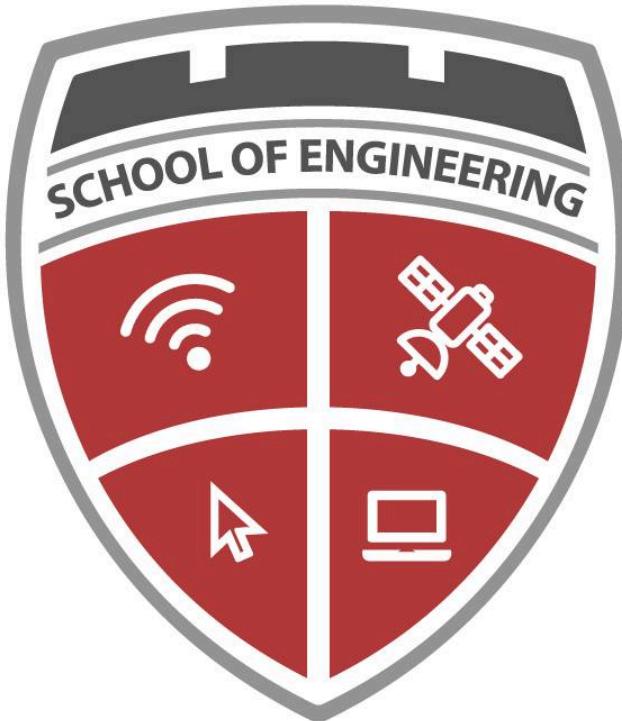


UNIVERSIDAD AUTÓNOMA GABRIEL RENÉ MORENO
SCHOOL OF ENGINEERING SOE - UNIDAD DE POSGRADO
MAESTRÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL V1E2



**CONTEO DE SERES VIVOS MEDIANTE VISIÓN ARTIFICIAL CLÁSICA CON
OpenCV
LABORATORIO 1 - GRUPO 3**

Participantes: ♦ Karen Torrico
♦ Karen Huacota
♦ Yesika Luna
♦ Elvis Miranda
♦ Ivan Mamani Condori

Docente: Msc. Danny Luis Huanca Sevilla

Módulo: GENERATIVE MODELS AND COMPUTER VISION

Enero, 2026

ÍNDICE

INTRODUCCIÓN.....	3
1. ANTECEDENTES.....	4
2. OBJETIVOS.....	4
3. DATOS DE ENTRADA.....	5
4. IMPLEMENTACIÓN.....	5
Procesamiento inicial y conversión a escala de grises.....	5
Detección de movimiento y umbralización.....	7
Detección y filtrado de contornos.....	8
Definición de la línea virtual y lógica de conteo.....	9
Visualización de resultados.....	9
5. RESULTADOS Y ANÁLISIS DE ERRORES.....	10
Resultados obtenidos.....	10
Análisis de errores.....	11
7. REFLEXIÓN FINAL.....	12
8. ANEXOS.....	12

INTRODUCCIÓN

El análisis de video constituye una herramienta esencial en múltiples áreas de investigación y aplicación profesional, incluyendo la vigilancia, la automatización industrial y el monitoreo de fauna. La generación continua de grandes volúmenes de datos visuales exige métodos eficientes que permitan extraer información relevante de manera automatizada. No obstante, en muchos escenarios prácticos no se cuenta con modelos entrenados ni con infraestructura de cómputo avanzada para implementar algoritmos de aprendizaje profundo. Por esta razón, resulta necesario recurrir a técnicas clásicas de visión por computadora, las cuales ofrecen soluciones robustas, computacionalmente eficientes y de bajo costo, capaces de abordar problemas concretos de análisis de movimiento y conteo.

Las técnicas clásicas de visión artificial permiten identificar movimiento y localizar objetos mediante operaciones matemáticas y geométricas elementales. Entre estas destacan la diferencia entre frames consecutivos, la umbralización binaria para segmentación de movimiento, la eliminación de ruido mediante operadores morfológicos y la detección de contornos para la identificación de los objetos de interés. Al combinar estas herramientas con razonamiento geométrico simple, es posible diseñar sistemas de conteo que determinen de manera precisa el cruce de un límite virtual dentro de la escena, generando información cuantitativa confiable sin la necesidad de modelos de aprendizaje complejo.

En el marco de esta práctica, los integrantes del grupo desarrollaron un sistema de conteo de seres vivos u objetos en movimiento utilizando exclusivamente métodos de visión clásica. Se trabajó con videos grabados con cámara fija, bajo condiciones controladas que permitieran evaluar la eficacia del pipeline implementado. El objetivo central fue demostrar que, aun en ausencia de modelos de Deep Learning, es factible construir un sistema que detecte y contabilice cruces de un límite virtual, proporcionando evidencia visual interpretativa y permitiendo reflexionar sobre las limitaciones y alcances de la visión artificial clásica en entornos reales y desafiantes.

1. ANTECEDENTES

Las cámaras de seguridad generan grandes volúmenes de video continuo; sin embargo, en muchos contextos reales no se dispone de modelos entrenados ni infraestructura avanzada, por lo que se requiere resolver problemas de análisis visual utilizando únicamente técnicas clásicas de visión por computadora.

En entornos como fábricas para contar productos, en establos para contar vacas, en salas de espera para contar personas y muchas otras aplicaciones, esta solución puede ayudar a controlar los objetos de análisis citados.

En esta práctica, cada grupo deberá diseñar e implementar un sistema simple de conteo, capaz de detectar y contabilizar objetos o personas que atravesen un límite virtual (línea) dentro de un video grabado con un celular en condiciones controladas.

El sistema deberá estar basado exclusivamente en:

- ❖ diferencia entre frames,
- ❖ umbralización,
- ❖ eliminación de ruido,
- ❖ detección de contornos,
- ❖ y razonamiento geométrico simple.

2. OBJETIVOS

Implementar un pipeline de visión clásica que:

- ❖ Detecte movimiento en un video de cámara fija.
- ❖ Identifique el objeto en movimiento dominante.
- ❖ Calcule su posición relativa respecto a una línea virtual.
- ❖ Cuente correctamente los eventos de cruce.

3. DATOS DE ENTRADA

Para la presente práctica, se emplearon dos conjuntos de video distintos, grabados bajo condiciones controladas con cámara fija. El primer video corresponde a una carrera de atletas, en la cual los participantes cruzan una línea de meta claramente definida. Este video tiene una duración de aproximadamente 24 segundos y se utilizó para probar la detección de objetos humanos en movimiento y la capacidad del sistema para contabilizar correctamente los cruces de una línea vertical central. El formato del archivo es MP4, lo que facilita su manipulación y procesamiento en Python mediante OpenCV.

El segundo video documenta un escenario natural, donde un grupo de vacas cruza un río. Este material visual, con duración aproximada de 32 segundos, permitió evaluar la eficacia del pipeline de visión clásica en un entorno más complejo, con elementos de ruido generados por el movimiento del agua y la vegetación circundante. La inclusión de ambos videos permitió validar la flexibilidad del sistema frente a distintos tipos de objetos en movimiento y condiciones ambientales variables.

4. IMPLEMENTACIÓN

La implementación del sistema de conteo se basó exclusivamente en técnicas clásicas de visión por computadora, cumpliendo con los requerimientos mínimos de la práctica. El pipeline fue diseñado para procesar dos tipos de videos: un caso controlado de **atletas cruzando la línea de meta** y un escenario natural de **vacas cruzando un río**, cada uno con características y desafíos particulares.

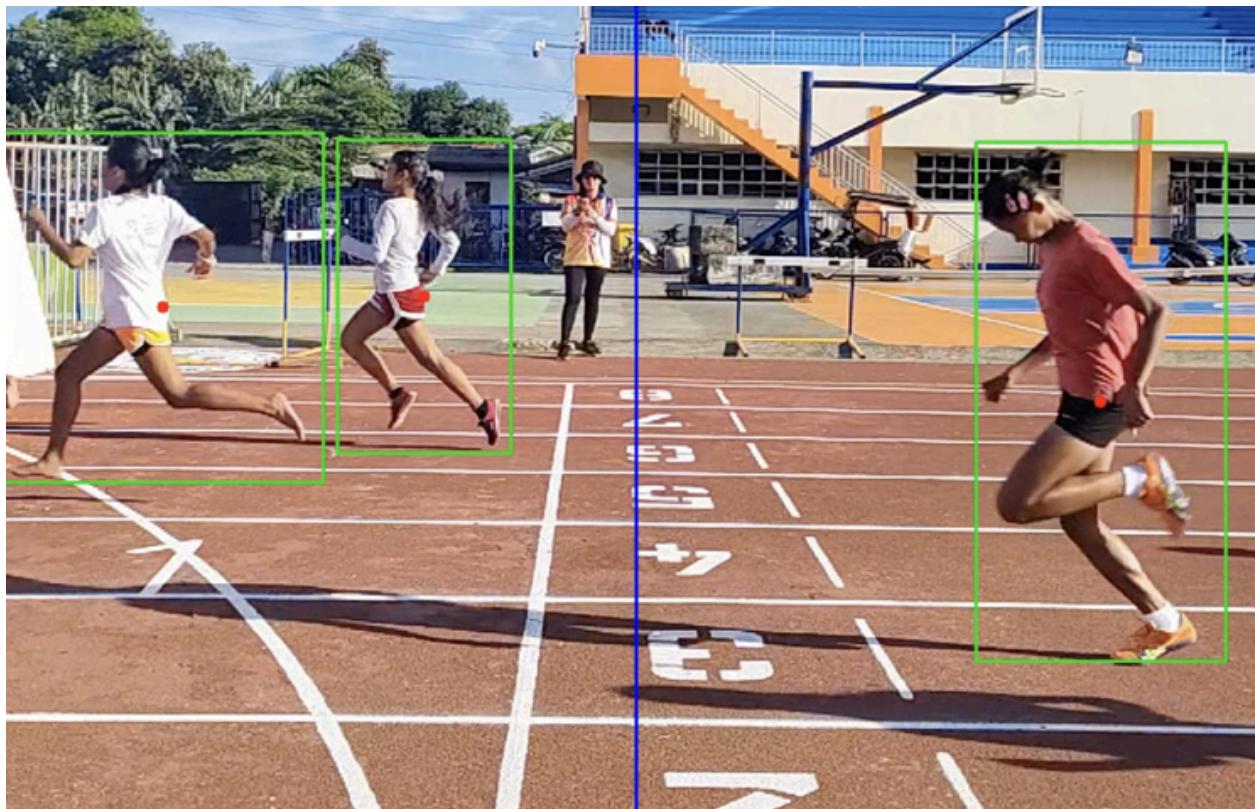
Procesamiento inicial y conversión a escala de grises

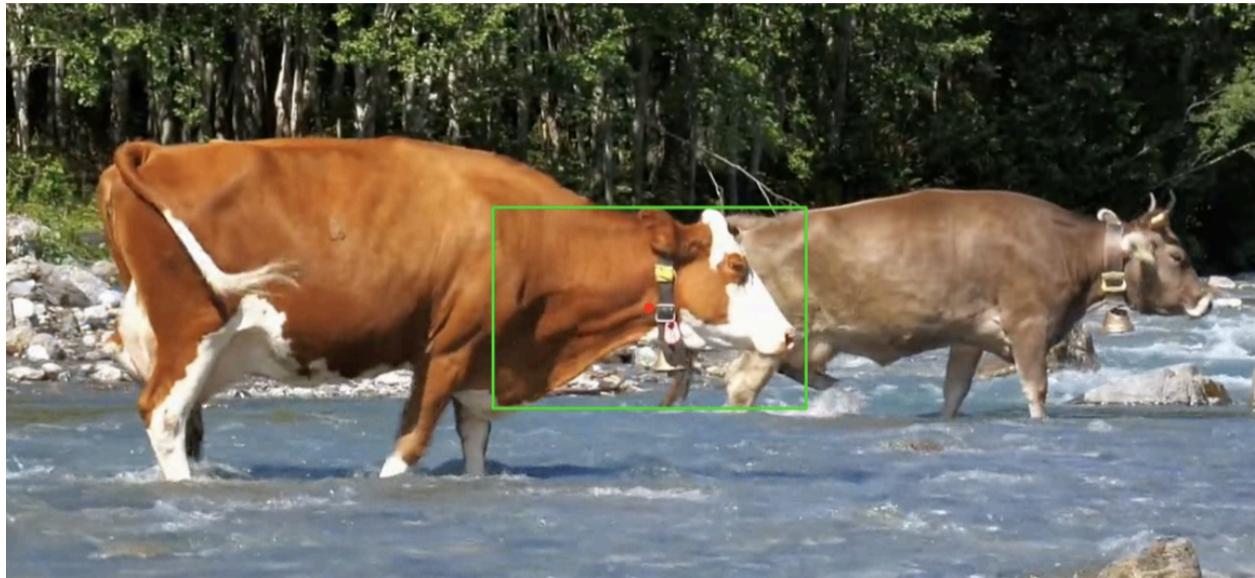
En ambos casos, cada frame del video se convirtió a **escala de grises**, simplificando el análisis y reduciendo el costo computacional al eliminar la información de color. Para minimizar el ruido y las pequeñas variaciones de iluminación entre frames consecutivos, se aplicó un **filtro Gaussiano (5x5)**. Este paso aseguró que los cambios detectados en la escena correspondieran principalmente al movimiento de los objetos de interés, como se evidencia en la **imagen** para los atletas.



Detección de movimiento y umbralización

- ❖ **Caso atletas:** Se utilizó la **diferencia absoluta entre frames consecutivos**, adecuada para un entorno controlado con mínima interferencia de ruido.
- ❖ **Caso vacas:** Se implementó un **sustractor de fondo MOG2**, que permitió ignorar el ruido generado por el movimiento del agua y la vegetación circundante, destacando únicamente los objetos relevantes.





En ambos casos, se aplicó un **umbral binario global**, seguido de **dilatación y operaciones morfológicas**, para eliminar pequeñas partículas irrelevantes y unir regiones fragmentadas.

Detección y filtrado de contornos

Los contornos de las regiones en movimiento se detectaron utilizando `cv2.findContours`. Para cada contorno, se aplicaron filtros específicos:

- ❖ **Área mínima:** 3000 píxeles para atletas y 7000 píxeles para vacas, adaptándose al tamaño relativo de los objetos.
- ❖ **Proporción de aspecto:**
 - **Atletas:** rectángulos verticales (altura mayor que el ancho).
 - **Vacas:** rectángulos horizontales (ancho mayor que la altura).
- ❖ **Posición en la imagen:** para descartar detecciones en bordes superiores o inferiores irrelevantes.

Cada contorno válido generó un **centroide**, utilizado posteriormente para el conteo. Las **imágenes 2 y 3** muestran ejemplos de detección de objetos y centroides para atletas y vacas, respectivamente, incluyendo los **bounding boxes** y los **puntos centrales** que se rastrean a lo largo de los frames.

Definición de la línea virtual y lógica de conteo

La línea virtual se definió de manera que correspondiera al flujo natural de los objetos:

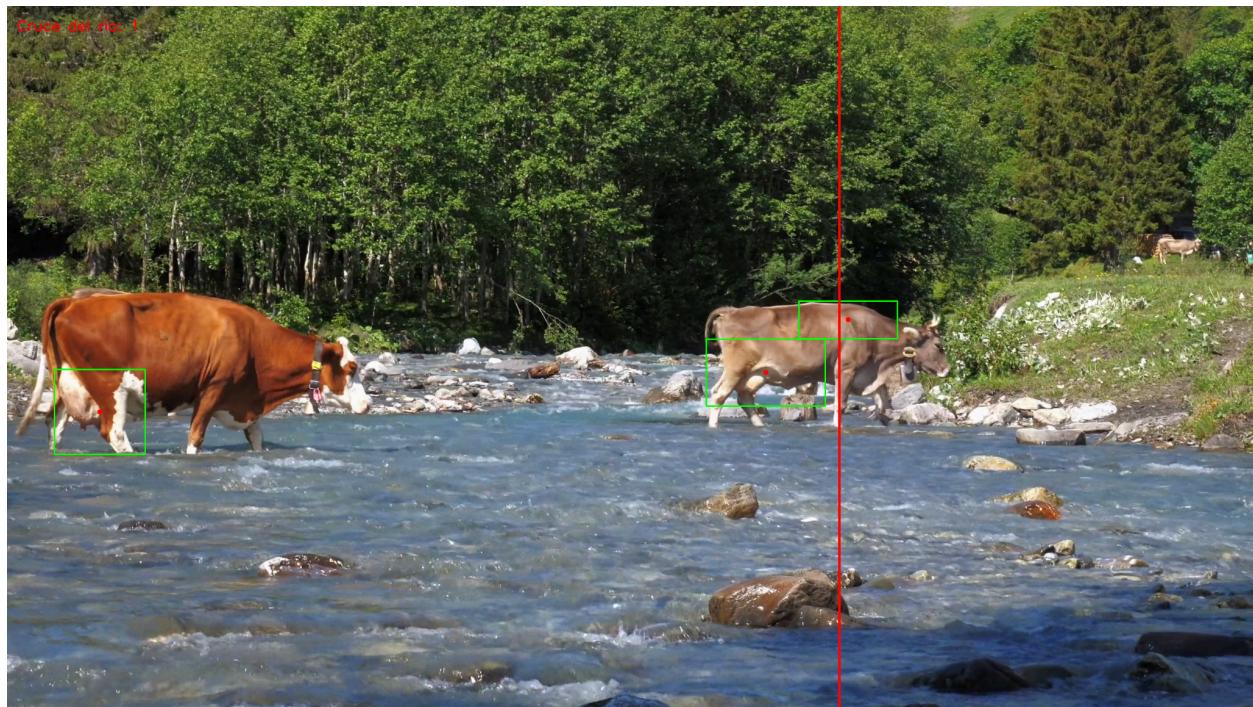
- ❖ **Atletas:** línea vertical al centro del video.
- ❖ **Vacas:** línea vertical a un tercio desde el lado derecho, considerando que se mueven de izquierda a derecha.

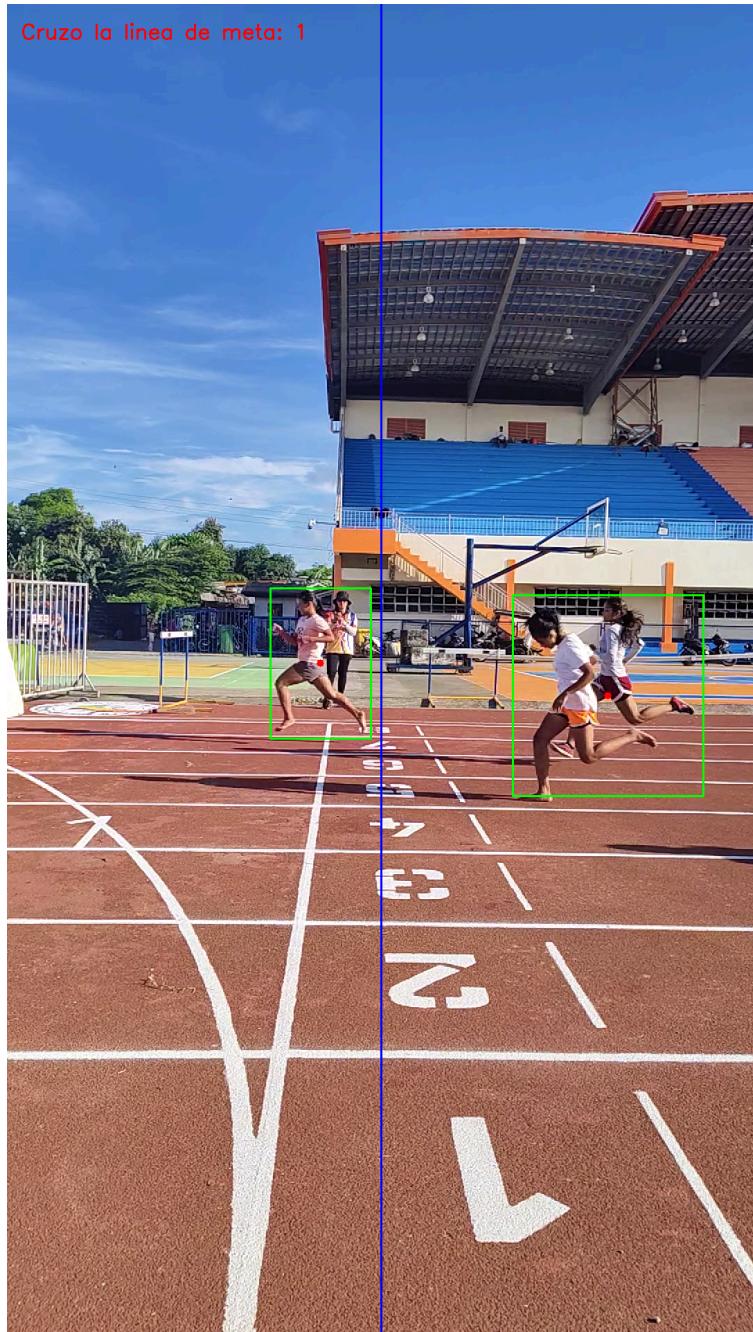
El conteo se realiza comparando la posición de cada centroide con respecto a la línea virtual. Cada objeto se contabiliza únicamente una vez al cruzar la línea, evitando duplicaciones mediante un seguimiento simple basado en la distancia máxima permitida entre centroides de frames consecutivos (`max_match_distance`).

Visualización de resultados

El sistema genera un **video de salida con overlays**, mostrando:

- ❖ Línea virtual en azul (atletas) o roja (vacas).
- ❖ Bounding boxes verdes alrededor de los objetos detectados.
- ❖ Centroides en rojo.
- ❖ Contador de cruces actualizado en tiempo real.





Estas visualizaciones permiten verificar de manera directa la correcta detección y conteo de objetos, tal como se observa en las **imágenes de arriba**, y constituyen evidencia del correcto funcionamiento del pipeline en ambos escenarios.

5. RESULTADOS Y ANÁLISIS DE ERRORES

Resultados obtenidos

El sistema implementado permitió contar de manera automática los cruces de objetos en ambos videos, validando la efectividad del pipeline basado en visión clásica.

- ❖ **Caso atletas:** El conteo fue preciso, detectando correctamente a cada participante al cruzar la línea de meta. La diferencia de frames consecutivos y la filtración por contorno vertical proporcionaron una segmentación robusta, sin detecciones falsas significativas, como se aprecia en la **imagen 1**.
- ❖ **Caso vacas:** A pesar de las condiciones más complejas, con agua en movimiento y vegetación circundante, el sustractor de fondo MOG2 combinado con filtrado por área y proporción horizontal permitió detectar y contar la mayoría de los cruces de vacas de manera confiable. La línea virtual ubicada a un tercio desde el lado derecho facilitó el seguimiento del movimiento de izquierda a derecha, como se muestra en la **imagen 2**.

Análisis de errores

Si bien el sistema logró resultados satisfactorios, se identificaron limitaciones inherentes a la visión clásica:

1. **Ruido de fondo:** En el video del río, pequeñas salpicaduras o reflejos del agua fueron interpretadas ocasionalmente como objetos en movimiento, generando falsas detecciones.
2. **Objetos múltiples y cercanos:** Cuando varios objetos se superponen o se mueven muy próximos entre sí, el sistema puede fusionar sus contornos, afectando la precisión del conteo.
3. **Variaciones de iluminación:** Cambios bruscos de luz, sombras o reflejos pueden alterar la segmentación por umbral, produciendo tanto falsas detecciones como pérdidas de objeto.
4. **Forma y orientación de objetos:** El filtro de proporción de aspecto puede descartar parcialmente objetos parcialmente visibles o con posturas no estándar, lo que podría ocasionar omisiones en el conteo.

7. REFLEXIÓN FINAL

El estudio evidencia que un pipeline de visión clásica, aunque simple, puede proporcionar resultados precisos en escenarios controlados y relativamente predecibles. Su principal ventaja es la **rapidez de ejecución y bajo requerimiento computacional**, haciéndolo útil en aplicaciones donde no es viable entrenar modelos de Deep Learning. Sin embargo, en escenarios dinámicos, con iluminación cambiante, objetos superpuestos o fondos ruidosos, los métodos clásicos presentan limitaciones que dificultan garantizar un conteo exacto.

La práctica permite concluir que la visión clásica sigue siendo una herramienta valiosa para problemas de conteo y seguimiento simples, pero su efectividad disminuye en condiciones del mundo real más complejas. Esto resalta la necesidad de combinar estos métodos con técnicas más avanzadas cuando se requiera mayor robustez frente a ruido, iluminación variable y alta densidad de objetos.

8. ANEXOS

Se adjunta Notebook en Python con el script aplicado al laboratorio en el siguiente repositorio:

<https://github.com/mc-ivan/computer-vision/tree/main/lab1>