Facultad de Ciencias Exactas, Ingeniería y Agrimensura

Tecnicatura Universitaria en Inteligencia Artificial

Procesamiento de Imágenes I

Trabajo Práctico 3

Docentes:

- Gonzalo Sad
- Juan Manuel Calle
- Joaquín Allione

Integrantes:

- Sofía Grando: G-6004/6
- Jeremías Olivieri:
- Miranda Pilar Onega: O-1779/5

15 de diciembre de 2025

Índice

Problema 1 - Cinco dados

1. Descripción del problema

En este problema se presentan 4 videos, donde cada uno se corresponde con una tirada de 5 dados, como se puede observar en el ejemplo de la Figura 1. En primer lugar, se solicita desarrollar un algoritmo para detectar automáticamente los frames donde los dados se encuentren detenidos y mostrar por la terminal cada uno de los valores obtenidos. En segundo lugar, se pide generar un video para cada archivo en el cual, al detenerse los dados, se muestren con sus bounding boxes asociados, un nombre que los identifique y su valor obtenido.

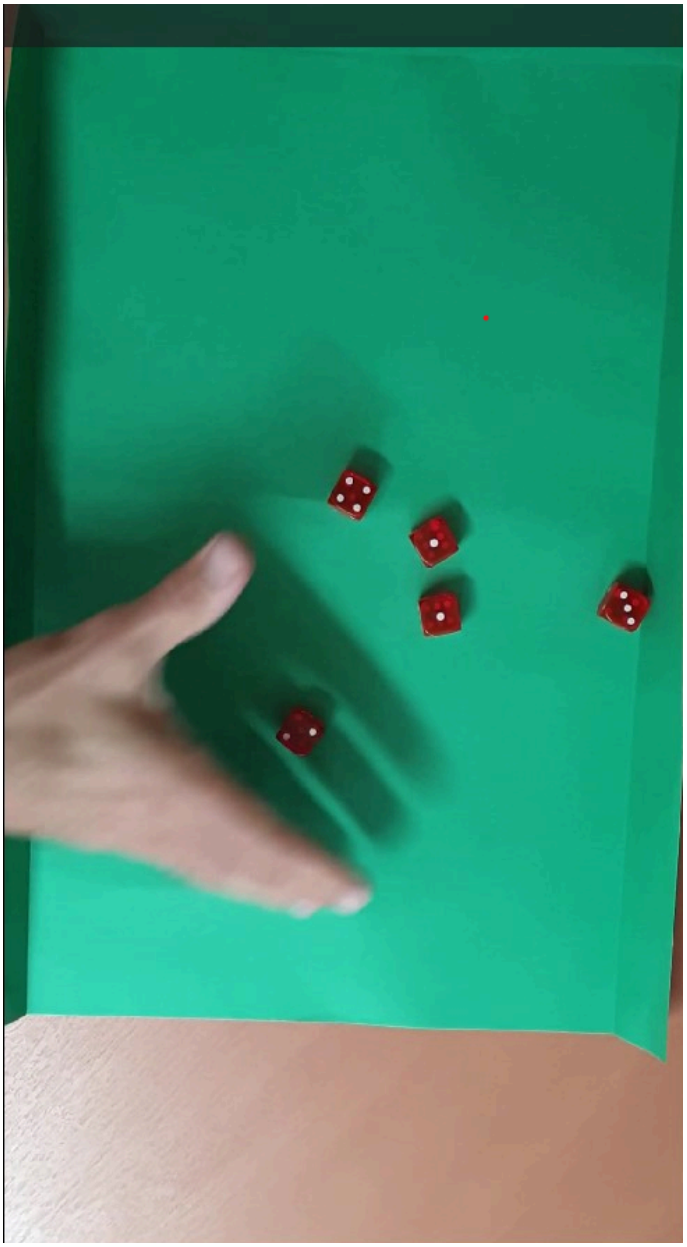

Figura 1: Captura del video tirada_1.

2. Análisis de desafíos enfrentados

El desarrollo del algoritmo para detectar dados y determinar su valor, presentó varios desafíos.

Uno de estos desafíos fue la segmentación de los objetos (dados rojos) sobre el fondo verde, complicada por la variación de la iluminación. Para abordar esto, primero se requirió experimentar y ajustar los umbrales HSV. La dificultad de aislar el color rojo en el espacio HSV, donde el Hue es cíclico, requirió una estrategia de doble umbral para asegurar que el color se capturara completamente.

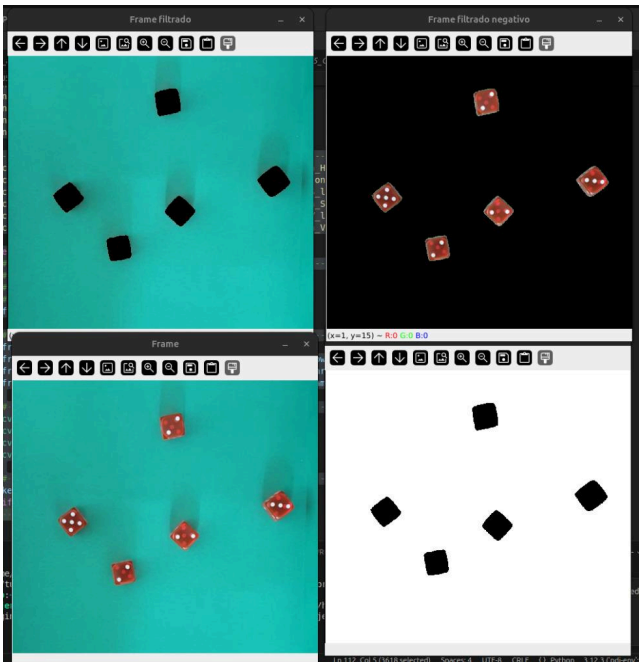

Figura 2: Filtros aplicados.

Para eliminar el ruido y mejorar la precisión, se implementó una estrategia de localización jerárquica. En lugar de buscar los dados rojos en todo el frame, primero se segmentaron y localizaron las áreas verdes, que definían las ROIs donde se esperaban los dados. La segmentación del color rojo y las operaciones posteriores se aplicaron únicamente dentro de estas ROIs, lo que simplificó drásticamente el problema de la separación objeto-fondo.

Finalmente, el requisito de determinar el valor del dado una vez que estaba estático fue otro desafío a enfrentar. Esto se resolvió mediante la comparación de la posición de los centroides de los dados entre frames consecutivos. Solo cuando la distancia euclidiana entre los centroides de un dado era inferior a un umbral predefinido se consideraba que el dado estaba en reposo.

3. Técnicas utilizadas

La base del procesamiento fue la segmentación basada en color, utilizando el espacio de color HSV debido a su mejor desacoplamiento de la intensidad lumínica. La definición de los umbrales óptimos para los colores rojo, verde y blanco se logró a través de un proceso iterativo de prueba y error.

En el archivo fitlros_colroes_hsv.py se fueron filtrando los colores para una imagen estática. Esta herramienta permitió la visualización en tiempo real de la máscara binaria y los resultados de los filtros. La técnica de segmentación jerárquica permitió aplicar umbrales específicos: doble rango de Hue para el rojo y un umbral de baja Saturación/Alto Valor para el blanco de los puntos.

Posteriormente, la detección del estado estático de los dados se implementó comparando la posición de los centroides entre frames consecutivos. La función emparejar_centroides calcula la distancia euclidiana entre un centroide actual y uno previo. Si esta distancia es menor que un umbral, se considera que el objeto está estático. Esta condición de estaticidad es fundamental para luego realizar el conteo de puntos y asegurar que la lectura se realice en el momento correcto.

Luego, para determinar el valor numérico, se aplicó la segmentación del color blanco únicamente dentro del bounding box del dado estático. A esta sub-imagen se le aplicaron operaciones morfológicas de apertura y clausura, mejorando la obtención de los puntos disminuyendo el ruido y se buscaron componentes conectadas para obtener el valor instantáneo del dado. El proceso completo para la detección de los valores de los dados se probó primero en una imagen estática con el programa filtrado_deteccion.py, para luego implementar el mismo procedimiento en los frames de los videos.

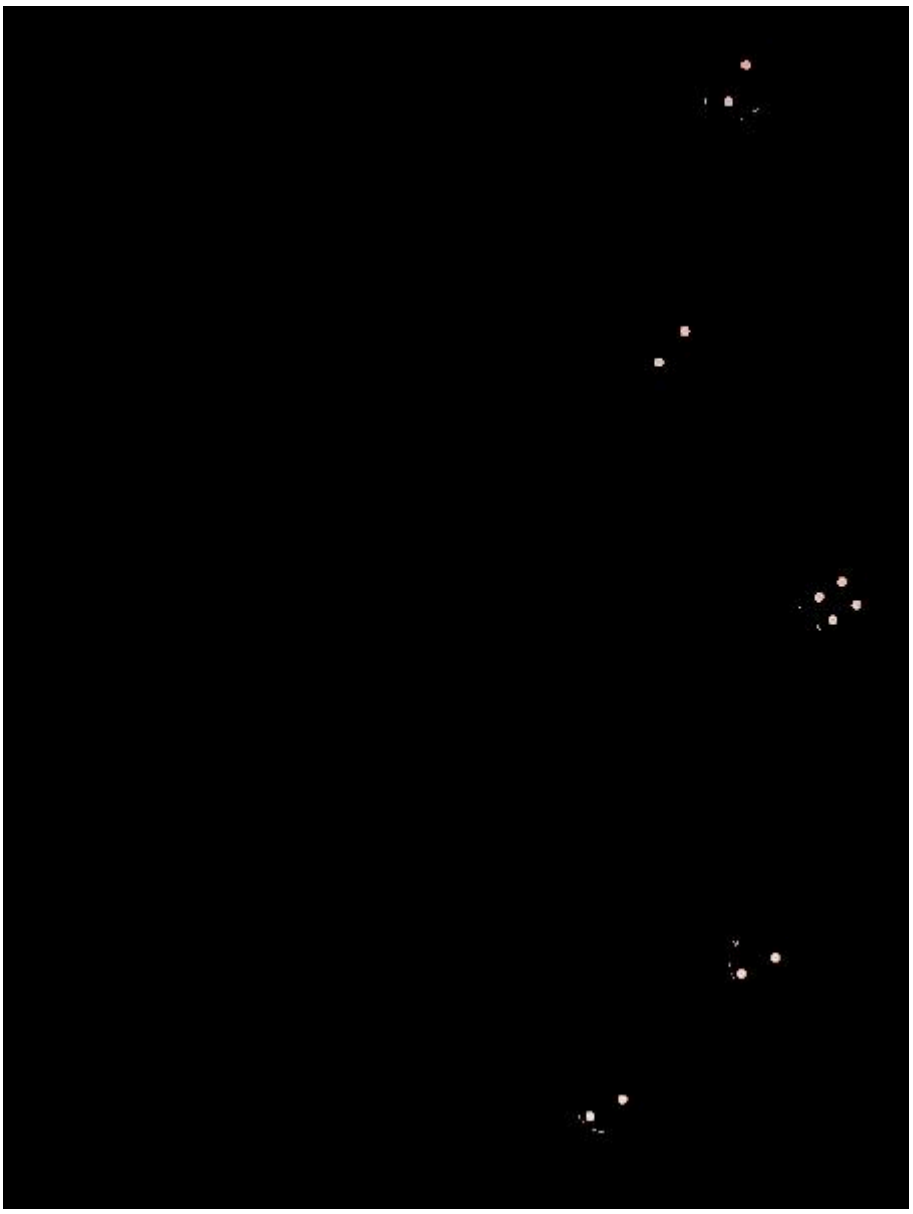

Figura 3: Filtro para valores en dados.

Para la estimación final del valor de la tirada se determinó mediante el cálculo de la moda de los valores obtenidos en todos los frames del video para un mismo dado. Los valores contados para cada dado en los sucesivos frames estáticos se acumularon en la lista tiradas y la función detectar_moda determinó el valor que apareció con mayor frecuencia a lo largo de los frames con los dados estáticos.

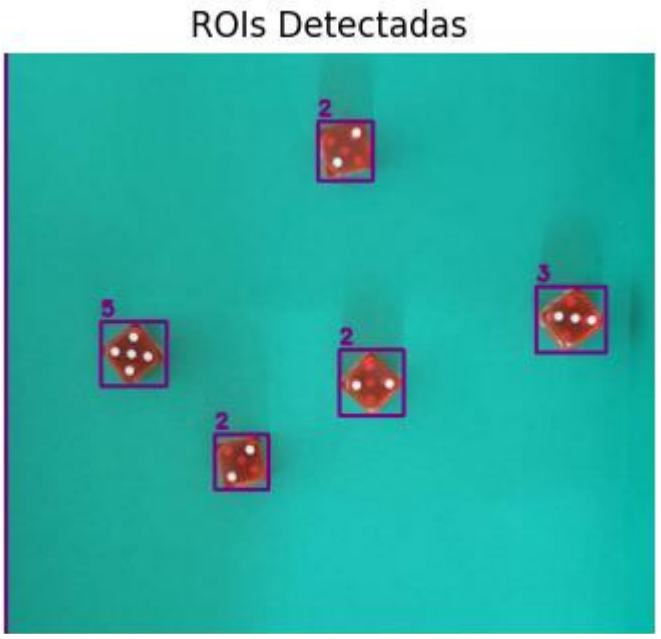

Figura 4: ROIs de dados con bounding boxes y valores.

4. Conclusiones

En resumen, se puede decir que el problema de automatizar la detección de dados en los videos fue resuelto con éxito. Se logró resolver mediante la combinación de la segmentación en HSV y el filtrado morfológico, superando los problemas de iluminación y ruido. La detección de estaticidad a través del análisis de centroides fue la clave para sincronizar la lectura. Y finalmente, el conteo se obtuvo al usar componentes conectadas y la aplicación de la moda, que ayudó a determinar los valores finales en cada tirada. Los resultados obtenidos se observan por terminal y también en la carpeta outputs, y corresponden a las Figuras 5 y 6 respectivamente.

```
Se encontraron 4 videos para procesar

Procesando: tirada_1

Julio: 2
Chuck: 1
Dani: 3
Noether: 1
Ernesto: 4

Video procesado: outputs\tirada_1.mp4
Frames procesados: 146
```

```
Procesando: tirada_2

Julio: 2
Chuck: 3
Dani: 1
Noether: 2
Ernesto: 3

Video procesado: outputs\tirada_2.mp4
Frames procesados: 145
```

```
Procesando: tirada_3

Julio: 2
Chuck: 2
Dani: 4
Noether: 2
Ernesto: 2

Video procesado: outputs\tirada_3.mp4
Frames procesados: 146
```

```
Procesando: tirada_4

Julio: 2
Chuck: 2
Dani: 5
Noether: 3
Ernesto: 2

Video procesado: outputs\tirada_4.mp4
Frames procesados: 88

¡Procesamiento completo!
```

Figura 5: Resultados que se muestran por terminal.

Figura 6: Captura de video output tirada_1.