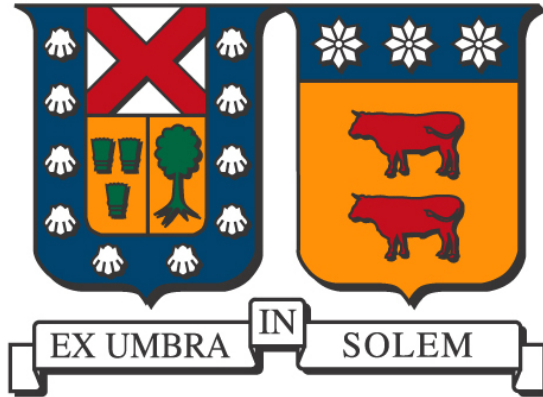


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTRÓNICA
VALPARAÍSO - CHILE



**“Modelo de Evaluación de Impacto Cefálico en Niños con TEA
mediante Visión por Computador”**

SERGIO ENRIQUE ROJAS ESCANILLA

**MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE INGENIERO CIVIL
TELEMÁTICO**

PROFESOR GUÍA:

Nicolás Torres

PROFESOR CORREFERENTE:

Patricio Olivares

SEPTIEMBRE - 2025



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título; ☐ Tesis de Postgrado;

Título del trabajo: Modelo de Evaluación de Impacto Cefálico en Niños con TEA
mediante Visión por Computado

Nombre del candidato(a): Sergio Rojas Escanilla

Carrera / Grado: Ingeniería civil Telemática

Campus: Santiago San Joaquín ; **Departamento:** Electrónica

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Nicolás Torres Rudloff, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL

☒ El trabajo **NO contiene información que amerite confidencialidad** y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (embargo) por:

☐ 6 meses; ☐ 12 meses; ☐ 2 años; ☐ 3 años; ☐ 5 años; ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 06/10/2025

; Firma:

Estudiante o Candidato(a):

Fecha: 06/10/2025

; Firma:

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

Agradecimientos

Quiero expresar mi más sincero agradecimiento a todos los docentes que fueron parte de mi formación académica, en especial a los profesores del Departamento de Telemática. Sin su guía, no sería el profesional en el que me convierto hoy. Dondequiera que la vida me lleve, llevaré siempre algo de lo que me enseñaron.

También agradezco profundamente a mi familia y a mi pareja. Sin su apoyo incondicional no habría logrado esta etapa: estuvieron cuando necesité una palabra de ánimo y me acompañaron en mis momentos de estrés, entre muchas otras ocasiones.

Asimismo, no quiero dejar de agradecer a mis amigos de la universidad, que se convirtieron en verdaderos hermanos. Ellos fueron el soporte constante para estudiar, hacer trabajos y resolver dudas. Como suele decirse, el paso por la universidad se hace mucho más fácil cuando se tienen buenos amigos en el camino.

Este documento está dedicado a mi papá y mamá que siempre me dieron su apoyo incondicional, para poder terminar mis estudios, ya que ellos no pudieron entonces estoy haciendo un sueño realidad de una familia.

Resumen

La creciente necesidad de **medición objetiva** y **seguimiento continuo** de conductas de riesgo en el TEA exige herramientas tecnológicas que operen más allá de la observación subjetiva. Este proyecto presenta una **solución integral de análisis de video** para *detección y cuantificación de golpes* en contextos de autolesión, orientada a entornos clínicos y domésticos. La metodología combina un detector 2D (**YOLOv8s**) para localizar mano, cabeza y golpe, con mapas de **profundidad monocular relativa (Depth Anything V2)** que permiten estimar, por evento, la **velocidad mano–cabeza** y derivar una **intensidad** interpretable (leve, media, fuerte). El sistema incorpora fusión detección–profundidad con normalización de escena y filtrado espacio–temporal, lo que estabiliza la señal y reduce errores por oclusiones, cambios de iluminación o ruido puntual.

Implementado como un **pipeline modular**, el enfoque facilita la ampliación a nuevos contextos y cámaras, y entrega **productos de salida** legibles para profesionales: video anotado, recortes por evento y un archivo **resultados.txt** que resume cada episodio con su velocidad, intensidad y cuadro de referencia para verificación inmediata. La arquitectura está pensada para **escalabilidad** y **integración** en flujos clínicos, priorizando la reproducibilidad del proceso y el resguardo de la privacidad. Entre los aportes destacan la **trazabilidad** desde una métrica física simple hacia una etiqueta clínica, y la posibilidad de **monitoreo longitudinal** de frecuencia e intensidad de eventos como insumo para decisiones terapéuticas. El proyecto sienta las bases para sistemas de apoyo a la intervención basados en evidencia, con potencial de incorporación en escenarios reales de atención y cuidado.

Palabras clave: TEA; YOLOv8s; Depth Anything V2; velocidad relativa; análisis de video;.

Abstract

There is a growing need for **objective measurement** and **continuous monitoring** of risk behaviors in Autism Spectrum Disorder, beyond subjective observation. This project presents an **end-to-end video analysis** solution for the *detection and quantification of head-impact events* in self-injurious contexts, targeting clinical and home environments. The method combines a 2D detector (**YOLOv8s**) that localizes **hand**, **head**, and **impact** with **monocular relative-depth** maps (**Depth Anything V2**) to estimate **hand-head relative velocity** per event and assign an interpretable **intensity** label (mild, moderate, strong). The system integrates detection–depth fusion with scene normalization and spatio–temporal filtering, stabilizing measurements and reducing errors due to occlusions, lighting changes, or local noise.

Implemented as a **modular pipeline**, the approach eases extension to new settings and cameras, and produces **clinician-readable outputs**: an annotated video, per-event crops, and a `resultados.txt` file that summarizes each episode with its velocity, intensity, and frame index for immediate verification. The architecture emphasizes **scalability**, **integration** into clinical workflows, **reproducibility**, and privacy safeguards. Key contributions include **traceability** from a simple physical metric to a clinically interpretable label, and the potential for **longitudinal monitoring** of event frequency and intensity to inform therapeutic decisions. The project lays the groundwork for evidence-based support systems suited for real-world care environments.

Keywords: TEA; YOLOv8s; Depth Anything V2; Relative velocity; Video analysis.

Glosario

Curva Precisión–Recall Gráfica que muestra el compromiso entre precisión y recall para distintos umbrales.

Depth Anything V2 Modelo de estimación de *profundidad monocular relativa* que produce mapas de profundidad por cuadro a partir de una sola imagen.

F1 Media armónica entre precisión y recall, $F1 = 2 \frac{PR}{P+R}$.

Fusión detección–profundidad Proceso de asignar a cada detección 2D una medida de profundidad relativa para derivar distancia y velocidad mano–cabeza.

Headbanging Conducta de golpeo cefálico repetitivo, a veces presente como conducta autolesiva.

Matriz de confusión Tabla de conteos que contrasta clases verdaderas vs. clases predichas.

NPY Formato binario de NumPy para almacenar arreglos; en este proyecto guarda mapas de profundidad por cuadro.

Precisión Proporción de verdaderos positivos sobre todas las predicciones positivas.

Profundidad relativa Medida de proximidad por cuadro sin escala métrica absoluta; útil para comparar distancias *dentro* de la misma escena.

Recuperación (Recall) Proporción de verdaderos positivos sobre todas las instancias positivas reales.

ROI (Region of Interest) Subárea de la imagen usada para calcular estadísticas de profundidad.

TEA Trastorno del Espectro Autista; condición del neurodesarrollo caracterizada por diferencias en comunicación, interacción social y conductas repetitivas.

YOLOv8s Variante ligera del detector de objetos *You Only Look Once* (versión 8), usada para detección en tiempo real de mano, cabeza y golpe.

Siglas

CPU Central Processing Unit.

FN Falsos Negativos.

FP Falsos Positivos.

FPS Frames Per Second.

GPU Graphics Processing Unit.

IoU Intersection over Union.

NMS Non-Maximum Suppression.

PR Precisión–Recall.

ROI Region of Interest.

TP True Positive.

Tabla de Contenido

1	Introducción	1
2	Marco General	3
2.1	Definición del Problema	3
2.2	Acercamiento de la Solución	4
2.3	Descripción del Sistema	5
2.4	Tecnologías Utilizadas	6
2.4.1	Roboflow	6
2.4.2	YOLOv8	7
2.4.3	Python	7
2.4.4	Depth Anything V2	8
2.4.5	Resumen de Ventajas y Desventajas	9
3	Estado del Arte	10
3.1	Detección de objetos en tiempo real para escenas dinámicas	10
3.2	Reconocimiento de gestos y conductas en TEA: importancia de la di- námica temporal	10
3.3	Detección de cabeza y regiones críticas	11
3.4	Estimación de movimiento y profundidad para cuantificar impactos . . .	11
4	Objetivos	12
4.1	Objetivo General	12
4.2	Objetivos Específicos	12
5	Requisitos del Sistema	13
5.1	Requisitos funcionales	13
5.2	Requisitos No Funcionales	14
5.3	Requisitos de interfaces	16
5.4	Requisitos de Ambiente	17

TABLA DE CONTENIDO

5.5	Perfiles de Usuario	19
6	Diseño de Arquitectura del Sistema	21
6.1	Diagrama de contexto	21
6.2	Diagrama de arquitectura	22
6.3	Descripción de módulos	23
6.3.1	Módulo de Recopilación de Datos (MRD)	23
6.3.2	Módulo de Dataset (MD)	24
6.3.3	Módulo de Entrenamiento (ME)	25
6.3.4	Módulo de Profundidad (MP)	26
6.3.5	Módulo de Detección y Análisis (MDA)	27
6.3.6	Módulo de Video Anotado (MVA)	28
6.3.7	Módulo de Reporte (MR)	29
7	Gestión de Riesgos	31
7.1	Supuestos	31
7.2	Dependencias	32
7.3	Restricciones	32
7.4	Riesgos	33
8	Desarrollo	36
8.1	Introducción a la arquitectura YOLOv8 + Depth Anything V2	36
8.2	Proceso de entrenamiento con Roboflow y YOLOv8	37
8.2.1	Análisis de matrices de confusión	39
8.2.2	Curvas F1–Confianza y Recall–Confianza	41
8.2.3	Curvas Precision–Confianza y Precision–Recall	43
8.3	Detección con YOLOv8s	45
8.4	Estimación de profundidad monocular con <i>Depth Anything</i> (mapas re- lativos)	46
8.4.1	Ejemplo visual	48

8.4.2	Combinación: YOLOv8l + Depth Anything V2	49
8.5	Cálculo de la velocidad relativa mano–cabeza	51
8.5.1	Profundidad por objeto (mano/cabeza).	52
8.5.2	Separación relativa por cuadro.	52
8.5.3	Elección del par temporal y Δt	53
8.5.4	Variación de distancia y velocidad	53
8.5.5	Etiqueta de intensidad (según v_t en m/s).	53
8.5.6	Entradas y salidas del módulo	53
9	Conclusión	55
9.1	Trabajo Futuro	57

Índice de figuras

6.1	Diagrama de contexto	22
6.2	Diagrama de arquitectura	23
8.1	Imágenes train/val/test	38
8.2	Matriz de confusión absoluta en validación. Columnas: clase verdadera; filas: clase predicha. La diagonal corresponde a aciertos (TP); las celdas fuera de la diagonal reflejan errores (FN/FP) según su orientación.	40
8.3	Matriz de confusión normalizada por columna en validación. La normalización permite interpretar el <i>recall</i> de cada clase como el valor de la diagonal en su respectiva columna.	41
8.4	Curva F1–Confianza en validación. La línea agregada (azul grueso) alcanza $\approx 0,98$ en $\text{conf} \approx 0,368$; las líneas finas corresponden a cada clase (cabeza, golpe, mano).	42
8.5	Curva Recall–Confianza en validación. El recall macro parte cercano a 1,00 y decrece al aumentar conf ; se observa una caída pronunciada a partir de $\text{conf} \in [0,8, 0,9]$	43

8.6	Curva Precision–Confianza en validación. La curva agregada alcanza precisión $\approx 1,00$ en $\text{conf} \approx 0,879$; las líneas individuales muestran el mismo patrón creciente por clase.	44
8.7	Curva Precision–Recall en validación. Se reportan $\text{AP}@0.5$ por clase (cabeza 0.989, golpe 0.995, mano 0.986) y $\text{mAP}@0.5 = 0.990$ (macro). 45	
8.8	Evolución temporal de las detecciones con YOLOv8s . El aumento de k de 3 a 4 corresponde a la aparición de golpe . El contador acumulado <i>Golpes</i> : N sólo se incrementa si la hipótesis persiste en una ventana temporal y se valida su coherencia inter-clase, evitando inflar eventos por detecciones efímeras.	46
8.9	Izquierda: cuadro RGB del video. Derecha: mapa de profundidad relativa estimado con Depth Anything V2 y normalizado para análisis y visualización.	49
8.10	Vista conceptual de la fusión: a la izquierda, detecciones 2D con YOLOv8l ; a la derecha, el mapa de profundidad relativa de Depth Anything V2 . La idea es asociar a cada caja (p. ej., mano, cabeza) un valor de profundidad relativa obtenido del mapa del mismo cuadro.	50

Índice de tablas

1	Resumen de ventajas y desventajas de las tecnologías utilizadas en el sistema	9
2	Requisitos de interfaces del sistema: entradas y salidas principales. . . .	16
3	Perfil de usuario: Ingeniero Biomédico / Científico de Datos	19
4	Perfil de usuario: Investigador Clínico	20
5	Perfil de usuario: Estudiante / Investigador en Formación	21
6	Módulo de Recopilación de Datos (MRD)	24
7	Módulo de Dataset (MD)	25
8	Módulo de Entrenamiento (ME)	26

ÍNDICE DE TABLAS

9	Módulo de Profundidad (MP)	27
10	Módulo de Detección y Análisis (MDA)	28
11	Módulo de Video Anotado (MVA)	29
12	Módulo de Reporte (MR)	30

1. Introducción

El Trastorno del Espectro Autista (TEA) es una condición del neurodesarrollo que afecta de manera significativa la comunicación, el comportamiento social y la interacción con el entorno. En niños diagnosticados con TEA, se observa una amplia gama de comportamientos atípicos que incluyen movimientos repetitivos, dificultades para expresar emociones, y en algunos casos, conductas autolesivas como golpes dirigidos a la cabeza (*headbanging*). [9] Este tipo de comportamiento representa un desafío importante tanto para los profesionales de la salud como para los cuidadores, ya que puede derivar en lesiones físicas, dificultades en el aprendizaje y deterioro en la calidad de vida del niño.

Actualmente, la evaluación clínica de estos comportamientos se realiza principalmente a través de la observación directa o mediante entrevistas estructuradas a padres y cuidadores. Si bien estas estrategias pueden ser útiles en determinados contextos, presentan limitaciones importantes en cuanto a objetividad, precisión temporal y escalabilidad. Además, muchas de estas evaluaciones ocurren fuera del contexto cotidiano del niño, lo que puede afectar la naturalidad del comportamiento observado y limitar la efectividad del diagnóstico o tratamiento. En consecuencia, existe una necesidad urgente de desarrollar herramientas tecnológicas que permitan una monitorización continua, no invasiva y objetiva del comportamiento de niños con TEA en entornos naturales como el hogar o la sala de clases [6].

En este escenario, los avances recientes en visión por computador, aprendizaje profundo y estimación de movimiento abren nuevas posibilidades para abordar este tipo de problemáticas. Modelos como YOLO (You Only Look Once), especialmente en sus versiones más recientes como YOLOv8, han demostrado ser altamente eficaces para tareas de detección de objetos en tiempo real. Su capacidad para identificar y clasificar múltiples objetos en escenas dinámicas, sumada a su eficiencia computacional, los convierte en candidatos ideales para ser implementados en sistemas de monitoreo continuo.

El presente trabajo propone el desarrollo de un sistema de evaluación automática de impactos cefálicos en niños con TEA, utilizando técnicas de visión por computador. En particular, se integra un modelo de detección basado en YOLOv8 con un estimador de profundidad monocular denominado *Depth Anything V2*, permitiendo no solo detectar visualmente el evento de golpeo, sino también estimar la velocidad relativa entre la mano y la cabeza para inferir la intensidad del impacto. Este enfoque se apoya en la construcción de un conjunto de datos personalizado, compuesto por imágenes etiquetadas de situaciones simuladas de golpeo, las cuales fueron procesadas y organizadas a través de la plataforma Roboflow.

La arquitectura del sistema está diseñada para operar con videos capturados en entornos clínicos, y genera como salida visualizaciones anotadas, registros de eventos en texto, y un análisis cuantitativo de los impactos detectados. Esta propuesta no busca reemplazar a los métodos clínicos tradicionales, sino más bien complementarlos con una herramienta de evaluación objetiva que pueda ser utilizada de forma continua, sin necesidad de supervisión directa, y adaptable a distintos escenarios terapéuticos.

El enfoque adoptado en esta tesis considera también aspectos de robustez técnica, como el suavizado de los valores de profundidad mediante operaciones estadísticas locales (mediana sobre regiones 5×5), y la implementación de un filtro para descartar eventos no válidos (por ejemplo, golpes con velocidad cero o con diferencias de tiempo ínfimas). Además, el sistema clasifica cada evento en categorías de intensidad (*leve*, *media* o *fuerte*) según la magnitud de la velocidad relativa calculada, lo cual representa un aporte innovador al análisis automático de comportamientos autolesivos.

Desde el punto de vista práctico, la principal motivación de este proyecto es aportar una solución tecnológica concreta al monitoreo de comportamientos de riesgo en niños con TEA, con un enfoque centrado en la accesibilidad, la escalabilidad y la integración futura en plataformas más complejas.

2. Marco General

2.1. Definición del Problema

El Trastorno del Espectro Autista (TEA) presenta una amplia gama de manifestaciones conductuales, entre las que se encuentran movimientos repetitivos y, en algunos casos, conductas autolesivas como los golpes dirigidos a la cabeza (*headbanging*) [9]. Este tipo de comportamiento, además de generar un riesgo inmediato de lesiones físicas, puede tener repercusiones a largo plazo en el desarrollo cognitivo, el aprendizaje y la calidad de vida del niño.

En la práctica clínica actual, la identificación y evaluación de estos eventos se basa principalmente en observaciones realizadas por cuidadores, terapeutas o profesionales de la salud, así como en entrevistas estructuradas y registros cualitativos. Este enfoque tradicional presenta limitaciones significativas:

- **Subjetividad en la evaluación:** la interpretación de la intensidad y frecuencia del impacto depende de la percepción del observador, lo que puede conducir a variaciones y sesgos.
- **Falta de continuidad:** la supervisión directa no es constante, lo que impide registrar todos los episodios que ocurren fuera del entorno clínico.
- **Entornos no representativos:** muchas evaluaciones se realizan en espacios controlados, que no siempre reflejan el comportamiento natural del niño en su vida diaria.
- **Ausencia de datos cuantitativos precisos:** las observaciones no suelen incluir métricas físicas como velocidad, distancia o energía del impacto, dificultando un análisis objetivo.

La ausencia de un sistema capaz de detectar de manera automática, precisa y en tiempo real los impactos cefálicos en niños con TEA representa una brecha importante

en la prevención y seguimiento de conductas autolesivas. Este vacío impide implementar intervenciones preventivas oportunas, dificulta el seguimiento objetivo de la evolución del paciente y limita la capacidad de los cuidadores para actuar de forma inmediata ante episodios de riesgo [6].

Si bien la visión por computador y las técnicas de aprendizaje profundo ofrecen una solución prometedora, su implementación práctica requiere hardware con capacidades de procesamiento gráfico avanzadas, lo que incrementa los costos y limita la accesibilidad para un uso doméstico generalizado. No obstante, la adopción de este tipo de tecnología en entornos clínicos o de investigación podría aportar datos precisos y continuos que complementen las evaluaciones tradicionales y faciliten la toma de decisiones terapéuticas, contribuyendo a una mejor prevención de lesiones y un seguimiento más riguroso de la conducta del paciente.

2.2. Acercamiento de la Solución

Para abordar las limitaciones identificadas en la evaluación de impactos cefálicos en niños con TEA, se propone el desarrollo de un sistema automatizado que combine técnicas de visión por computador y aprendizaje profundo. Este sistema integra dos componentes principales: un modelo de detección de objetos de última generación (**YOLOv8**) y un estimador de profundidad monocular (**Depth Anything V2**).

El modelo **YOLOv8** se encarga de detectar en tiempo real las clases relevantes para el problema: *mano*, *cabeza* y *golpe*. Estas detecciones se realizan cuadro a cuadro a partir de videos capturados en entornos domésticos o clínicos, lo que permite localizar con precisión las regiones de interés. Por su parte, **Depth Anything V2** genera mapas de profundidad relativos para cada imagen, posibilitando la estimación de la distancia entre la mano y la cabeza en el momento del impacto.

El procesamiento conjunto de la información de detección y profundidad permite calcular la velocidad relativa del golpe a partir de la variación de distancia y el tiempo transcurrido entre eventos, clasificando la intensidad del impacto en categorías defini-

das: *leve*, *media* o *fuerte*. Este enfoque no solo confirma visualmente el evento, sino que también proporciona una cuantificación física que favorece un análisis más objetivo y reproducible.

Dada la alta demanda de cómputo de los modelos utilizados, la implementación del sistema requiere hardware con capacidades de procesamiento gráfico avanzadas. Por este motivo, su uso inicial se proyecta principalmente en entornos clínicos o de investigación, donde existan los recursos necesarios para su ejecución en tiempo real.

2.3. Descripción del Sistema

El sistema propuesto para la detección y evaluación de impactos cefálicos en niños con TEA está diseñado como una solución modular que combina detección de objetos y estimación de profundidad, permitiendo identificar eventos de golpeo y cuantificar su intensidad de forma automática. La arquitectura general del sistema se organiza en tres componentes principales:

- **Adquisición de datos:** Captura de secuencias de video en entornos clínicos o de investigación, que posteriormente se procesan mediante un modelo de estimación de profundidad para generar los mapas de profundidad en formato `.npy`. Estos archivos son utilizados junto con el video original en la etapa de procesamiento para detectar impactos y calcular métricas de intensidad.
- **Procesamiento:** Aplicación de un modelo de detección de objetos para identificar las clases relevantes (*mano*, *cabeza*, *golpe*) y cálculo de la velocidad relativa e intensidad del impacto a partir de la información de profundidad.
- **Generación de resultados:** Creación de salidas visuales y registros estructurados que contienen la información de cada evento, incluyendo la clasificación de la intensidad del golpe y los videos anotados.

Este diseño modular permite que cada componente sea independiente pero alta-

mente integrado, facilitando la actualización de modelos, la incorporación de nuevas métricas o la adaptación del sistema a distintos escenarios de análisis sin alterar la estructura global.

2.4. Tecnologías Utilizadas

Para implementar los componentes descritos en el sistema, se seleccionaron tecnologías clave que permiten garantizar un funcionamiento eficiente y preciso. A continuación, se enumeran las principales, asociadas al componente correspondiente:

- **Adquisición de datos:** Roboflow, Python.
- **Procesamiento:** YOLOv8, Depth Anything V2.
- **Generación de resultados:** Python.

2.4.1. Roboflow

Descripción: Roboflow [2] es una plataforma en línea para la gestión de datasets, incluyendo etiquetado, preprocesamiento y exportación de datos para modelos de visión por computador.

Funcionamiento Básico: Permite cargar imágenes o videos, etiquetarlos de forma manual o semiautomática, aplicar técnicas de aumento de datos y exportar el dataset en múltiples formatos compatibles con frameworks como YOLO.

Ventajas:

- Interfaz intuitiva y fácil de usar.
- Integración directa con Google Colab y entornos de entrenamiento.

Desventajas:

- Dependencia de conexión a internet.

- Algunas funciones avanzadas requieren suscripción de pago.

Justificación en el Proyecto: Se utilizó para construir y gestionar el dataset personalizado, así como para facilitar el entrenamiento del modelo YOLOv8.

2.4.2. YOLOv8

Descripción: YOLOv8 [5] es un modelo de detección de objetos de última generación desarrollado por Ultralytics, optimizado para lograr alta precisión y velocidad en la detección en tiempo real.

Funcionamiento Básico: Divide la imagen en una cuadrícula y predice, en una sola pasada por la red, las coordenadas de los cuadros delimitadores y la probabilidad de cada clase detectada.

Ventajas:

- Alta velocidad de inferencia en tiempo real.
- Precisión elevada en detección de objetos.

Desventajas:

- Requiere hardware potente para un rendimiento óptimo.
- Puede ser menos eficiente en detecciones de objetos muy pequeños.

Justificación en el Proyecto: Se emplea para la detección de las clases *mano*, *ca-beza* y *golpe* en cada cuadro del video.

2.4.3. Python

Descripción: Python es un lenguaje de programación de alto nivel ampliamente utilizado en inteligencia artificial, visión por computador y análisis de datos.

Funcionamiento Básico: Utiliza un intérprete para ejecutar código de forma secuencial y cuenta con un ecosistema extenso de librerías para procesamiento de imágenes, datos y machine learning.

Ventajas:

- Sintaxis simple y legible.
- Gran cantidad de librerías y comunidad activa.

Desventajas:

- Rendimiento inferior comparado con lenguajes compilados.
- Consumo de recursos más alto en tareas muy intensivas.

Justificación en el Proyecto: Es el lenguaje principal para integrar el flujo completo del sistema, desde la detección y estimación de profundidad hasta la generación de resultados.

2.4.4. Depth Anything V2

Descripción: Depth Anything V2 [11] es un modelo de estimación de profundidad monocular que predice mapas de profundidad a partir de imágenes RGB sin necesidad de sensores especializados.

Funcionamiento Básico: Analiza patrones visuales y relaciones espaciales para asignar un valor de profundidad relativo a cada píxel de la imagen.

Ventajas:

- No requiere hardware especializado como cámaras estéreo.
- Fácil integración con pipelines de visión por computador.

Desventajas:

- Los valores son relativos y no métricos absolutos.
- Precisión dependiente de la calidad de la imagen.

Justificación en el Proyecto: Permite calcular la distancia relativa entre manos y cabeza para determinar la velocidad e intensidad de los impactos detectados.

2.4.5. Resumen de Ventajas y Desventajas

Tecnología	Ventajas	Desventajas
Roboflow	Interfaz intuitiva, integración con Colab	Dependencia de internet, funciones avanzadas de pago
YOLOv8	Alta velocidad y precisión	Requiere hardware potente, menos eficiente en objetos pequeños
Python	Sintaxis simple, gran ecosistema de librerías	Rendimiento inferior a lenguajes compilados, alto consumo de recursos en tareas intensivas
Depth Anything V2	No requiere hardware especializado, fácil integración	Valores de profundidad relativos, precisión dependiente de la imagen

Tabla 1: Resumen de ventajas y desventajas de las tecnologías utilizadas en el sistema

3. Estado del Arte

3.1. Detección de objetos en tiempo real para escenas dinámicas

La detección en video con restricciones de latencia ha evolucionado notablemente con la familia *YOLO*. Estudios comparativos recientes muestran que **YOLOv8** ofrece un equilibrio sólido entre precisión y velocidad en secuencias rápidas, superando a variantes previas como **YOLOv5** y compitiendo con arquitecturas más recientes como **YOLOv9** en contextos de alta exigencia temporal [7]. Resultados adicionales en reconocimiento de gestos manuales confirman la robustez de YOLOv8 frente a cambios de escala, iluminación y pose, respaldando su idoneidad para aplicaciones que requieren respuesta en tiempo real [4]. En conjunto, estas evidencias justifican el uso de **YOLOv8** como base para la detección de manos, cabeza y eventos de golpeo en este proyecto.

3.2. Reconocimiento de gestos y conductas en TEA: importancia de la dinámica temporal

La literatura subraya que la mera detección estática es insuficiente para caracterizar conductas complejas asociadas al TEA. En particular, se han propuesto enfoques que combinan detección cuadro a cuadro con modelos *temporales* para capturar patrones repetitivos y su evolución en el tiempo. Por ejemplo, la integración de **YOLOv7** con **VideoMAE** y estrategias de aumento de video mejora la robustez ante variaciones de entorno y postura [8]. De forma complementaria, el uso de descriptores de **puntos clave** procesados con **CNN-LSTM** ha demostrado eficacia para distinguir *headbanging* de movimientos no relevantes en videos no controlados [10]. Estas líneas de trabajo motivan, en esta tesis, el uso de una señal dinámica explícita basada en **velocidad relativa mano–cabeza** para estimar la *intensidad* del impacto.

3.3. Detección de cabeza y regiones críticas

La localización estable de la cabeza es un componente crítico para el análisis de impactos. Extensiones de **YOLOv8** centradas en este objetivo, como la incorporación de módulos de *feature fusion* auxiliares y pérdidas calibradas al ruido, reportan mejoras en escenarios densos y con oclusiones [1]. Aunque dichas modificaciones no se adoptan en este trabajo, sus resultados respaldan la elección de YOLOv8 como marco detector y enfatizan la necesidad de una **detección confiable de la cabeza** para cuantificar correctamente el acercamiento mano–cabeza.

3.4. Estimación de movimiento y profundidad para cuantificar impactos

Más allá de la detección, la **cuantificación** del evento requiere incorporar señales de movimiento y distancia. Los enfoques de estado del arte integran información temporal (p. ej., *optical flow*, modelos de secuencia) o **profundidad monocular** para inferir proximidad y velocidad relativa [10]. Siguiendo esta dirección, el sistema propuesto combina detecciones de **YOLOv8** con mapas de profundidad estimados (e. g., *Depth Anything V2*) para calcular $\Delta d/\Delta t$ y clasificar la *intensidad* del impacto, aportando una medida objetiva que complementa la evaluación clínica.

4. Objetivos

4.1. Objetivo General

Fortalecer el monitoreo clínico objetivo y no invasivo de conductas autolesivas en niños con Trastorno del Espectro Autista (TEA) mediante un enfoque de visión por computador capaz de detectar eventos de golpeo cefálico y cuantificar su intensidad a partir de video, entregando evidencia cuantificable y reproducible para la toma de decisiones en entornos clínicos o de investigación.

4.2. Objetivos Específicos

- Diseñar un flujo de procesamiento que combine detección de objetos e información de profundidad para analizar eventos de impacto.
- Entrenar un modelo YOLOv8 personalizado para la detección de las clases relevantes: *mano*, *cabeza* y *golpe*.
- Generar mapas de profundidad a partir de videos utilizando el modelo Depth Anything V2, y emplearlos para calcular la velocidad relativa del impacto.
- Clasificar automáticamente la intensidad de los impactos en categorías (*leve*, *media*, *fuerte*) en función de la velocidad estimada.
- Implementar un sistema de generación de resultados que incluya videos anotados, registros en formato estructurado y métricas cuantitativas para cada evento.

5. Requisitos del Sistema

5.1. Requisitos funcionales

- **RF1 – Detección de clases:** El sistema debe ser capaz de detectar de manera precisa las clases *mano*, *cabeza* y *golpe* en secuencias de video, utilizando un modelo de detección de objetos basado en YOLOv8 entrenado sobre un dataset personalizado. La detección debe realizarse en cada cuadro del video, permitiendo identificar la presencia y localización de las regiones de interés en tiempo real o cercano al tiempo real.
- **RF2 – Procesamiento de video y profundidad:** El sistema debe ser capaz de procesar simultáneamente videos de entrada junto con sus correspondientes mapas de profundidad en formato `.npy`, los cuales han sido generados previamente por el modelo Depth Anything V2. Este requisito garantiza que el análisis de impactos se realice considerando tanto la información visual como la espacial.
- **RF3 – Cálculo de velocidad relativa:** El sistema debe calcular la velocidad relativa entre *mano* y *cabeza* en el instante del impacto, utilizando para ello la variación de valores de profundidad entre cuadros consecutivos y el tiempo transcurrido entre ellos. Este cálculo permitirá cuantificar el movimiento en términos físicos y proporcionar un parámetro objetivo para la evaluación del impacto.
- **RF4 – Clasificación de intensidad:** El sistema debe clasificar automáticamente la intensidad del impacto en tres categorías predeterminadas: *leve*, *media* y *fuerte*, basándose en los valores obtenidos de velocidad relativa. Esta clasificación permitirá interpretar los resultados de manera comprensible y estandarizada para los usuarios clínicos.
- **RF5 – Generación de video anotado:** El sistema debe generar un video de salida anotado, en el cual se visualicen los cuadros delimitadores de las detecciones realizadas, las etiquetas de clase correspondientes y la intensidad de cada impacto

detectado. Este video anotado constituye una herramienta visual de validación y comunicación de resultados.

- **RF6 – Exportación de registros:** El sistema debe exportar registros estructurados de cada evento de impacto en archivos de salida (CSV y TXT). Estos registros deben contener información detallada como: número de cuadro, tiempo del evento, velocidad calculada, intensidad clasificada y coordenadas de las detecciones, asegurando así la trazabilidad y análisis posterior de los resultados.
- **RF7 – Almacenamiento de imágenes:** El sistema debe ser capaz de almacenar de manera automática las imágenes correspondientes a cada cuadro donde se confirme un evento de impacto cefálico, facilitando la revisión visual y el uso de estos datos en futuras etapas de entrenamiento o validación.

5.2. Requisitos No Funcionales

- **RNF1 – Rendimiento en tiempo real:** El sistema debe ser capaz de operar con un tiempo de procesamiento cercano al tiempo real, minimizando la latencia entre la entrada del video y la generación de resultados. Aunque la eficiencia dependerá del hardware disponible, se espera que el sistema mantenga un flujo continuo de análisis sin interrupciones perceptibles.
- **RNF2 – Precisión del sistema:** El sistema debe garantizar un nivel de precisión superior al 95 % en la detección de las clases entrenadas (*mano, cabeza y golpe*). Este requisito busca reducir al mínimo los falsos positivos y negativos, asegurando que los resultados sean confiables y clínicamente relevantes.
- **RNF3 – Robustez ante variaciones:** El sistema debe ser robusto frente a variaciones en las condiciones de grabación, tales como cambios en la iluminación, ángulos de cámara, resolución del video o posturas del sujeto. Esto permitirá que la detección y el análisis de impactos se mantengan estables en escenarios clínicos y de investigación no controlados.

- **RNF4 – Modularidad del diseño:** El sistema debe estar construido bajo un enfoque modular, permitiendo que los distintos componentes (detección de objetos, estimación de profundidad, análisis de impactos y generación de resultados) puedan actualizarse, reemplazarse o ampliarse sin afectar el funcionamiento global.
- **RNF5 – Portabilidad del sistema:** El sistema debe ser portable a distintos entornos de ejecución, garantizando su correcto funcionamiento tanto en laboratorios clínicos como en centros de investigación, siempre que se disponga de hardware compatible con GPU. Esto asegura que la herramienta no quede limitada a un único contexto o configuración.
- **RNF6 – Documentación y mantenibilidad:** El sistema debe contar con documentación técnica completa y clara, incluyendo instrucciones de instalación, uso, mantenimiento y resolución de problemas. Esta documentación facilitará la comprensión del sistema por parte de investigadores y técnicos, además de permitir su futura adaptación o ampliación.

5.3. Requisitos de interfaces

La Tabla 2 presenta los requisitos asociados a las interfaces de entrada y salida del sistema. Dado que la ejecución se realiza en entornos de desarrollo como Jupyter Notebook, no existen interfaces gráficas de usuario, sino flujos de datos que permiten la integración entre los módulos.

Evento	Descripción	Parámetros	Respuesta
Recepción de video	El sistema carga un archivo de video como entrada para el análisis.	Formato de video (MP4/AVI), Resolución	Video cargado y disponible para procesamiento.
Recepción de mapas de profundidad	El sistema recibe archivos .npy generados previamente con Depth Anything V2.	Archivos .npy, Resolución de mapas, Ruta de almacenamiento	Mapas de profundidad almacenados y sincronizados con el video.
Generación de resultados visuales	Creación de un video anotado que muestra detecciones y niveles de intensidad.	Video procesado, Detecciones confirmadas	Video de salida exportado en MP4.
Generación de reportes	Exportación de métricas y frames asociados a los eventos detectados.	Formato de salida (CSV, TXT, JPG)	Archivos generados con registros detallados e imágenes asociadas.

Tabla 2: *Requisitos de interfaces del sistema: entradas y salidas principales.*

5.4. Requisitos de Ambiente

Hardware de Desarrollo

- Estación de trabajo o servidor con **GPU NVIDIA RTX 3090 / 4090 (mínimo 24 GB VRAM)** o superior, para permitir la ejecución en tiempo cercano al real de videos de 30 a 60 minutos de duración.
- **Memoria RAM mínima de 32 GB**, recomendada 64 GB, para garantizar la carga y manipulación simultánea de videos largos y mapas de profundidad en formato .npy.
- **Procesador multinúcleo de alto rendimiento** (mínimo 16 núcleos físicos, ej. Intel i9, AMD Ryzen 9, Threadripper o equivalentes) que permita el preprocesamiento y análisis de datos en paralelo.
- **Unidad SSD NVMe de al menos 2 TB** para almacenar de manera eficiente los videos de entrada, mapas de profundidad, resultados intermedios y salidas generadas.
- Conexión de red rápida (≥ 1 Gbps) en caso de transferencia de grandes volúmenes de datos entre servidores o entornos de investigación.

Software de Desarrollo

- **Sistema Operativo:** Ubuntu 20.04/22.04 LTS o Windows 10/11 Pro con drivers NVIDIA CUDA/cuDNN actualizados para soporte de procesamiento acelerado en GPU.
- **Lenguaje de Desarrollo:** Python 3.10 o superior, elegido por su ecosistema amplio en inteligencia artificial, visión por computador y análisis de datos.
- **Frameworks de Deep Learning:**

- **PyTorch 2.0+:** framework principal para la ejecución de modelos de detección (YOLOv8) y estimación de profundidad (Depth Anything V2).
- **Ultralytics YOLOv8:** librería especializada para detección en tiempo real de objetos, entrenada con dataset personalizado.
- **Bibliotecas de visión por computador y análisis:**
 - **OpenCV 4.8+:** para procesamiento de imágenes y videos, manipulación de frames, anotación de detecciones y exportación de resultados visuales.
 - **NumPy 1.26+:** para manejo de matrices y cálculo numérico eficiente, especialmente para los mapas de profundidad.
 - **Pandas 2.0+:** para estructuración, exportación y análisis de datos en formato tabular (CSV, TXT).
 - **Matplotlib / Seaborn:** para visualización de métricas y generación de gráficos interpretativos.
- **Modelos y librerías específicas:**
 - **Depth Anything V2:** modelo de estimación de profundidad monocular, integrado al pipeline para generar los archivos `.npy`.
 - **Torchvision:** librería auxiliar para transformaciones de imágenes y compatibilidad con PyTorch.

5.5. Perfiles de Usuario

Perfil	Socioeconómico y Cultural	Ocupacional	Etario	Competencias y Características
Ingeniero Biomédico / Científico de Datos	Nivel socioeconómico medio; inserto en entornos de investigación tecnológica.	Profesional técnico/científico con conocimientos en programación, IA y visión por computador.	25–45 años.	Dominio de Python y librerías de IA; análisis de métricas; optimización de ejecución; soporte a equipos clínicos.

Tabla 3: Perfil de usuario: Ingeniero Biomédico / Científico de Datos

CAPÍTULO 5 : REQUISITOS DEL SISTEMA

Perfil	Socioeconómico y Cultural	Ocupacional	Etario	Competencias y Características
Investigador Clínico	Nivel socioeconómico medio–alto; contexto cultural vinculado a la salud y la investigación.	Profesional del área médica especializado en neurodesarrollo o psiquiatría infantil.	30–60 años.	Experiencia clínica; capacidad de análisis de resultados; interés en herramientas tecnológicas para apoyo diagnóstico.

Tabla 4: Perfil de usuario: Investigador Clínico

Perfil	Socioeconómico y Cultural	Ocupacional	Etario	Competencias y Características
Estudiante / Investigador en Formación	Nivel socioeconómico variable; contexto cultural académico.	Estudiante de posgrado o investigador junior.	20–35 años.	Conocimientos básicos de programación; interés en investigación aplicada; capacidad de aprendizaje supervisado; apoyo en preparación de datos.

Tabla 5: Perfil de usuario: Estudiante / Investigador en Formación

6. Diseño de Arquitectura del Sistema

6.1. Diagrama de contexto

El diagrama de contexto de la Figura 5.1 muestra la interacción principal entre los actores externos y el sistema desarrollado. En el extremo izquierdo se ubican los pacientes, quienes participan en sesiones de grabación de movimientos. Estas grabaciones constituyen la fuente primaria de datos de entrada que alimenta al software de identificación y evaluación.

El software, ubicado en el centro del diagrama, procesa los videos capturados aplicando técnicas de visión por computador y análisis de profundidad, con el fin de iden-

tificar eventos de golpeo dirigidos hacia la cabeza y calcular métricas objetivas relacionadas con la intensidad del impacto.

En el extremo derecho se encuentran los médicos, quienes actúan como usuarios finales del sistema. Ellos acceden a los resultados generados, los cuales incluyen videos anotados, reportes en formato estructurado y evidencias gráficas. Dichos resultados permiten complementar la evaluación clínica tradicional mediante información objetiva y cuantificable, facilitando la toma de decisiones en contextos de investigación o intervención terapéutica.

De esta manera, el diagrama refleja la naturaleza del sistema como un intermediario entre los datos capturados de los pacientes y el análisis clínico realizado por los especialistas, mostrando un flujo unidireccional de entrada de información y un flujo de salida de resultados procesados hacia los profesionales de la salud.



Figura 6.1: Diagrama de contexto

6.2. Diagrama de arquitectura

El diagrama de arquitectura de la Figura 5.2 muestra la organización modular del sistema y el flujo de información entre sus componentes. El proceso comienza con el Módulo de Recopilación de Datos, encargado de reunir los videos crudos, que luego son gestionados en el Módulo de Dataset para su anotación y preparación.

A partir de este dataset, el Módulo de Entrenamiento genera el modelo YOLOv8 entrenado, mientras que el Módulo de Profundidad produce los mapas con información

espacial. Ambos insumos confluyen en el Módulo de Detección y Análisis, que constituye el núcleo del sistema, encargado de identificar los golpes y calcular su intensidad.

Finalmente, los resultados se distribuyen en dos salidas principales: el Módulo de Video Anotado, que genera un archivo MP4 con las detecciones visuales, y el Módulo de Reporte, que produce registros estructurados en CSV/TXT junto con imágenes de evidencia.

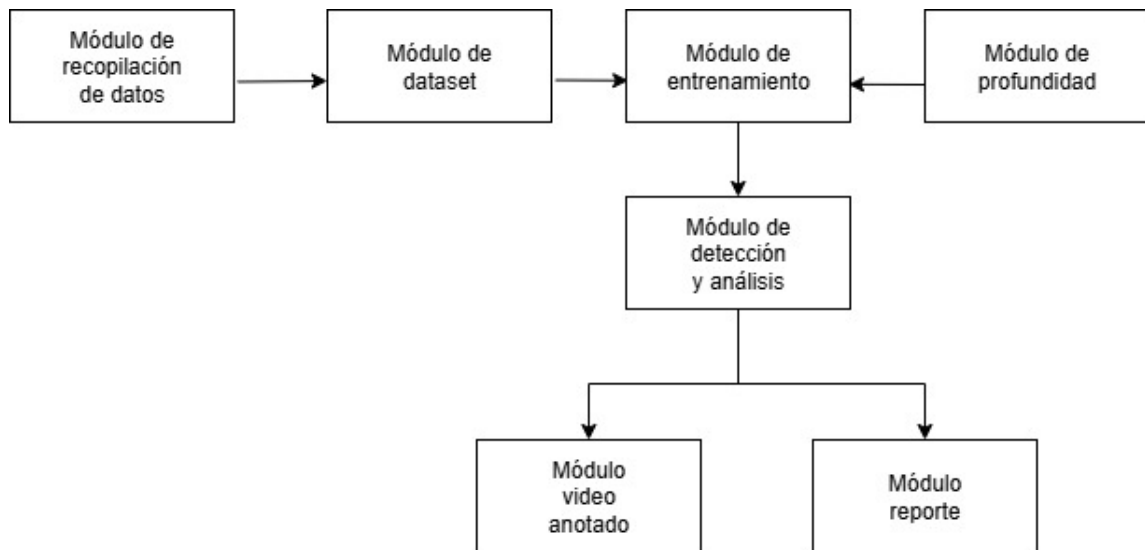


Figura 6.2: Diagrama de arquitectura

6.3. Descripción de módulos

6.3.1. Módulo de Recopilación de Datos (MRD)

El **MRD** es el punto de partida del sistema: concentra la obtención de videos crudos de sesiones clínicas o simuladas junto con metadatos básicos, resguardando criterios éticos, de privacidad y trazabilidad antes de su uso en el resto del pipeline.

Propósito	Proveer insumos audiovisuales válidos, trazables y autorizados que alimenten la construcción del dataset y la validación del sistema.
Alcance	Adquisición/recepción de videos; registro de metadatos (fecha, dispositivo, condiciones de captura); resguardo inicial en almacenamiento seguro. No realiza anotación ni procesamiento.
Dependencias	Dispositivos de captura, repositorios/autorizaciones de acceso, políticas de consentimiento y almacenamiento seguro.
Supuestos	Los videos cumplen estándares mínimos de calidad (iluminación, resolución, estabilidad) y cuentan con permisos de uso.
Restricciones	Límites de espacio en disco, compatibilidad de formatos, y cumplimiento normativo/ético.

Tabla 6: Módulo de Recopilación de Datos (MRD)

6.3.2. Módulo de Dataset (MD)

El **MD** transforma los videos recopilados en un conjunto de datos utilizable: etiqueta clases (mano, cabeza, golpe), controla versiones y prepara particiones y exportaciones compatibles con YOLOv8.

Propósito	Disponer de un dataset limpio, balanceado y versionado para entrenamiento/validación.
Alcance	Anotación y revisión de etiquetas en Roboflow; augmentations; definición de <i>splits</i> (train/val/test); exportación en formato YOLO/Ultralytics.
Dependencias	MRD (videos crudos), plataforma de etiquetado (p. ej., Roboflow), guías de anotación.
Supuestos	Etiquetado consistente y cobertura de variabilidad (ángulos, iluminación, distancias).
Restricciones	Tiempo/costo de etiquetado; sesgos de muestreo; límites de la plataforma para tamaños de dataset.

Tabla 7: Módulo de Dataset (MD)

6.3.3. Módulo de Entrenamiento (ME)

El **ME** entrena el detector basado en YOLOv8 con el dataset anotado y selecciona el mejor conjunto de pesos (`best.pt`) de acuerdo con métricas estándar.

Propósito	Obtener un modelo robusto para detectar mano, cabeza y golpe en video.
Alcance	Configuración de hiperparámetros; entrenamiento supervisado; validación con precisión/recall/mAP; selección de <code>best.pt</code> .
Dependencias	MD (dataset etiquetado), Ultralytics YOLOv8, PyTorch, CUDA/cuDNN y GPU adecuada.
Supuestos	Dataset representativo y recursos de cómputo suficientes para iterar y converger.
Restricciones	Consumo de VRAM/tiempo de entrenamiento; riesgo de over/underfitting si el dataset es limitado.

Tabla 8: *Módulo de Entrenamiento (ME)*

6.3.4. Módulo de Profundidad (MP)

El **MP** emplea *Depth Anything V2* para estimar profundidad monocular cuadro a cuadro y persistirla como mapas `.npy` sincronizados con el video, habilitando la medición relativa mano–cabeza.

Propósito	Aportar información espacial relativa (profundidad por píxel) para el cálculo de distancia/velocidad en impactos.
Alcance	Descarga de <i>checkpoints</i> ; inferencia por frame; guardado de <code>depth_{id}.npy</code> y verificación de sincronía.
Dependencias	Videos de entrada, repositorio/pesos de Depth Anything V2 (HuggingFace), PyTorch y hardware compatible.
Supuestos	Los mapas de profundidad reflejan relaciones espaciales consistentes aunque no métricas absolutas.
Restricciones	Alto costo de cómputo y almacenamiento (un <code>.npy</code> por frame); dependencia de la calidad del video.

Tabla 9: *Módulo de Profundidad (MP)*

6.3.5. Módulo de Detección y Análisis (MDA)

El **MDA** es el núcleo operativo: ejecuta la detección con YOLOv8, asocia mano–golpe–cabeza, muestrea y normaliza profundidad en ROIs, calcula $\Delta d/\Delta t$ y clasifica la intensidad (leve/media/fuerte) aplicando filtros temporales y de velocidad.

Propósito	Identificar automáticamente golpes dirigidos a la cabeza y cuantificar su intensidad con criterios objetivos.
Alcance	Inferencia por frame (mano/cabeza/golpe); asociación espacio-temporal; muestreo de profundidad (mediana 5×5); normalización por percentiles; cálculo de velocidad relativa; clasificación de intensidad; filtros de calidad.
Dependencias	ME (best.pt); MP (.npy sincronizados); OpenCV/-NumPy; FPS del video.
Supuestos	Detecciones con confianza suficiente; sincronía video-profundidad estable; FPS constante.
Restricciones	Sensibilidad a oclusiones, ángulos e iluminación; limitaciones de tiempo real por hardware; precisión condicionada al dataset y a la calidad de profundidad.

Tabla 10: Módulo de Detección y Análisis (MDA)

6.3.6. Módulo de Video Anotado (MVA)

El **MVA** genera un MP4 con superposiciones visuales (cuadros delimitadores, etiquetas, contador de eventos e intensidad) a partir de las salidas del MDA.

Propósito	Proveer evidencia visual clara para revisión clínica o de investigación.
Alcance	Render de anotaciones por frame; composición de video; codificación/compresión en MP4.
Dependencias	Salidas del MDA; OpenCV; códecs disponibles en el entorno.
Supuestos	Orden y disponibilidad de frames procesados; tasa de escritura suficiente.
Restricciones	Tamaño de archivo; compatibilidad de códecs/resoluciones; tiempo de renderizado.

Tabla 11: Módulo de Video Anotado (MVA)

6.3.7. Módulo de Reporte (MR)

El **MR** exporta registros estructurados de cada evento (CSV/TXT) y extrae imágenes de evidencia (JPG) para trazabilidad, análisis y auditoría de resultados.

Propósito	Entregar resultados cuantitativos y evidencias reutilizables para análisis posterior.
Alcance	Escritura de CSV/TXT con metadatos (frame, tiempo, Δd , Δt , velocidad, intensidad, causas de descarte); exportación de frames relevantes en JPG; organización por carpetas.
Dependencias	Salidas del MDA; sistema de archivos; permisos de escritura; utilidades estándar de E/S.
Supuestos	Rutas válidas y nomenclatura estable para ubicar/consumir los artefactos.
Restricciones	Espacio en disco y tiempos de E/S; políticas de retención/-backup; posible anonimización si se comparte externamente.

Tabla 12: Módulo de Reporte (MR)

7. Gestión de Riesgos

La presente sección identifica los supuestos bajo los cuales se desarrolló el proyecto, las dependencias técnicas y organizacionales, las restricciones que acotan su alcance y, finalmente, los riesgos principales con sus estrategias de mitigación y seguimiento. El sistema en cuestión integra **YOLOv8s** (detección 2D) con **Depth Anything V2** (profundidad relativa) para estimar eventos de *golpe* y su intensidad en videos.

7.1. Supuestos

1. **Disponibilidad de datos:** se cuenta con videos representativos de los escenarios de uso (entornos clínicos y/o domésticos controlados), con consentimiento informado y anotaciones consistentes para *cabeza*, *mano* y *golpe*.
2. **Calidad de señal:** las grabaciones mantienen una tasa mínima de ≥ 25 fps, resolución $\geq 720p$, iluminación suficiente y campo visual estable.
3. **Sujeto en cuadro:** en la mayor parte de los cuadros el sujeto principal permanece visible de cintura hacia arriba, sin oclusiones totales prolongadas.
4. **Escena moderadamente estática:** cambios de iluminación/exposición entre cuadros no invalidan la comparabilidad de la profundidad relativa tras la normalización global.
5. **Acceso a GPU:** para entrenamiento e inferencia en tiempo razonable se dispone de GPU con soporte CUDA/cuDNN compatible.
6. **Uso clínico complementario:** el sistema es una herramienta de apoyo y no reemplaza la evaluación profesional.

7.2. Dependencias

1. **Software y librerías:** Python (≥ 3.10), Ultralytics/YOLOv8, PyTorch (CUDA/-cuDNN), OpenCV, NumPy, Depth Anything V2 (pesos y código del repositorio), controladores NVIDIA.
2. **Plataformas de datos:** repositorio del dataset (p. ej., Roboflow u otro almacenamiento), control de versiones (Git), sistema de archivos con espacio suficiente para videos y NPY.
3. **Infraestructura:** equipo con GPU (memoria $\geq 8-12$ GB recomendada), disco de alta velocidad para lectura/escritura de cuadros y mapas de profundidad.
4. **Procesos y personas:** disponibilidad de personal para anotación/validación de datos, comité ético o equivalente para resguardo de datos sensibles, responsables de TI para instalación y mantenimiento.

7.3. Restricciones

1. **Profundidad relativa:** los mapas de *Depth Anything V2* no son métricos absolutos; requieren normalización y anclaje de escena para comparaciones temporales.
2. **Variabilidad de dominio:** cambios extremos de iluminación, poses inusuales, accesorios (gorros, lentes brillantes) u oclusiones pueden degradar la detección y la estimación de profundidad.
3. **Privacidad y uso de datos:** almacenamiento y tratamiento de videos deben cumplir la normativa vigente (consentimiento, anonimización, retención mínima necesaria).
4. **Cómputo:** la inferencia conjunta detección+profundidad puede estar limitada por GPU y E/S de disco; en equipos sin GPU la latencia aumenta.

5. **Tiempo y presupuesto:** número de iteraciones de mejora (re-anotación/entrenamiento/ablation) acotado por calendario y recursos.
6. **Estándares clínicos:** el sistema no realiza diagnóstico; su salida debe integrarse a flujos institucionales sin interferir con protocolos asistenciales.

7.4. Riesgos

- **Deriva de datos (*data drift*) en nuevos entornos.** El rendimiento puede disminuir cuando el dominio real difiere del dataset de entrenamiento (nuevos fondos/luces/sujetos).
 - **Medida de mitigación:** ampliar diversidad del dataset, validar por dominio, programar reentrenos periódicos y monitorear métricas por escena.
- **Inconsistencias de profundidad relativa entre cuadros.** La normalización por cuadro puede introducir variaciones no deseadas en secuencias cambiantes.
 - **Medida de mitigación:** usar escala global por percentiles, mediana temporal/espacial, umbral mínimo de Δt y acotar $|\Delta d|$.
- **Falsos positivos de golpe (fatiga de alarmas).** Señales espurias incrementan la carga de revisión.
 - **Medida de mitigación:** ajustar conf/iou con curvas P-R/F1, exigir confirmación en ventana de cuadros y excluir zonas problemáticas (reflejos, objetos recurrentes).
- **Falsos negativos (eventos omitidos).** Golpes reales no detectados reducen utilidad clínica.
 - **Medida de mitigación:** balancear el umbral de conf , enriquecer con *augmentation* (blur, motion, iluminación), incluir casos límite en entrenamiento y monitorear *recall* por clase.

- **Desincronización video–NPY.** Cuadros faltantes o reenumerados impiden la fusión detección–profundidad.
 - **Medida de mitigación:** convención estricta de nombres/índices, tolerancia $\pm K$ cuadros con registro, verificación previa de integridad y bitácora de emparejamiento.
- **Limitaciones de hardware.** GPU insuficiente o incompatibilidades de drivers afectan latencia y estabilidad.
 - **Medida de mitigación:** especificar HW mínimo, realizar pruebas de desempeño, definir modo offline en CPU y documentar instalación de CUDA/cuDNN.
- **Privacidad y aspectos legales.** Manejo inadecuado de videos puede vulnerar normativa.
 - **Medida de mitigación:** consentimiento informado, anonimización/enmascaramiento, control de accesos, retención mínima y cumplimiento regulatorio.
- **Dependencia de repositorios externos.** Cambios en APIs o pesos pueden romper compatibilidad.
 - **Medida de mitigación:** versionar por tag/commit, *vendorizar* artefactos críticos, mantener caché local de pesos y pruebas de regresión.
- **Umbral de intensidad no calibrados clínicamente.** Las bandas de v_t (m/s) podrían no corresponder a criterios del centro de salud.
 - **Medida de mitigación:** revisión con especialistas, ajuste por percentiles de casos reales y parametrización por institución.
- **Rendimiento en tiempo real insuficiente.** FPS por debajo del objetivo limita uso continuo.

- **Medida de mitigación:** perfilado por etapa, ajuste de `imgsz` y arquitectura (n/s), exportación a ONNX/TensorRT y E/S asincrónica.
- **Sesgos del dataset.** Distribuciones desbalanceadas (edad, tono de piel, vestimenta, contexto) afectan generalización.
 - **Medida de mitigación:** auditoría de distribución, muestreo estratificado, métricas por subgrupo y recolección adicional para balance.
- **Ambigüedad en la interpretación clínica.** La lectura de resultados puede requerir contexto adicional.
 - **Medida de mitigación:** archivo `resultados.txt` claro (m/s e intensidad) con referencia por frame al video anotado y guía breve para profesionales.

8. Desarrollo

Este informe aborda el desarrollo de un sistema automatizado para la detección y evaluación de impactos cefálicos en niños con TEA a partir de video. El enfoque principal se centra en los siguientes aspectos: adquisición y curación de datos de video en entornos clínicos controlados; anotación y preparación de un conjunto de datos con clases *mano*, *cabeza* y *golpe*; integración de un detector en tiempo real basado en **YOLOv8**; estimación de profundidad monocular mediante **Depth Anything V2** para obtener mapas de distancia por cuadro; asociación temporal de detecciones para calcular la velocidad relativa mano–cabeza en torno al evento de impacto; clasificación de la intensidad del golpe (*leve*, *media*, *fuerte*) en función de umbrales de velocidad; implementación de filtros de validez (p. ej., descarte de eventos con velocidad nula o Δt ínfimo) y suavizado estadístico local de profundidad (mediana en ventanas 5×5); generación de salidas en formato audiovisual (video anotado) y registros estructurados (logs de eventos); y, finalmente, evaluación del rendimiento del sistema en términos de precisión de detección, robustez temporal y requisitos computacionales para su operación en tiempo real.

8.1. Introducción a la arquitectura YOLOv8 + Depth Anything V2

La arquitectura propuesta combina detección de objetos y estimación de profundidad para cuantificar de manera objetiva los eventos de impacto. **YOLOv8** actúa como núcleo de percepción 2D: procesa cada cuadro del video y produce cajas delimitadoras y puntuaciones de confianza para las clases *mano*, *cabeza* y *golpe*, priorizando baja latencia y buena precisión en escenas dinámicas. En paralelo, **Depth Anything V2** estima un mapa de profundidad relativo por cuadro, entregando, para cada píxel, una medida de distancia compatible con la escena observada.

La fusión de ambas salidas se realiza proyectando las detecciones sobre el mapa de profundidad y extrayendo estadísticas robustas (p. ej., mediana) de las regiones de

interés, lo que permite estimar la distancia relativa entre la mano y la cabeza. A partir de la variación de esa distancia a lo largo del tiempo ($\Delta d/\Delta t$), se calcula la *velocidad relativa* asociada al evento. Estos valores se filtran con reglas de validez (para mitigar ruido, falsos positivos o discontinuidades) y se clasifican en niveles de intensidad predefinidos. Este enfoque habilita no solo la confirmación visual del golpe, sino también una cuantificación física reproducible, facilitando el seguimiento objetivo de conductas autolesivas y su análisis longitudinal.

8.2. Proceso de entrenamiento con Roboflow y YOLOv8

Datos y etiquetado

El proceso de entrenamiento se basó en un conjunto de **1 046 imágenes** obtenidas a partir de **videos** capturados en entornos controlados (planos medios y variación moderada de iluminación y pose). A partir de esos videos se extrajeron cuadros (*frames*) y se **etiquetaron manualmente** con tres clases de interés: mano, cabeza y golpe. El **split** del conjunto fue **85 %/10 %/5 %** para **train/val/test** (890/104/52 imágenes), evitando mezclar cuadros de un mismo video entre diferentes particiones para reducir *leakage* temporal.

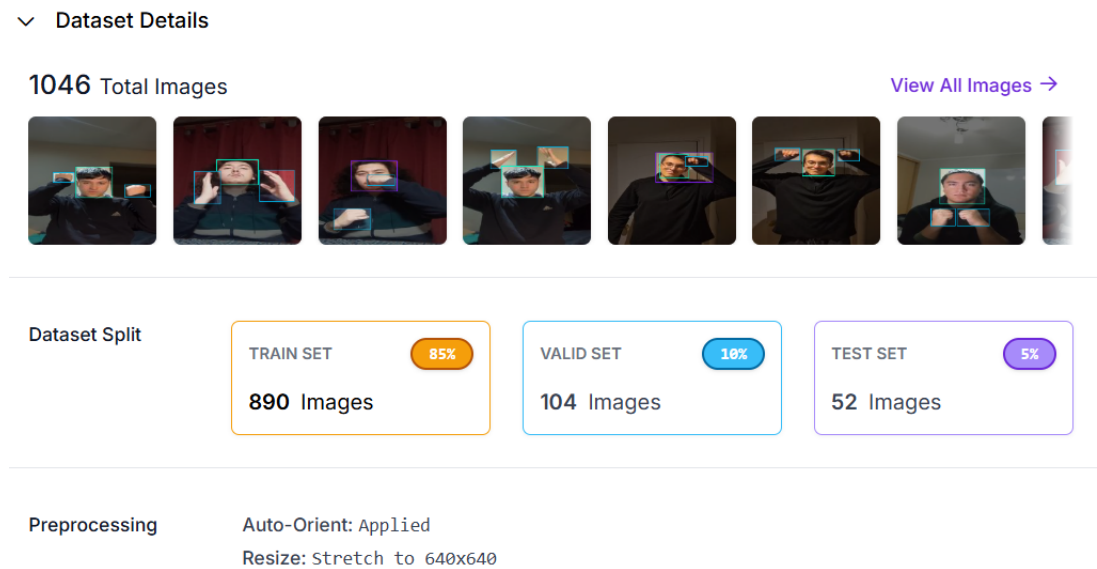


Figura 8.1: *Imagenes train/val/test*

Hiperparámetros y comando de entrenamiento

Se utilizó **YOLOv8s** pre-entrenado como punto de partida, en el cual en conjunto de google colab haremos un fine-tuning, con el siguiente hiperparámetros principal:

- **Épocas:** 150

Comando (Ultralytics):

```
yolo task=detect mode=train model=yolov8s.pt \
  data={dataset.location}/data.yaml epochs=150 imgsz=800 plots=True
```

Métricas generadas

Durante el entrenamiento se produjeron: curvas *Precision–Recall*, *F1–Confidence*, *Precision–Confidence*, *Recall–Confidence*, evolución de pérdidas y métricas , así como la *matriz de confusión* (absoluta y normalizada) y el `checkpoint best.pt`.

8.2.1. Análisis de matrices de confusión

Las Fig. 8.2 y 8.3 resumen el comportamiento del detector en validación mediante dos visualizaciones complementarias: (i) *matriz de confusión absoluta* (conteos) y (ii) *matriz de confusión normalizada por columna* (proporciones). En ambas, las **columnas** representan la **clase verdadera** (*True*) y las **filas** la **clase predicha** (*Predicted*). La diagonal principal recoge los **aciertos** (*true positives*, *TP*); las celdas fuera de la diagonal corresponden a *errores*: *false negatives* (FN, cuando una columna se reparte en otras filas) y *false positives* (FP, cuando una fila acumula conteos de otras columnas).

Matriz absoluta. Los valores reportados por clase son:

- **cabeza (True):** Pred.=cabeza = 70 (TP); Pred.=background = 2 (FN).
- **golpe (True):** Pred.=golpe = 54 (TP).
- **mano (True):** Pred.=mano = 204 (TP).
- **background (True):** Pred.=cabeza = 3, Pred.=golpe = 3, Pred.=mano = 6 (FP para cada clase).

La masa principal se concentra en la diagonal, con algunos FP sobre *background*, mayormente asignados a *mano*.

Matriz normalizada (proporciones por clase verdadera). La normalización por columna permite leer directamente el *recall* de cada clase como el valor de la diagonal:

- **cabeza:** diagonal $\approx 0,97$, es decir, ~ 97 % de las cabezas verdaderas se predijeron como cabeza; el ~ 3 % restante se asignó a *background*.
- **golpe:** diagonal = 1,00 (todas las instancias verdaderas se predijeron como golpe).
- **mano:** diagonal = 1,00 (todas las instancias verdaderas se predijeron como mano).

- **background:** la columna se reparte aprox. en cabeza $\approx 0,25$, golpe $\approx 0,25$, mano $\approx 0,50$.

Estas proporciones muestran que, dado **background** verdadero, el modelo emitió con mayor frecuencia predicciones de mano, seguido de cabeza y golpe.

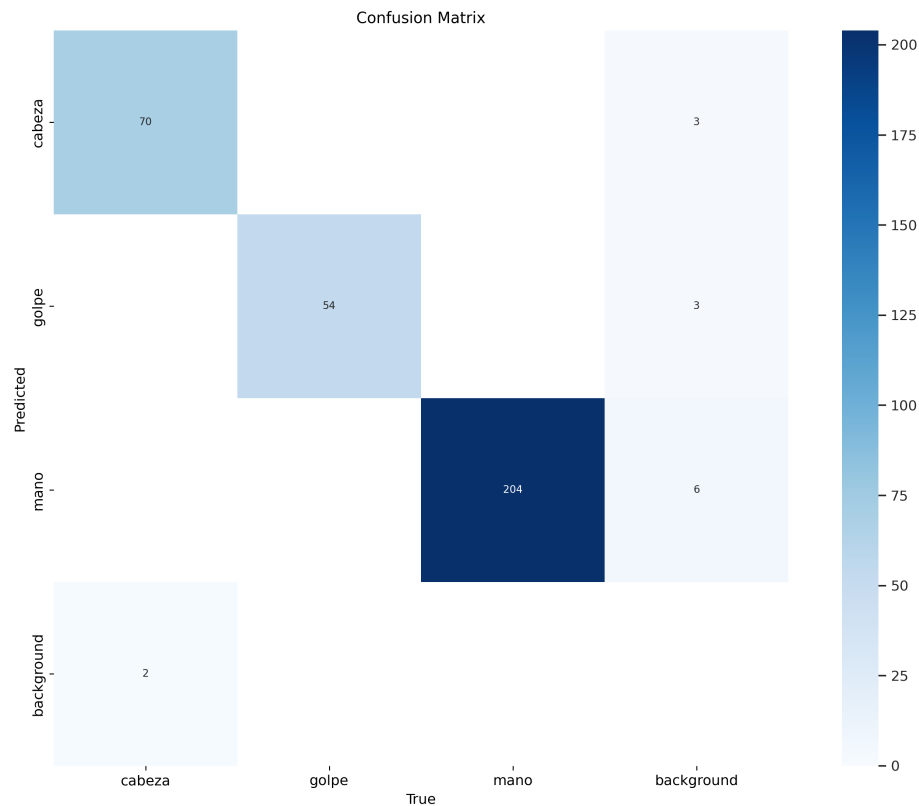


Figura 8.2: Matriz de confusión *absoluta* en validación. Columnas: clase verdadera; filas: clase predicha. La diagonal corresponde a aciertos (TP); las celdas fuera de la diagonal reflejan errores (FN/FP) según su orientación.

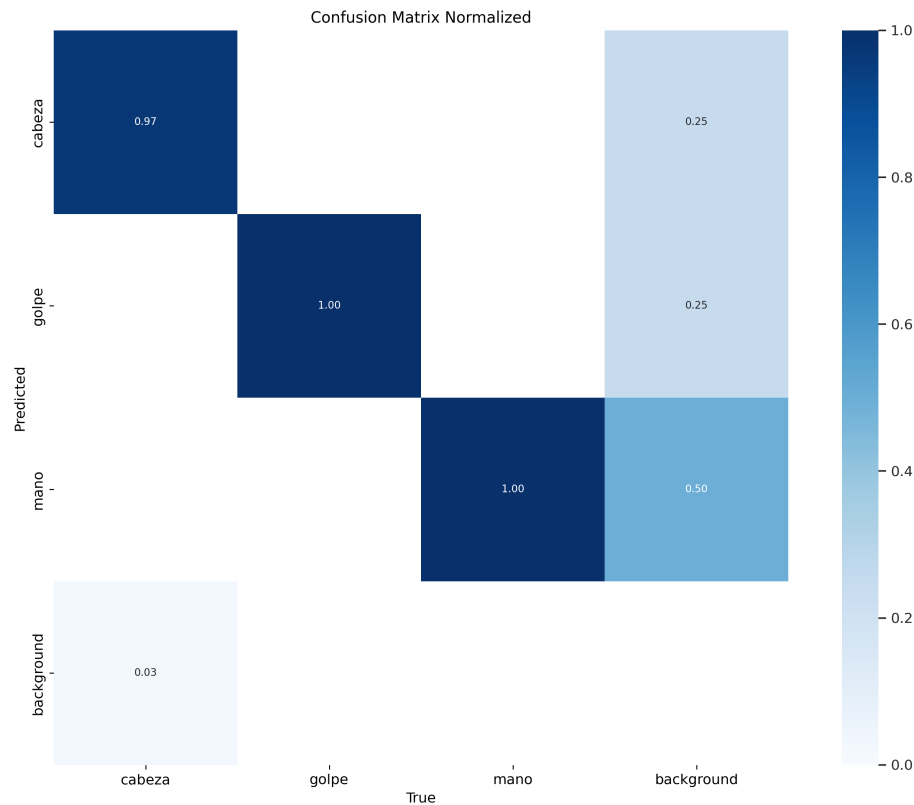


Figura 8.3: Matriz de confusión *normalizada por columna* en validación. La normalización permite interpretar el recall de cada clase como el valor de la diagonal en su respectiva columna.

8.2.2. Curvas F1–Confianza y Recall–Confianza

La relación entre el umbral de confianza (conf) y el desempeño del detector se analiza mediante dos figuras independientes: (i) *F1–Confianza*, que resume el compromiso entre precisión y recall para cada umbral; y (ii) *Recall–Confianza*, que muestra la fracción de instancias verdaderas recuperadas a medida que aumenta conf.

F1–Confianza.

- **Ejes:** F1 (vertical) vs. conf (horizontal), macro–promedio y por clase.
- **Óptimo macro:** máximo aproximado de **0,98** en conf $\approx$ **0,368**.

- **Comportamiento:** F1 crece rápidamente en umbrales bajos, se mantiene en meta alta y cae de forma pronunciada a partir de $\text{conf} \gtrsim 0,8$.

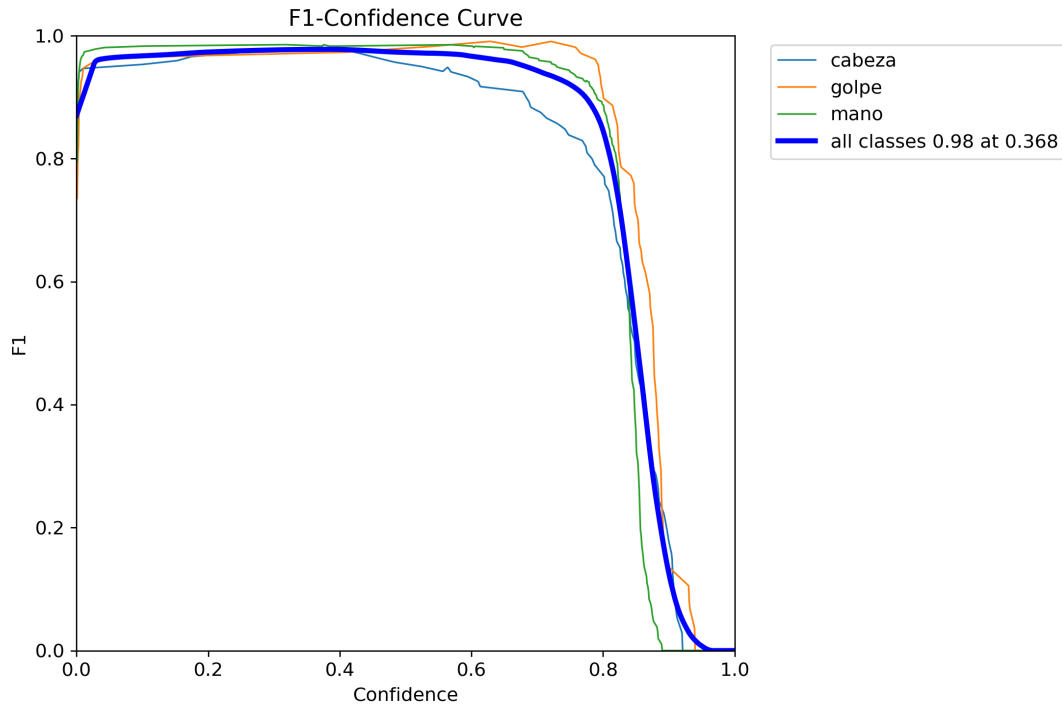


Figura 8.4: Curva F1–Confianza en validación. La línea agregada (azul grueso) alcanza $\approx 0,98$ en $\text{conf} \approx 0,368$; las líneas finas corresponden a cada clase (cabeza, golpe, mano).

Recall–Confianza.

- **Ejes:** recall (vertical) vs. conf (horizontal), macro–promedio y por clase.
- **Extremos:** para $\text{conf} \approx 0$, el recall macro $\approx 1,00$; decrece de manera monótona al aumentar conf .
- **Zona de caída:** descenso marcado en $[0,8, 0,9]$, donde se descartan predicciones menos confiables.

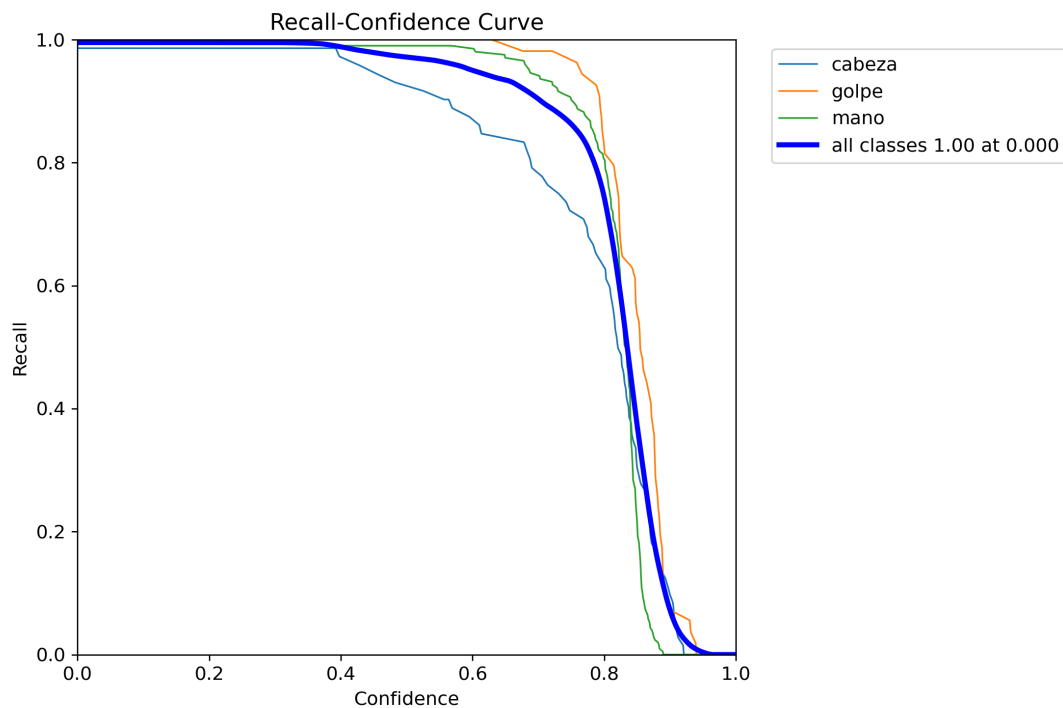


Figura 8.5: Curva Recall–Confianza en validación. El recall macro parte cercano a 1,00 y decrece al aumentar conf ; se observa una caída pronunciada a partir de $\text{conf} \in [0,8, 0,9]$.

8.2.3. Curvas Precision–Confianza y Precision–Recall

A continuación se presentan de forma separada las visualizaciones que caracterizan (i) la evolución de la *precisión* en función del umbral de confianza (conf) y (ii) la relación *Precisión–Recall* agregada y por clase sobre el conjunto de validación.

Precision–Confianza.

- **Ejes:** precisión (vertical) vs. conf (horizontal), con curvas por clase y macro.
- **Punto anotado (macro):** la leyenda indica **1,00** de precisión global en $\text{conf} \approx 0,879$.
- **Comportamiento:** la precisión crece de forma monótona con el umbral; las tres clases (cabeza, golpe, mano) se aproximan a 1,0 para conf altos.

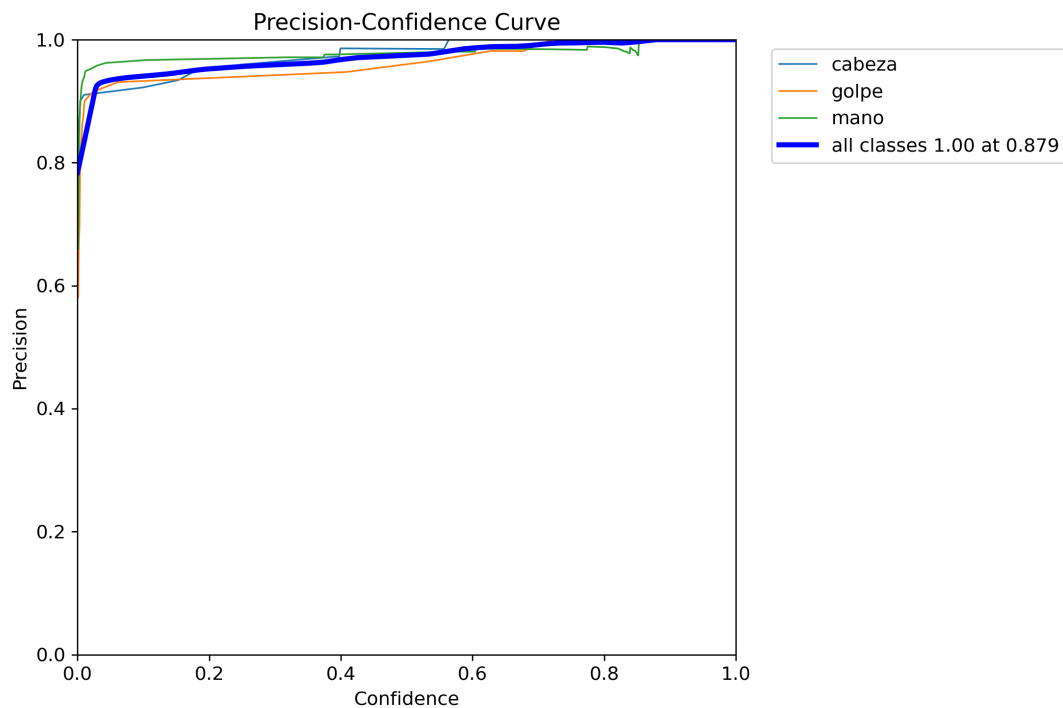


*Figura 8.6: Curva **Precision–Confianza** en validación. La curva agregada alcanza precisión $\approx 1,00$ en $conf \approx 0,879$; las líneas individuales muestran el mismo patrón creciente por clase.*

Precision–Recall (PR).

- **Ejes:** precisión (vertical) vs. recall (horizontal), por clase y macro.
- **AP@0.5 por clase (según leyenda):** cabeza = **0,989**, golpe = **0,995**, mano = **0,986**.
- **mAP@0.5 (macro):** **0,990**.
- **Forma de las curvas:** trayectorias cercanas al techo (alta precisión) a lo largo de la mayor parte del rango de recall.

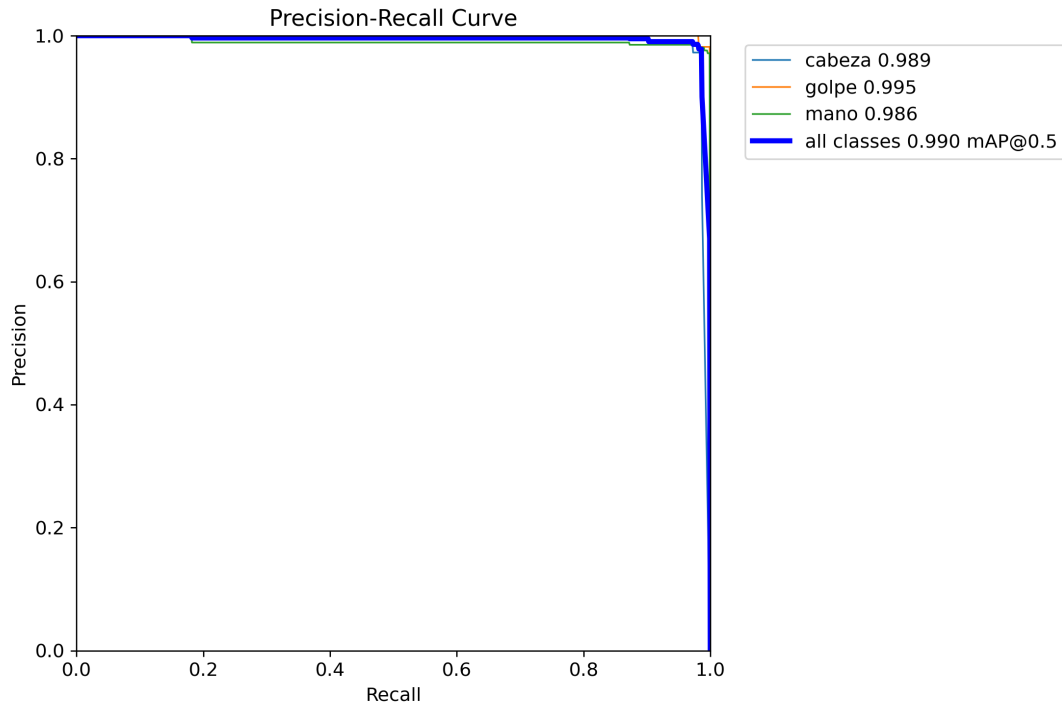


Figura 8.7: Curva *Precision-Recall* en validación. Se reportan $AP@0.5$ por clase (cabeza 0.989, golpe 0.995, mano 0.986) y $mAP@0.5 = 0.990$ (macro).

8.3. Detección con YOLOv8s

La inferencia cuadro a cuadro se realiza con **YOLOv8s** entrenado en mano, cabeza y golpe. Para cada cuadro I_t el modelo entrega hipótesis (clase, caja, confianza) sobre las que se aplica *Non-Maximum Suppression* ($iou_nms = 0,50$) y umbral de confianza ($conf = 0,25$). Las detecciones válidas se superponen como cajas rotuladas, y en la esquina superior izquierda se muestra el **conteo instantáneo Detecciones: k** . De forma independiente, el sistema mantiene un **contador acumulado de eventos golpe** (*Golpes: N*), que sólo aumenta cuando la hipótesis de golpe es consistente temporalmente y coherente con mano y cabeza (permitiendo luego el cálculo de velocidad).



(a) Cuadro inicial con Detecciones: 3. Se observan cabeza (0.83) y dos mano (0.74 y 0.33). Aún no aparece la hipótesis golpe.

(b) Cuadro siguiente con Detecciones: 4. Emerge la hipótesis golpe (0.66) sobre la región de la frente, además de cabeza y mano.

Figura 8.8: Evolución temporal de las detecciones con *YOLOv8s*. El aumento de k de 3 a 4 corresponde a la aparición de golpe. El contador acumulado Golpes: N sólo se incrementa si la hipótesis persiste en una ventana temporal y se valida su coherencia inter-clase, evitando inflar eventos por detecciones efímeras.

8.4. Estimación de profundidad monocular con *Depth Anything* (mapas relativos)

Esta etapa procesa un **video RGB** y produce, para cada cuadro, un **mapa de profundidad relativa** que se guarda en archivos `.npy`. Dichos mapas se consumen posteriormente junto con las detecciones 2D (*YOLOv8*) para extraer distancias/variaciones temporales por región de interés (ROI). A continuación se describe exclusivamente este componente.

Modelo y naturaleza de la predicción

Depth Anything V2 es un modelo de *monocular depth* que, dado un cuadro $I_t \in \mathbb{R}^{H \times W \times 3}$, infiere un mapa $D_t \in \mathbb{R}^{H \times W}$. Los valores son **relativos** por cuadro: preservan el orden y las relaciones cercano/lejos dentro de la misma imagen, pero *no* constituyen una escala métrica absoluta. Para visualización y cálculo robusto en ROI, se normaliza por percentiles a $[0, 1]$:

$$\tilde{D}_t = \text{clip}\left(\frac{D_t - P_{2.5}(D_t)}{P_{97.5}(D_t) - P_{2.5}(D_t)}, 0, 1\right), \quad \hat{D}_t = \text{mediana}_{5 \times 5}(\tilde{D}_t).$$

El repositorio proporciona **cuatro escalas** del modelo para profundidad *relativa*: Small (vits), Base (vitb), Large (vitl) y Giant (vitg);

Se cargan con `DepthAnythingV2(encoder={vits, vitb, vitl, vitg})` y `.pth` correspondientes.¹

Flujo de procesamiento para video

Dado un video $V = \{I_t\}_{t=1}^T$:

1. **Decodificación.** Se extraen los cuadros I_t manteniendo $H \times W$ originales cuando se prevé fusionar con cajas 2D.
2. **Inferencia.** Para cada cuadro, el modelo genera D_t (mapa crudo). El repositorio incluye `run_video.py` con:

```
python run_video.py \
  --encoder <vits|vitb|vitl|vitg> \
  --video-path <path> --outdir <outdir> \
  [--input-size <size>] [--pred-only] [--grayscale]
```

¹Ver `depth_anything_v2/dpt.py` y las instrucciones de `README.md`. El tamaño de entrada por defecto para inferencia es `-input-size 518`, ajustable para más detalle. Asimismo, V2 emplea *intermediate features* de DINOv2 para el decodificador.

3. **Normalización y suavizado.** Se aplica normalización por percentiles y filtro de mediana 5×5 , obteniendo $\hat{D}_t \in [0, 1]$.
4. **Persistencia de artefactos.** Por eficiencia de carga y trazabilidad, se guarda un `.npy` por cuadro (o un `stack (T, H, W)`), además de una versión coloreada `.png` para inspección.

Estructura de salida

```
video_original.mp4
depth_maps/
    frame_0000001.npy
    frame_0000002.npy
    ...
```

Cada `.npy` contiene una matriz `float32` ($H \times W$) con **profundidad relativa** del cuadro homólogo del video original.

8.4.1. Ejemplo visual

La Fig. 8.9 muestra un cuadro de entrada y su mapa de profundidad relativa. Los tonos (colormap) codifican valores normalizados en $[0, 1]$, útiles para identificar regiones cercanas/lejos dentro del mismo cuadro.



Figura 8.9: Izquierda: cuadro RGB del video. Derecha: mapa de *profundidad relativa* estimado con Depth Anything V2 y normalizado para análisis y visualización.

8.4.2. Combinación: YOLOv8l + Depth Anything V2

La combinación se concibe como dos ramas paralelas por *frame* del video: (i) **detección 2D** con **YOLOv8l** para obtener cajas de mano y cabeza; (ii) **profundidad monocular relativa** con **Depth Anything V2** para el mismo cuadro. El objetivo conceptual es asociar a cada detección una medida de *profundidad relativa* extraída del mapa, de modo que se pueda seguir la *separación relativa mano–cabeza* a lo largo del tiempo y, con ello, estimar su *variación temporal*.

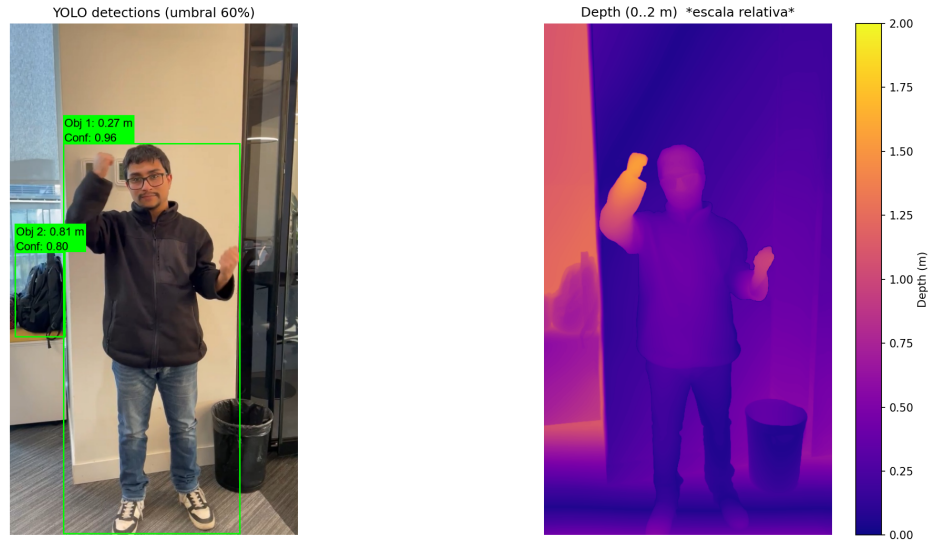


Figura 8.10: Vista conceptual de la fusión: a la izquierda, detecciones 2D con YOLOv8l; a la derecha, el mapa de **profundidad relativa** de Depth Anything V2. La idea es asociar a cada caja (p. ej., mano, cabeza) un valor de profundidad relativa obtenido del mapa del mismo cuadro.

Input del Programa.

1. El **video original** donde YOLOv8 entrega las cajas y sus confianzas.
2. Un **mapa de profundidad relativa** en .npz generado por Depth Anything V2.

La profundidad es **relativa por cuadro** ; por tanto, sirve para comparar “más cerca/lejos” entre regiones del mismo frame y entre frames usando la misma política de normalización.

Aplicación pensada a la secuencia (video + NPZ). Con el video y la serie de .npz por cuadro, la idea es construir una señal temporal de *separación relativa mano–cabeza* y su *variación*:

$$\Delta_t = d_t^{\text{cabeza}} - d_t^{\text{mano}}, \quad v_t = \frac{\Delta_{t-1} - \Delta_t}{\Delta t},$$

donde $d_t^{(\cdot)}$ proviene del mapa de profundidad relativa del mismo cuadro, y Δt es el intervalo entre frames. Esta formulación usa exclusivamente **YOLOv8l + Depth Anything V2 (NPZ relativos)**.

8.5. Cálculo de la velocidad relativa mano–cabeza

El cálculo se basa en (i) las detecciones 2D de mano y cabeza de YOLOv8s por cuadro, y (ii) el mapa de **profundidad relativa** $\hat{D}_t$ (NPY) del mismo cuadro. Los parámetros operativos usados en el módulo son:

AUTO_MAX_REAL_DISTANCE = 2.0 m Límite superior de la escala física utilizada para convertir la profundidad relativa del NPY a metros. Tras fijar una escala global de escena por percentiles ($scene_min$, $scene_max$), cada valor se normaliza a $[0, 1]$ y se proyecta al rango $[0, AUTO_MAX_REAL_DISTANCE]$ mediante $d_m = \text{norm}(d) \times AUTO_MAX_REAL_DISTANCE$. Este parámetro define el techo operativo con el que se computan Δd y, en consecuencia, la velocidad; su elección debe reflejar la profundidad típica de la escena (interacciones cercanas o más profundas).

MAX_PENETRATION_M = 0.35 m Cota física aplicada a la variación efectiva de distancia $|\Delta d|$ empleada en el cálculo de velocidad, de modo que $|\Delta d| \leftarrow \min(|\Delta d|, MAX_PENETRATION_M)$. Con ello se previenen incrementos espurios debidos a ruido u oclusiones en el mapa de profundidad y se asegura que las magnitudes de acercamiento permanezcan en un rango verosímil para el tipo de impacto que se modela.

MAX_VALID_VELOCITY = 10.0 m/s Umbral superior para la velocidad relativa resultante v , aplicado como $v \leftarrow \min(v, MAX_VALID_VELOCITY)$. Actúa como salvaguarda ante outliers causados por Δt muy pequeños o estimaciones erráticas de profundidad, estabilizando la etiqueta de intensidad sin permitir valores físicamente inverosímiles.

WINDOW_MEDIAN = 3 Radio r de la ventana espacial usada para estimar la mediana de profundidad alrededor del centro de la detección; induce un kernel $(2r+1) \times (2r+1)$, por lo que $r = 3$ corresponde a 7×7 . Esta agregación local robustece d_t frente a grano y discontinuidades del mapa, preservando a la vez suficiente

detalle para representar planos locales de mano y cabeza.

MANO_MAX_FRAMES = 55 Horizonte temporal máximo, en número de cuadros, para buscar y asociar la mano previa al evento de golpe actual con fines de cálculo de Δt y Δd . La mano candidata debe, además, estar próxima en imagen (distancia en píxeles) al centro del golpe; este parámetro determina hasta qué punto se permite retroceder en la secuencia para formar el par temporal mano–cabeza que sustenta la estimación de velocidad.

8.5.1. Profundidad por objeto (mano/cabeza).

Para cada cuadro t , y para el centro (c_x, c_y) de mano o cabeza:

1. Se carga el mapa $\hat{D}_t$ (profundidad relativa).
2. Se toma una **mediana espacial** en una vecindad $(2r+1) \times (2r+1)$ con $r = \text{WINDOW_MEDIAN}$.
3. Se aplica **mediana temporal** en $[t-1, t, t+1]$ si hay mapas disponibles.
4. Se fija una **escala global de escena** (P_2, P_{98}) y se convierte a metros en $[0, \text{AUTO_MAX_REAL_DISTANCE}]$ (mapeo lineal).

Así se obtienen d_t^{mano} y d_t^{cabeza} (metros, consistentes dentro de la escena).

8.5.2. Separación relativa por cuadro.

Se define la separación mano–cabeza en el cuadro del evento:

$$\Delta_t = d_t^{\text{mano}} - d_t^{\text{cabeza}}.$$

Si mano aparece claramente *más lejos* que cabeza (tolerancia > 3 cm), el evento se invalida para ese cuadro.

8.5.3. Elección del par temporal y Δt .

Para un golpe confirmado en t , se busca la mano previa más cercana en espacio y dentro de los MANO_MAX_FRAMES últimos cuadros. Si esa mano ocurrió en $t-k$, entonces

$$\Delta t = \frac{k}{\text{fps}}, \quad \Delta t \geq \text{MIN_DELTA_T} (\approx 0,03 \text{ s}).$$

8.5.4. Variación de distancia y velocidad.

Con la mano previa y la cabeza en el cuadro actual:

$$\Delta d = d_{t-k}^{\text{mano}} - d_t^{\text{cabeza}}, \quad |\Delta d| \leftarrow \min(|\Delta d|, \text{MAX_PENETRATION_M}),$$

$$v_t = \frac{|\Delta d|}{\Delta t}, \quad v_t \leftarrow \min(v_t, \text{MAX_VALID_VELOCITY}).$$

Es decir, la velocidad es la **razón entre el acercamiento efectivo** (en metros) y el tiempo transcurrido entre la mano previa y el cuadro del evento.

8.5.5. Etiqueta de intensidad (según v_t en m/s).

La intensidad del golpe se etiqueta por tramos (velocidad v_t en m/s):

$$\text{leve } (v_t < 0,5 \text{ m/s}), \quad \text{media } (0,5 \text{ m/s} \leq v_t < 1,5 \text{ m/s}), \quad \text{fuerte } (v_t \geq 1,5 \text{ m/s}).$$

8.5.6. Entradas y salidas del módulo

Entradas. El módulo recibe: (i) un **video RGB** en formato MP4 (VIDEO_PATH); (ii) una **carpeta de mapas de profundidad** con un archivo .npy por cuadro (DEPTH_MAPS); y (iii) los **pesos del detector** entrenado (best.pt, modelo YOLOv8s). Con estos insumos se infieren detecciones 2D por cuadro y, en paralelo, se consulta la profundidad relativa en las regiones de interés.

Salidas. Se generan: (i) **video anotado** con cajas e identificadores de pista; (ii) **recortes** (*crops*) de cada evento confirmado (`frames_golpe/`); (iii) **registro detallado** (`log_golpes.txt`) con caja, razón de validación y campos de depuración; (iv) **traza técnica** (`debug_deltas.csv`) con profundidades de mano/cabeza, Δd , Δt , v_t ; y (v) un **resumen legible** orientado a personal clínico (`resultados.txt`).

`resultados.txt` es el archivo pensado para lectura rápida por parte de profesionales de salud. Cada línea resume un evento confirmado mediante cuatro campos separados por comas:

- `id_evento`: identificador incremental del episodio.
- v_t (m/s): **velocidad relativa mano–cabeza** al momento del evento, ya acotada por restricciones físicas y temporales.
- **intensidad**: clasificación a partir de v_t — leve ($v_t < 0,5$ m/s), media ($0,5 \leq v_t < 1,5$ m/s), fuerte ($v_t \geq 1,5$ m/s).
- **frame**: número de cuadro del video donde se registró el evento (para verificación visual directa).

Ejemplo:

1,0.732,media,418

2,1.874,fuerte,563

3,0.412,leve,607

Este formato minimalista prioriza claridad y rapidez de interpretación: permite a un médico identificar *cuántos* episodios hubo, *qué tan intensos* fueron (en m/s y en categorías comprensibles) y *dónde* revisarlos en el video. Puede abrirse con cualquier editor de texto o planilla de cálculo y, al compartir el `id_evento` con `log_golpes.txt` y `debug_deltas.csv`, habilita trazabilidad hacia los detalles técnicos cuando sea necesario sin sobrecargar la vista principal orientada a clínica.

9. Conclusión

Esta tesis desarrolló y evaluó un sistema para la **detección automática de golpes** en contextos asociados a TEA, combinando un detector 2D YOLOv8s entrenado en mano, cabeza y golpe, con mapas de **profundidad relativa** generados mediante **Depth Anything V2** para estimar, por evento, la **velocidad relativa mano–cabeza** y derivar una **intensidad** en tramos (leve, media, fuerte). El aporte principal radica en *cerrar el ciclo* entre (i) una percepción visual robusta y eficiente en tiempo real, (ii) un mecanismo de fusión con profundidad que controla ruido y deriva de escala, y (iii) una *salida interpretable y operacional* para el ámbito clínico (`resultados.txt`) con identificadores de evento, velocidad en m/s, intensidad y cuadro correspondiente para verificación inmediata.

Desde el punto de vista metodológico, se estableció una *tubería reproducible* que abarca: curación/partición del dataset (entrenamiento/validación/prueba), entrenamiento de YOLOv8s con `imgsz = 800` durante 150 épocas, validación con curvas P–R/F1, matrices de confusión y análisis por clase, e integración con estimaciones de profundidad normalizadas a una *escala global de escena*. Se incorporaron *salvaguardas* para la cinemática: medianas espacial y temporal de profundidad, umbral mínimo de Δt , asociación física de $|\Delta d|$ y de v , y criterios de confirmación temporal/espacial para elevar una detección instantánea a *evento* validado. Este diseño permite **contener la variabilidad** de la profundidad relativa y atenuar falsos positivos relacionados con oclusiones puntuales, reflejos o jitter de medición.

Los **resultados** de entrenamiento y validación muestran *altos niveles de precisión y recobrado* en la banda operativa de confianza, con F1 macro cercano a la unidad y curvas P–R con mesetas amplias que facilitan la elección de umbrales. El análisis por clase sugiere **buena separabilidad** de golpe y estabilidad de mano/cabeza; las confusiones residuales se concentran en `background` y se reducen sustancialmente con la lógica de confirmación temporal. En términos de **utilidad clínica**, el archivo `resultados.txt` —*line-buffered*, legible y mínimo— permite a un profesional revisar

CAPÍTULO 9 : CONCLUSIÓN

el conteo de episodios, su intensidad y saltar de forma directa al cuadro del video donde ocurrió el evento, creando una trazabilidad simple entre la señal automatizada y la inspección visual humana.

No obstante, el sistema presenta **limitaciones** que contextualizan su alcance. (1) La profundidad es *relativa*; su uso operativo exige normalización global y sigue siendo sensible a cambios bruscos de iluminación o a escenas con fuerte textura especular. (2) Los *umbrales de intensidad* (0.5 y 1.5 m/s) son *heurísticos informados*; resultan razonables desde la cinemática humana, pero requieren **calibración clínica** con datos de uso real. (3) La **sincronía** entre video y NPY es un requisito práctico; errores de numeración o pérdidas de cuadros degradan el emparejamiento. (4) La **generalización** a dominios distintos (sujetos, fondos, cámaras) depende de la amplitud del dataset y del uso de *data augmentation* bien sintonizado. (5) El **desempeño temporal** puede verse afectado en hardware sin GPU, lo que desplaza el sistema a un modo *near-real-time/offline*. Pese a estas restricciones, el prototipo alcanza un **balance pragmático** entre precisión, interpretabilidad y costo de integración, y demuestra la *viabilidad* de cuantificar automáticamente eventos de golpe con una salida útil para profesionales de salud.

En términos generales, la integración de detección 2D y estimación de profundidad relativa permite cuantificar eventos de golpe y ofrecer salidas clínicamente interpretables. El sistema conecta la percepción visual con variables físicas (velocidad en m/s e intensidad categórica) y entrega **productos de salida** para revisión (video anotado, recortes y `resultados.txt`), estableciendo una base técnica sólida para su calibración clínica y despliegue controlado. Para favorecer la reproducibilidad y la adopción en otros contextos, el código fuente, las configuraciones y ejemplos de salida se encuentran disponibles en [3].

9.1. Trabajo Futuro

Calibración clínica y validación externa. Realizar estudios piloto con profesionales para **calibrar umbrales de intensidad** por acuerdo experto, evaluando distintas parejas (τ_1, τ_2) de cortes de v_t en m/s (grid search) y reportando F1 macro y kappa de Cohen. Incluir *análisis de sensibilidad* ($\pm 0,1$ m/s) y validación externa en centros distintos al de desarrollo, con protocolos de consentimiento y anonimización.

Ampliación de dominio y robustez. Recolectar datos en **entornos heterogéneos** (clínica, hogar, aula) con variación de cámaras, iluminación y vestimenta. Incorporar *aprendizaje activo* para priorizar la anotación de casos difíciles y *evaluación estratificada* por subgrupos (edad, tono de piel, contexto), identificando y mitigando sesgos. Explorar *ensembles* ligeros o *test-time augmentation* para estabilizar inferencia en condiciones adversas.

Profundidad y consistencia temporal. Investigar **fusión multimodal** cuando sea posible (estéreo o ToF) y métodos de *depth scale recovery* para reducir deriva entre cuadros. Evaluar filtros temporales avanzados (Kalman/Bayesian smoothing) o pérdidas de consistencia temporal auto-supervisadas para *suavizar* el campo de profundidad relativo y mejorar la estimación de Δd y v_t .

Modelado secuencial y anticipación de eventos. Integrar **modelos temporales** (Transformers/LSTM/TCN) que consuman trayectorias de mano/cabeza y señales de profundidad para *predecir* la probabilidad de impacto antes del contacto y reducir falsos positivos persistentes. Evaluar esquemas de *early warning* con ventanas de 0.5–1.0 s previas al golpe.

Optimización para despliegue. Optimizar el pipeline para **tiempo real** mediante exportación a ONNX/TensorRT, cuantización y *pipeline paralelizado* (lectura de NPY

CAPÍTULO 9 : CONCLUSIÓN

asincrónica, prefetching). Diseñar perfiles de *calidad-latencia* (p. ej., n/s/m de YOLO, imgsiz adaptativo) y pruebas de esfuerzo en GPUs de gama media y CPU de escritorio.

Interfaces y flujo clínico. Desarrollar un **tablero** que consuma `resultados.txt` como fuente canónica: línea de tiempo de eventos con enlace directo al cuadro del video, métricas por sesión/paciente (frecuencia e intensidad), filtros por rango temporal e informes descargables. Definir **políticas de alerta** (umbrales por paciente) y un *workflow* de revisión humana (*human-in-the-loop*) con registro de correcciones para retroalimentar el modelo.

Evaluación longitudinal y métricas de impacto. Diseñar estudios **longitudinales** que midan tendencias de frecuencia e intensidad a lo largo de semanas/meses, y evaluar *impacto clínico* (p. ej., reducción de episodios o de lesiones, tiempo de respuesta de cuidadores). Incorporar *análisis de supervivencia* para tiempo hasta el siguiente evento y métricas por intervención.

Incertidumbre y explicabilidad. Añadir **estimación de incertidumbre** (p. ej., *MC dropout*, ensambles) para priorizar revisión de casos ambiguos. Incluir *explicaciones visuales* (mapas de atención/atribución) que faciliten auditorías técnicas y clínicas sin exponer datos sensibles.

Privacidad, seguridad y gobernanza. Establecer **políticas de privacidad** (enmascaramiento, borrado selectivo, mínima retención) y **gobernanza de modelos** (versionado, trazabilidad de pesos y datos, auditorías periódicas). Documentar acuerdos de uso y planes de *rollback* ante regresiones de desempeño.

En conjunto, estas líneas de trabajo orientan la transición desde un prototipo técnicamente sólido hacia un **sistema clínicamente calibrado, robusto y sostenible**, con métricas interpretables, integración a flujos institucionales y garantías de privacidad y seguridad acordes al contexto sanitario.

Referencias

- [1] Jiezhou Chen, Guankun Wang, Weixiang Liu, Xiaopin Zhong, Yibin Tian, and ZongZe Wu. Perception reinforcement using auxiliary learning feature fusion: A modified yolov8 for head detection, 2023. URL: <https://arxiv.org/abs/2310.09492>, arXiv:2310.09492.
- [2] B. Dwyer, J. Nelson, T. Hansen, et al. Roboflow (version 1.0), 2024. Computer vision. URL: <https://roboflow.com>.
- [3] Sergio Rojas Escanilla. Detección y cálculo de velocidad (yolov8 + depth anything). https://github.com/Sergio1NG/Deteccion-Calculo_de_Velocidad, 2025. Repositorio GitHub.
- [4] Nouridine Herbaz, Hassan El Idrissi, and Abdelmajid Badri. Deep learning empowered hand gesture recognition: using yolo techniques. In *2023 14th International Conference on Intelligent Systems: Theories and Applications (SITA)*, pages 1–7, 2023. URL: <https://ieeexplore.ieee.org/document/10373734>, doi:10.1109/SITA60746.2023.10373734.
- [5] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8, 2023. URL: <https://github.com/ultralytics/ultralytics>.
- [6] Chinyere Okoye, Chisom M. Obialo-Ibeawuchi, O. A. Obajeun, S. Sarwar, Christine Tawfik, M. S. Waleed, A. U. Wasim, I. Mohamoud, A. Y. Afolayan, and R. N. Mbaezue. Early diagnosis of autism spectrum disorder: A review and analysis of the risks and benefits. *Cureus*, 15(8):e43226, 2023. doi:10.7759/cureus.43226.
- [7] Evan Quinn and Niall Corcoran. Comparative analysis of yolov5/v8/v9 for object detection, tracking, and human action recognition in combat sports. In *Proceedings of the 4th International Conference on AI Research*

REFERENCIAS

- (ICAIR), Midlands Midwest, Ireland, 2024. ICAIR. URL: <https://papers.academic-conferences.org/index.php/icaair/article/view/3031>.
- [8] Amit Kumar Singh, Trapti Shrivastava, and Vrijendra Singh. Advanced gesture recognition in autism: Integrating yolov7, video augmentation and videomae for video analysis, 2024. URL: <https://arxiv.org/abs/2410.09339>, arXiv: 2410.09339.
- [9] Casper Steinfeldt-Kristensen, Catherine A. Jones, and Caroline Richards. The prevalence of self-injurious behaviour in autism: A meta-analytic study. *Journal of Autism and Developmental Disorders*, 50(11):3857–3873, 2020. doi: 10.1007/s10803-020-04443-1.
- [10] Peter Washington, Aaron Kline, Onur Cezmi Mutlu, Emilie Leblanc, Cathy Hou, Nate Stockham, Kelley Paskov, Brianna Chrisman, and Dennis P. Wall. Activity recognition with moving cameras and few training examples: Applications for detection of autism-related headbanging, 2021. URL: <https://arxiv.org/abs/2101.03478>, arXiv:2101.03478.
- [11] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv:2406.09414*, 2024.