20-11-2023

PROCESAMIENTO DIGITAL DE IMÁGENES I

Trabajo Práctico Nº2 - INFORME

Integrantes:

- Sanchez, Salvador
- Ponce, Daniel
- Yañez, Mirian

PROBLEMA 1 – Detección y clasificación de Monedas y Dados

Estuvimos trabajando con la imagen "monedas.jpg", que tiene monedas de diferentes tamaños y valores, además de algunos dados en un fondo que no tiene una iluminación uniforme. Básicamente, quisimos analizar la imagen de manera automática y resolver tres problemas: separar las monedas y los dados, contar y clasificar las monedas por tipo, y descifrar automáticamente el número de puntos en cada dado.

En la resolución de este ejercicio, inicialmente abordamos la detección de objetos. Para ello, empleamos la gradiente de la imagen, filtrada previamente para reducir el ruido. Esta aproximación nos permitió obtener contornos, que son esenciales para la identificación de monedas y dados.

Después de aplicar la reducción de ruido y obtener las gradientes, identificamos los contornos exteriores, descartando aquellos con un área inferior a un umbral predefinido para eliminar el ruido y garantizar la relevancia de los objetos detectados. Monedas que estaban demasiado cercanas fueron separadas mediante la aplicación de una operación de apertura con un kernel circular.

Una vez que obtuvimos los contornos de los objetos en escalas de grises, los guardamos para poder utilizarlos en otros scripts de Python.

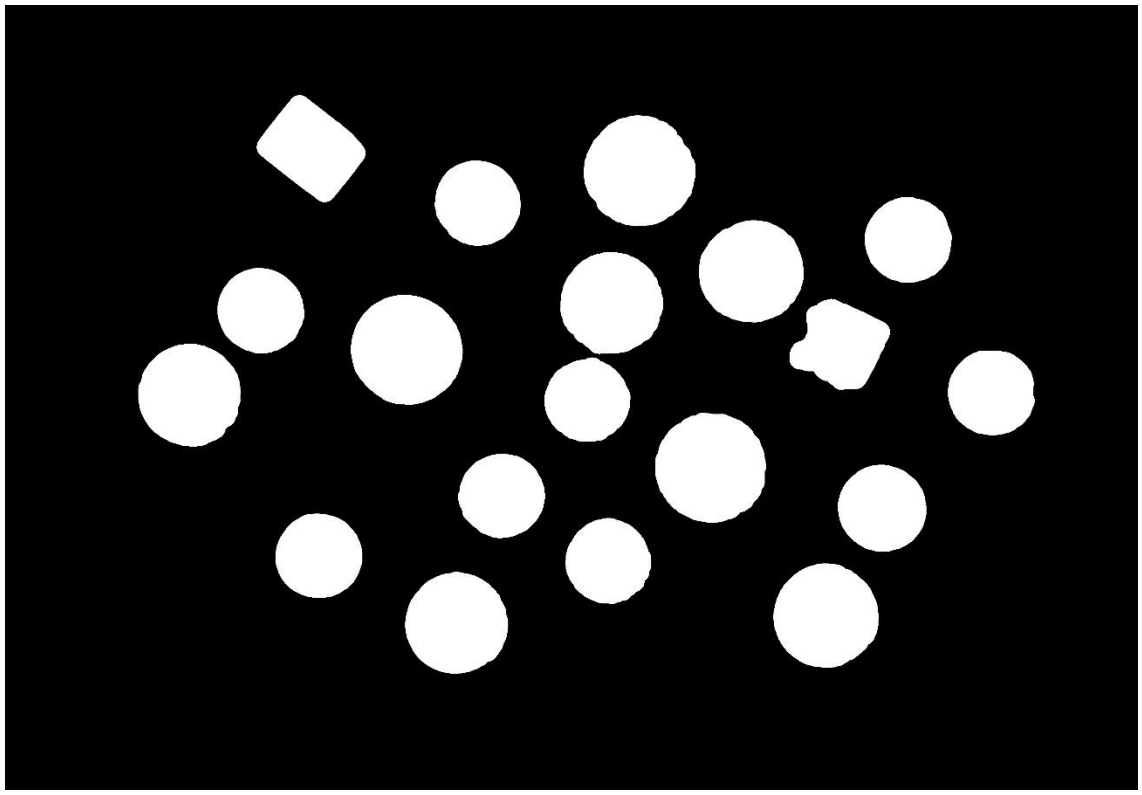
En la etapa de clasificación, utilizamos la función **monedas_y_dados**. Esta función toma la imagen binarizada y obtiene los contornos de cada objeto. Al evaluar el factor de forma de los contornos, distinguimos si es redondo o no. Las monedas, caracterizadas por su forma circular, fueron clasificadas y etiquetadas con sus respectivos valores.

Para mejorar la visualización y comprensión de los resultados, decidimos utilizar una **imagen_resultado**. Esto proporcionó una representación clara de los valores asignados a cada contorno y facilitó la verificación de la precisión de la clasificación.

En cuanto a la identificación de los valores de los dados, implementamos un enfoque específico. Analizamos la subimagen que contenía los dados, calculamos un umbral para los puntos oscuros (representativos de los puntos en los dados) y obtuvimos los contornos correspondientes. La suma de estos puntos proporcionó el valor final asignado a la **imagen_resultado**.

Concluimos la resolución del ejercicio mostrando la **imagen_resultado**, que integra tanto la identificación y clasificación de monedas, la suma total del valor de las monedas y la evaluación de los valores de los dados.

imagen_contornos:



imagen_resultado:

PROBLEMA 2 – Detección de patentes

En este problema, tuvimos que desarrollar un algoritmo de procesamiento de imágenes para la detección automática y segmentación de patentes en vehículos. El proceso consta de dos partes: detección y segmentación de patentes y segmentación de caracteres.

Partimos de la imagen original del vehículo.

Imagen original:

Convertimos la imagen a escala de grises para simplificar el procesamiento.

Imagen en escala de grises:

Aplicamos un filtro de blur para suavizar la imagen y mejorar la detección.

Imagen con filtro blur:

Utilizamos un filtro Canny para detectar los bordes en la imagen.

imagen Canny:

Realizamos iteraciones para completar líneas y mejorar la detección, ajustando el parámetro "iteration" a 3.

Buscamos los contornos y creamos rectángulos alrededor de las áreas detectadas.

Filtramos los contornos según el área definida para identificar la región de la patente. Tratamos de buscar un área genérica pero no nos coincidía, por lo cual seleccionamos el rectángulo que mejor se ajustaba a las condiciones de cada imagen. Para ello, observamos cuál de los rectángulos contenia la patente y seleccionamos el aspect_ratio que mejor se ajustaba.

Imagen con el rectángulo:

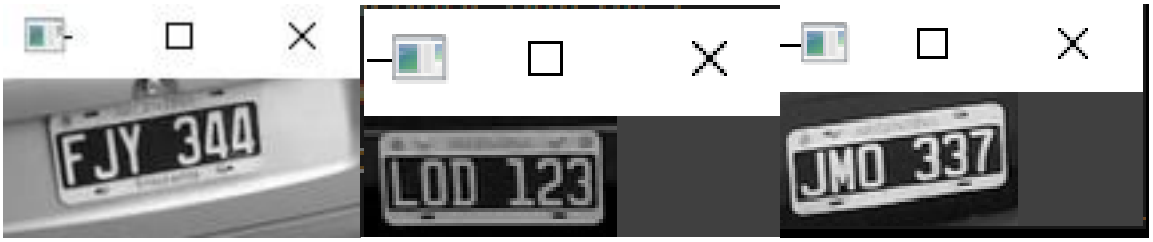
Mostramos la imagen del recuadro que contiene la patente en escala de grises.

imagen procesada:

Las coordenadas identificadas se transfieren a la clase **PatenteOCR** para procesar los caracteres.

La clase **PatenteOCR** se encarga de procesar los caracteres y devuelve el texto de la patente.

Imagenes del texto obtenido a partir de las imágenes procesadas:




```
[] Procesadas/img01.png
[] Procesadas/img02.png
[] Procesadas/img03.png
[] Procesadas/img04.png
['FJY344'] Procesadas/img05.png
Patente 1: FJY344
[] Procesadas/img06.png
['LOD123'] Procesadas/img07.png
Patente 1: LOD123
['JMO337'] Procesadas/img08.png
Patente 1: JMO337
['FFD765'] Procesadas/img09.png
Patente 1: FFD765
[] Procesadas/img10.png
[] Procesadas/img11.png
['BQX387'] Procesadas/img12.png
Patente 1: BQX387
PS C:\Users\mirian\TP_PDI2>
```