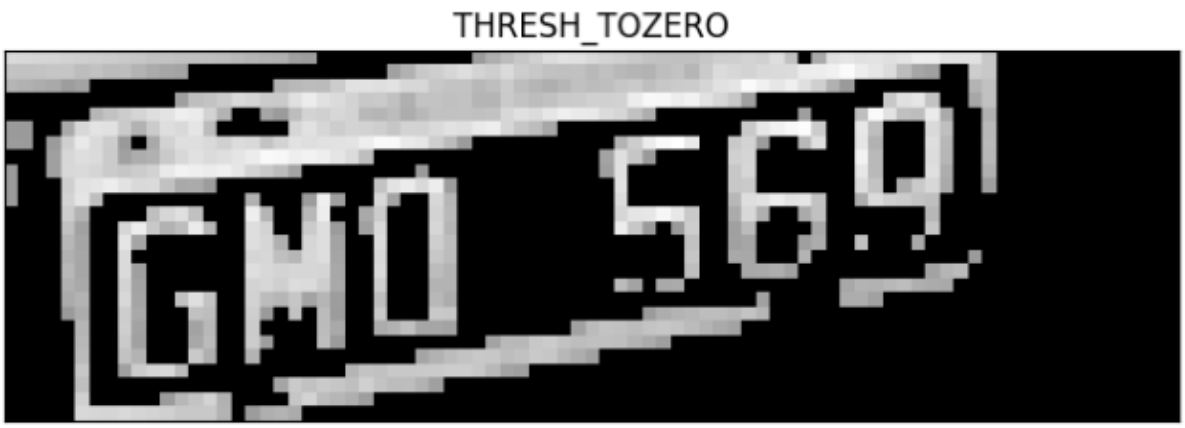

Figura 2.11: THRESH_TOZERO

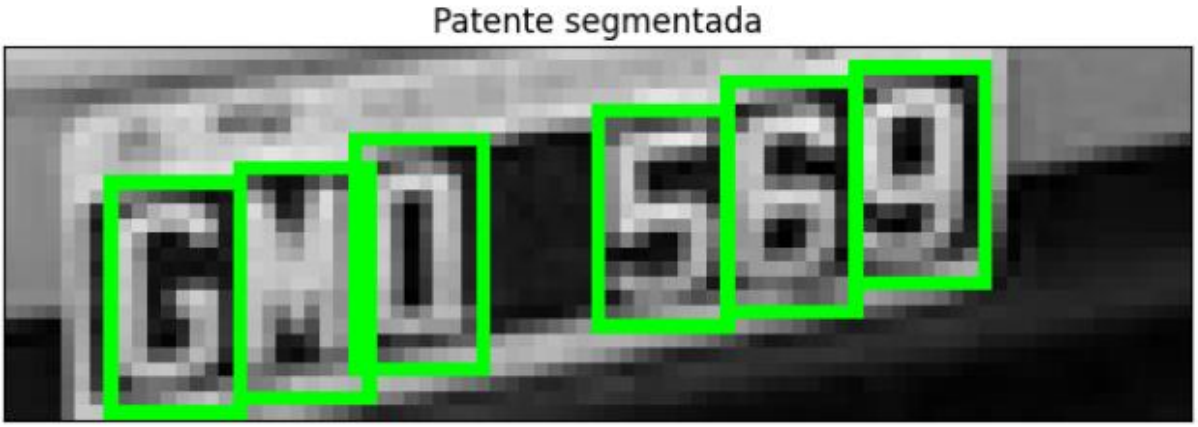

Figura 2.11: Patente segmentada correctamente

Una vez obtenido todas las patentes y de haber segmentado sus caracteres se hace el marcado en la imagen original.

Figura 2.11: Detección de patentes