

TRABAJO PRÁCTICO N° 3

Procesamiento de Imágenes I

Tecnicatura Universitaria en Inteligencia Artificial

FECIA - UNR

Integrantes:

Florencia Fernández
Leandro Salvañá
Leonel Palermo

Resumen

Este trabajo aborda la detección y seguimiento de datos en videos, utilizando técnicas de procesamiento de imágenes y visión por computadora. La metodología se centra en la detección de datos a través de la manipulación del canal rojo en imágenes RGB, seguido de la identificación de componentes conexas y la aplicación de criterios de filtrado basados en área y relación de aspecto. La determinación de la estaticidad de los datos se logra mediante la medición de la distancia euclíadiana entre centroides en frames consecutivos. Una vez identificada la estabilidad, se recorta la capa roja para aislar las subimágenes de los datos, binarizándolas y detectando el número mediante la identificación de componentes conexas. El procesamiento final incluye el dibujo de bounding boxes y números sobre los datos, así como la generación de videos resultantes.

1. Ítem (a).....	1
1.1. Detección de datos.....	1
1.2. Detectar cuándo se quedan quietos.....	2
1.3. Detectar el número en la cara superior del dado.....	2
2. Ítem (b): Dibujar bounding boxes, los números y generar los videos.....	3
2.1. Dibujar bounding boxes y números.....	3
2.2. Guardar los videos resultado.....	4
3. Dificultades.....	4
4. Conclusiones.....	6

1. Ítem (a)

1.1. Detección de datos

El objetivo inicial consistió en la detección de datos en los cuadros del video. Para ello, capitalizamos la presencia de colores "primarios" de la codificación de color "RGB" en los objetos (dados rojos) y el fondo (pañó verde), seleccionando exclusivamente el canal rojo. Subsecuentemente, binarizamos la imagen, dejando los dados (y otros elementos de la imagen) de color blanco, y el paño verde en negro. Posteriormente, procedimos a identificar las componentes conexas mediante la función `cv2.ConnectedComponentsWithStats()` de OpenCV. Una vez localizadas, filtramos dichas componentes en base a su área y relación de aspecto. Esta acción se fundamenta en la premisa de que los dados son uniformes en tamaño, tomando en consideración umbrales ajustados para compensar la reducción de tamaño causada por la perspectiva. Además, se consideraron las dimensiones similares en los ejes x e y, independientemente de su posición al ser proyectados sobre un plano 2D para la relación de aspecto.

1.2. Detectar cuándo se quedan quietos

El siguiente objetivo consistió en determinar cuándo los dados se encontraban inmóviles. Para lograrlo, medimos la distancia euclídea entre los centroides en frames consecutivos y verificamos que estuviera por debajo de un umbral predeterminado. Si esta condición se cumplía, los dados eran considerados estáticos.

1.3. Detectar el número en la cara superior del dado

Con respecto a la detección del número exhibido en la cara superior del dado, una vez confirmada la inmovilidad de los dados, se recortó la capa roja de la imagen para obtener subimágenes delimitadas por las bounding boxes correspondientes a los dados. Estas imágenes se binarizaron con un umbral elevado, de modo que solo los puntos blancos de los dados quedaran por encima del umbral, mientras que el resto quedara por debajo. Posteriormente, se procedió a identificar las componentes conexas mediante la función `cv2.ConnectedComponentsWithStats()`, extrayendo la cantidad de estas componentes identificadas que superen una cierta área umbral. Para determinar el número a exhibir, se esperó a que el mismo número fuera detectado durante un cierto número de frames, garantizando así la estabilización del resultado.

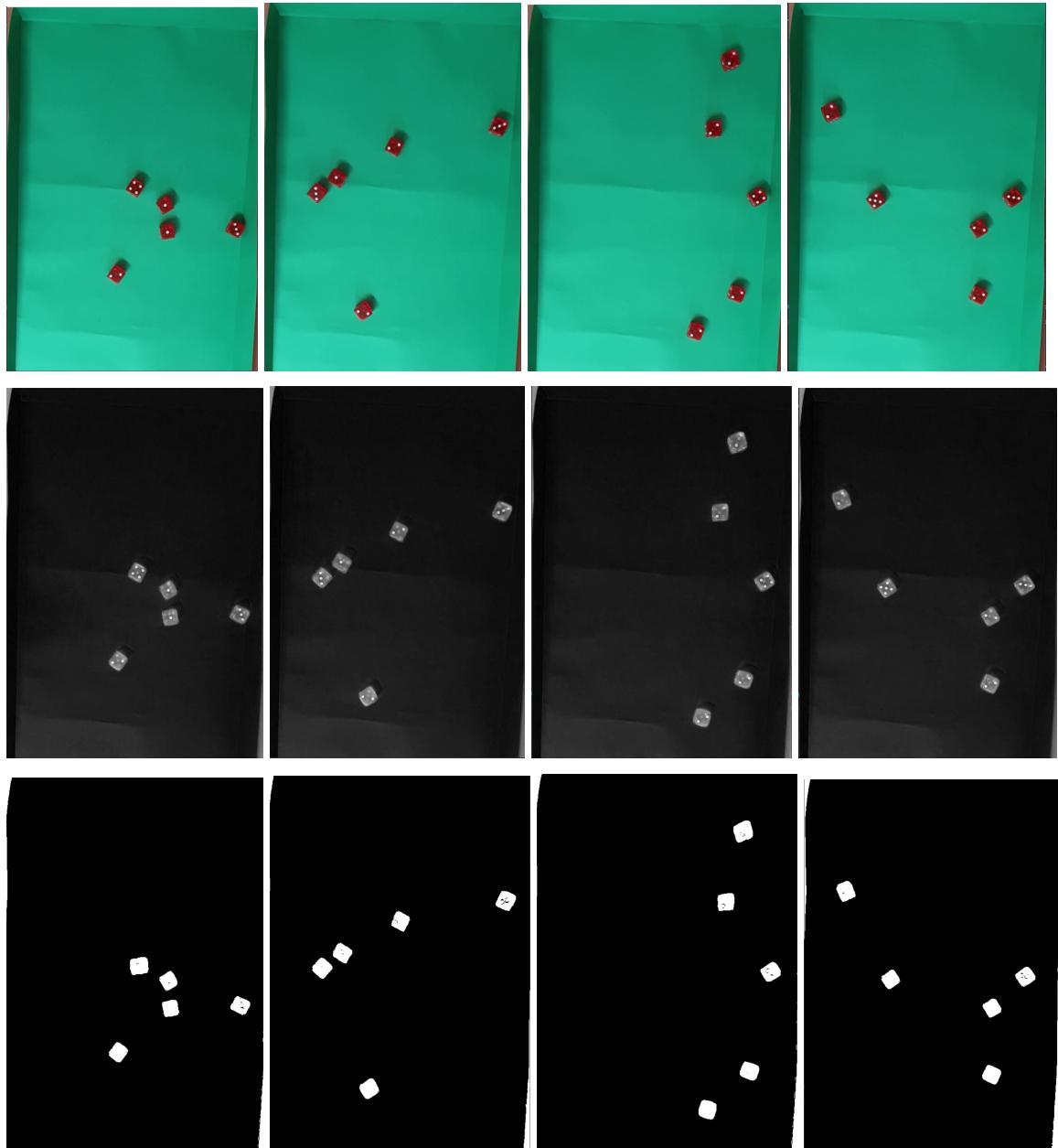


Figura 1. Fila superior: Cuadros originales del video. Fila del medio: Canal rojo de los mismos cuadros de la fila superior en escala de grises. Fila inferior: Binarización (umbral de 75) del canal rojo de los cuadros de la fila superior.

2. Ítem (b): Dibujar bounding boxes, los números y generar los videos.

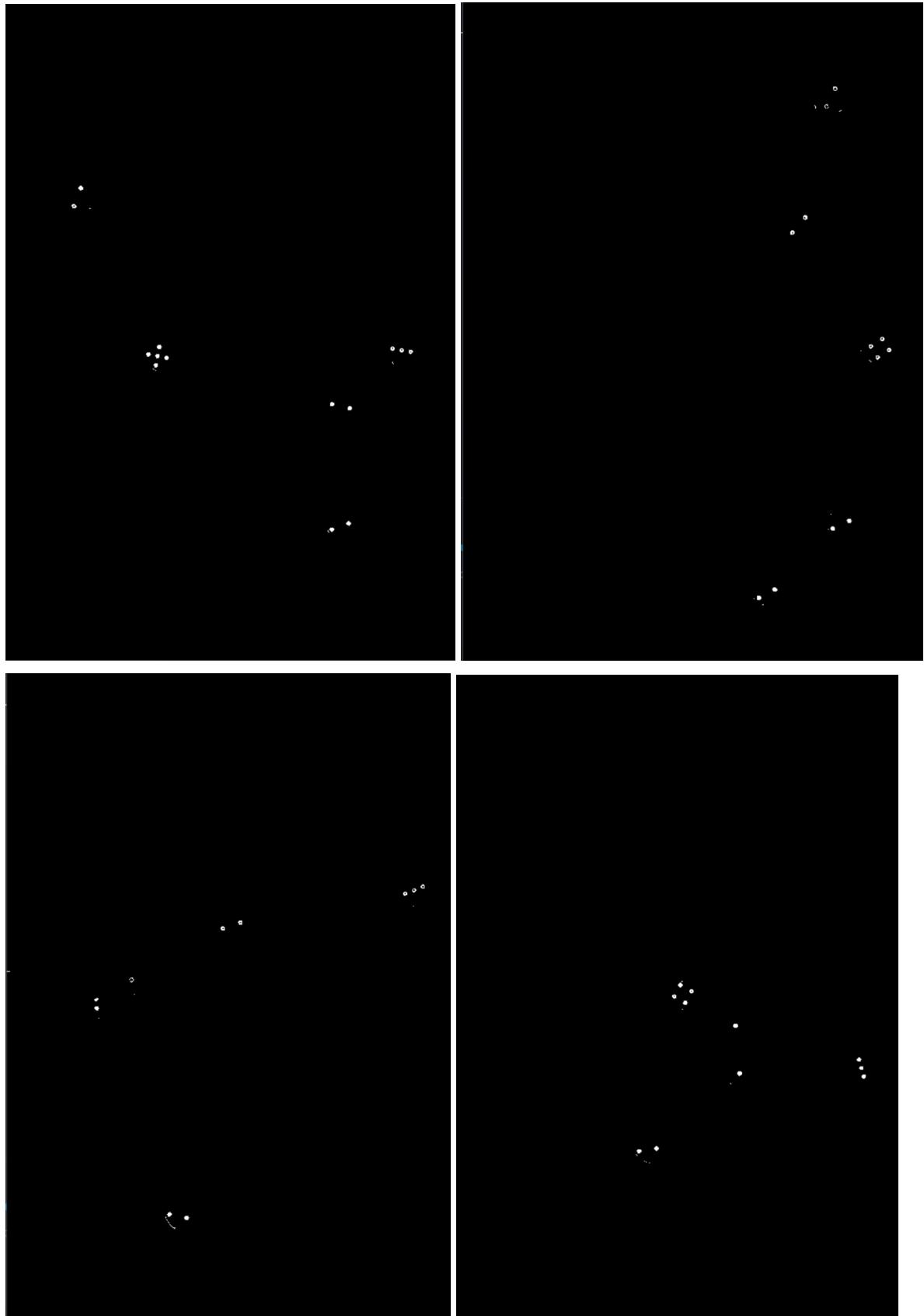


Figura 2. Canal rojo de los cuadros de la Figura 1 binarizada pero con un umbral más alto (197) para contar las componentes conexas que superen el área umbral y así obtener el número que muestra la cara superior del dado.

2.1. Dibujar bounding boxes y números

Una vez que el número se estabilizó, se emplearon las funciones de OpenCV, específicamente `cv2.rectangle()` y `cv2.putText()`, para trazar las bounding boxes alrededor de los datos y para añadir el número identificado, respectivamente.

2.2. Guardar los videos resultado

Tras el procesamiento del video, se implementó la función `cv2.VideoWriter()` para guardar los resultados. Con el propósito de sistematizar esta tarea para todos los videos, se encapsuló el algoritmo previamente detallado en una función. Posteriormente, se creó un diccionario con los nombres de los videos a procesar como claves y los nombres de los videos resultantes como valores. A continuación, se utilizó iterativamente la función para generar todos los videos resultantes.

3. Dificultades

Más allá de los desafíos técnicos inherentes a la implementación del código y la exploración de documentación y ejemplos, se suscitaron dificultades adicionales al buscar un conjunto de parámetros universalmente aplicable a todos los videos. Se afrontó la afortunada circunstancia de que el problema a resolver se presentó de manera indulgente en términos de condiciones. No obstante, se reconoce que, de haber variado el ángulo de la cámara, la iluminación sido más irregular, o el fondo menos homogéneo, se hubiera requerido la aplicación de una serie de métodos de procesamiento de imágenes, tales como aperturas, cierres, operaciones top-hat, erosiones, ecualización del histograma, o ajustes en la intensidad de la imagen. En este sentido, la adaptabilidad del algoritmo a condiciones menos favorables habría exigido un enfoque más sofisticado y diversificado en la manipulación de las imágenes capturadas.

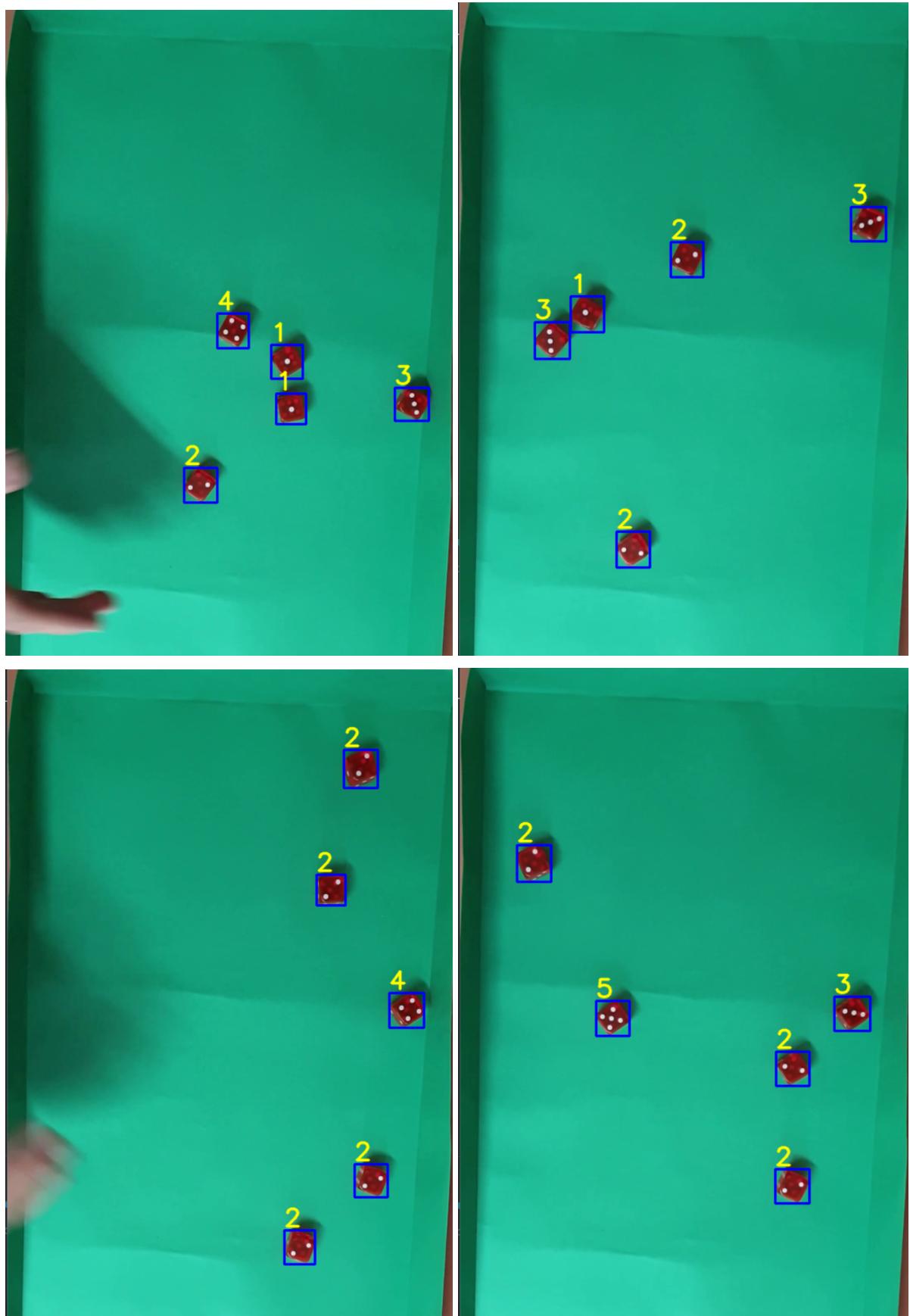


Figura 3. Resultado del procesamiento

4. Conclusiones

Se concluye que el enfoque más eficaz para procesar videos radica en un método de procesamiento cuadro a cuadro, almacenando la información únicamente del frame precedente. Este enfoque, centrado en la gestión secuencial de los frames, ha demostrado ser altamente eficiente para lograr el objetivo requerido.

Además, se subraya la importancia del uso estratégico del color en función del problema a resolver, destacando cómo esta táctica puede simplificar de manera significativa una tarea que inicialmente podría considerarse altamente desafiante. La selección inteligente de información basada en colores específicos, como se evidenció en la elección del canal rojo para la detección de datos, resalta la relevancia de adaptar las estrategias de procesamiento de imágenes a las características específicas de cada problema.

Tras llevar a cabo este trabajo práctico, se concluye que la visión por computadora se fundamenta principalmente en la aplicación de operaciones de intensidad de píxeles y morfológicas a las imágenes, para culminar con la obtención de una representación binaria que permite la identificación de componentes conexas y/o contornos, según la naturaleza del problema, y estos elementos identificados constituyen la información esencial subyacente en las imágenes, sobre la cual se aplican algoritmos lógicos con el fin de alcanzar los objetivos buscados.