

# Procesamiento de Imágenes

# **Trabajo práctico 2**

---

Manuel Spreutels

Augusto Rabbia

Diciembre, 2023



## Introducción

El trabajo práctico se centra en el análisis de un video. Debemos en una primera etapa encontrar los datos en una imagen, dibujando bounding boxes en sus posiciones, y luego, escribir el valor que estén mostrando estos datos.

## Setting

El trabajo se desarrolló en Python 3.11, aprovechando las funciones proveídas por la librería de OpenCV y numpy.

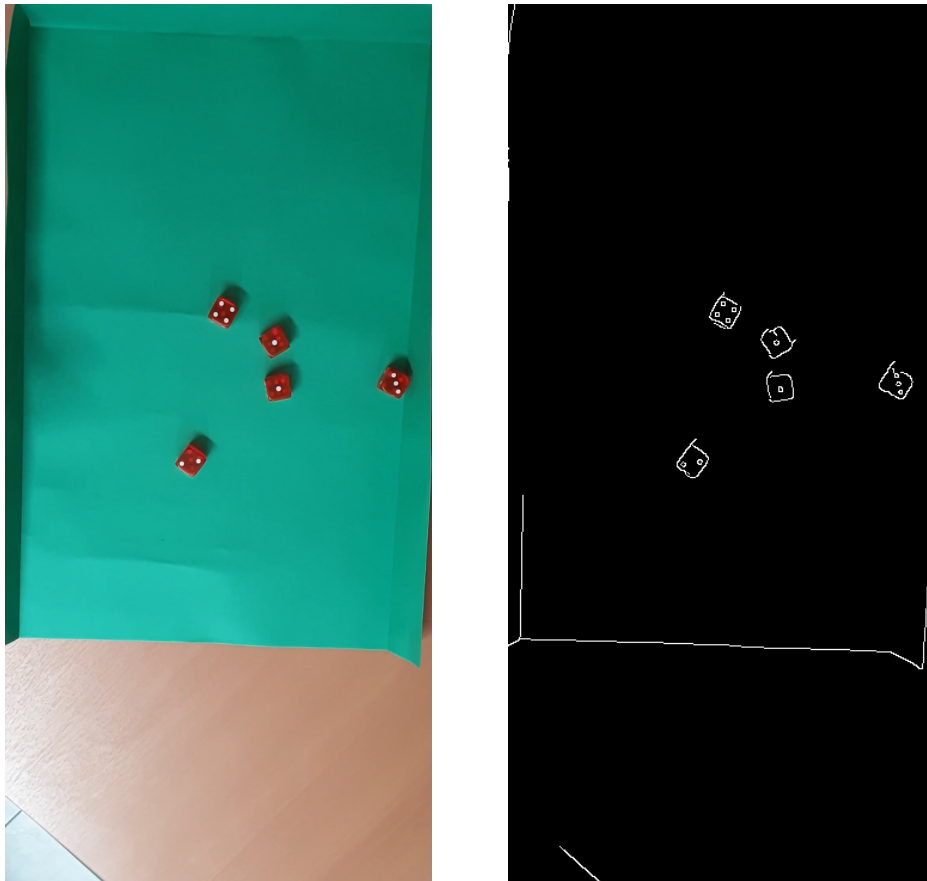
## Desarrollo

Para la solución del problema, dividimos el mismo en tres subproblemas más sencillos:

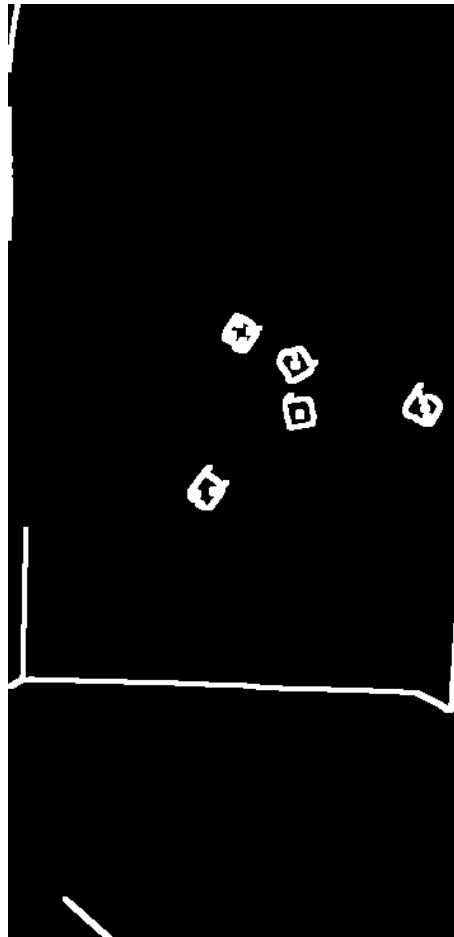
1. Encontramos la posición de los dados.
2. Comprobamos que estén quietos, comparándolos con la posición en el anterior frame.
3. Dada esta posición, obtenemos los valores que muestren.

### 1. Encontrando la posición de los dados

Para encontrar la posición de los dados, obtenemos los bordes de la imagen utilizando la técnica Gaussian Blur + Canny.



Una vez obtenidos los bordes, hacemos una dilatación para asegurarnos de que todos los bordes se unen formando una única componente conectada.



A partir de esa imagen, obtenemos las componentes conectadas con `cv2.connectedComponents`. Entre estas, filtramos las componentes conectadas de interés, es decir, las de los bordes de los dados, de las demás teniendo en cuenta su área y su forma: deberán ser relativamente cuadradas, y tener un área suficiente, y no demasiado grande.

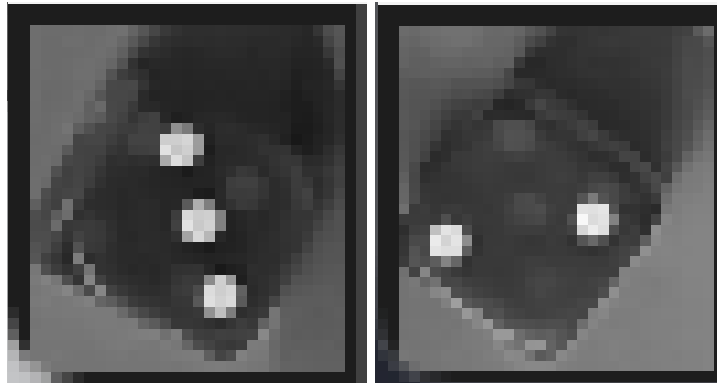


## 2. Comprobando que estén quietos.

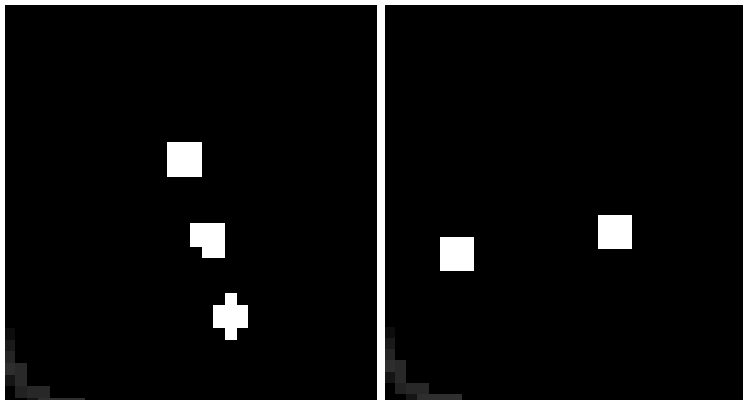
Como obtenemos la posición de los dados en cada frame, podemos compararlas con las posiciones inmediatamente anteriores y comprobar si existe un cambio lo suficientemente significativo como para considerarlos en movimiento. Esto es posible gracias a que la cámara en los vídeos está completamente fija, lo cual nos asegura que si la componente conectada más cercana tiene coordenadas con una diferencia significativa, entonces el dado está en movimiento.

## 3. Obteniendo los valores de los dados.

Como ya contamos con la posición de cada uno de los dados, podemos segmentar las subimágenes correspondientes a cada uno de ellos, previamente realizando una conversión de color a grayscale, lo cual nos será útil en el próximo paso:

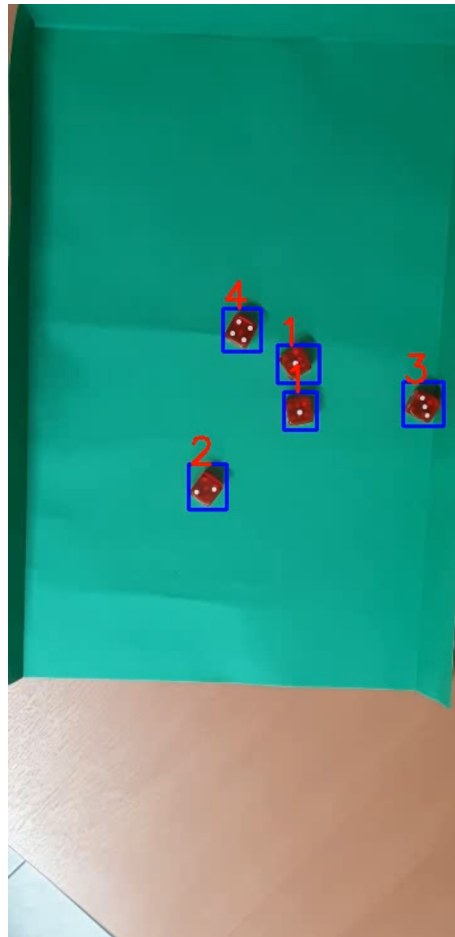


Luego, en cada una de estas subimágenes, aprovechando que los puntos de los dados son de un blanco muy claro, podemos umbralar:



Con este resultado, basta con simplemente contar la cantidad de componentes conectadas para obtener el valor del dado.

Por último, con la bounding box anteriormente obtenida y el valor del dado, podemos generar el vídeo que muestra estos detalles.




## Resultados

Los vídeos generados se pueden encontrar en [este link](#) .

## Conclusion

Este proyecto ha demostrado la viabilidad y eficacia de utilizar técnicas de visión por computadora para automatizar la detección y análisis de eventos específicos en videos, en este caso, el reconocimiento de dados en reposo y la lectura de los valores obtenidos.

La utilización de bibliotecas como OpenCV simplificó significativamente el procesamiento de videos, permitiendo abordar cada cuadro como una imagen independiente. Este enfoque modular y sistemático facilitó la implementación del algoritmo, ya que cada cuadro pudo ser



tratado de manera consistente, aplicando las mismas operaciones de detección y reconocimiento.