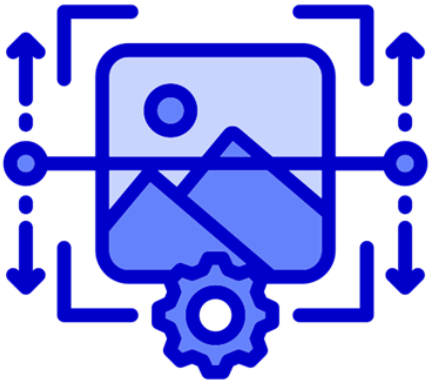
TP3 - Procesamiento de Imágenes 1

Tecnicatura Universitaria en Inteligencia Artificial.



Docentes:

- Gonzalo Sad
- Julián Alvarez
- Juan Manuel Calle

Integrantes (Grupo 11):

- Marcela Flaibani (F-3793/1)
- Facundo López Crespo (L-3339/1)
- Daniela Dito (D-4322/2)

Índice

Enunciado

TRABAJO PRÁCTICO N° 3 - Año 2024 - 2° Semestre

Problema 1 - Cinco dados

Las secuencias de **video** *tirada_1.mp4, tirada_2.mp4, tirada_3.mp4 y tirada_4.mp4* corresponden a tiradas de 5 dados. En la Figura 1 se muestran los dados luego de una tirada. Se debe realizar lo siguiente:

a. Desarrollar un algoritmo para detectar **automáticamente** cuando se detienen los dados y leer el número obtenido en cada uno. Informar todos los pasos de procesamiento.
b. Generar videos (uno para cada archivo) donde los dados, mientras estén en reposo, aparezcan resaltados con un bounding box de color azul y además, agregar sobre los mismos el número reconocido.

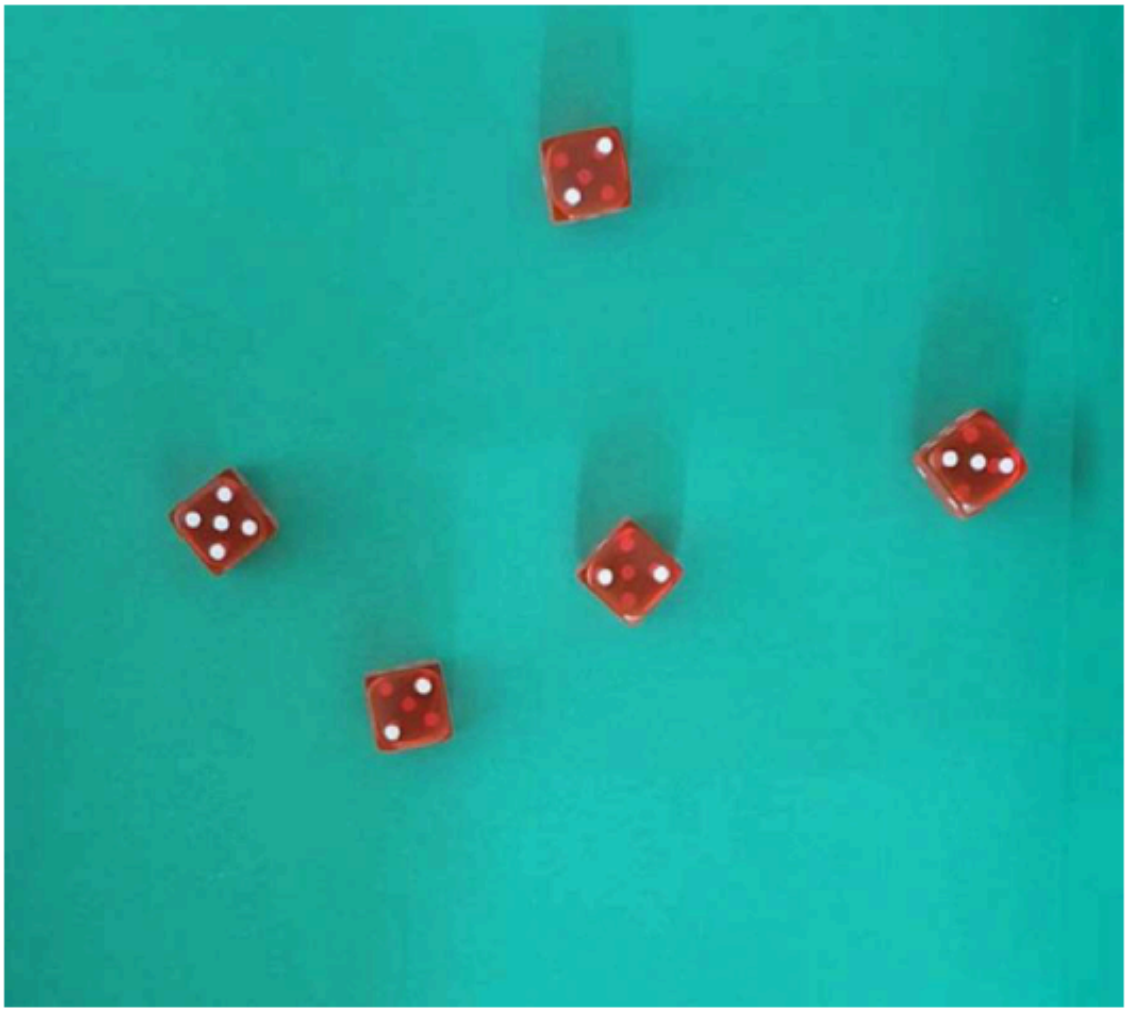


Figura 1 – Dados luego de una tirada.

Resolución

Parte a

Procesamiento del video

Los frames se procesan en tiempo real (no se guardan) en la función video_process(), reduciendo el uso del disco.

Se analizan los frames en forma consecutiva tratando de detectar el momento en que los dados quedan detenidos en el video.

Para ello primero se verifica que en la imagen haya 5 dados y luego que sus posiciones no hayan variado al menos en 5 frames consecutivos.

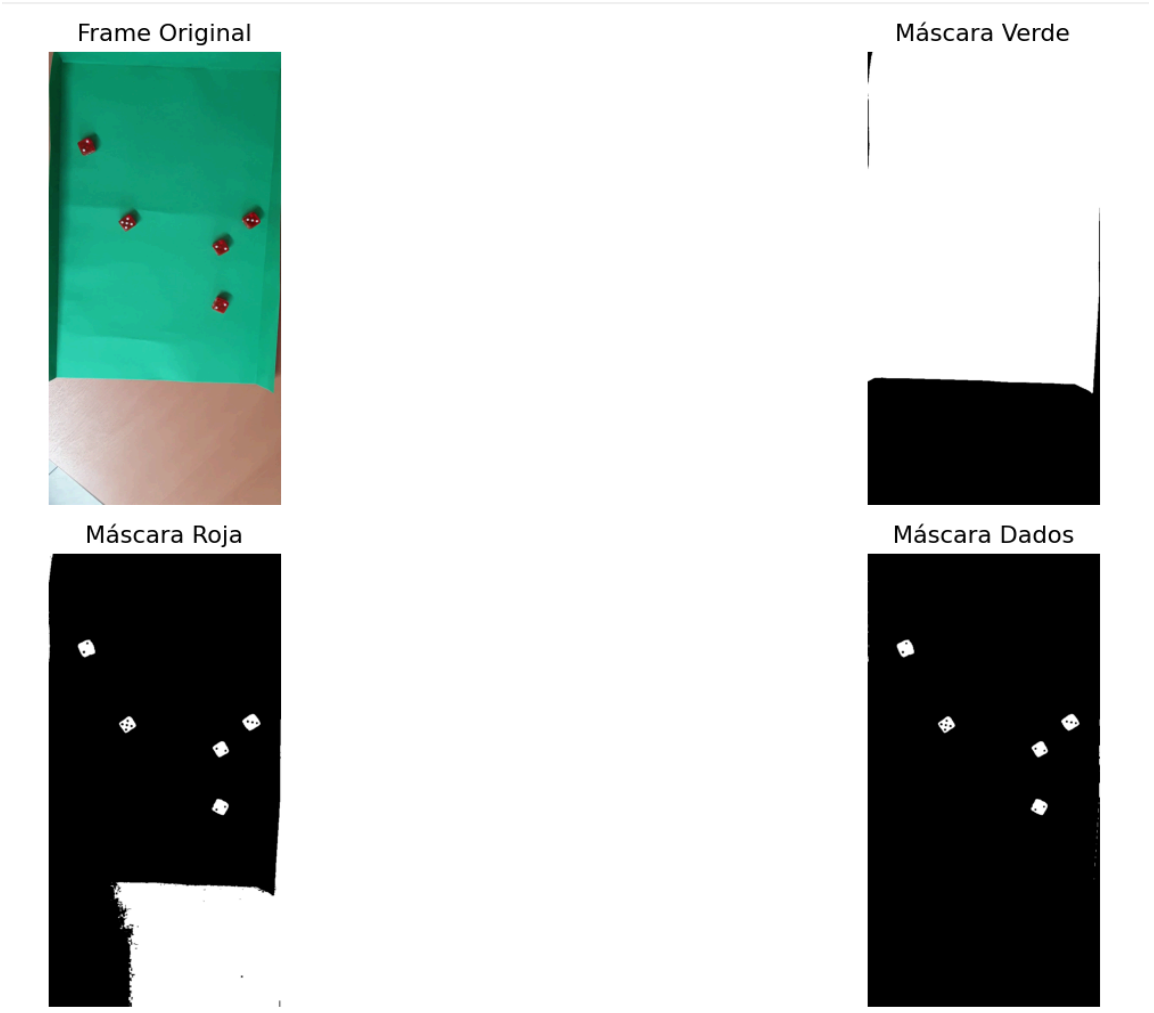
Este proceso se traduce en el código que se explica a continuación.

Se toma la imagen del primer frame (frame == 0) y se genera una máscara para filtrar todo lo que no es verde en la imagen. Esta máscara será usada para detectar la zona del paño verde sobre la mesa en todas las secuencias de frames.

La función detect_red_dados() genera sobre cada frame, una máscara para filtrar todo lo que no sea rojo, para detectar los dados.

Para generar ambas máscaras se utiliza el espacio HSV. Los umbrales se determinan empíricamente.

La intersección de ambas máscaras binarias permite detectar los dados rojos que se encuentran sobre el paño verde (Máscara Dados en la figura).

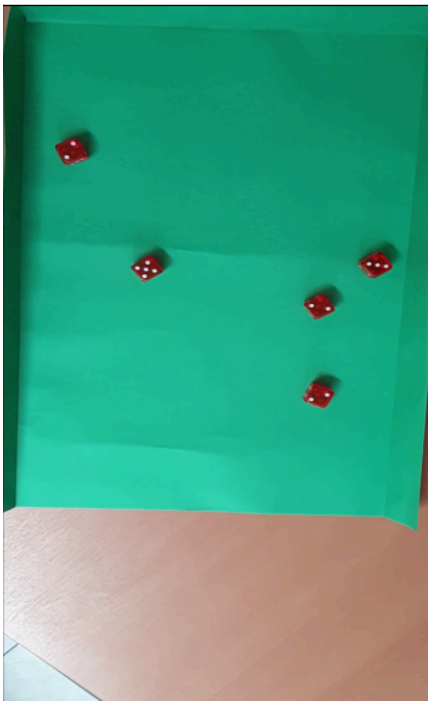


Con la imagen obtenida, fondo negro y zonas de interes (ROI) en blanco, se busca detectar los contornos de los dados, utilizando la función findContours de OpenCV. Se filtran los componentes por área, descartando ruido e imágenes de mayor tamaño que los dados (entre 3500 y 5500).
Los límites entre los que se encuentran los contornos de los dados, se determinan examinando los valores en los distintos videos.

Coords: [(796, 1243)] - Área: 3921.0
Coords: [(796, 1243), (798, 962)] - Área: 3659.5
Coords: [(796, 1243), (798, 962), (368, 842)] - Área: 3738.5
Coords: [(796, 1243), (798, 962), (368, 842), (940, 830)] - Área: 4196.0
Coords: [(796, 1243), (798, 962), (368, 842), (940, 830), (178, 472)] - Área: 4553.5

Si se detectan componentes con las características de un dado, se calcula su centroide con el método cv2.moment. Todos los centroides detectados en ese frame, se guardan en una lista (dados_coords).
Para verificar que los 5 dados están en reposo, se comparan las coordenadas actuales con las 5 anteriores, que se guardan en una cola (queue_coords) de longitud máxima igual 5.
Si se cumple esta condición, se guarda la siguiente información que será utilizada posteriormente para la grabación del video editado:

frame_number_start: el nro. frame donde comienza la detención (5 frames antes del actual).
frame_number_end: el nro. frame actual, que se irá modificando si los dados siguen detenidos en los frames siguientes.
frame: frame actual detenido
mask_dados_end: la imagen con los dados detenidos.
dados_coords_end: las coordenadas de los 5 dados estando en reposo.



En el bucle se verifican todas las condiciones posibles para determinar el comienzo y fin de la detención de los dados:

- Si hay exactamente 5 dados detectados,
- Si hay 5 dados y están quietos,
- Si estuvieron detenidos y se volvieron a mover,
- Si no hay 5 dados en la imagen
- Si termina el video con los dados detenidos
- Si los dados nunca estuvieron quietos

Procesamiento de los dados

Una vez terminado el procesamiento del video, con la información recolectada, se procesan cada uno de los dados, para determinar su puntaje.

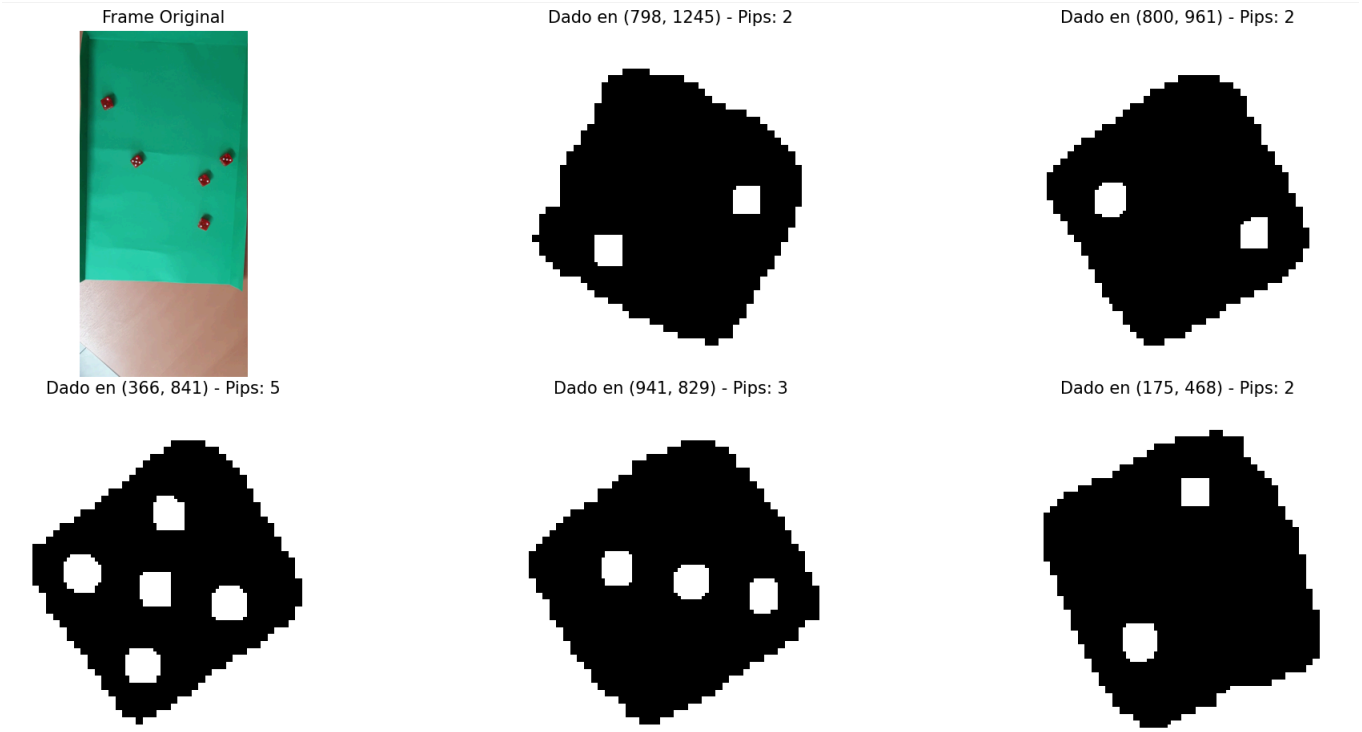
A partir de la imagen binaria con los dados detenidos y de las coordenadas de los centroides, se recortan las ROI de cada dado.

En este punto, el crop de cada dado tiene el fondo y los puntos en negro y la imagen del dado en blanco. Para aplicar componentes conectados, se necesita que las zonas de interés estén en blanco (255) por lo que se invierte la imagen.

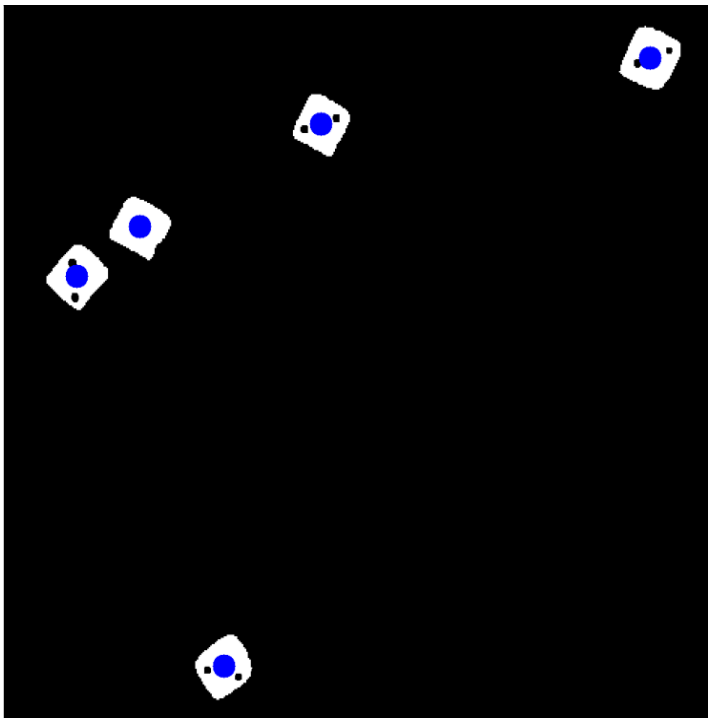
Se aplica morfología (apertura con kernel 5x5) para eliminar ruido pequeño en el fondo.

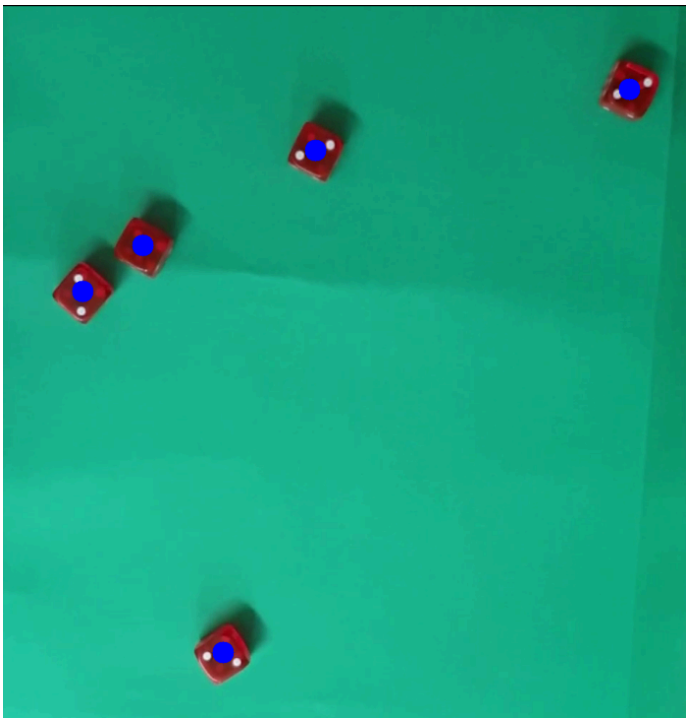
Con componentes conectados, filtrando por área, se detecta y cuenta el puntaje de cada dado.

Se verifica que el puntaje se encuentre entre [1,6].



Las coordenadas del centroide, la imagen procesada y el puntaje de la cara de cada dado, se guardan en un diccionario (dado) que será anexado a una lista que contiene la información de los 5 dados (dados_info).





Resultado de la tirada - Generala

Con la información del puntaje de los 5 dados, se determina el resultado del juego:

- ¡Generala! Cinco dados iguales.
- ¡Póker! Cuatro dados iguales.
- ¡Full! Tres y dos dados iguales .
- ¡Escalera! Todos los dados con puntajes consecutivos (1-5 o 2-6)
- No se logró una combinación especial.

Se utiliza la clase **Counter** de la librería **collections** para contar cuántas veces aparece cada valor (pip) en la lista **valores_dados.** El resultado es un diccionario donde las claves son los valores de los dados y los valores son las cantidades de veces que esos valores aparecen.
Se ordena el diccionario conteo en una lista de tuplas, donde cada tupla contiene un valor y su cantidad de ocurrencias. El método **most_common()** devuelve una lista ordenada de las ocurrencias de mayor a menor.

Parte b

La función **video_record** utiliza la información generada en el procesamiento previo del video, **videos_info**, que contiene la información necesario de cada una de las tiradas.
Se reproduce el video y solo procesan los fotogramas cuyo número de cuadro se encuentre entre **frame_number_start** y **frame_number_end,** que delimitan el lapso en que los dados comienzan y terminan de estar en reposo. Este proceso consiste en agregar a los frames un recuadro alrededor de cada dado y su puntaje como etiqueta.
Los videos generados, uno por tirada, se guardan en la carpeta **videos_procesados**.

Repositorio

https://github.com/Facunditos/PDI-TUIA-TP3-LopezCrespo-Flaibani-Dito.git