Procesamiento de Imágenes

Trabajo práctico N°2 - Informe.

Dimenna, Valentín : D-43366/4

Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Tec. Universitaria en Inteligencia Artificial
2024

Índice

Introducción

Este informe corresponde al trabajo práctico n°2 de la asignatura Procesamiento de Imágenes Digitales de la carrera Tecnicatura Universitaria en Inteligencia Artificial.

El trabajo consta de dos ejercicios: “Detección y clasificación de monedas y dados” y “Detección de patentes”.

En este informe se explica cómo se encaró cada ejercicio, con imágenes paso a paso y comentarios u observaciones.

Ejercicio 1: Clasificación de Monedas y Dados.

Solución.

Dados:

Para la resolución de este ejercicio, como primer paso se define la función que se solicita:

```
def segmentar(imagen,margen_inferior:np.array,margen_superior:np.array):
```

Donde:

- Imagen : imagen en BGR.
- margen_inferior: Valores mínimos de cada canal de un píxel en HSV para ser considerado parte del objeto de interés.
- margen superior: Valor máximo de cada canal de un píxel para ser considerado parte del objeto de interés .

Primero defino los límites para poder separar los dados de las monedas, creo una mascara en escala de grises y después en rgb.

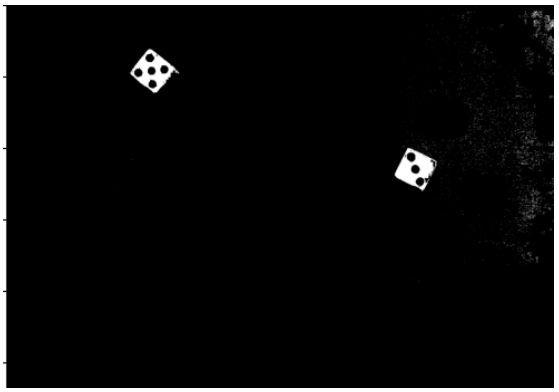
```
lower_bound = np.array([50, 3, 180])
upper_bound = np.array([110, 18, 250])

mask = segmentar(img,lower_bound,upper_bound)
result = cv2.bitwise_and(img, img, mask=mask)#Mascara en RGB
```

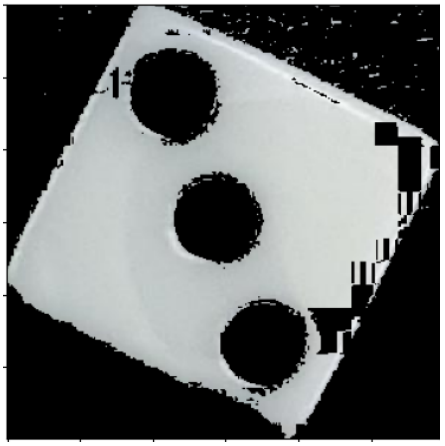
Luego, encuentro los contornos y las jerarquías en la imagen, donde los padres serán los dados y los hijos los puntos de adentro filtrando por área para evitar ruido.

```
    contours, jerarquia = cv2.findContours(mask, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
filtered_contours = [(i,cnt) for i,cnt in enumerate(contours) if
cv2.contourArea(cnt) > 2000]
```

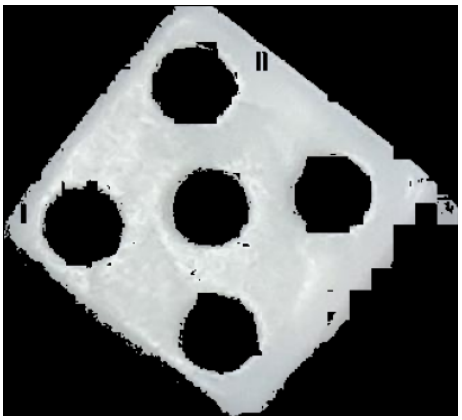
Máscara:



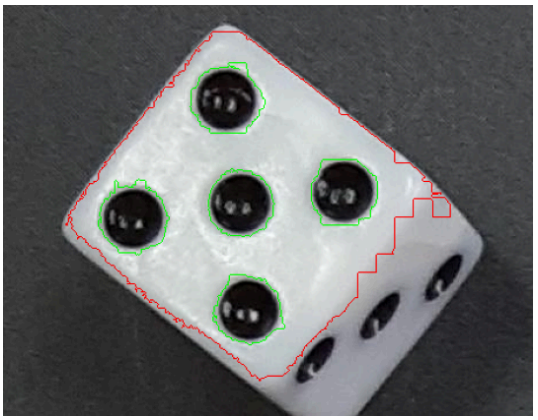
Contorno padre 1:

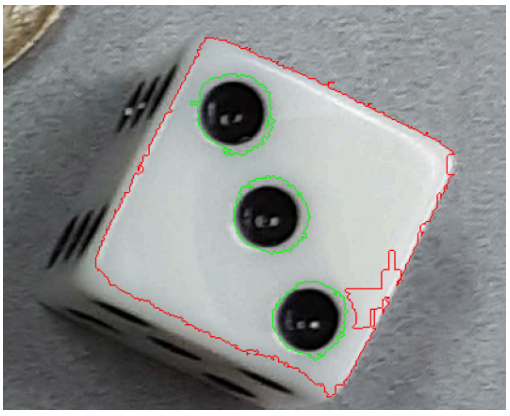


Contorno padre 2:



Luego se cuenta la cantidad de hijos de cada uno de los contornos padre se dibujan y se suman:



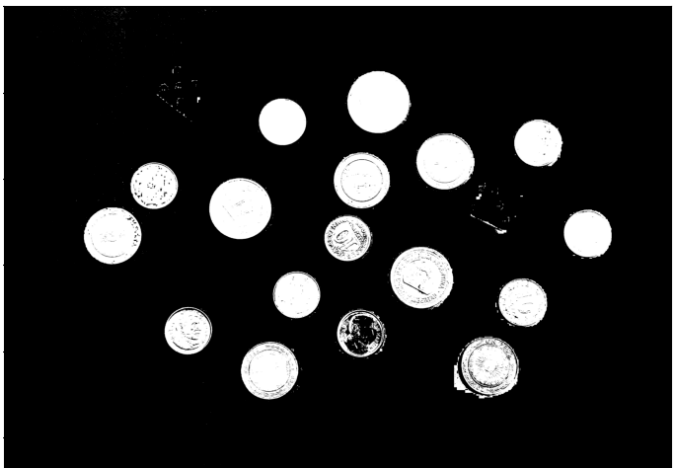


Monedas:

Procedemos a calcular la máscara igual que con los dados pero cambiando los márgenes:

```
lower_bound = np.array([0, 0, 90])
upper_bound = np.array([40, 255, 255])
mask = segmentar(img, lower_bound, upper_bound)
result = cv2.bitwise_and(img, img, mask=mask)
```

Máscara:



Buscamos los contornos y filtramos por área para reducir los contornos de ruido. Se recortaron todas las monedas de la siguiente manera y se guardaron en una lista.

Finalmente se clasificaron según el área de cada moneda si el area es mayor a 95000 es de $0.5, entre 80000 y 95000 es de $1 y si es menor a 80000 la moneda es de $0.1 se cuentan la cantidad de monedas de cada categoría y se suma para obtener el total. Hay $7,40 arriba de la mesa

Ejercicio 2: Detección de patentes.

Para comenzar, se recortaron,se pasaron a escala de grises y se umbralizan todas las imágenes.
Ejemplo de una de las imágenes de los autos.

Después se buscan las componentes conectadas y se filtran por relación de aspecto y área: