



FACULTAD DE CIENCIAS EXACTAS, INGENIERÍA Y AGRIMENSURA

Tecnicatura Universitaria en Inteligencia Artificial

Procesamiento de Imágenes I: Trabajo Práctico 3

Alumnos:

Facundo Geuna - G-5905/6

Pedro Casado - C-7439/1

Máximo Alva - A-4671/1

**Grupo: 5
Año: 2024**

Introducción

Con el propósito de resolver este trabajo práctico utilizando Python, implementamos las siguientes librerías:

- [cv2](#)
- [numpy](#)
- [matplotlib](#)
- [os](#)

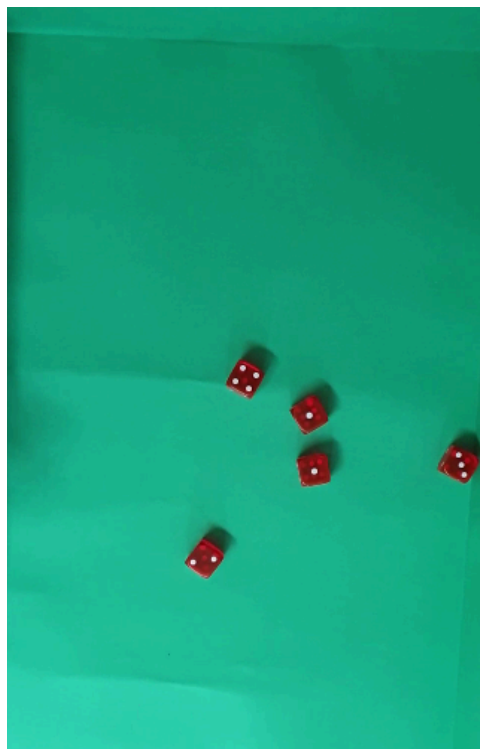
Problema 1 - Cinco dados

El objetivo de este desafío es desarrollar un algoritmo para detectar automáticamente cuando se detienen los dados y leer el número obtenido en cada uno. Para esto, se utilizan cuatro videos de lanzamientos de dados. Se debe, además, generar videos donde los dados estén resaltados y sobre los mismos se visualice el número reconocido.

Pasos del Algoritmo:

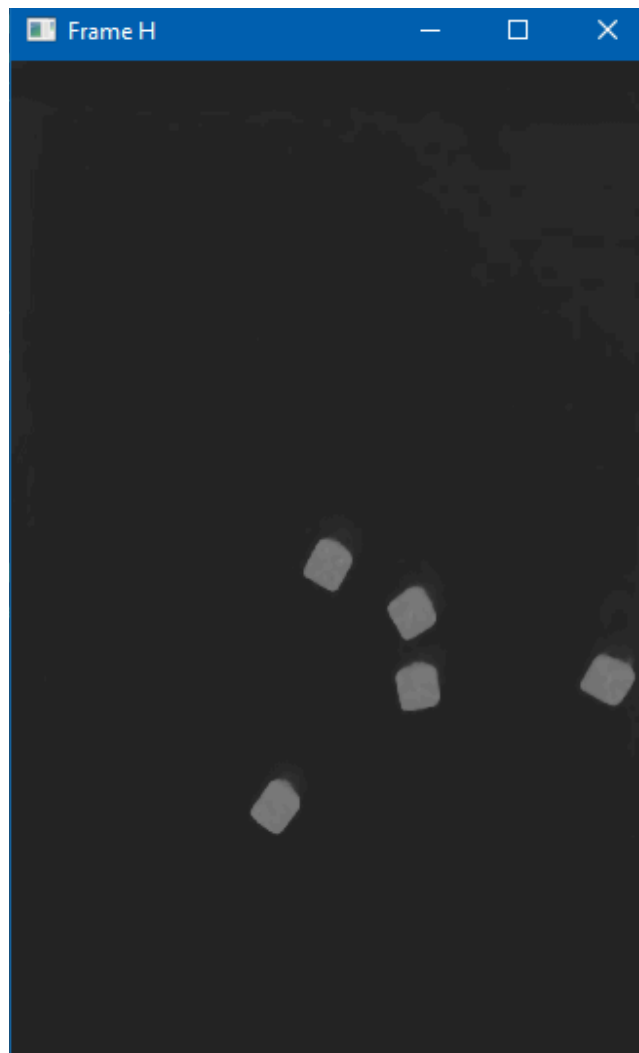
1. Lectura y preprocesamiento de los videos:

Se leen los videos de entrada y se transforma frame a frame en una imagen de color HSV para luego aplicarle un umbral al canal verde. Se utiliza componentes conectadas para detectar el área más grande de color verde, que corresponde al paño donde caen los dados. Ese área se recorta de todos los frames y se guardan todos los frames recortados en diferentes carpetas para cada video.

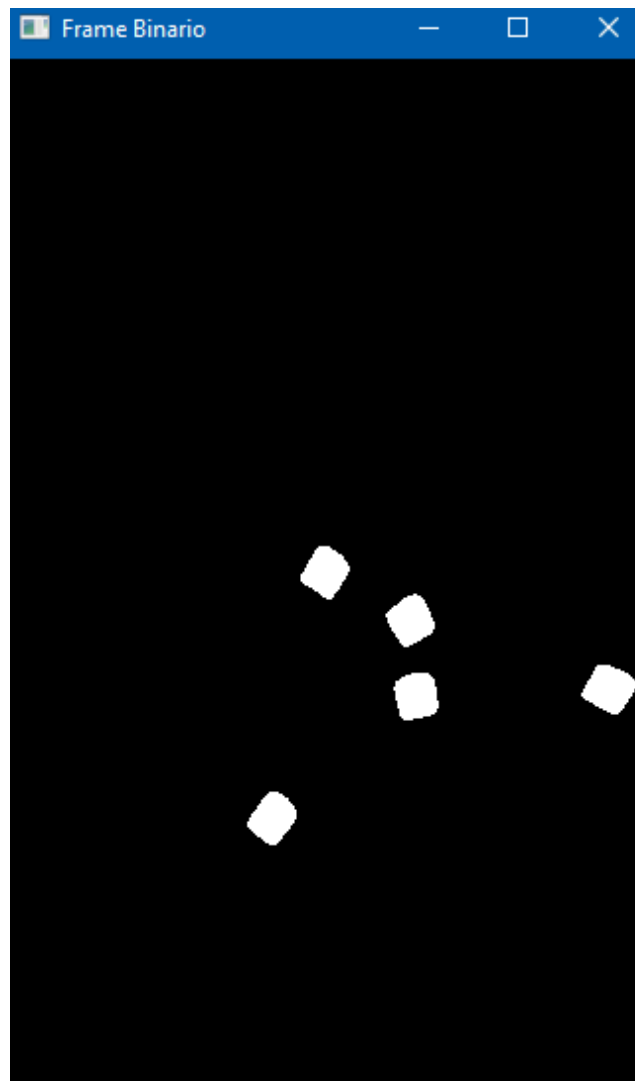


2. Detección de dados en reposo:

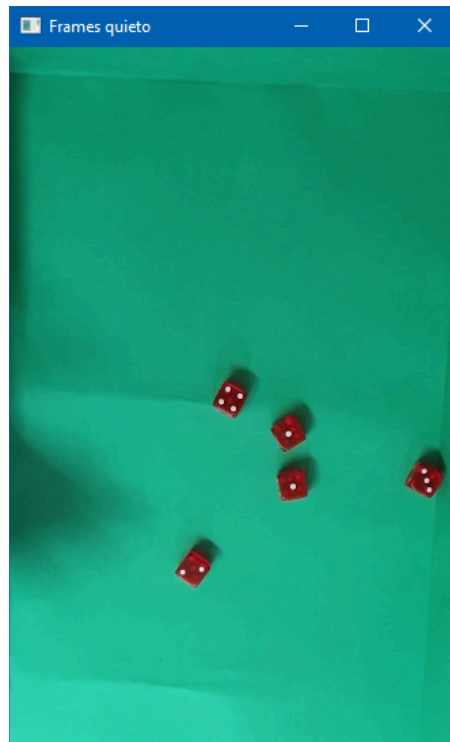
Se detecta el momento en que los dados están en reposo en cada directorio de frames. Para eso se utiliza la función `hsv_bin` que toma cada frame y lo transforma en HSV para después aplicar un umbral y binarizar la imagen. Con esa imagen binaria se utilizan componentes conectadas para identificar los frames donde hay 6 componentes (fondo + 5 dados). Con estos frames, se compara la posición de los dados y si la diferencia entre un frame y el siguiente es menor a cierto umbral durante suficientes frames consecutivos, se considera que los dados están en reposo. Estos frames se guardan para continuar el análisis.



Frame utilizando solo el canal h



Frame después de la binarización



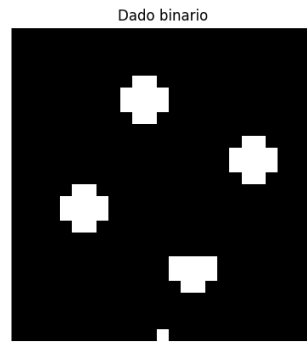
Frame encontrado con dados quietos.

3. Reconocimiento de números en los dados:

Detecta el número de cada dado. Se vuelve a binarizar los frames quietos con la función `hsv_bin`. Se utilizan componentes conectadas para detectar los dados y se hace un recorte de la imagen original de cada dado. Después se itera cada uno de ellos y pasando el frame a escala de grises se aplica un umbral para binarizar la imagen. Esta imagen binaria pasa de nuevo por componentes conectadas para encontrar todos los puntos de los dados, filtrando por área se cuentan los puntos de cada dado y se devuelve el valor.



Captura de uno de los dados encontrados



Dado después del umbralizado binario

4. Generación de video con detecciones resaltadas

Se utilizan los frames iniciales de todo el video original, sumado a los frames donde conocemos que los dados están quietos y el número de cada dado en esos frames. Con esto se construye un video donde en el momento que los dados están quietos, se dibujan los bounding boxes de cada uno y se etiqueta el número correcto. Luego guarda el video en formato .mp4 en la misma carpeta del programa.



Frame etiquetado dentro del video final

Conclusión:

El código implementa un flujo automatizado para procesar los videos de tiradas de dados. Divide el problema en pasos (segmentación, detección de reposo, reconocimiento de números, y generación de video). Cada paso utiliza técnicas de procesamiento de imágenes como componentes conectadas, umbralización, y análisis de cuadros consecutivos para lograr el objetivo. El producto final es un video que resalta y anota los dados con su número.