

Trabajo Práctico N° 3. - Procesamiento de Imágenes - TUIA.

Integrantes:

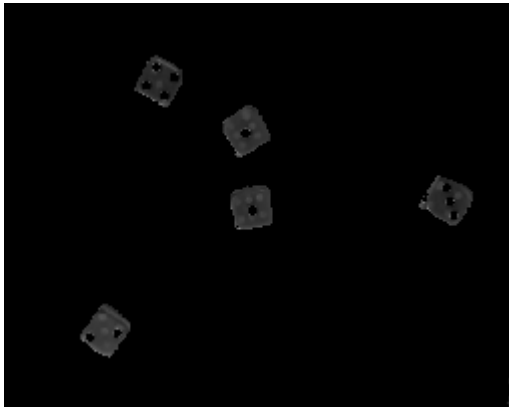
- Mateo Gravi Fiorino
- Tomás Navarro
- Ramiro Sagrera
- Alejo LoMenzo

En este script de Python, la implementación se centra en el procesamiento de videos para la detección de dados en reposo, la lectura de los números presentes en ellos y la generación de un nuevo video con resaltado y etiquetas. Para llevar a cabo estas tareas, se utiliza principalmente la biblioteca OpenCV.

El programa comienza importando las bibliotecas necesarias, principalmente OpenCV (`cv2`) y NumPy (`np`). Además, se define una función `filter_color_hsv` que desempeña un papel crucial al realizar una filtración de color en el espacio de color HSV. Esta función es esencial para aislar los dados, los cuales son de tono rojo en el video.

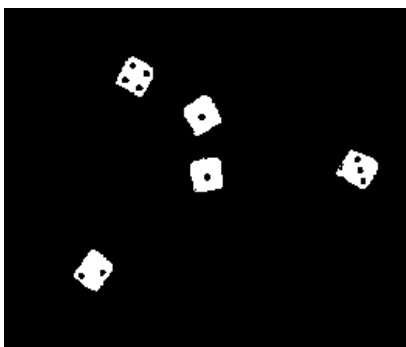
El archivo de video especificado en la variable `video_path` se abre mediante `cv2.VideoCapture()`, y se extraen propiedades clave como el ancho, alto y la velocidad de fotogramas (fps). Este conocimiento preliminar es fundamental para el procesamiento posterior.

El bucle principal del programa recorre cada fotograma del video. En cada iteración, el fotograma se redimensiona para facilitar el procesamiento, se aplica un filtro de color para resaltar la presencia de dados y se pasa a escala de grises.



El uso de la función `connectedComponentsWithStats` de OpenCV permite identificar componentes conectados en la imagen, que en este contexto representan los dados. Estos componentes se filtran con criterios específicos de área y relación de aspecto para asegurar una detección precisa.

Luego se aplica cierre morfológico sin dilatación sobre una máscara previamente creada para mejorar la calidad de la máscara y encontrar la región de interés para cada dado. Esta máscara se umbraliza y se buscan los contornos de la misma.

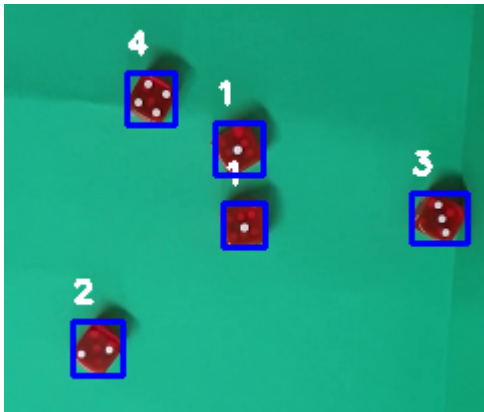


Estos contornos encontrados se filtran por área y se guardan sus centroides para identificar cuando se frenan los dados. Se comparan los centroides de los dados en el fotograma actual con los centroides en el fotograma anterior. Si la diferencia entre ellos es menor que un umbral predefinido y se detectan cinco dados, se considera que los dados están en reposo.

Luego, el código incluye una sección para el reconocimiento de números en cada dado. Se itera dentro de los contornos validados y se procesa nuevamente con binarización la región del dado para contar los puntos presentes, que representan el número en el dado.



Como último paso para la detección del número del dado, se buscan los contornos de las imágenes anteriores y se aplica un filtro por área. En esta situación, se resaltan los dados mediante un bounding box azul y se etiqueta el número reconocido.



Finalmente, se crea un nuevo video llamado "dados_quietos.mp4", en el cual los fotogramas previamente redimensionados que cumplen las condiciones de dados en reposo se resaltan y etiquetan. El código cierra adecuadamente el video original y las ventanas de visualización, liberando los recursos utilizados.