

UNIVERSIDAD AMAZÓNICA DE PANDO

ÁREA DE CIENCIAS Y TECNOLOGÍA

CARRERA DE INGENIERÍA DE SISTEMAS



TESIS DE GRADO

**“MODELO DE DETECCIÓN DEL USO DE CASCO EN MOTOCICLISTAS
MEDIANTE VISIÓN POR COMPUTADORA EN LA CIUDAD DE
COBIJA”**

TRABAJO DE GRADO PRESENTADO PARA OBTENER EL TÍTULO ACADÉMICO DE
LICENCIADO EN INGENIERÍA DE SISTEMAS

Postulante: Univ. Jorge Rodrigo Ugarte Gutierrez

Tutor: Msc. Ing. Alex Yanahuaya Arce

Cobija - Pando - Bolivia

2025

AGRADECIMIENTOS

A Dios, por ser mi guía espiritual y fortaleza en cada etapa de este arduo pero gratificante camino académico, iluminando mi entendimiento y brindándome la perseverancia necesaria para culminar este proyecto de investigación.

A mi madre, pilar fundamental de mi vida, cuyo amor, sacrificio y apoyo incondicional han sido el motor que impulsó cada paso de mi formación personal y profesional. Su ejemplo de esfuerzo, fortaleza y valores ha guiado mi camino y hecho posible la culminación de este logro académico.

Al Dr. Humberto Fernández Calle, por su orientación, paciencia y valiosas observaciones que fortalecieron este trabajo.

Al Msc. Ing. Alex Yanahuaya Arce, por su guía técnica, apoyo constante y contribución esencial al desarrollo del modelo.

A la Universidad Amazónica de Pando, por la formación académica que me permitió desarrollar las competencias necesarias para realizar esta investigación en beneficio de la comunidad.

A mis compañeros y amigos de Ingeniería de Sistemas, por el compañerismo, los aprendizajes compartidos y el compromiso con el desarrollo tecnológico de nuestra región.

A los docentes de la carrera, por su enseñanza y por inculcar valores éticos y compromiso social en mi formación profesional.

DEDICATORIA:

A mi madre, por su amor incondicional, sacrificio y fortaleza, que han sido mi mayor inspiración y apoyo en cada etapa de este camino.

A mis abuelos, por su ejemplo y sabiduría que han guiado mi vida; a mis tíos, tías y hermanos, por su cariño y constante acompañamiento.

A toda mi familia, por su comprensión, ánimo y fe en mí. Este logro también les pertenece.

INDICE

CAPITULO I. MARCO INTRODUCTORIO	1
1.1 INTRODUCCIÓN.....	2
1.2 ESTADO DEL ARTE	3
1.3 DESCRIPCIÓN DEL PROBLEMA	5
1.4 FORMULACIÓN DEL PROBLEMA	6
1.5 HIPÓTESIS	6
1.6 OBJETIVOS.....	8
1.6.1 Objetivo general.....	8
1.6.2 Objetivos específicos.....	8
1.7 JUSTIFICACIÓN.....	8
1.8 DISEÑO DE LA INVESTIGACIÓN.....	9
1.8.1 Enfoque de investigación.....	9
1.8.2 Tipo y nivel de investigación.....	9
1.8.3 Diseño de investigación.....	10
1.8.4 Métodos de investigación	10
1.8.5 Población y muestra.....	11
1.8.6 Técnicas e instrumentos de recolección de datos	12
1.8.7 Validación del instrumento.....	13
1.8.8 Validación de hipótesis.....	13
1.8.9 Procedimientos de análisis de datos	14
1.9 ALCANCES	16
CAPITULO II. MARCO REFERENCIAL	18
2.1 MARCO TEÓRICO	19
2.1.1 Inteligencia artificial.....	19
2.1.2 Machine learning	20

2.1.3	Aprendizaje supervisado.....	20
2.1.4	Aprendizaje no supervisado.....	21
2.1.5	Aprendizaje por esfuerzo.....	21
2.1.6	Visión por computadora	22
2.1.7	Calidad de imagen	22
2.1.8	Redes neuronales convolucionales	23
2.1.9	Descenso de gradiente y convergencia en entrenamiento	24
2.1.10	Robustez y aprendizaje invariante en redes neuronales profundas ..	25
2.1.11	Sobreajuste.....	25
2.1.12	Equilibrio sesgo varianza en redes neuronales profundas	26
2.1.13	Memorización y artefactos en el entrenamiento de redes profundas	27
2.1.14	Redes de clasificación de imágenes.....	27
2.1.15	Redes de detección de objetos	28
2.1.16	Factores que afectan la complejidad de detección de objetos	28
2.1.17	Modelo de visión por computadora	29
2.1.18	Aprendizaje por transferencia (Transfer Learning)	30
2.1.19	Yolo (you only look once).....	30
2.1.20	Beneficio.....	31
2.1.21	Funcionamiento	31
2.1.22	Algoritmo de Yolov1 lm.....	34
2.1.23	Métricas de rendimiento	35
2.1.24	Umbral de confianza.....	38
2.1.25	Matriz de confusión	39
2.1.26	Precision y Recall	39
2.1.27	Curva de Precision-Recall	40
2.1.28	Precisión media (mAP).....	42

2.1.29	Seguridad vial y uso del casco en motociclistas	42
2.1.30	Lenguaje de Programación: Python.....	43
2.1.31	Entorno de desarrollo: KagGle	43
2.1.32	Entorno de prueba: Google Colab	43
2.1.33	Conjunto de Datos (Dataset).....	43
2.1.34	Modelo no probabilístico de selección de datos	44
2.1.35	Justificación tamaño dataset y validación del modelo	44
2.2	MARCO CONTEXTUAL.....	45
2.2.1	Introducción al Problema de Seguridad Vial en Bolivia	45
2.2.2	Magnitud Global y Regional del Problema	45
2.2.3	Situación Nacional y local: Bolivia en la Seguridad Vial	46
2.2.4	Contexto Específico de la Ciudad de Cobija	46
2.2.5	Limitaciones y Mecanismos de Fiscalización	48
2.2.6	Brecha de Conocimiento y Oportunidad Tecnológica.....	49
2.2.7	Relevancia y Justificación del Presente Trabajo	50
CAPITULO III. RESULTADOS Y ANALISIS		52
3.1	Desarrollo del modelo de visión por computadora.....	53
3.2	Resultados de evaluación del modelo.....	54
3.3	Análisis de resultados	66
3.3.1	Identificación del punto óptimo de entrenamiento	66
3.3.2	Análisis Diferencial por Clase	68
3.3.3	Análisis Estado del Arte	74
3.3.4	Análisis de Errores Críticos	75
3.3.5	Validación Empírica de la Hipótesis	75
CAPITULO IV. CONCLUSIONES		77
4.1	CONCLUSIONES.....	78

CAPITULO V. RECOMENDACIONES.....	81
5.1 RECOMENDACIONES	82
REFERENCIAS.....	84
ANEXOS	90

ÍNDICE DE TABLAS

Tabla I.1: Operacionalización de variables	6
Tabla I.2: Descripción de las etapas y procedimiento de la investigación	15
Tabla III.1: <i>Resumen modelo A</i>	54
Tabla III.2: <i>Resumen modelo B</i>	55
Tabla III.3: <i>Resumen modelo C</i>	56
Tabla III.4: <i>Resumen modelo D</i>	58
Tabla III.5: <i>Resumen modelo E</i>	59
Tabla III.6: <i>Resumen modelo F</i>	60
Tabla III.7: <i>Resumen modelo G</i>	61
Tabla III.8: <i>Resumen modelo H</i>	62
Tabla III.9: <i>Resumen modelo I</i>	64
Tabla III.10: <i>Resumen modelo J</i>	65
Tabla III.11: Comparación con estado del arte.....	75

INDICE DE FIGURAS

Figura II.1: Categorías de IA.....	19
Figura II.2: Arquitectura de CNN.....	23
Figura II.3: Sobreajuste	26
Figura II.4: Ejemplo de calcular coordenadas en una imagen.....	32
Figura II.5 Descripción gráfica de la intersección sobre la unión	33
Figura II.6: Obtención de la puntuación de confianza del objeto.....	33
Figura II.7: Rendimiento de las arquitecturas Yolo.....	34
Figura II.8: <i>Variantes de la Arquitectura YOLOv11</i>	35
Figura II.9: Explicación gráfica de la intersección sobre la unión	36
Figura II.10: Representación de VP, FP, FN, VN	37
Figura II.11: Ejemplo de Matriz de confusión.....	39
Figura II.12: <i>Definición matemática de Precisión</i>	40
Figura II.13: Definición matemática de Recall.....	40
Figura II.14: Curvas de precisión- recall	41
Figura II.15: Ejemplo de una curva perfecta	41
Figura III.1: <i>Matriz de confusión normalizada modelo A</i>	54
Figura III.2: <i>Matriz de confusión normalizada modelo B</i>	56
Figura III.3: <i>Matriz de confusión normalizada modelo C</i>	57
Figura III.4: <i>Matriz de confusión normalizada modelo D</i>	58

Figura III.5: <i>Matriz de confusión normalizada modelo E</i>	59
Figura III.6: <i>Matriz de confusión normalizada modelo F</i>	61
Figura III.7: <i>Matriz de confusión normalizada modelo G</i>	62
Figura III.8: <i>Matriz de confusión normalizada modelo H</i>	63
Figura III.9: <i>Matriz de confusión normalizada modelo I</i>	64
Figura III.10: <i>Matriz de confusión normalizada modelo J</i>	65
Figura III.11: <i>Gráfico de líneas mAP vs Épocas</i>	66
Figura III.12: <i>Gráfico de línea evolución de la clase sin casco</i>	69
Figura III.13: <i>Grafica de línea de la evolución de la clase motociclista</i>	71
Figura III.14: <i>Grafica de línea de evolución de la clase sin casco</i>	73

INDICE DE ANEXOS

Anexo A	Árbol de problemas	91
Anexo B	Árbol de objetivos	92
Anexo C	Instrumentos de recolecciones de datos	93
Anexo D	Rubrica de expertos validadores	96
Anexo E	Recopilación y preparación del conjunto de datos	111
Anexo F	Etiquetado del conjunto de datos locales	114
Anexo G	Configuración experimental y entrenamiento del modelo	117
Anexo H	Implementación para la Evaluación de los Modelos.....	120
Anexo I	Código Fuente de Entrenamiento	123
Anexo J	Código Fuente Test de Rendimiento	125
Anexo K	Código Fuente ejecución básica.....	127

RESUMEN

La presente investigación desarrolló un modelo de visión por computadora basado en la arquitectura YOLOv11m para detectar automáticamente el uso de casco en motociclistas en la ciudad de Cobija, Pando, Bolivia. La problemática abordada surge de la incidencia de accidentes de tránsito involucrando motocicletas (69% de los siniestros en Pando en 2022) y la ausencia de mecanismos tecnológicos para monitorear el cumplimiento de la normativa de uso obligatorio del casco.

La investigación adoptó un enfoque cuantitativo con nivel explicativo y diseño cuasiexperimental, con la variable independiente, modelo de visión por computadora manipulado (número de épocas de entrenamiento: 10 a 100 en incrementos de 10) para evaluar su efecto sobre la variable dependiente (precisión promedio media - mAP@0.5). El modelo se entrenó con un dataset público de 1,592 imágenes (80% entrenamiento, 20% validación) y se evaluó sobre 200 imágenes locales capturadas en condiciones diurnas reales, cubriendo tres clases: "con casco", "sin casco" y "motociclista".

Los resultados demostraron que el modelo óptimo (50 épocas) alcanzó un mAP@0.5 del 97,2%, precisión global del 96,5% y recall global del 94,2%, superando ampliamente el umbral del 90% establecido en la hipótesis. El análisis por clase reveló rendimientos consistentes: "con casco" (96,2% AP), "motociclista" (98,6% AP) y "sin casco" (96,9% AP). Las matrices de confusión evidenciaron que el modelo redujo la confusión crítica entre "con casco" y "sin casco" al 1%, minimizando clasificaciones erróneas. El análisis de curvas de entrenamiento confirmó ausencia de sobreajuste, validando la robustez del modelo. La investigación confirma la viabilidad técnica de aplicar visión por computadora para detección de uso de casco en motociclistas en contextos urbanos reales. El modelo de detección desarrollado y validado proporciona una base sólida para futuras implementaciones operativas que podrían contribuir al fortalecimiento de la seguridad vial, reducción de lesiones craneoencefálicas y generación de datos sistemáticos para informar políticas públicas en Cobija y Pando.

Palabras clave: Visión por computadora, detección de objetos, YOLOv11m, precisión, recall, mAP.

ABSTRACT

This research developed a computer vision model based on the YOLOv11m architecture to automatically detect helmet use by motorcyclists in the city of Cobija, Pando, Bolivia. The problem addressed arises from the high incidence of motorcycle-involved traffic accidents (69% of crashes in Pando in 2022) and the lack of technological mechanisms to monitor compliance with mandatory helmet-use regulations.

A quantitative approach with an explanatory level and a quasi-experimental design was adopted. The independent variable was the computer-vision model as manipulated through the number of training epochs (10 to 100, in increments of 10), and the dependent variable was the mean average precision (mAP@0.5). The model was trained on a public dataset of 1,592 images (80% training, 20% validation) and evaluated on 200 locally captured daytime images, covering three classes: with helmet, without helmet, and motorcyclist.

Results show that the optimal model (50 epochs) achieved $mAP@0.5 = 97.2\%$, overall precision = 96.5%, and overall recall = 94.2%, substantially exceeding the 90% threshold stated in the hypothesis. Per-class APs were consistent: with helmet (96.2% AP), motorcyclist (98.6% AP), and without helmet (96.9% AP). Confusion matrices indicated that the model reduced critical confusion between with helmet and without helmet to approximately 1%, minimizing misclassifications. Training-curve analysis confirmed the absence of overfitting, validating the robustness of the model.

The study confirms the technical viability of applying computer vision for helmet-use detection in motorcyclists within real urban contexts. The developed prototype provides a solid technical foundation for future operational implementations that could contribute to improving road safety, reducing cranioencephalic injuries and fatalities, and generating systematic data to inform public policy in Cobija and the Pando department.

Keywords: computer vision; object detection; YOLOv11m; precision; recall; mAP.

CAPITULO I.

MARCO INTRODUCTORIO

1.1 INTRODUCCIÓN

El presente trabajo se enmarca en el área de la Inteligencia Artificial y la Visión por Computadora, con el objetivo de desarrollar un modelo capaz de detectar y clasificar el uso del casco en motociclistas a partir de imágenes estáticas. La problemática abordada surge de la creciente incidencia de accidentes de tránsito que involucran motocicletas y el incumplimiento sistemático de la normativa sobre uso obligatorio del casco, elemento de protección crítico para prevenir lesiones graves en accidentes de tránsito.

Según la Organización Mundial de la Salud (2024), los motociclistas representan el 30% de las muertes viales globales, y el uso correcto del casco puede reducir el riesgo de muerte en más de seis veces y de traumatismos hasta en 74%. En Bolivia, el año 2024 cerró con 20,942 siniestros de tránsito, siendo las motocicletas protagonistas del 45.8% de estos accidentes. En el departamento de Pando, específicamente, el 69% de los accidentes registrados en 2022 involucraron vehículos automotores menores, evidenciando una particular vulnerabilidad de los motociclistas en este grupo de usuarios viales.

La ciudad de Cobija ha experimentado un incremento notable en el uso de motocicletas como principal medio de transporte, motivado por su bajo costo y facilidad de movilidad. Sin embargo, esta tendencia no ha sido acompañada de mejoras proporcionales en la cultura de seguridad vial ni en mecanismos efectivos de fiscalización del cumplimiento normativo. El marco legal boliviano, establecido en la Ley N° 3988 y el Reglamento del Código Nacional de Tránsito, estipula la obligatoriedad del uso del casco protector para conductores y pasajeros, sancionando el incumplimiento. No obstante, la Dirección Departamental de Tránsito enfrenta limitaciones estructurales para monitorear efectivamente esta normativa: escasez de personal, cobertura espaciotemporal limitada, ausencia de datos sistemáticos, e imposibilidad de fiscalización continua.

El objetivo principal del estudio consiste en desarrollar y validar un modelo que permita identificación del uso del casco en motociclistas, proporcionando información cuantificable que puedan ser aplicados a futuro por instituciones de control de tráfico. Se adoptó un enfoque cuantitativo con tipo de investigación aplicada y nivel explicativo, dirigido a establecer relación causal entre la manipulación de la variable independiente (modelo de visión por computadora,

mediante variación del número de épocas de entrenamiento) y el rendimiento obtenido (variable dependiente: precisión promedio media - mAP@0.5).

El diseño adoptado es cuasiexperimental con múltiples grados de manipulación experimental. La hipótesis que orienta la investigación establece que Mediante el desarrollo de un modelo de visión por computadora basado en la arquitectura YOLOv11m, se logrará detectar el uso de casco en motociclistas con un nivel de precisión media (mAP@0.5) superior al 90%, evaluado sobre imágenes capturadas en condiciones reales en el contexto urbano de Cobija.

La arquitectura tecnológica seleccionada fue YOLOv11m (You Only Look Once, versión 11 medium), por su equilibrio óptimo entre precisión y eficiencia computacional, superando a versiones anteriores y siendo compatible con dispositivos de recursos limitados. El entrenamiento se realizó con un dataset público de 1,592 imágenes anotadas, dividido en 80% entrenamiento (1,274 imágenes) y 20% validación (318 imágenes). La evaluación del modelo se ejecutó sobre un conjunto independiente de 200 imágenes capturadas localmente en la ciudad de Cobija, en condiciones diurnas reales, cubriendo tres clases: "con casco", "sin casco" y "motociclista".

Los beneficiarios potenciales incluyen instituciones de control de tránsito, sistema de salud pública, y población general de Cobija. El documento se estructura en cinco capítulos: Capítulo I presenta el planteamiento del problema, marco metodológico y diseño de investigación; Capítulo II desarrolla los fundamentos teóricos y tecnológicos; Capítulo III documenta el proceso de recopilación de datos, configuración, entrenamiento e implementación del modelo; Capítulo IV presenta el análisis exhaustivo e interpretación de resultados; Capítulos V ofrecen conclusiones y recomendaciones derivadas del estudio, respectivamente.

1.2 ESTADO DEL ARTE

Para fundamentar adecuadamente el desarrollo de la investigación, se realizó una revisión sistemática de investigaciones previas relacionadas con la detección del uso de casco en motociclistas mediante técnicas de visión por computadora y aprendizaje profundo.

JAIME MERCADO REYNA (2023) realizó una investigación titulada, Detección del uso del casco en conductores de motocicleta, utilizando redes neuronales convolucionales. (Universidad Autónoma de Zacatecas). El objeto de estudio es la detección automatizada del uso de casco en motociclistas, mediante el desarrollo de un modelo inteligente basado en visión por

computadora. El objetivo general fue generar un modelo de detección de uso del casco con enfoque preventivo, utilizando imágenes captadas por una cámara montada en la motocicleta, con el fin de verificar en tiempo real si el conductor utiliza el casco durante todo el trayecto. Las conclusiones indican que, el modelo propuesto, basado en redes neuronales convolucionales y técnicas de transferencia de aprendizaje, logró una precisión del 97.24% en la detección del uso de casco, representando un avance importante hacia sistemas preventivos de seguridad vial para motociclistas.

K. N. HANDE, POOJA TAKLIKER, LEKHA BHOYAR, PRANJALI BAWANKAR, PIYUSH NAVGHARE (2023) realizaron un estudio titulado, Diseño y desarrollo de un sistema de detección automática para motociclistas sin casco utilizando aprendizaje automático. El objeto de estudio consiste en el sistema automatizado de detección de motociclistas que no usan casco, mediante algoritmos de procesamiento de imágenes y aprendizaje automático. El objetivo general fue, proponer un sistema que automatice la detección de motociclistas sin casco y extraiga automáticamente la matrícula para emitir multas de tránsito de forma más rápida y eficiente. Las conclusiones indican que, la implementación del sistema propuesto permitirá agilizar el proceso de emisión de multas, mejorar el cumplimiento de las leyes sobre el uso del casco, y contribuiría a una disminución de accidentes y fatalidades en motociclistas, promoviendo así una mayor seguridad vial.

JUAN PABLO QUILINGKING TOMÁS Y BONIFACIO DOMA (2023) realizaron un estudio titulado, Detección de cascos de motocicleta y clasificación de uso en Filipinas mediante el algoritmo YOLOv5. El estudio se centra en el desarrollo de un modelo de clasificación y detección del uso de cascos en motociclistas, utilizando el algoritmo YOLOv5. El objetivo general fue diseñar e implementar un modelo basado en YOLOv5 que permita detectar motociclistas y clasificar el uso adecuado del casco, utilizando imágenes capturadas en la ciudad de Makati, Filipinas, para entrenar y validar el sistema. Las conclusiones indican que, el modelo alcanzó un rendimiento deseable en la detección y clasificación del uso de cascos. Se identificaron valores óptimos de hiperparámetros con el método de validación niñera, recomendando mantener consistencia en las muestras y ajustar gradualmente los hiperparámetros en futuros estudios.

WEI JIA, SHIQUAN XU, ZHEN LIANG, YANG ZHAO, HAI MIN, SHUJIE LI, YE YU (2021) realizaron un estudio titulado, Detección automática en tiempo real del casco de los motociclistas en el tráfico urbano mediante el detector YOLOv5 mejorado. El objeto de estudio

fue la detección automática del uso de cascos por motociclistas mediante técnicas de aprendizaje profundo. El objetivo general fue desarrollar un método eficiente basado en el algoritmo YOLOv5 mejorado, capaz de identificar motocicletas y verificar si los conductores usan casco a partir de imágenes de videovigilancia, para ello, se implementó una versión optimizada de YOLOv5 incorporando mecanismos de atención triplete y NMS suave. Las conclusiones indican que, el método propuesto demostró una alta precisión, alcanzando un mAP del 97.7%, un F1-score del 92.7% y una velocidad de procesamiento de 63 FPS, superando a otros enfoques actuales en términos de efectividad y eficiencia.

1.3 DESCRIPCIÓN DEL PROBLEMA

Los accidentes de tránsito representan una de las principales causas de muerte a nivel mundial, y una proporción significativa de estas fatalidades involucra a conductores de motocicletas. Según la Organización Mundial de la Salud (2024), los motociclistas y otros conductores de vehículos de motor de dos o tres ruedas representan el 30% de los fallecidos. Esta cifra revela una situación crítica en la seguridad vial, especialmente en países en vías de desarrollo, donde el uso adecuado del casco aún no está generalizado. Se comprobó que el uso de un casco de calidad puede reducir el riesgo de muerte en más de seis veces y el riesgo de traumatismos hasta en 74% (OMS, 2024).

En el contexto latinoamericano, las lesiones traumáticas provocadas por accidentes de tránsito han constituido una preocupación sostenida en salud pública. Desde 2004 a 2030, se estimó que las lesiones craneoencefálicas traumáticas (LCT) representaban el 3.4% de los años de vida saludable perdidos, reflejando una tendencia persistente, especialmente en motociclistas, que continuaron siendo uno de los grupos más vulnerables en la región (Montoya et al., 2016).

En Bolivia, el año 2022 cerró con un total de 17,761 hechos y accidentes de tránsito registrados a nivel nacional. De este conjunto, un 16.55% (2,940 casos) correspondió a incidentes protagonizados por la categoría vehículo automotor menor (OBSCD, 2023). En el departamento de Pando para el año 2022, se registraron 547 hechos y accidentes, representando un 3.08% del total nacional, de los cuales 378 correspondieron a la categoría de vehículo automotor menor, lo que equivale al 69.1% del total departamental (OBSCD, 2023). Esta proporción evidenció que, pese al tamaño reducido del parque automotor de este tipo de vehículos, estos representaron una fracción significativa del total de siniestros viales registrados en el departamento.

La problemática central identificada radica en la ausencia de sistemas tecnológicos automatizados que permitan monitorear de forma continua, objetiva y sistemática el cumplimiento de la normativa del uso obligatorio del casco en motociclistas en la ciudad de Cobija. Esta carencia genera consecuencias múltiples: imposibilidad de fiscalización efectiva en tiempo real, ausencia de datos estadísticos confiables sobre tasas de cumplimiento, limitaciones en la capacidad de respuesta de autoridades de tránsito, y perpetuación de comportamientos de riesgo que incrementan la probabilidad y severidad de lesiones craneoencefálicas en accidentes viales.

1.4 FORMULACIÓN DEL PROBLEMA

¿En qué medida un modelo de visión por computadora basado en la arquitectura YOLOv11m permitirá la detección del uso de casco en motociclistas con un nivel de precisión media (mAP@0.5) superior al 90% en el contexto urbano de Cobija?

1.5 HIPÓTESIS

Mediante el desarrollo de un modelo de visión por computadora basado en la arquitectura YOLOv11m, se logrará detectar el uso de casco en motociclistas con un nivel de precisión media (mAP@0.5) superior al 90%, evaluado sobre imágenes capturadas en condiciones reales en el contexto urbano de Cobija.

- **Variables:**

Independiente: Modelo de visión por computadora YOLOV11m.

Dependiente: Precisión Promedio Media (mAP)

Tabla I.1: Operacionalización de variables

	Definición conceptual	Dimensiones	Indicadores	Instrumentos
Modelo de visión por computador YOLOv11m Variable Independiente	Sistema computacional entrenado mediante redes neuronales convolucionales para interpretar datos visuales y realizar tareas de detección, clasificación o segmentación de	Proceso de entrenamiento	Dataset, de Entrenamiento (Constante), número de épocas	Registros de entrenamiento, ficha técnica, scrips
		Implementación	Entorno de prueba en la	Ficha técnica, Registros de ejecución/logs

	objetos en imágenes (Szeliski, 2022).		nube (Constante)	
Precisión Promedio media (mAP) Variable Dependiente	. La precisión promedio media (mAP) se define como el promedio de las precisiones promedio (AP) obtenidas para cada clase, proporcionando una medida unificada para evaluar la exactitud global del detector de objetos (Padilla et al., 2021).	Precisión por clase	Valor de precisión para cada categoría	Librería Ultralytics
		Recall por clase	Valor de recall para cada categoría.	Librería Ultralytics
		AP por clase	Average Precision calculada para cada clase.	Librería Ultralytics
		Rendimiento o global	mAP como valor final promediado.	Librería Ultralytics
Calidad de imagen (Variable de control)	Características visuales de las imágenes resolución espacial, resolución de amplitud y nitidez que afectan la precisión de detección de objetos en redes neuronales (Hao et al., 2022).	Resolución	Número de píxeles por imagen o cuadro	Metadatos de imágenes, ficha técnica
		Condiciones de captura	Iluminación, ángulo de cámara, presencia de ruido visual	Registro de condiciones, metadatos

Nota. En el cuadro se describe las variables identificadas para la investigación a partir de la revisión documental. Fuente: Elaboración propia, 2025.

1.6 OBJETIVOS

1.6.1 Objetivo general

Desarrollar un modelo de visión por computadora basado en la arquitectura YOLOv11m para detectar el uso de casco en motociclistas con un nivel de precisión media (mAP@0.5) superior al 90% en el contexto urbano de Cobija.

1.6.2 Objetivos específicos

- Procesar y etiquetar un conjunto de datos que combine imágenes públicas y locales representativas de Cobija para el entrenamiento del modelo.
- Entrenar la arquitectura YOLOv11m variando el hiperparámetro épocas, para optimizar la detección de las clases 'con casco', 'sin casco' y 'motociclista'.
- Implementar el modelo con todos sus componentes para su correcta ejecución en un entorno simulado.
- Evaluar el desempeño del modelo mediante métricas técnicas (mAP, Recall, Precisión) sobre imágenes inéditas del contexto local.

1.7 JUSTIFICACIÓN

La presente investigación se realiza con la finalidad de desarrollar un Modelo de Visión por Computadora para detectar y clasificar el uso del casco en motociclistas en la ciudad de Cobija. El estudio se justifica en función de su conveniencia, relevancia social e implicaciones prácticas.

El estudio se realiza con la finalidad de desarrollar y validar un modelo de detección que detecta y clasifica el uso del casco en motociclistas en la ciudad de Cobija. Esta iniciativa surge ante el aumento de accidentes de tránsito protagonizados por motociclistas y la escasa cultura vial asociada con el no uso del casco, lo que constituye una problemática social relevante por el riesgo que implica para la vida humana.

Desde el ámbito social, la evidencia empírica generada por este modelo sirve como base para la concientización ciudadana y el fortalecimiento de políticas de prevención vial, promoviendo el uso del casco como medida de protección.

Desde el punto de vista económico, el desarrollo de sistemas automáticos de detección podría contribuir a la mitigación de los costos derivados de los accidentes de tránsito, incluyendo los asociados a servicios de salud y daños materiales. La exploración de tecnologías accesibles, como cámaras y Modelos de Visión por Computadora, se presenta como una alternativa de inversión eficiente frente a los métodos de control tradicionales.

Técnicamente, la investigación representa un aporte relevante al aplicar Modelos de Visión por Computadora en el ámbito de la seguridad vial urbana. El desarrollo de un sistema capaz de detectar con precisión el uso del casco en motociclistas en la ciudad de Cobija introduce una herramienta tecnológica innovadora que fortalece el monitoreo automatizado del cumplimiento de normas de tránsito. La propuesta es escalable, adaptable a otros entornos urbanos y ofrece un enfoque replicable que contribuye a la prevención de accidentes, a la toma de decisiones basadas en datos y al uso eficiente de la inteligencia artificial en espacios públicos. Esta utilidad técnica y la posibilidad de ofrecer un enfoque replicable cumplen con el criterio de Valor Teórico y Generalización que debe poseer una investigación.

1.8 DISEÑO DE LA INVESTIGACIÓN

1.8.1 Enfoque de investigación

La presente investigación se conduce bajo un enfoque cuantitativo. Para (La Madriz, 2019) el enfoque cuantitativo “...usa la recolección de datos para probar hipótesis, con base en la medición numérica y el análisis estadístico, para establecer patrones de comportamiento y probar teorías” (p. 69).

Este enfoque permite medir y estimar el rendimiento del modelo mediante indicadores empíricos como mAP, precisión y recall, vinculando conceptos abstractos con datos numéricos objetivos para someter a prueba la hipótesis.

1.8.2 Tipo y nivel de investigación

La presente investigación es de tipo aplicada. Para (Hernández et al., 2014), la investigación aplicada tiene fines prácticos en el sentido de solucionar problemas detectados en un área del conocimiento. Está ligada a la aparición de necesidades o problemas concretos y al deseo del investigador de ofrecer solución a estos.

Este tipo se adopta porque aborda un problema real que afecta a la sociedad mediante el desarrollo y la validación de un Modelo de Visión por Computadora, el cual proporciona un aporte al conocimiento y demuestra el potencial de la tecnología para contribuir a la seguridad vial.

A su vez, la investigación se sitúa en un nivel explicativo. De acuerdo a (Hernández et al., 2014), la investigación explicativa se enfoca en determinar las causas de un fenómeno, buscando entender por qué ocurre y en qué condiciones se manifiesta.

La elección de este nivel permite establecer la relación causal entre la Variable Independiente (el entrenamiento del Modelo de Visión por Computadora, manipulando el Número de Épocas) y el rendimiento obtenido (Variable Dependiente: Precisión Promedio Media o mAP). Este nivel es sumamente estructurado y brinda el sentido de entendimiento del fenómeno y de sus causas. Además, la naturaleza explicativa permite someter a prueba la hipótesis de la investigación, lo cual es un requisito en este alcance del estudio.

1.8.3 Diseño de investigación

El diseño es cuasiexperimental. Según Hernández et al. (2014), en el diseño cuasiexperimental se manipulan deliberadamente, al menos, una variable independiente, pero los grupos ya están conformados, es decir, no se asignan al azar.

Este diseño permite manipular intencionalmente la Variable Independiente y establecer su efecto causal sobre el rendimiento, proporcionando un modelo de verificación que contrasta la visión teórica con los datos empíricos.

1.8.4 Métodos de investigación

- **Método deductivo:** Para Maya (2014), el método deductivo "es una forma de razonamiento que parte de una verdad universal para obtener conclusiones particulares" (p. 14).

Este método guía el proceso al partir de la teoría para derivar hipótesis que se someten a prueba empírica, proporcionando la base lógica para la construcción y verificación del modelo.

- **Método analítico:** Según La Madriz (2019), el análisis "es un procedimiento lógico que posibilita descomponer mentalmente un todo en sus partes y cualidades, en sus múltiples

relaciones, propiedades y componentes, que permite estudiar el comportamiento de cada parte" (p. 72).

Este método permite examinar cómo cada componente del modelo influye en su precisión y efectividad.

- **Método sintético:** Según La Madriz (2019), la síntesis "es la operación inversa, que establece mentalmente la unión o combinación de las partes previamente analizadas y posibilita descubrir relaciones y características generales entre los elementos de la realidad" (p. 73).

Este método integra los conceptos teóricos con los elementos técnicos para configurar un modelo funcional, reconstruyendo la totalidad a partir de sus componentes.

- **Método experimental:** Para Arias (2012), el método experimental es un proceso donde se somete un objeto o grupo de individuos a condiciones, estímulos o tratamientos específicos para observar los efectos o reacciones resultantes.

Este método manipula deliberadamente el modelo de visión por computadora con el estímulo número de épocas para observar sus efectos en el rendimiento, permitiendo someter a prueba la hipótesis mediante escrutinio empírico.

Esta combinación metodológica es coherente con el enfoque cuantitativo y el nivel explicativo de la investigación, permitiendo una validación técnica rigurosa del modelo propuesto en condiciones reales de la ciudad de Cobija.

1.8.5 Población y muestra

- **Población:** Para (Guillen *et al.*, 2020), la población es "el conjunto de todos los individuos (objetos, personas, situaciones, entre otros) a investigar" [p. 89]. El proceso inicia con la definición de la unidad de muestreo/análisis y la delimitación de la población que se estudia.

La población de esta investigación está conformada por todos los motociclistas que circulan en la ciudad de Cobija. Esta población representa el universo de casos que concuerdan con las especificaciones de la investigación. La población se sitúa claramente por sus características de contenido ("motociclistas"), lugar ("Ciudad de Cobija") y tiempo

(“durante el periodo de recolección de datos”). Esta delimitación es necesaria para establecer los parámetros muestrales y asegurar la viabilidad del estudio.

- **Muestra:** La muestra según (Hernández y Mendoza, 2018), “es un subgrupo del universo o población del cual se recolectan los datos y que debe ser representativo de esta, si se desean generalizar los resultados” (p. 196).

La selección se realiza mediante un muestreo no probabilístico por conveniencia. Este enfoque es adecuado porque la unidad de análisis no es el motociclista, sino la imagen en sí misma, lo cual prioriza la representatividad técnica sobre la generalización estadística poblacional. La muestra de campo local que consiste en 200 imágenes capturadas en puntos estratégicos, específicamente en la Av. 9 de Febrero, entre Av. Otto Felipe Braun y Av. Pando, así como en la Av. 27 de Mayo. La recolección se lleva a cabo durante horarios diurnos en la ciudad de Cobija, lo que garantiza la variedad de datos necesaria para la validación.

1.8.6 Técnicas e instrumentos de recolección de datos

La técnica empleada es la observación estructurada. Para Cabrera (2020), la observación estructurada consiste en el registro sistemático, válido y confiable de comportamientos o conducta manifiesta.

Esta técnica es fundamental dentro del enfoque cuantitativo debido a que permite el uso sistemático de los sentidos orientado a la captación de la realidad que se quiere estudiar, con la finalidad de que los fenómenos deban poder observarse o referirse al “mundo real”. La elección de la modalidad estructurada se realiza porque implica el uso de una guía diseñada previamente que especifica los elementos a observar, lo que garantiza la homogeneidad de los datos y facilita su cuantificación

El instrumento es la ficha de recolección de datos estructurada. Para (Cabrera, 2020), una ficha de recolección de datos es un documento o formulario diseñado para recopilar y organizar información relevante de manera estructurada.

Se utilizan tres fichas estructuradas:

- **Ficha de Revisión de Datasets Públicos:** Este instrumento evalúa la calidad, relevancia y adecuación del conjunto de datos secundarios empleados durante el entrenamiento, verificando clases relevantes, variedad de escenarios y formato de anotación.
- **Ficha de Registro de Recolección de Imágenes:** Este instrumento documenta los detalles técnicos y contextuales de los datos primarios obtenidos en campo. convirtiendo las observaciones en datos cuantificables mediante la asignación de valores numéricos o simbólicos.
- **Ficha de Registro de Pruebas:** Este instrumento documenta los datos cuantitativos del rendimiento del modelo durante la evaluación final, tales como el número de épocas y las métricas de evaluación, permitiendo que los datos sean procesados y analizados con procedimientos estadísticos apropiados.

1.8.7 Validación del instrumento

La validación se realiza mediante juicio de expertos. Según Escobar-Pérez y Cuervo-Martínez (2008), el juicio de expertos "se define como una opinión informada de personas con trayectoria en el tema, que son reconocidas por otros como expertos cualificados en éste, y que pueden dar información, evidencia, juicios y valoraciones" (p. 29). Este procedimiento permite obtener evidencias de validez de contenido al evaluar sistemáticamente si los instrumentos representan adecuadamente las dimensiones que se pretenden medir. Para esta validación se contara con la participación de tres profesionales especializados en Inteligencia Artificial y ciencia de datos, quienes evalúan la claridad, coherencia y pertinencia de las fichas técnicas. Este proceso garantiza la validez del instrumento en el contexto del desarrollo de modelos de detección de objetos.

1.8.8 Validación de hipótesis

El método empleado para la validación de hipótesis es el enfoque práctico. Para Lazos Martínez (2004), "la validación es la confirmación y provisión de evidencia objetiva de que se cumplen los requisitos para un uso o aplicación prevista". Este enfoque es aplicable a la validación de modelos, sistemas y herramientas computacionales, permitiendo asegurar que la herramienta propuesta satisface una función determinada mediante la comparación sistemática entre los requisitos establecidos y las características de desempeño observadas.

El enfoque práctico de validación proporciona un marco riguroso y estructurado para verificar si una herramienta técnica cumple los requisitos definidos para la función que se le asigna, mediante la comparación sistemática entre dichos requisitos y las características de desempeño observadas en la herramienta validada. En este estudio, dicho enfoque se aplica al modelo YOLOv11m concebido como herramienta de detección, cuya función consiste en identificar correctamente el uso de casco en motociclistas a partir de imágenes capturadas en el contexto urbano de Cobija, bajo el requisito cuantitativo de alcanzar una precisión promedio media (mAP) igual o superior al 90%. La validación se organiza en cinco etapas interrelacionadas: la identificación explícita de la función del modelo, la especificación de requisitos y métricas de desempeño, la determinación empírica del rendimiento mediante experimentación controlada, la comparación objetiva de los resultados obtenidos con los umbrales establecidos y, finalmente, la formulación de un juicio de validez sustentado en evidencias cuantitativas. De este modo, el enfoque práctico permite que la validación de la hipótesis se fundamente en resultados medibles, evitando interpretaciones subjetivas y asegurando que el modelo cumple de manera demostrable con la función para la cual se diseña.

1.8.9 Procedimientos de análisis de datos

El análisis de datos sigue un enfoque cuantitativo técnico-evaluativo centrado en métricas de rendimiento del modelo, según los estándares de evaluación en aprendizaje profundo propuestos por Padilla et al. (2021).

A continuación, se describe el análisis cuantitativo:

- **Precision:** Proporción de detecciones correctas entre todas las detecciones realizadas por el modelo.
- **Recall:** Proporción de objetos reales detectados correctamente entre todos los objetos presentes en las imágenes.
- **Precisión Promedio Media (mAP):** Métrica global que promedia el AP a través de umbrales de IoU (0.5), evaluando el rendimiento general del modelo.
- **Matriz de Confusión Normalizada:** Matriz visual que resume verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos por clase.

- **Representación gráfica:** Análisis de errores de clasificación y detección en el conjunto de prueba local.

En el siguiente cuadro se describe las etapas que permitirán obtener los productos descritos en los objetivos específicos:

Tabla I.2: Descripción de las etapas y procedimiento de la investigación

Etapas	Descripción	Técnicas	Herramientas	Producto
Revisión teórica	Revisión documental intensiva sobre Visión por Computadora para sustentar teóricamente el estudio	Análisis documental	Artículos científicos, Documentación técnica, libros especializados	Marco teórico y conceptual finalizado
Búsqueda del dataset público	Identificación, descarga y análisis de un dataset público que contiene imágenes de motociclistas, adecuado para el entrenamiento del modelo	Observación estructurada, Revisión de repositorios y fuentes abiertas	Repositorios abiertos, Plataforma Roboflow	Dataset publico seleccionado, verificado y listo para entrenamiento
Recolección de imágenes locales	Captura de 200 imágenes de motociclistas para la creación del Conjunto de Prueba (Test Set) y validación del modelo en el contexto específico de Cobija	Observación estructurada	Cámara del dispositivo móvil Infinix Smart 8	Conjunto de imágenes local (200 imágenes de campo)
Etiquetado de imágenes locales.	Anotación de las 200 imágenes locales para establecer la ground truth (verdad fundamental). Etiquetado de tres clases: "cabeza con casco", "cabeza sin casco" y "motociclista"	Etiquetado manual	Plataforma RoboFlow	Dataset test etiquetado y codificado listo para la fase de evaluación.
Entrenamiento del modelo	Ejecución del entrenamiento del modelo de Visión	Observación estructurada,	Kaggle, Framework modelo	Modelo entrenado

	por Computadora utilizando el dataset seleccionado. Manipulación intencional de diferentes cantidades de épocas (de 10 a 100) como indicador	Entrenamiento supervisado	Yolov11m, Python, RoboFlow	
Implementación del modelo	Integración del modelo entrenado con sus componentes finales (dataset local de prueba, dependencias de Ultralytics y configuración del entorno), para habilitar su funcionalidad en el entorno de simulación y preparar la ejecución de pruebas de evaluación.	Implementación técnica en entorno simulado	Google Colab, Python, Ultralytics,	Modelo implementado en entorno simulado, listo para evaluación
Evaluación del modelo	Evaluación del rendimiento del modelo sobre el conjunto de prueba de 200 imágenes inéditas. Prueba de la hipótesis basada en el cumplimiento de un umbral específico de desempeño	Prueba de rendimiento	Librerías model.val() de Ultralytics	Informe de desempeño (precisión, recall, mAP@0.5)

Nota. En el cuadro se describe las etapas y procedimiento del desarrollo de la investigación las cuales se estructuraron en función de los objetivos específicos planteados. Fuente: Elaboración propia, 2025.

1.9 ALCANCES

La presente investigación se delimita en dimensiones temáticas, temporales y contextuales claramente definidas:

- **Temático:** La investigación se desarrolla en el campo de la Visión por Computadora, una rama de la Inteligencia Artificial enfocada en permitir que los sistemas informáticos interpreten y comprendan imágenes del entorno. Dentro de este campo, se abordaron temáticas clave como el procesamiento digital de imágenes, la extracción de características, y el uso de técnicas de aprendizaje profundo para tareas de detección de objetos.

Se profundiza en el uso de redes neuronales convolucionales (CNN), específicamente mediante la arquitectura YOLO (You Only Look Once), reconocida por su eficiencia en la detección en tiempo real. Estas técnicas se aplicaron al problema de la detección del uso de casco en motociclistas, con el objetivo de validar un modelo de detección enfocado en la detección del uso de casco como caso de estudio técnico en visión por computadora.

- **Temporal:** El estudio se desarrolló durante el periodo comprendido entre marzo a octubre del año 2025, abarcando todas las fases de la investigación: revisión bibliográfica, recolección de datos, entrenamiento del modelo, implementación en entorno simulado, evaluación de resultados y redacción del informe final.
- **Contextual:** El presente trabajo se realiza en la ciudad de Cobija, departamento de Pando, Estado Plurinacional de Bolivia, donde se llevó a cabo el desarrollo y la validación de un modelo de detección basado en visión por computadora, orientado a la detección del uso de casco en motociclistas.
- **Delimitaciones y Restricciones:** El estudio se limitó al análisis de imágenes estáticas de motociclistas, capturadas exclusivamente en condiciones diurnas, por lo que no se abordó la detección de motociclistas o el uso del casco en condiciones nocturnas ni otro tipo de comparaciones del modelo ante situaciones naturales climáticas, vestuarios etc. Esta decisión respondió a las limitaciones técnicas y de alcance propias de la etapa inicial del proyecto.

Asimismo, no se generan estadísticas poblacionales sobre el uso del casco ni se realiza un análisis social, legal o de impacto directo en la seguridad vial. Tampoco se contempló la implementación operativa del sistema en espacios públicos, ya que el alcance se restringió a una etapa técnica preliminar, desarrollada en un entorno de prueba simulado.

CAPITULO II.

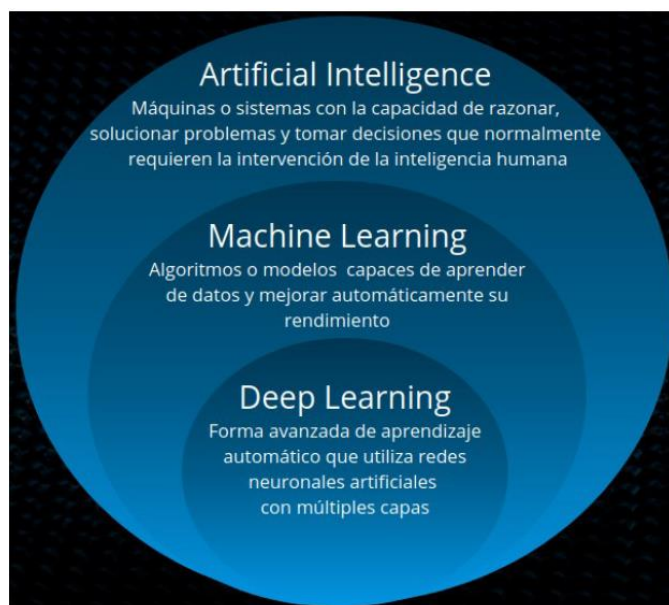
MARCO REFERENCIAL

2.1 MARCO TEÓRICO

2.1.1 Inteligencia artificial

La inteligencia artificial (IA) es una rama de la informática que se ocupa del diseño de sistemas capaces de realizar tareas que, normalmente, requieren inteligencia humana. Estas tareas incluyen el reconocimiento de patrones, el aprendizaje a partir de datos, la toma de decisiones y la comprensión del lenguaje natural. Según Vera-Rubio (2023), la IA se puede definir como “el estudio de los agentes que reciben percepciones del entorno y realizan acciones”, como se muestra en la (Figura 2.1).

Figura II.1: Categorías de IA



Nota. Adaptado de *sistema de monitorización de tráfico en directo mediante detección de objetos basada en deep learning (p. 10)*, por F. E. García Borges, 2025.

En este sentido, la IA busca simular procesos cognitivos humanos a través de algoritmos y modelos computacionales que permiten a las máquinas “pensar” o actuar de manera autónoma.

Desde una perspectiva más aplicada, la inteligencia artificial puede entenderse como el conjunto de técnicas y metodologías que permiten a una máquina aprender de la experiencia y adaptarse a nuevas situaciones sin intervención humana directa. La IA abarca desde sistemas simples de automatización hasta redes neuronales profundas capaces de identificar objetos,

reconocer voz o diagnosticar enfermedades. Esta capacidad de análisis y adaptación ha hecho que la IA se convierta en una herramienta fundamental en campos como la medicina, la robótica, la seguridad y, recientemente, en la gestión del tránsito urbano (Peinado García, 2022).

2.1.2 Machine learning

El machine learning o aprendizaje automático es una rama de la inteligencia artificial cuyo objetivo es desarrollar algoritmos capaces de identificar patrones dentro de grandes volúmenes de datos. Estos algoritmos tienen la capacidad de aprender de manera automática y mejorar su rendimiento en la ejecución de tareas o en la formulación de predicciones, sin ser programados explícitamente para cada caso. Según esta perspectiva, el aprendizaje automático permite que los sistemas se adapten a nuevos datos, incrementando su precisión y eficacia a lo largo del tiempo (Martínez Heras, 2022).

El proceso de machine learning se compone de varias etapas. Primero, se realiza una exploración de los datos disponibles para comprender el problema que se desea resolver. Luego, se define un criterio de evaluación para medir el rendimiento del modelo. Posteriormente, los datos se preparan y dividen en conjuntos de entrenamiento, validación y prueba, cada uno con funciones específicas. Después, se selecciona el algoritmo adecuado al tipo de problema y finalmente se entrena el modelo, ajustándolo para que aprenda a partir de los datos. Existen distintos tipos de aprendizaje, como el supervisado, no supervisado y por refuerzo, cada uno con su propia metodología y aplicación (Martínez Heras, 2022).

2.1.3 Aprendizaje supervisado

El aprendizaje supervisado es el tipo de aprendizaje automático que utiliza un conjunto de datos etiquetados y su objetivo es predecir tales etiquetas. Este método de aprendizaje requiere la utilización de menos datos en el proceso de entrenamiento, pero tales datos tienen que estar bien etiquetados. Si el conjunto de datos utilizado no es lo suficientemente preciso, el modelo puede estar sesgado y no se obtendrán resultados adecuados al enfrentarse a nuevos datos (Peinado García, 2022).

Existen dos tipos de aprendizaje supervisado, la clasificación y la regresión. En clasificación, los algoritmos son entrenados para clasificar los datos en variables discretas. Hay distintos tipos de clasificación, entre los más importantes se encuentran: la clasificación binaria, la

clasificación multiclase, la clasificación de etiquetas múltiples y la clasificación con datos desbalanceados (Peinado García, 2022).

Respecto a la regresión, se pretenden establecer relaciones entre un número de características y una variable objetivo-continua. El ejemplo más común es el de la regresión lineal que se utiliza para establecer una relación lineal entre los datos de entrada y salida. Además de la regresión lineal, existen otros tipos de regresión como pueden ser la regresión logística o la regresión polinomial (Peinado García, 2022).

2.1.4 Aprendizaje no supervisado

El aprendizaje no supervisado es una modalidad del aprendizaje automático que se centra en analizar los datos sin necesidad de etiquetas previas. A diferencia del aprendizaje supervisado, en este enfoque los algoritmos trabajan con datos no etiquetados, buscando identificar estructuras, similitudes, patrones ocultos y formar agrupaciones. Este método resulta especialmente útil para explorar información desconocida, ya que permite detectar patrones no evidentes o analizar grandes volúmenes de datos que no podrían examinarse manualmente. Los hallazgos obtenidos facilitan la comprensión de los datos y la generación de vectores de características (Martinez Morales, 2022).

2.1.5 Aprendizaje por esfuerzo

En esta forma de aprendizaje automático, también se trabaja sin datos etiquetados y los algoritmos deben aprender de manera autónoma. A diferencia del aprendizaje no supervisado, aquí no se busca agrupar los datos, sino que el aprendizaje se basa en un sistema de recompensas y castigos dentro de un entorno dinámico y cambiante. En lugar de minimizar una función de error, el objetivo es maximizar las recompensas obtenidas. Este sistema se compone de dos elementos principales: el agente, encargado de tomar decisiones, y el entorno, que representa el contexto en el que actúa el agente, con sus propias reglas y restricciones que pueden variar con el tiempo. La interacción entre ambos se da a través de acciones del agente, los estados del entorno y las recompensas o penalizaciones generadas por cada decisión tomada (Martinez Morales, 2022).

2.1.6 Visión por computadora

La visión por computador es una de las aplicaciones de las técnicas del deep learning para extraer información a partir del análisis de imágenes y videos. Gracias a la aplicación de estas técnicas y métodos podemos procesar, analizar y extraer información numérica que los algoritmos utilizarán para diversos fines. Estas técnicas pueden ser útiles para detectar objetos y clasificar imágenes, reconstruir escenas, estimar movimientos, restaurar imágenes, seguimiento de vídeo, conducción autónoma, entre otras. Dentro de este campo podemos destacar las redes de clasificación de imágenes y las de detección de objetos (Peinado García, 2022).

La visión por computadora es de gran importancia porque permite a las máquinas interpretar y comprender el contenido de imágenes y videos de manera similar al ojo humano. Esta capacidad abre un amplio abanico de aplicaciones en distintos sectores, como la seguridad, la salud, el transporte, la industria y la agricultura. Gracias a esta tecnología, se pueden automatizar tareas como la detección de objetos, el reconocimiento facial, el monitoreo en tiempo real o el diagnóstico médico, mejorando la eficiencia y reduciendo errores humanos (Sarmah et al., 2024).

Además, la visión por computadora contribuye significativamente a la transformación digital, ya que facilita la toma de decisiones basada en datos visuales y en tiempo real. Por ejemplo, en la seguridad vial, puede ser utilizada para detectar infracciones, como el no uso del casco en motociclistas; en medicina, para detectar anomalías en radiografías; o en la industria, para realizar controles de calidad automatizados. Por tanto, su importancia radica en su capacidad para resolver problemas complejos con rapidez, precisión y escalabilidad (Jia et al., 2021).

2.1.7 Calidad de imagen

La calidad de imagen se refiere a las características técnicas de una imagen digital que impactan el rendimiento de los algoritmos de detección de objetos, siendo determinada principalmente por dos factores: la resolución espacial (cantidad de píxeles de la imagen) y la resolución de amplitud (controlada por el tamaño del paso de cuantización del código de compresión).

La resolución espacial afecta significativamente la precisión de detección, ya que imágenes de mayor resolución permiten que los objetos sean representados con más píxeles, facilitando su reconocimiento por parte de las redes neuronales convolucionales. Por otro lado, la resolución de

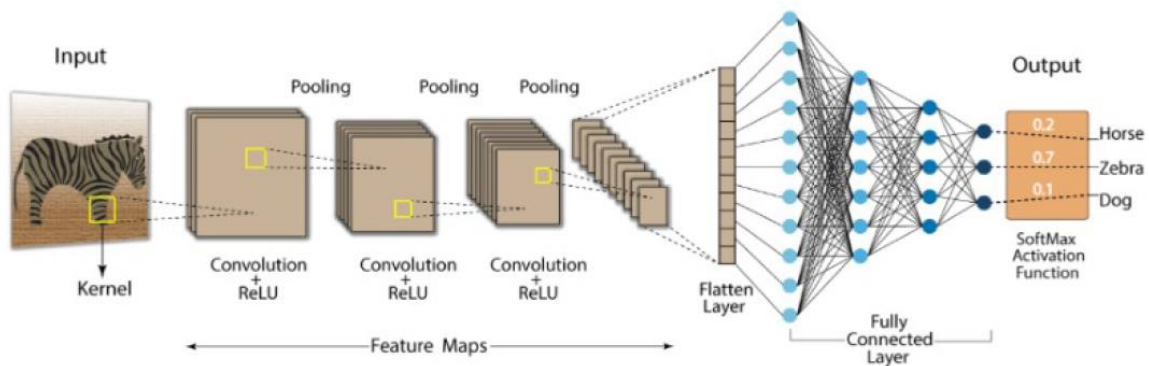
amplitud relacionada con el nivel de compresión aplicado a la imagen presenta un efecto menos pronunciado hasta alcanzar umbrales críticos de cuantización, punto en el cual la precisión de detección disminuye abruptamente.

La estandarización de estas características en estudios de visión por computadora es fundamental para controlar variables que podrían afectar el rendimiento del modelo de manera independiente a los factores experimentales bajo investigación (Hao et al., 2022).

2.1.8 Redes neuronales convolucionales

Las redes neuronales convolucionales, conocidas como Convolutional Neural Networks (CNN), son un tipo especializado de red neuronal diseñado para procesar datos con estructura en forma de imagen. Estas redes transforman los píxeles de una imagen en matrices numéricas que permiten identificar patrones visuales. A través de capas convolucionales, las CNN extraen características relevantes de las imágenes como observa en la (Figura 2.2), comenzando por formas simples como líneas o bordes, y progresando hasta reconocer figuras complejas como siluetas u objetos completos (Mercado Reyna, 2023).

Figura II.2: Arquitectura de CNN



Nota. Adaptado de *sistema de monitorización de tráfico en directo mediante detección de objetos basada en deep learning (p. 15)*, por F. E. García Borges, 2025.

Las CNN operan aplicando convoluciones, que son operaciones matemáticas que combinan los datos de entrada con filtros o kernels. Estas capas permiten que distintas neuronas se especialicen en identificar ciertos aspectos visuales, conectándose solo a regiones específicas de la imagen para detectar patrones locales. Gracias a esta arquitectura jerárquica, las redes neuronales

convolucionales son altamente eficaces en tareas como clasificación de imágenes, detección de objetos y reconocimiento facial (Mercado Reyna, 2023).

2.1.9 Descenso de gradiente y convergencia en entrenamiento

El entrenamiento de redes neuronales profundas funciona mediante un proceso iterativo de ajuste de pesos que busca minimizar el error del modelo. El método principal utilizado para este ajuste se denomina descenso de gradiente, el cual calcula qué tan lejos está el modelo de la respuesta correcta y modifica los pesos de la red en la dirección que reduce este error. Existen tres formas de aplicar este método: procesando todo el conjunto de datos a la vez (descenso de gradiente por lotes), procesando un solo dato por vez (descenso de gradiente estocástico), o procesando grupos pequeños de datos (descenso de gradiente por mini-lotes). Esta última variante es la más utilizada en la práctica (Géron, 2019).

Durante el entrenamiento, es común observar fluctuaciones temporales en el rendimiento del modelo, donde las métricas pueden subir y bajar entre épocas consecutivas. Este comportamiento ocurre porque el algoritmo de descenso de gradiente estocástico introduce variabilidad al utilizar subconjuntos aleatorios de datos en cada iteración, lo que genera cambios en la dirección del ajuste de pesos. Aunque estas oscilaciones pueden parecer problemas de entrenamiento, en realidad son beneficiosas porque permiten al modelo explorar diferentes soluciones y evitar quedar atrapado en configuraciones subóptimas. Con suficientes iteraciones de entrenamiento, el modelo tiende a estabilizarse y alcanzar un punto de equilibrio donde el error se minimiza (Géron, 2019).

El proceso completo de entrenamiento atraviesa tremendamente tres etapas: una fase inicial donde el modelo aprende rápidamente los patrones más evidentes y reduce el error de forma acelerada; una fase intermedia donde se observan oscilaciones mientras el modelo refina su aprendizaje; y una fase final de estabilización donde se alcanza el mejor equilibrio entre recordar los ejemplos de entrenamiento y poder funcionar correctamente con datos nuevos (Géron, 2019).

2.1.10 Robustez y aprendizaje invariante en redes neuronales profundas

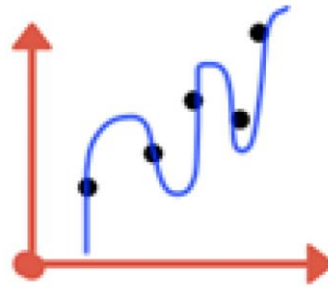
Las redes neuronales profundas para visión por computadora enfrentan el desafío central de aprender representaciones invariantes a distintas condiciones visuales que pueden presentarse en imágenes del mundo real. Estas condiciones incluyen cambios en la iluminación, ángulos de captura, presencia de accesorios o variaciones contextuales que modifican la apariencia de los objetos. Para lograr una detección robusta y generalizable, el modelo debe aprender a discriminar las características esenciales e ignorar o compensar aquellas que son irrelevantes o cambian frecuentemente. Cuando una clase exhibe alta heterogeneidad visual (gran variabilidad intraclase), como sucede en la detección de cascos de diferentes formas, colores y estados, el modelo requiere una mayor capacidad representacional y mayor cantidad de datos y entrenamiento para poder generalizar. Por el contrario, clases con baja variabilidad interna y gran diferencia interclase, como la silueta anatómica básica de la cabeza humana o la estructura común de motocicletas, son más sencillas de aprender y clasificar (Chollet, 2018).

El proceso de aprendizaje de redes profundas se caracteriza por distintas dinámicas a lo largo de las épocas: en una fase inicial, las redes suelen aprender con rapidez (“fase de alto gradiente”) los patrones visuales más generales y repetibles presentes en los datos, como bordes, texturas básicas o formas predominantes. Este aprendizaje acelerado permite que el modelo alcance mejoras significativas en las primeras iteraciones, consolidando la base para el reconocimiento de patrones cada vez más complejos en fases posteriores del entrenamiento (Chollet, 2018).

2.1.11 Sobreajuste

El sobreajuste, también conocido como overfitting, es un problema común en modelos de aprendizaje supervisado donde el modelo se ajusta excesivamente a un conjunto de datos específico. Esto hace que el modelo "memorice" esos datos y parezca que ha aprendido perfectamente, pero en realidad no puede generalizar ni predecir correctamente con datos nuevos. En redes neuronales artificiales, el sobreajuste puede ocurrir independientemente de la arquitectura o algoritmos usados, por lo que se aplican técnicas para mitigarlo. El primer paso para combatir el sobreajuste (Figura 2.3) es poder identificarlo.

Figura II.3: Sobreajuste



Nota. Adaptado de *Entrenamiento de la Red Neuronal Convolutiva YOLO para objetos propios*, por E. R. Silva Guzmán, 2020.

Una de las maneras más fáciles de identificarlo es dividir la base de datos en dos conjuntos, uno de entrenamiento y el otro de validación. Teniendo los conjuntos de entrenamiento y validación se pueden obtener las gráficas de la función de pérdida de cada uno de ellos durante el entrenamiento de la ANN (Yamashita et al., 2018).

2.1.12 Equilibrio sesgo varianza en redes neuronales profundas

El desempeño de los modelos de aprendizaje profundo depende del equilibrio entre dos tipos de error: el sesgo y la varianza, elementos esenciales para garantizar que el modelo generalice bien a nuevos datos. Según Goodfellow et al. (2016), el error total de un modelo se puede descomponer en tres partes: el sesgo (error por simplificaciones excesivas, que lleva a no captar toda la complejidad del problema), la varianza (sensibilidad a los cambios en los datos de entrenamiento, que conduce al sobreajuste), y el ruido irreducible (información imposible de modelar). Es fundamental encontrar un punto de equilibrio porque minimizar demasiado uno de estos errores tiende a aumentar el otro (Goodfellow et al., 2016).

En el contexto de redes neuronales convolucionales aplicadas a tareas de detección de objetos, este equilibrio se ve influenciado por la capacidad de representación de la red y la variabilidad en los datos. Cuando existe gran diversidad dentro de una clase, como ocurre con diferentes tipos y estilos de cascos, el modelo necesita más capacidad y datos para aprender los patrones relevantes y evitar caer en el sobreajuste. Goodfellow et al. (2016) destacan que las redes profundas pueden desarrollar “representaciones robustas”, extrayendo múltiples características

visuales útiles para identificar un objeto aunque algunos elementos estén ausentes o cambien de aspecto, lo que favorece la detección confiable en contextos variados.

2.1.13 Memorización y artefactos en el entrenamiento de redes profundas

El proceso de entrenamiento de redes neuronales profundas presenta un comportamiento característico cuando se extiende más allá del punto de convergencia óptima: el modelo comienza a memorizar características específicas del conjunto de entrenamiento en lugar de aprender patrones generalizables. Este fenómeno, conocido como sobreajuste u overfitting, ocurre cuando la red ajusta sus pesos de manera excesiva a las particularidades idiosincráticas de los datos de entrenamiento, incluyendo ruido visual, variaciones aleatorias y artefactos propios del proceso de captura o etiquetado. Como resultado, el rendimiento del modelo mejora en los datos de entrenamiento pero se degrada en datos nuevos no vistos, indicando una pérdida de capacidad de generalización (Szeliski, 2022).

Durante las fases avanzadas del entrenamiento, cuando el modelo ya ha capturado los patrones fundamentales de las clases, las iteraciones adicionales pueden llevar a que la red aprenda detalles irrelevantes que no son representativos de la verdadera distribución de datos en el mundo real. Esto se manifiesta como un incremento en la diferencia entre el rendimiento en el conjunto de entrenamiento y el conjunto de validación, señal clara de que el modelo está perdiendo su habilidad para funcionar correctamente con información nueva (Szeliski, 2022).

2.1.14 Redes de clasificación de imágenes

Las redes de clasificación de imágenes son un tipo de modelo de inteligencia artificial, usualmente basado en redes neuronales profundas, que tienen como objetivo principal asignar una categoría o etiqueta específica a una imagen completa. Estas redes aprenden a identificar patrones, formas, texturas y colores dentro de una imagen para determinar a qué clase pertenece. Un ejemplo común es una red que clasifica imágenes de animales en categorías como "gato", "perro" o "caballo". Estas redes suelen entrenarse con grandes conjuntos de datos etiquetados y emplean arquitecturas como las redes neuronales convolucionales (CNN) (Peinado García, 2022).

2.1.15 Redes de detección de objetos

La detección de objetos es una tarea fundamental dentro de la visión por computadora que consiste en localizar e identificar múltiples objetos dentro de una imagen, mediante el uso de algoritmos capaces de predecir tanto la categoría de los objetos como su ubicación espacial a través de cajas delimitadoras (bounding boxes).

A diferencia de la clasificación de imágenes, donde se determina una sola etiqueta para toda la imagen, en la detección de objetos se pueden identificar varios elementos simultáneamente, incluso si están parcialmente ocultos o en diferentes escalas.

Existen diferentes algoritmos que abordan esta tarea. Entre los más destacados se encuentran:

YOLO (You Only Look Once): realiza detección en tiempo real, dividiendo la imagen en celdas y prediciendo las cajas y clases en una sola pasada.

SSD (Single Shot Multibox Detector): realiza predicciones en múltiples escalas, combinando velocidad y precisión.

Faster R-CNN: una arquitectura en dos etapas, muy precisa pero más lenta, ideal para contextos donde el tiempo de procesamiento no es crítico.

En este proyecto se emplea un modelo de la familia YOLO, por su eficacia en tareas de vigilancia urbana donde se requiere rapidez y precisión, permitiendo detectar motociclistas y determinar si llevan casco protector.

“Los sistemas de detección de objetos permiten identificar múltiples entidades en una imagen, lo que los convierte en herramientas clave para la vigilancia automatizada, el transporte inteligente y la interacción con entornos físicos” (Szeliski, 2022, p. 407).

2.1.16 Factores que afectan la complejidad de detección de objetos

La dificultad para detectar objetos en imágenes mediante sistemas de visión por computadora está determinada por múltiples factores relacionados con las características visuales y contextuales de los objetos. Uno de los factores más significativos es la dimensión espacial del objeto en la imagen: existe una correlación directa entre el área de píxeles que ocupa un objeto y la facilidad con que puede ser detectado. Objetos más grandes proporcionan mayor cantidad de

información visual a las capas convolucionales de la red, permitiendo la extracción de múltiples características discriminativas en diferentes escalas de resolución. Por el contrario, los objetos pequeños que ocupan pocas posiciones de píxeles presentan mayor dificultad de detección debido a la información visual limitada disponible (Szeliski, 2022).

La variabilidad intraclasa constituye otro factor determinante en la complejidad de detección. Las clases que presentan alta heterogeneidad visual en términos de forma, color, textura o apariencia requieren que el modelo aprenda un espacio de características más amplio y complejo, incrementando la dificultad del problema de reconocimiento. En contraste, clases con características visuales consistentes y baja variabilidad interna son internamente más fáciles de detectar, ya que el modelo puede aprender patrones más uniformes y predecibles (Szeliski, 2022).

El fenómeno de oclusión parcial, donde elementos del entorno obstruyen porciones del objeto de interés, representa un desafío adicional significativo. Las redes neuronales convolucionales tienen mayor dificultad para detectar objetos cuando la información visual crítica está fragmentada o parcialmente ausente, especialmente si la oclusión afecta las características distintivas del objeto. Este problema se acentúa cuando se combina con objetos pequeños, ya que la proporción de información visible se reduce aún más (Szeliski, 2022).

Finalmente, la detección de ausencia de un objeto presenta características particulares: en lugar de identificar un vacío, los sistemas de visión por computadora deben aprender a reconocer características visuales positivas alternativas que indiquen la no presencia del objeto objetivo. Por ejemplo, detectar la ausencia de un casco implica identificar características como cabello visible, rasgos faciales expuestos o la geometría de una cabeza descubierta. Esta transformación conceptual convierte el problema en una tarea de detección de características específicas, lo cual puede presentar ventajas o desventajas dependiendo de la uniformidad y consistencia de dichas características alternativas (Szeliski, 2022).

2.1.17 Modelo de visión por computadora

Un modelo de visión por computadora es una representación computacional entrenada para interpretar datos visuales como imágenes o videos. Estos modelos utilizan técnicas de aprendizaje automático, especialmente redes neuronales convolucionales (CNN), para extraer características relevantes y realizar tareas como clasificación, detección de objetos o segmentación.

Durante su entrenamiento, el modelo aprende a asociar patrones visuales con categorías específicas, lo que le permite generalizar y realizar predicciones en datos nuevos. En esta investigación, el modelo está diseñado para detectar si un motociclista usa o no casco, a partir del análisis automático de imágenes captadas en la vía pública.

“Los modelos de visión por computadora permiten extraer información estructurada a partir de datos visuales, lo cual los hace esenciales en aplicaciones como la seguridad, la medicina, el transporte y la industria” (Szeliski, 2022, p. 5)

2.1.18 Aprendizaje por transferencia (Transfer Learning)

El aprendizaje por transferencia es una técnica del aprendizaje automático que consiste en utilizar un modelo previamente entrenado en una tarea grande y general (como la detección de objetos en imágenes comunes), y adaptarlo a una tarea nueva pero relacionada, con menos datos.

Esta técnica es útil cuando no se dispone de grandes volúmenes de datos o recursos computacionales para entrenar un modelo desde cero. El modelo conserva el conocimiento aprendido en las primeras capas (características generales como bordes, formas), y se reentrena solo en las últimas capas para reconocer clases específicas del nuevo problema.

En esta investigación, se utiliza transfer learning para adaptar un modelo YOLOv11m previamente entrenado en el conjunto COCO a la tarea de detectar el uso del casco en motociclistas en contextos locales. Esto reduce el tiempo de entrenamiento y mejora el rendimiento del modelo con pocos datos.

“El aprendizaje por transferencia permite reutilizar conocimientos de una red ya entrenada para nuevas tareas, acelerando el proceso y mejorando resultados incluso con datos limitados” (Chollet, 2018).

2.1.19 Yolo (you only look once)

Se trata de una arquitectura de detección de objetos muy rápida y precisa, que fue creada por Joseph Redmond como principal autor. El hecho de que sea una arquitectura muy rápida, la hace idónea para que se use en la detección en video en tiempo real. YOLO consiste en una red neuronal convolucional que predice simultáneamente múltiples cuadros delimitadores y las

probabilidades de la clase de objeto que delimitan dichos cuadros delimitadores (Redmon et al., 2016).

2.1.20 Beneficio

Es un sistema muy rápido, debido a que se reduce la detección a un problema de regresión lineal y esto conlleva a que no se requiera un (conjunto de pasos interconectados para procesar y modelar datos de manera automatizada) complejo. Simplemente se ejecuta la red neuronal en una imagen en la prueba para hacer predicciones. Para hacer las predicciones, a diferencia de los métodos anteriores, este sistema consume la imagen completa, en lugar de regiones de la misma. Esto hace que se limiten los errores a la hora de reconocer las clases de objetos que hay en la imagen. Aprende representaciones generalizables de objetos, lo que hace que, si se introducen datos de entrada nuevos, tenga menos probabilidades de fallo que las técnicas descritas anteriormente (Redmon et al., 2016)

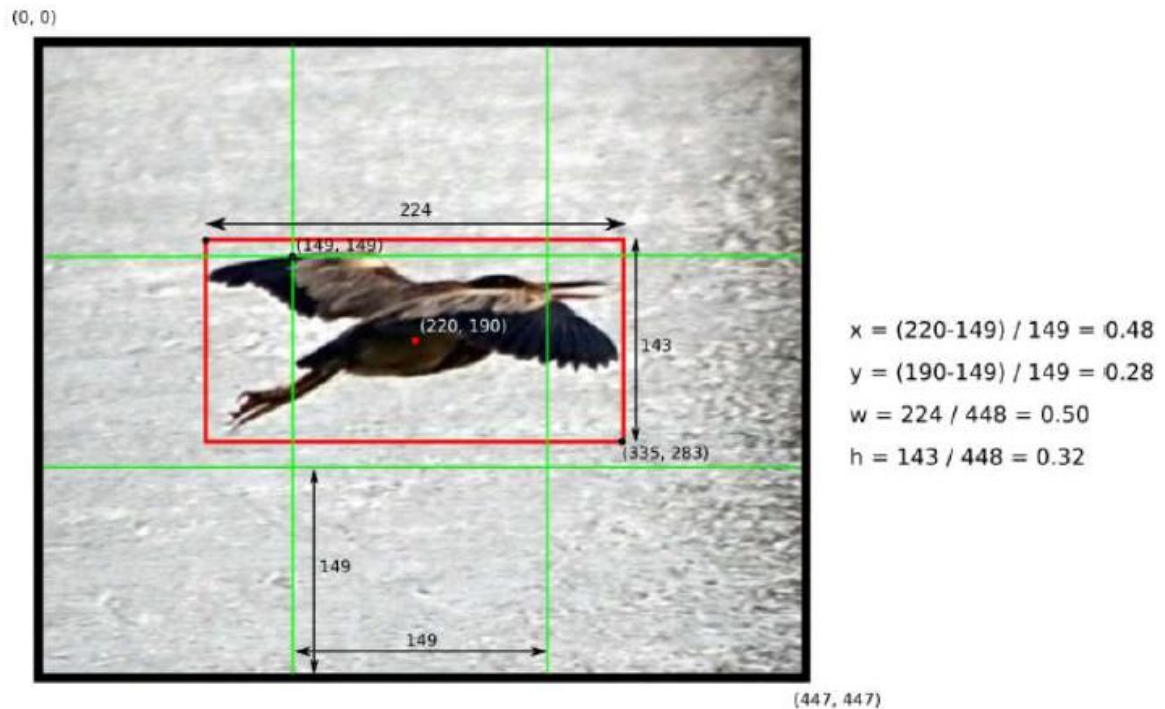
2.1.21 Funcionamiento

El sistema funciona a través de una red neuronal convolucional que utiliza características de la imagen completa para generar los cuadros delimitadores (Bounding Box), permitiendo que la red detecte todos los objetos presentes en la imagen. La imagen se divide en una cuadrícula de tamaño $S \times S$, donde cada celda es responsable de detectar los objetos que caen dentro de ella, tal como se muestra en la Figura 2.4, donde se señala la casilla central. Cada celda predice B cuadros delimitadores junto con sus respectivas puntuaciones de confianza (score), que indican tanto la fiabilidad del modelo al detectar un objeto en esa celda como la precisión de la predicción.

La confianza se define formalmente como $\Pr(\text{Objeto})$ multiplicado por IOU, como se puede observar en la Figura 2.5. Si una celda no contiene un objeto, su puntuación de confianza debe ser cero; de lo contrario, la puntuación debe equivaler a la intersección sobre unión (IOU). Más adelante se explicará esto en mayor detalle. Cada cuadro delimitador incluye cinco predicciones: x , y , w , h y la confianza. Las coordenadas (x, y) representan el centro del cuadro en relación con los límites de la celda de la cuadrícula, ejemplificado en la Figura 2.4 por un punto rojo con coordenadas $(220, 190)$. Estas coordenadas se normalizan con respecto al tamaño de la celda, que en el ejemplo es de 149×149 , por lo que los valores resultantes oscilan entre 0 y 1, como se muestra en los cálculos; el mismo principio se aplica para la coordenada y .

En cuanto al ancho y alto, estos se normalizan respecto al tamaño completo de la imagen, que en este caso es 448×448, tomando para ello el ancho y la altura del objeto detectado (cuadro rojo en la Figura 2.4).

Figura II.4: Ejemplo de calcular coordenadas en una imagen



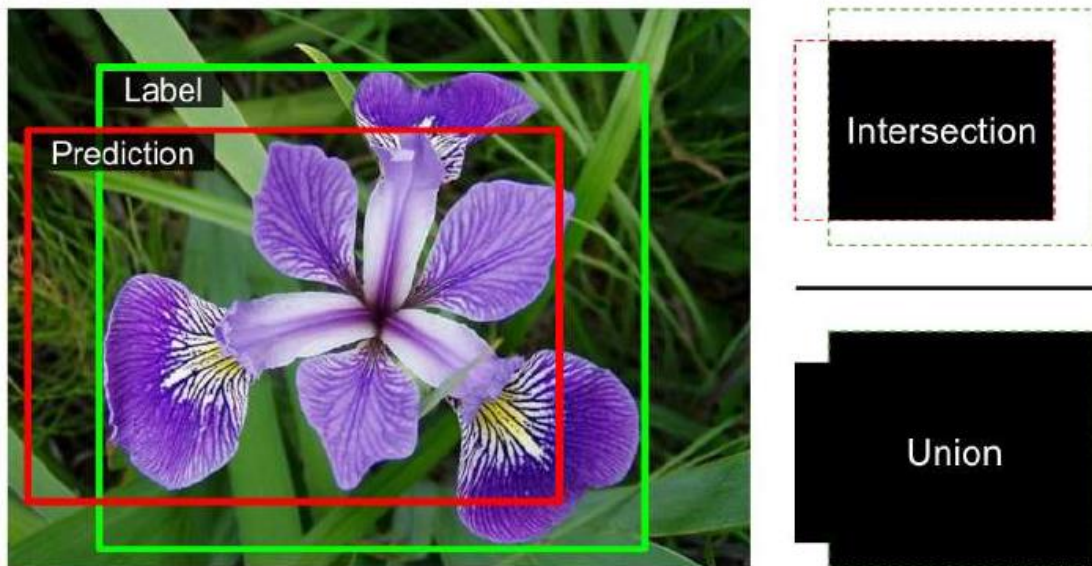
Nota. Adaptado de *Estudio de la arquitectura YOLO para la detección de objetos mediante deep learning*, por D. S. Rozada Raneros, 2021.

Finalmente, para obtener la puntuación de confianza de la detección, necesitamos varios parámetros. En primer lugar, se necesita la intersección sobre la unión (IOU) del cuadro delimitador y la anotación realizada sobre la imagen (Groundtruth), que podemos ver en la Figura 2.5.

Calculamos la IOU como el cociente entre la intersección y la unión. Esto nos da como resultado la intersección relativa entre ambos cuadros delimitadores, lo cual es fundamental para evaluar la precisión de la detección.

Figura II.5

Descripción gráfica de la intersección sobre la unión



Nota. Adaptado de *Estudio de la arquitectura YOLO para la detección de objetos mediante deep learning*, por D. S. Rozada Raneros, 2021.

Posteriormente, la puntuación de confianza se calcula considerando varias probabilidades. Cada celda de la cuadrícula predice C probabilidades condicionales de clases, $\Pr(\text{Class} | \text{Object})$, que están condicionadas a que la celda contenga un objeto. Además, la predicción se realiza para un solo objeto, independientemente del número de cuadros delimitadores B . En la detección, se multiplica la probabilidad condicional de la clase por la probabilidad del cuadro individual, obteniendo así las puntuaciones de confianza específicas para cada cuadro delimitador. Estas puntuaciones indican la probabilidad de que el objeto esté contenido en el cuadro y qué tan bien se ajusta el cuadro predicho al objeto (Rozada Raneros, 2021).

Figura II.6: Obtención de la puntuación de confianza del objeto

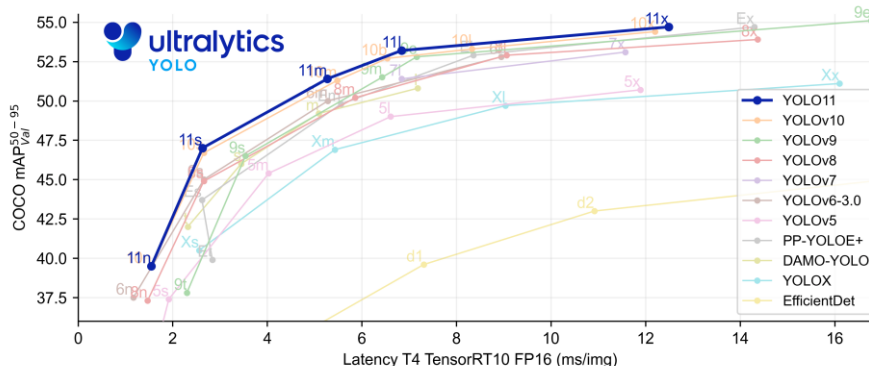
$$\Pr(\text{Class}_i | \text{Object}) * \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

Nota. Adaptado de *Estudio de la arquitectura YOLO para la detección de objetos mediante deep learning*, por D. S. Rozada Raneros, 2021.

2.1.22 Algoritmo de Yolov11m

YOLOv11 es una de las últimas iteraciones de la serie Ultralytics, que redefine lo que es posible con una precisión, velocidad y eficacia de vanguardia. La selección de un modelo de detección de objetos a menudo implica un compromiso fundamental entre la precisión alcanzada y la velocidad de procesamiento. Para ilustrar este compromiso en arquitecturas recientes, en la (Figura 2.7) se compara el rendimiento de diversos modelos, permitiendo comparar la eficiencia de modelos como YOLOv11 frente a versiones anteriores y otros modelos, observándose la posición de YOLOv11 en relación con sus predecesores y otros modelos. Basándose en los impresionantes avances de las versiones anteriores, YOLO11 introduce mejoras significativas en la arquitectura y los métodos de entrenamiento, lo que lo convierte en una opción versátil para una amplia gama de tareas de visión de computadoras.

Figura II.7: Rendimiento de las arquitecturas Yolo



Nota. Adaptado de *Documentación de Ultralytics YOLO* [Grafica], por Ultralytics, 2024, Flickr (<https://docs.ultralytics.com/es/models/yolo11/>).

Entre las principales características encontramos:

Extracción de características mejoradas: Emplea una arquitectura mejorada de columna vertebral y cuello, que mejora las capacidades de extracción de características. Optimizado para la eficiencia y velocidad: Introduce diseños arquitectónicos refinados y canales de formación optimizados, lo que proporciona velocidades de procesamiento más rápidas y mantiene un equilibrio óptimo entre precisión y rendimiento.

Adaptabilidad en distintos entornos: Puede implementarse sin problemas en distintos entornos, incluidos dispositivos periféricos, plataformas en la nube y sistemas compatibles con GPU NVIDIA, lo que garantiza la máxima flexibilidad. Amplia gama de tareas soportadas: Ya se trate de la detección de objetos, la segmentación de instancias, la clasificación de imágenes, la estimación de poses o la detección de objetos orientada, YOLO11 está diseñado para atender a un conjunto diverso de retos de visión por ordenador.

Existen diferentes variantes dentro de la familia YOLOv11, la elección de esta depende del balance deseado entre precisión, velocidad y recursos. La (Figura 2.8) detalla las métricas para cada tamaño del modelo, identificado por una escala, comenzando por ‘n’(nano), ‘s’(small), ‘m’(medium), ‘l’(large) y ‘x’(extra-large). Se observa cómo incrementos en precisión (mAP) y complejidad (parámetros, FLOPs) al pasar de las versiones más pequeñas (‘n’, ‘s’) a las más grandes (‘l’, ‘x’) conllevan un aumento en la latencia de inferencia, tanto en CPU como en GPU (T4 TensorRT). Estos resultados se obtuvieron al entrenar las variantes con el conjunto de datos COCO, con más de 330k de imágenes y 80 categorías de objetos (Ultralytics, 2024).

Figura II.8: *Variantes de la Arquitectura YOLOv11*

Modelo	tamaño (píxeles)	mAP ^{val} ₅₀₋₉₅	Velocidad CPU ONNX (ms)	Velocidad T4 TensorRT10 (ms)	parámetros (M)	FLOPs (B)
YOLO11n	640	39.5	56.1 ± 0.8	1.5 ± 0.0	2.6	6.5
YOLO11s	640	47.0	90.0 ± 1.2	2.5 ± 0.0	9.4	21.5
YOLO11m	640	51.5	183.2 ± 2.0	4.7 ± 0.1	20.1	68.0
YOLO11l	640	53.4	238.6 ± 1.4	6.2 ± 0.1	25.3	86.9
YOLO11x	640	54.7	462.8 ± 6.7	11.3 ± 0.2	56.9	194.9

Nota. Adaptado de *Documentación de Ultralytics YOLO* [Tabla], por Ultralytics, 2024, Flickr (<https://docs.ultralytics.com/es/models/yolo11/>).

Teniendo en cuenta estos resultados, la decisión dependerá de las necesidades y recursos para desarrollar el proyecto (Ultralytics, 2024).

2.1.23 Métricas de rendimiento

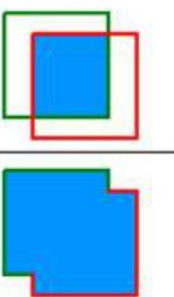
En realidad, no solo vale con que el sistema de detección detecte los objetos que nosotros deseamos, sino que debemos de saber cuál es el rendimiento de dicho sistema de detección de

objetos. Para ello debemos evaluarlo mediante una serie de métricas que nos permitan obtener un análisis claro de si el sistema funciona bien o funciona mal.

Una técnica muy interesante para evaluar las prestaciones es la validación cruzada. Consiste en dividir el conjunto de entrenamiento en conjuntos más pequeños de entrenamiento y validación, para luego entrenar el modelo con cada conjunto de entrenamiento que se acaba de conseguir y a continuación aplicarlo al conjunto de validación para evaluar el error cometido en el mismo. Este método nos permite obtener una estimación del rendimiento del modelo (promediando los errores de validación obtenidos), pero también nos da cuánto de precisa es la estimación. Para entender cómo funcionan las métricas y su funcionamiento, debemos de realizar algunas definiciones importantes.

- **Intersección sobre la unión (IOU, Intersection over Union):** En este caso es la medida que evalúa el porcentaje de superposición de dos cuadros delimitadores (Bounding box). Concretamente tenemos 2 tipos de cuadros delimitadores, los Groundbox, que son las anotaciones que se han realizado sobre las imágenes y que permiten poder realizar el entrenamiento y validación en el sistema de detección de objetos, y por otro tenemos los cuadros que se obtienen de la detección que ha realizado el sistema. Esto lo podemos ver en la Figura 2.9. IOU será 1, si coinciden tanto la detección como los cuadros delimitadores de las anotaciones, mientras que cuanto menor sea el valor, más alejado estará de una detección fiable

Figura II.9: Explicación gráfica de la intersección sobre la unión

$$IOU = \frac{\text{area of overlap}}{\text{area of union}} = \frac{\text{área de superposición}}{\text{área de unión}}$$


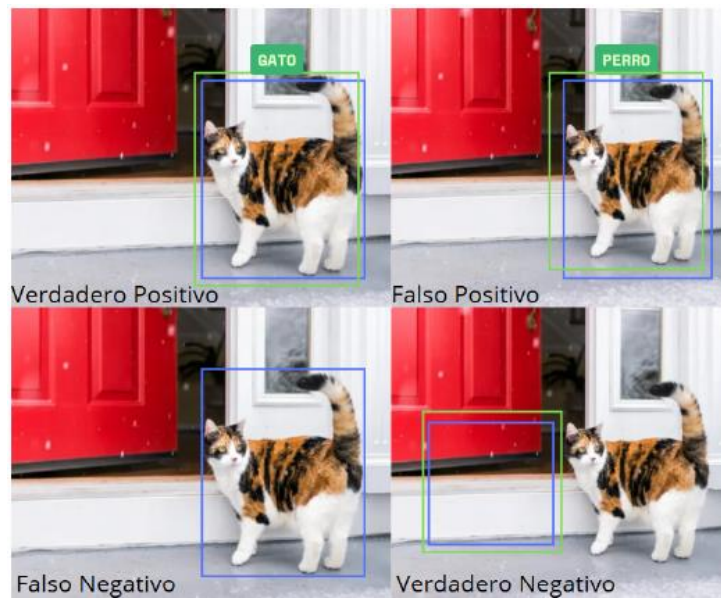
Nota. Adaptado de *Estudio de la arquitectura YOLO para la detección de objetos mediante deep learning*, por D. S. Rozada Raneros, 2021.

Concretamente, para conocer si una detección es o no buena tenemos los siguientes conceptos.

- **Verdadero Positivo (True Positive, TP):** Se produce cuando se ha detectado el objeto y $IOU \geq \text{Umbral}$. Es una detección correcta.
- **Falso Positivo (False Positive, FP):** Se produce cuando el sistema informático detecta el objeto, pero al calcular el IOU se obtiene que $IOU < \text{Umbral}$. Se trata por tanto de una detección incorrecta.
- **Falso Negativo (False Negative, FN):** Es cuando existe un groundbox en la imagen, pero no se detecta.
- **Verdadero Negativo (True Negative, TN):** No se utiliza puesto que serían todos los cuadros delimitadores posibles que no contienen y en los que no se detectan objetos. En detección de objetos hay una infinidad de cuadros delimitadores posibles que no se deben detectar en una imagen.
- **Umbral:** El que se establece para que se detecte un objeto. Generalmente 50%, 75%, o 95%.

A continuación, en la Figura 2.10 se observa el ejemplo visual.

Figura II.10: Representación de VP, FP, FN, VN



Nota. Adaptado de *sistema de monitorización de tráfico en directo mediante detección de objetos basada en deep learning (p. 15)*, por F. E. García Borges, 2025.

Teniendo en cuenta lo anterior, el sistema debe evaluarse con métricas precisas para medir su rendimiento (Rozada Raneros, 2021).

2.1.24 Umbral de confianza

El umbral de confianza es el valor mínimo que debe alcanzar la puntuación de confianza de una detección para que esta sea aceptada como válida por el modelo (Redmon et al., 2016). En este estudio se implementó el valor estándar de 0.25, que corresponde a la configuración predeterminada en YOLOv11m establecida por Ultralytics (2024). Este umbral representa un equilibrio validado empíricamente entre precisión y recuperación en tareas de detección de objetos, minimizando tanto los falsos positivos como los falsos negativos sin comprometer la capacidad del modelo para detectar objetos en condiciones desafiantes como oclusión parcial, variaciones de iluminación y ángulos complejos.

Diversos estudios recientes respaldan el uso de este umbral estándar. Papacostas et al. (2024) utilizaron un umbral de 0.25 en su investigación sobre detección automática de objetos con YOLOv8, mientras que Song et al. (2025) aplicaron el mismo valor en su algoritmo para detección de cascos de seguridad, demostrando resultados robustos en escenarios reales. Mantener el valor predeterminado garantiza la comparabilidad con otros estudios y preserva el rigor metodológico sin introducir ajustes arbitrarios que podrían comprometer la generalización del modelo (Ultralytics, 2024).

Asimismo, es importante considerar que el umbral de confianza debe ajustarse al tipo de aplicación: en sistemas de seguridad vial como el presente, mantenerlo en 0.25 permite que el modelo detecte motociclistas aun en situaciones de oclusión, movimiento o baja visibilidad, evitando omitir infracciones genuinas por exigir una certeza excesiva. Sin embargo, para asegurar la justicia y confiabilidad del proceso, toda detección debe pasar por un mecanismo de revisión adicional ya sea humana o automatizada antes de emitir una sanción, evitando errores derivados del umbral bajo y garantizando la validez de las infracciones reportadas. Esta estrategia se alinea con las recomendaciones internacionales y ofrece un balance óptimo entre sensibilidad del sistema y rigurosidad en la aplicación de medidas correctivas.

2.1.25 Matriz de confusión

Las métricas que acabamos de definir suelen representarse en la denominada matriz de confusión. Se trata de una matriz cuadrada que se usa para evaluar el rendimiento de un modelo, normalmente de clasificación. En ella se compara los valores predichos por el sistema con los reales. Esto nos da una forma más de ver cómo de bien está funcionando nuestro modelo. Para poder interpretarla nos podemos fijar en la Figura 2.11.

Figura II.11: Ejemplo de Matriz de confusión

		Real Values	
		Positive	Negative
Predicted Values	Positive	TP = True Positive	FP = False Negative
	Negative	FN = False negative	TN = True Negative

Nota. Adaptado de *Estudio de la arquitectura YOLO para la detección de objetos mediante deep learning*, por D. S. Rozada Raneros, 2021.

Si nos fijamos, encontramos que la variable de destino tiene dos posibles valores, bien positivo o negativo. Por otra parte, los valores reales se representan en la columna, mientras que en las filas se representan los valores predichos. Si por ejemplo tenemos un sistema en el que tenemos 1000 datos y en el que se obtiene una matriz de confusión con los siguientes valores: TP=600, TN=300, FN=50, FP=50. Tenemos que es un sistema bastante con una tasa de acierto del 90% (pues se identifican correctamente 900 de los 1000 datos) pero también nos permite ver qué tipos de errores comete, pues según la aplicación puede tener más relevancia un tipo de errores que otros (Bhandari, 2020; Gerón, 2019).

2.1.26 Precision y Recall

La precisión nos da una medida de la calidad del modelo propuesto. Indica, de todas las cajas que devuelve el sistema informático (es decir, de todos los objetos que encuentra), qué fracción de ellas realmente contienen el objeto a detectar. Su valor ideal sería la unidad (Padilla et al., 2020). En la Figura 2.12 vemos cómo lo podemos calcular.

Figura II.12: Definición matemática de Precisión

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{\text{TP}}{\text{all detections}}$$

Nota. Adaptado de *A survey on performance metrics for object detection algorithms*, por R. Padilla, S. L. Netto y E. A. B. da Silva, 2020.

El Recall o sensibilidad, indica qué fracción de objetos, de los que verdaderamente existen, encuentra el detector. Si el sistema es capaz de encontrar todos los casos relevantes, es decir, que encuentre todos los ground truth, que son todas las anotaciones que hay en una imagen, diremos que es un sistema muy sensible (Padilla et al., 2020). En la Figura 2.13 vemos cómo se calcula.

Figura II.13: Definición matemática de Recall

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{TP}}{\text{all ground truths}}$$

Nota. Adaptado de *A survey on performance metrics for object detection algorithms*, por R. Padilla, S. L. Netto y E. A. B. da Silva, 2020.

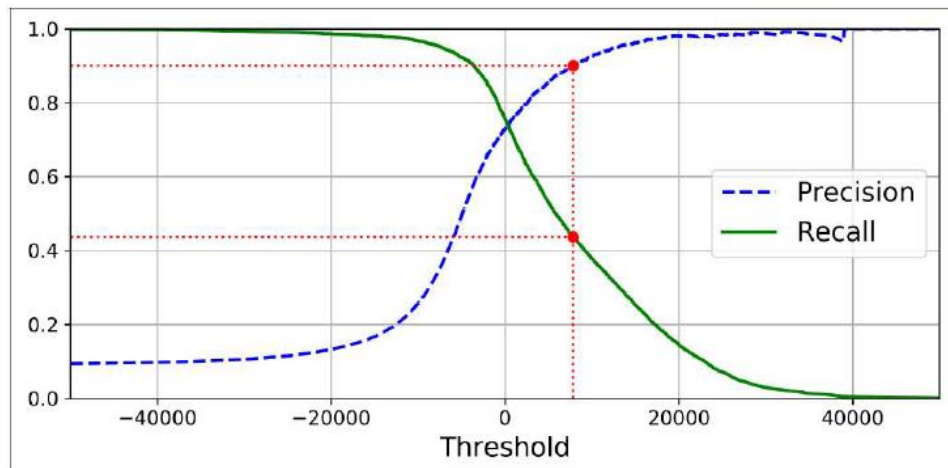
2.1.27 Curva de Precision-Recall

Los dos conceptos descritos anteriormente están estrechamente relacionados entre sí. Si nosotros configuramos el sistema para obtener una mayor precisión, tendremos en general como resultado una disminución del Recall, es decir el sistema será menos sensible y por tanto se detectarán menos objetos.

Por otra parte, si configuramos el sistema para tener más sensibilidad, vemos que el sistema será menos preciso puesto que a la hora de realizar detecciones, el sistema es más laxo a la hora de decidir.

Todo esto se gestiona a partir del umbral de puntuación (Threshold score). En la Figura 2.14 vemos gráficamente lo que sucede, si aumentamos el umbral, vemos como decae la sensibilidad, pero aumenta la precisión. Por otro lado, si lo bajamos, tendremos en general el caso contrario, mayor sensibilidad y peor precisión (Gerón, 2019; Ramírez, 2020).

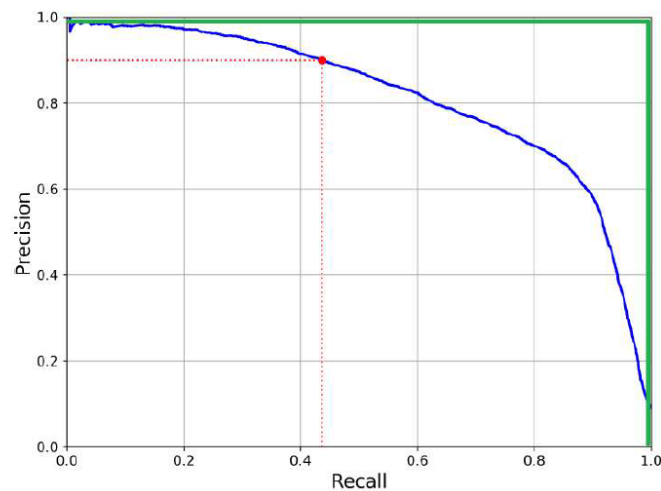
Figura II.14: Curvas de precisión- recall



Nota. Adaptado de *Curvas PR y ROC*, por J. Ramírez, 2020.

Un detector lo consideramos bueno, si para cuando va aumentando el Recall la precisión se va manteniendo alta. Esto lo podemos ver en la Figura 2.15, en la cual encontramos un caso de un detector ideal y otro de un detector real. Vemos que la línea verde es la línea ideal de cómo se debería de comportar un sistema perfecto, mientras que la línea azul representa un ejemplo de un sistema real (Gerón, 2019; Ramírez, 2020).

Figura II.15: Ejemplo de una curva perfecta



Nota. Adaptado de *Curvas PR y ROC*, por J. Ramírez, 2020.

2.1.28 Precisión media (mAP)

Es otra de las formas de encontrar un buen o mal detector de objetos es utilizar la precisión media o mAP. Se trata de encontrar el área bajo la curva de Precision-Recall, de esta forma encontramos una medida numérica que facilita la tarea de saber si es bueno un detector o no (Gerón, 2019).

El proceso de cálculo sigue estos pasos:

1. Para cada clase, se calcula su Average Precision (AP), que corresponde al área bajo la curva Precisión-Recall.
2. Se promedian los valores de AP de todas las clases, obteniendo el mAP.

La notación mAP@X indica el umbral de IoU utilizado. Por ejemplo:

- **mAP@0.5:** Considera una detección como correcta si $\text{IoU} \geq 0.5$ (umbral estándar en detección de objetos).
- **mAP@0.75:** Umbral más estricto que requiere $\text{IoU} \geq 0.75$.
- **mAP@0.5:0.95:** Promedia los valores de mAP calculados con umbrales desde 0.5 hasta 0.95 en incrementos de 0.05, proporcionando evaluación más rigurosa del modelo (estándar COCO).

En esta investigación se utilizó mAP@0.5 como métrica principal por ser el estándar más ampliamente adoptado en la literatura de detección de objetos y permitir comparabilidad con estudios previos. Un mAP@0.5 de 0.90 (90%) indica que, en promedio, el modelo alcanza 90% de precisión considerando todas las clases cuando se aplica un umbral de IoU de 0.5.

2.1.29 Seguridad vial y uso del casco en motociclistas

La seguridad vial agrupa políticas, normas, acciones y tecnologías destinadas a prevenir accidentes y minimizar sus consecuencias (OMS, 2023a; OMS, 2023b). Es evidente que el uso correcto de cascos reduce el riesgo de muerte por traumatismo craneal entre seis y siete veces, y disminuye lesiones cerebrales graves hasta en 74 % (OMS, 2023a; OMS, 2023b).

La OMS aboga por leyes universales que incluyan a conductores y pasajeros, estándares de calidad, fiscalización efectiva, educación y evaluación constante (OMS, 2023b).

Bolivia cuenta con regulación clara al respecto: la Ley N.º 3988 (2008) y su reglamento, Resolución Suprema N.º 187444 (1978), establecen la obligatoriedad del uso de casco homologado y correctamente ajustado para motociclistas. Estas normas siguen vigentes.

Además, iniciativas como la Declaración de Marrakech (2025) y la Década de Acción para la Seguridad Vial 2021-2030 de la OMS refuerzan el compromiso global de reducir a la mitad las muertes viales, intensificando los esfuerzos en uso del casco (OMS, 2025; OMS, 2023a).

2.1.30 Lenguaje de Programación: Python

Para esta investigación se utiliza el lenguaje de programación Python por su sintaxis sencilla, su amplia comunidad de usuarios y su ecosistema de librerías especializadas en inteligencia artificial, aprendizaje profundo y procesamiento de imágenes (Chollet, 2018). Además, Python cuenta con bibliotecas específicas como OpenCV, NumPy, Pandas, PyTorch y TensorFlow, que facilitan el desarrollo de modelos de visión artificial y detección de objetos.

2.1.31 Entorno de desarrollo: Kaggle

El desarrollo y entrenamiento del modelo se efectúa en Kaggle, una plataforma en la nube que provee acceso a recursos de GPU gratuitos y facilita el manejo de grandes conjuntos de datos para proyectos de aprendizaje automático. Esta herramienta permite a los usuarios ejecutar notebooks de Python con frameworks populares y colaborar en comunidad, sin necesidad de hardware local avanzado (Ultralytics, 2025).

2.1.32 Entorno de prueba: Google Colab

La validación del modelo se lleva a cabo en Google Colab, una plataforma gratuita basada en la nube que permite la ejecución de notebooks de Python con acceso a unidades de procesamiento gráfico (GPU). Esta herramienta facilita el entrenamiento de modelos sin la necesidad de hardware local avanzado (Bisong, 2019).

2.1.33 Conjunto de Datos (Dataset)

El dataset utilizado estará compuesto por imágenes de motociclistas, tanto con casco como sin él. Se recurrirá a fuentes públicas (como bases de datos open-source) y eventualmente a la recolección propia de imágenes. Las imágenes serán anotadas manualmente para delimitar las

regiones de interés (bounding boxes). No se aplicarán técnicas de aumento de datos (data augmentation), ya que el enfoque del proyecto se limita a la incorporación directa de las imágenes en la base de datos para su uso en el entrenamiento inicial del modelo.

2.1.34 Modelo no probabilístico de selección de datos

La metodología de esta investigación se basa en un modelo no probabilístico, en el cual las imágenes que conforman el conjunto de datos no se seleccionan al azar, sino que se eligen de forma intencionada y basada en criterios técnicos (Yin & Liu, 2021). La selección busca representar visualmente distintos escenarios comunes del tránsito urbano, tales como motociclistas con y sin casco en diversas condiciones de iluminación, ángulos de cámara y tipos de casco.

Esta estrategia responde a la naturaleza del problema abordado y al enfoque computacional de la investigación, donde la unidad de análisis no es una persona, sino la imagen en sí misma (Szeliski, 2022). En contextos de visión por computadora, el uso de un conjunto de datos balanceado y diverso, aunque no aleatorio, es suficiente para el entrenamiento y validación efectiva del modelo (Chollet, 2018).

2.1.35 Justificación tamaño datase y validación del modelo

En lugar de aplicar un muestreo probabilístico basado en la población de motociclistas de la ciudad, este proyecto trabaja con un conjunto de imágenes representativas que serán utilizadas para entrenar y validar un modelo de visión por computadora. La unidad de análisis no es el sujeto, sino la imagen como dato visual.

Se empleará un conjunto de 200 imágenes. Esta cantidad se considera adecuada para validar el modelo de detección, ya que se utilizará una técnica de aprendizaje por transferencia (transfer learning), que permite reutilizar modelos previamente entrenados en grandes bases de datos (como COCO) y adaptarlos a un contexto local con menor cantidad de datos.

Además, estudios previos han demostrado que modelos como YOLOv5 pueden alcanzar altos niveles de precisión (por ejemplo, 94%) utilizando conjuntos de entrenamiento moderados, cuando se aplican buenas prácticas de anotación y evaluación (Mercado Reyna, 2023).

La validación del modelo se realiza mediante el uso de métricas propias del aprendizaje automático, como Precisión, Recall, y mAP (mean Average Precision), que permiten cuantificar el desempeño del sistema en la tarea específica de detección de uso del casco.

2.2 MARCO CONTEXTUAL

2.2.1 Introducción al Problema de Seguridad Vial en Bolivia

La seguridad vial constituye un desafío multidimensional de salud pública en el Estado Plurinacional de Bolivia, donde los accidentes de tránsito representan una de las principales causas de morbilidad, generando no solo sufrimiento humano sino también costos económicos y sociales significativos para el desarrollo nacional. Dentro de este panorama complejo, los usuarios de motocicletas y ciclomotores emergen como el grupo poblacional de mayor vulnerabilidad, enfrentando riesgos diferenciados que demandan atención prioritaria tanto desde perspectivas normativas como tecnológicas.

El uso inadecuado o la ausencia del casco protector constituye el factor de riesgo modificable más crítico asociado a la severidad de las lesiones en accidentes de tránsito involucrando motociclistas. A pesar de la existencia de marcos normativos claros que establecen su obligatoriedad y de evidencia científica contundente sobre su efectividad preventiva, las tasas de cumplimiento permanecen insatisfactorias en numerosos contextos urbanos bolivianos, evidenciando brechas significativas entre la norma formal y su implementación efectiva en la práctica cotidiana de movilidad urbana.

2.2.2 Magnitud Global y Regional del Problema

A nivel mundial, los accidentes de tránsito representan una de las principales causas de muerte, y una proporción significativa de estas fatalidades involucra a conductores de motocicletas. Según la Organización Mundial de la Salud (2024), "los motociclistas y otros conductores de vehículos de motor de dos o tres ruedas representan el 30% de los fallecidos" (p. 3). La evidencia internacional establece que el uso correcto de cascos de calidad puede reducir el riesgo de muerte en más de seis veces y disminuir el riesgo de traumatismos craneoencefálicos hasta en 74% (OMS, 2024). Estos datos posicionan el casco protector como una de las intervenciones de prevención de lesiones de mayor costo-efectividad disponibles, comparable en su impacto potencial a otras medidas preventivas consolidadas en salud pública.

2.2.3 Situación Nacional y local: Bolivia en la Seguridad Vial

Bolivia, como país que sigue enfrentando una situación crítica de inseguridad vial que anualmente cobra miles de vidas, genera decenas de miles de heridos y provoca discapacidades permanentes directamente atribuibles a hechos de tránsito, presenta una problemática particularmente grave en el segmento de los vehículos automotores menores. Según el informe oficial del Observatorio Boliviano de Seguridad Ciudadana y Lucha Contra las Drogas, en 2022 se registraron 17.761 hechos y accidentes de tránsito a nivel nacional, de los cuales 2.940 casos (16,55%) fueron protagonizados por la categoría “vehículo automotor menor”, que agrupa exclusivamente motocicletas y mototaxis (OBSCD, 2023).

Datos reportados por la Agencia de Noticias Fides documentan que en 2024 la tendencia se agravó: de los 20.942 siniestros nacionales, las motocicletas fueron protagonistas en el 45,8% (9.592 casos), lo que significa un incremento del 226% en solo dos años respecto a la participación de 2022. Este salto posiciona a los motociclistas como el grupo vial más vulnerable del país (OBSCD, 2023; ANF, 2025).

En el departamento de Pando, durante el mismo año 2022, se registraron 547 siniestros viales (3,08% del total nacional), de los cuales 378 correspondieron a vehículos automotores menores, lo que representa un 69,1% del total departamental. Esta proporción evidencia que, pese a su menor parque automotor, las motocicletas generan casi siete de cada diez accidentes en Pando, convirtiéndose en el principal factor de riesgo vial de la región (OBSCD, 2023).

2.2.4 Contexto Específico de la Ciudad de Cobija

La ciudad de Cobija, capital del departamento de Pando, presenta características sociodemográficas, geográficas y de movilidad que configuran un contexto crítico para la seguridad vial de motociclistas. Según el Registro Único para la Administración Tributaria Municipal (RUAT), en 2022 Pando registró 7,629 vehículos, de los cuales 6,169 (80,85%) fueron motocicletas. Para 2024 la cifra escaló a 10,271 vehículos totales y 8,721 motocicletas, alcanzando el 84,9% del parque automotor departamental. Este crecimiento exponencial consolida a la motocicleta como el medio de transporte absolutamente predominante en Cobija y todo Pando (INE, 2024).

Múltiples factores contextuales explican esta preferencia: la configuración urbana compacta de la ciudad que facilita desplazamientos cortos, el clima tropical cálido que favorece el uso de vehículos abiertos, limitaciones en la cobertura y calidad del transporte público formal, y condiciones socioeconómicas que posicionan la motocicleta como la opción de movilidad motorizada más accesible económicamente para amplios sectores poblacionales. Esta democratización del acceso a movilidad motorizada genera, paradójicamente, nuevos riesgos cuando no es acompañada de cultura vial preventiva ni fiscalización efectiva.

En noviembre de 2013, se implementó en Bolivia un estudio de campo observacional urbano sobre uso de casco en motocicletas y ciclomotores, desarrollado mediante colaboración entre la Agencia Nacional de Seguridad Vial de Argentina (ANSV), el Observatorio Iberoamericano de Seguridad Vial (OISEVI), la Policía Boliviana y la Organización Panamericana de la Salud (OPS/OMS). Este estudio representó un esfuerzo pionero para cuantificar objetivamente los comportamientos viales relacionados con el uso de sistemas de protección en contextos urbanos bolivianos (Puppo et al., 2013).

Los datos del estudio observacional de 2013 documentaron que en Cobija solo el 23,9% de los conductores utilizaba casco, posicionando a la ciudad con la segunda tasa de incumplimiento más alta entre las principales ciudades evaluadas (Puppo et al., 2013). Este dato, aunque corresponde a un período específico hace más de una década, proporciona una línea de base empírica valiosa que evidencia la persistencia histórica del problema.

El análisis por sexo en Cobija reveló que las conductoras femeninas presentaban mayor tasa de uso (29.7%) comparado con conductores masculinos (21.3%), patrón consistente con evidencia de otras ciudades que sugiere mayor propensión de mujeres hacia comportamientos preventivos (Puppo et al., 2013). Respecto a grupos etarios, los conductores adultos utilizaban casco con mayor frecuencia (24.3%) que conductores jóvenes (16.1%), aunque ambos valores permanecían sustancialmente por debajo del umbral deseable.

La situación de pasajeros en Cobija reflejó el patrón nacional: solo el 6.4% del primer pasajero utilizaba casco, y el uso en pasajeros adicionales era prácticamente nulo (Puppo et al., 2013). Esta situación es particularmente preocupante considerando las observaciones de campo que documentaron frecuentemente más de dos ocupantes por motocicleta, configuración que incrementa exponencialmente el riesgo en caso de siniestro.

En el período reciente, aunque no se dispone de estudios observacionales sistemáticos actualizados que repliquen la metodología de 2013, reportes de autoridades locales de tránsito indican que la problemática persiste. Las autoridades han implementado planes especiales de control y campañas de concienciación; sin embargo, la efectividad de estas intervenciones no ha sido cuantificada mediante estudios rigurosos que permitan evaluar cambios en las tasas de cumplimiento. Esta ausencia de datos actualizados constituye, en sí misma, una limitación significativa para el diseño de políticas basadas en evidencia.

2.2.5 Limitaciones y Mecanismos de Fiscalización

La fiscalización del cumplimiento del uso obligatorio del casco en Cobija se realiza mediante métodos tradicionales que enfrentan limitaciones estructurales significativas. El modelo prevalente consiste en fiscalización manual por efectivos de la Dirección Departamental de Tránsito, Transporte y Seguridad Vial de Pando, quienes realizan controles mediante observación visual directa en puntos específicos de la red vial urbana.

Este enfoque tradicional presenta múltiples restricciones que limitan su efectividad:

- **Cobertura espacial limitada:** El número reducido de efectivos de tránsito imposibilita la vigilancia simultánea de múltiples arterias viales, generando "zonas ciegas" donde el cumplimiento normativo no es fiscalizado. Esta cobertura fragmentada permite que conductores adapten sus rutas para evitar puntos de control conocidos, reduciendo el efecto disuasivo de la fiscalización.
- **Inconsistencia temporal:** La fiscalización manual opera en horarios limitados y días específicos, generando patrones predecibles que facilitan estrategias de evasión. La ausencia de fiscalización continua, elemento crítico para la efectividad de estrategias de control.
- **Subjetividad en la identificación:** La determinación visual de infracciones está sujeta a variabilidad en la percepción y criterio de diferentes fiscalizadores. Factores como velocidad del vehículo, distancia de observación, pueden generar inconsistencias en la identificación de infracciones, potencialmente generando percepciones de arbitrariedad en la aplicación de sanciones.

- **Limitación de recursos:** Ampliar significativamente el número de efectivos dedicados exclusivamente a fiscalización de uso de casco representa un costo económico elevado que compite con otras prioridades presupuestarias institucionales. Además, el enfoque basado exclusivamente en incremento de recursos humanos no aborda eficientemente los problemas estructurales de cobertura y sistematización.
- **Inexistencia de evidencia objetiva:** La fiscalización manual genera registros de infracciones basados en constancia del efectivo fiscalizador, sin respaldo de evidencia visual objetiva (fotografías, videos) que documente inequívocamente la infracción. Esta limitación puede generar controversias en procesos administrativos de apelación y debilita la robustez legal del procedimiento sancionatorio.

Estas limitaciones no son exclusivas de Cobija; constituyen características estructurales de modelos de fiscalización manual que afectan la efectividad de aplicar la norma de tránsito en numerosos contextos similares. La superación de estas restricciones requiere complementar estrategias tradicionales con enfoques tecnológicos que ofrezcan cobertura ampliada, sistematización de datos, y generación de evidencia objetiva.

2.2.6 Brecha de Conocimiento y Oportunidad Tecnológica

A pesar del valioso estudio observacional de 2013 y del reconocimiento institucional de la problemática por autoridades locales, persisten brechas significativas tanto en conocimiento empírico actualizado como en desarrollo de herramientas tecnológicas adaptadas al contexto local.

Ausencia de datos actualizados: No se han realizado estudios observacionales sistemáticos posteriores a 2013 que cuantifiquen si las tasas de uso de casco se han modificado en la última década, período durante el cual se han implementado diversas campañas de concienciación. Esta ausencia de líneas temporales de datos imposibilita evaluar efectividad de intervenciones previas y detectar tendencias evolutivas del comportamiento vial.

Inexistencia de sistemas automatizados locales: No se han desarrollado en Cobija sistemas tecnológicos de detección de objetos del uso de casco adaptados a las características específicas del contexto local. Esta carencia contrasta con la disponibilidad creciente de tecnologías de visión por computadora y aprendizaje profundo que han demostrado efectividad en otros contextos geográficos.

Brecha entre capacidad tecnológica y aplicación práctica: Aunque estudios previos internacionales han demostrado la viabilidad técnica para la detección del casco, ninguno de estos desarrollos se ha adaptado, validado o específicamente para el contexto de Cobija. Esta brecha representa simultáneamente una limitación actual y una oportunidad para investigación aplicada que conecte capacidades tecnológicas disponibles con necesidades locales específicas.

Ausencia de validación en contexto local: Los modelos de visión por computadora reportados en la literatura científica se han validado típicamente con datasets de contextos geográficos diferentes (países asiáticos, europeos o canadienses), donde las características de motocicletas, patrones de uso de casco, tipo de casco, condiciones de infraestructura vial, e incluso características étnico-fenotípicas de poblaciones pueden diferir sustancialmente. Esta falta de validación contextual genera incertidumbre sobre la transferibilidad y efectividad de estos modelos en contextos bolivianos específicos.

2.2.7 Relevancia y Justificación del Presente Trabajo

El presente trabajo busca abordar directamente estas brechas mediante el desarrollo y validación de un modelo de visión por computadora específicamente adaptado y evaluado para el contexto de la ciudad de Cobija. La relevancia de este esfuerzo investigativo se fundamenta en múltiples dimensiones complementarias:

Validación contextual de tecnología: La investigación ejecuta un proceso completo de validación en imágenes locales que representan las condiciones específicas de Cobija (tipos de motocicletas prevalentes, infraestructura vial urbana, características demográficas, condiciones ambientales diurnas tropicales y tipos de casco oficiales). Esta validación contextual busca establecer si estas tecnologías pueden ser técnicamente viables en contextos urbanos de ciudades intermedias bolivianas.

Evaluación de viabilidad técnica: La investigación ejecuta un proceso completo de validación en imágenes locales que representan las condiciones específicas de Cobija (tipos de motocicletas prevalentes, infraestructura vial urbana, características demográficas, condiciones ambientales diurnas tropicales y tipos de casco oficiales). Esta validación contextual busca establecer si estas tecnologías pueden ser técnicamente viables en contextos urbanos de ciudades intermedias bolivianas.

Base técnica para futuras implementaciones: El modelo desarrollado y validado en este estudio puede constituir una base técnica concreta que informe decisiones de inversión tecnológica por parte de autoridades de tránsito. Aunque el presente estudio se limita a una etapa de validación técnica en entorno simulado, busca establecer fundamentos necesarios para fases subsiguientes de implementación operativa con videos, proporcionando especificaciones técnicas, métricas de rendimiento esperadas y lecciones aprendidas del proceso de desarrollo.

Potencial de impacto en salud pública: Aunque el presente estudio no implementa el sistema operativamente ni evalúa su impacto directo sobre comportamientos viales o indicadores epidemiológicos, establece las bases técnicas para intervenciones futuras que, basadas en evidencia de efectividad de sistemas de fiscalización automatizada en otros contextos, tienen potencial de contribuir a incrementar tasas de uso de casco y, consecuentemente, reducir incidencia y severidad de lesiones craneoencefálicas en accidentes de motocicletas.

En síntesis, el presente trabajo se posiciona en la intersección entre una problemática social documentada históricamente, limitaciones de enfoques tradicionales de fiscalización, disponibilidad de tecnologías emergentes con potencial aplicado, y ausencia de validaciones contextuales locales. Al abordar estas múltiples dimensiones de manera integrada, la investigación contribuye tanto al conocimiento científico sobre aplicabilidad de tecnologías de visión por computadora en contextos específicos, como a la fundamentación empírica de potenciales decisiones de inversión en infraestructura tecnológica para seguridad vial en Cobija, Bolivia.

CAPITULO III.

RESULTADOS Y ANALISIS

3.1 Desarrollo del modelo de visión por computadora

El desarrollo del modelo de visión por computadora se estructuró en una secuencia de etapas coherentes con los objetivos específicos de la investigación, abarcando desde la construcción del conjunto de datos hasta la implementación para su evaluación. En primer lugar, se conformó un conjunto de datos híbrido que combinó un dataset público procedente de la plataforma Roboflow Universe con imágenes locales capturadas en la ciudad de Cobija, equilibrando volumen, diversidad visual y validez contextual para el problema de estudio (véase Anexo E y Anexo F).

Una segunda etapa correspondió al etiquetado manual de las imágenes locales, en la que se estableció la verdad fundamental para tres clases de análisis: motociclista, con casco y sin casco. Este proceso se llevó a cabo siguiendo un protocolo de anotación estandarizado en formato YOLO, garantizando consistencia en la codificación de clases y en las coordenadas de los cuadros delimitadores que describen las regiones de interés (véase Anexo F).

Posteriormente, se desarrolló la fase de entrenamiento del modelo, utilizando la arquitectura YOLOv11m bajo un esquema de aprendizaje por transferencia y manipulando de forma sistemática el número de épocas de entrenamiento para analizar su efecto sobre las métricas de rendimiento. En esta etapa se integraron los componentes de software y hardware necesarios para ejecutar el entrenamiento en un entorno de cómputo en la nube, documentando tanto la configuración experimental como los scripts empleados (véase Anexo H y Anexo I).

Finalmente, se implementaron dos componentes complementarios: por un lado, el procedimiento de evaluación del modelo sobre el conjunto de prueba local, mediante rutinas de validación que generaron métricas como precisión, recall y mAP, y por otro, una interfaz gráfica de ejecución básica que permite cargar imágenes individuales y visualizar las detecciones del modelo de forma interactiva. Ambos elementos materializan la integración funcional del modelo entrenado (véase Anexo I y Anexo J).

3.2 Resultados de evaluación del modelo

A continuación, se presentan los resultados de los diez modelos (A a J), entrenados con 10 a 100 épocas, evaluados sobre 200 imágenes locales de Cobija.

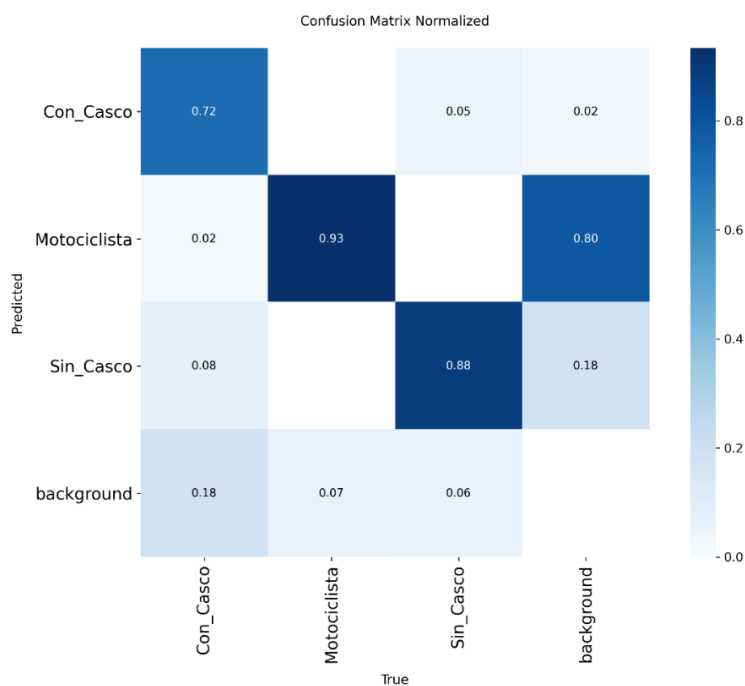
- **Modelo A-10 épocas:**

Tabla III.1: *Resumen modelo A*

Clase	Precisión	Recall	mAP@50
Con casco	0.889	0.62	0.802
Motociclista	0.656	0.902	0.911
Sin casco	0.857	0.865	0.915
Total	0.801	0.796	0.876

Nota. Fuente: Elaboración propia, 2025.

Figura III.1: *Matriz de confusión normalizada modelo A*



Nota. La matriz de confusión normalizada muestra el porcentaje de predicciones correctas e incorrectas para cada clase del modelo A.

El modelo A mostró un comportamiento inicial moderado ($mAP@0.5 = 87.6\%$), destacando una significativa debilidad en la detección de la clase "Con casco" con un recall de 62.0% véase Tabla 3.2. Por otro lado, "Sin casco" y "Motociclista" alcanzaron recalls más aceptables (86.5% y 90.2%, respectivamente), aunque la precisión en "Motociclista" (65.6%) revela la tendencia del modelo a detectar erróneamente otros objetos como motociclistas. En general la Figura 3.2 matriz de confusión evidencia un aprendizaje inicial centrado en patrones visuales más evidentes, pero con criterio conservador para los objetos pequeños como el casco.

- **Modelo B-20 épocas:**

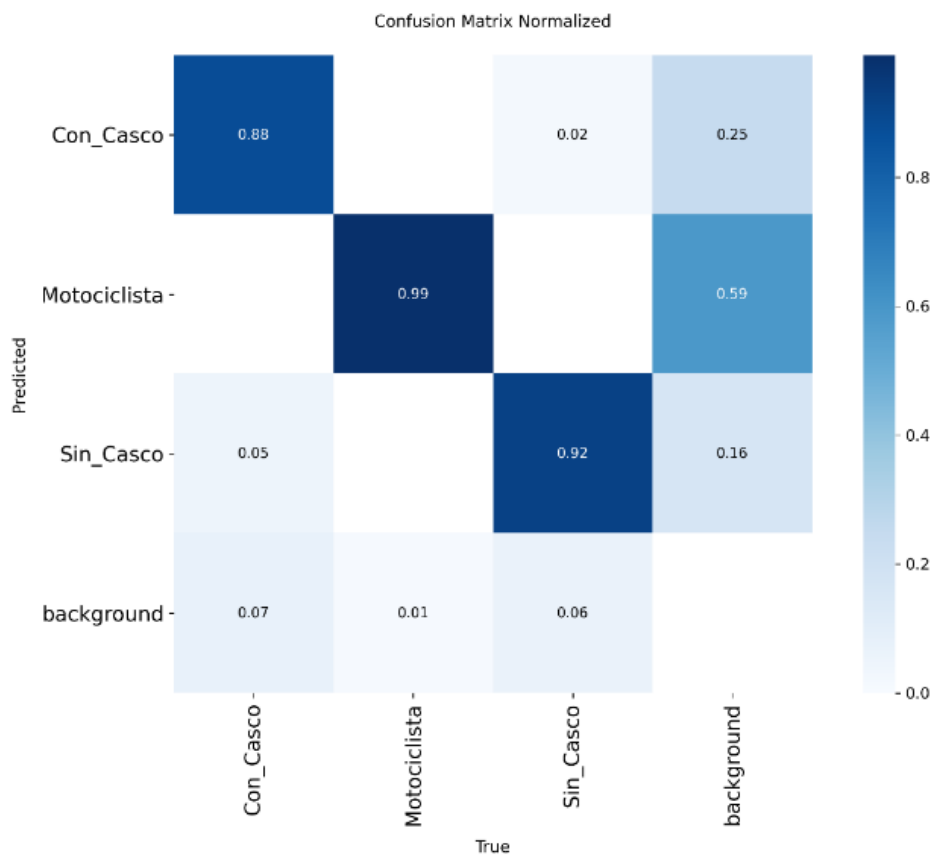
Tabla III.2: *Resumen modelo B*

Clase	Precisión	Recall	$mAP@0.5$
Con casco	0.884	0.86	0.929
Motociclista	0.889	0.976	0.982
Sin casco	0.969	0.866	0.964
Total	0.914	0.9	0.958

Nota. Fuente: Elaboración propia, 2025.

Al aumentar a 20 épocas, se observa una mejora global significativa ($mAP@0.5 = 95.8\%$). Las tres clases muestran un incremento notable en la capacidad de detección: "Con casco" sube su recall a 86.0%, "Sin casco" llega al 86.6% y "Motociclista" supera el 97%. El análisis de la matriz revela que los errores entre "Con casco" y "Sin casco" disminuyen respecto al modelo anterior véase Figura 3.3, aunque persisten casos de confusión entre ambos estados de protección. Este avance evidencia la adquisición progresiva de rasgos distintivos, sobre todo en la clase "Motociclista", que se consolida como la de mejor desempeño véase Figura 3.3.

Figura III.2: *Matriz de confusión normalizada modelo B*



Nota. La matriz de confusión normalizada muestra el porcentaje de predicciones correctas e incorrectas para cada clase del modelo B.

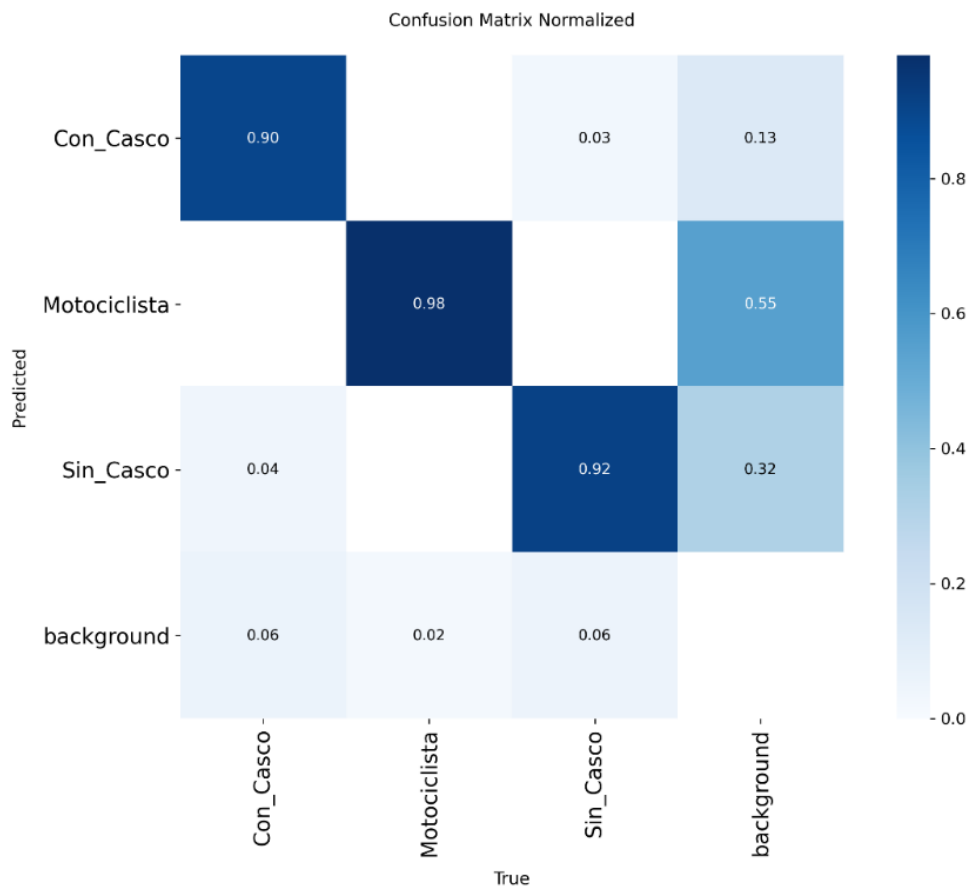
- **Modelo C-30 épocas:**

Tabla III.3: *Resumen modelo C*

Clase	Precisión	Recall	mAP@0.5
Con casco	0.927	0.9	0.954
Motociclista	0.959	0.983	0.982
Sin casco	0.965	0.878	0.965
Total	0.95	0.92	0.967

Nota. Fuente: Elaboración propia, 2025.

Figura III.3: *Matriz de confusión normalizada modelo C*



Nota. La matriz de confusión normalizada muestra el porcentaje de predicciones correctas e incorrectas para cada clase del modelo C.

El modelo C registro la mejora en el reconocimiento de patrones específicos ($mAP@0.5 = 96.7\%$). "Con casco" logra un recall de 90.0%, mientras "Sin casco" alcanza 87.8% y "Motociclista", 98.3% véase Tabla 3.4. La matriz arroja una confusión de 4% Con casco, mostrando un avance relevante en la capacidad discriminativa del modelo. No obstante, la estabilidad en "Motociclista" se convierte en una constante, mientras "Sin casco" aún evidencia falencias menores que podrían atribuirse a la diversidad de perfiles visuales véase Figura 3.4.

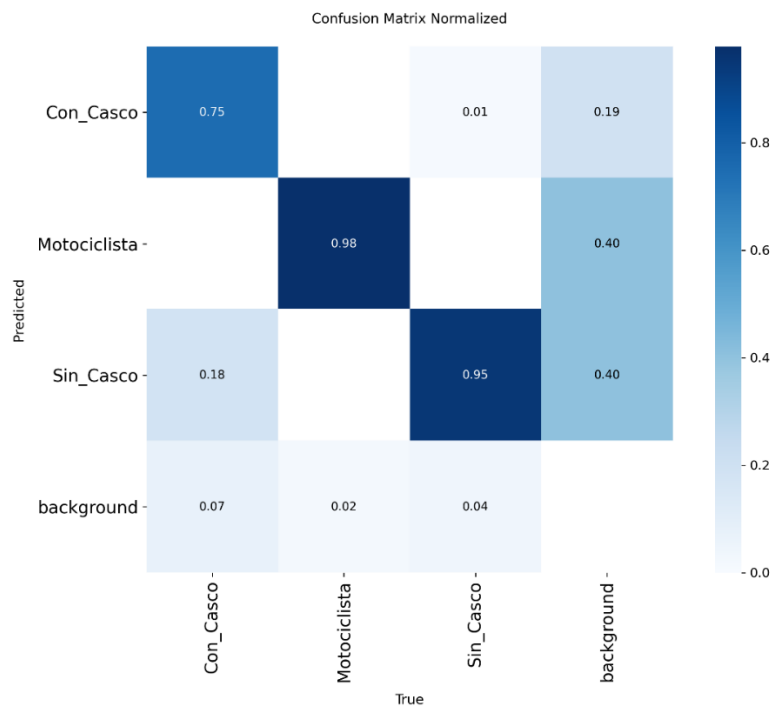
- **Modelo D 40 épocas:**

Tabla III.4: *Resumen modelo D*

Clase	Precisión	Recall	mAP@0.5
Con casco	0.952	0.76	0.902
Motociclista	0.952	0.979	0.98
Sin casco	0.869	0.949	0.954
Total	0.924	0.896	0.945

Nota. Fuente: Elaboración propia, 2025.

Figura III.4: *Matriz de confusión normalizada modelo D*



Nota. La matriz de confusión normalizada muestra el porcentaje de predicciones correctas e incorrectas para cada clase del modelo D.

En la cuarta iteración, aunque el mAP@0.5 permanece alto (94.5%), surge una caída abrupta en la sensibilidad de "Con casco" (recall de 76.0%) véase Tabla 3.5. La confusión de "Sin

casco" hacia "Con casco" alcanza 18%, el nivel más crítico hasta este punto véase Figura 3.5; esta tendencia puede comprometer la labor de fiscalización, al sobreestimar el cumplimiento normativo. Sin embargo, "Motociclista" mantiene su desempeño excelente (97.9% recall), reafirmando la robustez de esta clase ante distintas configuraciones del modelo.

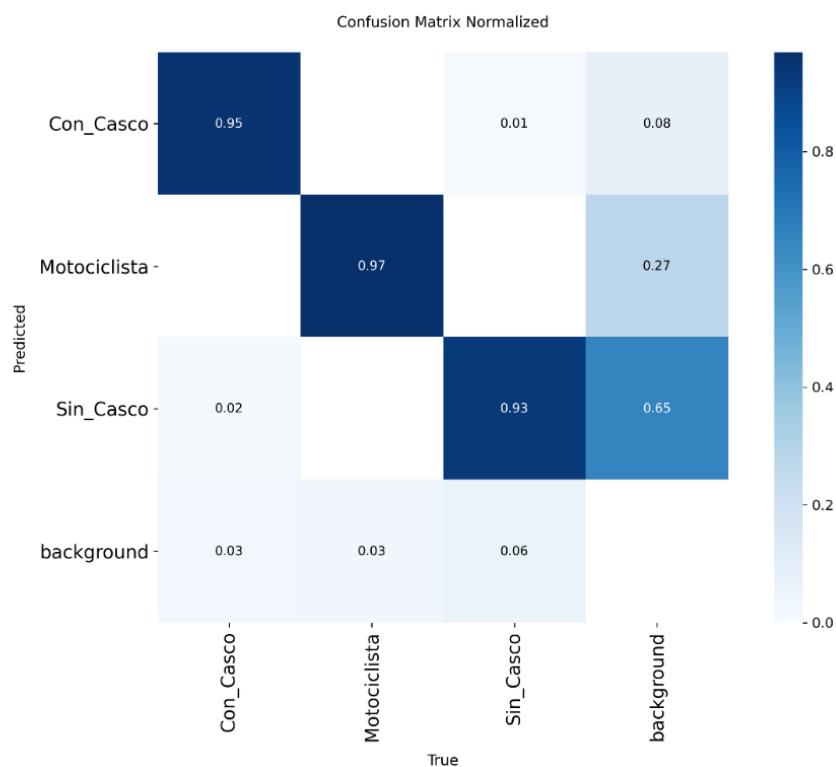
- **Modelo E-50 épocas:**

Tabla III.5: Resumen modelo E

Clase	Precisión	Recall	mAP@0.5
Con casco	0.969	0.95	0.962
Motociclista	0.976	0.969	0.986
Sin casco	0.95	0.908	0.969
Total	0.965	0.942	0.972

Nota. Fuente: Elaboración propia, 2025.

Figura III.5: Matriz de confusión normalizada modelo E



Nota. La matriz de confusión normalizada muestra el porcentaje de predicciones correctas e incorrectas para cada clase del modelo E.

Este modelo registró el valor más alto de la serie ($mAP@0.5 = 97.2\%$). Todas las clases muestran valores sobresalientes: "Con casco" y "Sin casco" superan el 90% en recall y precisión, y "Motociclista" se estabiliza en 96.9% véase Tabla 3.6. Las confusiones críticas en especial entre "Con casco" y "Sin casco" se reducen a valores mínimos (1–2%), lo cual otorga al modelo alto valor de rendimiento véase Figura 3.6. La matriz revela que el modelo distingue fiable y consistentemente entre los tres estados, siendo especialmente eficaz en evitar errores graves de seguridad vial.

- **Modelo F-60 épocas:**

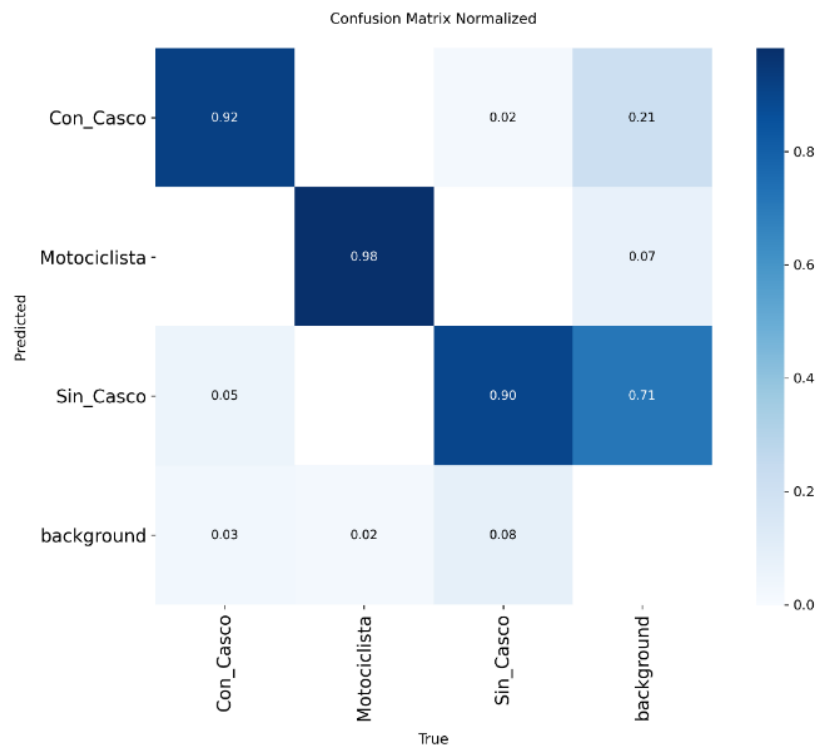
Tabla III.6: *Resumen modelo F*

Clase	Precisión	Recall	$mAP@0.5$
Con casco	0.969	0.937	0.961
Motociclista	0.991	0.983	0.984
Sin casco	0.928	0.904	0.969
Total	0.962	0.941	0.971

Nota. Fuente: Elaboración propia, 2025.

El modelo F mantiene cifras elevadas ($mAP@0.5 = 97.1\%$), con "Motociclista" alcanzando un recall y precisión cercanos al 99%. "Con casco" y "Sin casco" registran ligeras disminuciones en sus métricas, particularmente "Sin casco", cuyo recall baja a 90.4% véase Tabla 3.7. Los errores entre "Casco" y "Sin casco" se mantienen bajos; sin embargo, el modelo exhibe cierta fluctuación que podría indicar el inicio de sobreajuste en clases menos robustas. La matriz refleja una tendencia hacia estabilización de las detecciones correctas véase Figura 3.7, aunque sin mejoras sustanciales respecto al modelo E.

Figura III.6: *Matriz de confusión normalizada modelo F*



Nota. La matriz de confusión normalizada muestra el porcentaje de predicciones correctas e incorrectas para cada clase del modelo F.

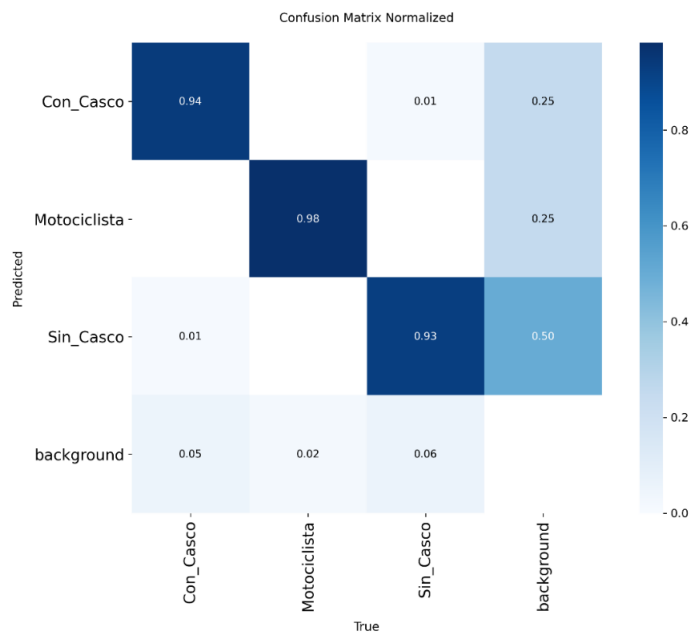
- **Modelo G-70 épocas:**

Tabla III.7: *Resumen modelo G*

Clase	Precisión	Recall	mAP@0.5
Con casco	0.958	0.94	0.951
Motociclista	0.982	0.975	0.989
Sin casco	0.966	0.916	0.974
Total	0.922	0.920	0.971

Nota. Fuente: Elaboración propia, 2025.

Figura III.7: *Matriz de confusión normalizada modelo G*



Nota. La matriz de confusión normalizada muestra el porcentaje de predicciones correctas e incorrectas para cada clase del modelo G.

La consistencia global persiste ($mAP@0.5 = 97.1\%$). "Con casco" logra un recall de 94.0%, "Sin casco" aumenta a 91.6% y "Motociclista" se mantiene en 97.5% véase Tabla 3.8. La matriz de confusión evidencia una reducción en la confusión entre "Con casco" y "Sin casco" (1–1%), y una homogeneidad en las tasas de acierto véase Figura 3.8. El modelo demuestra estabilidad en la predicción, donde todas las clases se aproximan a su mejor rendimiento sin evidencia clara de sobreajuste.

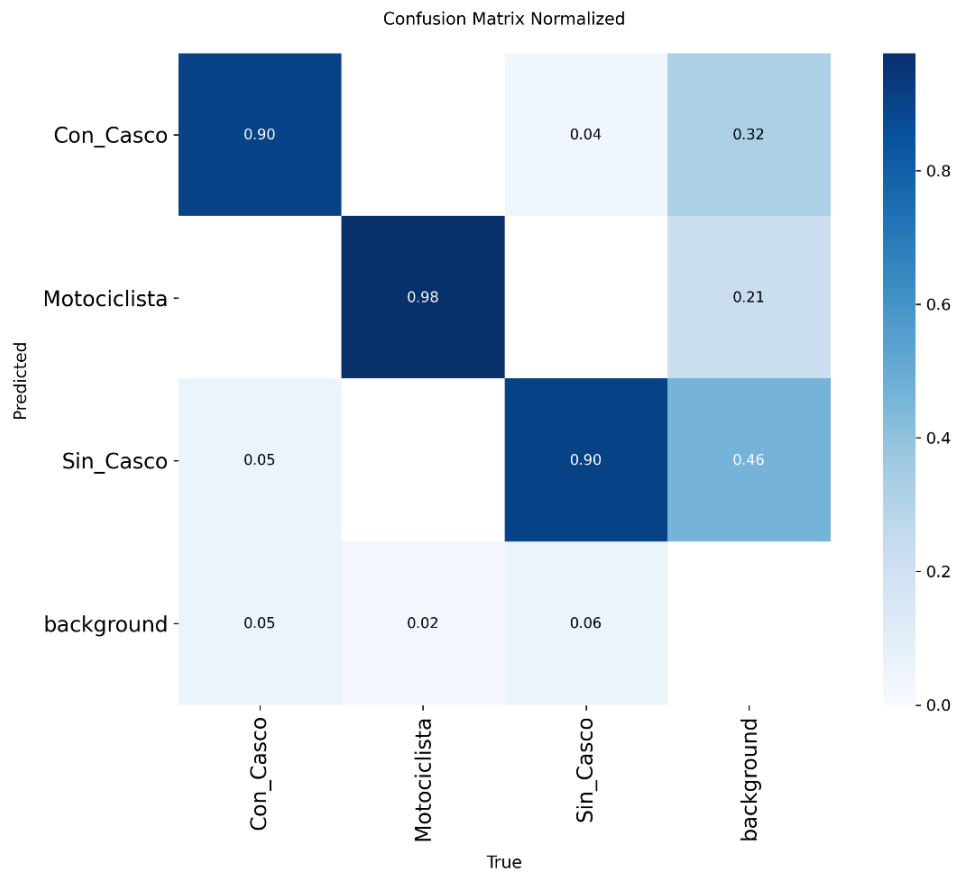
- **Modelo H-80 épocas:**

Tabla III.8: *Resumen modelo H*

Clase	Precisión	Recall	$mAP@0.5$
Con casco	0.865	0.93	0.952
Motociclista	0.979	0.976	0.986
Sin casco	0.947	0.912	0.971
Total	0.931	0.939	0.969

Nota. Fuente: Elaboración propia, 2025.

Figura III.8: *Matriz de confusión normalizada modelo H*



Nota. La matriz de confusión normalizada muestra el porcentaje de predicciones correctas e incorrectas para cada clase del modelo H.

Si bien el rendimiento general permanece elevado ($mAP@0.5 = 96.9\%$), se observa que "Con casco" sufre una leve reducción en precisión (86.5%), aunque su recall sigue siendo alto (93.0%). La clase "Sin casco" mantiene valores aceptables, y "Motociclista" se consolida como la más robusta (recall 97.6%) véase Tabla 3.9. Los errores entre " Sin casco " y " Con casco " crecen moderadamente (4–5%) véase Figura 3.9, lo que sugiere que el modelo empieza a perder generalización para clasificar objetos pequeños, síntoma posible de sobreajuste.

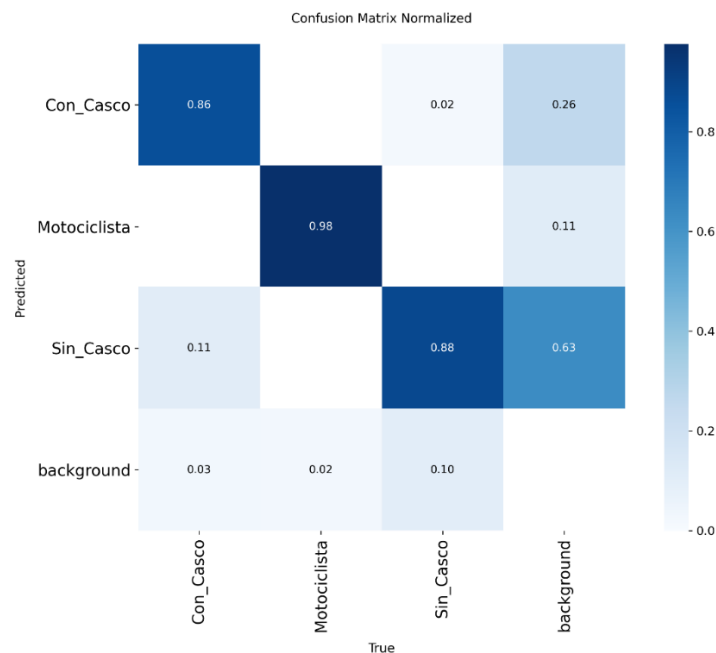
- **Modelo I-90 épocas:**

Tabla III.9: *Resumen modelo I*

Clase	Precisión	Recall	mAP@0.5
Con casco	0.926	0.91	0.949
Motociclista	0.989	0.977	0.986
Sin casco	0.898	0.876	0.951
Total	0.938	0.921	0.962

Nota. Fuente: Elaboración propia, 2025.

Figura III.9: *Matriz de confusión normalizada modelo I*



Nota. La matriz de confusión normalizada muestra el porcentaje de predicciones correctas e incorrectas para cada clase del modelo I.

En el modelo I, la degradación empieza a hacerse perceptible. El mAP@0.5 cae ligeramente a 96.2%, "Con casco" disminuye a 91.0% de recall y "Sin casco" a 87.6% véase Tabla 3.10. Las confusiones entre ambas clases que asciende a 13% véase Figura 3.10. "Motociclista" aún se mantiene alto (97.7%) en recall. Estos resultados sugieren que el entrenamiento excesivo comienza a revertir las ganancias, sobre todo en la capacidad de distinguir motociclistas con y sin casco en escenarios complejos.

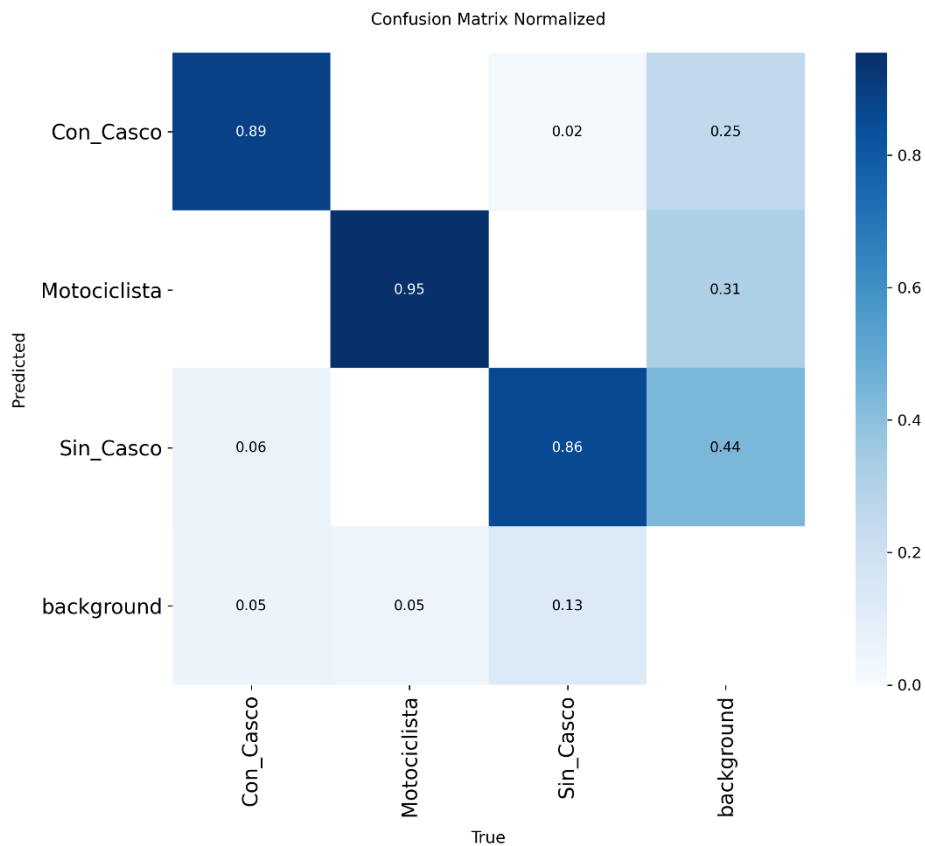
- **Modelo J 100 épocas:**

Tabla III.10: Resumen modelo J

Clase	Precisión	Recall	mAP@0.5
Con casco	0.926	0.91	0.936
Motociclista	0.965	0.955	0.983
Sin casco	0.922	0.849	0.948
Total	0.938	0.905	0.956

Nota. Fuente: Elaboración propia, 2025.

Figura III.10: Matriz de confusión normalizada modelo J



Nota. La matriz de confusión normalizada muestra el porcentaje de predicciones correctas e incorrectas para cada clase del modelo J.

Finalmente, el modelo J ($mAP@0.5 = 95.6\%$) muestra un descenso sistemático en varias métricas clave: "Con casco" y "Sin casco" caen a 91.0% y 84.9% de recall, respectivamente, y "Motociclista" baja a 95.5% véase Tabla 3.11. La matriz indica que las confusiones entre las clases "Con casco y Sin casco" aumentan véase Figura 3.11.

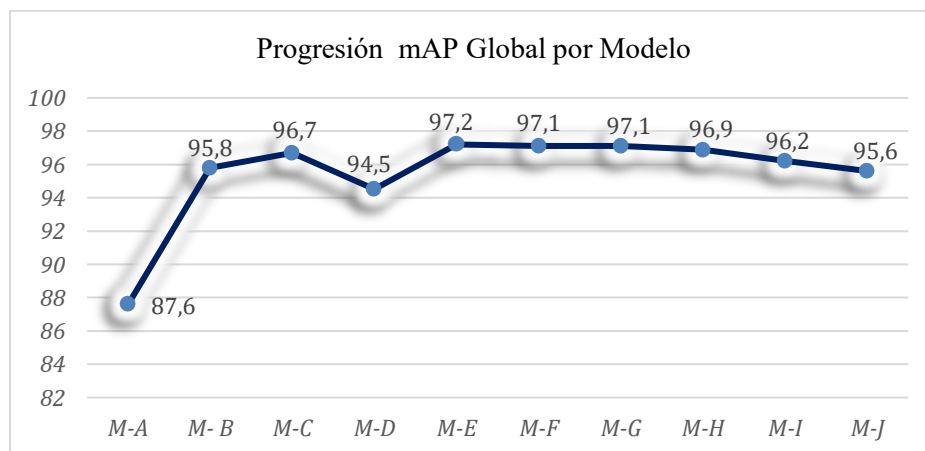
3.3 Análisis de resultados

3.3.1 Identificación del punto óptimo de entrenamiento

El análisis comparativo de los diez modelos entrenados revela un patrón evolutivo que se ajusta precisamente a la teoría del equilibrio sesgo-varianza descrita por Goodfellow et al. (2016) en el contexto del aprendizaje profundo. El comportamiento observado en la métrica $mAP@0.5$ a través de las configuraciones experimentales permite identificar tres fases distintivas del proceso de aprendizaje que son fundamentales para comprender la dinámica de optimización de redes neuronales convolucionales en tareas de detección de objetos.

A continuación, se presenta la gráfica de líneas que muestra la variación del rendimiento ($mAP@0.5$) en función del número de épocas utilizadas en el entrenamiento de los diez modelos evaluados véase Figura 3.12.

Figura III.11: Gráfico de líneas mAP vs Épocas



Nota. Cada punto representa el mejor resultado de validación alcanzado por cada modelo (A–J).

La primera fase, que abarca los modelos A, B y C (correspondientes a 10, 20 y 30 épocas de entrenamiento), muestra un crecimiento rápido característico del aprendizaje inicial: el modelo mejora su $mAP@0.5$ desde 87.6% hasta 96.7%. Este patrón concuerda con lo que Chollet (2018)

describe como una etapa donde el modelo realiza grandes ajustes en sus parámetros internos para comenzar a identificar patrones visuales clave. Durante este período, la red aprende a reconocer características simples pero esenciales, como bordes, texturas y formas básicas, que sirven de base para tareas más complejas de detección. El notable aumento de 9.1 puntos porcentuales en solo 20 épocas entre los modelos A y C indica que la arquitectura YOLOv11m, previamente entrenada en un gran conjunto de imágenes generales (COCO), ya posee conocimientos previos reutilizables, lo que facilita un aprendizaje más eficiente y rápido en la detección específica de cascos en motociclistas.

La segunda fase, representada por los modelos D y E (40 a 50 épocas), marca la transición hacia el mejor rendimiento posible del modelo. El modelo D presenta una caída temporal en su rendimiento (94.5%), constituye un fenómeno esperado según Géron (2019), quien indica que durante el entrenamiento pueden surgir pequeñas variaciones debido al uso de diferentes subconjuntos de datos, generando fluctuaciones antes de alcanzar estabilidad. Sin embargo, el modelo E alcanza el punto máximo de rendimiento con un mAP@0.5 del 97.2%, representando el balance ideal donde la red neuronal logra dos objetivos importantes al mismo tiempo: aprender bien los patrones específicos de los datos de entrenamiento y poder funcionar correctamente con imágenes nuevas que nunca ha visto.

Este punto óptimo no es casualidad, sino el resultado de lograr un equilibrio, explicado por Goodfellow et al. (2016), donde el error total de un modelo se puede descomponer en tres componentes: sesgo, varianza y un mínimo de error inevitable debido al ruido. En el modelo E, el sesgo se reduce porque el entrenamiento permite que el modelo aprenda patrones complejos, como diferentes tipos de cascos, ángulos de captura y condiciones de iluminación en Cobija. Al mismo tiempo, la varianza se mantiene controlada porque el modelo no ha sido entrenado en exceso, lo que le permite adaptarse correctamente a nuevas imágenes que no están exactamente representadas en los datos de entrenamiento. Así, se logra un modelo capaz de captar la complejidad de los datos sin perder la capacidad de generalizar, cumpliendo los principios matemáticos del equilibrio entre sesgo y varianza descritos en la literatura.

La tercera fase, representada por los modelos F a J (60 a 100 épocas), muestra que las mejoras del modelo comienzan a disminuir y aparece el inicio del sobreajuste. El mAP@0.5 disminuye progresivamente desde 97.1% en el modelo F hasta 95.6% en el modelo J, una pérdida de 1.6 puntos porcentuales. De acuerdo con Szeliski (2022), este comportamiento ocurre porque el

modelo empieza a aprender detalles demasiado específicos o poco representativos (conocidos como “artefactos”) de las imágenes de entrenamiento, incluyendo ruido visual o variaciones aleatorias. En lugar de seguir enfocándose en los patrones generales que definen las clases “con casco”, “sin casco” y “motociclista”, la red empieza a “memorizar” características particulares del conjunto de entrenamiento, lo que reduce su capacidad para reconocer correctamente nuevas imágenes no vistas.

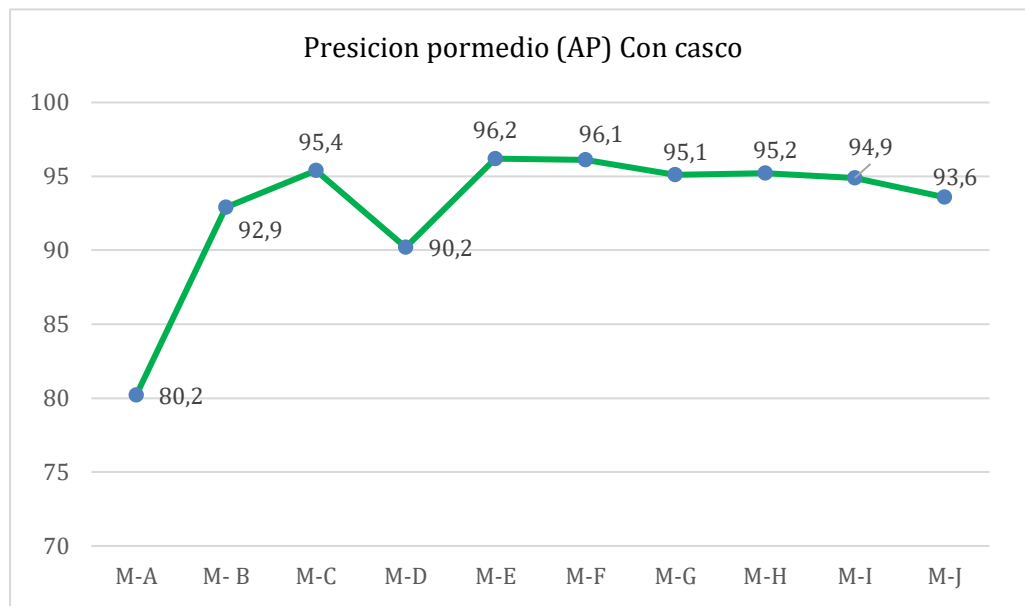
Este comportamiento valida empíricamente el principio de parsimonia en aprendizaje automático, establecido por Géron (2019), que sostiene que configuraciones de entrenamiento moderadas suelen ser suficientes y más eficientes que entrenamientos prolongados. En el contexto específico de esta investigación, 50 épocas representan el punto de equilibrio óptimo entre costo computacional y máximo rendimiento predictivo.

3.3.2 Análisis Diferencial por Clase

El análisis detallado del rendimiento por clase muestra que el modelo YOLOv11m no aprende igual para todas las categorías, sino que responde de manera diferente según las características propias de cada una. Observar cómo evoluciona la Precisión Promedio (AP) de cada clase a lo largo de los diez modelos permite identificar tendencias específicas. Por ejemplo, es posible que la clase "Motociclista" mantenga un AP alto debido a su tamaño y forma distintivos, mientras que "Con casco" y "Sin casco" pueden ser más difíciles de diferenciar si comparten elementos visuales o aparecen en situaciones de iluminación compleja.

A continuación, se presenta el gráfico comparativo del rendimiento por clase, donde se visualizan los valores de Precisión Promedio (AP) obtenidos para “Con casco”, véase Figura 3.13.

Figura III.12: Gráfico de línea evolución de la clase sin casco



Nota. Nota. Cada punto representa el resultado para la clase con casco alcanzado por cada modelo (A–J).

La clase "Con casco" presenta la trayectoria de aprendizaje más compleja y variable entre las tres categorías evaluadas. El modelo A inició con una precisión promedio (AP) del 80.2%, el más bajo entre todas las clases, evidenciando que la detección de cascos protectores constituye el desafío técnico más significativo para el modelo. Esta dificultad inicial se explica por múltiples factores que Szeliski (2022) identifica como determinantes de la complejidad de detección en visión por computadora.

Primero, la dimensión espacial reducida del casco en comparación con el motociclista completo implica que este objeto ocupa significativamente menos píxeles en las imágenes capturadas. Mientras un motociclista completo puede abarcar entre 15,000 y 30,000 píxeles en una imagen de 640×640 píxeles, un casco típicamente ocupa solo entre 1,500 y 3,000 píxeles, representando aproximadamente el 10% del área del motociclista. Esta proporción reducida limita la cantidad de información visual disponible para las capas convolucionales de la red neuronal, dificultando la extracción de características distintivas.

En segundo lugar, los cascos presentan una gran variedad dentro de la misma categoría, mucho mayor que otros objetos urbanos. El conjunto de datos contiene cascos de diferentes tipos

homologados por la Policía boliviana (integrales, abiertos, modulares), con una amplia diversidad de colores (negro, blanco, rojo, azul, multicolor), acabados (mate, brillante) y estados (nuevos, desgastados). Esta mezcla de características visuales hace que sea más difícil para el modelo aprender a reconocerlos correctamente, porque debe identificar muchos patrones distintos para generalizar bien. Goodfellow et al. (2016) explican que cuanto mayor es la diversidad interna dentro de una clase, mayor capacidad y entrenamiento necesita la red neuronal para aprender y distinguir todos esos detalles relevantes.

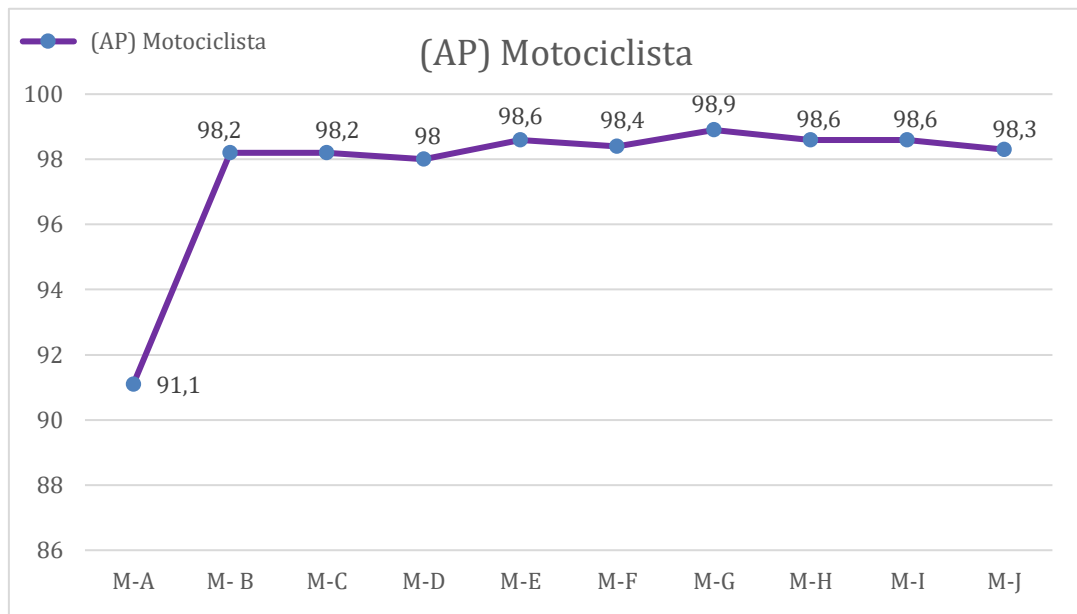
Tercero, el fenómeno de oclusión parcial afecta desproporcionadamente a los cascos. En el contexto urbano de Cobija, es común que los cascos aparezcan parcialmente cubiertos en las imágenes debido a elementos del entorno, como señalización, vehículos sobrepuestos, o por la postura del conductor, como la inclinación de la cabeza o el movimiento. Szeliski (2022) establece que las redes convolucionales tienen mayor dificultad para detectar objetos pequeños parcialmente ocultos, ya que la información visual crítica puede estar fragmentada o ausente.

La curva de aprendizaje muestra una mejora acelerada entre los modelos A y E, donde el AP (Con casco) se incrementa de 80.2% a 96.2%, ganando 16 puntos porcentuales. Este crecimiento evidencia que la arquitectura YOLOv11m, con suficientes épocas de entrenamiento, logra desarrollar representaciones robustas capaces de manejar la variabilidad y complejidad visual de los cascos protectores. El modelo E representa el punto donde la red ha aprendido a identificar características invariantes del casco (forma hemisférica característica, posición relativa sobre la cabeza del conductor, textura superficial) independientemente de color, tipo o condiciones de captura.

Sin embargo, los modelos subsiguientes (F a J) muestran una tendencia descendente, con el modelo J alcanzando solo 93.6% de AP, perdiendo 2.6 puntos porcentuales respecto al óptimo. Esta degradación refleja sobreajuste específico en esta clase, donde el modelo comienza a memorizar configuraciones particulares del conjunto de entrenamiento en lugar de patrones generalizables.

A continuación, se presenta el gráfico comparativo del rendimiento por clase, donde se visualizan los valores de Precisión Promedio (AP) obtenidos para “Motociclista”, véase Figura 3.14.

Figura III.13: Grafica de línea de la evolución de la clase motociclista



Nota. Nota. Cada punto representa el resultado para la clase motociclista alcanzado por cada modelo (A-J).

La clase "Motociclista" demuestra el comportamiento más consistente y robusto a través de todas las configuraciones experimentales, alcanzando una precisión promedio (AP) del 91.1% en el modelo inicial A y manteniéndose establemente por encima del 98% desde el modelo D en adelante, con un máximo de 98.6% en el modelo E. Esta superioridad en rendimiento y estabilidad responde a características estructurales del objeto que facilitan significativamente su detección por redes neuronales convolucionales.

La primera razón fundamental es la dimensión espacial superior del motociclista completo. Como establece Szeliski (2022), existe una correlación directa entre el área de píxeles que ocupa un objeto en una imagen y la facilidad de detección por modelo de visión por computadora. Un motociclista con su vehículo presenta una silueta grande y distintiva que típicamente abarca entre 20% y 40% del área total de la imagen en las capturas realizadas en Cobija. Esta proporción sustancial proporciona abundante información visual para las capas convolucionales, permitiendo la extracción de múltiples características discriminativas en diferentes escalas de resolución.

La segunda razón radica en la redundancia de características visuales distintivas. Un motociclista en su vehículo presenta múltiples elementos identificables: la geometría circular de las ruedas, la forma característica del manubrio, la postura corporal del conductor, la estructura del

chasis de la motocicleta, y la relación espacial entre estos componentes. Esta multiplicidad de características crea lo que Goodfellow et al. (2016) denominan "representaciones robustas", donde la red neuronal no depende de un único patrón visual sino de un conjunto de características complementarias. Incluso si algunos elementos están parcialmente ocultos u ocurren variaciones en la apariencia, la presencia de múltiples características alternativas asegura una detección confiable.

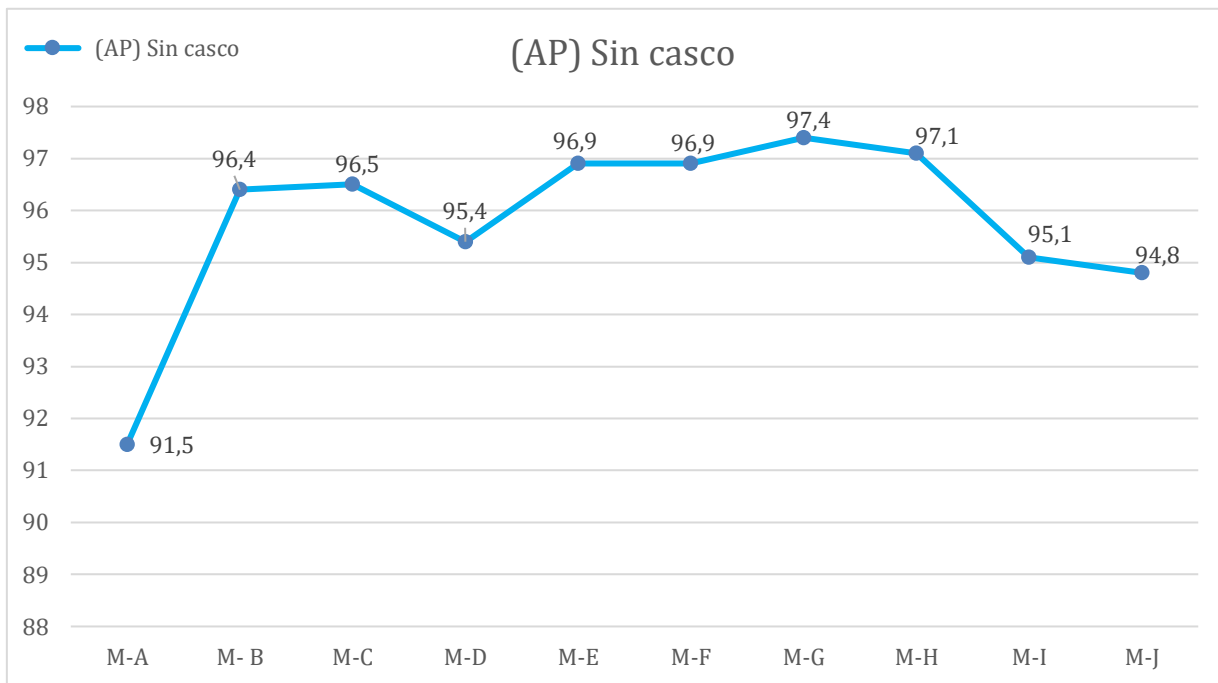
La tercera razón se relaciona con la variabilidad intraclase controlada. Aunque existen diferentes tipos de motocicletas circulando en Cobija, todas comparten una configuración estructural fundamental: dos ruedas en línea, un manubrio frontal, un asiento y la presencia de un conductor en postura característica. Esta estructura básica común reduce la complejidad del espacio de características que el modelo debe aprender, facilitando la generalización. Chollet (2018) explica que cuando las diferencias dentro de una misma clase son pequeñas y, al mismo tiempo, las diferencias entre clases son grandes, las redes neuronales pueden clasificar esas categorías con mayor facilidad.

La evolución de la precisión promedio (AP) para esta clase muestra un incremento rápido desde 91.1% (modelo A) hasta alcanzar una meseta de alto rendimiento en 98.6% (modelo E), manteniéndose establemente en el rango 98.3%-98.6% en modelos subsiguientes. Esta estabilidad contrasta marcadamente con el comportamiento más errático observado en las clases "Con casco" y "Sin casco", evidenciando que la detección de motociclistas es una tarea más robusta ante variaciones en la configuración del entrenamiento.

Es particularmente significativo que incluso el modelo J, que muestra degradación en otras clases, mantiene un AP de 98.3% para "Motociclista", solo 0.3 puntos porcentuales por debajo del óptimo. Esto sugiere que esta clase es relativamente inmune al sobreajuste debido a sus características visuales fuertemente distintivas y consistentes. La red neuronal puede continuar refinando su representación interna de motociclistas sin caer en memorización excesiva de artefactos del conjunto de entrenamiento, ya que los patrones visuales fundamentales son suficientemente claros y repetibles.

A continuación, se presenta el gráfico comparativo del rendimiento por clase, donde se visualizan los valores de Precisión Promedio (AP) obtenidos para "Sin casco", véase Figura 3.15.

Figura III.14: Grafica de línea de evolución de la clase sin casco



Nota. Nota. Cada punto representa el resultado para la clase sin casco alcanzado por cada modelo (A–J).

La clase "Sin casco" muestra un comportamiento intermedio, comenzando con una precisión promedio (AP) de 91.5% en el modelo A, superior a "Con casco" pero inferior a "Motociclista". Su rendimiento mejora de forma sostenida hasta el modelo E (96.9% de AP) y luego presenta una disminución moderada hasta 94.8% en el modelo J. Este patrón refleja que detectar la ausencia de un casco implica desafíos diferentes comparado con la detección de objetos presentes.

Szeliski (2022) explica que detectar "ausencia" implica en realidad identificar características positivas alternativas: la presencia de cabello, rasgos faciales visibles, piel expuesta, o la geometría de la cabeza humana sin la forma hemisférica característica del casco. Esta reconfiguración conceptual del problema transforma la tarea en una detección de características específicas de "cabeza descubierta".

Goodfellow et al. (2016) documenta que las redes neuronales convolucionales aprenden eficientemente a detectar patrones consistentes y repetibles. La cabeza humana descubierta, con sus características anatómicas universales, proporciona señales visuales más uniformes que la

diversidad artificial de los cascos protectores. Esto resulta en una curva de aprendizaje inicial más favorable para la clase "Sin casco".

Chollet (2018) señala que estas variaciones contextuales incrementan la complejidad del problema, requiriendo que el modelo aprenda representaciones invariantes a condiciones de iluminación y ángulo de visualización.

La evolución del AP muestra crecimiento sostenido hasta alcanzar el máximo en el modelo E (96.9%), ganando 5.4 puntos porcentuales respecto al inicial. Este incremento refleja que el modelo desarrolla progresivamente la capacidad de identificar cabezas descubiertas bajo diversas condiciones. Las capas convolucionales aprenden a detectar características invariantes como la forma general de la cabeza humana, la presencia de cabello o piel expuesta, y la ausencia de texturas artificiales, independientemente de iluminación o ángulo.

Los modelos posteriores al óptimo (F-J) muestran degradación moderada, perdiendo 2.1 puntos porcentuales hasta alcanzar 94.8% en el modelo J. Esta disminución es menos severa que en la clase "Con casco" (2.6 puntos), sugiriendo que la detección de cabezas descubiertas es moderadamente robusta ante sobreajuste. Sin embargo, la degradación es más pronunciada que en "Motociclista" (0.3 puntos), evidenciando que objetos pequeños, como las cabezas descubiertas, son más susceptibles a perder precisión si el modelo comenzó a memorizar detalles específicos de las imágenes de entrenamiento en vez de aprender patrones generales útiles.

3.3.3 Análisis Estado del Arte

Los resultados obtenidos son consistentes con investigaciones previas en detección de cascos mediante YOLO. Mercado Reyna (2023) reportó 97.24% de precisión con arquitecturas Inception v3 reentrenado basado en CNN, mientras que Jia et al. (2021) alcanzaron 97.7% con YOLOv5 mejorado. El presente estudio valida que estas arquitecturas mantienen su efectividad en contextos urbanos con dataset locale de tamaño moderado.

A continuación, se presenta una tabla comparativa del rendimiento obtenido frente a otras investigaciones previas, véase Tabla 3.12.

Tabla III.11: Comparación con estado del arte

Autor	Año	Modelo	Métrica	Resultado
Mercado Reyna	2023	Inception v3 reentrenado	Precisión	97.24%
Jia et al.	2021	YOLOv5 mejorado	mAP@0.5	97.7%
Presente estudio	2025	YOLOv11m	mAP@0.5	97.2%

Nota. Los valores reflejan el desempeño reportado por cada autor y por el presente estudio; la comparación permite evaluar la efectividad de distintas arquitecturas CNN y YOLO en la detección de casco.

3.3.4 Análisis de Errores Críticos

Las matrices de confusión revelan que el error más crítico para seguridad vial (clasificar "Con casco" como "Sin casco") se redujo del 5% (modelo A) al 1% (modelo E). Este resultado es fundamental para la viabilidad operativa del sistema, ya que minimiza los falsos negativos que podrían resultar en infracciones no detectadas.

3.3.5 Validación Empírica de la Hipótesis

La validación de la hipótesis en este estudio se fundamenta en el enfoque práctico de validación de métodos, el cual establece que una herramienta se considera validada cuando existe evidencia objetiva de que cumple los requisitos definidos para la función específica que se le asigna. En este caso, la herramienta corresponde al modelo de visión por computadora YOLOv11m y su función consiste en detectar el uso de casco en motociclistas a partir de imágenes estáticas capturadas en el contexto urbano de Cobija, bajo el requisito cuantitativo de alcanzar una precisión promedio media (mAP0.5) igual o superior al 90%.

En coherencia con dicho enfoque, el proceso de validación se organiza en cinco etapas interrelacionadas. La primera etapa implica la identificación explícita de la función del modelo, la cual se define como la detección automática de tres clases de objetos ("con casco", "sin casco" y "motociclista") en imágenes diurnas de tránsito urbano. La segunda etapa consiste en la especificación de requisitos y características de desempeño, donde se establece el umbral mínimo de $mAP_{0.5} \geq 0.90$ como criterio de aceptación, junto con métricas complementarias de precisión y recall por clase. La tercera etapa comprende la determinación empírica del desempeño mediante la evaluación de diez configuraciones de entrenamiento (épocas de 10 a 100) sobre un conjunto de

prueba independiente de 200 imágenes locales, seleccionadas intencionalmente para representar condiciones reales de iluminación, ángulos de captura y densidad de tráfico en Cobija, lo que sitúa el análisis en un marco no probabilístico orientado a la comprobación técnica más que a la inferencia estadística poblacional.

La cuarta etapa implica la comparación objetiva de los resultados obtenidos contra los requisitos preestablecidos. Los datos evidencian que la configuración óptima (50 épocas) alcanza un mAP0.5 de 97.2%, con valores de precisión por clase superiores al 96% y una tasa de confusión crítica entre "con casco" y "sin casco" inferior al 1%, indicando que las características de desempeño del modelo superan el umbral mínimo y satisfacen los requisitos funcionales definidos para la tarea de detección. La quinta etapa consiste en la expresión formal de los alcances y límites de la evidencia, reconociendo que la comprobación alcanzada no garantiza necesariamente el mismo nivel de rendimiento en escenarios nocturnos, condiciones climáticas adversas, o ciudades con características urbanísticas diferentes, por lo que la transferencia y generalización de los resultados requerirán futuras validaciones adicionales, idealmente bajo esquemas probabilísticos o pruebas multicéntricas que involucren mayor diversidad contextual.

CAPITULO IV.

CONCLUSIONES

4.1 CONCLUSIONES

En base en el análisis exhaustivo de los resultados obtenidos durante el desarrollo de esta investigación y su interpretación crítica en el marco de los objetivos general y específicos establecidos, así como en relación con la hipótesis planteada, el marco teórico desarrollado y el contexto metodológico cuantitativo aplicado, se formulan las siguientes conclusiones que sintetizan los hallazgos científicos más relevantes del estudio, evalúan el cumplimiento de los propósitos investigativos y establecen las implicaciones técnicas, metodológicas y contextuales derivadas del desarrollo y validación del modelo de visión por computadora para la detección del uso de casco en motociclistas en la ciudad de Cobija, Pando, Bolivia.

El presente estudio cumple satisfactoriamente el objetivo general al desarrollar y validar técnicamente un modelo de visión por computadora basado en la arquitectura YOLOv11m que alcanza un rendimiento de mAP@0.5 del 97.2% en la detección del uso de casco en motociclistas, superando ampliamente el umbral del 90% establecido como criterio de éxito. Este resultado valida la hipótesis de investigación y demuestra la viabilidad técnica de aplicar modelos de aprendizaje profundo en escenarios urbanos reales. El modelo desarrollado distingue efectivamente tres clases de objetos ("con casco", "sin casco" y "motociclista") con precisión consistente, minimiza errores críticos de clasificación (1% de confusión entre "con casco" y "sin casco") y evidencia capacidad de generalización sobre datos inéditos del contexto local. La validación empírica confirma que la detección automatizada del uso de casco mediante visión por computadora es técnicamente viable y potencialmente escalable para su implementación operativa en sistemas de fiscalización vial automatizada en Cobija.

El proceso de construcción del conjunto de datos mediante la combinación estratégica de un dataset público de 1,592 imágenes (80% entrenamiento, 20% validación) con un conjunto de 200 imágenes locales inéditas capturadas en Cobija resulta metodológicamente apropiado y técnicamente efectivo. El dataset público proporciona diversidad visual y volumen suficiente para el entrenamiento robusto del modelo mediante aprendizaje por transferencia (Transfer Learning), mientras que el conjunto local etiquetado manualmente garantiza la validación contextualizada en condiciones urbanas específicas de la ciudad. Esta estrategia híbrida optimiza recursos al eliminar la necesidad de recolectar y etiquetar masivamente datos locales, reduciendo costos operativos y

tiempos de desarrollo sin comprometer la capacidad de generalización del modelo. El etiquetado manual supervisado de las 200 imágenes locales, que generó 638 instancias distribuidas entre las tres clases objetivo, establece una ground truth (verdad fundamental) confiable para la evaluación independiente del rendimiento. La validación empírica confirma que el modelo entrenado con datos públicos generaliza correctamente sobre el contexto local, evidenciando que conjuntos de datos moderados, pero bien estructurados y representativos son suficientes cuando se aplican técnicas de aprendizaje por transferencia con arquitecturas preentrenadas.

El análisis experimental de diez configuraciones de entrenamiento (10 a 100 épocas en incrementos de 10) establece que el punto de equilibrio óptimo entre costo computacional y precisión se sitúa en 50 épocas de entrenamiento, configuración que alcanza el máximo rendimiento global ($mAP@0.5 = 97.2\%$) y por clase ("con casco" 96.2% AP, "motociclista" 98.6% AP, "sin casco" 96.9% AP). La curva de aprendizaje evidencia tres fases distintivas: (1) crecimiento acelerado inicial (10-30 épocas: 87.6% a 96.7%), donde el modelo aprende patrones visuales fundamentales; (2) convergencia hacia el óptimo (40-50 épocas: 94.5% a 97.2%), alcanzando equilibrio entre sesgo y varianza; y (3) degradación progresiva por sobreajuste (60-100 épocas: 97.1% a 95.6%), donde el modelo memoriza artefactos específicos del conjunto de entrenamiento. Este comportamiento valida empíricamente los principios teóricos del equilibrio sesgo-varianza y confirma que configuraciones moderadas de entrenamiento son suficientes que entrenamientos prolongados para tareas de detección de objetos en contextos urbanos. El análisis diferencial por clase revela que "motociclista" mantiene rendimiento consistente y robusto ante variaciones de entrenamiento (98.3%-98.6%), mientras que "con casco" y "sin casco" exhiben mayor sensibilidad al sobreajuste, justificando la necesidad de seleccionar rigurosamente el punto de parada del entrenamiento.

La implementación técnica del modelo en Google Colab, plataforma en la nube con acceso a recursos GPU, mediante la librería Ultralytics y el framework Python, demuestra la viabilidad de ejecutar modelos de visión por computadora en entornos simulados sin requerir infraestructura computacional local avanzada. El modelo implementado integra exitosamente todos sus componentes esenciales: arquitectura YOLOv11m con pesos optimizados (50 épocas), configuración de hiperparámetros validados (umbral de confianza 0.25, IoU 0.5), procesamiento de imágenes de entrada (640×640 píxeles) y generación automatizada de métricas de evaluación (mAP, Precisión, Recall, matrices de confusión). La funcionalidad del modelo se verifica mediante

la ejecución de inferencias sobre el conjunto de prueba de 200 imágenes locales, obteniendo predicciones consistentes, reproducibles y documentadas. Esta implementación establece una base técnica sólida y replicable que facilita futuras fases de desarrollo, incluyendo la transición hacia procesamiento de video en tiempo real y despliegue operativo en dispositivos de borde (edge computing). El entorno simulado cumple satisfactoriamente su función de validación técnica preliminar, confirmando que el modelo opera correctamente con las especificaciones definidas y genera resultados confiables antes de considerar implementaciones operativas en campo.

La evaluación sistemática del modelo mediante métricas técnicas estándar sobre 200 imágenes inéditas capturadas en Cobija revela rendimiento consistente y altamente satisfactorio en todas las dimensiones evaluadas. El modelo alcanza un mAP@0.5 global del 97.2%, con precisión global del 96.5% y recall global del 94.2%. El análisis diferencial por clase confirma capacidad discriminativa robusta: la clase "con casco" obtiene 96.2% AP con recall del 95.0%, "motociclista" alcanza 98.6% AP con recall del 96.9%, y "sin casco" logra 96.9% AP con recall del 90.8%. Las matrices de confusión normalizadas evidencian que el modelo reduce el error crítico de seguridad vial clasificar motociclistas con casco como "sin casco" al 1%, minimizando falsos negativos que podrían resultar en infracciones no detectadas. Esta capacidad discriminativa resulta fundamental para la confiabilidad operativa de futuros sistemas de fiscalización automatizada. El análisis comparativo con estudios previos posiciona el modelo desarrollado en niveles competitivos de rendimiento (Mercado Reyna 2023: 97.24%; Jia et al. 2021: 97.7%), validando la efectividad de YOLOv11m en contextos urbanos con datasets de tamaño moderado. Los resultados confirman que el modelo cumple satisfactoriamente los criterios técnicos de precisión, sensibilidad y especificidad requeridos para aplicaciones de seguridad vial, estableciendo evidencia empírica sólida de su viabilidad técnica en el contexto específico de Cobija.

CAPITULO V.

RECOMENDACIONES

5.1 RECOMENDACIONES

En base en los hallazgos obtenidos, las limitaciones identificadas y las proyecciones derivadas del desarrollo y validación del modelo de visión por computadora, se formulan las siguientes recomendaciones técnicas, metodológicas y operativas dirigidas a investigadores, autoridades de tránsito y desarrolladores tecnológicos interesados en expandir, optimizar o implementar sistemas automatizados de detección del uso de casco en motociclistas.

Se recomienda evaluar el modelo en procesamiento de video en tiempo real mediante implementación en dispositivos de borde (edge computing) con capacidad de cómputo acelerado (GPU integradas o unidades de procesamiento neuronal), instalando el sistema en puntos estratégicos de alta densidad vehicular de Cobija. Esta evaluación permitirá caracterizar el rendimiento del modelo en detección continua, determinar la tasa de fotogramas procesados por segundo (FPS), identificar desafíos operativos no evidentes en procesamiento de imágenes estáticas (oclusión dinámica, desenfoque por movimiento, variaciones abruptas de iluminación) y establecer la viabilidad técnica real del modelo más allá del entorno simulado validado en esta investigación.

Se sugiere expandir el conjunto de datos de entrenamiento incorporando imágenes capturadas durante condiciones de iluminación nocturna, eventos climáticos adversos característicos de la región amazónica (lluvia intensa, neblina matutina, tormentas eléctricas) y escenarios de tráfico denso con oclusión parcial severa. Esta expansión puede realizarse mediante captura dirigida de nuevas imágenes en estos escenarios específicos o mediante técnicas de aumento sintético de datos (data augmentation) que simulen estas condiciones (ajuste de brillo, adición de ruido, rotaciones, transformaciones geométricas). La incorporación de diversidad ambiental y temporal permitirá evaluar y mejorar la robustez del modelo ante la variabilidad contextual que podría presentarse en una implementación operativa continua de 24 horas, garantizando confiabilidad en cualquier condición climática o de iluminación.

Se recomienda realizar evaluación sistemática de potenciales sesgos algorítmicos del modelo mediante análisis diferenciado de rendimiento según variables no relevantes para la detección (tipo de motocicleta, marca del vehículo, color del casco, género del conductor, características étnico-fenotípicas). Esta evaluación debe verificar que la precisión del modelo se mantenga consistente independientemente de estas variables, asegurando equidad en la detección

y evitando discriminación algorítmica. En caso de identificarse sesgos estadísticamente significativos, se debe proceder a rebalancear el conjunto de entrenamiento o aplicar técnicas de mitigación de sesgos (re-ponderación de clases, ajuste de umbrales diferenciados, entrenamiento adversarial). Esta verificación resulta fundamental para garantizar transparencia, justicia y aceptabilidad social del sistema en una eventual implementación operativa.

Se sugiere ampliar las capacidades del modelo para detectar infracciones complementarias de seguridad vial en motociclistas, tales como uso de teléfonos móviles durante la conducción, exceso de pasajeros (más de dos ocupantes por motocicleta), transporte inadecuado de menores sin elementos de protección, o conducción sin uso de elementos reflectivos en horario nocturno. Esta expansión puede realizarse aprovechando la arquitectura YOLOv11m validada y el conjunto de datos base de motociclistas, mediante reentrenamiento con nuevas clases objetivo o implementación de modelos en cascada. La detección multiclase de comportamientos de riesgo maximizaría el valor del modelo desarrollado, transformándolo en un sistema integral de monitoreo de seguridad vial que aporte información cuantitativa diversificada para políticas públicas preventivas.

Se recomienda integrar el modelo de detección de cascos con un sistema de Reconocimiento Óptico de Caracteres (OCR) especializado en lectura automatizada de placas vehiculares bolivianas, utilizando arquitecturas como CRNN (Convolutional Recurrent Neural Networks) o modelos transformer especializados (TrOCR). Esta integración permitiría no solo detectar la infracción sino también identificar automáticamente el vehículo infractor, completando el ciclo de fiscalización automatizada mediante la generación de reportes estructurados que incluyan: imagen de evidencia, coordenadas geográficas de captura, fecha y hora precisa, tipo de infracción detectada y placa vehicular. Este sistema integrado facilitaría significativamente la labor de las autoridades de tránsito, reduciendo tiempos de procesamiento de infracciones y mejorando la trazabilidad del proceso sancionatorio.

REFERENCIAS

Afzal, A., Draz, HU, Khan, MZ y Khan, MUG (2021). Detección automática de infracciones de casco en motociclistas a partir de vídeos de vigilancia mediante enfoques de aprendizaje profundo de visión artificial. En *2021, Conferencia Internacional sobre Inteligencia Artificial (ICAI)* (pp. 252-257). IEEE. <https://doi.org/10.1109/ICAI52203.2021.9445249>

Agencia de Noticias Fides. (2025, 13 de enero). *En 2024 se registraron 20.942 siniestros de tránsito y dejaron 26.299 personas accidentadas*. <https://noticiasfides.com/nacional/seguridad/en-2024-se-registraron-20-942-siniestros-de-transito-y-dejaron-26-299-personas-accidentadas>

Arias, FG (2012). *El proyecto de investigación: Introducción a la metodología científica* (7ª ed.). Episteme.

Bhandari, A. (28 de abril de 2020). Todo lo que debe saber sobre la matriz de confusión para el aprendizaje automático. *Analytics Vidhya*. <https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/>

Bisong, E. (2019). Google Colaboratory. En E. Bisong, *Creación de modelos de aprendizaje automático y aprendizaje profundo en Google Cloud Platform: Una guía completa para principiantes* (pp. 59-64). Apress. https://doi.org/10.1007/978-1-4842-4470-8_7

Bolivia. (1978). *Resolución Suprema N.º 187444, 8 de junio de 1978: Reglamento del Código Nacional de Tránsito*. Gaceta Oficial del Estado Plurinacional de Bolivia.

Bolivia. (2008). *Ley N.º 3988, 18 de diciembre de 2008: Código de Tránsito*. Gaceta Oficial del Estado Plurinacional de Bolivia.

Cabrera, P. (2020). *Metodología de la investigación: Un enfoque pedagógico* (3ª ed.). CognitusCA

Escobar-Pérez, J., & Cuervo-Martínez, Á. (2008). Validez de contenido y juicio de expertos: Una aproximación a su utilización. *Avances en Medición*, 6, 27–36.

Chollet, F. (2018). *Aprendizaje profundo con Python*. Publicaciones Manning.

Géron, A. (2019). *Aprendizaje automático práctico con Scikit-Learn, Keras y TensorFlow: Conceptos, herramientas y técnicas para construir sistemas inteligentes* (2.^a ed.). O'Reilly Media.

Gholamy, A., Kreinovich, V. y Kosheleva, O. (2018). *¿Por qué existe una relación 70/30 u 80/20 entre los conjuntos de entrenamiento y de prueba? Una explicación pedagógica* (Informe Técnico UTEP-CS-18-09). Universidad de Texas en El Paso, Departamento de Ciencias de la Computación. https://scholarworks.utep.edu/cs_techrep/1209

Goodfellow, I., Bengio, Y. y Courville, A. (2016). *Aprendizaje profundo*. MIT Press. <http://www.deeplearningbook.org>

Guillén, O., Sánchez, M., & Begazo, L. (2020). *Pasos para elaborar una tesis de tipo correlacional, bajo el enfoque cuantitativo, variable categórica, escala ordinal y la estadística no paramétrica*. Magdalena del Mar. http://cliic.org/2020/Taller-Normas-APA-2020/libro-elaborar-tesis-tipo-correlacional-octubre-19_c.pdf

García Borges, F. E. (2025). Sistema de monitorización de tráfico en directo mediante detección de objetos basado en aprendizaje profundo [Trabajo Fin de Grado, Universidad de Cádiz].

Hao, Y., Pei, H., Lyu, Y., Yuan, Z., Rizzo, J.-R., Wang, Y. y Fang, Y. (2022). Comprensión del impacto de la calidad de imagen y la distancia entre objetos y el rendimiento de detección de objetos. *arXiv*. <https://arxiv.org/abs/2209.08237>

Hernández, R., Fernández, C., & Baptista, P. (2014). *Metodología de la investigación* (6^a ed.). McGraw-Hill Interamericana.

Hernández, R., & Mendoza, C. (2018). *Metodología de la investigación: Las rutas cuantitativa, cualitativa y mixta*. Educación McGraw-Hill.

Instituto Nacional de Estadística. (2024). *Cuadro N° 2.9 Pando: Parque automotor, según tipo de servicio y clase de vehículo, 2003-2024*. <https://www.ine.gob.bo/index.php/estadisticas-economicas/transportes/parque-automotor-cuadros-estadisticos/>

Jia, W., Xu, S., Liang, Z., Zhao, Y., Min, H., Li, S. y Yu, Y. (2021). Detección automática de cascos de motociclistas en tiempo real en el tráfico urbano mediante el detector YOLOv5 mejorado. *IET Image Processing*, 15 (14), 3623-3637. <https://doi.org/10.1049/ipr2.12295>

Jocher, G., Chaurasia, A., Qiu, J. y Stoken, A. (2020). *YOLOv5* (Versión 7.0) [Software]. Ultralíticos. <https://github.com/ultralytics/yolov5>

La Madriz, J. (2019). *Metodología de la investigación: Actuación humana orientada al conocimiento de la realidad observable*. Centro de Investigación y Desarrollo Ecuador. <http://repositorio.cidecuador.org/bitstream/123456789/75/1/Metodologia%20de%20la%20Investigacion.pdf>

Lazos Martínez, R. J. (2004). La validación de métodos: Un enfoque práctico. *Simpósio de Metrología 2004*, 25-27.

Martínez Heras, J. (2022, 18 de septiembre). *Las 7 fases del proceso de aprendizaje automático*. https://github.com/jmartinezheras/2018-machinelearning-lectures-esa/tree/master/iartificial_net

Martínez Morales, G. (2022). *Sistema de detección de motocicletas y del uso de casco de seguridad por medio de algoritmos de inteligencia artificial* [Tesis de licenciatura]. Tecnológico de Estudios Superiores de Ecatepec.

Maya, E. (2014). *Métodos y técnicas de investigación*. Universidad Nacional Autónoma de México. http://www.librosoa.unam.mx/bitstream/handle/123456789/2418/metodos_y_tecnicas.pdf?sequence=3&isAllowed=y

Mercado Reyna, J. (2023). *Detección del uso del casco en conductores de motocicleta, utilizando redes neuronales convolucionales* [Tesis de maestría]. Universidad Autónoma de Zacatecas.

Ministerio de Gobierno. (2008). *Reglamento del Código Nacional de Tránsito*. Gobierno del Estado Plurinacional de Bolivia.

Ministerio de Obras Públicas, Servicios y Vivienda. (2019). *Plan Nacional de Seguridad Vial 2019-2025*. Gobierno del Estado Plurinacional de Bolivia.

Montoya, S., Rodríguez, J., Albavera, C. y Valero, O. (2016). Evidencias para la prevención y control de lesiones en motociclistas. *Revista Cubana de Salud Pública*, 42 (4), 630-644.

Observatorio Boliviano de Seguridad Ciudadana y Lucha Contra las Drogas. (2023). *Dossier de estadísticas en seguridad vial 2017-2022*.

Organización Mundial de la Salud. (2023). *Informe sobre la situación mundial de la seguridad vial 2023*. <https://www.who.int/publications/i/item/9789240086517>

Organización Mundial de la Salud. (2023b). *Cascos: un manual de seguridad vial para tomadores de decisiones y profesionales* (2ª ed.). <https://www.who.int/publications/i/item/9789240069824>

Organización Mundial de la Salud. (2024). *Informe sobre la situación mundial de la seguridad vial 2023*. <https://www.who.int/es/publications/i/item/9789240086517>

Padilla, R., Netto, SL, y da Silva, EAB (2020). Un estudio sobre métricas de rendimiento para algoritmos de detección de objetos. *Actas de la Conferencia Internacional de Sistemas, Señales y Procesamiento de Imágenes (IWSSIP) de 2020*. IEEE. <https://doi.org/10.1109/IWSSIP48289.2020.9145130>

Padilla, R., Passos, WL, Dias, TLB, Netto, SL y da Silva, EAB (2021). Análisis comparativo de métricas de detección de objetos con un conjunto de herramientas complementario de código abierto. *Electronics*. <https://doi.org/10.3390/electronics10030279>

Peinado García, R. (2022). *Estudio de la calidad en las detecciones de la red YOLOv5 con transformación de imágenes* [Tesis de licenciatura]. Universidad de Málaga.

Puppo, C., Centeno, P., Rojas, P. y Serulnicoff, M. (2013). *Estudio de campo observacional urbano sobre uso de cinturón de seguridad, sistemas de retención infantil en vehículos particulares y uso de cascos en motocicletas y ciclomotores en el Estado Plurinacional de Bolivia*. Observatorio Iberoamericano de Seguridad Vial (OISEVI), Agencia Nacional de Seguridad Vial (ANSV), Policía Boliviana, Organización Panamericana de la Salud/Organización Mundial de la Salud (OPS/OMS).

Quilingking, J. y Doma, B. (2023). Detección y clasificación del uso de cascos de motocicleta en Filipinas mediante el algoritmo YOLOv5. En *CIIS '22: Actas de la 5.ª Conferencia Internacional sobre Inteligencia Computacional y Sistemas Inteligentes 2022* (pp. 21-25). Asociación para la Maquinaria de Computación. <https://doi.org/10.1145/3573428.3573432>

Ramírez, J. (2020, 15 de septiembre). Curvas PR y ROC. *Medio*. <https://medium.com/bluekiri/curvaspr-y-roc-1489fbd9a527>

Redmon, J., Divvala, S., Girshick, R. y Farhadi, A. (2016). Solo miras una vez: Detección unificada de objetos en tiempo real. *Actas de la Conferencia IEEE sobre Visión Artificial y Reconocimiento de Patrones (CVPR)*. IEEE. <https://doi.org/10.1109/CVPR.2016.91>

Rozada Raneros, DS (2021). *Estudio de la arquitectura YOLO para la detección de objetos mediante aprendizaje profundo* [Tesis de maestría]. Universidad de Valladolid.

Sarmah, R., Lahon, P. y Ahmed, T. (2024). Mayor precisión en la detección de cascos de motocicleta: enfoque YOLOv5 y Darknet. *Research Square*. <https://doi.org/10.21203/rs.3.rs-3879213/v1>

Silva Guzmán, ER (2020). *Entrenamiento de la red neuronal convolucional YOLO para objetos propios* [Informe técnico]. Universidad Nacional Autónoma de México, Programa de Maestría y Doctorado en Ingeniería.

Szeliski, R. (2022). *Visión artificial: Algoritmos y aplicaciones* (2.^a ed.). Springer. <https://doi.org/10.1007/978-3-030-34372-9>

Song, C., Zhang, Y., Liu, J. y Wang, Z. (2025). Un algoritmo YOLOv8 para la detección del uso de casco de seguridad en obras de construcción. *Scientific Reports*. <https://doi.org/10.1038/s41598-025-08828-z>

Ultralíticos. (2024). *Modelos YOLOv11*. <https://docs.ultralytics.com/es/models/yolo11/>

Ultralíticos. (2025). *Una guía sobre cómo usar Kaggle para entrenar sus modelos YOLO11*. <https://docs.ultralytics.com/es/integrations/kaggle/>

Ultralytics. (2024). *Configuración - Documentación YOLO de Ultralytics*. <https://docs.ultralytics.com/usage/cfg/>

Vera-Rubio, P. (2023). La inteligencia artificial en la educación superior: Un enfoque transformador. *Revista Polo del Conocimiento*. <https://doi.org/10.23857/pc.v8i11>

Wei, C., Tan, Z., Qing, Q., Zeng, R. y Wen, G. (2023). Detección rápida de casco y matrícula basada en el sensor ligero YOLOv5. *Sensors*. <https://doi.org/10.3390/s23094335>

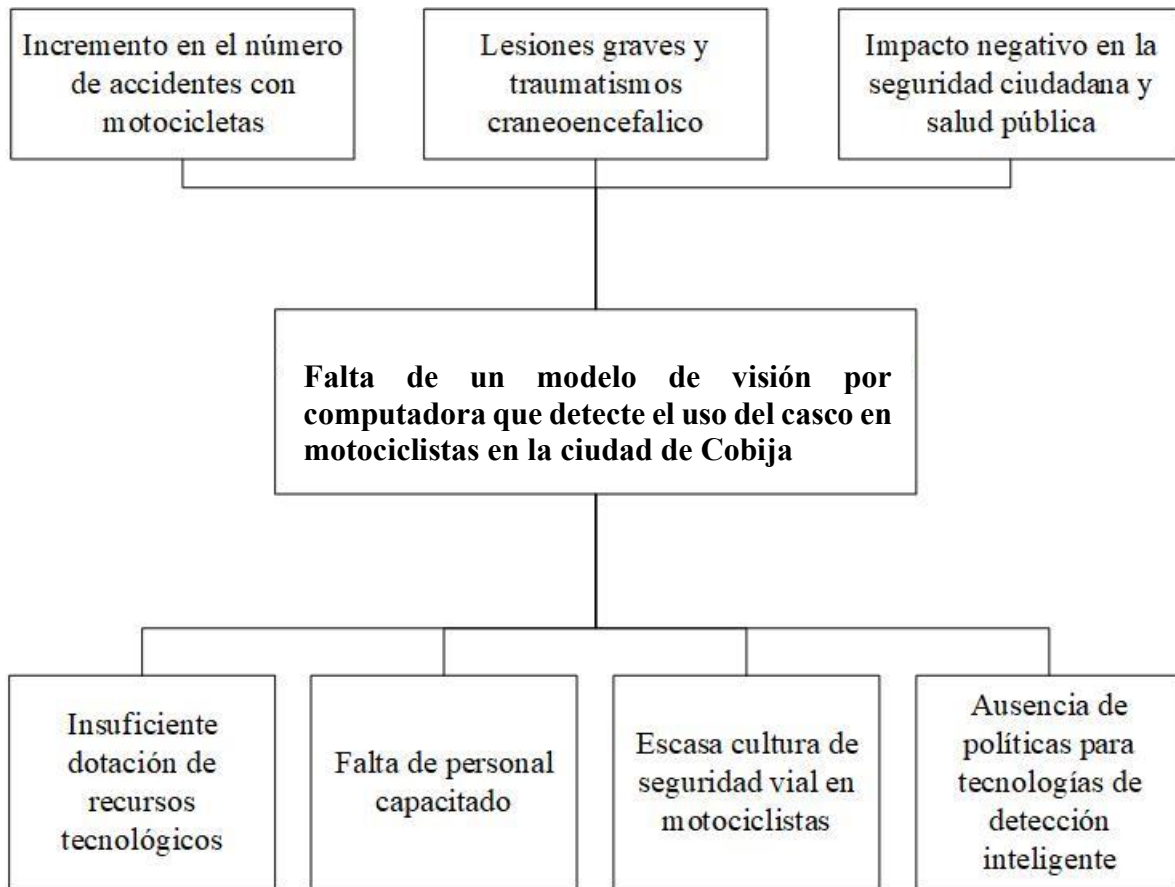
Yamashita, R., Nishio, M., Do, RKG y Togashi, K. (2018). Redes neuronales convolucionales: Resumen y aplicación en radiología. *Insights into Imaging*. <https://doi.org/10.1007/s13244-018-0639-9>

Yin, X. y Liu, X. (2021). Estrategias de selección de datos en aprendizaje profundo: Una encuesta. *Knowledge-Based Systems*. <https://doi.org/10.1016/j.knosys.2021.107212>

You, Q., y Chen, G. (2025). Detección de objetos de tráfico vial basada en el algoritmo mejorado YOLOV11m. *Patentes recientes sobre ingeniería mecánica*. Publicación anticipada en línea. <https://doi.org/10.2174/0122127976412590250730075138>

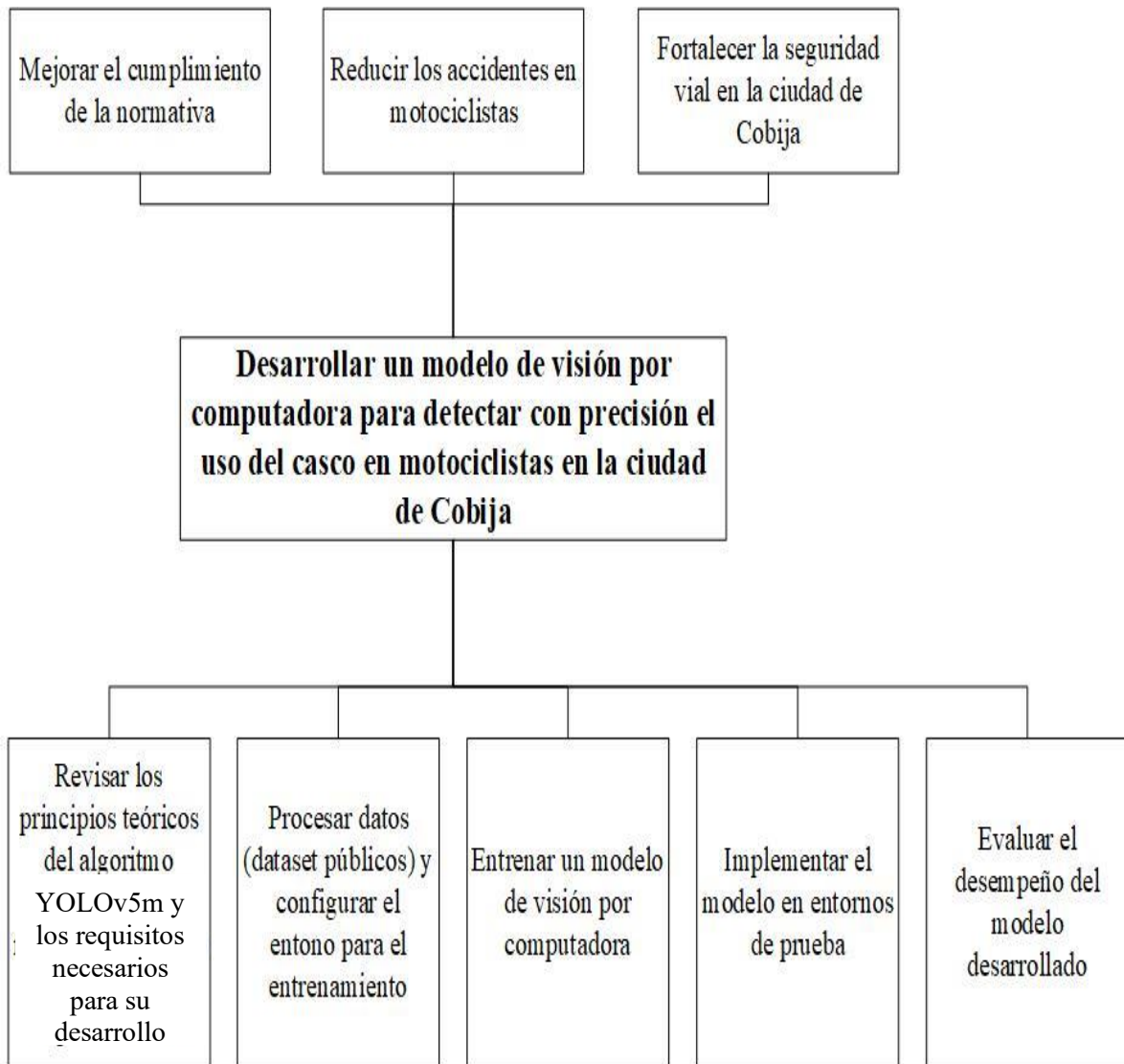
ANEXOS

Anexo A
Árbol de problemas



Anexo B

Árbol de objetivos



Anexo C

Instrumentos de recolecciones de datos

Tabla C1

Ficha de revisión del dataset público

FICHA DE RECOLECCIÓN DE DATOS INSTRUMENTO DE INVESTIGACIÓN		Nº1
<i>Ficha de revisión del dataset público:</i> Para evaluar la pertinencia del dataset que se utilizará en el entrenamiento del modelo.		
Categoría	Descripción	
Nombre del dataset		
Fuente del dataset		
Enlace		
Número total de imágenes		
Presencia de clases		
Etiquetas disponibles		
Tamaño de imagen		
Diversidad de escenarios		
Formato de anotación		
Requiere preprocesamiento		
Aprobación para entrenamiento		

Tabla C2

Registro de recolección de imágenes locales

INSTRUMENTO DE INVESTIGACIÓN FICHA DE RECOLECCIÓN DE DATOS		Nº2
Registro de recolección de imágenes: Para documentar y registrar de forma clara y organizada los detalles técnicos y contextuales de la recolección de datos del estudio.		
Categoría	Descripción	
Dispositivo/Cámara utilizado		
Numero de Imagen		
Fecha de captura		
Angulo		
Resolución		
Formato de archivo		
Lugar o zona de captura		
Horario de captura		
Tipo de iluminación		
Presencia Con casco		
Presencia Sin casco		
Presencia motociclista		
Observaciones		

Tabla C3

Registro de prueba del modelo

INSTRUMENTO DE INVESTIGACIÓN		Nº3
FICHA DE RECOLECCIÓN DE DATOS		
Registro de pruebas del modelo: Para evaluar el desempeño del modelo		
Categoría	Descripción	
Numero de épocas del modelo		
Número de imágenes que contienen al menos un objeto con casco		
Número de imágenes que contienen al menos un objeto sin casco		
Número total de imágenes		
Número de motociclistas con casco		
Número de motociclistas sin casco		
Número total de motociclistas detectados		
Valor de Precisión con casco		
Valor de Precisión sin casco		
Total valor de precisión		
Valor Recall con casco		
Valor Recall sin casco		
Total valor Recall		
Valor de Precisión promedio (AP) con casco		
Valor de Precisión promedio (AP) sin casco		
Valor Total, Precisión Media Promedio (mAP)		
Observaciones relevantes		

Anexo D

Rubrica de expertos validadores de los instrumentos de recolección de datos

Figura E1

Constancia de validación de instrumento experto 1



UNIVERSIDAD AMAZÓNICA DE PANDO
INGENIERIA DE SISTEMAS



Cobija, 24 de junio del 2025

Señor:

Ing. Roger Rolando Gutierrez Ramirez
INGENIERO DE SISTEMAS
Presente. -

REF. SOLICITUD DE VALIDACIÓN DE INSTRUMENTOS DE RECOLECCION DE DATOS

De mi mayor consideración:

Mediante la presente, me es grato dirigirme a usted para expresarle mis cordiales saludos y mis mejores deseos de éxito en sus funciones.

En virtud de su amplia experiencia y conocimiento en el área, me permito solicitar la **VALIDACIÓN DE LOS DIFERENTES INSTRUMENTO DE RECOLECCIÓN DE DATOS**, el cual será aplicado en la ciudad de Cobija. Dicho instrumento forma parte de mi investigación para la tesis titulada: "**MODELO DE DETECCIÓN DEL USO DE CASCO EN MOTOCICLISTAS MEDIANTE VISIÓN POR COMPUTADORA EN LA CIUDAD DE COBIJA**", con la cual aspiro al grado de Licenciatura en Ingeniero de Sistemas.

Agradeciendo de antemano su valioso tiempo y colaboración, quedo atenta a sus observaciones y a la posibilidad de coordinar cualquier aspecto necesario para la validación del instrumento.

Sin otro particular, me despido de usted reiterándole mi más alta estima y consideración.

Atentamente,

Jorge Rodrigo Ugarte Gutierrez

R.U.: 28529
C.I.: 7455996 or
Cel.:73552233

Recibido
27 - JUNIO - 25 14:00
S.I.S.
Roger Rolando Gutierrez Ramirez
INGENIERO DE SISTEMAS
R.N.I. 61712



**UNIVERSIDAD AMAZONICA DE PANDO
INGENIERIA DE SISTEMAS**



CONSTANCIA DE VALIDACION DE INSTRUMENTO

Yo, Rogar Rolando Gutierrez Ramirez, con CI. N° 7389748
Especialista en: Ciencias de Datos, y ejerzo la carrera profesional en:
Ingeniero de Sistemas.

Por medio de la presente hago constar que he revisado, con fines de validación de la FICHA TECNICA N° 1 de la tesis titulada: "**MODELO DE DETECCIÓN DEL USO DE CASCO EN MOTOCICLISTAS MEDIANTE VISIÓN POR COMPUTADORA EN LA CIUDAD DE COBIJA**"
Que será aplicado en el mes de julio 2025 desde el 10 al 16 de julio en el desarrollo de la investigación del cursante Est. Jorge Rodrigo Ugarte Gutierrez de la carrera de Ingeniería de Sistemas de Universidad Amazónica de Pando.

Luego de hacer las verificaciones pertinentes, puedo formular las siguientes apreciaciones:

EVALUACION DE INSTRUMENTO

N°	INDICADORES	VALORES			
		Deficiente	Regular	Bueno	Excelente
1	El instrumento presenta coherencia con el problema de investigación			✓	
2	El instrumento evidencia el problema a solucionar			✓	



UNIVERSIDAD AMAZONICA DE PANDO
INGENIERIA DE SISTEMAS



3	El instrumento guarda relación con los objetivos propuestos en la investigación.			✓	
4	El instrumento facilita la comprobación de la hipótesis que se plantea en la investigación				✓
5	Los indicadores son los correctos para cada dimensión			✓	
6	En general, el instrumento permite un manejo ágil de la información			✓	

Observaciones:

Lugar y fecha: Oruro - 30 - 06 - 2025

 Roger Rolando Gutierrez Ramirez
INGENIERO DE SISTEMAS
R.N.J. 61712

Firma y sello del Experto Informante: 

Cl. 7389748 Or. Cel.: 60420970



UNIVERSIDAD AMAZONICA DE PANDO
INGENIERIA DE SISTEMAS



CONSTANCIA DE VALIDACION DE INSTRUMENTO

Yo, Roger Rolando Gutierrez Ramirez, con CI. N° 7289748
Especialista en: Ciencias de Datos, y ejerzo la carrera profesional en:
Ingeniero de Sistemas.

Por medio de la presente hago constar que he revisado, con fines de validación de la **FICHA TECNICA N° 2** de la tesis titulada: **"MODELO DE DETECCIÓN DEL USO DE CASCO EN MOTOCICLISTAS MEDIANTE VISIÓN POR COMPUTADORA EN LA CIUDAD DE COBIJA"**
Que será aplicado en el mes de julio 2025 desde el 17 al 31 de julio en el desarrollo de la investigación del cursante Est. Jorge Rodrigo Ugarte Gutierrez de la carrera de Ingeniería de Sistemas de Universidad Amazónica de Pando.

Luego de hacer las verificaciones pertinentes, puedo formular las siguientes apreciaciones:

EVALUACION DE INSTRUMENTO

N°	INDICADORES	VALORES			
		Deficiente	Regular	Bueno	Excelente
1	El instrumento presenta coherencia con el problema de investigación			✓	
2	El instrumento evidencia el problema a solucionar			✓	
3	El instrumento guarda relación con los objetivos propuestos en la investigación.			✓	




UNIVERSIDAD AMAZONICA DE PANDO
INGENIERIA DE SISTEMAS



4	El instrumento facilita la comprobación de la hipótesis que se plantea en la investigación			✓	
5	Los indicadores son los correctos para cada dimensión				✓
6	En general, el instrumento permite un manejo ágil de la información			✓	

Observaciones:

Lugar y fecha: Oruro - 30-06-2025

 Roger Rolando Gutierrez Ramirez
INGENIERO DE SISTEMAS
R.N.I. 61712

Firma y sello del Experto Informante: 

Cl. 7389748 Or. Cel.: 60420970



**UNIVERSIDAD AMAZONICA DE PANDO
INGENIERIA DE SISTEMAS**



CONSTANCIA DE VALIDACION DE INSTRUMENTO

Yo, Roger Rolando Gutierrez Ramirez, con CI. N° 7389748
Especialista en: Ciencias de Datos y ejerzo la carrera profesional en:
Ingeniero de Sistemas.

Por medio de la presente hago constar que he revisado, con fines de validación de la FICHA TECNICA N° 3 de la tesis titulada: **"MODELO DE DETECCIÓN DEL USO DE CASCO EN MOTOCICLISTAS MEDIANTE VISIÓN POR COMPUTADORA EN LA CIUDAD DE COBIJA"**

Que será aplicado en el mes de septiembre 2025 desde el 11 al 15 de septiembre en el desarrollo de la investigación del cursante Est. Jorge Rodrigo Ugarte Gutierrez de la carrera de Ingeniería de Sistemas de Universidad Amazónica de Pando.

Luego de hacer las verificaciones pertinentes, puedo formular las siguientes apreciaciones:

EVALUACION DE INSTRUMENTO

N°	INDICADORES	VALORES			
		Deficiente	Regular	Bueno	Excelente
1	El instrumento presenta coherencia con el problema de investigación			✓	
2	El instrumento evidencia el problema a solucionar			✓	
3	El instrumento guarda relación con los objetivos propuestos en la investigación.				✓



UNIVERSIDAD AMAZONICA DE PANDO
INGENIERIA DE SISTEMAS



4	El instrumento facilita la comprobación de la hipótesis que se plantea en la investigación				✓
5	Los indicadores son los correctos para cada dimensión			✓	
6	En general, el instrumento permite un manejo ágil de la información			✓	

Observaciones:

Lugar y fecha: Oruro - 30-06-2015

 Roger Rolando Gutiérrez Ramírez
INGENIERO DE SISTEMAS
R.N.I. 61712

Firma y sello del Experto Informante: 

Cl. 7389748 Or. Cel.: 60420970

Figura E2

Constancia de validación de instrumento experto 2



UNIVERSIDAD AMAZONICA DE PANDO
INGENIERIA DE SISTEMAS



Cobija, 10 de octubre del 2025

Señora:
Ing. Delfina Villafuente Romero
INGENIERA DE SISTEMAS
Presente. -

REF. SOLICITUD DE VALIDACIÓN DE INSTRUMENTOS DE RECOLECCION DE DATOS

De mi mayor consideración:

Mediante la presente, me es grato dirigirme a usted para expresarle mis cordiales saludos y mis mejores deseos de éxito en sus funciones.

En virtud de su amplia experiencia y conocimiento en el área, me permito solicitar la **VALIDACIÓN DE LOS DIFERENTES INSTRUMENTO DE RECOLECCIÓN DE DATOS**, el cual se utilizaron en la ciudad de Cobija. Dicho instrumento forma parte de mi investigación para la tesis titulada: "**MODELO DE DETECCIÓN DEL USO DE CASCO EN MOTOCICLISTAS MEDIANTE VISIÓN POR COMPUTADORA EN LA CIUDAD DE COBIJA**", con la cual aspiro al grado de Licenciatura en Ingeniero de Sistemas.

Agradeciendo de antemano su valioso tiempo y colaboración, quedo atenta a sus observaciones y a la posibilidad de coordinar cualquier aspecto necesario para la validación del instrumento.

Sin otro particular, me despido de usted reiterándole mi más alta estima y consideración.

Atentamente,



Jorge Rodrigo Ugarte Gutierrez

R.U.: 28529
C.I.: 7455996 or
Cel.:73552233



Recibido
Delfina C. Villafuente



CONSTANCIA DE VALIDACION DE INSTRUMENTO

Yo, Delfino Ulloa Reste Romero, con CI. N° 4203139 Pdo
Especialista en: Inteligencia Artificial y ejerzo la carrera profesional en:
Ingeniería de Sistemas.

Por medio de la presente hago constar que he revisado, con fines de validación de la FICHA TECNICA N° 1 de la tesis titulada: **"MODELO DE DETECCIÓN DEL USO DE CASCO EN MOTOCICLISTAS MEDIANTE VISIÓN POR COMPUTADORA EN LA CIUDAD DE COBIJA"**

Que se utilizaron en el mes de julio 2025 desde el 10 al 16 de julio en el desarrollo de la investigación del cursante Est. Jorge Rodrigo Ugarte Gutierrez de la carrera de Ingeniería de Sistemas de Universidad Amazónica de Pando.

Luego de hacer las verificaciones pertinentes, puedo formular las siguientes apreciaciones:

EVALUACION DE INSTRUMENTO

N°	INDICADORES	VALORES			
		Deficiente	Regular	Bueno	Excelente
1	El instrumento presenta coherencia con el problema de investigación			✓	
2	El instrumento evidencia el problema a solucionar				✓



**UNIVERSIDAD AMAZONICA DE PANDO
INGENIERIA DE SISTEMAS**



3	El instrumento guarda relación con los objetivos propuestos en la investigación.				✓
4	El instrumento facilita la comprobación de la hipótesis que se plantea en la investigación			✓	
5	Los indicadores son los correctos para cada dimensión			✓	
6	En general, el instrumento permite un manejo ágil de la información				✓

Observaciones:

El instrumento es coherente, sin embargo en la ficha 1 Debuset, el volumen de 2000 imágenes es limitado para un modelo robusto. Considero que no se debe usar el mismo debuset para entendimiento y prueba

Lugar y fecha: 18/10/2025

Ing. Delfina C. Villafuerte R.
INGENIERA DE SISTEMAS
C.I. 4203139
Cebija, / / 2025

Firma y sello del Experto Informante: _____

Ci. 4203139 Cel. 67664066



CONSTANCIA DE VALIDACION DE INSTRUMENTO

Yo, Delfino Villaparte Roman, con CI. N° 4203139 Pd
Especialista en: Inteligencia Artificial y ejerzo la carrera profesional en:
Ingeniería de Sistemas.

Por medio de la presente hago constar que he revisado, con fines de validación de la FICHA TECNICA N° 2 de la tesis titulada: "MODELO DE DETECCIÓN DEL USO DE CASCO EN MOTOCICLISTAS MEDIANTE VISIÓN POR COMPUTADORA EN LA CIUDAD DE COBIJA" Que se utilizaron en el mes de julio 2025 desde el 17 al 31 de julio en el desarrollo de la investigación del cursante Est. Jorge Rodrigo Ugarte Gutierrez de la carrera de Ingeniería de Sistemas de Universidad Amazónica de Pando.

Luego de hacer las verificaciones pertinentes, puedo formular las siguientes apreciaciones:

EVALUACION DE INSTRUMENTO

N°	INDICADORES	VALORES			
		Deficiente	Regular	Bueno	Excelente
1	El instrumento presenta coherencia con el problema de investigación			✓	
2	El instrumento evidencia el problema a solucionar			✓	

UNIVERSIDAD AMAZONICA DE PANDO
INGENIERIA DE SISTEMAS



3	El instrumento guarda relación con los objetivos propuestos en la investigación.				✓
4	El instrumento facilita la comprobación de la hipótesis que se plantea en la investigación			✓	
5	Los indicadores son los correctos para cada dimensión			✓	
6	En general, el instrumento permite un manejo ágil de la información			✓	

Observaciones:

Se podría beneficiarse con alguna descripción operativa por ej:
iluminación o ángulo

Lugar y fecha: _____

Ing. Delfino C. Villafuerte P.
INGENIERA DE SISTEMAS
CI. 4203139

Firma y sello del Experto Informante: _____

Ci. 4203139 Pdo

Cel.: _____

[Signature]
62664066



**UNIVERSIDAD AMAZONICA DE PANDO
INGENIERIA DE SISTEMAS**



CONSTANCIA DE VALIDACION DE INSTRUMENTO

Yo, Dafne C Vilhuenle Romero, con CI. N° 4203139 Pa
Especialista en: Inteligencia Artificial y ejerzo la carrera profesional en:
Ingenieria de Sistemas.

Por medio de la presente hago constar que he revisado, con fines de validación de la FICHA TECNICA N° 3 de la tesis titulada: **"MODELO DE DETECCIÓN DEL USO DE CASCO EN MOTOCICLISTAS MEDIANTE VISIÓN POR COMPUTADORA EN LA CIUDAD DE COBIJA"**

Que se utilizaron en el mes de septiembre 2025 desde el 11 al 15 de septiembre en el desarrollo de la investigación del cursante Est. Jorge Rodrigo Ugarte Gutierrez de la carrera de Ingeniería de Sistemas de Universidad Amazónica de Pando.

Luego de hacer las verificaciones pertinentes, puedo formular las siguientes apreciaciones:

EVALUACION DE INSTRUMENTO

N°	INDICADORES	VALORES			
		Deficiente	Regular	Bueno	Excelente
1	El instrumento presenta coherencia con el problema de investigación				✓
2	El instrumento evidencia el problema a solucionar				✓



**UNIVERSIDAD AMAZONICA DE PANDO
INGENIERIA DE SISTEMAS**



3	El instrumento guarda relación con los objetivos propuestos en la investigación.				✓
4	El instrumento facilita la comprobación de la hipótesis que se plantea en la investigación			✓	
5	Los indicadores son los correctos para cada dimensión				✓
6	En general, el instrumento permite un manejo ágil de la información			✓	

Observaciones:

Debe incluir métricas que cuantifiquen la viabilidad operacional del modelo en la infraestructura de Cobija, esto exige registrar el rendimiento

Lugar y fecha: _____

Ing. Delfina C. Villa fuerte R.
INGENIERA DE SISTEMAS
C.I. 4203139

Cobija,/202.....

Firma y sello del Experto Informante: 

CI. 4203139 Pdo Cel. 97664066



Cobija, 10 de octubre del 2025

Señor:
Ing. Luis Enrique Villca Mamani
INGENIERO DE SISTEMAS
Presente. -

REF. SOLICITUD DE VALIDACIÓN DE INSTRUMENTOS DE RECOLECCION DE DATOS

De mi mayor consideración:

Mediante la presente, me es grato dirigirme a usted para expresarle mis cordiales saludos y mis mejores deseos de éxito en sus funciones.

En virtud de su amplia experiencia y conocimiento en el área, me permito solicitar la **VALIDACIÓN DE LOS DIFERENTES INSTRUMENTO DE RECOLECCIÓN DE DATOS**, el cual se utilizaron en la ciudad de Cobija. Dicho instrumento forma parte de mi investigación para la tesis titulada: **"MODELO DE DETECCIÓN DEL USO DE CASCO EN MOTOCICLISTAS MEDIANTE VISIÓN POR COMPUTADORA EN LA CIUDAD DE COBUJA"**, con la cual aspiro al grado de Licenciatura en Ingeniero de Sistemas.

Agradeciendo de antemano su valioso tiempo y colaboración, quedo atenta a sus observaciones y a la posibilidad de coordinar cualquier aspecto necesario para la validación del instrumento.

Sin otro particular, me despido de usted reiterándole mi más alta estima y consideración.

Atentamente,

Jorge Rodrigo Ugarte Gutierrez

R.U.: 28529
C.I.: 7455996 or
Cel.:73552233

Ing. Luis Enrique Villca Mamani
Ingeniero de Sistemas de
Información y Comunicación (ISI)
Universidad Amazónica de Pando

16 / 10 / 2025

11 01

Anexo E

Recopilación y preparación del conjunto de datos

La construcción del conjunto de datos implementó una estrategia dual que combinó un dataset público con imágenes locales de Cobija. Esta aproximación equilibró tres requisitos fundamentales: volumen suficiente para entrenamiento robusto, diversidad visual para capacidad de generalización y validación contextualizada en condiciones locales específicas.

El componente público provino del repositorio Roboflow Universe, específicamente el dataset "motociclista-con-y-sin-casco3" con 1,592 imágenes previamente anotadas en formato YOLO. Este conjunto, recolectado por la comunidad, incorpora variabilidad natural en ángulos de cámara, distancias, densidad de tráfico e iluminación. Define tres clases: "con_casco" (cascos protectores visibles), "sin_casco" (cabezas descubiertas) y "motociclista" (vehículo completo con ocupante).

Tabla E1: Resumen de *datasets públicos*

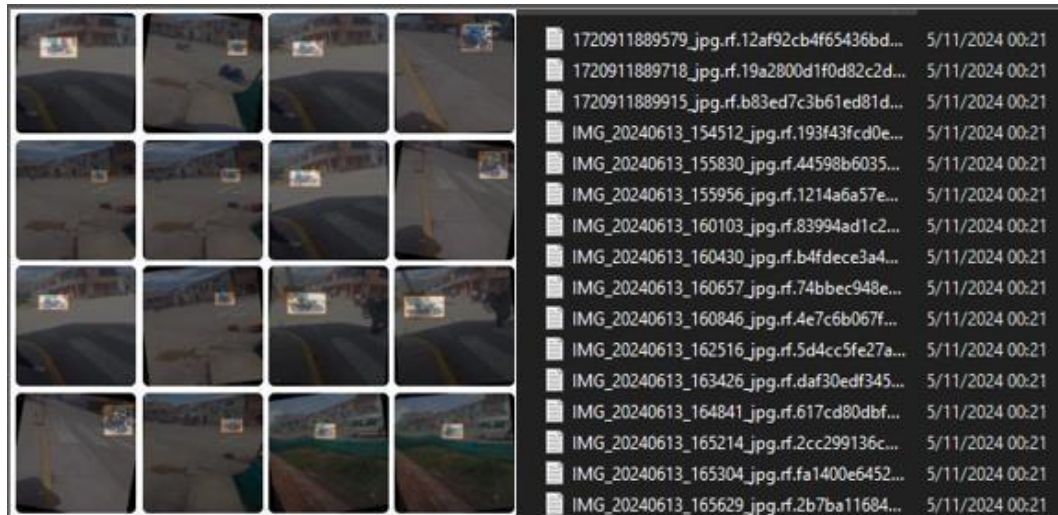
Nº	Dataset	Plataforma	Total, imágenes	Clases	Anotación	Estado
1	detection-of-motorcyclists-without-helmet	GitHub	665	Con casco, Sin casco	Caja delimitadora	No aprobado
2	Detección de casco	Kaggle	764	Con casco, Sin casco	Caja delimitadora	No aprobado
3	motociclista-con-y-sin-casco3	RoboFlow	1.592	Con casco, Sin casco, Motociclista	Caja delimitadora	Sí aprobado

Nota. La tabla muestra la número de datasets públicos encontrados y sus volúmenes de datos. Fuente: Elaboración propia, 2025.

La tabla del Anexo F1 documenta características técnicas del dataset público: procedencia, licenciamiento CC BY 4.0, formato de anotación, resoluciones predominantes y

distribución de instancias por clase. Su selección se fundamentó en volumen adecuado para transfer learning, presencia de las tres clases necesarias y calidad verificada de anotaciones.

Figura E1: *Muestra representativa dataset publico*



Nota. La ilustración muestra la variabilidad y etiquetas del conjunto de datos publico. Fuente: Elaboración propia, 2025.

En la Figura E1 se muestra que por cada imagen tiene un archivo de texto en donde están anotadas todas las etiquetas de cada clase en donde están las coordenadas.

El dataset se dividió en 80% entrenamiento (1,274 imágenes) para ajustar parámetros del modelo y 20% validación (318 imágenes) para monitorear rendimiento durante entrenamiento. Esta proporción 80/20 es óptima para conjuntos de tamaño moderado. Ninguna imagen pública se empleó como prueba final, reservándose exclusivamente las imágenes locales para evaluación independiente.

Paralelamente se recolectaron 200 imágenes originales en Cobija mediante cámara móvil Infinix Smart 8 (13MP, resolución 1920×1080, formato JPG).

Figura E1: *Muestra representativa del dataset local*



Nota. La ilustración muestra la variabilidad del conjunto de datos local. Fuente: Elaboración propia, 2025.

La figura del Anexo E1 muestra seis imágenes representativas ilustrando variabilidad del conjunto: motociclistas con/sin casco en ángulos frontal, lateral y posterior, diversidad de motocicletas locales, variaciones en densidad de tráfico y características de infraestructura vial de Cobija.

Las ubicaciones de captura fueron Av. 9 de Febrero entre Av. Otto Felipe Braun y Av. Pando, y Av. 27 de Mayo, seleccionadas por ser arterias de alta circulación de motocicletas. En tres franjas horarias diurnas: mañana, mediodía y tarde, Todas las capturas fueron exclusivamente diurnas con luz natural, sin flash ni escenarios nocturnos.

Anexo F

Etiquetado del conjunto de datos locales

Las 200 imágenes locales requirieron etiquetado manual riguroso para establecer la verdad fundamental (ground truth) contra la cual evaluar el modelo. El proceso se ejecutó mediante Roboflow, plataforma con interfaz gráfica para trazar cajas delimitadoras y asignar etiquetas de clase. Su selección respondió a compatibilidad con formato YOLO.

El formato YOLO codifica información en archivos .txt, uno por imagen con mismo nombre base. Cada línea representa un objeto: número entero de clase seguido de cuatro valores flotantes (coordenadas normalizadas del centro X,Y y dimensiones ancho/alto). Esta normalización garantiza independencia de resolución absoluta.

Tabla F1: *Codificación de clases*

Variable / Concepto	Código	Definición operativa	Función en la detección
Uso de casco (con)	0	Cabeza con casco visible en el área del rostro	Clase positiva; indica cumplimiento de la norma
Uso de casco (sin)	1	Cabeza sin casco visible	Clase negativa; indica incumplimiento de la norma
Motociclista	2	Cuerpo completo/motocicleta	Clase de apoyo para proporcionar contexto del conductor o pasajero

Nota. La tabla muestra los códigos asociados a cada clase definida. Fuente: Elaboración propia, 2025

La tabla del Anexo F1 documenta la codificación: 0="con_casco" (casco protector visible cubriendo cabeza), 1="sin_casco" (cabeza descubierta o protección inadecuada como gorras), 2="motociclista" (vehículo completo con ocupante).

El protocolo de etiquetado estableció procedimiento sistemático: identificar todos los motociclistas, trazar caja ajustada al contorno del motociclista completo, luego etiquetar específicamente región craneal como "con_casco" o "sin_casco" según corresponda. En casos de oclusión completa o imposibilidad de determinar uso de casco por distancia/ángulo, se etiquetó solo el motociclista completo evitando anotaciones ambiguas.

Figura F2: Ejemplo visual de imágenes etiquetado



Nota. La ilustración muestra las etiquetas por clase con sus valores numéricos según codificación asignada. Fuente: Elaboración propia, 2025.

La figura F1 ilustra un ejemplo mostrando imagen con cajas delimitadoras trazadas y su archivo de texto correspondiente con coordenadas normalizadas, facilitando comprensión de la transformación de información visual a datos cuantitativos procesables.

Instrucción de etiquetado:

- Delimitar el bounding box ajustado al contorno visible del motociclista (incluyendo torso y cabeza).
- Asignar "con casco" solo si el casco cubre completamente la cabeza y es claramente visible.
- Asignar "sin casco" cuando no hay protección craneal o esta es inadecuada (gorra, pañuelo, etc.).

- No etiquetar objetos secundarios (peatones, vehículos de fondo, cascos sueltos).
- Validar anotaciones en modo revisión para consistencia.

El proceso generó 638 instancias etiquetadas totales, distribuidas en 287 "motociclista", 100 "con_casco" y 251 "sin_casco". Esta cifra superior al número de imágenes refleja que cada escena contiene frecuentemente múltiples motociclistas. El desbalance hacia "sin_casco" refleja la realidad observada de incumplimiento prevalente en Cobija, proporcionando evaluación más realista del rendimiento en condiciones operativas reales.

Tabla F2: *Distribución de instancias etiquetadas por clase*

Clase	Cantidad de instancias
motociclista	287
con casco	100
sin casco	251
Total etiquetas	638

Nota. La tabla muestra las cantidades asociados a cada clase. Fuente: Elaboración propia, 2025.

La figura F2 ilustra un ejemplo mostrando imagen con cajas delimitadoras trazadas y su archivo .txt correspondiente con coordenadas normalizadas, facilitando comprensión de la transformación de información visual a datos cuantitativos procesables.

El proceso generó 638 instancias etiquetadas totales, distribuidas en 287 "motociclista", 100 "con_casco" y 251 "sin_casco". Esta cifra superior al número de imágenes refleja que cada escena contiene frecuentemente múltiples motociclistas. El desbalance hacia "sin_casco" refleja la realidad observada de incumplimiento prevalente en Cobija, proporcionando evaluación más realista del rendimiento en condiciones reales.

Anexo G

Configuración experimental y entrenamiento del modelo

El diseño cuasiexperimental manipuló sistemáticamente el número de épocas de entrenamiento como variable independiente, evaluando su efecto sobre el rendimiento (variable dependiente medida como mAP). Se entrenaron diez modelos (A-J) con épocas de 10 a 100 en incrementos de 10, para identificando empíricamente el punto óptimo que maximiza rendimiento antes del sobreajuste.

Tabla G1: *Configuración experimental por modelo*

Modelos	A	B	C	D	E	F	G	H	I	J
Épocas	10	20	30	40	50	60	70	80	90	100

Nota. La tabla muestra los diferentes estímulos de número de épocas por cada modelo. Fuente: Elaboración propia, 2025.

La tabla documenta la asignación específica de épocas por modelo (A=10, B=20... J=100), estableciendo trazabilidad completa. Esta manipulación mediante incrementos uniformes permitió caracterizar la curva de aprendizaje y detectar patrones de mejora o degradación.

El entrenamiento se ejecutó en Kaggle con dos GPU NVIDIA Tesla T4 (16GB cada una, 32GB total), permitiendo procesamiento paralelo que redujo significativamente tiempos de cómputo. El entorno incluyó Linux, Python 3.10, PyTorch y Ultralytics.

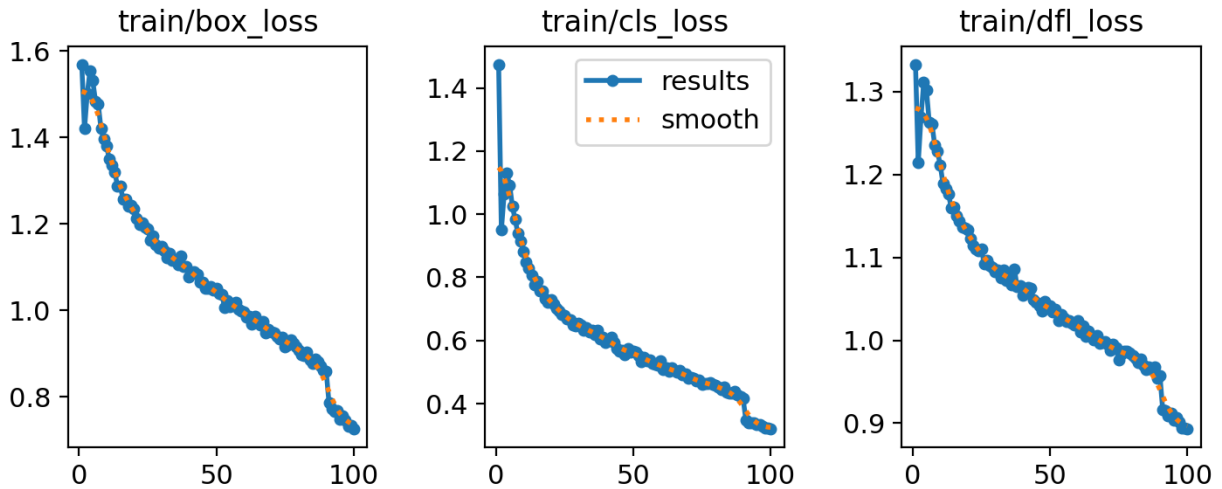
La arquitectura seleccionada fue YOLOv11m (versión medium), equilibrio óptimo entre precisión y eficiencia computacional. El modelo se inicializó mediante transfer learning cargando pesos pre-entrenados.

La configuración de hiperparámetros se mantuvo constante en todos los modelos: batch size 8 imágenes, tasa de aprendizaje inicial 0.01 con ajuste automático progresivo, tamaño de entrada 640×640 píxeles.

El entrenamiento operó mediante retropropagación del error con descenso de gradiente estocástico. En cada época se procesan las 1,274 imágenes en lotes de 8. Para cada lote, la red

ejecuta pasada hacia adelante generando predicciones, las compara con anotaciones mediante funciones de pérdida (`box_loss` para localización espacial, `cls_loss` para clasificación, `dfl_loss` para límites de cajas), calcula gradientes y actualiza pesos para reducir error.

Figura G1: *Resumen de pérdida de entrenamiento*



Nota. La ilustración muestra las diferentes graficas lineales de perdida de entrenamiento.

Fuente: Elaboración propia, 2025.

La Figura G1 ilustra la evolución de tres funciones de pérdida del modelo YOLOv11m durante 100 épocas de entrenamiento: `box_loss` (localización de cajas delimitadoras), `cls_loss` (clasificación de objetos) y `dfl_loss` (distribución focal para refinamiento de bordes).

El `box_loss` mide la precisión espacial de las cajas delimitadoras alrededor de los objetos detectados. Inició en 1.6 y descendió progresivamente hasta estabilizarse en 0.8, reflejando una mejora sostenida en la capacidad del modelo para localizar objetos con exactitud. La `cls_loss` evalúa el error en la clasificación de categorías (casco vs. no casco), partiendo de 1.4 y descendiendo hasta 0.4, lo que evidencia un aprendizaje efectivo para distinguir entre clases. Por su parte, la `dfl_loss` refina la distribución de probabilidad en los límites de las cajas, iniciando en 1.3 y alcanzando 0.9, mejorando la precisión en casos complejos o difusos.

La convergencia observada en las tres curvas sugiere que el modelo alcanzó un punto de estabilidad donde el aprendizaje de características adicionales se vuelve marginal. La

ausencia de fluctuaciones abruptas o incrementos inesperados en las pérdidas indica que no se presentó sobreajuste temprano (overfitting), lo cual es fundamental para asegurar la capacidad de generalización del modelo sobre datos no vistos. Este comportamiento confirma que el proceso de entrenamiento fue eficiente y que el modelo logró un equilibrio entre la minimización del error en el conjunto de entrenamiento y su desempeño en condiciones reales.

Anexo H

Implementación para la Evaluación de los Modelos

La implementación para evaluación se ejecutó en Google Colab, plataforma en la nube que proporciona acceso a recursos computacionales con GPU necesarios para validar el rendimiento de los diez modelos entrenados. La selección de este entorno respondió a su capacidad de ejecutar evaluaciones independientes de manera reproducible, separando claramente la fase de entrenamiento (Kaggle) de la fase de evaluación.

El proceso de implementación siguió un procedimiento sistemático replicado para cada uno de los diez modelos. Primero se configuró el entorno mediante la instalación de la librería Ultralytics, que integra la arquitectura YOLOv11m y proporciona funcionalidades completas para cargar modelos pre-entrenados y ejecutar evaluaciones cuantitativas mediante métricas estándar:

```
!pip install ultralytics
from ultralytics import YOLO
import os
```

La instalación mediante pip descarga automáticamente la librería junto con sus dependencias necesarias como PyTorch para operaciones de aprendizaje profundo y herramientas de cálculo de métricas de detección de objetos. El módulo os permite gestionar las rutas de archivos dentro del sistema, facilitando el acceso a los modelos almacenados y al dataset de prueba.

Posteriormente se procedió a cargar cada modelo entrenado especificando la ruta del archivo de pesos correspondiente en formato .pt. Una vez cargado el modelo, se ejecutó la validación sobre el conjunto de prueba local mediante el método val() proporcionado por Ultralytics:

```
model = YOLO(model_path)
results = model.val(
    data=data_yaml_path,
    split='test',
    project=results_dir,
    name='val_test_10'
)
```

Este comando especifica la ruta del archivo de configuración data.yaml que define la estructura del dataset y las clases a detectar, el parámetro split='test' para evaluar específicamente sobre el conjunto de prueba de 200 imágenes locales, y establece el directorio de proyecto donde se almacenarían los resultados junto con un nombre identificador único para cada evaluación.

El comando model.val() ejecuta automáticamente todo el proceso de evaluación sin requerir intervención manual adicional. Internamente, el método realiza las siguientes operaciones:

- Lectura de las 200 imágenes del conjunto de prueba.
- Detección de objetos en cada imagen aplicando el modelo cargado.
- Comparación de las detecciones predichas con las etiquetas reales anotadas.
- Cálculo de métricas de rendimiento completas.

Los resultados de cada evaluación se almacenan automáticamente en el directorio de resultados especificado, generando archivos que documentan las métricas calculadas (mAP, precisión, recall) y las matrices de confusión en formato visual. Este almacenamiento automático garantiza la trazabilidad completa del proceso experimental y facilita el análisis posterior comparativo entre los diez modelos evaluados.

Tabla II: Componentes técnicos del entorno de implementación

Componente	Especificación	Función en la implementación
Modelo de IA	YOLOv11m (.pt)	Motor de inferencia para las pruebas
Plataforma de inferencia	Google Colab (GPU T4)	Ejecución independiente y reproducible
Librería principal	Ultralytics	Carga y validación de modelos YOLOv11m
Dataset de evaluación	200 imágenes locales etiquetadas (Cobija)	Validar desempeño real del modelo

Archivo de configuración	data.yaml	Define la estructura y clases de los datos
Formato de salida	Métricas, curvas, visualizaciones	Documentación del desempeño del modelo

Nota. Esta tabla describe la implementación de los diferentes componentes para su correcta evaluación con imágenes inéditas de Cobija.

La tabla II sintetiza los componentes técnicos esenciales del entorno de implementación. El modelo YOLOv11m en formato .pt actúa como motor de inferencia ejecutando las predicciones. La plataforma Google Colab con GPU Tesla T4 proporciona la capacidad computacional para evaluaciones reproducibles. La librería Ultralytics cumple la función de cargar y validar los modelos mediante su interfaz de alto nivel. El dataset de 200 imágenes locales etiquetadas de Cobija valida el desempeño real en el contexto objetivo. El archivo data.yaml define la estructura organizacional de los datos y las clases a detectar. El formato de salida genera métricas cuantitativas, curvas de rendimiento y matrices de confusión que documentan exhaustivamente el desempeño alcanzado.

La implementación completa de los diez modelos generó conjuntos independientes de resultados que permitieron establecer la relación entre el número de épocas de entrenamiento y el rendimiento alcanzado sobre datos reales del contexto de Cobija, completando exitosamente la fase de desarrollo y dejando el sistema preparado para el análisis comparativo de resultados que se presenta en las secciones subsiguientes.

Anexo I

Código Fuente de Entrenamiento

En esta etapa se implementó el script responsable de la descarga del conjunto de datos y del entrenamiento del modelo YOLOv11m. En primer lugar, se instaló explícitamente el paquete de Roboflow, utilizado para gestionar el dataset remoto, y el paquete de Ultralytics, empleado para la carga y entrenamiento del modelo de detección de objetos. A continuación, se importaron las clases necesarias de cada biblioteca:

```
!pip install roboflow ultralytics
from roboflow import Roboflow
from ultralytics import YOLO
```

Seguidamente, se estableció la conexión con el espacio de trabajo de Roboflow mediante una clave de API asociada al proyecto. Dentro de este espacio se seleccionó el proyecto específico de detección de motociclistas con y sin casco y, en él, la versión 1 del conjunto de datos. Esta versión se descargó en formato directo para YOLOv11, lo que generó la estructura de directorios y el archivo de configuración data.yaml requerido para el entrenamiento:

```
rf = Roboflow(api_key="4OPr8CjYdi4JSvtN6c6j")
project = rf.workspace("jr-lpzd5").project("motociclista-con-y-
sin-casco3-e68dd")
version = project.version(1)
dataset = version.download("yolov11")
```

Una vez disponible el conjunto de datos en el entorno de ejecución, se procedió a cargar la arquitectura YOLOv11m con pesos pre-entrenados, adoptando un enfoque de aprendizaje por transferencia. De forma complementaria, se definió la ruta hacia el archivo data.yaml generado por Roboflow, que contiene la descripción de las rutas de las imágenes y de las clases utilizadas en el entrenamiento.

```
model = YOLO("yolo11m.pt")

data_path = "/kaggle/working/motociclista-con-y-sin-casco3-
1/data.yaml"
```

Finalmente, se ejecutó el proceso de entrenamiento del modelo, parametrizando los elementos centrales del experimento. En particular, se configuró el entrenamiento para 100 épocas, con imágenes redimensionadas a 640×640 píxeles y un tamaño de lote de 8 imágenes, lo que define el número de iteraciones y la granularidad con la que el modelo ajusta sus pesos. Asimismo, se especificó el directorio de salida donde se almacenan los resultados del entrenamiento y se indicó el uso de dos unidades de procesamiento gráfico para acelerar la optimización.

```
results = model.train(  
    data=data_path,  
    epochs=100,  
    imgsz=640,  
    batch=8,  
    workers=1,  
    amp=True,  
    cache=True,  
    project='/kaggle/working/runs/train',  
    name='motociclistas_v5_100',  
    augment=True,  
    mosaic=1.0,  
    mixup=0.2,  
    device='0,1' )
```

En conjunto, este bloque de código implementa el pipeline completo necesario para el experimento: descarga del conjunto de datos en formato YOLOv11, inicialización del modelo YOLOv11m y entrenamiento durante 100 épocas divididas de 10 en 10 con imágenes de 640×640 píxeles y lotes de 8 instancias, bajo una configuración consistente con los objetivos de rendimiento planteados para el modelo.

Anexo J

Código Fuente Test de Rendimiento

En esta etapa se implementó el script de evaluación del rendimiento del modelo entrenado sobre el conjunto de prueba independiente. Primero se montó Google Drive en el entorno de ejecución de Colab, con el fin de acceder tanto a los archivos de pesos del modelo como al conjunto de datos de prueba y al directorio donde se almacenarían los resultados de la validación

```
# Enlace del modelo de detección  
https://drive.google.com/drive/folders/1MxpQ4e141AwMC-  
r1LJpHWarunH0NJ392?usp=sharing  
from google.colab import drive  
drive.mount('/content/drive')
```

```
!pip install ultralytics
```

```
from ultralytics import YOLO  
import os
```

A continuación, se definieron explícitamente tres rutas clave: la ubicación del archivo de pesos correspondiente al modelo entrenado para 10 épocas, la ruta del archivo data.yaml asociado al conjunto de prueba, y el directorio donde se guardarían los resultados de la validación. Antes de ejecutar la evaluación, se verificó la existencia de estos archivos para evitar errores de ejecución, y se creó, en caso de ser necesario, la carpeta destinada a los resultados.

```
model_path = '/content/drive/MyDrive/Colab Notebooks/10 en 10 hasta  
100/epoch10.pt'
```

```
data_yaml_path = '/content/drive/MyDrive/Colab Notebooks/10 en 10  
hasta 100/datatest/data.yaml'
```

```
results_dir = '/content/drive/MyDrive/Colab Notebooks/10 en 10 hasta  
100/RESULTADOS'
```

```
if not os.path.exists(model_path):  
    raise FileNotFoundError(f"Modelo no encontrado: {model_path}")  
if not os.path.exists(data_yaml_path):
```

```
raise FileNotFoundError(f"YAML no encontrado: {data_yaml_path}")

os.makedirs(results_dir, exist_ok=True)
```

Una vez validadas las rutas, se cargó en memoria el modelo YOLOv11m entrenado, utilizando el archivo de pesos previamente generado en la fase de entrenamiento. Sobre esta instancia del modelo se ejecutó el modo de validación, indicando como fuente de datos el archivo data.yaml del conjunto de prueba, especificando el “split” de evaluación como test y definiendo el directorio y el nombre de la carpeta donde se almacenarían las métricas y artefactos generados

```
model = YOLO(model_path)

results = model.val(
    data=data_yaml_path,
    split='test',
    project=results_dir,
    name='val_test_10'
)
```

Este bloque de código realiza, de manera automatizada, la evaluación cuantitativa del modelo entrenado sobre un conjunto de imágenes no vistas, generando métricas técnicas como precisión, recall y mAP, así como salidas gráficas (matrices de confusión) que permiten analizar el comportamiento del modelo para cada clase en el contexto de prueba definido.

Anexo K

Código Fuente ejecución básica

Para la ejecución interactiva del modelo entrenado se desarrolló una interfaz básica en Python, basada en la biblioteca Tkinter para la interfaz gráfica y en la librería Ultralytics para la inferencia con YOLOv11m. El script inicia declarando los metadatos del archivo y cargando las dependencias necesarias: manejo de rutas, procesamiento de imágenes, creación de ventanas, conversión de formatos y uso del modelo de detección.

```
import os
import cv2
import tkinter as tk
from tkinter
import filedialog
from PIL
import Image, ImageTk
from ultralytics
import YOLO
import numpy as np
```

A continuación, se carga el modelo YOLO correspondiente al mejor experimento identificado en la fase de entrenamiento (50 épocas), almacenado en el archivo de pesos epoch50OFI.pt. Además, se declaran variables globales que almacenan la imagen original cargada y su ruta en disco, las cuales serán reutilizadas por las funciones de la interfaz.

```
# Cargar modelo YOLO
model = YOLO("epoch50OFI.pt")

# Variables globales
imagen_original = None
ruta_imagen = None
```

Posteriormente, se inicializa la ventana principal de la aplicación gráfica. Se define el título, el tamaño de la ventana y el color de fondo. Adicionalmente, se intenta cargar una imagen de bienvenida (yolo_pipeline.png) que funciona como banner visual en la parte superior de la interfaz, sin bloquear la ejecución en caso de que el archivo no esté disponible.

```
# Crear ventana principal
root = tk.Tk()
root.title("Detección de Motociclistas con YOLO")
root.geometry("1100x800") # ventana más grande
root.configure(bg = "#f0f0f0")
```

```

#📁 Imagen de bienvenida
try:
    banner_img = Image.open("yolo_pipeline.png")
    banner_img = banner_img.resize((500, 80), Image.Resampling.LANCZOS)
    banner_tk = ImageTk.PhotoImage(banner_img)
    banner_label = tk.Label(root, image = banner_tk, bg = "#f0f0f0")
    banner_label.image = banner_tk
    banner_label.grid(row = 0, column = 0, columnspan = 2, pady = (10,
0))
except Exception as e:
    print("No se pudo cargar la imagen de bienvenida:", e)

```

La interfaz se organiza en dos marcos principales: uno para mostrar la imagen original seleccionada por el usuario y otro para visualizar la salida del modelo con las detecciones superpuestas. En cada marco se coloca una etiqueta (label) donde se renderizarán las imágenes. Adicionalmente, se incorporan tres botones: uno para cargar una imagen desde el sistema de archivos, otro para ejecutar la detección y un tercero para cerrar la aplicación. Se incluye también un área de texto destinada a mostrar, en formato legible, las clases detectadas y sus niveles de confianza.

```

# Marcos visuales
frame_left = tk.LabelFrame(root, text = " Imagen original", width =
500, height = 500, bg = "white")
frame_left.grid(row = 1, column = 0, padx = 10, pady = 10)
frame_left.grid_propagate(False)

frame_right = tk.LabelFrame(root, text = " Detección YOLO", width =
500, height = 500, bg = "white")
frame_right.grid(row = 1, column = 1, padx = 10, pady = 10)
frame_right.grid_propagate(False)

label_left = tk.Label(frame_left, bg = "white")
label_left.pack(expand = True)

label_right = tk.Label(frame_right, bg = "white")
label_right.pack(expand = True)

#🔘 Botones
boton_cargar = tk.Button(root, text = " CARGAR IMAGEN", command =
lambda: cargar_imagen(),
font = ("Arial", 11), bg = "#007acc", fg = "white", padx = 10,
pady = 5)
boton_cargar.grid(row = 2, column = 0, pady = (5, 0))

boton_calcular = tk.Button(root, text = " CALCULAR", command =
lambda: calcular_deteccion(),
font = ("Arial", 11), bg = "#28a745", fg = "white", padx = 10,
pady = 5)

```

```

boton_calcular.grid(row = 2, column = 1, pady = (5, 0))

boton_salir = tk.Button(root, text = " SALIR", command = lambda:
salir(),
font = ("Arial", 11), bg = "#dc3545", fg = "white", padx = 10,
pady = 5)
boton_salir.grid(row = 2, column = 4, colspan = 2, pady = (10,
20))

#📄 Área de texto para mostrar resultados(más grande y con negrilla)
resultado_texto = tk.Text(root, height = 6, width = 140, font =
("Arial", 10))
resultado_texto.grid(row = 4, column = 0, colspan = 2, padx = 10,
pady = (5, 10))
resultado_texto.tag_configure("negrita", font = ("Arial", 10,
"bold"))

```

La función `cargar_imagen()` implementa el flujo de selección e incorporación de una imagen en la aplicación. Mediante un cuadro de diálogo se permite al usuario elegir un archivo en formato común (JPG, PNG, JPEG). La imagen se lee con OpenCV, se convierte a espacio de color RGB y se redimensiona a 480×480 píxeles para su proyección en el marco izquierdo. Además, se registra el nombre del archivo en el área de texto para confirmar la correcta carga.

```

def cargar_imagen():
    global imagen_original, ruta_imagen
    ruta_imagen = filedialog.askopenfilename(filetypes = [("Imagen",
"*.*jpg *.png *.jpeg")])
    if ruta_imagen:
        imagen_original = cv2.imread(ruta_imagen)
    if imagen_original is None:
        resultado_texto.insert(tk.END, " Error al cargar la imagen.\n")
    return
    image_rgb = cv2.cvtColor(imagen_original, cv2.COLOR_BGR2RGB)
    img_pil = Image.fromarray(image_rgb)
    img_pil = img_pil.resize((480, 480), Image.Resampling.LANCZOS)
    img_tk = ImageTk.PhotoImage(img_pil)
    label_left.config(image = img_tk)
    label_left.image = img_tk
    resultado_texto.delete("1.0", tk.END)
    resultado_texto.insert(tk.END, f " Imagen cargada:
{os.path.basename(ruta_imagen)}\n")

```

La función `calcular_deteccion()` ejecuta la inferencia del modelo sobre la imagen cargada. Si no hay imagen disponible, se notifica al usuario. En caso contrario, la imagen se envía al modelo YOLO, se obtiene la salida con las detecciones y se genera una versión anotada de la imagen (con cuadros delimitadores y etiquetas), que se muestra en el marco derecho. En

paralelo, se recorre cada una de las cajas detectadas y se registran en el área de texto la clase identificada y su nivel de confianza.

```
def calcular_deteccion():
    global imagen_original
    if imagen_original is None:
        resultado_texto.insert(tk.END, " Primero cargue una imagen.\n")
    return

results = model(imagen_original)
annotated = results[0].plot()

annotated_rgb = cv2.cvtColor(annotated, cv2.COLOR_BGR2RGB)
annotated_pil = Image.fromarray(annotated_rgb)
annotated_pil = annotated_pil.resize((480, 480),
Image.Resampling.LANCZOS)
annotated_tk = ImageTk.PhotoImage(annotated_pil)
label_right.config(image = annotated_tk)
label_right.image = annotated_tk

resultado_texto.delete("1.0", tk.END)
resultado_texto.insert(tk.END, " Resultados de detección:\n",
"negrita")
for r in results[0].boxes:
    cls = results[0].names[int(r.cls[0])]
    conf = float(r.conf[0])
    resultado_texto.insert(tk.END, f " {cls}: precisión {conf:.2f}\n")
```

Finalmente, se define una función simple para cerrar la y se inicia el bucle principal de Tkinter, que mantiene activa la ventana hasta que el usuario decida terminar la ejecución.

```
def salir():
    root.destroy()

root.mainloop()

root.mainloop()
```

Este módulo constituye la interfaz de ejecución básica, permitiendo cargar imágenes individuales, visualizar simultáneamente la imagen original y las detecciones generadas por el modelo YOLOv11m, y consultar en texto las clases identificadas y sus niveles de confianza para cada instancia detectada.