

# INTRODUCCIÓN

## Comprensión del problema

Este informe tiene como objetivo la comprensión del Trabajo Práctico N° 3 de la materia Procesamiento de Imágenes.

Este trabajo abarca detección y reconocimiento de objetos en movimiento. Estas prácticas tienen diversas aplicaciones, como por ejemplo en áreas de desarrollo de videojuegos.

El objetivo principal de este trabajo es desarrollar un algoritmo capaz de identificar de manera automática los momentos en que cinco dados dejan de moverse en un video y, posteriormente, reconocer el número que muestra cada dado. Para ello, se emplearán técnicas de procesamiento de imágenes como la segmentación por color, la detección de bordes, el seguimiento de objetos y el reconocimiento de patrones.

## Estructura de este informe

Este informe se estructura de la siguiente manera:

* Metodología empleada, incluyendo herramientas y técnicas utilizadas.
* Desarrollo: explicación del código y las funciones implementadas.
* Resultados obtenidos al aplicar el algoritmo a los videos de prueba.
* Conclusión final y posibles mejoras futuras.

# METODOLOGÍA

## Herramientas y tecnologías empleadas

Se empleó el editor de código fuente Visual Studio Code y el lenguaje de programación Python. Se utilizaron las siguientes bibliotecas:

* OpenCV (cv2): biblioteca que proporciona las herramientas necesarias para el procesamiento de imágenes y videos.
* Matplotlib: para la visualización de los resultados obtenidos, generando figuras.

## Información que tuvimos en cuenta acerca de los videos

Antes de comenzar a implementar el código, verificamos que ciertas condiciones se cumplen para todos los videos. Son las siguientes:

* Todas las secuencias de videos están grabadas desde el mismo ángulo.
* Se lanzan 5 (cinco) dados por secuencia de video.
* Los dados se detienen sobre un fondo verde.

Esto nos ayuda a generar ciertas hipótesis que nos facilitan la implementación del algoritmo, como por ejemplo que las métricas de los dados no varían por cada secuencia de video, y el fondo de los mismos se mantiene uniforme por cada uno.

## Métodos y técnicas utilizadas

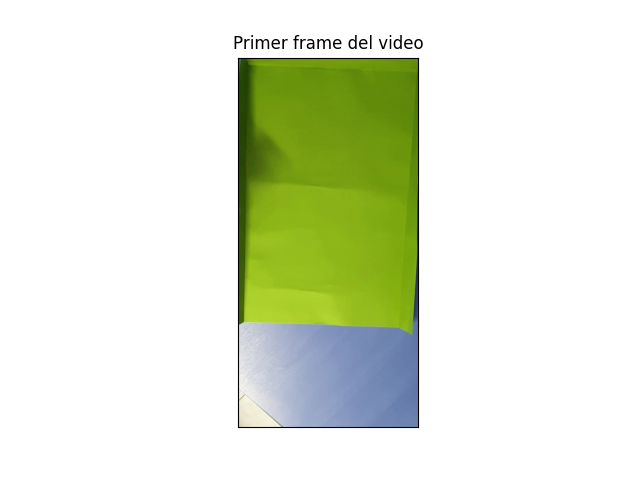
Para la implementación del algoritmo, se utilizó la librería OpenCV en el lenguaje de programación Python. El proceso se dividió en las siguientes etapas:

1. **Segmentación por color**:
   * Se aplicó una máscara en el espacio de color HSV para aislar el fondo verde del área de juego.
   * Posteriormente, se utilizó el espacio de color LAB para detectar los dados rojos mediante el canal A, que representa la diferencia entre los colores verde y rojo.
2. **Detección de bordes y contornos**:
   * Se detectaron los contornos de los objetos utilizando técnicas de umbral y detección jerárquica en imágenes binarias.
   * Los contornos se filtraron con base en el área, perímetro y una métrica de forma para identificar aquellos que corresponden a dados.
3. **Reconocimiento de patrones**:
   * Se empleó la detección de puntos blancos en cada dado, utilizando un rango específico en el espacio de color HSV para identificar las tonalidades blancas correspondientes a los puntos.
4. **Control de estabilidad**:
   * Se verificó que los dados estuvieran quietos durante un número fijo de frames consecutivos antes de proceder al conteo de puntos y la evaluación final.

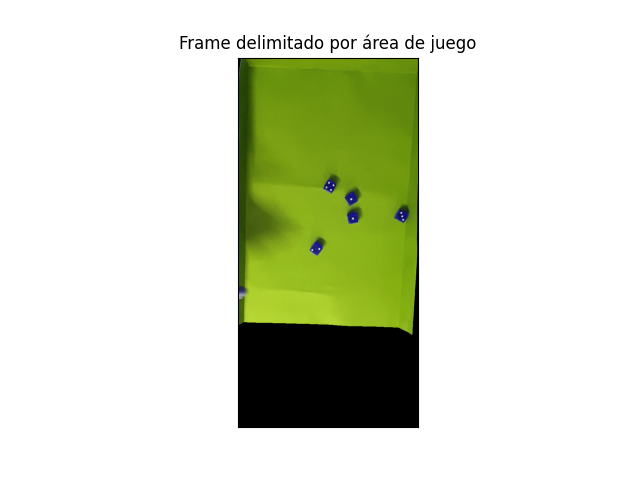
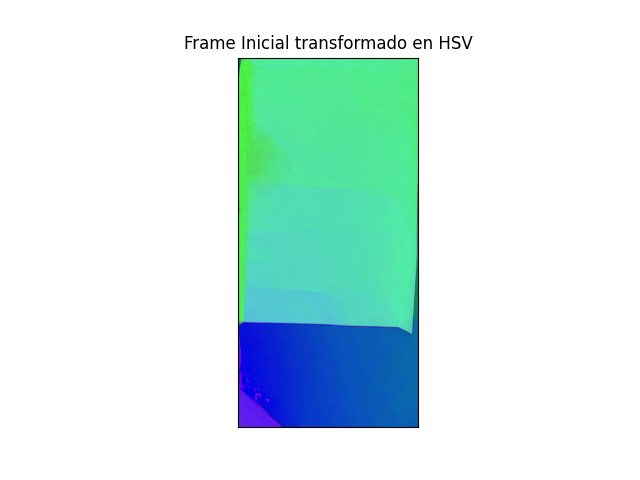
# DESARROLLO

El programa sigue los siguientes pasos:

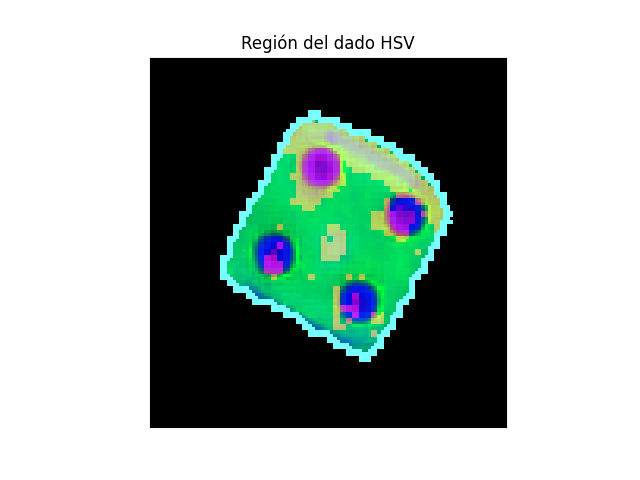
1. **Carga del video**:
   * El algoritmo comienza cargando el video de entrada y extrae el primer frame para analizar el área de juego.



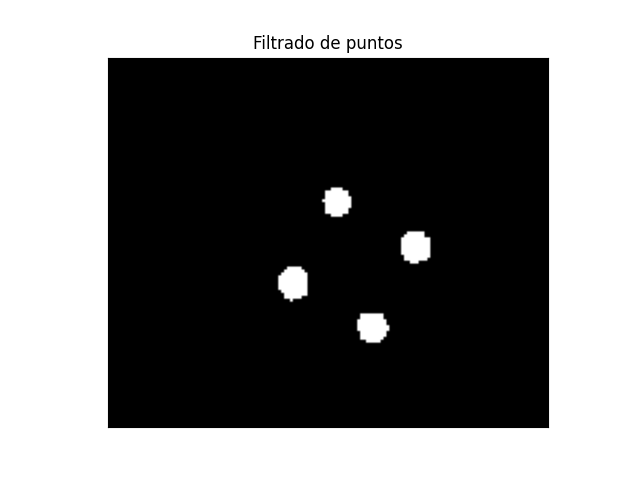
1. **Máscara del área de juego**:
   * Una máscara basada en el color verde permite aislar la región del área de juego donde caen los dados.



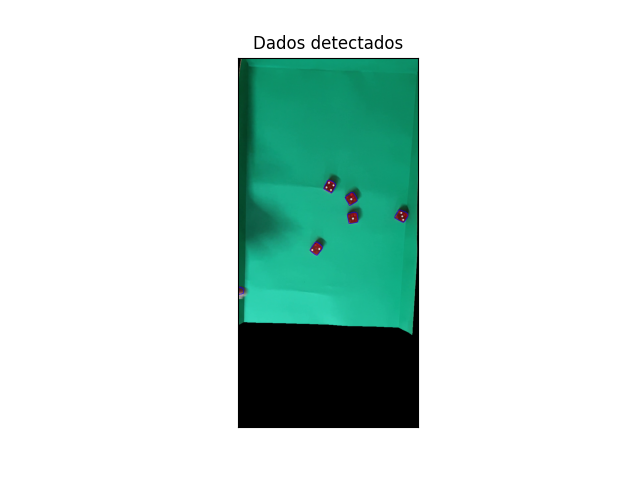
1. **Detección de dados rojos**:
   * Se procesó la imagen en el espacio LAB para detectar las áreas rojas que correspondían a los dados. Los contornos resultantes fueron filtrados para descartar formas que no cumplieran con las características esperadas de un dado.



1. **Conteo de puntos**:
   * Cada dado fue segmentado individualmente, y dentro de su región se identificaron los puntos blancos mediante una segunda máscara. El número de contornos detectados dentro de cada región se utilizó como representación del valor del dado.



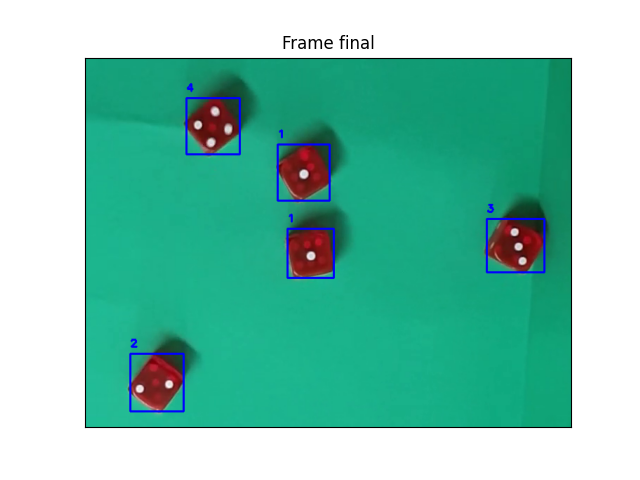
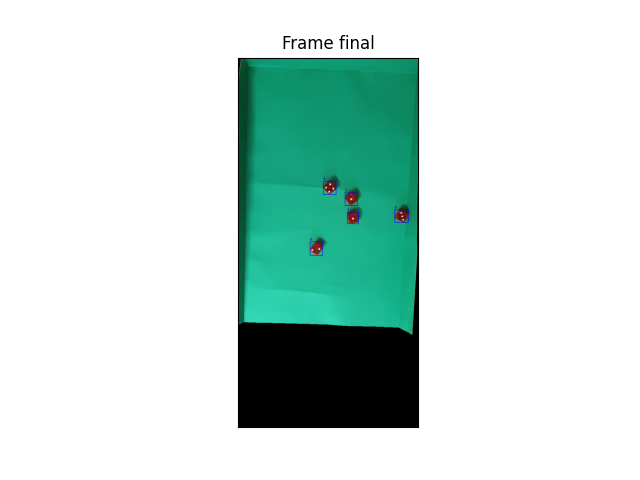
1. **Control de estabilidad**:
   * El algoritmo monitoreó el estado de movimiento de los dados y solo procedió al análisis cuando estos estuvieron inmóviles durante un número suficiente de frames consecutivos.



# RESULTADOS

El algoritmo fue aplicado a las secuencias de video proporcionadas y obtuvo los siguientes resultados:

1. **Detección de dados**:
   * Se detectaron consistentemente los cinco dados en cada secuencia de video cuando los mismos se encontraban completamente visibles sobre el fondo verde. Existen intermitencias dependiendo del movimiento de estos, requiriendo al menos 20 frames (0.66 segundos a 30fps) para determinar la finalización del movimiento.
2. **Conteo de puntos**:
   * El número de puntos en cada dado fue identificado correctamente en la todos los casos, con un margen de error mínimo en situaciones de movimiento residual por lo anteriormente mencionado.
3. **Estabilidad**:
   * El sistema logró identificar correctamente los momentos en que los dados permanecieron inmóviles, cumpliendo con el criterio de estabilidad definido.



# CONCLUSIONES

El algoritmo desarrollado ha demostrado ser eficaz en la tarea de detectar y reconocer los números de cinco dados en movimiento sobre un fondo verde uniforme. La combinación de técnicas de procesamiento de imágenes como segmentación por color, detección de contornos y análisis de patrones ha permitido obtener resultados precisos y confiables.

* Precisión en la detección: El algoritmo ha demostrado ser capaz de identificar de manera consistente los dados y sus respectivos números. La segmentación por color y la detección de contornos demostraron ser herramientas efectivas para identificar los dados y sus puntos respectivos, aun así, teniendo puntos que a simple vista se notan por la transparencia de los dados.
* El control de estabilidad permitió asegurar que el análisis solo se realizara en condiciones óptimas, reduciendo errores en el conteo y en la detección de estos.

Limitaciones y futuras mejoras:

* Sensibilidad a condiciones de iluminación: El algoritmo podría ser más sensible a variaciones significativas en la iluminación, lo que podría afectar la precisión de la segmentación por color.
* El sistema podría tener dificultades para detectar los dados cuando están parcialmente ocultos, dañados o cuando se superponen.
* Generalización a otros objetos: Aunque el algoritmo ha sido diseñado para detectar dados, podría ser necesario realizar ajustes para aplicarlo a otros objetos con características visuales diferentes.
* Optimización del código: La optimización del código podría reducir el tiempo de procesamiento y permitir el análisis de videos en tiempo real en dispositivos con recursos limitados.

En conclusión, el algoritmo desarrollado ha demostrado ser una herramienta valiosa para la detección y reconocimiento de dados en videos. Sin embargo, existen oportunidades para mejorar su desempeño en términos de robustez y generalización.