



Facultad de Ingeniería
Universidad de Concepción

Universidad de Concepción
Facultad de Ingeniería
Departamento de Ingeniería
Informática y Ciencias de la Computación

Detección Automática de Carga de Rollizos por medio de Visión Artificial

POR

Dayan Antonio Sáez Cubillos

Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de Concepción
para optar al título profesional de Ingeniero Civil Informático

Profesor Patrocinante: Ricardo Flores Huenchullanca

Supervisores: Fernando Flores Cerna
Víctor Sanhueza Salazar

Concepción, 31 de julio de 2025

© 2025 Dayan Antonio Sáez Cubillos

Índice

1. Introducción	3
Presentación del problema	4
2. Objetivos	6
2.1 Objetivo general	6
2.2 Objetivos específicos.....	6
3. Marco teórico	7
3.1 Estado del arte	7
3.2 Evaluación de alternativas.....	11
3.2.1 Evaluación de alternativas para OE1	11
3.2.2 Evaluación de alternativas para OE2	13
3.2.3 Evaluación de alternativas para OE3	15
4. Metodología	19
5. Desarrollo del proyecto	21
6. Evaluación experimental.....	33
6.1 Diseño de experimentos	33
6.2 Desarrollo de experimentos.....	36
6.3 Análisis de resultados.....	38
7. Operación.....	42
7.1 Estrategia de implementación	42
7.2 Arquitectura e integración del modelo	43
7.2.1 Arquitectura del sistema e interoperabilidad	43
7.2.2 Flujo y acceso a datos	45
7.2.3 Plan de integración y despliegue técnico	45
7.3 Mantenimiento y monitoreo continuo	46
7.4 Consideraciones adicionales de operación	47
8. Análisis de impacto	48
8.1 Análisis de impacto económico.....	48
8.2 Análisis de impacto ambiental	48
8.3 Análisis de implicaciones éticas y responsabilidad social	49
9. Conclusiones	51
10. Bibliografía	53

1. Introducción

El robo de madera es un tipo de crimen organizado que hoy está presente en la macrozona centro sur y sur generando peligrosidad a la estabilidad económica y social del país (Rodríguez Jiménez, 2024). Este ilícito se centra en el robo de rollizos, por medio de la sustracción desde predios legales, propiedades de forestales, o el robo de camiones cargados. El problema es considerable, ya que, cada camión con acoplado cargado con madera sustraída genera ganancias por más de \$1.500.000. Existe documentación sobre diversos casos asociados al robo de madera como lo fue hacia la forestal Arauco descubierto en 2022, que da cuenta de organizaciones criminales que operan desde hace años, se evidencia una trama de sobornos, vínculos con comunidades, tráfico de armas y enfrentamientos a tiros (González, 2024). Según estimaciones de la Corporación Chilena de la Madera (Corma), las pérdidas por este delito entre 2021 y 2023 ascendieron a \$167.000.000.000 (González, 2024).

Frente a la gravedad del robo de madera, el Congreso modificó el Código Penal, el 27 de septiembre de 2022, cuando fue publicada en el Diario Oficial la Ley N°21.488, que tipifica el delito de sustracción y otros relacionados, habilitando el uso de técnicas especiales de investigación para su persecución y agrava las penas por estos delitos, modificando también el Decreto con Fuerza de Ley N°5, del Ministerio de Agricultura, la Ley N°19.913 y también la Ley N°20.393.

Paralelamente, las forestales han incorporado diversas medidas tecnológicas para combatir el hurto de madera, una de ellas es la adopción de herramientas técnicas, como drones para perseguir el material robado, sistemas de georreferenciación y cámaras que operan desde tierra, fotografías de las placas de los camiones y la presencia de operadores en terreno para poder denunciar ante Carabineros y la Fiscalía (Emily Avendaño, 2022). También, se destaca el sistema desarrollado en conjunto por las principales forestales denominado como “Buena Madera, confiable desde su origen”, que tiene como objetivo dar a conocer al consumidor de este tipo de material, sobre el origen de la madera que compra, proteger a los trabajadores y a las pequeñas y medianas empresas de la industria maderera junto con desarrollar un sistema de alerta para rastrear la madera ilegal, empezando por la supervisión de cada camión que transporta la madera (Michael, 2021).

En este contexto, la empresa chilena Ecofor contribuye a mitigar el robo de madera al promover la trazabilidad de esta desarrollando el sistema Trazo, el cuál es utilizado por aproximadamente el 90% de las forestales chilenas para monitorear la trazabilidad de la madera, proveyendo

diversas soluciones tecnológicas, desde monitoreo vía GPS de camiones a sistemas avanzados de verificación de guías de despacho.

Presentación del problema

Un área de mejora del sistema Trazo es la identificación de camiones y su carga antes del almacenaje en canchas de acopio, concretamente, identificar la carga de rollizos de camiones. Esto se intentó llevar a cabo por medio de PostCom un subsistema de Trazo que consiste en dos cámaras, que, al detectar movimiento, toman imágenes de la patente y la carga del camión en la entrada del recinto destinado para almacenaje.

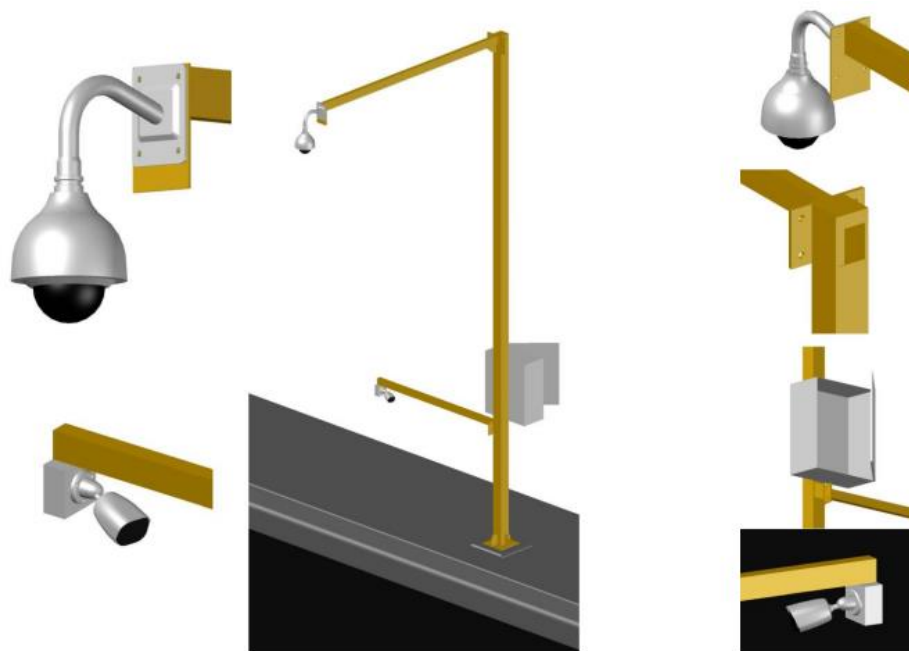


Ilustración 1: Sistema de cámaras PostCom

Originalmente, PostCom fue desarrollado para permitir la identificación del camión y la carga de estos, para luego habilitar la entrada o salida de un camión. Las imágenes son enviadas a un servidor, donde, la identificación de la patente es llevada a cabo mediante inteligencia artificial, esta identificación es exitosa, sin embargo, la identificación de carga de rollizos y otros tipos es deficiente, con una baja precisión, por lo que debe ser validada por un trabajador. Actualmente, debido a la poca confianza que tienen las predicciones (entre 45% y 60%), el sistema es usado como vía de registro de entrada y salida de camiones, donde la fotografía que debiese predecir la carga es usada como imagen de respaldo almacenada en una base de datos.

Al depender de validación humana sobre la carga de los camiones se tienen costos adicionales, factor de error humano, y, según lo expuesto sobre el robo de madera, se puede inferir el riesgo de colusión o coacción entre bandas de crimen organizado con personas a cargo de un punto vital del proceso de transporte de rollizos. Por tanto, mitigar este riesgo resulta relevante para las empresas forestales y colabora con la trazabilidad de la madera.

Comprometidos con la mejora continua de sus sistemas, Ecofor, requiere una herramienta que disponga de un modelo con alta efectividad de clasificación, superior al actual que posee una confianza entre 45% y 65% dependiendo del punto de recepción. Debido al robo de madera el enfoque está en la identificación de rollizos. El sistema PostCom, cada mes toma aproximadamente 1500 imágenes de camiones cargados con rollizos en distintos puntos de recepción, esto da cabida al etiquetado de imágenes históricas que permitan el entrenamiento de un modelo más robusto que obtenga mejores resultados en la identificación de la carga.

Por tanto, el problema que aborda esta memoria es la deficiencia del sistema PostCom para identificar correctamente la carga de rollizos. Se requiere obtener un modelo de inteligencia artificial que mejore sustancialmente la precisión de esta tarea, eliminando la necesidad de validación humana, para así fortalecer la trazabilidad de la madera, y de forma indirecta, contribuir a la mitigación del robo de madera.

2. Objetivos

Para resolver el problema planteado, se establece el siguiente objetivo general y tres objetivos específicos.

2.1 Objetivo general

Desarrollar un modelo que prediga de forma automática la carga de rollizos de camiones usando imágenes de cámaras estáticas y permitiendo su implementación en tiempo real.

2.2 Objetivos específicos

1. Crear un conjunto de datos etiquetados a partir de imágenes históricas para el entrenamiento de los modelos.
2. Entrenar y evaluar modelos de Deep Learning pre entrenados para clasificar si un camión está cargado e identificar si dicha carga corresponde a rollizos.
3. Medir el rendimiento de los modelos entrenados bajo criterios de alta precisión y bajo costo computacional para determinar su viabilidad en un entorno de producción.

3. Marco teórico

3.1 Estado del arte

Los modelos de Deep Learning basados en redes neuronales convolucionales (CNN) han mostrado ser útiles para la clasificación de imágenes alcanzando altos niveles de precisión (Tamuly et al., 2020). Además, se ha visto que son superiores para la clasificación de imágenes en comparación a modelos de Machine Learning que utilizan técnicas de regresión o clasificación, debido a que pueden captar diferencias sutiles presentes en las imágenes (J. Lu et al., 2021).

De forma similar (G. Liu, 2020) destacan las CNN para la extracción de características a partir de imágenes, concretamente métodos de Deep Learning para la detección de defectos superficiales en la producción industrial, en donde se tienen enormes diferencias en la forma, tamaño, textura, color, fondo e iluminación de la imagen del objeto de interés, lo que es extrapolable a la clasificación e identificación de rollizos en la carga de un camión debido a la variabilidad y condiciones climáticas de las zonas geográficas de los puntos de recepción.

Los modelos de Deep Learning desde cero requieren una gran cantidad de datos y un alto poder de cómputo. Por ello, los modelos pre entrenados, como aquellos de detección de objetos, como los de la familia YOLO (You Only Look Once), desarrollados por Ultralytics (*Ultralytics YOLO Docs*, 2023), resultan interesantes para solventar el problema. El modelo YOLOv8 (8.^a versión) se compone de dos partes principales: backbone (columna vertebral o red troncal) y head (cabeza), ambas utilizan una red neuronal completamente convolucional. Este modelo sobresale por su velocidad y precisión a la hora de predecir, además de poder implementarse en diversos tipos de hardware, con un bajo costo computacional, permitiendo su uso en tiempo real (Sohan et al., 2024).

Con lo revisado hasta el momento se observa que modelos de Deep Learning y modelos de detección de objetos son prometedores para atacar el problema. Así, entrando a aplicaciones específicas, Sikora y Fojtík (2020) abordan un problema de clasificación de troncos según su forma para su posterior apilado. Este estudio se centra en clasificar el tipo de carga de madera en camiones que entran a una planta de procesamiento. Usan imágenes tomadas por cámaras fijas en un portón de inspección; entrenan una red neuronal convolucional basada en la arquitectura VGG-19, realizando fine-tuning, con dos clases de carga de rollizos. Logran automatizar la inspección de seguridad distinguiendo por visión artificial el modo en que se debiesen apilar los troncos, obteniendo resultados para la métrica de clasificación exactitud de

85% por clase y al clasificar un camión según el resultado promedio de todas las imágenes, alcanzan una exactitud promedio de casi 98% (Sikora & Fojtík, 2020).

Al ser el eje la clasificación de camiones cargados con rollizos, trabajos en donde solo se identifique o trabaje con rollizos y/o troncos de madera son extrapolables al problema de esta memoria, pues al tener como entrada una imagen de la carga de un camión, la identificación de los rollizos en diversos contextos o entornos es equivalente a distinguir un camión cargado con rollizos.

Establecido lo anterior, Da Silva et al. (2021) aborda la detección de troncos de árboles vivos (sobre suelo) usando imágenes visibles y térmicas. Compilaron un conjunto de 2895 imágenes anotadas. Entrenan y comparan cinco modelos (SSD MobileNetV2, SSD InceptionV2, SSD ResNet50, SSD Lite MobileDet y YOLOv4 Tiny). El mejor desempeño lo obtuvo YOLOv4-Tiny, alcanzó un AP de aproximadamente 90% y F1-Score de 89%, además, obtiene los tiempos de inferencia más bajos, usando GPU, con aproximadamente 8 ms por imagen. Estos resultados dan cuenta de cómo modelos compactos, más livianos detectan con alta precisión troncos en entornos naturales usando visión RGB y térmica, sumado al hecho que los conjuntos de datos de validación con solo imágenes de visión RGB obtuvieron resultados superiores en todos los casos (da Silva et al., 2021).

Carratù et al. (2023) diseñaron un sistema de visión artificial que emplea una red CNN tipo YOLO (en su arquitectura de la versión 4) junto con una transformación lineal directa (DLT) para localizar los rollizos cargados en camiones y medir sus diámetros. El modelo fue calibrado en un entorno real donde se realizó un etiquetado manual de aproximadamente 1500 imágenes de camiones cargados con troncos con una posterior augmentación de datos logrando que en el 80% de los casos el error medio de medida fuera inferior a 4 cm (aproximadamente el 8% del diámetro medido), demostrando una alta precisión. Al tener como objetivo la medición de los rollizos la evaluación y análisis de la detección de estos no es tratada explícitamente, pero se observa que los modelos alcanzan valores para la métrica mAP[0.5] (mean Average Precision sobre umbrales IoU de 0,5) de aproximadamente el 80% siendo resultados muy buenos al considerar que se detecta solamente la cara del corte del tronco de la cual se mide el diámetro de este (Carratù et al., 2023).

YOLACT_WOOD es un sistema de segmentación de instancia diseñada específicamente para detectar y segmentar toda la carga de madera en un camión completo, buscando abordar el problema de baja velocidad de detección, baja precisión, apilamiento denso de madera y, por ende, facilidad de ocultación y omisión por parte de trabajadores. Usando como base

YOLACT, un trabajo previo de los autores, con backbone ResNeXt más un bloque de atención CBAM, aumentan el tamaño de entrada y emplean pérdidas CIOU/DIOU y Fast-NMS para mejorar la localización y reducir falsos negativos. Reportan mAP[0.5] de 79,1% mejorando en 5,6% sobre trabajo anterior (YOLACT), junto a una precisión de 96,61% (Zheng et al., 2023).

Una tarea sencilla pero tediosa que puede llevar gran cantidad de tiempo es el conteo de rollizos en el rubro forestal. Para esto, Casas et al. (2023) aplican YOLOv8 en su versión nano para detectar y contar troncos apilados de eucalipto en imágenes. Usaron 230 imágenes de pilas de rollizos con diámetros desde 4 a 31 centímetros y entrenaron el modelo para reconocer cada tronco. Obtuvieron una precisión de 74,1% y mAP[0.5] de 79,9%, además prueban la capacidad de generalización de detección y conteo del modelo de forma manual, si bien la detección es buena el error relativo del conteo es alto, sin embargo, se vislumbra la gran capacidad de detección de rollizos del modelo YOLOV8n (nano) (Casas et al., 2023).

De la mano con los problemas anteriores, el conteo y cálculo de diámetro de troncos de madera por medio de la segmentación, Lu et al proponen un modelo dual que combina YOLOv3 para detectar extremos de rollizos y DeepLabv3+ para segmentarlos, esto con el fin de contar y medir el diámetro de los troncos. En su validación reportan mAP[0.5] de 97,28% y mIoU de segmentación de 92,22% con exactitud promedio de 96,26%. En pruebas forestales, el error promedio en el volumen de un camión completo fue 1,2% y en el diámetro individual 0,73%, cumpliendo estándares de la industria, permitiendo su implementación (Z. Lu et al., 2024).

A continuación, se resumen los artículos analizados que abordan problemas relacionados a la detección de rollizos:

Referencia	Objetivo	Modelos	Resultados
Sikora & Fojtík (2020)	Clasificación de troncos según su forma en punto de recepción por medio de imágenes.	Red neuronal VGG-19	Logran la clasificación exhibiendo una exactitud de 98%
(da Silva et al., 2021)	Detección de troncos en árboles vivos por medio de imágenes RGB y térmicas.	SSD MobileNetV2, SSD InceptionV2, SSD ResNet50, SSDLite MobileDet y YOLOv4 Tiny	Detectan troncos, independiente del tipo de imagen donde el mejor modelo (YOLOv4 Tiny) exhibe métricas de: AP = 90%

			F1-Score = 89%
(Carratù et al., 2023)	Medir el diámetro de rollizos en camiones de carga lo que implica la detección de los rollizos.	Red neuronal CNN basada en Yolo con DLT.	Logran una medición con un error medio inferior al 8% con un mAP aproximado del 80%
(Zheng et al., 2023a)	Detectar y segmentar toda la carga de madera de un camión.	Mask R-CNN Cascade Mask R-CNN YOLACT YOLACT_WOOD	Logran la detección de la carga con una Precision de 96,6% y mAP[0.5] de 79,1%
(Casas et al., 2023a)	Identificar y contar rollizos apilados usando imágenes de cara de corte de los rollizos.	YOLOv8 nano	Obtienen un error relativo alto en el conteo de rollizos, sin embargo, destacan la detección de rollizos con un mAP[0.5] de 79,9%.
(Z. Lu et al., 2024)	Detectar y segmentar rollizos en camiones cargados.	YOLOv3 para detección y Deeplabv3+ para segmentación.	Muestran excelentes resultados para ambos modelos donde YOLOv3 obtuvo los siguientes resultados: mAP[0.5] = 97,28% Accuracy = 96,26%

Tabla 1: Resumen Estado del Arte

Donde se referencia al trabajo analizado, el objetivo que resume el propósito del artículo relacionado con el problema a resolver, los modelos utilizados para cumplir con el objetivo y un resumen de los resultados asociados a la detección de rollizos.

3.2 Evaluación de alternativas

Los trabajos anteriores dieron cuenta de claras tendencias hacia el uso de modelos pre entrenados de detección de objetos y clasificación de imágenes, los cuales resultaron relevantes para abordar el segundo objetivo específico (OE2). Es por esto, se abordó cada objetivo como una tarea a evaluar.

En base a los objetivos específicos (OE) se tienen tres tareas: el etiquetado de las imágenes (OE1), el entrenamiento de modelos que permitieran la identificación de rollizos (OE2) y la validación de los modelos entrenados en base a métricas de precisión y costo computacional (OE3). Considerando lo anterior, se detallan los objetivos de la sección, los cuales son abordados de forma secuencial:

- Definir la cantidad de imágenes necesarias y la herramienta que permita el etiquetado de los datos.
- Fundamentar la elección de modelos de clasificación y detección de objetos a entrenar.
- Elegir métricas de evaluación para medir la precisión de clasificación y costo computacional de los modelos.

3.2.1 Evaluación de alternativas para OE1

Para el etiquetado de las imágenes, se consideraron dos aspectos: la calidad y el volumen de imágenes a etiquetar. Estos fueron abordados simultáneamente, gracias a la revisión bibliográfica del estado del arte, donde exhiben la cantidad de datos utilizados para el entrenamiento de sus modelos junto con los buenos resultados obtenidos, ya que la base de un buen modelo es el conjunto de datos con el que se entrena (Kaur et al., 2021).

Para determinar la cantidad de datos que fue necesario utilizar, se presenta la siguiente tabla:

Referencia	Tarea	Cantidad imágenes	Herramienta utilizada
(Sikora & Fojtík, 2020)	Clasificación	1136	-
(da Silva et al., 2021)	Detección de objetos	2895	Computer Vision Annotation Tool (CVAT)
(Y. Lin et al., 2022)	Detección de objetos	~ 200	LabelImg
(Carratù et al., 2023)	Detección de objetos	~ 600	-

(Zheng et al., 2023)	Detección y segmentación de objetos	~150	Labelme
(Casas et al., 2023)	Detección y segmentación de objetos	230	LabelImg
(Z. Lu et al., 2024)	Detección y segmentación de objetos	~100	LabelImg

Tabla 2: Tamaño de conjunto de datos en trabajos relacionados.

Donde se referencia al trabajo analizado, el tipo de tarea que desarrollan los modelos entrenados, la cantidad de imágenes en el conjunto de datos, y, si es detallado, la herramienta utilizada para etiquetar los datos. Hay que mencionar que las cifras anteriores contemplan la suma de la cantidad de imágenes usadas para entrenamiento, validación y prueba (este último no siempre está presente).

Es con las etiquetas presentes por imagen con las que, principalmente, los modelos de detección de objetos entrenan y aprenden a detectar patrones visuales, es decir, identifican regiones (Distante & Distante, 2020). Por tanto, aquellos trabajos de detección o segmentación de objetos, en realidad, tienen múltiples etiquetas correspondientes a bounding boxes (cuadros delimitadores).

Ahora, en el contexto del problema de la presente memoria, se tiene que, por imagen en la mayoría de los casos hay solo un objeto de interés, siendo un camión cargado o vacío. Con el análisis realizado, a la vez de tener en cuenta que el problema se puede reducir a clasificar imágenes, donde además se tiene un entorno de entrenamiento con recursos limitados, se consideró que para efectos de obtener un modelo robusto con buena capacidad de generalización y que permita visualizar resultados en tiempos pertinentes, se utilice un conjunto de al menos 1500 imágenes. Esto prácticamente corresponderá a la cantidad total de etiquetas.

La cifra anterior representa un gran volumen de datos, donde además se busca heterogeneidad, ya que se ha visto que las redes neuronales entrenadas en dichas condiciones aprenden patrones más generales (Sikora & Fojtík, 2020), lo que es útil en el contexto del problema donde principalmente hay camiones cargados y vacíos, siendo el ambiente del punto de recepción lo que da mayor variabilidad a las imágenes.

Una vez definida la cantidad de datos a utilizar, se procedió a definir la herramienta de etiquetado para el conjunto de imágenes. Observando que en la literatura revisada se repite el uso del software LabelImg, el cual permite etiquetar los datos por medio de cuadros delimitadores. Se revisa el repositorio oficial (Tzuralin, 2015) de este software y se observa que, actualmente, es parte de la plataforma de código abierto Label Studio. Por tanto, se decidió usar esta herramienta para el etiquetado del conjunto de imágenes, dado que mantiene soporte activo y ofrece mayores funcionalidades.

3.2.2 Evaluación de alternativas para OE2

Establecida la cantidad de imágenes y la herramienta de etiquetado, se definió la parte vital del presente proyecto: la elección de los modelos a entrenar.

En base al estado del arte, se observó una tendencia y preferencia clara por los modelos YOLO. Según Vijayakumar y Vairavasundaram (2024), los modelos YOLO desde su octava versión destacan en la detección de objetos y funcionan principalmente en entornos de tiempo real con una precisión y velocidad sobresalientes (Vijayakumar & Vairavasundaram, 2024) lo que correspondía con los requerimientos de Ecofor, al tener la necesidad de predecir en tiempo real, de manera local y aumentar la efectividad actual. Sumado al hecho de mostrar buenos resultados en entornos abiertos e iluminados, los modelos YOLO han mostrado ser capaces de detectar y segmentar objetos, como maquinaria pesada de construcción, en entornos nocturnos (Zhang & Shen, 2024), lo que es relevante al ser un escenario presente en los puntos de entrada y salida de camiones de carga en el rubro forestal. Por consiguiente, se seleccionó la octava versión de los modelos YOLO debido a que se encontró que fue la última versión utilizada en un trabajo relacionado a la detección de rollizos.

Los modelos YOLO están a la vanguardia en la detección y segmentación de objetos en diversos contextos (Xu et al., 2024), por lo que, como segundo modelo se quiere uno que supere al detector de objetos de Ultralytics. Se identificó el modelo D-FINE, un modelo reciente basado en Transformers (DETR) que propone mejorar la regresión de los bounding boxes en DETR y derivados, en vez de predecir coordenadas directamente (como $[x1, y1, x2, y2]$), D-FINE introduce una refinación de distribución fina para cada vértice del bounding box, siendo la idea principal que el modelo sea más preciso en localizar objetos, especialmente complejos (Peng et al., 2024).

Para validar el modelo D-FINE, utilizan los conjuntos de datos COCO (T.-Y. Lin et al., 2015) y Objects365 (Tian et al., 2019) conjunto de pruebas ampliamente utilizados para analizar y comparar modelos de detección de objetos. D-FINE mostró mejoras sobre YOLOv8 con

ganancias de entre 2 y 3 puntos porcentuales para mAP y mayor precisión bajo un IoU estricto de 75 superando por 2 puntos porcentuales a YOLOv8 y su última versión (11.^a). Además, a pesar de ser un modelo más costoso computacionalmente, al ser basado en Transformers, mantiene tiempos de inferencia promedios de aproximadamente 25 ms por imagen siendo similar a YOLOv8 con 20 ms. Estos resultados dieron cabida a la posibilidad de obtener un modelo con mejores resultados que el modelo YOLO, por lo que se definió el modelo D-FINE como segundo candidato a entrenar.

Teniendo, en este punto, dos modelos de detección de objetos y considerando que se busca una implementación en tiempo real, resultaba interesante contar con un modelo de menor coste computacional que realice inferencias rápidas, incluso, con uso de CPU. Para esto, se consideró el uso de un modelo con arquitectura VGG, que ya ha sido utilizado en trabajos de clasificación de rollizos obteniendo buenos resultados (Sikora & Fojtík, 2020).

El modelo VGG16 ha sido utilizado en tareas comparativas para clasificación de camiones de obra vacíos y cargados por medio de imágenes reales donde alcanzó los mayores valores de precisión frente a otras 11 arquitecturas CNN, donde está incluido el modelo VGG19 (Q. Liu et al., 2019). El modelo pre entrenado VGG16 ha mostrado ser superior a su versión con mayor número de capas VGG19 en otros casos, un ejemplo de esto es en el contexto de predecir enfermedades foliares en plantas de tomates por medio de imágenes de hojas de la planta en cuestión. En dicho estudio el modelo VGG16 obtuvo una exactitud de 90,83% frente a un 88,55% del modelo VGG19, donde también se evaluó el modelo ResNet50 que obtuvo una exactitud de 51,57% (Mishra et al., 2023).

Por lo tanto, como tercer modelo se utiliza el modelo pre entrenado de clasificación VGG16.

Definidos los modelos, se presenta la siguiente tabla resumen:

Modelo	Tipo	Justificación
YOLOv8	Detección de objetos	Modelo de mayor uso en estado del arte. Usados en entornos de tiempo real. Robusto para ambientes variables.
D-FINE	Detección de objetos	Modelo innovador que ha mostrado ser superior frente a YOLO en conjuntos COCO y Object365. Captura con mayor fidelidad el objeto de interés con alto AP75.
VGG16	Clasificación	Menor coste computacional.

		Superioridad en contextos específicos a otros modelos de clasificación.
--	--	---

Tabla 3: Resumen de modelos elegidos para entrenar.

3.2.3 Evaluación de alternativas para OE3

Es necesario definir las métricas con las cuales se evaluaron y validaron los modelos. Como se establece en el OE3, se quiere medir los modelos en términos de costo computacional y alta precisión. Para esto se tienen métricas que son ampliamente usadas en el área de visión artificial, como son, por ejemplo, Accuracy (exactitud) y F1-Score para clasificación de imágenes o mAP para detección de objetos.

Primeramente, se abordó el criterio de medir los modelos en términos de alta precisión de clasificación. Para esto, la base es una matriz de confusión de predicciones versus valor real de las etiquetas utilizadas, la cual permite observar verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos, los cuales se explican a continuación:

- Verdadero Positivo (TP): predicción “positiva” y realmente lo es.
- Falso Positivo (FP): predicción “positiva” pero es incorrecto.
- Verdadero Negativo (TN): predice “negativo” y realmente lo es.
- Falso Negativo (FN): predice “negativo” cuando debería ser “positivo”.

Los valores anteriores se utilizan para calcular métricas que evalúan la precisión de clasificación de un modelo, dichas métricas son: Precision, Recall, F1-Score y Accuracy, las cuales se definen como:

- Precision: Proporción de predicciones positivas que son correctas para una clase. Se usa como referencia si se quiere minimizar falsos positivos. Se calcula como:
 - $\frac{TP}{(TP+FP)}$
- Recall: Proporción de positivos reales detectados para una clase. Se usa como referencia si se quiere minimizar falsos negativos. Se calcula como:
 - $\frac{TP}{(TP+FN)}$
- F1-Score: Media armónica entre Precision y Recall. Usada para evaluar balance entre ambos. Se calcula como:
 - $2 \cdot \frac{(Precision \cdot Recall)}{Precision + Recall}$
- Accuracy: Proporción de predicciones correctas sobre el total. Solo es relevante en conjuntos balanceados. Se calcula como:

$$\circ \frac{TP + TN}{TP + FP + TN + FN}$$

Estas métricas están presentes en la revisión bibliográfica realizada. Se ejemplifica su cálculo por medio de la matriz de confusión a continuación:

		Etiqueta real		
		+	-	
Etiqueta Predicha	+	Verdadero positivo (TP)	Falso positivo (FP)	Precision = $\frac{TP}{(TP + FP)}$
	-	Falso negativo (FN)	Verdadero negativo (TN)	
		Recall = $\frac{TP}{(TP + FN)}$		Accuracy = $\frac{TP + TN}{TP + FP + TN + FN}$

Tabla 4: Matriz de confusión.

Para los modelos de detección de objetos, además de la precisión de clasificación fue necesario medir la precisión espacial, esto es, la precisión con la que el modelo asigna un cuadro delimitador sobre un objeto. Para esto, se utiliza la métrica Intersection over Union (IoU):

- IoU: Área de intersección entre la caja predicha y la caja real, dividido por su unión, esto es:

- $$\frac{\text{Área de intersección}}{\text{Área de unión}}$$

Un IoU de 0,5 suele ser suficiente para considerar una predicción como correcta (Szeliski, 2022). La métrica IoU es usada como base, dado un umbral mínimo para considerar una predicción como verdadero positivo, para el cálculo de precisión promedio (Average precisión (AP)) correspondiente al área bajo la curva Precision-Recall que exhibe la métrica Precision en función de la métrica Recall por clase o etiqueta. Se tiene la métrica del promedio de AP, esta es, mAP (mean Average Precision) que proporciona una puntuación única y completa que resume la capacidad de un modelo para identificar y localizar correctamente objetos en varias clases y niveles de confianza. Equilibra eficazmente el compromiso entre encontrar todos los objetos relevantes (Recall) y garantizar que los objetos encontrados son realmente correctos (Precision). Existen las métricas mAP con porcentajes IoU comunes que suelen ser 0,5 y 0,75, donde solo se consideran Precision y Recall para predicciones donde el cuadro predicho supera

los umbrales anteriores. También, se tiene la métrica mAP[0.5:0.95] (comúnmente llamada solo mAP) la cual calcula el área bajo la curva para los umbrales IoU entre 0,50 y 0,95 por pasos de 0,05 y luego se calcula el promedio para cada uno de los valores obtenidos, esto es:

$$\bullet \quad \text{mAP}_{[0.5:0.95]} = \frac{1}{10} \sum_{i=1}^{10} \text{AP}_{\text{IoU}=t_i}, \quad \text{donde } t_i \in \{0,50, 0,55, \dots, 0,95\}$$

Para evaluar el costo computacional de los modelos se tienen métricas usadas comúnmente, como lo es Floating-point Operation (FLOP), como su nombre indica, mide las operaciones de punto flotante. En el contexto de los modelos de aprendizaje profundo y detección de objetos, cuantifica el número total de cálculos en coma flotante necesarios para una sola pasada del modelo. Dado que los modelos modernos implican miles de millones de operaciones de operaciones de este tipo se expresan en GigaFLOPs (GFLOPs).

Es importante no confundir los FLOPs con los FLOPS (operaciones de coma flotante por segundo), ya que esta métrica es una medida de la velocidad de procesamiento del hardware, como la capacidad de una GPU. Los FLOPs miden las operaciones totales, la carga de trabajo computacional. Por medio del recuento total de operaciones luego de un proceso de inferencia.

Además de medir los GFLOPs, considerando que se busca un modelo que entregue predicciones en tiempo real, se midió la latencia de los modelos. También, debido a los altos tiempos de entrenamiento, se consideró el tiempo necesario por cada época.

Finalmente, se establece el criterio laboral como eje central para evaluar la idoneidad de la solución propuesta.

El sistema actual no solo presenta deficiencias técnicas, sino que impone una condición laboral específica y riesgosa. La dependencia de una validación humana para una tarea crítica en la cadena de suministro expone al personal a dos factores clave:

1. **Ineficiencia y repetitividad:** El trabajador debe validar constantemente las predicciones de un sistema con una baja tasa de confianza (entre 45% y 65%), lo que convierte su labor en una tarea repetitiva y de bajo valor añadido, centrada en corregir errores tecnológicos en lugar de supervisar el proceso de forma estratégica.
2. **Riesgo de seguridad y ético:** Como se ha expuesto en la introducción, el robo de madera es un ilícito vinculado al crimen organizado. La necesidad de que un operario valide manualmente la carga de cada camión lo posiciona como un eslabón vulnerable a la colusión, ya sea mediante soborno o coacción. Esto representa un riesgo tangible para la seguridad e integridad del empleado y una amenaza para la seguridad operacional de la empresa.

Por consiguiente, el diseño de la solución se ha guiado primordialmente por el objetivo de mitigar y eliminar este riesgo laboral. No se busca únicamente una mejora en la precisión, sino desarrollar una herramienta lo suficientemente robusta y autónoma como para reemplazar la intervención humana en un punto de control.

4. Metodología

Para lograr el objetivo general fueron propuestas las siguientes tareas asociadas directamente a los objetivos específicos:

Crear un conjunto de datos etiquetados a partir de imágenes históricas para el entrenamiento de los modelos.

- **Tarea 1:** Gestionar la entrega de parte de Ecofor de las imágenes necesarias para entrenar el modelo, definiendo que tipo de imágenes son útiles para este fin.
- **Tarea 2:** Etiquetar las imágenes por medio de bounding boxes utilizando el software Label Studio, con el objetivo de construir un conjunto de datos adecuado para el entrenamiento de los modelos.

Entrenar y evaluar modelos de Deep Learning pre entrenados para clasificar si un camión está cargado e identificar si dicha carga corresponde a rollizos.

- **Tarea 3:** Desarrollar un pipeline de preprocesamiento de imágenes que contemple técnicas de normalización y aumentación de datos, respetando y manteniendo los bounding boxes.
- **Tarea 4:** Entrenar y descartar modelos de clasificación por medio de métricas cuantitativas.
- **Tarea 5:** Entrenar y descartar modelos de detección de objetos por medio de métricas cuantitativas.

Medir el rendimiento de los modelos entrenados bajo criterios de alta precisión y bajo costo computacional para determinar su viabilidad en un entorno de producción.

- **Tarea 6:** Evaluar y medir modelos en términos de costo computacional, velando por un balance con una alta precisión.
- **Tarea 7:** Validar modelos entrenados con imágenes recientes capturadas por el sistema PostCom.

Estas tareas se agrupan en tres grandes áreas de trabajo: recolección de datos, modelamiento, y evaluación y validación de modelos. De este modo, la metodología general se estructura de la siguiente manera:

1. Recolección de datos

- a. Establecer criterios de calidad para la entrega de imágenes por parte de Ecofor.
- b. Limpieza de datos.

- c. Etiquetado de imágenes de forma manual.

2. Preprocesamiento de datos y modelamiento

- a. Preprocesamiento de datos, por medio del desarrollo de un pipeline de augmentation.
- b. Entrenar los modelos propuestos con el conjunto de datos preparado.

3. Evaluación y validación

- a. Evaluar modelos en etapas tempranas por medio de métricas de precisión de clasificación.
- b. Validación y evaluación con últimas imágenes de sistema PostCom por medio de métricas de precisión de clasificación y costo computacional.

Cabe destacar que las actividades 1.c, 2.a, 2.b y 3.a representan un ciclo iterativo de trabajo orientado a mejorar progresivamente la precisión de clasificación de los modelos propuestos. Este ciclo previo es necesario ya que se busca un modelo para ser implementado en múltiples entornos reales y que prediga con alta efectividad.

5. Desarrollo del proyecto

El desarrollo del proyecto siguió un enfoque práctico e iterativo, partiendo por la recolección de datos, avanzando a través de ciclos de entrenamiento, evaluación y ajuste.

Inicialmente, debido a que se definió usar modelos de detección de objetos, para solicitar imágenes útiles a la empresa Ecofor fue necesario una revisión exhaustiva de literatura en donde se explique cómo etiquetar correctamente los objetos en las imágenes y como se define una imagen útil para ser etiquetada.

Un punto importante y que en general se debe respetar en todos los conjuntos de datos, es la variabilidad de las clases de interés, es decir, contar con la mayor cantidad de escenarios reales en que se puede tener un objeto (Szeliski, 2022). En el contexto del problema, esto se traduce a la variabilidad de tipos de camiones que ingresan a una cancha de acopio, la forma de estos, tipo de cabina, carros de carga, si presenta grúa con garra, y, más importante aún, variabilidad de las posiciones en donde se pueda observar la carga de rollizos, esto es, zona superior, lateral izquierdo y derecho.

Ahora, para el etiquetado por medio de cuadros delimitadores, se encontró el enfoque “Label what can you see” (Carratù et al., 2023) que indica etiquetar objetos claramente visibles, no superpuestos por otro. Un punto importante a la hora de realizar el cuadro delimitador es que este debe ser lo más ajustado posible al contorno del objeto, evitando la mayor cantidad de píxeles que no aportan en la identificación del objeto y que solo introducen ruido al aprendizaje del modelo (Z. Lu et al., 2024).

Teniendo más claridad respecto a las imágenes que serían útiles, se pidió a Ecofor imágenes de puntos de recepción donde idealmente cumplan con:

- Variabilidad de camiones:
 - Distintas formas
 - Vistas desde diversos ángulos
- Solo un camión por imagen
- Evitar presencia de otros objetos en las imágenes
 - Haciendo énfasis en evitar elementos que cubran parcialmente al camión

Luego de esto, Ecofor entregó imágenes históricas de dos puntos de recepción. Se realizó un análisis exploratorio de los datos donde se encontró un conjunto de imágenes de aproximadamente 15000 elementos, donde el 51% era realmente útil.

Para poder comenzar el etiquetado, se definieron las etiquetas o clases del conjunto de imágenes. Al revisar estas, se observó que se repetían 3 elementos además de los camiones cargados con rollizos, por lo que se establecieron las siguientes clases:

- **Cargado Rollizos:** Camiones cargados con rollizos.
- **Cargado otro:** Camiones cargados con cualquier otro tipo de carga, como astillas, madera aserrada, entre otros. Por otro lado, se consideran camiones con acoplado para carga de astillas sin carga, debido a que se requiere una vista superior de la carga para distinguir si se encuentra vacío o no.
- **No cargado:** Camiones sin carga, excluyendo a camiones de transporte de astillas sin carga.
- **Otro:** Cualquier elemento presente en las imágenes que se repiten con frecuencia, como camionetas, camiones de menor tamaño, buses de transporte y personas. También, aquellas instancias donde solo se aprecie la cabina del camión y no su área de carga.

Se presenta la siguiente figura con ejemplos de las clases definidas:



Ilustración 2: Etiquetas definidas con imagen de ejemplo.

El desarrollo de la presente memoria se llevó a cabo bajo el sistema operativo Windows, considerando esto y que la instalación directa de Label Studio en este entorno puede presentar

ciertas incompatibilidades relacionadas con dependencias, se optó por su ejecución mediante Docker. Esto permitió una instalación limpia y controlada, evitando conflictos.

La configuración de Docker concedió levantar un contenedor con Label Studio de forma local, accediendo a la interfaz gráfica a través de un navegador web. Desde allí, y habiendo realizado una limpieza manual, se cargaron las imágenes, se definieron las etiquetas descritas (“Cargado Rollizos”, “Cargado otro”, “No cargado” y “otro”) y se configuró el esquema de anotación mediante bounding boxes. Esto permitió una visualización clara y consistente durante todo el proceso de etiquetado, asegurando que las imágenes fueran anotadas según los criterios estudiados.

Luego de un exhaustivo etiquetado, se obtuvo un conjunto de datos de casi 1500 etiquetas. Como uno de los objetivos planteados en la evaluación de alternativas era obtener un conjunto de datos de al menos 1500 elementos, se decidió entrenar un primer modelo con fin de observar la calidad de predicción en este punto y vislumbrar que técnicas adoptar para mejorar la precisión de clasificación.

El modelo elegido para esta tarea fue el modelo YOLOv8 nano, debido a que los modelos YOLO son los más usados en la literatura revisada y su versión más pequeña es robusta y ofrece tiempos de entrenamiento más cortos (*Ultralytics YOLO Docs*, 2023).

Dividiendo el conjunto en una proporción 80% entrenamiento y 20% validación, se obtuvieron los siguientes valores para métricas Precision, Recall y F1-Score:

- **Precision:** 0,685
- **Recall:** 0,651
- **F1-Score:** 0,667

Estos fueron buenos resultados para un modelo inicial y más aún para la clase de interés “Cargado Rollizos” que predijo con gran precisión, teniendo un 95% de verdaderos positivos. Esto refuerza el uso de modelos YOLO, en especial su octava versión más pequeña. Sin embargo, estos resultados no son del todo representativos, ya que el modelo debe ser validado y mostrar buenos resultados en puntos de recepción distintos a los usados para su entrenamiento, por lo que es necesario probar con un nuevo punto de entrenamiento, distinto a los usados para entrenar el modelo.

Por otro lado, en Chile, las macrozonas centro sur y sur presentan una variedad de climas adversos caracterizados por ser húmedos, con altas precipitaciones (GARCÉS et al., 2023). Al

realizar inferencias de forma unitaria del modelo en estos contextos se observa como el modelo falla en la predicción:



Ilustración 3: Ejemplo de predicciones erróneas en clima de baja visibilidad.

Evidenciado lo anterior, es necesario considerar este factor de riesgo para que no afecte la confianza de las predicciones del modelo. Para resolver esto, se decide aumentar el conjunto de datos por medio de dos tareas:

1. Etiquetar más imágenes, buscando que sea en días donde se presentaron climas adversos y priorizando las clases “Cargado otro” y “No cargado”, ya que son las clases minoritarias.
2. Enfocar el desarrollo del pipeline de augmentación en replicar climas adversos como lluvia y neblina.

Además, se solicitaron a Ecofor más imágenes de un punto de recepción distinto a los iniciales, bajo los mismos criterios establecidos.

Para la primera tarea se filtró el conjunto de imágenes por fechas en donde se presentaron climas con precipitaciones, esto se revisó en el sitio de la Dirección Meteorológica de Chile.

De este modo, se seleccionaron más imágenes y se subieron a Label Studio para su etiquetado, obteniendo un conjunto de casi 2000 etiquetas.

Se ha visto como el proceso de augmentación de datos favorece a la capacidad de generalización de los modelos, sin embargo, es necesario validar el proceso, pues también puede desfavorecer el entrenamiento (Kaur et al., 2021), como el hecho de aplicar transformaciones geométricas en exceso o poco realistas (Lei et al., 2019).

De la mano de factores asociados al clima, otro factor relevante es la iluminación en la imagen, más aún en ambientes al aire libre. Condiciones de baja iluminación afectan al rendimiento de inferencia de los modelos, afectando la precisión (Mekled et al., 2024; Zhang & Shen, 2024) del mismo modo, que lo hacen entornos con alta luminosidad (Y. Lin et al., 2022).

Con fin de mejorar las predicciones en climas adversos y en base a lo estudiado, se desarrolló un pipeline de augmentación utilizando la librería Albumentations. Convenientemente, permite realizar transformaciones sobre imágenes, ajustando los cuadros delimitadores en formato YOLO.

El pipeline contempla transformaciones sobre las imágenes y cuadros delimitadores que son aplicadas según un grado de probabilidad para tratar el posible problema de pérdida de precisión por exceso de transformaciones. Se resume las transformaciones utilizadas en la siguiente tabla:

Tipo de transformación	Función específica	Descripción
Geométricas	<i>HorizontalFlip</i>	Voltea la imagen horizontalmente (espejo).
	<i>RandomRotate90</i>	Rota la imagen aleatoriamente en múltiplos de 90 grados.
	<i>ShiftScaleRotate</i>	Combinación de desplazamiento, escalada y rotación en una sola transformación.
Color y brillo	<i>RandomBrightnessContrast</i>	Modifica aleatoriamente el brillo y el contraste de la imagen.
	<i>RandomGamma</i>	Ajusta la corrección gamma de la imagen, afectando principalmente el brillo de los tonos medios.

	<i>HueSaturationValue</i>	Modifica el tono (color), saturación (intensidad del color) y valor (brillo) de la imagen.
Efectos ambientales	<i>RandomRain</i>	Simula efecto de la lluvia añadiendo gotas y ajustando el brillo.
	<i>RandomFog</i>	Simula niebla o neblina en la imagen.
	<i>RandomShadow</i>	Añade sombras aleatorias en la imagen.
Problemas de cámara	<i>MotionBlur</i>	Simula el desenfoque de movimiento cuando la cámara o el objeto se mueve durante la exposición.
	<i>GaussianBlur</i>	Aplica un desenfoque general a la imagen, simulando problemas de enfoque.
	<i>ImageCompression</i>	Simula la pérdida de calidad por compresión JPEG.
	<i>Downscale</i>	Reduce la resolución de la imagen y luego la escala nuevamente, simulando mala calidad.

Tabla 5: Resumen de transformaciones utilizadas para augmentación de imágenes.

Las transformaciones fueron aplicadas de forma compuesta, esto es, seleccionando una transformación de cada categoría según un grado de probabilidad controlado como parámetro, luego, la transformación en específico tenía una probabilidad única de ser aplicada sobre una imagen. Lo que permitió que no se apliquen todas las transformaciones en cada imagen aumentada, velando por mantener una imagen más realista.

Teniendo un conjunto de datos más balanceado con una mayor cantidad de imágenes en climas adversos y desarrollado un pipeline de augmentación, se procedió a entrenar nuevamente el modelo YOLOv8n para validar estos procedimientos. Para esto se dividió el conjunto de datos en una proporción de 80% entrenamiento y 20% validación.

También, con el fin de validar el pipeline de augmentación, este se aplicó sobre cada imagen en tres instancias, obteniendo un conjunto de entrenamiento cuatro veces mayor al original.

Se obtuvieron los siguientes resultados para las métricas F1-Score y mAP[0.5] en el conjunto de validación:

	Modelo sin augmentación	Modelo con augmentación
F1-Score	0,883	0,901
mAP[0.5]	0,923	0,924

Tabla 6: Resultados de F1-Score y mAP[0.5] para validación de pipeline de augmentación.

Estos resultados son muy buenos para ambos procedimientos, siendo superior el modelo con el conjunto de entrenamiento augmentado, por lo que las técnicas aplicadas pudieron ser validadas.

Se decidió comenzar a probar los demás modelos propuestos para distinguir tempranamente si son herramientas útiles para la clasificación de camiones cargados con rollizos. Para esto, se procedió con el entrenamiento de los modelos YOLOv8n, YOLOv8m y D-FINE en su versión media.

Los datos utilizados hasta el momento son exportados desde Label Studio en formato YOLO el cual contempla dos directorios, uno que contiene las imágenes (images) y otro los archivos de texto donde cada uno corresponde a una imagen (labels). Estos archivos de texto presentan una etiqueta por cada línea, las cuales mantienen el formato YOLO (class id, x, y, height, width):

- class_id: Número entero que referencia a una clase a la cual pertenece la etiqueta.
- x: valor de la coordenada en eje x (respecto a la imagen) para el centro del cuadro delimitador.
- y: valor de la coordenada en eje y (respecto a la imagen) para el centro del cuadro delimitador.
- height: Número decimal que exhibe el alto normalizado del cuadro delimitador (respecto a la imagen).
- width: Número decimal que exhibe el ancho normalizado del cuadro delimitador (respecto a la imagen).

También, es posible dividir el directorio principal en los conjuntos de entrenamiento, validación y test, donde estos contienen los subdirectorios de imágenes y etiquetas respectivos.

Para poder entrenar el modelo D-FINE con el mismo conjunto de datos que los modelos YOLO fue necesario transformar al tipo de datos que D-FINE espera, el cual corresponde a una estructura que contiene las etiquetas en formato COCO e incluye, entre otros, los siguientes parámetros:

- image_id: ID de la imagen de ejemplo

- image: un objeto *PIL.Image.Image* que contiene la imagen
- width: ancho de la imagen
- height: alto de la imagen
- objects: un diccionario que contiene metadatos de las cajas delimitadoras (bounding boxes) de los objetos en la imagen:
 - id: ID de la anotación
 - area: área de la caja delimitadora
 - bbox: caja delimitadora del objeto (en formato COCO)
 - category: la categoría o clase del objeto con valores posibles.

Para el entrenamiento y evaluación de los modelos se dividió el conjunto de datos en una proporción de 70% entrenamiento, 15% validación y 15% test utilizando una estrategia “Stratified K-Fold” para mantener la distribución de las etiquetas.

Esta vez, se aplicó augmentación solo sobre la clase “No cargado” y se realizó un downsampling sobre la clase “otro” buscando un balance.

De este modo, se obtuvieron los siguientes resultados para las métricas mAP:

Modelo	Data augmentation	mAP[0.5:0.95]	mAP[0.5]	mAP[0.75]
YOLOv8n	No	0,843	0,939	0,908
	Si	0,830	0,924	0,887
YOLOv8m	No	0,849	0,946	0,917
	Si	0,828	0,926	0,893
D-FINE	No	0,758	0,860	0,814

Tabla 7: Resultados para métricas mAP en distinto umbral IoU.

Se observan buenos resultados en general, siendo el modelo YOLOv8 en su formato medio el que presenta mejores resultados para todas las métricas aplicadas. También, se vio que el proceso de augmentación en este experimento no mejoró la precisión de los modelos, por lo que, con el fin de ahorrar recursos, se decidió no entrenar el modelo D-FINE con el conjunto de datos balanceado. Se infiere que no se obtuvieron mejores resultados con este conjunto debido a que solo se aumentó una clase y además se eliminó algunas imágenes de la clase “otro”. Esto permitió identificar que es necesario mejorar la estrategia de augmentación.

En este punto, Ecofor entrega imágenes de un nuevo punto de recepción, donde predomina el flujo de camiones cargado con rollizos. Por lo que, se seleccionaron aproximadamente 150 imágenes para ser etiquetadas.

Se realizó una validación de los modelos con este nuevo conjunto de datos, obteniéndose los siguientes resultados:

Modelo	Data augmentation	mAP[0.5:0.95]	mAP[0.5]	mAP[0.75]
YOLOv8n	No	0,365	0,482	0,393
	Si	0,319	0,483	0,369
YOLOv8m	No	0,262	0,400	0,283
	Si	0,236	0,413	0,203
D-FINE	No	0,335	0,450	0,377

Tabla 8: Resultados para métrica mAP en nuevo punto de recepción

Estos resultados son bastante desfavorables en comparación con los obtenidos previamente. Nuevamente, el balanceo de los datos no tuvo un impacto positivo. En comparativa, D-FINE presenta una menor disminución de su capacidad predictiva frente a los modelos YOLO y generaliza mejor que YOLOv8m.

Frente a los bajos valores obtenidos, se decide analizar estas métricas solo para la clase de interés “Cargado Rollizos”:

Modelo	Data augmentation	AP[0.5:0.95]	AP[0.5]
YOLOv8n	No	0,805	0,962
	Si	0,733	0,968
YOLOv8m	No	0,642	0,933
	Si	0,396	0,827
D-FINE	No	0,587	-

Tabla 9: Resultados para métricas AP en nuevo punto de recepción para clase “Cargado Rollizos”.

Esta métrica permitió evidenciar la buena capacidad de los modelos YOLO para aprender patrones de los rollizos, al lograr predecir y detectar camiones cargados con rollizos con alta precisión. Se ve como el modelo YOLO en su formato nano es superior a los demás. Esto podría deberse a la menor complejidad del modelo, lo cual permite que se sobreajuste menos al dominio de entrenamiento.

Los camiones cargados con rollizos suelen ser objetos de, por lo menos, tamaño medio. El formato nano fue diseñado con anchors (cajas delimitadoras que sirven como referencia de puntos en la imagen) y escalas que se adaptan adecuadamente a este rango (*Ultralytics YOLO Docs*, 2023). En contraste, los modelos más complejos pueden destinar más capacidad a objetos más pequeños, los cuales suelen estar presentes en la clase “otro”.

La idea que modelos más ligeros obtengan mejores resultados de predicción en dominios fuera del conjunto de entrenamiento, donde cambian el entorno, iluminación y fondo, es algo que se ha estudiado y validado en contextos forestales en trabajos previos. Se ha visto que una menor complejidad y tamaño de modelos ofrecen mejor capacidad de generalización en ambientes distintos y adversos, manteniendo una precisión alta (aproximadamente 80%), mientras que los modelos más pesados decaen (Jin et al., 2024). Resultados similares se han encontrado en otros contextos, como el trabajo de Wu et al. (2024), donde desarrollan una herramienta ligera basada en YOLO para la detección en tiempo real de defectos superficiales en acero (Wu et al., 2024).

Esto sugiere que, en escenarios nuevos, YOLOv8n muestre una capacidad de generalizar patrones característicos y propios de rollizos sin captar ruido específico de los puntos de entrenamiento. Esto es prometedor considerando que se trata de un modelo menos costoso computacionalmente.

Los resultados sobre el nuevo punto de recepción para la clase “Cargado Rollizos” permitieron validar tempranamente la obtención de un modelo que solventa el problema. Por tanto, se apuntó a mejorar las predicciones mediante el reentrenamiento de los modelos luego de las siguientes tareas:

1. Etiquetar mayor cantidad de imágenes del nuevo punto de recepción, buscando un balance en las etiquetas.
2. Etiquetar más imágenes de puntos de recepción anteriores solo en climas adversos o situaciones que dificulten una predicción correcta.
3. Unificar todos los conjuntos de datos.
4. Aplicar el pipeline de augmentación a todas las clases.

Para la primera tarea, dado que el nuevo conjunto de imágenes tenía mayor cantidad de camiones cargados con rollizos (en proporción) que puntos anteriores, se priorizaron imágenes

con condiciones que dificulten su predicción, como la sobreexposición de luz sobre los rollizos o la presencia de polvo:



Ilustración 4: Ejemplo de camiones en condiciones de alta luminosidad y presencia de polvo.

Por otro lado, la cantidad de imágenes disponibles para las demás etiquetas eran limitadas, sobre todo para los camiones vacíos.

Se realizó la selección de imágenes en situaciones adversas de la misma forma en que se hizo anteriormente y se procedió con el etiquetado de este conjunto. Finalmente, se unieron los tres conjuntos.

Con el conjunto de datos final, se decidió llevar a cabo el entrenamiento de los modelos anteriores junto con el modelo de clasificación de imágenes. Para poder entrenar este último fue necesario transformar la data en formato YOLO a una útil para el modelo VGG. Al ser un modelo de clasificación, por cada imagen debe existir únicamente una etiqueta. Por lo que se realizó un análisis de esto sobre el conjunto de datos y se vio que mayoritariamente hay solo una etiqueta por objeto:

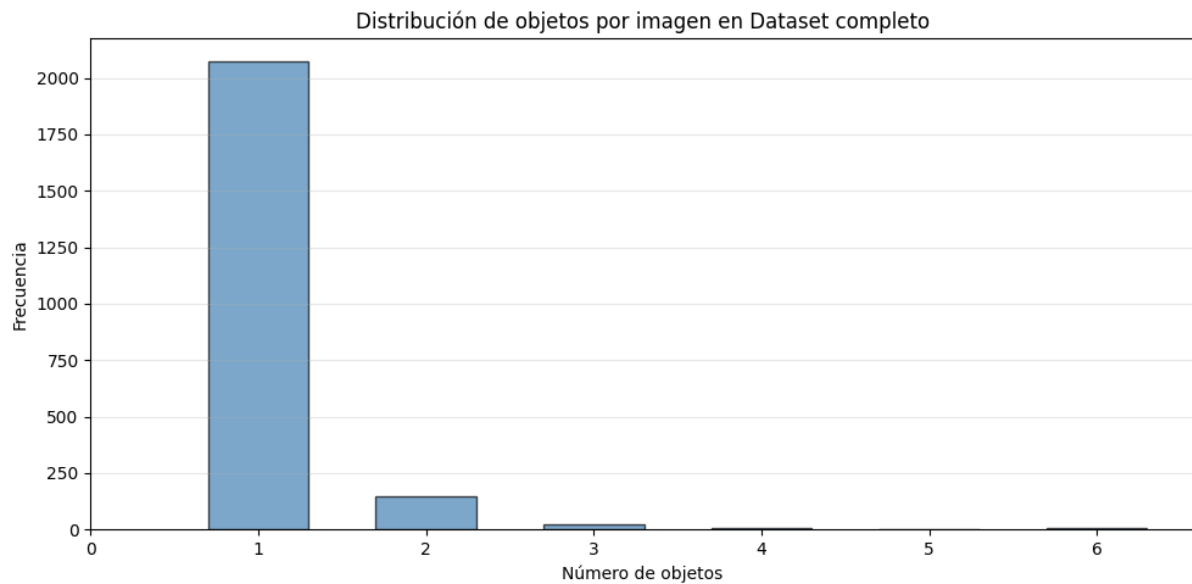


Ilustración 5: Distribución de objetos por imagen en conjunto de datos final.

Esto se justifica en el hecho que las cámaras difícilmente pueden capturar más de un camión de carga en una imagen. Por lo que, para transformar el conjunto de datos a uno útil para el modelo de clasificación, se distribuyó cada una de las imágenes en subdirectorios etiquetados con el nombre de cada una de las clases. Para definir a que clase pertenece una imagen con más de una etiqueta, se tomó la clase de aquella caja delimitadora de mayor área.

Con los conjuntos de datos definidos para cada uno de los modelos solo resta entrenar, evaluar y validar los modelos, definiendo cual es el que mejor desempeño tiene para el problema.

6. Evaluación experimental

6.1 Diseño de experimentos

Para evaluar y validar el desempeño de los modelos entrenados se considera el uso de métricas de precisión de clasificación y de detección de objetos, detalladas en la evaluación de alternativas. También se utilizan las métricas para la evaluación del costo computacional. Esto permite comparar los distintos modelos aplicados al problema de identificación de carga de rollizos en camiones mediante visión artificial.

Por tanto, se definen las variables consideradas:

- **Métricas de evaluación del desempeño:** mAP[0.5:0.95], mAP[0.5] y mAP[0.75] para los modelos de detección de objetos. Matriz de confusión, Precision, Recall y F1-Score.
- **Costo computacional:** GFLOPs, latencia promedio por imagen y tiempo por época de entrenamiento.
- **Modelos:** YOLOv8n, YOLOv8m, D-FINE y VGG-16
- **Dominio de los conjuntos de validación:** Conjunto de prueba del mismo dominio que el entrenamiento (in-domain) y conjunto de prueba de un nuevo punto de recepción (out-domain).

La metodología experimental contempla el entrenamiento de los cuatro modelos. Cada modelo fue entrenado con el conjunto de datos final de casi 2400 etiquetas:

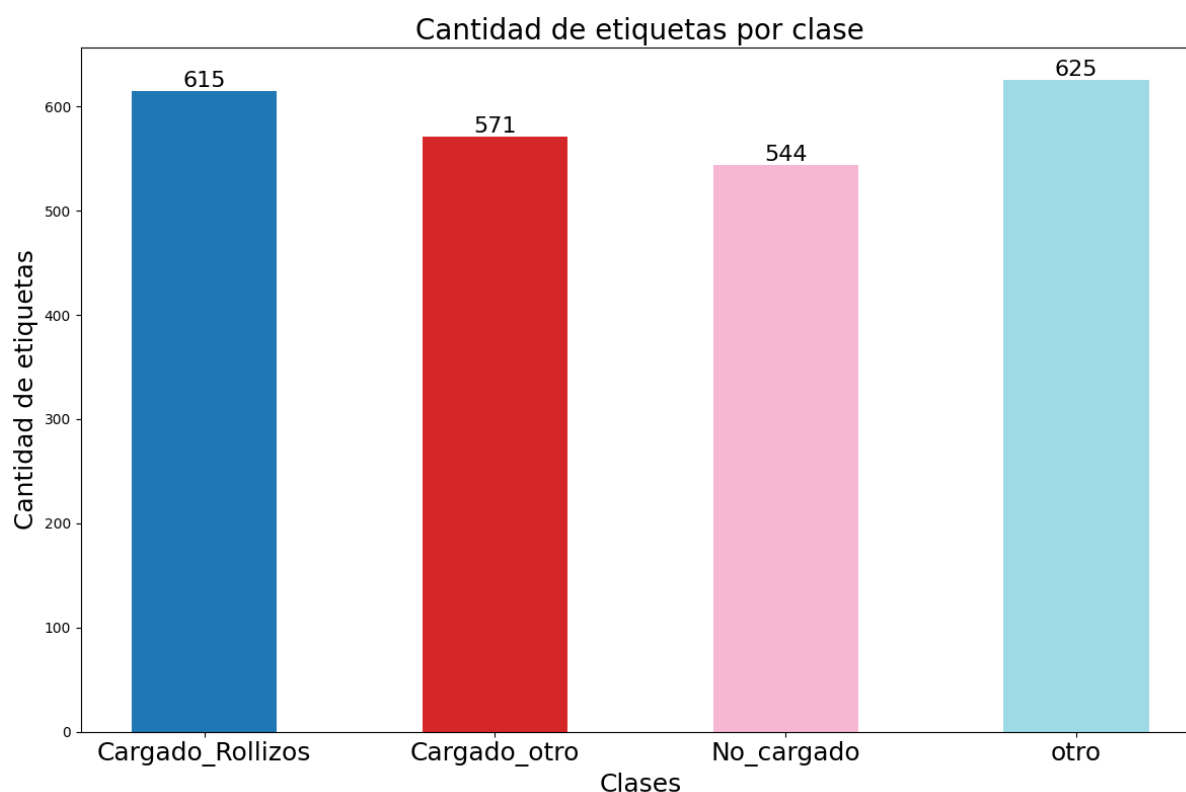


Ilustración 6: Distribución de etiquetas por clase en conjunto de datos final.

Con una división, utilizando una estrategia “Stratified K-Fold” para mantener la distribución de las etiquetas, en proporción: 70% entrenamiento, 15% validación y 15% test.

Aplicando el pipeline sobre el conjunto de entrenamiento realizando un 20% de augmentación para cada clase.

Como se muestra en el desarrollo, se vio que los modelos YOLO en su versión nano logran generalizar de mejor manera y predicen con efectividad la clase de interés en nuevos dominios (out-domain). Por tanto, se busca evaluar el desempeño de los modelos agregando el punto ajeno al dominio de entrenamiento:

- **Evaluación in-domain:** Usando el conjunto de prueba que proviene del mismo dominio (puntos de recepción) utilizado durante el entrenamiento.

Para la evaluación de los modelos de detección (YOLOv8n, YOLOv8m y D-FINE) se utilizaron métricas mAP. Para la evaluación final de desempeño en clasificación se aplicaron métricas tradicionales: matriz de confusión, Precision, Recall y F1-Score. Esto es realizado en modelos entrenados con datos con y sin augmentación. Finalmente se aplican las métricas de costo computacional.

La realización de la fase de experimentación utilizó a nivel de hardware: PC personal Lenovo IdeaPad L340 Gaming, con un procesador i5-9300H (8) @ 2.400GHz, con una memoria RAM 16Gb 2.666 MGz Ddr4 y uso de GPU NVIDIA GeForce GTX 1650.

En cuanto a software, se emplearon diversas bibliotecas y herramientas para el desarrollo, ejecución y experimentación de los modelos:

- Python 3.10.11: Versión seleccionada por su compatibilidad con la versión utilizada de CUDA y permitir el uso de GPU.
- Ultralytics: Utilizada para la implementación y entrenamiento de modelos YOLOv8.
- Huggingface y repositorio de D-FINE.
- Torch y torchvision: Framework para modelo VGG-16 en PyTorch.
- Scikit-learn: Para la evaluación de modelos de clasificación y análisis de métricas.
- Pandas y NumPy: Manipulación de datos tabulares y estructuras numéricas.
- Matplotlib y Seaborn: Visualización de resultados, incluyendo gráficas de métricas y matrices de confusión.
- TensorFlow y Keras: Soporte en algunos casos de comparación y pruebas exploratorias.
- Label Studio: Software utilizado para el etiquetado de imágenes mediante bounding boxes. Se ejecutó a través de Docker para garantizar su compatibilidad y funcionamiento adecuado en sistema operativo Windows.

El principal obstáculo del proceso de experimentación está en los altos tiempos de espera durante la fase de entrenamiento. Lo que conlleva definir con exactitud todos los parámetros necesarios para el correcto entrenamiento de los modelos y evitar fallas. También se tiene el proceso meticuloso de etiquetar manualmente los datos para las imágenes del nuevo punto de recepción. Otro obstáculo presente es la limitación de hardware que ralentiza el proceso de entrenamiento y aumenta la latencia.

Definida la experimentación solo se realizan inferencias sobre los conjuntos de prueba, obteniendo todas las métricas mencionadas. Por lo que se desglosa el tiempo utilizado para cada tarea:

Tarea	Tiempo (en días)
Etiquetado conjunto de datos	1
Entrenamiento de modelos YOLOv8n	1
Entrenamiento de modelos YOLOv8m	2
Entrenamiento de modelos D-FINE	3

Entrenamiento de modelos VGG-16	1
Desarrollo de experimentos	2

Tabla 10: Tiempo estimado de experimentación por tarea.

Este diseño permite una comparativa de los modelos donde se evaluará su efectividad en la identificación de rollizos y capacidad de generalización de los modelos, asegurando que los resultados finales respalden la elección del modelo a utilizar en el sistema de predicción y control de entrada PostCom.

6.2 Desarrollo de experimentos

Inicialmente, se llevó a cabo el etiquetado de las imágenes restantes del punto de recepción nuevo, junto con una mayor cantidad de imágenes en condiciones climáticas adversas, buscando un balance de clases. Obteniendo el conjunto de datos mencionado (Ilustración 6).

Con el conjunto de datos establecido, y habiendo realizado una augmentación para el conjunto de entrenamiento. Se procedió con el entrenamiento de los modelos utilizando la configuración de parámetros por defecto para el modelo YOLO, a excepción de los parámetros “epochs”, “batch” y “patience”. Estos parámetros son ajustados debido a las limitaciones de hardware, donde se definieron 50 épocas de entrenamiento, un “batch” de 8, ajustado a la RAM y GPU del equipo y una paciencia de 5 épocas en caso de no haber mejoras para las métricas de precisión de clasificación. Estos parámetros son replicados en los parámetros equivalentes de los demás modelos.

Se debe mencionar que no se realizó una búsqueda de mejores hiperparámetros debido a las limitaciones de hardware, ya que el proceso se realizó en un equipo personal. Haber realizado esta tarea habría implicado altos tiempos de espera para cada uno de los modelos y sus distintas variantes. Lo que, sumado a los buenos resultados en etapas tempranas, fundamentó esta decisión.

No obstante, se cree que realizar búsqueda de mejores hiperparámetros supondría mejoras significativas en la capacidad de inferencia de los modelos. Se ha visto como esta técnica ha contribuido a mejorar los resultados para las métricas de precisión de clasificación en contextos más complejos al del problema abordado. Tanto en trabajos de detección de objetos (Kafková et al., 2025; Munir et al., 2023) como de clasificación de imágenes (Chua & Lin, 2022; Sharma & Vardhan, 2023).

Teniendo los modelos entrenados, se continuó con el desarrollo necesario para el cálculo de las métricas. Es necesario aclarar que cada modelo se evalúa de forma independiente, ya que cada uno requiere una forma distinta para el cálculo de las métricas. Por lo tanto, se describe el desarrollo experimental para cada modelo.

En el caso de los modelos YOLO, estos permiten el cálculo integrado de todas las métricas. La función de evaluación para un conjunto de datos entrega un diccionario para cada una de las métricas de interés, incluidas las métricas de costo computacional. Esto agilizó el desarrollo para este modelo.

Continuando con el modelo D-FINE, el cálculo de las métricas mAP se realiza a través de una clase personalizada que luego se integra con el sistema de evaluación de Hugging Face Trainer. Esto permite que, durante la evaluación, se calculen y reporten automáticamente las métricas mAP. El proceso de cálculo de esta clase contempla 3 etapas: obtención de los tamaños de las imágenes, preparación de los valores reales para cada una de las imágenes, y procesamiento de las predicciones del modelo en función de un umbral IoU específico.

En cuanto a las métricas de clasificación en D-FINE, se implementa el cálculo de la matriz de confusión, utilizando un umbral IoU de 0,5 para considerar coincidencias válidas. Por cada imagen, se obtienen las predicciones del modelo, se calculan los valores de IoU entre predicciones y valores reales, y se aplica un conteo de verdaderos positivos, falsos positivos y falsos negativos. Con estos valores se calculan las métricas de interés: Precision, Recall y F1-Score.

El cálculo de los GFLOPs utiliza la biblioteca “fvcore” que proporciona herramientas para analizar el rendimiento computacional de modelos de Deep Learning. Entre sus funciones se permite estimar el cálculo de coste computacional por medio de FLOPs. El tiempo de inferencia para cada imagen es entregado directamente por el modelo, permitiendo luego calcular el promedio simple. El tiempo por época es registrado por medio del uso de TensorBoard durante la fase de entrenamiento.

Para el modelo VGG-16, se emplea la biblioteca de scikit-learn que permite el cálculo automatizado de las métricas de precisión. Se procesan los datos del conjunto prueba obteniendo las etiquetas reales y las predichas por el modelo. Estos valores se ingresan como parámetros de entrada en las funciones de scikit-learn para obtener la matriz de confusión y las métricas Precisión, Recall y F1-Score.

El cálculo de GFLOPs para VGG-16 se realiza utilizando la biblioteca “thop”, que permite medir las operaciones de multiplicación-acumulación (MACs). Dado que cada MAC es equivalente a dos FLOPs, los valores son convertidos fácilmente a GFLOPs. En cuanto a la medición de latencia, se mide el tiempo que toma procesar una imagen, se acumulan los tiempos de cada una y se calcula el promedio. Finalmente, el tiempo de cada época se obtiene registrando el tiempo al inicio y final de cada una.

En general, al tratarse del desarrollo de experimentos con métricas estandarizadas, no se tuvieron mayores obstáculos, a excepción del cálculo de la matriz de confusión en el modelo D-FINE, el cual requirió la integración del umbral IoU para considerar una predicción como correcta.

6.3 Análisis de resultados

Para la evaluación de desempeño en el dominio de entrenamiento se utilizó el conjunto de prueba reservado para este fin. Primeramente, se muestran los resultados de los modelos de detección de objetos para las métricas mAP:

Modelo	Data augmentation	mAP[0.5:0.95]	mAP[0.5]	mAP[0.75]
YOLOv8n	No	0,829	0,929	0,886
	Si	0,815	0,923	0,864
YOLOv8m	No	0,832	0,926	0,869
	Si	0,799	0,909	0,849
D-FINE	No	0,687	0,781	0,724
	Si	0,643	0,773	0,713

Tabla 11: Métricas mAP en distintos umbrales IoU en conjunto de datos final.

En general, se tienen buenos resultados, sin embargo, los modelos YOLO muestran ser superiores en la predicción de clasificación y asignación del cuadro delimitador para un objeto. La augmentación muestra no ser de ayuda, disminuyendo levemente los valores para cada modelo. El modelo YOLOv8n muestra ser más robusto y estable, mostrando una menor caída relativa en todas las métricas frente al efecto de la augmentación, por lo que se infiere que es menos sensible a las transformaciones. Esto indica que YOLOv8m parece sobre ajustarse más a las condiciones naturales del entrenamiento, perdiendo mayor porcentaje de mAP.

En cuanto a D-FINE, a pesar de tener buenos resultados, sus cifras son significativamente menores a las de los demás modelos. Lo que sugiere que su capacidad de generalización y refinamiento fino no compensa su uso para este conjunto de datos, donde los objetos de interés son de gran área en la imagen a predecir.

Ahora, se evalúan los modelos en cuanto a métricas de precisión de clasificación sobre las clases definidas, permitiendo comparar el desempeño de los 4 modelos.

Modelo	Data augmentation	Precision	Recall	F1-score Macro
YOLOv8n	No	0,870	0,885	0,876
	Si	0,879	0,865	0,872
YOLOv8m	No	0,883	0,873	0,878
	Si	0,860	0,838	0,846
D-FINE	No	0,827	0,642	0,640
	Si	0,803	0,627	0,623
Vgg-16	No	0,678	0,677	0,677
	Si	0,665	0,663	0,656

Tabla 12: Métricas de evaluación para clasificación de clases.

Nuevamente los modelos YOLO muestran superioridad, teniendo variaciones ligeras uno de otro. De forma similar a lo mostrado por las métricas mAP, el modelo YOLO en su formato nano muestra ser menos sensible a la augmentación. En particular, YOLOv8m presentó los valores más altos sin aplicar data augmentation, lo que refleja su alto desempeño base. Sin embargo, bajo el conjunto con augmentación, su rendimiento decreció notablemente, bajando su F1-Score de 0,878 a 0,846. Por otro lado, YOLOv8n refuerza la idea de ser más robusto al demostrar mayor estabilidad frente a las transformaciones introducidas, con una menor disminución en su desempeño (F1-Score: de 0,876 a 0,872), lo que sugiere que su arquitectura más ligera generaliza mejor ante nuevas condiciones visuales y variaciones sintéticas.

El modelo D-FINE, presentó una alta Precision, pero un Recall considerablemente más bajo en ambos modelos entrenados, lo que indica que, si bien sus predicciones tienden a ser correctas, omite una gran cantidad de instancias verdaderas, disminuyendo su capacidad de generalización. Este comportamiento puede atribuirse a una mayor sensibilidad a las condiciones del entorno forestal. Esto refuerza la necesidad de realizar búsqueda de mejores hiperparámetros para mejorar los resultados de clasificación.

Por su parte, el modelo VGG-16 mostró resultados aceptables pero inferiores en todos los indicadores. Esto es esperable dado que se trata de una arquitectura enfocada en clasificación de imagen completa, sin realizar localización de objetos, lo que limita su eficacia en contextos donde se requiere identificar y diferenciar instancias específicas en escenas complejas al aire libre.

En suma, los modelos YOLOv8 sobresalen como los más efectivos en el conjunto de prueba, logrando los mayores valores de F1-Score. Dentro de esta familia, YOLOv8n se posiciona

como el candidato más estable y robusto, al mantener un alto desempeño con y sin data augmentation. A esto se añade el hecho que, previamente, fue superior a la versión media al evaluar con imágenes fuera del dominio de entrenamiento, como se puede ver en la Tabla 9, donde fue superior tanto con datos aumentados como no aumentados.

Evaluados los modelos en cuanto a su efectividad de predicción, se presentan los resultados en cuanto a la medición de costo computacional asociado al entrenamiento y uso de los modelos considerando los indicadores definidos en el diseño de experimentación.

Los resultados se resumen en la siguiente tabla:

Modelo	GFLOPs	Latencia promedio	Tiempo por época
YOLOv8n	8,1	10,4 ms	1,3 min
YOLOv8m	78,7	27,2 ms	43,0 min
D-FINE	30,17	153,9 ms	65,1 min
Vgg-16	30,9	24,9 ms	1,2 min

Tabla 13: Métricas de evaluación para costo computacional.

Hay que mencionar que el cálculo de la latencia promedio se realizó en 100 iteraciones de inferencia para el conjunto de prueba, con fin de disminuir el efecto del ruido y variaciones de tiempo.

El modelo YOLOv8n destaca por su bajo costo computacional en todos los indicadores: presenta la menor cantidad de GFLOPs, la menor latencia y un tiempo por época muy reducido. Esto confirma que este modelo es altamente eficiente y adecuado para implementaciones en tiempo real y entornos con recursos limitados.

En contraste, YOLOv8m muestra un incremento significativo en el costo computacional, particularmente en GFLOPs y en el tiempo por época. Esto es esperable debido a su arquitectura con mayor número de parámetros que su formato nano.

D-FINE, aunque menos costoso en GFLOPs que YOLOv8m, presenta la mayor latencia y el mayor tiempo de entrenamiento. Esto se debe a su arquitectura basada en Transformers, que, si bien busca mejorar la precisión de los cuadros delimitadores, demanda más recursos computacionales y es menos eficiente para tareas en tiempo real.

Por otro lado, el modelo VGG-16 exhibe un perfil de costo computacional muy bajo en cuanto a tiempo de entrenamiento y una latencia moderada. A pesar de tener GFLOPs similares a D-

FINE, su eficiencia en entrenamiento lo convierte en una buena opción para tareas de clasificación livianas.

En conjunto con los resultados de precisión de clasificación se concluye que el modelo YOLOv8n representa el mejor equilibrio entre precisión y eficiencia computacional, lo que refuerza su elección como modelo candidato para ser implementado en terreno.

7. Operación

La transición de un modelo experimental a una herramienta operativa en un entorno forestal requiere un proceso de implementación y mantenimiento estructurado. Este proceso está diseñado para garantizar la robustez, fiabilidad y mejora continua del sistema de identificación de carga de rollizos a reemplazar, asegurando una integración fluida con las operaciones existentes del sistema PostCom.

A continuación, se detallan las fases de implementación y el plan de mantenimiento propuesto para el modelo YOLOv8n.

7.1 Estrategia de implementación

La implementación se concibe como un proceso iterativo y controlado, dividido en tres fases principales, que busca minimizar los riesgos y validar el rendimiento del modelo en condiciones reales antes de un despliegue completo. El flujo propuesto es el siguiente:

1. **Fase 1: Entrenamiento y validación inicial:** El punto de partida es el modelo YOLOv8n entrenado con el conjunto de datos final, como se detalla en la sección 5. Desarrollo del proyecto. El primer paso de la implementación consiste en realizar una validación adicional del modelo utilizando un conjunto de imágenes recientes capturadas por el sistema PostCom que no hayan sido parte del entrenamiento (Out-Domain). El objetivo es evaluar la utilidad y la capacidad de generalización del modelo en nuevos puntos actuales. Si el rendimiento (medido con métricas usadas en experimentación) no cumple con un umbral de aceptación predefinido, el modelo regresará a una fase de reentrenamiento, incorporando estas nuevas imágenes para mejorar la robustez.
2. **Fase 2: Puesta en marcha blanca y análisis comparativo:** Una vez que el modelo demuestra ser útil en la validación inicial, se procede a una "puesta en marcha blanca". Durante esta fase, el modelo YOLOv8n se despliega en un punto de recepción seleccionado y operará en paralelo al sistema actual. El sistema propuesto realizará predicciones en tiempo real, donde resultados serán registrados y almacenados. El propósito es evaluar el desempeño del modelo en un flujo continuo y en un entorno de producción sin interrumpir las operaciones. Además de realizar una comparación directa y cuantitativa con el sistema actual, analizando la tasa de aciertos, falsos positivos y falsos negativos de ambos sistemas bajo las mismas condiciones.
3. **Fase 3: Reemplazo y despliegue definitivo:** Si durante la puesta en marcha blanca el modelo YOLOv8n demuestra ser consistentemente superior al sistema actual, se

procederá con el reemplazo definitivo en el punto de recepción piloto. En caso contrario, se analizarán los fallos específicos y se volverá a la fase de reentrenamiento. Una vez validado el éxito en un primer punto, el modelo se convierte en un candidato para el despliegue escalonado en otros puntos de recepción.

Se resume la estrategia de implementación en la siguiente figura:

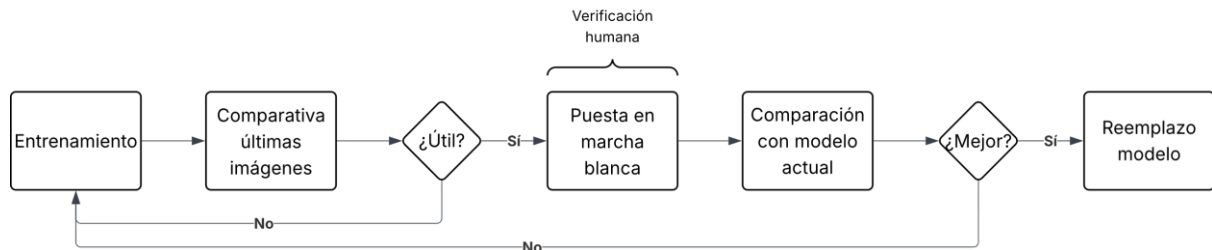


Ilustración 7: Estrategia de implementación.

7.2 Arquitectura e integración del modelo

Para que el modelo YOLOv8n pase de ser un prototipo experimental a una herramienta productiva, es fundamental definir su integración dentro del ecosistema tecnológico de Ecofor. Esta sección detalla la arquitectura propuesta, los flujos de datos y el plan técnico para su despliegue.

7.2.1 Arquitectura del sistema e interoperabilidad

El modelo no se inserta como componente dentro de un servidor central, sino como un agente de inferencia en el borde (Edge AI). Esta decisión se basa en la necesidad de procesar imágenes en tiempo real y de forma eficiente directamente en los puntos de recepción.

Se utilizará una placa BeagleY-AI, un computador de placa única diseñado para aplicaciones de visión por computador. Sus características permiten ejecutar el modelo YOLOv8n de manera local y optimizada, sin depender de conectividad a internet para la inferencia.

La arquitectura general se compone de dos entornos: el borde (punto de recepción) y la nube (AWS), como se muestra en la siguiente figura:

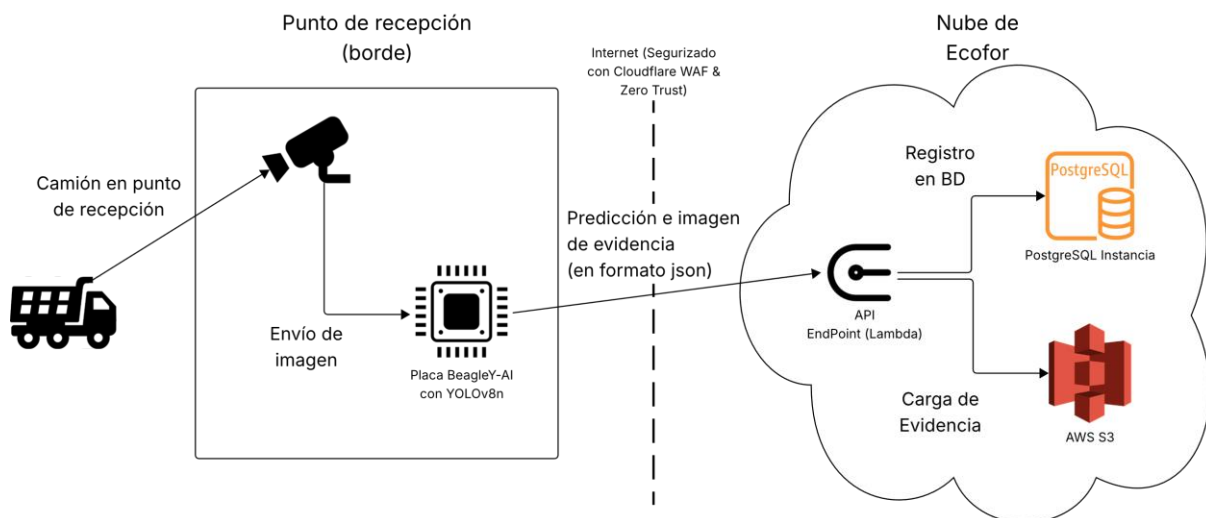


Ilustración 8: Diagrama de Arquitectura

Flujo de interoperabilidad:

1. **Captura en el borde:** Las cámaras de PostCom capturan una imagen del camión y la envían directamente a la placa BeagleY-AI conectada en la red local del punto de recepción.
2. **Inferencia local:** El servicio con el modelo YOLOv8n, ejecutándose en la placa, procesa la imagen. Realiza la predicción de la carga de forma local, obteniéndose la clase, el nivel de confianza y las coordenadas del cuadro delimitador.
3. **Envío de resultados:** Tras la inferencia, el servicio en la BeagleY-AI empaqueta los metadatos de la predicción (clase, confianza, etc.) y la imagen de evidencia (codificada) en un único objeto JSON. Este objeto es enviado en una sola petición segura al endpoint de API gestionado por AWS Lambda.
4. **Procesamiento en la nube:** Al recibir la petición, la función AWS Lambda actúa como un orquestador. Su lógica ejecuta las siguientes acciones en secuencia:
 - a. Decodifica la imagen de evidencia y la carga en el bucket de AWS S3, obteniendo una URL única.
 - b. Combina esta URL con los metadatos de la predicción.
 - c. Registra la información completa (referencia a la evidencia y resultado de la predicción) en la base de datos PostgreSQL.
5. **Consolidación:** Una vez que los datos están en la base de datos, el sistema Trazo/PostCom puede acceder a ellos de forma inmediata. El resultado de la clasificación se muestra en la interfaz de usuario final, reemplazando la necesidad de la auditoría manual y proporcionando una validación automática y en tiempo real.

7.2.2 Flujo y acceso a datos

Para la obtención de los datos de entrada al modelo, desplegado en la placa BeagleY-AI, obtiene las imágenes directamente de las cámaras IP de PostCom en la red local. Este enfoque minimiza la latencia, ya que la imagen no necesita viajar a la nube para ser procesada. La copia que se envía a AWS S3 sirve como evidencia y respaldo histórico.

El acceso a los datos generados (salidas) es distinguido en dos apartados:

- **Acceso para sistemas (Trazo y PostCom):** Los sistemas centrales no interactúan directamente con el modelo. En su lugar, consumen los resultados de forma asíncrona desde la base de datos PostgreSQL, la cual es actualizada por la función AWS Lambda. Este patrón de diseño (escritura vía API, lectura desde BD) desacopla los componentes y mantiene la integridad y seguridad de la base de datos.
- **Acceso para usuarios finales:** Los usuarios (clientes forestales) acceden a los resultados a través de una plataforma web de PostCom. En la interfaz, podrán visualizar las imágenes de los camiones junto con los resultados de la clasificación de carga. Esto reemplaza la validación manual posterior y agiliza la visibilidad de la operación.

7.2.3 Plan de integración y despliegue técnico

El plan de integración se concibe en fases para asegurar una transición controlada y sin impacto en la operación actual.

- **Fase 1: Preparación del entorno y modelo para el borde.**
 - **Optimización del modelo:** El modelo YOLOv8n entrenado (en formato .pt) deberá ser compilado y cuantizado utilizando las herramientas del toolchain de Texas Instruments para generar un artefacto optimizado que se ejecute eficientemente en la placa BeagleY-AI.
 - **Despliegue del endpoint:** Se desarrollará y desplegará la función AWS Lambda que actuará como endpoint para recibir las predicciones. Este endpoint será securizado utilizando servicios de Cloudflare WAF y Zero Trust.
- **Fase 2: Puesta en marcha blanca.**
 - Se instalará y configurará la placa en un punto de recepción piloto.
 - El servicio de inferencia operará en "modo sombra": procesará las imágenes en tiempo real, pero enviará sus predicciones a una tabla o base de datos de prueba, no a la de producción.
 - En paralelo, se continuará realizando una auditoría manual.

El objetivo de esta fase es comparar el rendimiento del modelo con los resultados manuales durante un periodo definido, validar su precisión en un entorno real y realizar ajustes necesarios.

- **Fase 3: Despliegue definitivo y activación.**

- Una vez validada la superioridad y fiabilidad del modelo, se procederá a la activación. La función AWS Lambda se reconfigurará para que escriba los resultados en la base de datos de producción de Trazo/PostCom.
- En el punto de recepción piloto, el proceso de validación manual para la clasificación de carga será formalmente reemplazado por el sistema automático.

- **Fase 4: Escalado.**

- Tras el éxito en el punto piloto, el plan de despliegue se extenderá de forma gradual al resto de los puntos de recepción, instalando una placa BeagleY-AI en cada uno. El mantenimiento y monitoreo del sistema serán gestionados por el área de mesa de ayuda de Ecofor, que canalizará las incidencias que puedan surgir.

7.3 Mantenimiento y monitoreo continuo

Un modelo de Deep Learning puede sufrir una degradación en su rendimiento con el tiempo debido a cambios en la distribución de los datos de entrada, como nuevas condiciones climáticas, tipos de camiones o cambios en la iluminación. Por ello, se propone un plan de mantenimiento proactivo:

1. **Monitoreo de rendimiento:** Tras el despliegue, el sistema deberá registrar continuamente la tasa de éxito de sus predicciones. Esto se puede lograr mediante validaciones humanas periódicas y muestreadas. Se medirá el F1-Score y la precisión por cada punto de recepción por periodos de tiempos a definir por Ecofor.
2. **Ciclo de reentrenamiento:** Si se detecta que el rendimiento en un punto de recepción específico cae por debajo de un umbral de calidad establecido (por ejemplo, un F1-Score inferior al 85% para la clase de interés), se activará un protocolo de reentrenamiento. Este proceso consistirá en recolectar las imágenes donde el modelo ha fallado, etiquetarlas y utilizarlas para reentrenar o ajustar (fine-tune) el modelo YOLOv8n. Este enfoque dirigido asegura que el modelo se adapte a las condiciones específicas del lugar que presenta problemas, optimizando los recursos de entrenamiento.

Se resume la estrategia de mantenimiento y monitoreo en la siguiente figura:

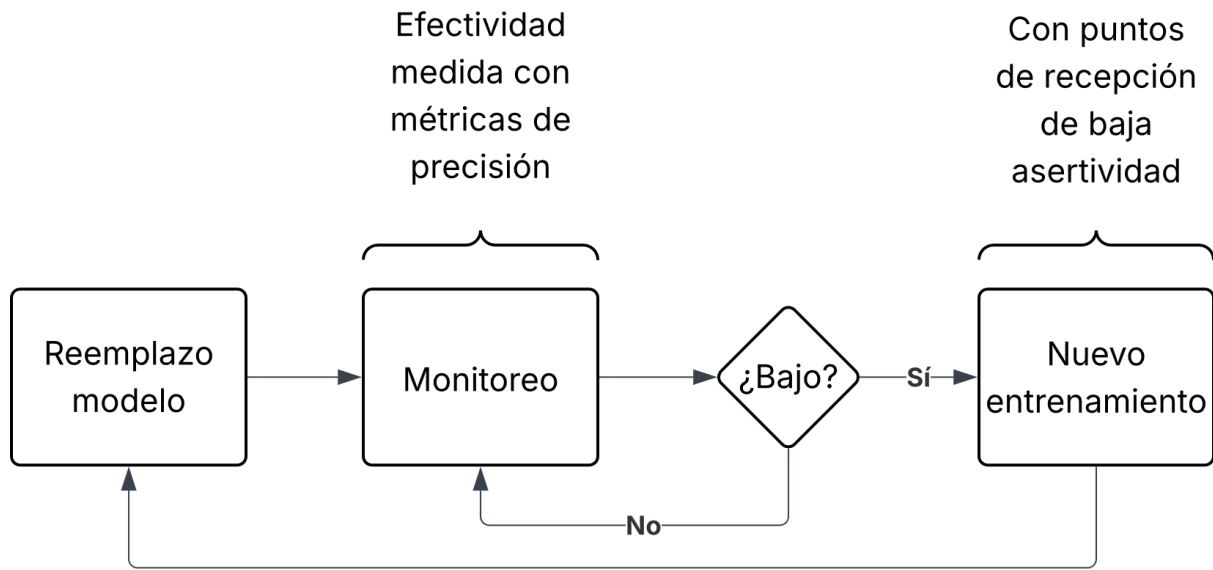


Ilustración 9: Estrategia de mantenimiento y monitoreo continuo.

7.4 Consideraciones adicionales de operación

- **Aspectos técnicos:** El entorno de hardware actual utilizado por el sistema PostCom es viable para la ejecución en inferencia del modelo YOLOv8n, dado su bajo costo computacional (8,1 GFLOPs). Sin embargo, para escalar la solución a múltiples puntos de recepción y agilizar los ciclos de reentrenamiento, se recomienda una eventual mejora del hardware de servidor, principalmente en la capacidad de la GPU.
- **Gestión del cambio y capacitación:** La adopción exitosa de esta nueva tecnología requiere una gestión del cambio adecuada. Se contempla una presentación formal de los resultados y del funcionamiento del modelo al personal técnico y directivo de Ecofor. Si bien el sistema es autónomo, se elaborará una documentación concisa que permita al equipo técnico de la empresa comprender el funcionamiento del modelo y, si fuese necesario, ejecutar los scripts de reentrenamiento. La usabilidad no se ve afectada, ya que el sistema opera de forma transparente para el usuario final.

8. Análisis de impacto

La implementación de un sistema de identificación de carga de rollizos basado en modelos de detección de objetos es innovadora en el área, pero también genera una serie de impactos de dimensiones económicas, ambientales, éticas y sociales que deben ser analizados.

8.1 Análisis de impacto económico

El análisis económico del presente proyecto, si bien es de carácter teórico al no disponer de cifras, permite proyectar beneficios significativos para la empresa y el sector forestal.

1. **Reducción de costos operativos directos:** El principal impacto económico radica en la automatización de una tarea de validación que actualmente requiere de supervisión humana y es propensa a errores. Al reemplazar el sistema actual, cuya precisión es baja (entre 45% y 65%), por un modelo con una alta tasa de acierto, se elimina la necesidad de tener personal dedicado exclusivamente a la verificación manual en los puntos de recepción. Esto se traduce en un ahorro directo en costos de personal, permitiendo a la empresa reasignar estos recursos humanos a tareas de mayor valor.
2. **Escalabilidad y retorno de la inversión (ROI):** La solución basada en el modelo YOLOv8n es altamente escalable. Una vez desarrollado y validado, el costo marginal de implementarlo en nuevos puntos de recepción es relativamente bajo, limitándose principalmente a la configuración del hardware, esto sin considerar el coste de los materiales asociados a las cámaras y demás materiales requeridos. Esto contrasta con el costo de contratar y capacitar personal para cada nueva ubicación. Por tanto, se proyecta un alto Retorno de la Inversión, donde la inversión inicial en desarrollo, entrenamiento y despliegue se recupera rápidamente a través de los ahorros operativos recurrentes y la capacidad de expansión.
3. **Mitigación de pérdidas por robo de madera:** Como se establece en la introducción de este informe, el robo de madera genera pérdidas millonarias para la industria. Un sistema de identificación automatizado, preciso y fiable como el propuesto, fortalece la cadena de trazabilidad de la madera. Al dificultar el ingreso de camiones con carga ilegal o no declarada, el sistema contribuye directamente a la mitigación de pérdidas económicas asociadas a este ilícito.

8.2 Análisis de impacto ambiental

El desarrollo e implementación de tecnologías de Inteligencia Artificial conlleva una huella ambiental que es importante considerar, centrada principalmente en el consumo energético.

- **Consumo energético en el entrenamiento:** El entrenamiento de modelos de Deep Learning, como YOLOv8, es un proceso computacionalmente intensivo que demanda un alto consumo de energía, especialmente por parte de las Unidades de Procesamiento Gráfico (GPU). Durante el desarrollo de este proyecto, se realizaron múltiples ciclos de entrenamiento para alcanzar el rendimiento óptimo, lo que implicó un consumo energético considerable.
- **Mitigación a través de la eficiencia:** Tomando consciencia de este impacto, una de las razones fundamentales para la elección del modelo YOLOv8 en formato nano fue su eficiencia. Este modelo representa un equilibrio entre alta precisión y bajo costo computacional, requiriendo significativamente menos recursos para entrenar en comparación con modelos más grandes (como YOLOv8m o arquitecturas más pesadas). Además, una vez entrenado, el costo energético de la inferencia (la predicción en tiempo real) es mínimo, lo que lo hace sostenible para una operación continua. Los ciclos de reentrenamiento, aunque necesarios, serán esporádicos y localizados, limitando así el impacto ambiental a lo largo del ciclo de vida del producto.

8.3 Análisis de implicaciones éticas y responsabilidad social

La implementación de esta tecnología trasciende lo técnico y económico, presentando importantes implicaciones éticas y de responsabilidad social.

Existe una contribución a la lucha contra el crimen organizado. El robo de madera en Chile no es solo un problema económico, sino también un problema social profundo, vinculado a la violencia y al crimen organizado (Rodríguez Jiménez, 2024). Al proporcionar una herramienta que mejora la trazabilidad y el control en la cadena de suministro forestal, este proyecto contribuye de manera directa a la lucha contra esta actividad ilícita. Un sistema automatizado y difícil de corromper fortalece el estado de derecho y promueve un comercio más justo y transparente.

Se tiene una reducción del riesgo de colusión y corrupción al automatizar el proceso de verificación de carga, se reduce drásticamente el "factor humano" en un punto crítico y vulnerable del proceso. Esto minimiza el riesgo de que un trabajador sea objeto de sobornos o presiones por parte de bandas criminales para permitir el paso de madera robada, protegiendo tanto la integridad de la empresa como la del propio empleado.

Por otro lado, hay un impacto en el empleo y la responsabilidad empresarial. Una implicación ética ineludible de la automatización es el desplazamiento de puestos de trabajo. La misma eficiencia que genera ahorros económicos implica que la labor de verificación manual ya no

será necesaria. Desde una perspectiva de responsabilidad social, es imperativo que la empresa implementadora, gestione esta transición de manera ética. Se recomienda la elaboración de un plan para los trabajadores afectados, que podría incluir programas de recapitación para asumir nuevos roles dentro de la organización (por ejemplo, en el monitoreo del sistema de IA, la gestión de datos o el mantenimiento técnico) o la reasignación a otras áreas de la compañía. Abordar este desafío de forma proactiva es fundamental para una implementación tecnológica socialmente responsable.

9. Conclusiones

La presente memoria de título abordó el desafío de reemplazar un sistema deficiente de identificación de carga de rollizos en el sector forestal chileno, cuyo rendimiento no solo era bajo en precisión, sino que además imponía un significativo riesgo laboral y operacional. El objetivo principal fue desarrollar y validar una herramienta basada en Deep Learning, capaz de automatizar esta tarea de manera fiable, precisa y en tiempo real.

A lo largo de este trabajo, se llevó a cabo un proceso metodológico y riguroso que abarcó desde la recolección y etiquetado exhaustivo de un conjunto de datos de casi 2400 imágenes hasta el entrenamiento y la evaluación comparativa de múltiples arquitecturas de detección de objetos (YOLOv8n, YOLOv8m, D-FINE) y clasificación (VGG-16). Los resultados experimentales demostraron de manera concluyente la superioridad de los modelos de la familia YOLO. Específicamente, el modelo YOLOv8n fue la solución óptima, presentando el mejor equilibrio entre una alta precisión, con un F1-Score macro de 0,876 y un mAP[0.5] de 0,929 en el conjunto de prueba in-domain, y una notable eficiencia computacional, con solo 8,1 GFLOPs y una latencia de 10,4 ms por inferencia.

Se concluye que el modelo YOLOv8n no solo cumple con los requisitos técnicos para ser implementado en un entorno de producción, sino que representa la solución más robusta y estable. Su menor sensibilidad a la aumentación de datos en comparación con su variante media (YOLOv8m) y su capacidad superior para generalizar en dominios no vistos lo posicionan como el candidato idóneo para un despliegue escalable y de bajo mantenimiento.

Más allá del rendimiento técnico, el éxito de este proyecto debe ser medido a la luz del criterio laboral establecido como eje central en la evaluación de la solución. El sistema propuesto no fue diseñado únicamente para ser más preciso, sino para resolver una problemática humana fundamental: eliminar la exposición del personal a un entorno de riesgo. La dependencia de la validación manual posicionaba al trabajador en una situación de vulnerabilidad frente a la coacción y la colusión por parte de bandas de crimen organizado, un riesgo tangible y significativo en el contexto del robo de madera. Al desarrollar una herramienta con la autonomía y fiabilidad suficientes para reemplazar la intervención humana en este punto de control, se cumple con el objetivo primordial de mitigar dicho riesgo laboral. La solución, por tanto, no es solo una optimización tecnológica, sino una medida de protección directa para los empleados y un refuerzo a la integridad de la operación.

En suma, este proyecto demuestra con éxito la viabilidad de aplicar modelos de Deep Learning para resolver problemas complejos y de alto impacto en la industria forestal. La solución

propuesta no solo ofrece beneficios económicos tangibles a través de la reducción de costos y la mitigación de pérdidas, sino que también aporta valor desde una perspectiva social y ética, al contribuir a la trazabilidad de la madera y a la seguridad de los trabajadores. No obstante, se reconoce la responsabilidad ineludible de gestionar el impacto de la automatización en el empleo, recomendando la implementación de programas de reconversión laboral para el personal afectado.

La empresa ha manifestado su conformidad con los resultados obtenidos, y su mira está puesta en la implementación y prueba del modelo YOLOv8n en un entorno productivo, validando la relevancia y aplicabilidad del proyecto.

De cara al futuro, el camino está trazado para una implementación gradual siguiendo el plan de operación descrito, comenzando con una marcha blanca y extendiéndose a otros puntos de recepción. Las líneas de trabajo futuras deberían centrarse en el desarrollo de un ciclo del monitoreo y reentrenamiento continuo del modelo, así como en la expansión del conjunto de datos para incluir una mayor variabilidad de condiciones, asegurando así la evolución y robustez del sistema a largo plazo.

10. Bibliografía

- Carratù, M., Gallo, V., Liguori, C., Lundgren, J., O’Nils, M., & Pietrosanto, A. (2023). Vision-Based System for Measuring the Diameter of Wood Logs. *IEEE Open Journal of Instrumentation and Measurement*, 2, 1–12. <https://doi.org/10.1109/OJIM.2023.3264042>
- Casas, G. G., Ismail, Z. H., Limeira, M. M. C., da Silva, A. A. L., & Leite, H. G. (2023a). Automatic Detection and Counting of Stacked Eucalypt Timber Using the YOLOv8 Model. *Forests*, 14(12), 2369. <https://doi.org/10.3390/f14122369>
- Casas, G. G., Ismail, Z. H., Limeira, M. M. C., da Silva, A. A. L., & Leite, H. G. (2023b). Automatic Detection and Counting of Stacked Eucalypt Timber Using the YOLOv8 Model. *Forests*, 14(12), 2369. <https://doi.org/10.3390/f14122369>
- Chua, Y. Y., & Lin, L. P. (2022). Hyperparameter-Tuned CNN for the Classification of Ten Tomato Plant Diseases. *2022 2nd International Conference on Electronic and Electrical Engineering and Intelligent System (ICE3IS)*, 1–6. <https://doi.org/10.1109/ICE3IS56585.2022.10010269>
- da Silva, D. Q., dos Santos, F. N., Sousa, A. J., & Filipe, V. (2021). Visible and Thermal Image-Based Trunk Detection with Deep Learning for Forestry Mobile Robotics. *Journal of Imaging*, 7(9), 176. <https://doi.org/10.3390/jimaging7090176>
- Dirección Meteorológica de Chile. (n.d.). *Dirección Meteorológica de Chile*. Retrieved July 6, 2025, from <https://climatologia.meteochile.gob.cl/>
- Distante, A., & Distante, C. (2020). *Handbook of Image Processing and Computer Vision* (4th ed., Vol. 3). Springer International Publishing. <https://doi.org/10.1007/978-3-030-42378-0>
- Emily Avendaño. (2022, June 13). Investigación en el epicentro del robo de madera en la Macrozona: cómo operan las mafias y cómo el Estado las deja actuar. *El Libero*.
- GARCÉS, A., PAREDES, M., JIMENEZ, H., LAGOS-ZUÑIGA, M., REBOLLEDO, S., LARA, M., JEREZ-TOLEDO, C., & SILVA, D. (2023). *Sociedad Chilena de Ingeniería Hidráulica XXVI Congreso Chileno de Ingeniería Hidráulica*.
- González, E. (2024, May 19). “El ISIS de Cayucupil”: las escuchas que develan el violento robo de madera en Arauco. *La Tercera*. <https://www.latercera.com/la-tercera-domingo/noticia/el-isis-de-cayucupil-las-escuchas-que-develan-el-violento-robo-de-madera-en-arauco/2AZ272O3WRC7HD4MXGPURI5YVE/#>
- Jin, L., Yu, Y., Zhou, J., Bai, D., Lin, H., & Zhou, H. (2024). SWVR: A Lightweight Deep Learning Algorithm for Forest Fire Detection and Recognition. *Forests*, 15(1), 204. <https://doi.org/10.3390/f15010204>
- Kafková, J., Pirník, R., Janota, A., & Kuchár, P. (2025). Hyperparameter Tuning for Effective Passenger Detection. *2025 26th International Carpathian Control Conference (ICCC)*, 1–5. <https://doi.org/10.1109/ICCC65605.2025.11022848>

- Kaur, P., Khehra, B. S., & Mavi, Er. B. S. (2021). Data Augmentation for Object Detection: A Review. *2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, 537–543. <https://doi.org/10.1109/MWSCAS47672.2021.9531849>
- Lei, C., Hu, B., Wang, D., Zhang, S., & Chen, Z. (2019). A Preliminary Study on Data Augmentation of Deep Learning for Image Classification. *Proceedings of the 11th Asia-Pacific Symposium on Internetware*, 1–6. <https://doi.org/10.1145/3361242.3361259>
- Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., & Dollár, P. (2015). *Microsoft COCO: Common Objects in Context*.
- Lin, Y., Cai, R., Lin, P., & Cheng, S. (2022). A detection approach for bundled log ends using K-median clustering and improved YOLOv4-Tiny network. *Computers and Electronics in Agriculture*, 194, 106700. <https://doi.org/10.1016/j.compag.2022.106700>
- Liu, G. (2020). Surface Defect Detection Methods Based on Deep Learning: a Brief Review. *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*, 200–203. <https://doi.org/10.1109/ITCA52113.2020.00049>
- Liu, Q., Feng, C., Song, Z., Louis, J., & Zhou, J. (2019). Deep Learning Model Comparison for Vision-Based Classification of Full/Empty-Load Trucks in Earthmoving Operations. *Applied Sciences*, 9(22), 4871. <https://doi.org/10.3390/app9224871>
- Lu, J., Tan, L., & Jiang, H. (2021). Review on Convolutional Neural Network (CNN) Applied to Plant Leaf Disease Classification. *Agriculture*, 11(8), 707. <https://doi.org/10.3390/agriculture11080707>
- Lu, Z., Yao, H., Lyu, Y., He, S., Ning, H., Yu, Y., Zhai, L., & Zhou, L. (2024). A Deep Learning Method for Log Diameter Measurement Using Wood Images Based on Yolov3 and DeepLabv3+. *Forests*, 15(5), 755. <https://doi.org/10.3390/f15050755>
- Mekled, A. S., Abdennadher, S., & Shehata, O. M. (2024). Performance Evaluation of YOLO Models in Varying Conditions: A Study on Object Detection and Tracking. *2024 International Conference on Computer and Applications (ICCA)*, 1–6. <https://doi.org/10.1109/ICCA62237.2024.10927807>
- Michael, T. (2021, August 21). Madereros de Chile ensayan nuevas normas contra el robo de madera. *InSightCrime*. <https://insightcrime.org/es/noticias/madereros-chile-ensayan-nuevas-normas-contra-robo-madera/>
- Mishra, A., Mishra, A., Tewari, A. K., & Gangrade, J. (2023). Deep Transfer Learning for Tomato Leaf Diseases Detection and Classification using Pre-trained Models. *2023 9th International Conference on Signal Processing and Communication (ICSC)*, 290–295. <https://doi.org/10.1109/ICSC60394.2023.10441215>
- Munir, N. S., Hossain, N., Zame, R. R., & Sarowar, G. (2023). Vehicle Detection of Bangladesh using YOLOv7 with Hyper-parameter Tuning. *2023 3rd International Conference on Artificial Intelligence and Signal Processing (AISP)*, 1–5. <https://doi.org/10.1109/AISP57993.2023.10135057>

- Peng, Y., Li, H., Wu, P., Zhang, Y., Sun, X., & Wu, F. (2024). *D-FINE: Redefine Regression Task in DETRs as Fine-grained Distribution Refinement*.
- Rodríguez Jiménez, C. (2024). Tráfico ilícito de madera en la Macrozona Sur de Chile: Análisis de la fenomenología y sus actores criminales. *Quadernos de Criminología (Segunda Época)*, 1, 18–26. <https://doi.org/https://doi.org/10.5281/zenodo.10565530>
- Sharma, S., & Vardhan, M. (2023). Hyperparameter Tuned Hybrid Convolutional Neural Network (H-CNN) for Accurate Plant Disease Classification. *2023 International Conference on Communication, Circuits, and Systems (IC3S)*, 1–6. <https://doi.org/10.1109/IC3S57698.2023.10169257>
- Sikora, J., & Fojtík, D. (2020). Classification of Timber Load on Trucks. *Advances in Science, Technology and Engineering Systems Journal*, 5(2), 683–687. <https://doi.org/10.25046/aj050284>
- Sohan, M., Sai Ram, T., & Rami Reddy, Ch. V. (2024). *A Review on YOLOv8 and Its Advancements* (pp. 529–545). https://doi.org/10.1007/978-981-99-7962-2_39
- Szeliski, R. (2022). *Computer Vision*. Springer International Publishing. <https://doi.org/10.1007/978-3-030-34372-9>
- Tamuly, S., Jyotsna, C., & Amudha, J. (2020). *Deep Learning Model for Image Classification* (pp. 312–320). https://doi.org/10.1007/978-3-030-37218-7_36
- Tian, Z., Shen, C., Chen, H., & He, T. (2019). *FCOS: Fully Convolutional One-Stage Object Detection*.
- Tzuralin. (2015). *LabelImg*. Git Code. <https://github.com/tzutalin/labelImg>
- Ultralytics YOLO Docs*. (2023, November 12). Ultralytics YOLO Docs. <https://docs.ultralytics.com/>
- Vijayakumar, A., & Vairavasundaram, S. (2024). YOLO-based Object Detection Models: A Review and its Applications. *Multimedia Tools and Applications*, 83(35), 83535–83574. <https://doi.org/10.1007/s11042-024-18872-y>
- Wu, Y., Chen, R., Li, Z., Ye, M., & Dai, M. (2024). SDD-YOLO: A Lightweight, High-Generalization Methodology for Real-Time Detection of Strip Surface Defects. *Metals*, 14(6), 650. <https://doi.org/10.3390/met14060650>
- Xu, J.-H., Li, J.-P., Zhou, Z.-R., Lv, Q., & Luo, J. (2024). A Survey of the Yolo Series of Object Detection Algorithms. *2024 21st International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, 1–6. <https://doi.org/10.1109/ICCWAMTIP64812.2024.10873779>
- Zhang, C., & Shen, J. (2024). YOLO-DS: Application of One-Stage Instance Segmentation in the Dark at Construction Sites. *2024 International Joint Conference on Neural Networks (IJCNN)*, 1–8. <https://doi.org/10.1109/IJCNN60899.2024.10650361>
- Zheng, J., Zheng, J., Zhang, S., Yu, H., Kong, L., & Zhigang, D. (2023a). Segmentation Method for Whole Vehicle Wood Detection Based on Improved YOLACT Instance

Segmentation Model. *IEEE Access*, 11, 81434–81448.
<https://doi.org/10.1109/ACCESS.2023.3300900>

Zheng, J., Zheng, J., Zhang, S., Yu, H., Kong, L., & Zhigang, D. (2023b). Segmentation Method for Whole Vehicle Wood Detection Based on Improved YOLACT Instance Segmentation Model. *IEEE Access*, 11, 81434–81448.
<https://doi.org/10.1109/ACCESS.2023.3300900>