Sistema de Estimación de Velocidad de Pelotas mediante Visión Computacional

Kristhian Santiago Palomino Fajardo
Data Scientist

15 de febrero de 2025

Resumen

Este documento describe la arquitectura, diseño e implementación de un sistema para estimación de velocidad de objetos en secuencias de video. El sistema combina técnicas de visión por computadora con procesamiento geométrico avanzado, ofreciendo una solución precisa y escalable para análisis deportivo y aplicaciones industriales.

Índice

1. Introducción

1.1. Contexto

En el ámbito del análisis de movimiento en deportes y procesos industriales, la medición precisa de velocidades de objetos móviles constituye un desafío técnico fundamental. Este sistema aborda dicho problema mediante técnicas innovadoras de visión computacional.

1.2. Objetivos Técnicos

- Diseñar un pipeline de procesamiento de video modular.
- Proveer una API RESTful para integración con sistemas externos.
- Permitir la calibración de la perspectiva en caso de que la cámara no esté alineada correctamente.

2. Arquitectura del Sistema

2.1. Diagrama de Componentes

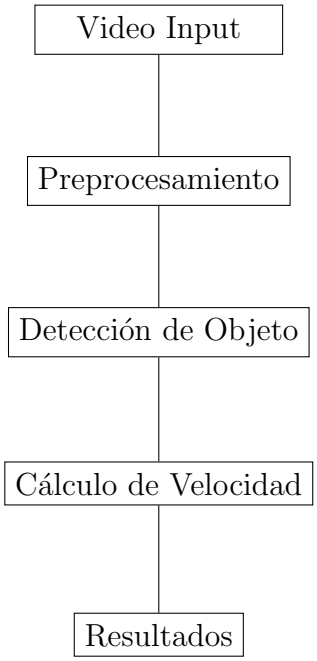

Figura 1: Flujo de procesamiento del sistema

2.2. Tecnologías Clave

Tecnología	**Versión**	**Uso**
OpenCV	4.7.0	Procesamiento de imágenes, detección de objetos y transformaciones geométricas
Streamlit	1.23.1	Interfaz web interactiva para visualización de resultados
FastAPI	0.95.2	API REST para integración con sistemas externos
NumPy	1.24.3	Cálculos matriciales y operaciones tensoriales

3. Diseño de Algoritmos

3.1. Detección de Objetos

Utilizamos un enfoque híbrido que combina:

- Filtrado espacial en espacio HSV.
- Segmentación morfológica con operaciones de apertura y cierre.

La máscara de color se define como:

$$M(x,y) = \begin{cases} 1 & \text{si } H(x,y) \in [H_{min}, H_{max}] \\ 0 & \text{de otro modo} \end{cases} \tag{1}$$

3.2. Corrección de Perspectiva

Si la cámara no está alineada correctamente con el plano de referencia, se aplica una transformación de homografía. Se usa una matriz de transformación basada en cuatro puntos de referencia tomados manualmente.

3.3. Cálculo de Velocidad

La velocidad media se calcula como:

$$v = \frac{d_{real}}{t_{transcurrido}} \tag{2}$$

donde d_{real} es la distancia real definida por el usuario y $t_{transcurrido}$ es el tiempo en segundos entre dos detecciones consecutivas.

4. Implementación

4.1. Estructura del Código

```
src/
 backend/              # API FastAPI
    main.py          # Endpoint para recibir y procesar video
    ball_tracker.py  # Detección de pelota y cálculo de velocidad
 frontend/             # UI Streamlit
    streamlit_app.py  # Aplicación principal
    pages/
       home.py      # Página de inicio
       velocity_estimation.py  # Página de análisis
 src/                  # Lógica del procesamiento
    constants.py     # Parámetros generales
    calculate_velocity.py  # Cálculo de velocidad
    homography.py    # Ajuste de perspectiva (en caso de ser necesario)
```

4.2. Pseudocódigo Principal

Algorithm 1 Procesamiento de Video

```
Cargar video ← input_path
(Opcional) Aplicar homografía si es necesario
Inicializar tracker ← CSRT()
while frame ∈ video do
    Aplicar filtro HSV
    Segmentar región de interés
    Calcular centroide
    Actualizar posición tracker
    Calcular velocidad media
end while
```