

Facultad de Ciencias Exactas, Ingeniería y Agrimensura

Tecnicatura Universitaria en Inteligencia Artificial



PROCESAMIENTO DE IMÁGENES

IA 4.4

Informe N°3

Año: 2025

Grupo: 1

Integrantes:

- Calabozo, Nicolas - C-7551/5
- Perez, Sebastián - P-5334/1
- Lapolla, Martín Facundo - L-3372/3

Fecha de presentación: 01/12/2025

Docentes:

- Ing. Gonzalo Sad
- Ing. Juan Manuel Calle
- Ing. Joaquín Allione

Introducción

El presente informe corresponde a la resolución del Trabajo Práctico N°3, cuyo propósito es aplicar conceptos de segmentación y detección de objetos en videos de tiradas de datos.

Interpretando los videos como una secuencia de imágenes estáticas (frames o cuadros), el objetivo es elaborar un algoritmo capaz de segmentar datos automáticamente, y detectar el valor de su cara una vez se encuentren en reposo. Para ello se utilizaron técnicas de detección por color, morfología, análisis de contornos, detección de forma y seguimiento temporal cuadro a cuadro.

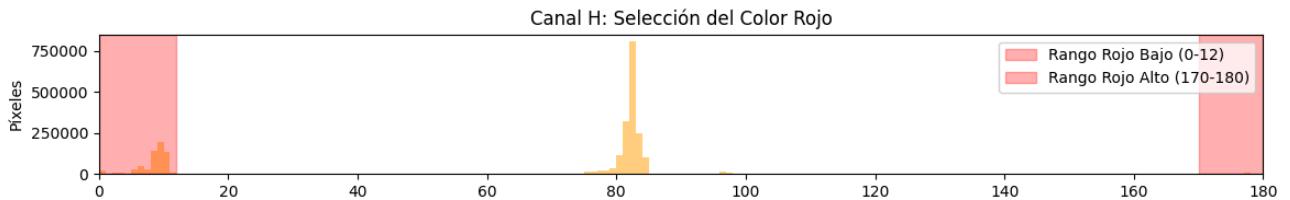
La implementación se realizó utilizando la biblioteca OpenCV, junto con NumPy y Matplotlib, asegurando la visualización de los resultados en cada etapa del procesamiento.

Etapa 1: Segmentación por Color

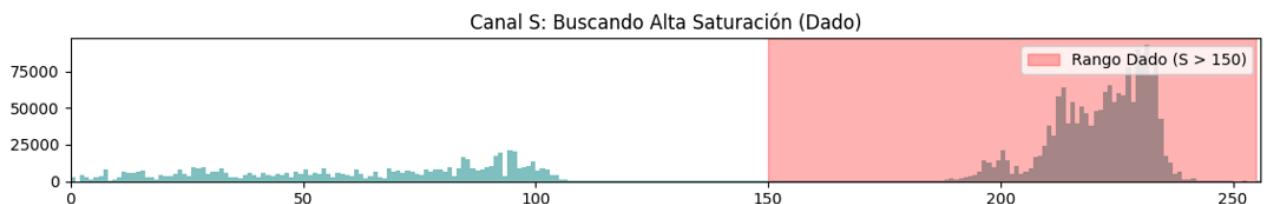
En esta primera etapa, nos centramos en poder aislar los dados color rojo del fondo verde no uniforme. Para lograrlo, implementamos un evento on-click que nos permite, dentro de un frame, extraer los valores HSV. La función traduce el click a una coordenada específica, convierte los valores BGR del pixel a HSV y luego los muestra por la terminal.



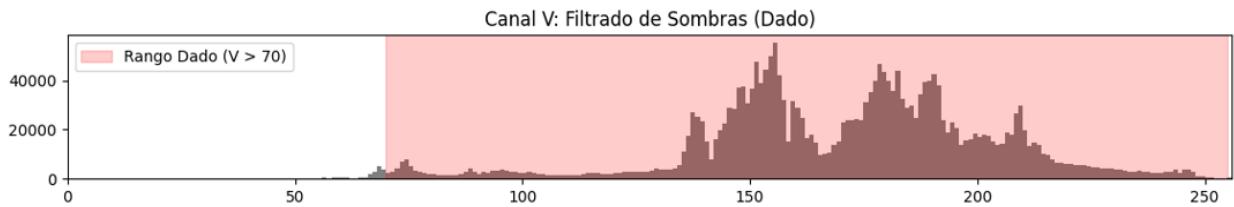
Este primer acercamiento a modo exploratorio nos permitió observar que los valores de tono rojo se encuentran al comienzo y al final de la banda de tonos.



Más específicamente, se encuentran en el rango [0,12] y el rango [170,180] de intensidad. Podemos observar un pico en el centro, en el rango [70,90] que corresponden a los tonos verdes de la imagen.



De la misma manera, podemos segmentar los colores vibrantes según su saturación. Dicho intervalo está definido en [150 , 255]. Dentro de este intervalo encontraremos los valores de saturación de los colores verdes y rojos.



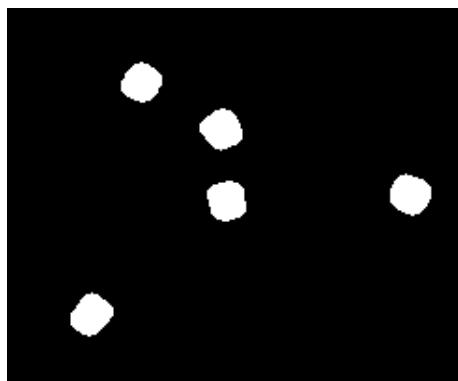
Finalmente, se establece un rango mínimo de brillo [70,255], esto nos permite ignorar todo lo que sea demasiado oscuro sea causado por sombras, o zonas oscuras de cada frame.

De esta manera, obtenemos los rangos para los tonos de rojo con valores bajos que van de [0,150,70] a [12,255,255] y para tonos de rojo con valores altos, de [170,150,70] a [180,255,255]. La unión de ambos rangos representa nuestra máscara de color.

Esta máscara, al aislar solamente los tonos de rojo, presenta huecos o pequeñas discontinuidades donde se encuentran los puntos blancos del dado, por lo que para consolidar cada dado como un único objeto segmentado se aplican operaciones morfológicas:

1. Clausura (Close) con un elemento estructural elíptico de tamaño (13×13) para llenar huecos internos.
2. Apertura (Open) con el mismo kernel para eliminar ruido aislado.

Filtrado por color + Morfología



Etapa 2: Tratamiento de imagen para segmentar contornos

Una vez obtenida la máscara binaria que aísla las regiones correspondientes a los datos, se procede al tratamiento de la imagen para segmentar correctamente cada dato como un objeto independiente y extraer la información geométrica necesaria para las etapas posteriores.

A partir de la máscara ya procesada mediante operaciones morfológicas, se realiza la detección de contornos utilizando únicamente los contornos externos, con el objetivo de identificar cada dato como una única región conectada. Esta elección evita considerar contornos internos o secundarios que puedan aparecer debido a imperfecciones en la segmentación o a variaciones de iluminación.

Para cada contorno detectado se calcula su área, descartando aquellos cuya superficie no se encuentre dentro de un rango preestablecido de 250 a 500 píxeles cuadrados. Este filtrado por área permite eliminar pequeñas regiones de ruido residual y evitar considerar objetos que no sean datos.

Sobre cada contorno válido se encuentra una bounding box, obteniendo así las coordenadas (x, y) y las dimensiones (w, h) del dato en la imagen. A partir de esta se determina además un centro del objeto, donde la coordenada X del centro corresponde a la suma entre el valor de 'x' y la mitad del ancho 'w', y la coordenada Y del centro corresponde al valor de y más la mitad del alto 'h'.

Este centro será utilizado posteriormente para evaluar el desplazamiento de los datos entre frames consecutivos y así detectar el estado de reposo.

Con la bounding box se recorta también el ROI correspondiente a cada dato directamente desde el frame original. Este ROI contiene exclusivamente la imagen del dato y será utilizada más adelante para el conteo de puntos blancos que determinan el valor de la cara visible.

Con fines tanto visuales como prácticos, a la bounding box se le agrega un margen fijo o padding en todas sus direcciones de cinco píxeles. Este margen permite:

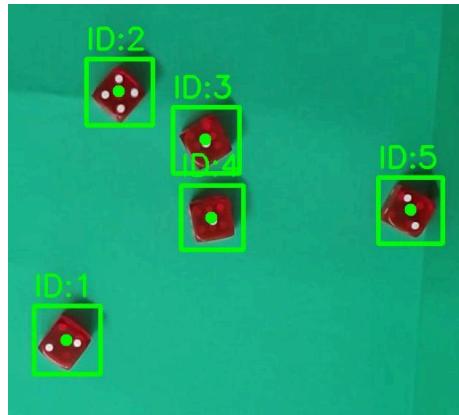
- Asegurar que el ROI contenga completamente al dato aún ante pequeñas imprecisiones en la segmentación
- Mejorar la legibilidad de las anotaciones gráficas (bounding boxes y texto) superpuestas en el video final.

Como resultado de esta etapa, para cada frame se construye una estructura de datos, que agrupa toda la información relevante de cada detección. Cada elemento de esta estructura contiene: el ROI del dato, el bounding box con padding, el centro del objeto, el área del contorno, un campo puntos, inicializado en cero, que será actualizado únicamente cuando el dato sea detectado en estado de reposo.

Esta estructura permite mantener organizada la información de cada dado y facilita su utilización en las etapas siguientes de detección de reposo y cálculo del valor de la tirada.

De manera preliminar, mientras los dados no están en reposo, tendremos la siguiente representación gráfica dentro del video.

Detección preliminar de datos



Etapa 3: Detección de Dados en reposo

Una vez segmentados y localizados los dados en cada frame, el siguiente objetivo es determinar automáticamente cuándo los dados se encuentran en reposo, es decir, cuándo dejan de moverse luego de la tirada. Esta detección es fundamental, ya que el valor de cada dado solo debe calcularse cuando su posición es estable y fija.

La estrategia implementada se basa en el análisis del desplazamiento de los dados entre frames consecutivos, utilizando como referencia la posición de sus centros geométricos.

Para cada frame procesado se obtiene el conjunto de centros correspondientes a los dados detectados. Con el fin de mantener consistencia entre frames, los dados se ordenan según su coordenada horizontal de izquierda a derecha, evitando así ambigüedades cuando se comparan posiciones entre imágenes sucesivas.

El criterio de reposo se evalúa comparando los centros del frame actual con los del frame anterior. Para cada centro actual se calcula la distancia mínima respecto de los centros del frame previo, y se acumula el desplazamiento total. Este enfoque resulta robusto ante pequeñas variaciones en el orden de detección de los contornos o leves fluctuaciones en la segmentación.

Si el desplazamiento total es inferior a un umbral, que por defecto son 10 píxeles, se considera que el movimiento entre ambos frames es despreciable y que los dados se encuentran potencialmente en reposo, esto ya que por ejemplo podría haber una pequeña diferencia de pixeles provocadas por el movimiento, o enfoque y desenfoque de la cámara. En caso contrario, se interpreta que aún existe movimiento significativo.

Con el objetivo de evitar falsos positivos , por ejemplo, detecciones erróneas debidas a ruido, vibraciones leves o errores momentáneos en la segmentación, no se declara el reposo a partir de un único frame estático. En su lugar, se implementa un criterio de estabilidad temporal, en este caso utilizando un contador de frames consecutivos sin movimiento apreciable.

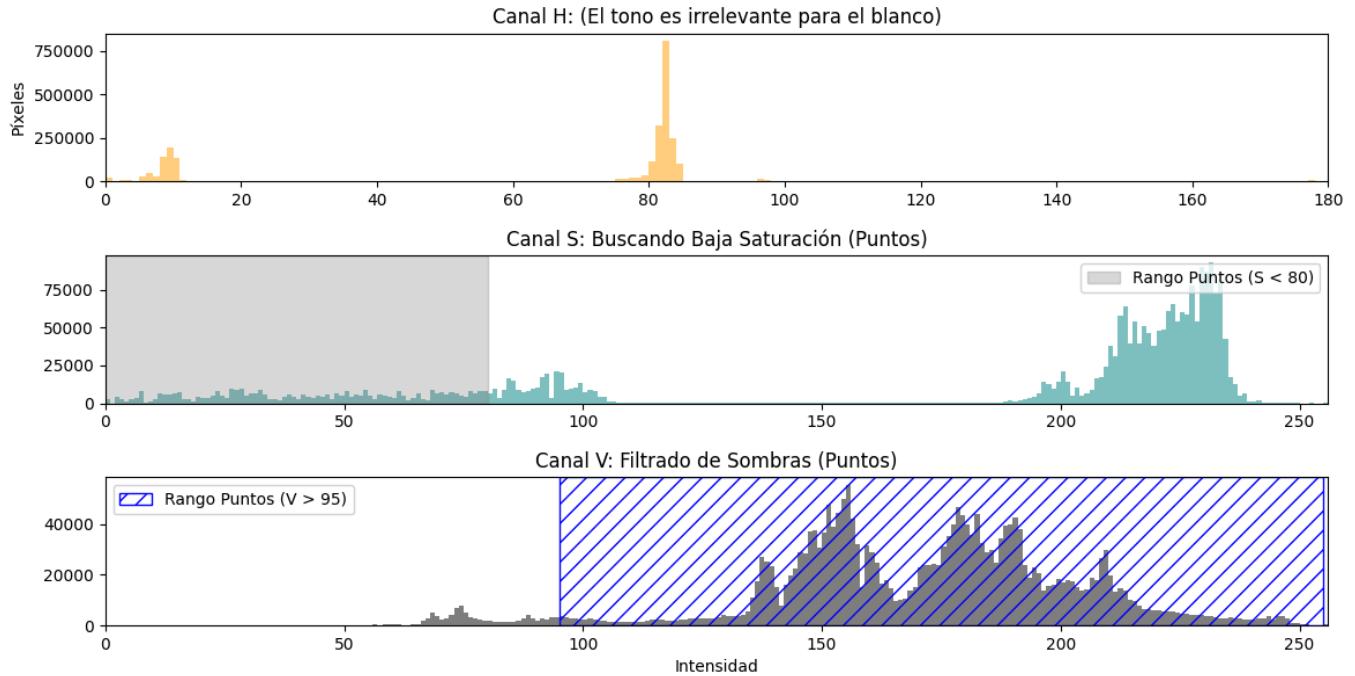
El procedimiento es el siguiente:

1. Si el frame cumple la condición de movimiento menor al umbral, se incrementa un contador de reposo.
2. Si el movimiento supera el umbral, el contador se reinicia y se habilita una nueva detección futura.
3. Se considera que los datos están efectivamente en reposo únicamente cuando el contador supera un número mínimo de cinco frames consecutivos predefinido.

Una vez validado el estado de reposo, se habilita la etapa de cálculo del valor de cada dato. Para evitar repetir la lectura del mismo evento de reposo en múltiples frames consecutivos, se utiliza una bandera de control que garantiza que la lectura y visualización del resultado se realicen una sola vez por tirada.

Etapa 4: Detección de valor

Para poder detectar el valor de la cara de los dados, recurrimos a la misma segmentación por color para poder aislar los puntos blancos. Al ser acromático, realizamos una segmentación por saturación y brillo.

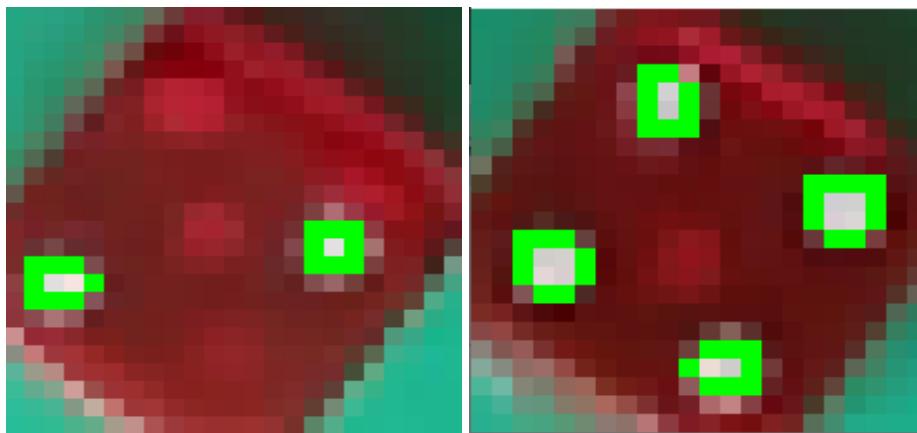


A diferencia de los tonos rojos, los valores de saturación que representan a los colores negro, blanco y grises están en el rango inferior. La elección del umbral ($S < 80$) nos permite segmentar esta gama de colores.

Dado que el color blanco, negro y las sombras comparten la propiedad de tener baja saturación, filtrar por saturación por sí solo es insuficiente. El umbral de brillo ($V > 95$) es lo que nos permite separar los puntos blancos de alta intensidad, del resto de las zonas oscuras o sombras, que poseen baja intensidad.

Utilizando los rangos provistos por los umbrales antes mencionados, generamos una nueva máscara que nos permitirá, a partir del análisis de contornos, contabilizar el valor final de la cara de cada dado. Para ello, filtramos los contornos por área para eliminar el ruido. Luego, regularizamos su geometría mediante Convex Hull y evaluamos su factor de forma. Si cumplen con el umbral establecido, son validados y contabilizados para obtener el valor del dado.

Detección de puntos preliminar



Etapa 5: Unificar implementaciones y graficar

Una vez obtenidos los valores de cada dado, graficamos las bounding boxes en color rojo para indicar el estado de reposo. Incluimos también el ID de cada dado (ordenado según sus centros de izquierda a derecha) y su valor alineado al centro de la caja.

Resultado Final		Salida por Terminal
	<pre>--- Video: tirada_1.mp4 --- --- Reposo alcanzado en Frame: 67 --- Valor del dado 1: 2 Valor del dado 2: 4 Valor del dado 3: 1 Valor del dado 4: 1 Valor del dado 5: 3 Dados detectados: 5 Suma Total: 11</pre>	

Conclusión

En este trabajo se desarrolló una solución para el análisis automático de tiradas de dados a partir de videos, aplicando de forma integrada los conceptos principales vistos en la materia de Procesamiento de Imágenes. El objetivo fue detectar los dados, determinar cuándo se encontraban en reposo y calcular el valor de la cara visible en cada tirada.

La segmentación por color en el espacio HSV permitió aislar de manera efectiva los dados del fondo, mientras que el uso de operaciones morfológicas mejoró la calidad de las máscaras obtenidas y facilitó la detección de contornos. A partir de estos contornos se extrajo la información geométrica necesaria para el seguimiento de los dados a lo largo del video.

La detección del estado de reposo se resolvió analizando el desplazamiento de los dados entre frames consecutivos, incorporando un criterio temporal que evitó lecturas erróneas ante pequeños movimientos o variaciones en la segmentación. Una vez confirmado el reposo, se realizó el conteo de puntos blancos sobre cada dado para determinar su valor.

En conjunto, el trabajo permitió integrar técnicas de segmentación, morfología y análisis temporal, logrando cumplir con los objetivos planteados en la consigna.