



## Tecnicatura Universitaria en Inteligencia Artificial

### Procesamiento de Imágenes

---

# Informe Trabajo Práctico 3

Integrantes:

- Herrera Morena - (H-1187/8)
- María Victoria Oviedo - (O-1820/1)
- Zorzolo Rubio Juana - (Z-1217/3)

## Índice

Resumen.....	3
Introducción.....	3
Metodología.....	3
1. Preprocesamiento de Imágenes y Videos.....	3
2. Detección de Área de Lanzamiento de los Dados.....	3
3. Detección de Movimiento.....	4
4. Captura de Frames con Dados Quietos.....	4
5. Procesamiento de los Dados.....	4
6. Agrupación de Contornos y Identificación de Dados.....	5
Desarrollo/Implementación.....	5
Funciones de Utilidad:.....	5
Creación de Carpeta:.....	5
Detección de Dados en Reposo:.....	5
Procesamiento de Dados y Detección de Números:.....	6
Generación de Videos con Bounding Boxes:.....	6
Desafíos Técnicos/Problemas a los que nos enfrentamos.....	6
Resultados.....	7
Pruebas y Ejemplos de Ejecución.....	7
Conclusiones.....	8
Anexos.....	9

## Resumen

En este trabajo se desarrolló un algoritmo para detectar automáticamente el momento en que cinco dados se detienen durante una tirada y leer el número obtenido en cada uno. Además, se generaron videos modificados que resaltan los dados en reposo con un bounding box azul y muestran el número reconocido sobre ellos. El desarrollo incluyó etapas de preprocesamiento, detección de reposo, reconocimiento de números y generación de resultados visuales.

## Introducción

Se abordó el problema específico de analizar secuencias de video de tiradas de dados, identificando el momento en que los mismos se detienen y extrayendo su resultado.

Las secuencias proporcionadas (tirada\_1.mp4, tirada\_2.mp4, tirada\_3.mp4 y tirada\_4.mp4) presentaron un desafío interesante: combinar detección de movimiento, segmentación de objetos y reconocimiento de patrones. Este informe describe la metodología utilizada, los resultados obtenidos y los problemas enfrentados durante el desarrollo.

## Metodología

Se desarrollaron una serie de pasos que incluyen preprocesamiento de imágenes, la detección de movimiento, y la clasificación de los dados. A continuación, se detallan las metodologías empleadas en cada etapa del proceso:

### **1. Preprocesamiento de Imágenes y Videos**

Se inició con la lectura de videos de una carpeta de entrada ("videos-dados"). Para cada video, se extrajeron los frames y se redimensionaron para un procesamiento más eficiente. Las dimensiones de los videos (ancho y alto) fueron obtenidas automáticamente utilizando la biblioteca OpenCV.

### **2. Detección de Área de Lanzamiento de los Dados**

Para identificar el área donde los dados caen, se utilizó el espacio de color HSV. Se definieron rangos de color (típicamente para tonos de verde) para identificar el área de interés. El proceso incluye:

- Conversión de la imagen de BGR a HSV.
- Aplicación de un umbral para detectar áreas dentro de los rangos de color definidos.

- Realización de operaciones morfológicas (cierre y apertura) para reducir el ruido y mejorar la detección.
- Localización del área mediante la detección de contornos.
- Se seleccionó el contorno más grande como el área de interés, y se dibujaron rectángulos alrededor de las regiones que contenían los dados. Esto permitió segmentar eficazmente el área donde los dados se encuentran dentro del frame.

### **3. Detección de Movimiento**

Una vez identificada el área de interés, se implementó un mecanismo de detección de movimiento para detectar si los dados están quietos o en movimiento. Para esto, se utilizó la diferencia entre dos frames consecutivos:

- Se extrajo la región de interés (ROI) del frame anterior y el actual.
- Se calculó la diferencia absoluta entre las dos imágenes, y se convirtió a escala de grises.
- Se aplicó un umbral binario para resaltar las áreas con movimiento.
- Se llevó a cabo un período de calibración durante los primeros 60 frames del video, donde se registraron los niveles de movimiento y se estableció un umbral calibrado para determinar si los dados estaban quietos o en movimiento

### **4. Captura de Frames con Datos Quietos**

Cuando el nivel de movimiento era bajo (indicando que los dados estaban quietos), se procedió a guardar una imagen del frame actual. Estas imágenes fueron almacenadas en una carpeta ("capturas\_datos") para su posterior procesamiento. Además, se redimensionaron y mostraron en una ventana separada.

### **5. Procesamiento de los Datos**

Para la identificación de los dados y la extracción de los números de los mismos, se utilizó el siguiente enfoque:

- Conversión a Escala de Grises: Se convirtió cada imagen a escala de grises para facilitar la detección de bordes.
- Desenfoque Gaussiano: Para reducir el ruido, se aplicó un desenfoque gaussiano sobre la imagen en escala de grises.

- **Detección de Bordes:** Se utilizó el detector de bordes de Canny para resaltar las áreas que podrían corresponder a los dados.
- **Operaciones Morfológicas:** Se aplicaron operaciones morfológicas (dilatación y erosión) para limpiar la imagen y mejorar la detección de los contornos de los dados.
- **Umbralización y Detección de Contornos:** Se utiliza la técnica de umbralización Otsu para binarizar la imagen, lo que permite separar los elementos de interés del fondo. Posteriormente, se detectan los contornos externos en la imagen binarizada.
- **Filtrado de Contornos:** Los contornos detectados se filtran para identificar únicamente aquellos que sean circulares y tengan áreas dentro de un rango específico.

## 6. Agrupación de Contornos y Identificación de Dados

Una vez detectados los contornos en la imagen, se calcularon los centroides de cada uno y se agruparon en función de su cercanía utilizando una medida de distancia euclidiana. Esto permitió agrupar los contornos que pertenecían al mismo dado.

Los dados fueron marcados en la imagen con un rectángulo delimitador (bounding box) y se asignaron diferentes colores a los grupos de contornos identificados. Esta visualización fue útil para asegurar que los dados fueran correctamente identificados y agrupados.

## Desarrollo/Implementación

El desarrollo se realizó en el lenguaje Python utilizando librerías como OpenCV y NumPy. A continuación, se describen las funciones principales:

### Funciones de Utilidad:

- `imshow()`: Muestra imágenes en ventanas con opciones de color, título y barra de color.
- `eliminar_carpetas_contenido()`: Elimina las carpetas de trabajo previas para reiniciar la ejecución.

### Creación de Carpeta:

- Se crea una carpeta `capturas_dados` para almacenar las imágenes procesadas y los videos resultantes.

### Detección de Dados en Reposo:

- Función `detectar_area()`: Detecta el área de los dados en cada cuadro del video utilizando un rango de colores en el espacio HSV.

- Función `detectar_movimiento()`: Detecta la quietud de los dados al comparar los frames anteriores con los actuales. Se utiliza un umbral para determinar si los dados están quietos.

#### **Procesamiento de Datos y Detección de Números:**

- Función `procesar_dado()`: Preprocesa la imagen de los dados (conversión a escala de grises, detección de bordes, operaciones morfológicas) y detecta contornos circulares que corresponden a los dados.
- Función `agrupar_contornos_por_cercania()`: Agrupa los contornos cercanos para identificar los dados individuales.
- Función `marcar_dados_y_nros()`: Dibuja las bounding boxes en los dados con sus respectivos números.

#### **Generación de Videos con Bounding Boxes:**

- Los videos resultantes se guardan con los dados detectados y numerados, agregando cuadros de delimitación azul y los números sobre los dados.

### **Desafíos Técnicos/Problemas a los que nos enfrentamos**

#### **- Detección Incorrecta del Reposo Inicial de los Dados**

Durante el desarrollo, utilizamos un umbral de movimiento menor a 1000 para determinar que los dados estaban quietos. Sin embargo, encontramos un problema al analizar los primeros frames de algunos videos: antes de que los dados fueran lanzados, el movimiento en la escena era muy bajo, lo que hacía que esos frames iniciales fueran erróneamente etiquetados como momentos en los que los dados estaban en reposo.

Solución Implementada: Se incorporó un periodo de calibración de 60 frames al inicio del video. Durante este periodo, aunque el nivel de movimiento fuese menor al umbral, los frames no se consideraban para la detección de reposo.

#### **- Detección de Dados a partir de sus Puntos**

Al intentar detectar los contornos de los dados, enfrentamos dificultades debido a la variabilidad en las formas y bordes visibles en las imágenes. En lugar de detectar directamente los dados, logramos identificar los contornos de los círculos que de los puntos de sus caras. Esto planteó el desafío de agrupar correctamente los puntos pertenecientes a cada dado.

Solución Implementada: Se desarrolló un algoritmo que agrupa los puntos detectados en función de su cercanía, asignándolos a un dado específico. A partir del centroide de estos grupos de puntos, generamos las bounding boxes correspondientes a cada dado. Cada bounding box se definió utilizando un margen preestablecido alrededor del centroide.

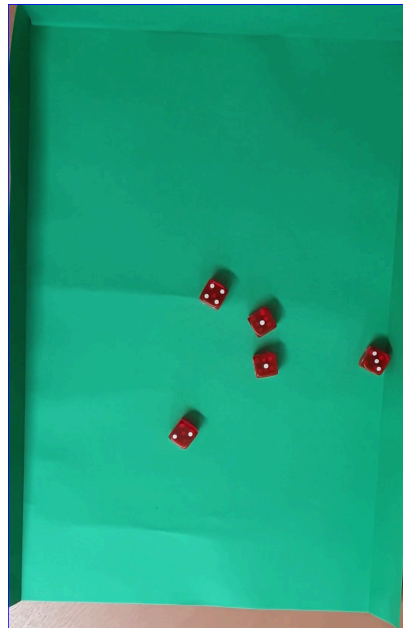
## Resultados

La resolución permitió detectar los dados en reposo y reconocer sus números. Los videos generados destacan visualmente los dados detectados, mostrando sus números y los bounding boxes en color azul.

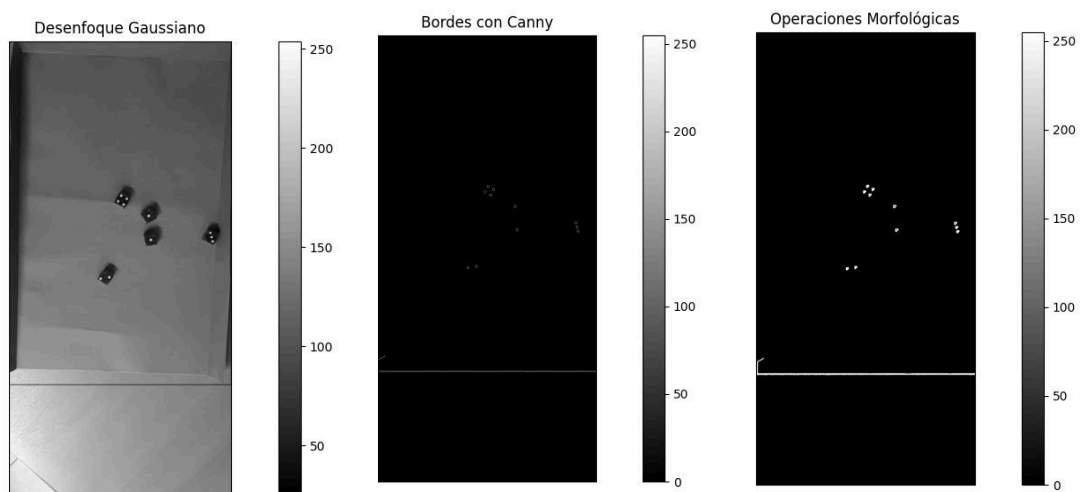
## Pruebas y Ejemplos de Ejecución

Para [tirada\\_1.mp4](#):

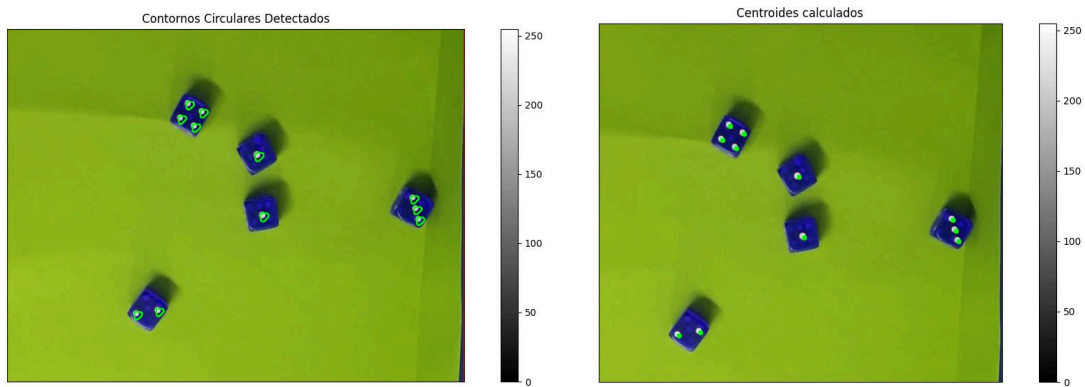
- Detección de dados en reposo:



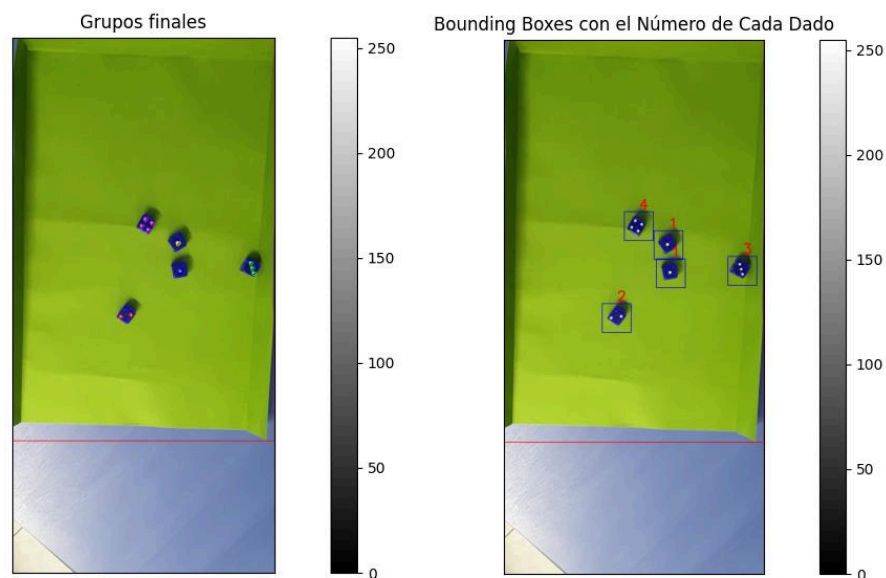
- Preprocesamiento:



- Reconocimiento número de dados:



- Resultado final para generación de video final:



## Conclusiones

El desarrollo de este trabajo logró un avance significativo en la detección y reconocimiento automatizado de dados y sus números en diferentes videos (tirada\_1.mp4, tirada\_2.mp4, tirada\_3.mp4 y tirada\_4.mp4).

El algoritmo implementado integra diversas etapas, como preprocesamiento, detección de movimiento, segmentación y reconocimiento de dados en reposo, resaltando sus valores numéricos en los resultados visuales generados.

En términos generales:

- **Implementación:** Se diseñó una resolución que combina técnicas para procesar los videos, identificar momentos clave y extraer información relevante.



- **Resultados:** Los videos generados validan la efectividad del método al mostrar claramente los datos detectados y sus números con alta precisión.
- **Mejoras:** Futuras iteraciones podrían optimizar la calibración y la segmentación para abordar variaciones en iluminación, movimiento inicial y formas de los datos.

En resumen, aunque se enfrentaron desafíos técnicos, las soluciones implementadas validan la solidez de los resultados obtenidos. Este trabajo puede ajustarse en un futuro para mejoras y aplicaciones más avanzadas en contextos similares.

## Anexos

Repositorio de github:

<https://github.com/juanazorzolo/TP3-PDI-TUIA-Herrera-Oviedo-Zorzolo>