



Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Universidad Nacional de Rosario

Procesamiento de Imágenes
Trabajo Práctico Nro. 3
09/12/2024

Pace, Bruno E. P-5295/7.
Crenna, Giuliano. C-7438/1.
Sancho, Sancho. S-5778/9.

Docentes: Sad Gonzalo, Alvarez Julián, Calle Juan Manuel.

Contenidos

1	Abstract	2
2	Introducción	2
3	Solución	2
3.1	Resultado del area de interés	2
4	Conclusión	4

1 Abstract

Este informe presenta el desarrollo y análisis de un problema que consiste en cuatro videos de tiradas de cinco dados, simulando el juego de La Generala. Se aborda desde un paradigma funcional, en el lenguaje Python, en conjunto con la librería OpenCV. Entre las operaciones que se aplican, encontramos filtros de colores HSV mediante umbrales para segmentar, conversión a escala de grises, elementos estructurales, dilatación y componentes conectadas. El fin es definir la ubicación de los dados, identificando el momento en el cual dejan de moverse en conjunto con su resultado de forma automática.

2 Introducción

Consiste en analizar cuatro tiradas de cinco dados para definir de forma automática si la jugada es válida y el resultado de la tirada (puntaje). Una vez realizado esto, se exportan los videos resultantes.

3 Solución

Este problema de tiradas de dados presenta desafíos. Es necesario analizar el video frame a frame para trabajarlos como imágenes individuales. Se tiene en cuenta el encuadre impreciso de la cámara, filtrando el color HSV mediante una máscara, creada por la función:

```
cv2.bitwise_and(frame, frame, mask=mask)
```

Para recortar la imagen, se utilizó:

```
crop_green_region(frame: Matlike) -> Matlike
```

que se encarga de recortar una imagen mediante una máscara. En este caso, quedándonos con la zona de interés verde.

3.1 Resultado del area de interés

Para detectar los dados, fue recurrir a filtrado blur para eliminar detalles no deseados con una ventana cuadrada de 5x5, un filtro de canny para detectar los bordes y la función:

```
dilate_image(image: np.ndarray, kernel_size: int = 3, iterations: int = 1) -> np.ndarray
```

Esta última binariza la imagen, toma elementos estructurales por medio de la morfología MORPH RECT con una ventana de 3x3 y dilata la imagen.

Luego, se obtienen las componentes conectadas de la imagen resultante. Aquí se identifican los dados, ya que en la función

```
filter_components(frame: Matlike, components: Tuple[Matlike, Matlike, Matlike, Matlike]) -> Tuple[Matlike, Tuple[int, int, int, int]]
```

Se filtran las componentes conectadas, por factor de forma de los dados y se obtienen las coordenadas de los bounding boxes.

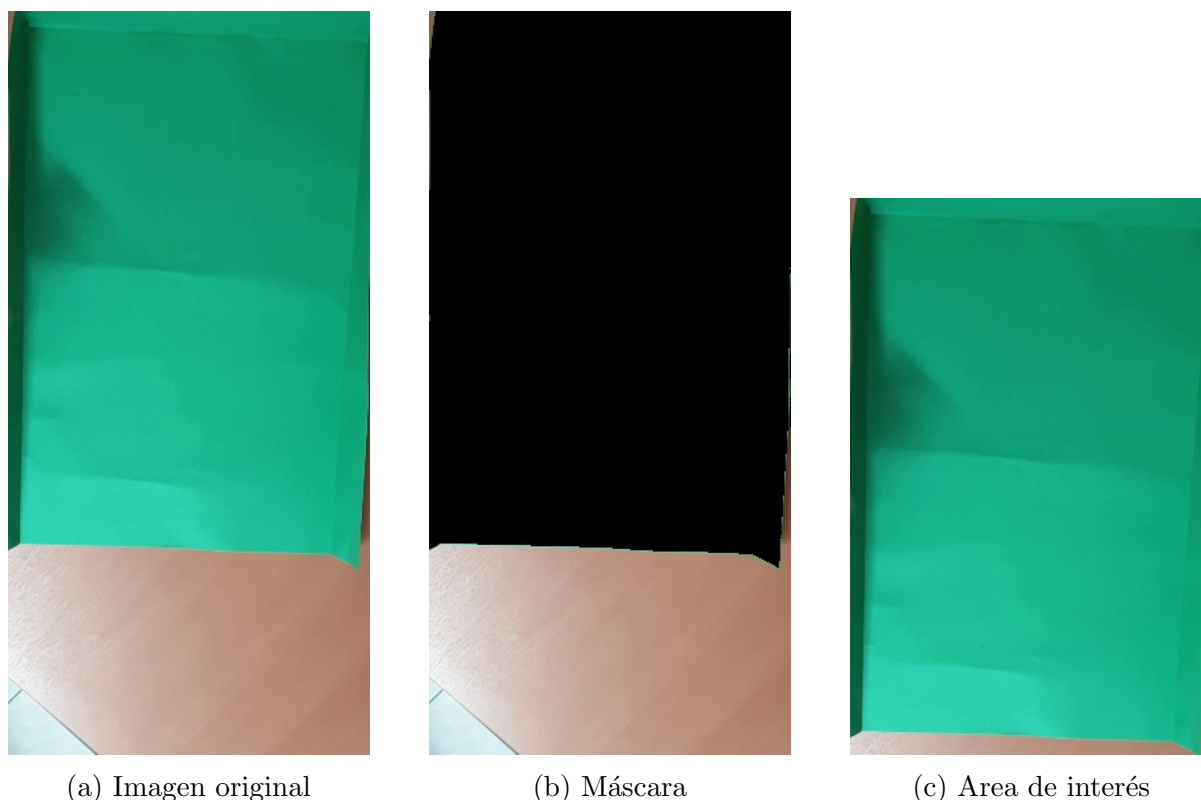


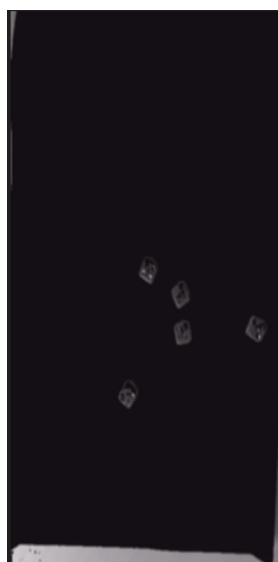
Figure 1: Comparación entre imagen original, imagen con región verde detectada para máscara e imagen recortada a zona de interés.



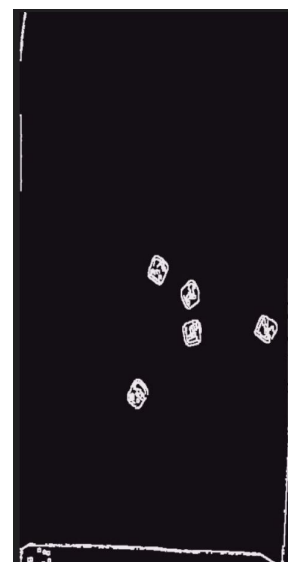
Hay un control de la cantidad de dados. Si son 5 dados, la jugada es válida y se procede a dibujar los bounding box y calcular la cantidad de puntos de cada dado.

Para los puntos cada cuadrado (dado), se aplica un filtro blur nuevamente para eliminar detalle no deseable y detectar mediante HoughCircles los círculos internos. De esta manera, se obtiene la cantidad de círculos de cada dado y en consecuencia, su puntaje.

Se dibujan los bounding boxes, escribiendo sobre el lado superior el puntaje de cada dado y en la esquina superior izquierda del frame, el puntaje de la



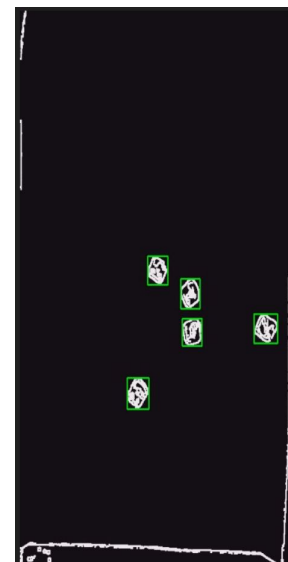
(a) Imagen original



(b) Máscara



(c) Área de interés



(d) Componentes

Figure 2: Comparación entre imagen original, imagen con región verde detectada para máscara, imagen recortada a zona de interés y componentes detectados.

tirada.

Finalmente, en la carpeta "output" se almacenan los videos de tiradas finales con sus procesos.

4 Conclusión

La solución presentada ha demostrado ser eficaz en la identificación y análisis automático de las tiradas de dados en un video, utilizando técnicas de procesamiento de imágenes sin recurrir a valores fijos ("hardcoding"). A través de la segmentación por color, la detección de bordes, y el uso de componentes conectadas, se logró detectar los dados en movimiento y calcular su puntaje de

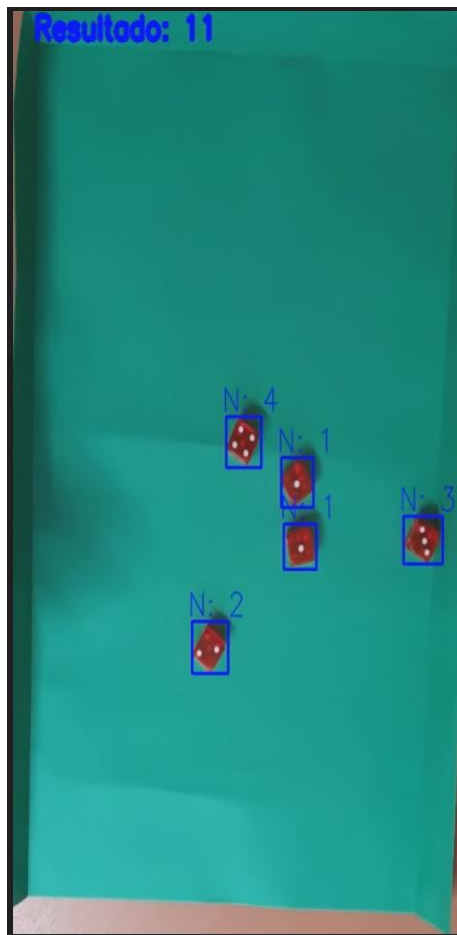


Figure 3: imagen final

manera precisa. Este enfoque flexible y escalable permite analizar múltiples tiradas y es aplicable a diversas situaciones en las que se requiere detectar objetos en movimiento en un contexto controlado.