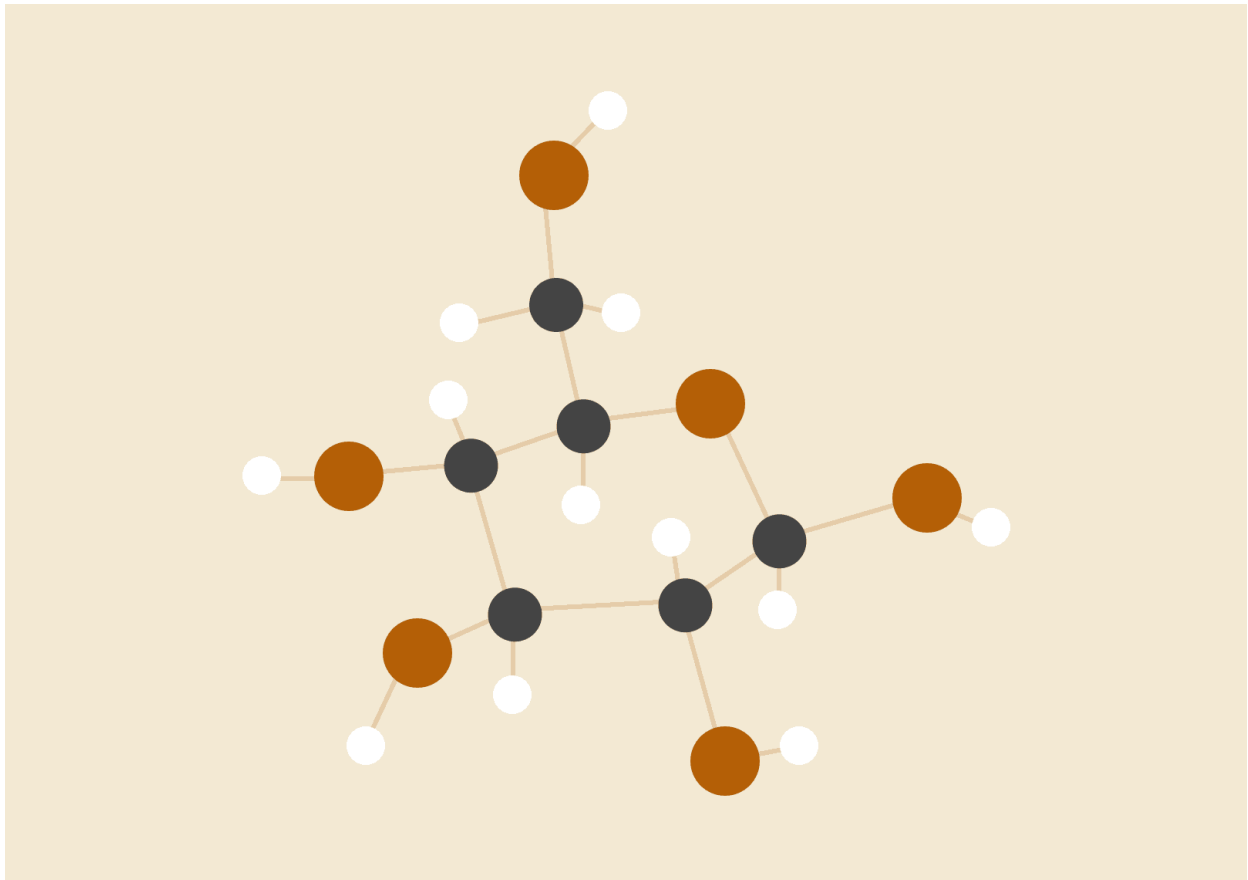


TRABAJO PRÁCTICO 3

IA4.4 Procesamiento de Imágenes I

Tecnicatura Universitaria en Inteligencia Artificial



Brizuela Cipolletti Sofía

Fontana Gustavo Julián

05/12/2023

Universidad Nacional de Rosario

Facultad de Ciencias Exactas, Ingeniería y Agrimensura

INTRODUCCIÓN

Se plantea el problema de los cinco dados, el que consiste en detectar dados en un video y poder identificar el número que muestran en su cara superior al momento en que se detienen.

Se solicita desarrollar un algoritmo para el propósito mencionado, informando cada paso del procedimiento, y además generar un video grabado cuando se produce la detección de los dados. El mismo debe contener un bounding box de color azul para cada dado y el numero asociado a el.

PROCEDIMIENTO

PROBLEMA 1 – Cinco dados

Se implementan tres funciones con el fin de detectar los dados y los números asociados a ellos:

1. Función ‘contar_huecos’: recibe una imagen (ROI del dado) y detecta los contornos internos en la imagen. Los contornos internos son contados, obteniendo su total al iterar sobre la jerarquía de contornos. La función retorna el total de contornos.
2. Función ‘encontrar_datos’: el objetivo de la función es segmentar la imagen a través del espacio HSV. Para eso se utilizan dos mascarar con umbrales de color, saturación y tono adecuados para detectar los dados y evitar el resto de los objetos. Luego las mascarar son aplicadas a la imagen original obteniéndose la imagen transformada. Esta nueva imagen es pasada a escalas de grises, binarizada y sometida a filtros morfológicos, apertura y clausura, que son implementados para mejorar la definición de los contornos de los dados.
3. Función ‘video_segmentado’: recibe un video, lo carga y lo reproduce. Durante su reproducción se realizan diversas tareas:
 - Se convierte a escalas de grises los frames y luego se binarizan. Luego se detecta la cantidad de pixeles blancos con la función `cv2.countNonZeros` y de pixeles negros por la diferencia entre el total de pixeles y los pixeles blancos. La detección del total de pixeles negros permite establecer un umbral óptimo para detectar el cambio de intensidad cuando los dados se detienen y quedan presentados limpiamente.
 - Luego, se itera filtrando los frames según un nivel de intensidad de pixeles negros, pasando una copia del frame a la función ‘video_segmentado’ que devuelva el frame procesado del cual se obtienen sus componentes conectadas.

La componentes conectadas sirven para extraer las coordenadas de la ROI de cada objeto con un determinado área. La ROI obtenida es pasada como argumento de la función ‘contar_huecos’ que retorna el total de contornos internos del dado, lo que represente el valor asociado a el.

Estas mismas coordenadas extraídas sirven para dibujar los rectángulos y colocar las etiquetas de los valores correspondientes a cada dado.

- Por último, los frames que no son filtrados son mostrados de igual manera en el video en reproducción. La función muestra el video de la tirada de dados, a la vez que dibuja los bounding box sobre las ROI detectadas y coloca la etiqueta del valor del dado que se muestra en la cara superior. A la par que se realiza la salida del video, el procesamiento de los frames y el etiquetado de los dados, se realiza una copia del video original con la detección de los dados y los valores que presentan.

El conjunto de estas funciones permiten llevar a cabo la resolución del problema, detectando los dados y sus valores en los videos proporcionados.

Dentro de la última función descrita se encuentran líneas de código comentadas, que en caso de ser requeridas, pueden ser utilizadas para mostrar diferentes pasos del procesamiento de los frames y los resultados parciales.

Resultados del procesamiento:



figura 1 Resultado de aplicar la binarización sobre un frame (mayor cantidad de pixeles negros)

En este caso se puede ver que cuando la imagen está en movimiento, la cantidad de pixeles negros va a aumentar por la sombra que genera la mano.



figura 2: Resultado de aplicar la binarización sobre un frame (menor cantidad de pixeles negros)

En este otro caso se puede ver que la cantidad de pixeles negros es menor, y eso sucede cuando los dados están quietos, lo que permite observarlos de una manera más clara.

Aclaración: la binarización solo se utiliza para comparar los umbrales de los frames y saber cuando los dados están quietos, el procesamiento y segmentación lo realizan otras funciones.



figura 3: Resultado de segmentar en el espacio HSV

La segmentación del espacio HSV permite quedarse solo con los dados, ya que se intenta detectar los colores rojos en tonos claros y oscuros que solo coinciden con ellos. Además, esta segmentación permite deshacerse de las transparencias que pueden resultar problemáticas durante la detección de contornos internos.

Si no nos deshacemos de las transparencias, los números de las caras inferiores podrían ser detectados como contornos superiores erróneamente.

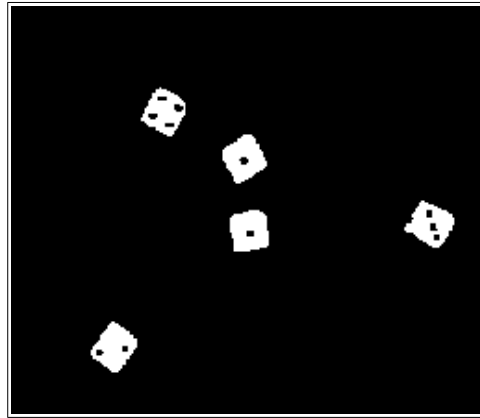


figura 4: Imagen binaria con morfología

El procesamiento de la imagen con la segmentación del espacio HSV y los diferentes filtros aplicados, devolvieron una imagen transformada bastante buena, lo que permitió una detección de contornos internos y bounding box bastante clara.

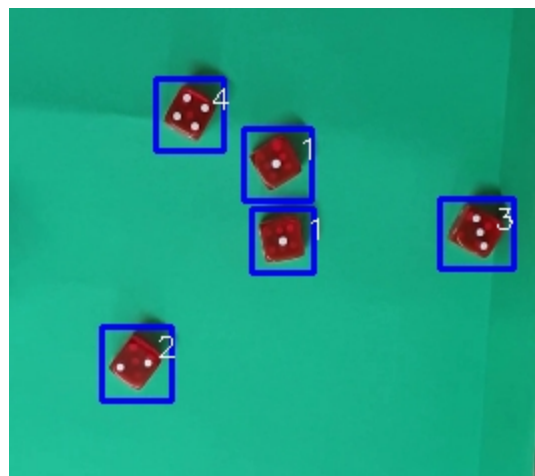


figura 5: Detección de los dados y clasificación numérica

La detección final de los dados y sus respectivos valores, se visualiza en el instante en que los dados dejan de moverse. Esta detección es susceptible al movimiento, y desaparece cuando se produce un cambio de intensidad en los píxeles.