

Trabajo de grado

**APLICATIVO MÓVIL PARA EL RECONOCIMIENTO DE
MEDICAMENTOS USANDO VISIÓN ARTIFICIAL PARA APOYAR EL
PROCESO DE INVENTARIOS**

Edwar Stiven Montaña Cely



UNIVERSIDAD DEL VALLE
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA
PROGRAMA ACADÉMICO DE INGENIERÍA ELECTRÓNICA
SANTIAGO DE CALI
2025

Trabajo de grado

**APLICATIVO MÓVIL PARA EL RECONOCIMIENTO DE
MEDICAMENTOS USANDO VISIÓN ARTIFICIAL PARA APOYAR EL
PROCESO DE INVENTARIOS**

Edwar Stiven Montaña Cely
201622505

Documento presentado al
Programa de Pregrado de Ingeniería Electrónica
como requisito para optar por el título de
Ingeniero electrónico

Director: Ing. Eval Bladimir Bacca Cortés, Ph.D.
Ing. Humberto Loaiza Correa, Ph.D.

UNIVERSIDAD DEL VALLE
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA
PROGRAMA ACADÉMICO DE INGENIERÍA ELECTRÓNICA
SANTIAGO DE CALI
2025

AGRADECIMIENTOS

En primer lugar, quiero expresar mi más profundo agradecimiento a Dios, quien me ha acompañado incondicionalmente en este camino académico. Su guía, fortaleza y amor han sido un faro de luz en los momentos más desafiantes y una fuente de esperanza que me permitió alcanzar esta importante meta en mi vida.

A mis padres, les debo todo lo que soy. Su apoyo incondicional, su amor infinito y su paciencia inquebrantable han sido los cimientos de mi formación, no solo como profesional, sino como ser humano. Gracias por enseñarme con su ejemplo el valor del esfuerzo, la perseverancia y los sueños. Sin ustedes, este logro simplemente no habría sido posible.

También quiero expresar mi gratitud a mis amigos, compañeros y profesores. Cada palabra de aliento, cada momento de camaradería y cada consejo brindado fueron piezas fundamentales para superar los retos de este trayecto. En especial, agradezco a mis directores de tesis, cuya dedicación, compromiso y valiosas contribuciones marcaron una diferencia trascendental en el desarrollo de este proyecto y en mi crecimiento como profesional.

A la Universidad del Valle, mi alma máter, le agradezco por ofrecerme un entorno lleno de oportunidades, recursos y desafíos que me impulsaron a crecer y dar lo mejor de mí. Su compromiso con la excelencia académica ha dejado una marca indeleble en mi vida, una que siempre llevaré con orgullo.

A todos ustedes —Dios, mi familia, mis amigos, mis profesores y mi universidad—, les ofrezco mi gratitud más profunda y sincera. Cada uno de ustedes ha dejado una huella imborrable en mi corazón y en mi camino, convirtiéndose en parte esencial de este logro. Este triunfo no es solo mío, es nuestro, fruto del esfuerzo conjunto de quienes caminaron conmigo en este trayecto lleno de aprendizajes, retos y sueños cumplidos.

FICHA RESUMEN

Título:	APLICATIVO MÓVIL PARA EL RECONOCIMIENTO DE MEDICAMENTOS USANDO VISIÓN ARTIFICIAL PARA APOYAR EL PROCESO DE INVENTARIOS
Área temática o línea de investigación del Trabajo de grado:	Visión artificial, Reconocimiento de patrones, Máquinas de aprendizaje y clasificación de parámetros.
Modalidad de trabajo de grado:	<i>Profesional</i> ☐ <i>Revisión crítica</i> ☐ <i>Investigación e innovación</i> ☒ <i>En la industria/empresa</i> ☐ <i>Creación de empresa</i> ☐ <i>Práctico social</i> ☐
Duración del trabajo de grado:	2 períodos académicos
Entidades participantes:	Universidad del Valle
Áreas académicas EIEE:	☐ Arquitecturas Digitales ☐ Bionanoelectrónica ☐ Conversión de energía ☒ Informática Industrial ☐ Sistemas de Control y Accionamientos ☐ Sistemas de Potencia ☐ Telecomunicaciones
Grupos de investigación:	☐ Alta tensión (GRALTA) ☐ Arquitecturas Digitales y Microelectrónica (GADYM) ☐ Bionanoelectrónica ☐ Control Industrial (GICI) ☐ Conversión de Energía (CONVERGIA) ☒ Percepción y Sistemas Inteligentes (PSI) ☐ Sistemas de Telecomunicaciones (SISTEL-UV) ☐ Sistemas Hidroeléctricos de Generación (SHG) ☐ Otro:
Estudiante	
Código: 201622505	Nombre: Edwar Stiven Montaña Cely
Tel: 316 4934384	Correo: edwar.montano@correounivalle.edu.co
Directores	
Nombre: Bladimir Bacca Cortes, Ph.D.	
Tel: 6023391780 Ext. 116	Correo: bladimir.bacca@correounivalle.edu.co
Nombre: Humberto Loaiza Correa, Ph.D.	
Tel: 6023391780 Ext. 116	Correo: humberto.loaiza@correounivalle.edu.co

Glosario

AI Artificial intelligence. 9, 10, 18

CNN Convolutional Neural Networks. 11, 20, 25

DL Deep Learning. 9–11, 16, 18, 19, 25

IoU Intersection Over Union. 25

ML Machine Learning. 9, 18, 19

NN Neural Network. 18

RNA Redes Neuronales Artificiales. 18, 19

SSP Spatial pyramid pooling. 25

VA Visión Artificial. 21–23

YOLO You Only Look Once. 10–16, 25

TABLA DE CONTENIDO

	Pag.
GLOSARIO	III
TABLA DE CONTENIDO	IV
TABLA DE FIGURAS.....	VI
TABLAS DE TABLAS.....	IX
1. INTRODUCCIÓN DEL TRABAJO	1
1.1 Introducción.....	1
1.2 Planteamiento del Problema	2
1.3 Objetivos	4
1.4 Solución Propuesta	4
1.5 Estructura del Documento	7
2. ANTECEDENTES Y MARCO TEÓRICO.....	9
2.1 Introducción.....	9
2.2 Antecedentes	9
2.2.1 Antecedentes Nacionales	11
2.2.2 Antecedentes internacionales	12
2.2.3 Antecedentes comerciales	15
2.2.4 Discusión	15
2.3 Marco Teórico.....	17
2.3.1 Medicamentos.....	17
2.3.2 Inteligencia Artificial	18
2.3.3 Aprendizaje Automático	18
2.3.4 Visión Artificial	21
2.3.4.1 Adquisición:	23
2.3.4.2 Preprocesado:.....	23
2.3.4.3 Segmentación:	24
2.3.4.4 Extracción de características:	24
2.3.4.5 Clasificación:	25
2.3.4.5.1 YOLO.....	25
2.3.4.6 Métricas de desempeño	26
2.4 Discusiones	27

3.	APLICATIVO MÓVIL PARA RECONOCIMIENTO DE MEDICAMENTOS.	29
3.1	Introducción.....	29
3.2	Reconocimiento de medicamentos	29
3.2.1	Descripción de la base de datos	29
3.2.1.1	Adquisición controlada.....	31
3.2.1.2	Adquisición no controlada	32
3.2.2	Etiquetado de imágenes	34
3.2.3	Separación del conjunto de datos de entrenamiento, validación y prueba	38
3.2.3.1	Pre-procesamiento de imágenes	38
3.2.3.2	Aumento de datos.....	39
3.2.4	Implementación del modelo YOLO.....	40
3.2.4.1	Espacio experimental uno: entrenamiento de los modelos YOLOv8	41
3.2.4.2	Espacio experimental dos: optimización del mejor modelo YOLO	43
3.3	Desarrollo del aplicativo móvil	45
3.3.1	Requerimientos funcionales y no funcionales	45
3.3.2	Diagrama conceptual	47
3.3.3	Historias de usuarios	48
3.3.4	Implementación	51
3.3.4.1	Exportación del modelo	51
3.3.4.2	Desarrollo del aplicativo	52
4.	PRUEBAS Y RESULTADOS	56
4.1	Introducción.....	56
4.1.1	Espacio experimental uno y dos: evaluación de los modelos YOLOv8 y optimización del mejor modelo	56
4.1.2	Exportación modelo YOLOv8n al ambiente móvil.....	59
4.1.3	Pruebas de Campo	61
4.2	Discusión	63
5.	CONCLUSIONES	65
5.1	Limitaciones.....	66
6.	TRABAJO FUTURO	67
7.	ANEXOS	68
	REFERENCIAS	69

ÍNDICE DE FIGURAS

FIGURA		Pag.
1.1	Aplicación móvil de visión artificial revolucionando la gestión de inventario de medicamentos. Fuente: Elaboración propia	3
1.2	Flujo MLOps recomendado por Ultralytics y adaptado al proyecto. Fuente: elaboración propia adaptado de ultralytics	5
1.3	Flujo de trabajo que ejecuta el aplicativo. Fuente: elaboración propia	7
2.1	Tendencia de la consulta de AI en Google desde 2019 al 2024. Fuente: Elaboración propia.....	10
2.2	Algunos tipos de empaque de medicamentos y etiquetas. Fuente: elaboración propia	17
2.3	Topología de una RNA. Fuente: Ponce Cruz.....	19
2.4	Diagrama de Venn conjunto de AI, ML Y DL [Deep Learning Ian goodfellow].....	20
2.5	Arquitectura de una CNN. Fuente: Megha Shroff 2023	21
2.6	Proceso en un sistema de VA. Fuente: Adaptación propia	22
2.7	Representación del calculo de la IoU. Fuente: Elaboración propia	26
3.1	Imágenes de las 10 cajas de medicamentos seleccionados para el estudio. Fuente: elaboración propia	30
3.2	Muestra conjunto de datos adquisición controlada. Fuente: elaboración propia.....	32
3.3	Muestra conjunto de datos adquisición no controlada. Fuente: elaboración propia	33
3.4	Muestra dataset etiquetado con las 10 clases del estudio. Fuente: elaboración propia	34

3.5	cantidad de etiquetas por cada clase. Fuente: Gráfico resultante etiquetado Roboflow	35
3.6	Distribución porcentual por clase según etiquetas de imágenes en adquisición controlada y no controlada. Fuente: Gráfico resultante etiquetado Roboflow	36
3.7	Gráfico de número de etiquetas vs. cantidad de imágenes. Fuente: Gráfico resultante etiquetado Roboflow	37
3.8	Gráfico mapa de calor, ocurrencias de la ubicación del objeto dentro de las estanterías. Fuente: Gráfico resultante etiquetado Roboflow	37
3.9	Distribución media del tamaño de las imágenes. Fuente: Gráfico resultante etiquetado Roboflow	39
3.10	Modelos pre-entrenados en YOLOv5. Fuente: repositorio Ultralytics	41
3.11	captura de pantalla código de programación. Fuente: elaboración propia .	44
3.12	captura de pantalla código de programación. Fuente: elaboración propia .	45
3.13	Diagrama conceptual. Fuente: elaboración propia	48
3.14	Captura de pantalla de las vistas de la aplicación móvil en teléfono Android. Fuente: elaboración propia según diseño de App	54
3.15	Captura de pantalla de las vistas de la aplicación móvil en teléfono Android. Fuente: elaboración propia según diseño de App	55
4.1	Precisión de los modelos YOLO sobre un conjunto de datos nunca antes visto. Fuente: Gráfico resultante de la evaluación del modelo en CometML	57
4.2	Sensibilidad de los modelos YOLO sobre un conjunto de datos nunca antes visto. Fuente: Gráfico resultante de la evaluación del modelo en CometML	57
4.3	mAP50 de los modelos YOLO sobre un conjunto de datos nunca antes visto. Fuente: Gráfico resultante de la evaluación del modelo en CometML	58
4.4	mAP50-95 de los modelos YOLO sobre un conjunto de datos nunca antes visto. Fuente: Gráfico resultante de la evaluación del modelo en CometML	58
4.5	Gráficas de validación y test del entrenamiento del mejor modelo	59

4.6	Tiempos de inferencia de los modelos YOLO. Fuente: Gráfico resultante de la evaluación del modelo en CometML	61
4.7	Matriz de confusión multi-label.....	62
7.1	Anexos. Fuente: Elaboración propia.....	68

ÍNDICE DE CUADROS

TABLA	Pag.
2.1 Comparación resúmenes de antecedentes	11
3.1 Gasto farmacéutico según grupo terapéutico.	30
3.2 Código de colores para las etiquetas del dataset de imágenes.	35
3.3 Métricas de CPU y GPU con valores convertidos a GiB y porcentajes expresados claramente.	42
3.4 Búsqueda exhaustiva del tamaño de lote óptimo	43
3.5 Métricas de CPU y GPU con valores convertidos a GiB y porcentajes expresados claramente.	43
3.6 Requerimientos funcionales del aplicativo.....	46
3.7 Requerimientos no funcionales del aplicativo	47
3.8 Historias de usuario y criterios de aceptación	49
4.1 Evaluación GPU	59
4.2 Evaluación CPU	60

RESUMEN

El desarrollo de una aplicación móvil para el reconocimiento de medicamentos surge como una respuesta a la necesidad de mejorar la gestión de inventarios en farmacias aumentando la precisión en la distribución y almacenamiento de medicamentos. El proyecto se enfoca inicialmente en diez medicamentos seleccionados por su alto volumen de ventas y relevancia en el tratamiento de condiciones comunes en Colombia. Para implementar esta solución, se utilizó un modelo YOLO versión 8, adaptado para ejecutarse en TensorFlow lite, lo cual permite su uso en una amplia gama de dispositivos móviles. La recopilación y preparación de datos involucraron procedimientos meticulosos en entornos controlados y comerciales para garantizar la robustez del sistema ante variaciones reales de iluminación y empaque.

El sistema de clasificación emplea la arquitectura YOLOv8, destacándose por su precisión y velocidad en la detección y clasificación de objetos, lo cual es esencial para la funcionalidad de la aplicación en tiempo real. Además, se integraron herramientas como PyTorch, Ultralytics, Roboflow y Comet para facilitar el flujo de desarrollo y la gestión de modelos. El entrenamiento del modelo se realizó ajustando cuidadosamente los hiperparámetros para equilibrar precisión y eficiencia computacional, y se exportó el modelo a TensorFlow Lite para asegurar su compatibilidad y rendimiento en dispositivos móviles.

El diseño de la aplicación móvil se centró en integrar todas las etapas del flujo MLOps, desde la adquisición de datos hasta el despliegue y monitoreo de modelos. En pruebas, el modelo alcanzó una precisión del 94% y una sensibilidad de 97.2% mostrando capacidad robusta para identificar medicamentos. La evaluación de desempeño en plataformas de GPU y CPU arrojó resultados de velocidades de inferencia de 8 y 71 milisegundos por imagen respectivamente. Aunque en CPU la velocidad de inferencia se redujo, la precisión y sensibilidad se mantuvieron, lo que sugiere que el modelo es adaptable y funcional en dispositivos con recursos computacionales limitados, lo que amplía considerablemente su aplicabilidad.

Palabras clave— Visión por Computadora, Inteligencia Artificial, Reconocimiento de objetos, Y.O.L.O, Medicamentos

ABSTRACT

The development of a mobile application for medication recognition arises as a response to the need to improve inventory management in pharmacies by increasing accuracy in the distribution and storage of medications. The project initially focuses on ten medications selected for their high sales volume and relevance in treating common conditions in Colombia. To implement this solution, a YOLO version 8 model was used, adapted to run on TensorFlow Lite, enabling its use on a wide range of mobile devices. Data collection and preparation involved meticulous procedures in controlled and commercial environments to ensure the system’s robustness against real-world variations in lighting and packaging.

The classification system employs the YOLOv8 architecture, distinguished by its accuracy and speed in object detection and classification, which is essential for real-time application functionality. Additionally, tools such as PyTorch, Ultralytics, Roboflow, and Comet were integrated to facilitate the development workflow and model management. The model was trained by carefully tuning hyperparameters to balance accuracy and computational efficiency and was subsequently exported to TensorFlow Lite to ensure compatibility and performance on mobile devices.

The design of the mobile application focused on integrating all stages of the MLOps pipeline, from data acquisition to model deployment and monitoring. In testing, the model achieved an accuracy of 94% and a sensitivity of 97.2%, demonstrating robust capability in identifying medications. Performance evaluation on GPU and CPU platforms yielded inference speeds of 8 and 71 milliseconds per image, respectively. Although inference speed decreased on the CPU, accuracy and sensitivity remained consistent, suggesting that the model is adaptable and functional on devices with limited computational resources, significantly expanding its applicability.

Keywords— Computer Vision, Artificial Intelligence, Object Recognition, YOLO, Medications.

1. INTRODUCCIÓN DEL TRABAJO

1.1. Introducción

En los últimos años, se han desarrollado investigaciones sobre la aplicación de la visión artificial en el comercio minorista, especialmente en el sector privado. Sin embargo, estas soluciones aún no se han implementado en sectores como las farmacias de las EPS colombianas, donde la gestión ineficiente de inventarios genera problemáticas de orden social que afectan la salud de las personas [1]. Considerando esta necesidad, el presente proyecto desarrolló un aplicativo móvil que, mediante técnicas de visión artificial y aprendizaje automático, permite identificar hasta diez (10) medicamentos de alta demanda en Colombia utilizando imágenes captadas con la cámara del dispositivo móvil. El aplicativo proporciona el nombre y la cantidad de los medicamentos según su formato de presentación—en este caso, cajas—lo que agiliza la gestión de inventarios y la detección de productos agotados.

Para responder a este desafío, se implementó un sistema de reconocimiento que emplea la arquitectura YOLOv8 de detección de objetos, entrenada con un dataset propio de imágenes de medicamentos, el cual incluye variaciones de iluminación, orientación y condiciones de captura. Dicho modelo fue optimizado y exportado a TensorFlow Lite con el propósito de garantizar su ejecución eficiente en dispositivos móviles, manteniendo un balance entre precisión y velocidad de inferencia.

El desarrollo metodológico incluyó cuatro fases principales: (i) la identificación de características relevantes en los empaques que facilitan su diferenciación visual, (ii) la construcción de un dataset de imágenes originales, (iii) la implementación y entrenamiento de la arquitectura YOLOv8 en sus variantes nano y medium, (iv) el diseño y desarrollo del aplicativo móvil que integra captura de imágenes, procesamiento e inferencia y (v) la definición de métricas y métodos para evaluar el rendimiento del sistema en términos de exactitud, sensibilidad y eficiencia computacional.

Los resultados alcanzados evidencian que el modelo YOLOv8n, optimizado para su implementación en dispositivos móviles, logró una precisión del 94 % y una sensibilidad del 97.2 %, con tiempos de inferencia de 7 ms por imagen en GPU. El aplicativo desarrollado no solo demuestra la viabilidad de integrar modelos de aprendizaje profundo en entornos móviles, sino que también representa una contribución práctica al campo de la visión artificial aplicada al sector salud, al ofrecer una herramienta tecnológica que apoya en las tareas de inventarios de medicamentos.

1.2. Planteamiento del Problema

Los propietarios de farmacias se enfrentan al desafío constante de gestionar su inventario. Según el Grupo IHL¹, los comerciantes de todo el mundo pierden alrededor de \$1,1 billones de dólares debido a la falta de disponibilidad de mercancía o al exceso de existencias [2]. Para las farmacias, esta situación adquiere mayor importancia debido a la naturaleza sensible de los medicamentos y a las responsabilidades sociales y económicas asociadas a su comercialización [3]. Las farmacias tienen un stock en constante variación, con entradas y salidas de productos en todo momento, esta dinámica aumenta las inexactitudes en los sistemas de inventario dada la alta incidencia del factor humano. Por lo anterior, se requiere que periódicamente las farmacias deban someterse a un proceso de ajuste donde se comparan las cantidades de cada medicamento registradas en el sistema con el valor numérico real en existencia [4]. Estas actividades de inventario suelen resultar agotadoras y consumen tiempos laborales aprovechables en tareas de mayor importancia.

Similar al inventario, la recepción de medicamentos también requiere la revisión de los productos y su documentación para ingresar la mercancía al almacén (para el caso de estudio, la farmacia) y de paso ingresar la información del producto en la base de datos del sistema de gestión de la empresa [5]. La información del producto se verifica en los rótulos y etiquetas de los envases y empaques de los medicamentos. Algunos ítems que el personal debe inspeccionar al recibo de cada medicamento son: nombre, concentración del ingrediente activo, cantidad, número de lote de fabricación, fecha de fabricación, fecha de vencimiento, y nombre del fabricante. En general, no todos los datos son valiosos por la gran cantidad de tiempo que requiere la inspección y en especial el registro manual de la información de la mercancía al sistema.

Este es un proceso repetitivo que conlleva una alta incidencia del factor humano y por ende genera una alta posibilidad de cometer errores, modelos de fallas probados por autores como Romero-Zayas [6] muestran como en una radio farmacia cerca de 96 ítems pueden proceder en fallas críticas en la administración y dispensación de medicamentos para pacientes. Desde la ingeniería de procesos, se han propuesto soluciones de tipo metodológicas para abordar esta situación [7], sin embargo, la era digital trajo consigo herramientas capaces de proponer soluciones efectivas y de fácil acceso.

Actualmente es posible extraer información del mundo físico utilizando dispositivos móviles con cámaras que captan imágenes para ser procesadas por medio de un computador. Esto se conoce como Visión Artificial. La detección de objetos es una tarea crucial en los sistemas de visión por computadora con una amplia gama de aplicaciones en conducción autónoma, imágenes médicas, comercio minorista,

¹<https://www.ihlservices.com>

seguridad, reconocimiento facial, robótica y otros [8] además es un problema de investigación atractivo y vigente [9]. En este estudio se utilizó esta tecnología para la detección automática de 10 productos exhibidos en estantes de farmacias.

Su implementación busca mejorar los tiempos y costos operativos, permitiendo una administración más eficiente del stock y garantizando la disponibilidad oportuna de medicamentos para la población. Una gestión eficiente del inventario no solo optimiza costos de adquisición y venta, sino que también asegura el abastecimiento adecuado de productos médicos para satisfacer la demanda de los pacientes de manera oportuna [10]. En la figura 1.1 se observa un regente de farmacia que, mediante un teléfono celular, supervisa el inventario de sus estanterías.



Figura 1.1: Aplicación móvil de visión artificial revolucionando la gestión de inventario de medicamentos. Fuente: Elaboración propia

Sin embargo, estas propuestas tecnológicas no están exentas de errores. Durante este proceso pueden presentarse dificultades ligadas a diferentes factores como lo son las condiciones ambientales que se presentan en el momento de obtener la imagen (Por ejemplo el ruido producido por un alta presencia de neblina u humedad en una imagen), las características del equipo con el cual es tomada la imagen a procesar (por ejemplo la estabilidad u capacidad de enfoque es diferente según el tipo de cámara u estabilizador empleado para tomar las imágenes), el tamaño y resolución de imagen al ser capturada desde diferentes dispositivos debe coincidir previo a su procesamiento y clasificación [11], etc.

Con el ánimo de contribuir en el desarrollo de sistemas automáticos que ayuden a superar las limitaciones mencionadas, en este trabajo se planteó como pregunta problema:

¿Cómo desarrollar un aplicativo móvil para el reconocimiento de medicamentos usando visión artificial para apoyar el proceso de inventarios?

1.3. Objetivos

Se planteó como objetivo general:

“Reconocer automáticamente un total de hasta diez medicamentos en empaque tipo caja a partir del análisis de imágenes utilizando técnicas de visión artificial y reconocimiento de patrones implementadas en una herramienta software ejecutable en un dispositivo móvil para apoyar el proceso de inventarios”

Para dar cumplimiento al objetivo general, se propusieron los siguientes objetivos específicos:

1. Identificar las características del empaque que faciliten la diferenciación entre medicamentos por medio de técnicas de visión artificial y reconocimiento de patrones.
2. Construir un data set de imágenes digitales de medicamentos comerciales distribuidos en Colombia con variaciones en su orientación e iluminación.
3. Implementar una arquitectura de máquinas de aprendizaje para el reconocimiento automático de medicamentos.
4. Desarrollar un aplicativo Móvil para la adquisición de las imágenes de medicamentos a reconocer.
5. Determinar métricas y métodos para evaluar el rendimiento de los modelos locales y remotos de reconocimiento de medicamentos

1.4. Solución Propuesta

Esta propuesta buscó contribuir en el desarrollo de sistemas automáticos que faciliten los procesos de inventario en farmacias por medio de una aplicación móvil que, usando la cámara del dispositivo, capture una imagen de un conjunto de cajas de medicamento, para posteriormente identificar el nombre de los medicamentos y su cantidad. La solución propuesta se puede dividir en dos grandes fases. La primera fase consistió en el entrenamiento de la arquitectura de red neuronal convolucional YOLO versión 8 especializada en detección de objetos en imágenes y video. Y la segunda fase consistió en, ya como tal, desarrollar el aplicativo móvil.

Para la fase 1 se propuso una solución utilizando un flujo de trabajo de operaciones de aprendizaje automático (Machine Learning Operations - MLOps), que abarcó desde la recopilación y preparación de datos hasta el despliegue del modelo en el dispositivo móvil como se aprecia en la figura 1.2.

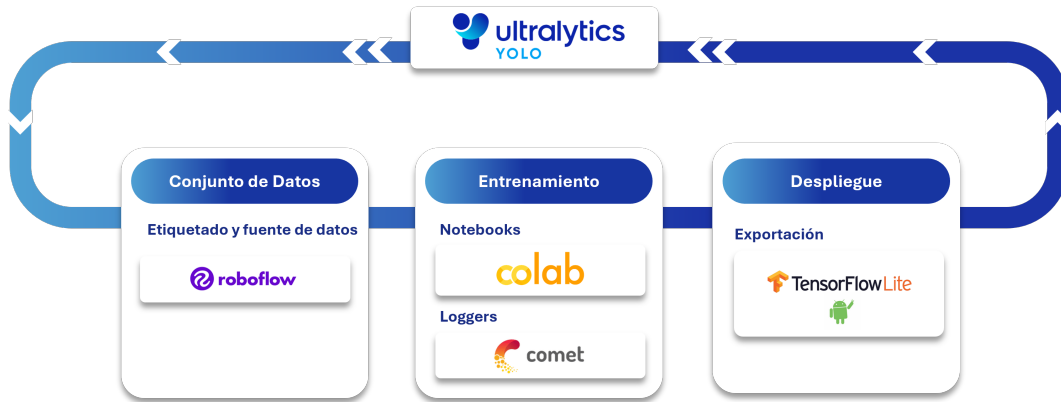


Figura 1.2: Flujo MLOps recomendado por Ultralytics y adaptado al proyecto. Fuente: elaboración propia adaptado de ultralytics

Este flujo inicia con la preparación de un conjunto de datos de imágenes con las cuales se debe entrenar el modelo, por lo tanto contiene las fotos de los medicamentos de interés en el estudio. Este conjunto de datos es manipulado en Roboflow que facilita el etiquetado de las imágenes donde se delimita la ubicación de cada caja de medicamento. Aquí mismo se generaliza el conjunto de datos por medio de un Data Augmentation con el fin de garantizar un entrenamiento óptimo del modelo. El conjunto de datos resultante se utilizó para entrenar dos tipos de modelos YOLOv8. Una vez finalizado el entrenamiento, se analizaron los pesos de la matriz obtenida y, con base en estos resultados, se ajustaron los hiperparámetros para realizar un nuevo entrenamiento en solo el modelo YOLO mas conveniente que resulto ser el YOLOv8 Nano.

La elección de YOLOv8, PyTorch, Ultralytics, Roboflow¹ y Comet se fundamenta en las ventajas complementarias que cada herramienta aporta; YOLOv8 asegura precisión y rapidez en la detección de objetos, PyTorch facilita tanto el entrenamiento como la implementación del modelo, las APIs de Ultralytics y Roboflow¹ optimizan el desarrollo y manejo de datos, y Comet permite una gestión integral de experimentos para maximizar el rendimiento del modelo. Esta combinación de herramientas garantiza consistencia en los datos, facilidad de uso, escalabilidad y capacidad de gestión, consolidándose como una solución idónea para aplicaciones prácticas en el mundo real.

Ya en la fase 2, se crea un proyecto desarrollado en Android Studio que integra la biblioteca de TensorFlow lite que permite el despliegue del modelo YOLO recientemente

entrenado en un entorno móvil para su ejecución en Android. Aquí se programan las tareas que debe cumplir el aplicativo móvil que se describen en la figura 1.3 las cuales inician con un usuario que por medio de su celular y habiendo ingresado en la aplicación de este proyecto, inicializa la cámara y apunta hacia el grupo de medicamentos que pretende reconocer. Al extraer la imagen, la aplicación procesa la información gracias a un modelo light YOLO específico para Android que hace inferencia de la imagen y reconoce el objeto de estudio. La salida de información será una etiqueta de texto con la información del nombre del medicamento y la cantidad.

El caso de estudio de este trabajo se aplicó al reconocimiento de diez (10) medicamentos comerciales contenidos en cajas existentes en el mercado colombiano. Esta aplicación capaz de detectar en una imagen el nombre de un medicamento y la cantidad de unidades visibles constituye una herramienta de apoyo para la gestión de inventarios en entornos de farmacia. Con esta herramienta el personal no necesita identificar manualmente cada referencia ni contabilizar una por una las unidades presentes. Con un solo registro visual, la aplicación ofrece de manera inmediata información puntual sobre el producto observado. Por otro lado, la precisión en la identificación de referencias es otro aspecto relevante. En muchos establecimientos se comercializan medicamentos con presentaciones similares o con nombres fácilmente confundibles. El reconocimiento automático del nombre impreso en el empaque contribuye a reducir estos errores, garantizando que la información obtenida corresponda exactamente al producto capturado en la imagen.

Asimismo, la aplicación funciona como un mecanismo de verificación en el proceso de inventario. El personal puede contrastar los datos generados por la aplicación con el conteo realizado de forma manual o con las listas de control existentes. De esta manera, se disminuye la probabilidad de inconsistencias y se fortalece la confiabilidad de los registros obtenidos. Otro aporte significativo es la posibilidad de realizar inventarios en lote. En lugar de analizar cada unidad de manera individual, la aplicación permite capturar imágenes de estantes completos o de grupos de productos organizados. Con ello, se obtiene en menor tiempo el nombre y la cantidad de los artículos visibles, lo que representa un aumento en la eficiencia operativa frente a los métodos tradicionales de conteo.

Si bien la aplicación presenta aún múltiples oportunidades de mejora para cubrir de manera integral las necesidades propias de los procesos de inventario, constituye una aproximación relevante hacia dicho objetivo. Su capacidad para reconocer el nombre de los productos y contabilizar las unidades visibles en una imagen no reemplaza por completo las tareas de registro y actualización de inventarios, pero sí ofrece un punto de partida valioso en la búsqueda de soluciones tecnológicas que optimicen la gestión de existencias en farmacias.

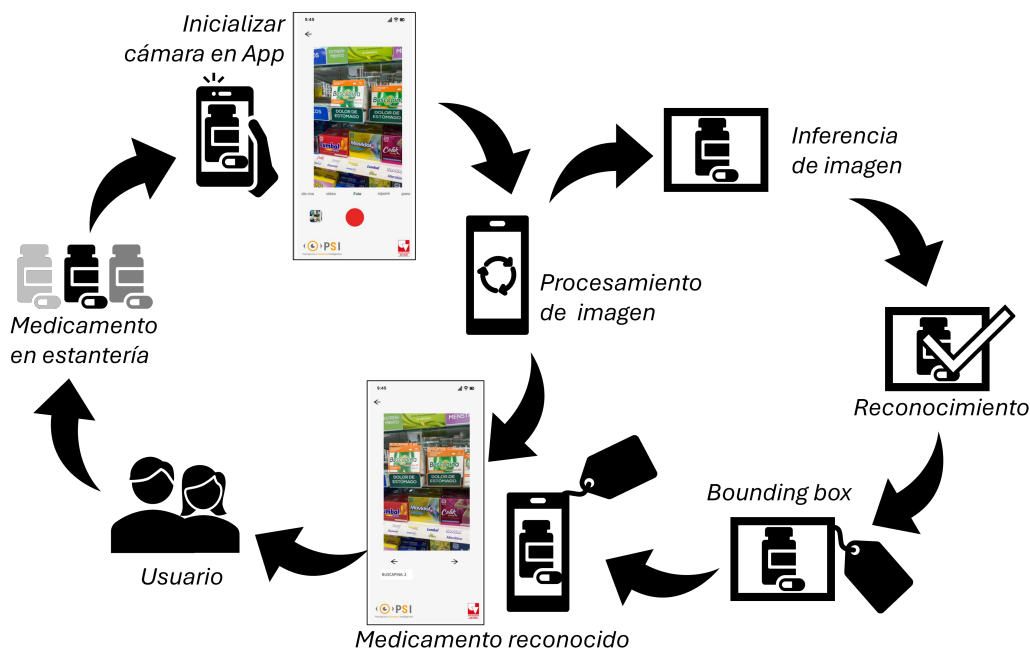


Figura 1.3: Flujo de trabajo que ejecuta el aplicativo. Fuente: elaboración propia

1.5. Estructura del Documento

Para una comprensión integral de esta investigación, el presente documento estructura la información en diversos capítulos numerados hasta el siete (7).

En el primer capítulo “INTRODUCCIÓN DEL TRABAJO” se presenta una contextualización y se destaca la relevancia del proyecto, sus respectivos objetivos, general y específicos y un diagrama con la solución planteada.

En el capítulo dos “ANTECEDENTES Y MARCO TEÓRICO”, primero se plantean los antecedentes que corresponde a resúmenes de trabajos previos que han servido de guía y referencia para este proyecto, después, se presenta una tabla comparativa en donde se resaltan los aspectos mas importantes a tener en cuenta. Finalmente, en el Marco Teórico se proporcionan definiciones y explicaciones esenciales para comprender los temas que se discuten a lo largo del documento, en especial para la comprensión de la metodología y los resultados de la investigación.

En el capítulo tres “APLICATIVO MÓVIL PARA RECONOCIMIENTO DE MEDICAMENTOS”, se detalla con precisión el proceso implementado para obtener la aplicación móvil, desde la selección de datos y preprocesamiento de imágenes, hasta el

entrenamiento de algoritmos de aprendizaje profundo. Además, se discutirán las consideraciones prácticas y los desafíos enfrentados durante este proceso.

En el capítulo cuatro “PRUEBAS Y RESULTADOS”, se muestra la validación de la funcionalidad del aplicativo por medio de diferentes pruebas que le permite al lector evaluar la satisfacción de los resultados y validar el aporte de la investigación.

En el quinto capítulo “CONCLUSIONES” se destaca la información más relevante de la investigación y el potencial que tiene esta tecnología en aplicaciones similares y

En el capítulo seis “TRABAJOS FUTUROS”, se discuten las oportunidades de mejoras que tiene el estudio ya sea para alcanzar mayores niveles de exactitud del modelo o para proponer actividades que amplíen el alcance de la investigación para que aumente la funcionalidad del aplicativo en un ambiente completamente real.

El capítulo siete corresponde a los “ANEXOS” donde se presentan los enlaces a los materiales complementarios en caso de que el lector quiera profundizar en algún punto del desarrollo del aplicativo.

Finalmente, en el capítulo siete “REFERENCIAS” se cita la bibliografía consultada a lo largo del proyecto, proporcionando a los lectores la posibilidad de ampliar y verificar los datos sustentados del estudio.

2. ANTECEDENTES Y MARCO TEÓRICO

2.1. Introducción

En esta sección se presentará el marco de referencia del estudio, que integra los antecedentes y fundamentos teóricos del sistema propuesto para reconocer automáticamente el nombre comercial de medicamentos en presentación tipo caja, a partir de imágenes capturadas por una aplicación móvil. La revisión bibliográfica incluyó estudios recientes sobre visión artificial, aprendizaje profundo y detección de objetos, destacando el uso de la arquitectura YOLOv8 por su eficiencia y precisión en tiempo real. Asimismo, se abordan conceptos clave como inteligencia artificial, redes neuronales, visión artificial, procesamiento de imágenes y aspectos normativos sobre el rotulado de medicamentos, necesarios para comprender el funcionamiento del sistema desarrollado.

2.2. Antecedentes

El presente capítulo sintetiza los hallazgos más relevantes de la revisión bibliográfica la cual incluyó estudios académicos nacionales e internacionales desarrollados entre los años 2015 y 2025 en torno a la aplicación de técnicas de visión artificial para la detección y clasificación de productos, especialmente medicamentos, a partir de imágenes capturadas mediante dispositivos móviles. Los estudios se seleccionaron a partir de una búsqueda sistemática en bases de datos científicas reconocidas como IEEE Xplore y Scopus. Para delimitar la búsqueda, se emplearon palabras clave como: *“reconocimiento de medicamentos”*, *“visión artificial”*, *“reconocimiento de imágenes”*, *“detección de objetos en el comercio minorista”*, *“aprendizaje profundo”* y *“aplicaciones de accesibilidad”*. También se incluyeron soluciones comerciales con las cuales se logró comprender tanto los avances como los desafíos actuales en el uso de redes neuronales para el reconocimiento visual en contextos reales.

Se destaca que, actualmente la Inteligencia Artificial (Artificial intelligence (AI)) ha encontrado una amplia variedad de aplicaciones en nuestra vida diaria gracias al desarrollo y la inversión realizada en los últimos años [12], (McKinsey¹). Campos como el Aprendizaje Automático (Machine Learning (ML)) y el Aprendizaje Profundo (Deep

¹<https://acortar.link/Cm9g3t>

Learning (DL)) han experimentado un notable impulso, permitiendo resolver tareas que anteriormente solo podían llevarse a cabo en supercomputadoras. Este avance ha sido posible gracias al mejoramiento del hardware destinado al procesamiento en paralelo [13].

Asimismo, se ha comenzado a reemplazar técnicas de procesamiento tradicionales, logrando realizar cálculos extremadamente complejos en cuestión de minutos o segundos, en comparación con el tiempo considerablemente mayor que requerían anteriormente. Según los datos recopilados a través de búsquedas en Google sobre AI, se ha observado un creciente interés mundial en estos temas en los últimos cinco años. La figura 2.1 presenta las tendencias de búsqueda relacionadas con el tema, reflejando este aumento de interés.

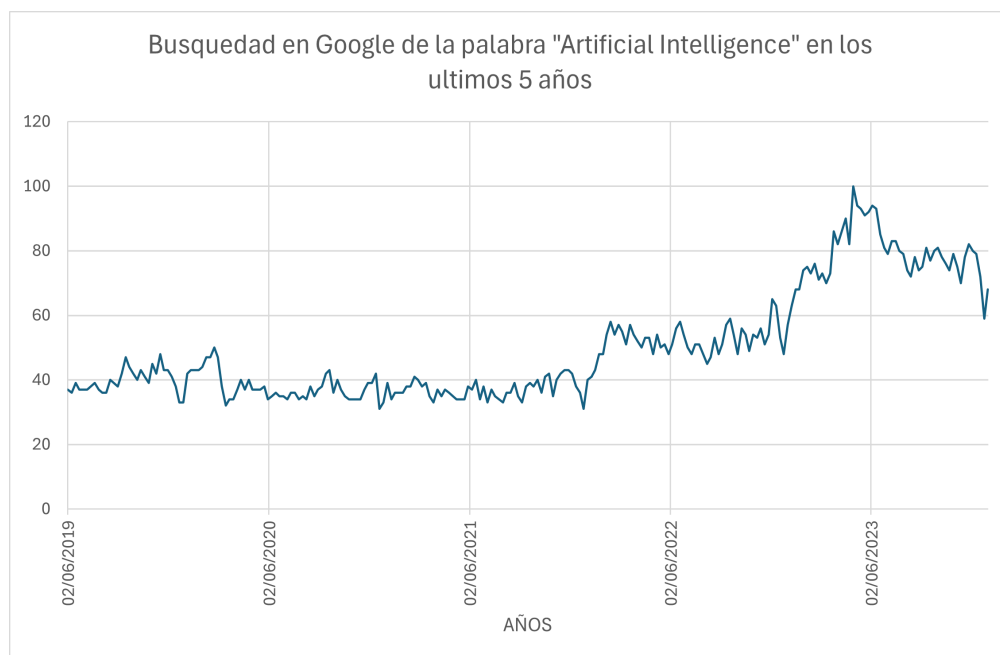


Figura 2.1: Tendencia de la consulta de AI en Google desde 2019 al 2024. Fuente: Elaboración propia

Finalmente, para una mayor facilidad hacia el lector, se presenta la tabla 2.1 la cual contiene un resumen de los antecedentes académicos presentados con las características más importantes que se tuvieron en cuenta para el desarrollo de esta investigación como tamaños de dataset, modelo utilizado, pruebas experimentales y resultados. Con base en esta revisión. Se identificó que la arquitectura YOLO, especialmente en su versión más reciente (YOLOv8), ha sido ampliamente utilizada por su eficiencia, precisión y capacidad de implementación en dispositivos móviles, lo cual motivó su selección como eje central para el desarrollo de la solución propuesta en esta investigación: un sistema

de reconocimiento de medicamentos mediante imágenes tomadas desde una aplicación móvil.

Tabla 2.1: Comparación resúmenes de antecedentes

AÑO	AUTORES	REF	OBJETIVO		TAMAÑO DATASET	PORCENTAJE ENTRENAMIENTO	PORCENTAJE VALIDACIÓN Y PRUEBA	MODELO	EVALUACION	
									METRICA	% ACCURACY
2020	I. A. Salazar Ortiz, et, al	[17]	Medidores de energía	Contador	1000	-	-	YOLO V4	mAP	87
				Numeros		-	-	YOLO V4	mAP	36
2021	M. Rivero-Palacio, et, al	[18]	Anemia en niños		671	80	20	YOLO V5	mAP PC	95
									mAP APP	79
2019	W.-J. Chang, et, al	[19]	Medicamento en pildoras		5000	-	-	Faster R-CNN/ Inception V3	Precisión	96,6
2019	C. G. Melek, et, al	[20]	Comercio minorista	Cajas de cigarrillos	354	90	10	YOLO V2	*Perdida 900ta interaccion	25,39
				Coca Cola	200	90	10	YOLO V2	*Perdida 2000va interaccion	2,63
2020	P. Wajire, et, al	[21]	Comercio minorista - Cajas de cigarrillo		1958	-	-	VGG16 / YOLO V2	mAP	95
2021	T. T. Mac, et, al	[22]	Medicamento en pildoras		2500	80	20	YOLO V3	*Perdida	≈0
2023	S. Li, et, al	[23]	Comercio minorista - Cajas de cigarrillo		1000	70	30	YOLO V5	mAP	95,1
2023	K. Kumar, et, al	[24]	Comercio minorista	botellas Sprite	200	80	20	YOLO V8	mAP50 mAP50-95	87
				Latas Nescafe						54

2.2.1. Antecedentes Nacionales

A nivel nacional, la literatura sobre la implementación de técnicas de visión artificial en el reconocimiento de medicamentos es muy escasa. A pesar de que se ha visto un auge en tecnologías como el Deep Learning (DL) en diferentes aplicaciones, este campo no cuenta aún con el desarrollo que se evidencia en otros países. Sin embargo, sí hay algunas investigaciones relacionadas con el reconocimiento de imágenes que resultaron importantes para esta investigación en especial para la reproducción metodológica.

En [14] del 2020, se desarrolló una aplicación móvil con capacidad de lectura de dispositivos de consumo de energía por medio de técnicas de deep learnig, específicamente redes neuronales convolucionales (*Convolutional Neural Networks (CNN)*) con la arquitectura de You Only Look Once (YOLO)-V4 . Se utilizó un dataset de imágenes de contadores (compartido en forma abierta por Vannetti y Laroca) que fue sometido a un etiquetado y posterior Data Augmentation. Con estos datos listos, se entrenaron los primeros modelos utilizando herramientas como Google Colab ², Jupyter y el framework de Darknet. Posteriormente, se convirtió el formato de los archivos entrenados de Darknet.weights a TensorFlow, haciéndolos aptos para

²<https://colab.research.google.com/>

plataformas móviles. Luego se transforman los archivos TensorFlow a TFLite para desplegar los modelos en teléfonos móvil. El modelo se implementó en una aplicación para pruebas en contextos reales, combinando ejemplos de detección de objetos de TensorFlow y TensorFlow-yolov4-tensorflowlite. Los resultados mostraron una buena detección del contador, sin embargo, la clasificación de números sin segmentación no alcanzó los resultados esperados, con un aumento del mAP del 20 % al 35.5 % incluso después de diferentes entrenamientos y técnicas de aumento de imágenes. probablemente debido a que para este tipo de modelos que no utilizan segmentación, para realizar la clasificación de los números necesitan muchos más datos de entrenamiento. La anterior investigación se desarrollo dentro del campo de aplicación para la Lectura automática de dispositivos de consumo de energía (contadores eléctricos) usando dispositivos móviles.

En [15] del 2021, los autores desarrollaron una aplicación móvil que con imágenes del interior del párpado permitió diagnosticar anemia en niños, esto gracias a la técnica de deep learning YOLO. Para ello se depuró un dataset de 1012 imágenes, quedando 457 imágenes validas (121 Anemia y 336 Sin Anemia). Por desbalanceo en el dataset, se aplicó un Data Augmentation a la clase Anemia utilizando la plataforma Robloflow³. Las imágenes fueron etiquetadas en su respectiva clasificación con la herramienta CVAT en formato *json*. El dataset se dividió en tres conjuntos: para el entrenamiento (80 %), validación(10 %) y prueba del modelo (10 %). Al final se obtuvieron 539 imágenes en entrenamiento y 66 imágenes para cada conjunto de validación y prueba. Posteriormente se entrenó la arquitectura YOLO con el dataset. Los resultados se evaluaron por medio de métricas derivadas de la matriz de confusión como la exactitud, sensibilidad y especificidad. Aprovechando el potencial de *transfer learning* se utilizo un modelo pre-entrenado para acelerar el proceso de entrenamiento. Finalmente se diseñó la aplicación móvil con *flutter* y *android studio* y se incorporó el modelo YOLO a la App con *TorchScript*. Las pruebas sobre la aplicación en el teléfono SAMSUNG A20 arrojaron una sensibilidad de 0,71 y una especificidad de 0,89. Aunque no se superan los resultados de las pruebas convencionales, es una gran paso en la detección de anemia. Esta investigación se desarrolló dentro del campo de aplicación para el diagnóstico preliminar de anemia en niños a partir de imágenes del párpado tomadas con una cámara móvil; aplicación en el sector salud/pediatría.

2.2.2. Antecedentes internacionales

En el ámbito internacional, el estado del arte es más amplio que el nacional. Algunas publicaciones utilizan técnicas tradicionales de reconocimiento de objetos mientras que otras emplean técnicas emergentes de inteligencia artificial, basándose en el uso de

³<https://roboflow.com/>

Convolutional Neural Network (CNN) para la detección de objetos.

En [16] del 2020 se propone un enfoque híbrido para la clasificación de imágenes con detección de objetos para verificar automáticamente el cumplimiento del planograma de un comercio minorista. Se busca clasificar una imagen como ‘Exacta 7 por 4’ o ‘No Exacta 7 por 4’. Si hay 7 artículos en cada una de las 4 filas de una mostrador se cumple con la política marcada por la empresa y se denomina ‘Exacto 7 por 4’. Si hay más o menos de 7 artículos en una o varias de las 4 filas se denomina clase ‘No exacta 7 por 4’. Para el entrenamiento utilizaron un dataset con 2000 imágenes, 1000 imágenes para cada clasificación. En primera instancia evaluaron la calidad de las imágenes con la herramienta BRISQUET obteniendo que de las 2000 imágenes 1958 resultaron de buena calidad. Después de la limpieza se clasificaron las imágenes con la red neuronal VGG16. para la detección de objetos utilizaron el modelo YOLO-V2. La tasa de clasificación fue del 70 % utilizando solo la red neuronal VGG16. Cuando la imagen se proceso con el modelo híbrido (se clasifica con VGG16 y se detecta con YOLO), la tasa de clasificación mejoró al 95 %. Esta investigación se desarrolló dentro del campo de aplicación para la verificación automática del cumplimiento de planogramas en comercios minoristas mediante clasificación de imágenes y detección de objetos; aplicación en retail y gestión de inventarios.

En [17] del 2021, se desarrolló un método efectivo y automatizado para la detección de pastillas en el proceso de manufacturación empleando diferentes técnicas que reducen la carga de cálculo por medio de Filtrado de Gauss, suavizado y extracción de características de Hog, en búsqueda de mejorar el modelo YOLO para la detección en línea de defectos de pastillas. Esta investigación se desarrolló dentro del campo de aplicación para la automatización en tareas de calidad dentro de la manufactura farmacéutica.

En [18] del 2023, los autores entrenan la YOLOv5 para detectar y etiquetar la ubicación de cajas de cigarrillos en una imagen y luego utilizan el algoritmo ArcFace para extraer la característica de estas cajas. El modelo logra una precisión promedio del 95,1 % en el conjunto de prueba, y la distribución mAP indica un buen rendimiento en la mayoría de las muestras. Se abordó la metodología en tres pasos: detección de cajas de cigarrillos, extracción de características y reconocimiento de las cajas. En el reconocimiento de cajas de cigarrillos, se utiliza el modelo YOLOv5 para entrenar los conjuntos de datos, lo que implica optimizar los parámetros del modelo para detectar y etiquetar con precisión las cajas en las imágenes. Luego, el modelo entrenado se utiliza para detectar nuevas imágenes y generar la ubicación de las cajas de cigarrillos. En la extracción de características, se emplea el algoritmo ArcFace para extraer las características de la imagen. Este algoritmo asigna la imagen de la caja a un vector de características que puede usarse para el reconocimiento. Finalmente, se compara el vector de características de la caja de cigarrillos detectada con una base de datos de cajas de cigarrillos conocidas. Si hay una coincidencia, se reconoce la caja de cigarrillos.

Esta investigación se desarrolló dentro del campo de aplicación del comercio minorista, con uso potencial en control de inventarios o vigilancia de productos regulados.

En [19] del 2023, se presenta una aplicación práctica de técnicas de detección de objetos en imágenes de vanguardia en el contexto de la gestión de estanterías minoristas. Para ello se pretendió clasificar dos objetos ‘Nescafé’ y ‘Sprite’ y cuantificar sus cantidades utilizando el algoritmo YOLO v8. Los resultados mostraron un mean Average Precision en el 50 % de la intersección sobre la unión (mAP50) de 0,871 en todas las clases de objetos, mientras que el mAP50-95 se situó en 0,541, lo que indica capacidades sólidas de detección de objetos. La metodología implementada consistió en recopilar un extenso conjunto de datos que incluyeron imágenes de productos de ‘Nescafé’ y ‘Sprite’ en estantes de tiendas (200 imágenes). Posteriormente se limpió, transformó y aumentó este dataset. Para el entrenamiento se empleó el modelo YOLOv8 que utilizó el conjunto de datos preparado. Para mejorar su rendimiento se aplicaron técnicas de aprendizaje por transferencia. La eficacia del modelo se evaluó con métricas que incluyeron Intersection over Union (IoU), precisión, recall y F1-score. La capacidad del modelo para clasificar con precisión los objetos ‘Nescafé’ y ‘Sprite’ y contar con precisión sus instancias se evaluaron con la comparación de los registros reales en el sitio. Esta investigación se desarrolló dentro del campo de aplicación para la gestión de estanterías y conteo automático de productos específicos (Nescafé y Sprite) en tiendas minoristas; sector retail, con uso potencial en control de inventarios o vigilancia de productos regulados.

En [20] del 2023 los autores proponen un sistema de reconocimiento de cajas de medicamentos basado en aprendizaje profundo y colaboración nube-borde para la automatización de tareas en farmacias. Para ello, mejoraron el modelo YOLOv5 mediante la integración de GhostNetv1 y GhostNetv2, lo que redujo en un 45 % las GFLOPs y mejoró la precisión en 1.6 %. El sistema opera en dos niveles: en el borde, un modelo ligero de YOLOv5 identifica las cajas de medicamentos en tiempo real, mientras que en la nube, un módulo de reconocimiento de texto basado en OCR (DBNet+DenseNet+CTC) verifica la información extraída. Los resultados experimentales muestran que la arquitectura propuesta reduce la latencia de reconocimiento en un 37.18 % y alcanza una precisión del 97.47 %, lo que la hace una solución viable para mejorar la eficiencia en la gestión de inventarios farmacéuticos. Esta investigación se desarrolló dentro del campo de aplicación del reconocimiento de cajas de medicamentos, apoyando tareas de gestión farmacéutica e inventarios.

En [21] del 2023, los autores desarrollan una aplicación basada en visión por computadora para la detección, conteo y verificación del estado de productos embotellados y enlatados en supermercados. Utilizaron el modelo YOLOv5 entrenado con un conjunto de datos propio compuesto por imágenes de productos en buen y mal estado. La aplicación permite a los usuarios registrar y supervisar el inventario de manera automatizada. En las pruebas, el modelo alcanzó un mAP@0.5 del 93.09 % y

un mAP@0.5:0.95 del 89.04 %, lo que demuestra su capacidad para detectar con precisión los productos y evaluar su estado. La metodología incluyó la recopilación y etiquetado de datos, la optimización de hiperparámetros mediante algoritmos genéticos y el desarrollo de una interfaz móvil. Los resultados sugieren que esta solución puede mejorar significativamente la gestión de inventarios en supermercados y reducir errores en la identificación de productos dañados. Esta investigación se desarrolló dentro del campo de aplicación para la automatización de la supervisión de inventarios y control de calidad en supermercados, detectando productos dañados mediante visión artificial.

2.2.3. Antecedentes comerciales

La startup colombiana AI Turing⁴ es una propuesta comercial que brinda servicios de mercadotecnia implementando algoritmos de inteligencia artificial para reconocer la ubicación de los productos de sus clientes en las estanterías de los supermercados. De acuerdo con la publicación de la revista Semana, esta herramienta permite enviar información precisa y en tiempo real de cómo se lleva a cabo una estrategia de manejo de marca en los puntos de venta. Actualmente tienen operaciones en Perú, Ecuador, Bolivia, República Dominicana, Panamá y México.

Su competencia internacional es TraxRetail⁵ una empresa fundada en Singapur con clientes como Coca-Cola, Nestlé, Unilever, P&G y Heineken. Trax recopila imágenes a través de una aplicación móvil, cámaras de estantería, cámaras domo y los ojos digitales de robots itinerantes. Este enfoque híbrido les permite prosperar en cualquier entorno minorista, desde farmacias hasta supermercados y tiendas. La plataforma Trax utiliza los datos de las estanterías para medir la ejecución, el cumplimiento de las tiendas, la cuota de mercado, el rendimiento del personal, entre otras.

2.2.4. Discusión

Con base en la literatura revisada sobre el reconocimiento de imágenes en especial de cajas de productos en el comercio minorista, similar al contexto de farmacias en Colombia, se puede observar que a pesar de los numerosos estudios que prueban una amplia gama de métodos y técnicas, YOLO destaca como la arquitectura más utilizada. Lo anterior probablemente debido a su versatilidad, eficiencia y velocidad en la detección de objetos en diversos contextos, además de su capacidad para ser implementado en dispositivos móviles. Sin embargo, se evidenció que la precisión y

⁴<https://aituring.co/>

⁵<https://traxretail.com/>

eficiencia de YOLO mejora significativamente en algunos estudios mediante la integración de técnicas de transferencia de aprendizaje, especialmente útiles para reconocer objetos de tipos complejos o poco comunes. Por otro lado, resulta interesante analizar la investigación de [14], donde las métricas obtenidas no cumplieron con las expectativas. Esto podría deberse a que la implementación de YOLO para el reconocimiento de números no era la opción más adecuada. En este contexto, posiblemente se habrían logrado mejores resultados mediante la aplicación de un modelo especializado en OCR. Se resalta la importancia de los procesos de aumento de datos como esenciales para mejorar el rendimiento, como se demostró en [15] y [22].

Las mejoras continuas en los algoritmos y en las técnicas de entrenamiento, así como la disponibilidad de los conjuntos de datos de alta calidad, son cruciales para superar las limitaciones actuales y expandir las aplicaciones del en diversos campos. En ese sentido se pudo evidenciar la evolución de YOLO a lo largo de los años desde su versión número 2 hasta la actual versión 8, también puede notarse el avance en general sobre las maneras en medir la eficiencia de la YOLO. La comparación entre los estudios revela que mientras algunos logran altas precisiones (como en [23] y [18]), otros enfrentan desafíos significativos debido a la naturaleza y calidad de los datos de entrenamiento (como en [14]).

Las técnicas de transferencia de aprendizaje y el uso de algoritmos híbridos han demostrado ser estrategias efectivas para mejorar el rendimiento de los modelos, sugiriendo su mayor adopción en futuros desarrollos. La calidad y el balanceo de los conjuntos de datos, así como las herramientas utilizadas para el etiquetado, juegan un papel crucial en el desempeño de los modelos de DL, como se evidencia en [15] y [16]. La inversión en la mejora de estas áreas puede conducir a resultados significativamente mejores y más consistentes. La implementación de modelos en aplicaciones móviles y su desempeño en escenarios reales son aspectos críticos que determinan el éxito de estos desarrollos. Las pruebas y ajustes en contextos reales, como se hizo en [14] y [15], son esenciales para validar y mejorar estos sistemas.

Con base en este levantamiento bibliográfico, se decidió utilizar la red neuronal YOLO para lograr los objetivos propuestos, reconocer medicamentos por medio de imágenes leídas en una aplicación móvil.

2.3. Marco Teórico

Para facilitar la comprensión sobre el desarrollo y los resultados obtenidos en este estudio, cuyo objetivo era reconocer medicamentos por medio de visión artificial a partir de una aplicación móvil, es necesario abordar algunos conceptos y técnicas que se mencionarán a lo largo del documento como: medicamentos, inteligencia artificial, aprendizaje automático y visión artificial.

2.3.1. Medicamentos

Los medicamentos son un preparado farmacéutico que se usa para la prevención, alivio, diagnóstico, tratamiento, curación o rehabilitación de la enfermedad. Como están disponibles al público, el Instituto Nacional de Vigilancia de Medicamentos y Alimentos (INVIMA) se encarga de establecer las políticas, regulaciones y controles de calidad de estos. Dentro de las regulaciones hay ciertas características sobre las etiquetas, empaques, envases y cajas que deben cumplir los laboratorios farmacéuticos [24]. Esta comprensión es importante ya que la investigación que se desarrolló buscaba reconocer la información de medicamentos cuyo empaque fuera de tipo caja. No incluye reconocimiento de medicamentos con otro tipo de presentaciones, ya sea blíster, sachets, píldoras, frascos, entre otros. En la Figura 2.2 se puede identificar algunos parámetros relevantes de la presentación de medicamentos.



Figura 2.2: Algunos tipos de empaque de medicamentos y etiquetas. Fuente: elaboración propia

Cabe aclarar que el alcance del presente trabajo se limita a reconocer de la caja de los medicamentos, únicamente el nombre comercial. Por ejemplo, en el caso de la Figura 2.2 la aplicación móvil reconocería la caja del medicamento por el nombre de -Aspirina-. La contribución a la gestión de inventarios de este estudio consiste en indicar al usuario la cantidad total de cada medicamento detectado en la imagen. Se propone para trabajos futuros el reconocimiento de los otros datos proporcionados por la caja como fecha de vencimiento etc.

2.3.2. Inteligencia Artificial

La Inteligencia Artificial -Artificial intelligence (AI)- es un campo general que abarca el Aprendizaje Automático -Machine Learning (ML)- y el Aprendizaje Profundo -Deep Learning (DL)-, pero que también incluye muchos más enfoques que no implican ningún aprendizaje. Durante bastante tiempo, muchos expertos creyeron que la AI a nivel humano podría lograrse haciendo que los programadores elaboraran a mano un conjunto suficientemente grande de reglas explícitas para manipular el conocimiento. Este enfoque se conoce como AI simbólica. Aunque la AI simbólica demostró ser adecuada para resolver problemas lógicos como jugar al ajedrez, resultó intratable descubrir reglas explícitas para resolver problemas más complejos y difusos, como la clasificación de imágenes, el reconocimiento de voz y la traducción de idiomas. Con esto surgió un nuevo enfoque para ocupar el lugar simbólico de la AI: el ML [25].

2.3.3. Aprendizaje Automático

Con Aprendizaje Automático -Machine Learning (ML)- los humanos introducen datos, así como las respuestas que se esperan de los datos, y salen las reglas. Estas reglas se pueden aplicar a nuevos datos para producir respuestas originales. Un sistema de aprendizaje automático se entrena en lugar de programarse explícitamente. Se presenta con muchos ejemplos relevantes para una tarea, y encuentra una estructura estadística en estos ejemplos que eventualmente permite al sistema crear reglas para automatizar la tarea [25].

Entre las técnicas más utilizadas para la clasificación usando métodos de **ML** se encuentra las **Redes Neuronales Artificiales (RNA)** o en inglés Neural Network (NN), es una referencia a la neurobiología [25] y corresponden a un sistema interconectado de funciones implementadas a nivel computacional basadas en la teoría de grafos, la cual sentó las bases matemáticas del estudio en RNA⁶ con el objetivo de

⁶<https://la.mathworks.com/discovery/neural-network.html>

aplicarlas a diferentes problemas. La estructura de las RNA se conforma por tres capas principales: La capa de entrada recibe los valores o datos de diferentes características; la segunda capa, conocida como capa oculta, se encarga de gestionar el entrenamiento; y la capa de salida obtiene el resultado de algún valor o función de predicción [23], [26].

La Figura 2.4 muestra el esquema de la red neural multicapas 10-10-5 donde las entradas (4A) representa cualquier variable. Capa de entradas (4B), capa oculta (4C) y capa de salidas (4D). Salidas (4E): Representa también cualquier variable de interés para el usuario. Los pesos entre cada neurona (4F) están representados por un punto negro (4G) [27]. El entrenamiento se puede hacer por medio del “*perceptrón*” o por medio del algoritmo de “*retropropagación del error*”.

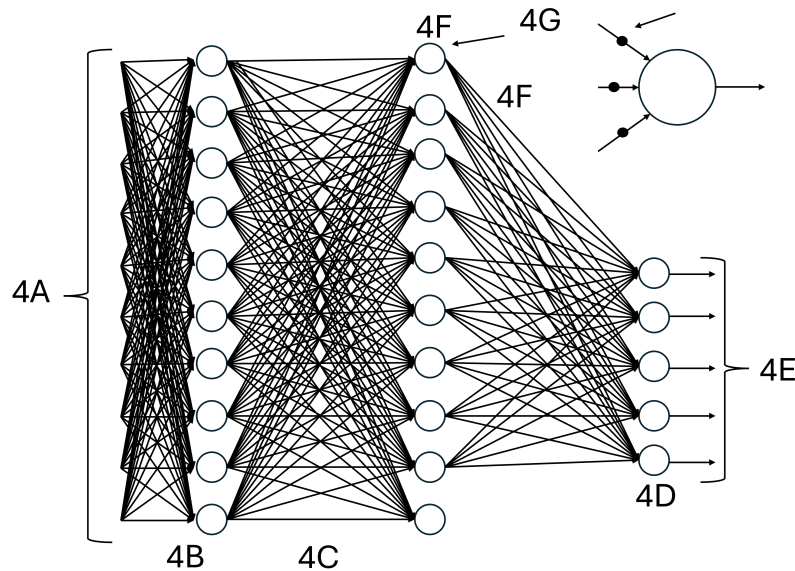


Figura 2.3: Topología de una RNA. Fuente: Ponce Cruz.

DL es un subcampo específico del ML que implica un aprendizaje a partir de capas sucesivas de representaciones cada vez más significativas. Lo profundo se refiere a la idea de capas sucesivas de representaciones. El número de capas que contribuyen a un modelo de datos se denomina *profundidad del modelo*. El DL moderno implica decenas o incluso cientos de capas sucesivas de representaciones y todas se aprenden automáticamente a partir de las Redes Neuronales Artificiales (RNA) [25]. La Figura 2.4 es una representación de como se organizan estos campos del conocimiento de la AI

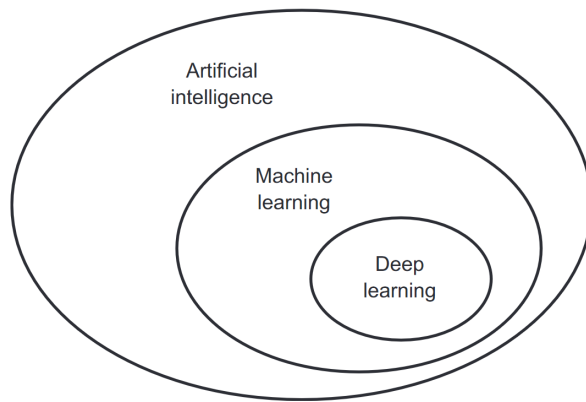


Figura 2.4: Diagrama de Venn conjunto de AI, ML Y DL [Deep Learning Ian goodfellow].

La **Convolutional Neural Networks (CNN)** es una red que contiene filtros entre sus diferentes capas. Es muy útil para el procesamiento de datos en cuadrícula, como lo son señales muestreadas con un intervalo regular de tiempo o una imagen, pensada como una cuadrícula 2D de píxeles [28]. Su nombre se deriva del hecho de que estas redes en algunas de sus capas emplean la operación matemática de convolución para extraer características de la entrada, además, hacen uso de una operación llamada pooling que calcula un nuevo valor, este valor usualmente es el máximo o el promedio de los píxeles. para un píxel en una imagen sustrayendo el píxel original con sus vecinos aledaños. Estas dos herramientas la hacen muy eficiente en problemas reales de detección de objetos en imágenes.

La arquitectura de estos modelos se puede entender apreciando la Figura 2.5. En primer instancia se tiene el Backbone, es quien recibe la entrada de información siendo el responsable de extraer y codificar las características de los datos entrantes, normalmente se compone de capas convolucionales. El Neck es el responsable de refinar aun más las características extraídas en el paso anterior, en sus capas busca combinar y mezclar las características de la imagen. En general, se compone de neuronas totalmente conectadas y redes tipo cuello de botella. Por último, el Head es quien realiza la inferencia, es decir, la predicción de la clase a la que pertenece el objeto de la imagen y el post-procesamiento con las cajas dibujadas sobre cada objeto [29].

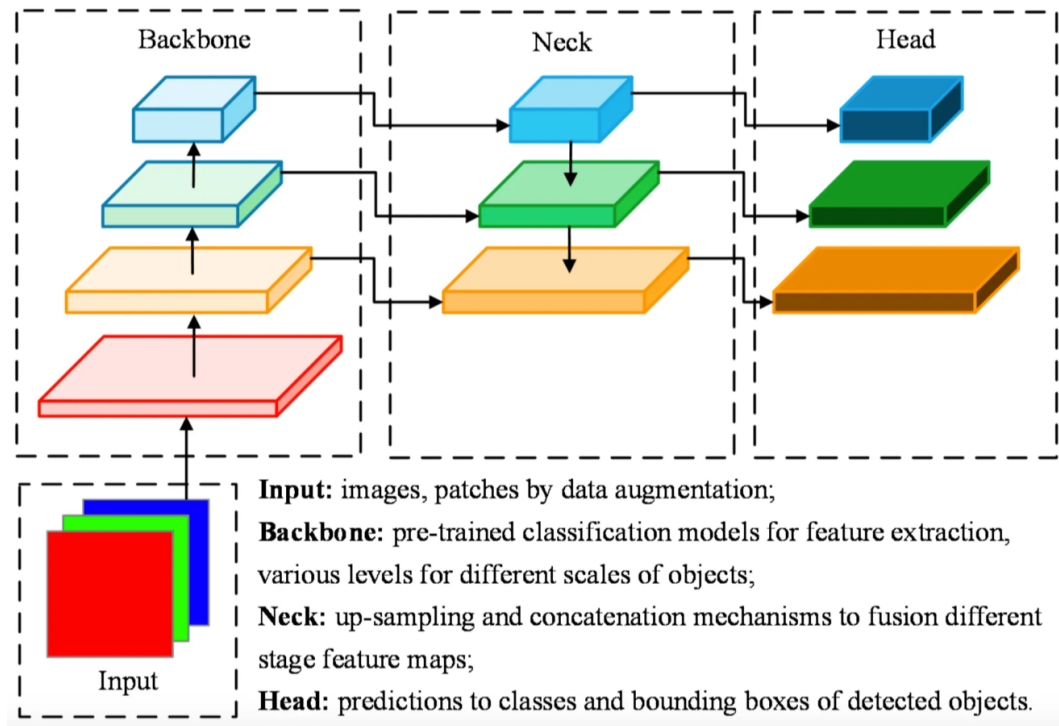


Figura 2.5: Arquitectura de una CNN. Fuente: Megha Shroff 2023

2.3.4. Visión Artificial

La Visión Artificial (VA), también llamada Visión por Computador, es un campo científico interdisciplinario que estudia problemas metodológicos y algorítmicos para interpretar o comprender escenas del mundo real mediante la captura, procesamiento y análisis de imágenes o vídeos [30]. Pero, ¿Qué son las imágenes? Antes de abordar a profundidad los tecnicismos de la VA es importante mencionar la naturaleza de las imágenes, en este caso, las digitales.

De acuerdo con [31], una imagen digital se concibe como una representación bidimensional de una escena que existe en el mundo tridimensional. La imagen es el resultado de la adquisición de una señal proporcionada por un sensor, que convierte la información del espectro electromagnético en codificaciones numéricas. Una imagen digital está compuesta por un número finito de elementos, cada uno de los cuales tiene una ubicación y un valor determinado; estos elementos se denominan picture elements, image elements, pels, y pixels. Pixels es el termino más utilizado para designar los elementos de una imagen digital. El tipo de información más elemental que se puede obtener de una imagen se llama información *sintáctica* y se ocupa únicamente de los valores de los píxeles. La otra información, llamada información

semántica, se ocupa del conocimiento y significado de la información [32].

Dentro del campo de la VA se pueden distinguir tres niveles de interpretación de imagen [32]. La **visión de bajo nivel** opera directamente sobre las imágenes y producen resultados basados en píxeles, que permanecen únicamente dentro de las coordenadas de la imagen original. La **visión de nivel medio** recibe la imagen o los resultados de la salida de algoritmos de visión de bajo nivel y producen alguna salida distinta de los píxeles, dentro de los sistemas de coordenadas de la imagen original. Los sistemas de **visión de alto nivel** se utilizan para tomar decisiones abstractas o, basándose en los datos visuales, realizar un análisis de la escena. En este nivel se utilizan los resultados producidos por los algoritmos de visión de bajo nivel o de nivel medio para analizar formas, reconocer y localizar objetos.

El reconocimiento de patrones, aplicable al reconocimiento de objetos, implica que cuando una imagen se presenta al sistema, este deba comparar la imagen con los modelos almacenados para ver si algún modelo coincide con un objeto presente en la imagen. Además de identificar el objeto, el sistema también determina la ubicación exacta del objeto dentro de la imagen. Esta ubicación se conoce como la “pose” del objeto. Los desafíos en el reconocimiento de objetos incluye situaciones como: El fondo de los objetos tiene mucha textura, las condiciones de iluminación son desconocidas e incontroladas, la escena comprende demasiados objetos, la cantidad de modelos de objetos es muy grande en el almacenamiento y algunos objetos pueden estarse tocando y/o ocluyendo con otros objetos [32]. Para desarrollar un sistema de VA, se necesita ejecutar los procesos que se aprecian en la Figura 2.6

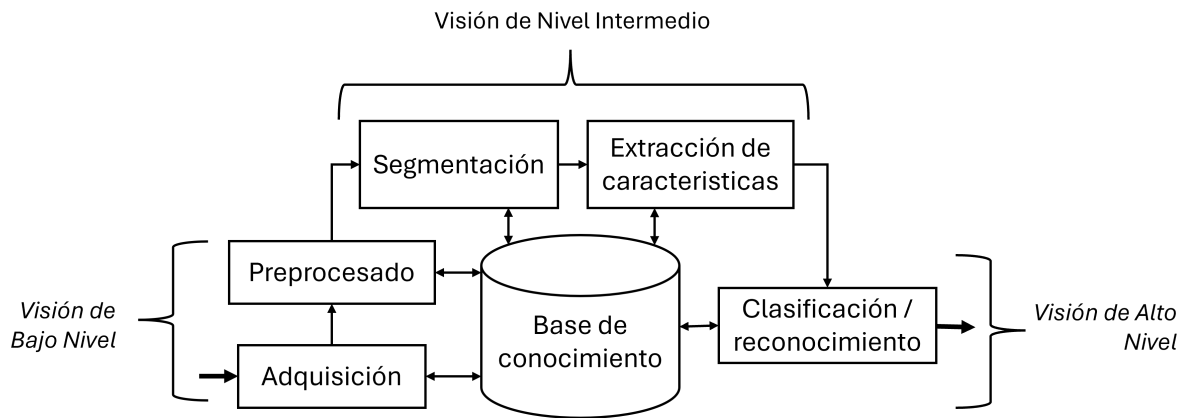


Figura 2.6: Proceso en un sistema de VA. Fuente: Adaptación propia

En resumen estos procesos consisten en: **Primero**, los estímulos más relevantes de las imágenes percibidos por los sensores como tamaño, forma, color o textura, se utilizan para caracterizar los objetos definiendo así un *espacio de medición*. **Segundo**,

se selecciona un subconjunto de atributos cuyos valores permitan una agrupación o clasificación coherente de los objetos, creando así un *espacio de características*. **Tercero**, se utilizan los valores de atributos seleccionados para crear modelos que aprendan a caracterizar los objetos. Estos modelos se almacenan para su posterior uso en reconocimiento de objetos. El rango de valores de funciones de decisión es el *espacio de decisión*. El reconocimiento de objetos similares se logra mediante la aplicación de las reglas aprendidas en el paso final, comparando las características del objeto con los modelos almacenados. Posteriormente se pueden realizar refinamientos y ajustes para mejorar la calidad y la velocidad del reconocimiento [32]. Se analizará detalladamente cada area del proceso.

2.3.4.1. Adquisición:

Consiste en obtener la imagen digital. Por ejemplo las cámaras que por medio de sistema de lentes puede generar imágenes de cualquier objeto o de cualquier escena. Para que los procesos de aprendizaje sean exitosos es fundamental contar con datos de entrenamiento representativos e imparciales en una cantidad suficiente para obtener buenos resultados, conocidos como conjunto de datos o dataset. En otras palabras, un dataset es el conjunto de muchas imágenes utilizadas para el entrenamiento de los modelos de VA. A lo largo de los años, se ha desarrollado una gran variedad de conjuntos de datos para probar y evaluar algoritmos de VA tales como PASCAL, ImageNet, KITTI, Sintel y COCO [33]. Resulta necesario en algunas ocasiones mejorar y corregir las imágenes del dataset para favorecer los resultados de los modelos. Esto es conocido como *Procesamiento de Imagen*.

2.3.4.2. Preprocesado:

El campo de la VA tiene una estrecha relación con el campo del procesamiento de imágenes, en el procesamiento de imágenes, la atención se centra en la transformación de imágenes, mientras que en la VA, la atención se centra en extraer información de los datos de la imagen [32]. Si bien algunos pueden considerar que el procesamiento de imágenes está fuera del ámbito de la VA, la mayoría de los métodos de clasificación requieren cuidado en el diseño de las etapas de procesamiento de imágenes para lograr resultados aceptables [33].

El procesamiento digital de imágenes busca transformar imágenes mediante la implementación de algoritmos computacionales; se puede considerar como un sistema donde su entrada y salida son imágenes digitales [34]. Se trata de mejorar la apariencia visual de una imagen para el análisis artificial. En este proceso se incluyen

técnicas de eliminación de ruido y realce de detalles. En cualquier caso, debe ser posible dividir la imagen de acuerdo a una característica de homogeneidad predefinida. Entre las técnicas esta la binarización de imágenes (thresholding) y técnicas de detección de bordes, entre muchas más [35].

De acuerdo con el artículo de Krut Patel⁷ las características son partes o patrones de un objeto en una imagen que ayudan a identificarlo e incluyen propiedades como esquinas, bordes, regiones de puntos de interés, crestas, etc.

2.3.4.3. Segmentación:

La segmentación de imágenes es el proceso que subdivide una imagen en sus partes constituyentes u objetos. El nivel al que se lleva a cabo esta subdivisión depende del problema que se esté resolviendo. La segmentación de imágenes no solo implica la discriminación de objeto y fondo, sino también la separación entre distintas regiones; un método para ello se conoce como watershed segmentation [31]. Los resultados de la misma serán utilizados en etapas sucesivas para su descripción, reconocimiento e interpretación. Los métodos de la segmentación asumen que las regiones a extraer poseen algunas características homogéneas distintivas. Pueden existir límites físicos entre regiones similares que no aparecen como tales en la imagen [35].

2.3.4.4. Extracción de características:

Una vez logrado el preprocesamiento y el nivel deseado de segmentación, donde se tienen los datos de píxeles en bruto, ya sean los puntos del contorno de una región o los puntos de toda la región [35], se aplica a los segmentos alguna técnica de extracción de características. La selección de un método de extracción de características es el factor más importante para lograr un alto rendimiento en el reconocimiento de imágenes. La extracción de características se ha definido como: extraer de los datos brutos la información más adecuada para la clasificación, minimizando la variabilidad del patrón dentro de la clase y mejorando la variabilidad del patrón entre clases [31]. Los algoritmos que son usados para encontrar puntos de interés, usualmente se denominan detectores, mientras que aquellos se encargan de describir las características son conocidos como descriptores [36].

⁷<https://n9.cl/metowardskrut>

2.3.4.5. Clasificación:

2.3.4.5.1. YOLO

Es un algoritmo presentado en el 2016 que aborda el reconocimiento por medio de *Una única CNN que calcula simultáneamente para varios cuadros delimitadores, la probabilidad de todas las clases*. A esto se debe su nombre YOLO: You Only Look Once, **mira una vez una imagen para predecir qué objetos están presentes y dónde están**. Esto implica que YOLO deba razonar globalmente sobre toda la imagen a la hora de hacer predicciones. Esto le permitió optimizar el rendimiento de la detección procesando imágenes en tiempo real en 45 fotogramas por segundo. La versión Fast llegando a procesar 155 fotogramas por segundo. Todo esto mientras se lograba mAP mejores que otros detectores en tiempo real [37].

Lo que hace YOLO es dividir la imagen de entrada en una cuadrícula $S \times S$. Si el centro de un objeto se encuentra en una celda, esa celda es responsable de detectar el objeto. Cada celda de la cuadrícula predice los cuadros delimitadores y las puntuaciones de confianza para esos cuadros. Estas puntuaciones reflejan el grado de confianza de que tan probables es que la caja contenga un objeto y también la precisión con la que cree que es la caja que predice. Si no existe ningún objeto en esa celda, las puntuaciones de confianza deben ser cero [37].

Existen distintas implementaciones de YOLO⁸, una de ellas es la de ultralytics 11, que cuenta con un desarrollo de scripts de entrenamiento, validación y arquitecturas de detección usando DL. El backbone de YOLO es la red **CSPDarknet**, esta es una CNN y de base para la detección de objetos. Emplea una estrategia CSPNet para dividir el mapa de características de la capa base en dos partes y luego las fusiona mediante una jerarquía de etapas cruzadas. El uso de una estrategia de división y fusión permite un mayor flujo de gradiente a través de la red. El Neck utiliza PA-Ne y un SSP. PA-Net facilita el flujo de información a través de diferentes resoluciones espaciales, lo que permite que el modelo capture características de múltiples escalas de manera efectiva. SSP o Spatial pyramid pooling realiza la conversión de un mapa de características de cualquier tamaño a una representación de longitud fija para ser enviada al head.

- **Supresión de no máximos** Un método utilizado para mejorar la detección eliminando cajas delimitadoras redundantes o innecesarias. Consiste en encontrar los mejores recuadros de clasificación con base en ciertos criterios. El algoritmo recibe una serie de cajas propuestas y devuelve una serie filtrada de cajas. Se utiliza la puntuación de confianza para realizar este filtro. Posteriormente, se calcula y compara el Intersection Over Union (IoU) de la caja actual con las

⁸<https://yolov8.org/yolov8-architecture/>

demás propuestas. Si este valor supera un umbral N , la caja se descarta, de lo contrario se conserva ⁹

- **Intersección sobre Unión (IoU)** Se utiliza en el algoritmo de Non-maximum suppression (NMS) para medir el nivel de solapamiento de las cajas propuestas por un algoritmo de detección. Esto se puede apreciar gráficamente en la figura

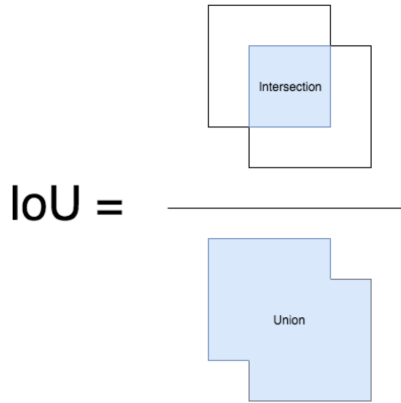


Figura 2.7: Representación del calculo de la IoU. Fuente: Elaboración propia

2.3.4.6. Métricas de desempeño

- **Exactitud:** La exactitud es la métrica más utilizada para medir el desempeño de una clasificación. Corresponde a la cantidad de veces que el clasificador identificó los datos de manera correcta respecto al número total de datos [38] . Se calcula a través de la ecuación 2.3

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.1)$$

Donde

- TP=True positive o verdaderos positivos.
 - FP=False positive o falsos positivos.
 - TN=True negative o verdaderos negativos.
 - FN=False negative o falsos negativos.
- **Precisión:** La precisión mide la proporción de datos correctamente identificados como positivos del total identificados como positivos. Esta se calcula con la ecuación 2.4

⁹<https://towardsdatascience.com/non-maximum-suppression-nms-93ce178e177c>

$$p = \frac{TP}{TP + FP} \quad (2.2)$$

- **Sensibilidad o Recall** Se usa cuando los F N son inaceptables en el clasificador, se puede considerar como la exactitud de los casos positivos, además es útil cuando se evalúan datasets no balanceados [38] . Esta se calcula con la ecuación 2.5

$$Recall = \frac{TP}{TP + FN} \quad (2.3)$$

- **mAP:** Esta metrica es la Precisión media. Se utiliza para medir cuán precisas son las predicciones de las cajas delimitadoras al comparar la superposición entre la caja delimitadora predicha y la caja delimitadora real (ground truth). La precisión media se calcula promediando los valores de precisión en la curva de precisión-sensibilidad como se muestra en la ecuación 2.6

$$mAP = \frac{1}{|classes|} \cdot \sum_{c \in classes} \frac{TP(c)}{TP(c) + FP(c)} \quad (2.4)$$

2.4. Discusiones

Anteriormente se realizó un levantamiento bibliográfico en un periodo de tiempo entre el 2013 y 2023 de estudios nacionales e internacionales sobre la detección y clasificación de objetos en imágenes usando visión por computador. La búsqueda se realizó en fuentes como IEEE Xplore y Scopus, utilizando palabras clave específicas para obtener literatura relevante. En el ámbito global, se destaca el impacto del Deep Learning y las CNN que han permitido avances significativos en estas tareas. Modelos como YOLO han sobresalido por su capacidad para realizar detecciones rápidas y precisas, especialmente en aplicaciones como la clasificación de productos en estanterías y el reconocimiento de píldoras, con precisiones de hasta el 96.6 %.

A nivel nacional, la implementación de visión artificial en el reconocimiento de medicamentos es limitada, aunque existen estudios relevantes. En contraste, los estudios internacionales ofrecen un panorama más amplio, con enfoques que combinan modelos híbridos y transferencia de aprendizaje para optimizar el rendimiento, evidenciando la evolución de YOLO desde su primera versión. YOLO es una herramienta destacada por su versatilidad y eficiencia, siendo ampliamente adoptada en tareas de detección de objetos. Sin embargo, su efectividad depende de factores como la calidad de los datos y el preprocesamiento. En ciertos casos, como el reconocimiento de números, los modelos especializados en OCR pueden ofrecer mejores resultados.

Como bien se identificó en la literatura, se decide usar YOLO por su capacidad para integrar modelos en aplicaciones móviles y su desempeño superior en tareas de detección para reconocer el nombre comercial de cajas de comercios minoristas. Ahora bien, de acuerdo con Ultralytics ¹⁰, Se destaca YOLOv8 entre las otras versiones por ser un modelo versátil capaz de realizar tareas que incluyen detección de objetos, segmentación de instancias, clasificación y estimación de poses. En comparación con versiones anteriores como YOLOv3 y YOLOv4, YOLOv8 ofrece mejoras significativas en velocidad y precisión debido a su arquitectura optimizada. Por esa razón, se decide usar la versión 8 del modelo en el desarrollo de la presente investigación.

Finalmente, se explican conceptos clave sobre visión artificial y el proceso de reconocimiento de objetos, desde la adquisición de imágenes hasta la clasificación para una mejor comprensión técnica e histórica de la visión por computador y del estudio de esta investigación.

¹⁰<https://docs.ultralytics.com/models/>

3. APLICATIVO MÓVIL PARA RECONOCIMIENTO DE MEDICAMENTOS

3.1. Introducción

En este capítulo se presenta el desarrollo del aplicativo móvil para el reconocimiento de diez medicamentos en presentación tipo caja y la especificación de sus cantidades. La descripción se organiza en dos secciones principales: la primera expone los procesos y métodos utilizados en el entrenamiento del modelo de aprendizaje profundo YOLO. La segunda describe los procesos y métodos implementados en el desarrollo funcional del aplicativo móvil que contiene dicho modelo.

3.2. Reconocimiento de medicamentos

3.2.1. Descripción de la base de datos

La base de datos de este proyecto se construyó desde cero, por lo tanto, primero se determinaron los medicamentos objeto de estudio. La selección se fundamentó en dos criterios principales: su relevancia comercial y su representatividad en el gasto farmacéutico en Colombia. En el primer caso, se eligieron medicamentos de venta libre como Dolex, Aspirina, Noraver, ibuprofeno y vitamina C, cuya adquisición de imágenes resultó más sencilla y que, según un artículo de la revista Semana¹ presentan un crecimiento significativo en ventas a nivel nacional. En el segundo criterio, de acuerdo con el Boletín de Gastos de Bolsillo en Medicamentos del Ministerio de Salud de Colombia [39], se identificó que los analgésicos y antigripales concentraron el 63,8% del gasto farmacéutico de la población, lo que motivó la inclusión de medicamentos como Ibuflash y Buscapina. Finalmente, se añadieron de manera arbitraria otros productos ampliamente reconocidos en los hogares colombianos, como Gaviscon, Alka-Seltzer y Levonorgestrel (Postday), por su uso frecuente en el tratamiento de condiciones comunes. La tabla 3.1 presenta un resumen del gasto farmacéutico del consumidor y la agrupación de los medicamentos seleccionados según su clasificación terapéutica, con base en la información del Boletín de Gasto de Bolsillo en Medicamentos de 2022.

¹<https://n9.cl/semana-medicamentos>

Tabla 3.1: Gasto farmacéutico según grupo terapéutico.

Grupo terapéutico	Medicamentos seleccionados	Gasto farmacéutico
Analgésicos	Aspirina, Ibuprofeno, Ibuprofeno	48.54 %
Antigripales	Dolex Gripe, Noraver Garganta	15.27 %
Antiácidos	Gaviscon, Alkaseltzer	5.29 %
Anticonceptivo	Levonorgestrel (Postday)	2.10 %
Antiespasmódico	Buscapina	1.25 %
Otros	Vita C mk	-

A continuación, se realizó un levantamiento de imágenes de los medicamentos seleccionados. El reconocimiento adecuado de los medicamentos mediante la implementación de la arquitectura YOLO depende en gran medida del conjunto de datos de imágenes con el que se entrena el modelo. Este debe ser representativo y coherente, es decir, contener unas características consistentes en todas las imágenes que compone el conjunto de datos. Este proceso fundamental debe considerar tanto las características de forma, la ubicación espacial y su información implícita. En este sentido, es crucial tener en cuenta la forma del contenedor del medicamento (definir los bordes y esquinas) para facilitar la identificación del contorno del objeto; asimismo, la textura, los patrones de color, la ubicación consistente del texto, el contraste entre el texto y el fondo de la caja etc. La figura 3.1 muestra las imágenes de las cajas de los 10 medicamentos utilizados para el reconocimiento.



Figura 3.1: Imágenes de las 10 cajas de medicamentos seleccionados para el estudio. Fuente: elaboración propia

Para garantizar la robustez del sistema de reconocimiento de medicamentos, se propone la implementación de dos procedimientos de toma de datos complementarios que se llevó a cabo a través de la adquisición de imágenes en formato RGB para los 10

medicamentos objetivo, mencionados anteriormente. En primer lugar, se plantea una adquisición de datos en un entorno controlado, donde se garantizan condiciones adecuadas de luminosidad y temperatura de color, capturando imágenes con variaciones en estos parámetros para aumentar la generalización del modelo y adicionalmente, se plantea un segundo procedimiento que consistió en la adquisición no controlada de imágenes en establecimientos comerciales.

Este proceso permitió capturar la variabilidad real en las condiciones de iluminación, el fondo y la disposición de los medicamentos, exponiendo al sistema a escenarios más desafiantes y mejorando su capacidad de adaptación a entornos reales. La combinación de estos dos procedimientos de toma de datos permitió crear un conjunto de datos robusto y diverso que representan adecuadamente la variabilidad real de los medicamentos en diferentes condiciones. Esto garantizará que el sistema de reconocimiento de medicamentos sea capaz de funcionar de manera precisa y confiable en entornos reales, ofreciendo un valor significativo para la gestión de inventarios en droguerías y farmacias.

3.2.1.1. Adquisición controlada

La adquisición de datos en un entorno controlado, se llevó a cabo en un espacio donde se garantizaban condiciones adecuadas de luminosidad y temperatura de color. Con el objetivo de capturar la variabilidad de las condiciones de iluminación, se capturaron imágenes con niveles de luminosidad que iban desde 10 lúmenes hasta 1400 lúmenes, con incrementos aproximados de 40 lúmenes en cada paso, sobre temperaturas de color, tanto cálidas (3000 kelvin) como frías (11000 kelvin), simulando diversos escenarios de iluminación real.

Adicionalmente, se realizaron tomas de registro sobre cada uno de los canales independientes de rojo, azul y verde. Este proceso permitió analizar el comportamiento individual de cada componente cromático en la identificación de los medicamentos objetivo, proporcionando información valiosa para el entrenamiento y la optimización del modelo. La figura 3.2 consiste en una muestra de 4 imágenes del conjunto de datos obtenido mediante adquisición controlada, específicamente, de la caja del Levonogestrel. Se puede observar que bajo una luz azul, la caja tiende a desvanecerse debido a su color monocromático, lo que hace que el color deje de ser una característica relevante.



Figura 3.2: Muestra conjunto de datos adquisición controlada. Fuente: elaboración propia

Así mismo, se empleó un trípode que permitió obtener ángulos de variación controlados para cada captura, esto garantizó la variabilidad en la perspectiva y la posición del medicamento en las imágenes, minimizando la incertidumbre introducida por la posición de la cámara en un entorno real. El proceso de captura se llevó a cabo utilizando un dispositivo móvil con una resolución de 3072 x 3072 píxeles en un formato 1:1 y con una focal de 1.8.

Finalmente, se adquirieron 1200 imágenes bajo este tipo de condición que corresponde al 23% del total de registros del conjunto de datos final. de las 1200, el 70% corresponde a imágenes con temperatura de color fría y cálida, mientras que el 30% restante se compone de imágenes de los canales cromáticos individuales; todos los registros se tomaron con una proporción de cantidad en una proporción 1:1. Es importante destacar que todos los registros presentan variaciones de luminosidad como se describió anteriormente.

La combinación de la adquisición en un entorno controlado, la variación sistemática de las condiciones de iluminación y temperatura de color, la captura de información de cada canal cromático y el uso de un trípode para controlar los ángulos de captura, dieron como resultado un conjunto de datos robusto y diverso que representa adecuadamente la variabilidad real de los medicamentos en diferentes condiciones.

3.2.1.2. Adquisición no controlada

En contraste con la adquisición de datos en un entorno controlado, donde se garantizaban condiciones artificiales, la recolección de datos en farmacias y tiendas de barrio presentó un escenario mucho más desafiante y realista. Aquí, la iluminación y los ángulos de toma estaban fuera de control. Para abordar estos desafíos, se adoptó un enfoque flexible y adaptable en la recolección de datos. Debido al ambiente de estrés en el que se recolecta este conjunto de datos, se utilizaron diferentes dispositivos

de captura, diferentes formatos y resoluciones, todo dependiendo del recurso que se tuviera al alcance en el momento de la adquisición. La amplia gama de datos recopilados permitió capturar variaciones en los empaques de medicamentos y las condiciones de iluminación. Esto hizo que los datos fueran más representativos del entorno real en el que se utilizaría la aplicación, aumentando su generalización y confiabilidad.

Esta etapa de adquisición de datos se llevó a cabo durante las primeras etapas de ejecución del proyecto, lo que permitió obtener una representatividad del 77 % (3965 imágenes) del conjunto total de datos. La figura 3.3 consiste en una muestra de 4 imágenes del dataset obtenido mediante adquisición no controlada en comercios minoristas y droguerías.



Figura 3.3: Muestra conjunto de datos adquisición no controlada. Fuente: elaboración propia

La adquisición de datos en un entorno no controlado se realizó a través de una herramienta desarrollada para el aplicativo móvil de reconocimiento de medicamentos, la cual permitía tomar registros fotográficos de los medicamentos objetivo y almacenarlos de manera centralizada en la nube para su posterior procesamiento y análisis.

Cabe mencionar que este proceso de adquisición se realizó de manera cíclica en conjunto con el etiquetado de imágenes -que se explicará en la siguiente sección-. A medida que se adquirían las imágenes, estas eran etiquetadas de inmediato, lo que permitía identificar los medicamentos con menor cantidad de etiquetas. Con esta información, se priorizaba la adquisición de más imágenes de aquellos medicamentos con escasa representación, asegurando así un conjunto de datos balanceado en la medida de lo posible. La adquisición temprana facilitó la identificación de posibles sesgos en los datos en las primeras etapas del proyecto. Esto permitió implementar estrategias de mitigación oportunas, como el sobremuestreo o el submuestreo, para

garantizar que el modelo no estuviera sesgado hacia ciertos medicamentos

Finalmente entre los dos conjuntos de datos - controlados y no controlados - se obtuvo un total de 5150 imágenes

3.2.2. Etiquetado de imágenes

De acuerdo con el flujo MLOps presentado en la figura 1.2, el proceso de etiquetado de datos se llevó a cabo a través de la plataforma Roboflow² y su módulo de anotaciones. Para garantizar una representatividad óptima de los medicamentos, se optó por un etiquetado manual utilizando cajas poligonales, esta herramienta permitió una aproximación del encuadre del objeto dentro de la imagen puesto que se adaptaba a la forma del empaque del medicamento sin importar su orientación, ángulo o incluso si había un sobre-posicionamiento con respecto a otro empaque. El etiquetado manual, a pesar de ser un proceso más lento, ofreció mayor precisión y control sobre la calidad de las etiquetas. Un ejemplo del resultado del etiquetado de las 10 clases dentro de las imágenes se observa en la figura 3.4.



Figura 3.4: Muestra dataset etiquetado con las 10 clases del estudio. Fuente: elaboración propia

En la figura 3.4 se aprecia cada recuadro de color que corresponde a una clase o medicamento según el código de colores de la tabla 3.2. Las zonas que no se etiquetaron corresponden al entorno farmacéutico o de comercio minorista y por ende se compone de estanterías y demás medicamentos que no son de interés. Estos otros objetos se pueden presentar en diferentes empaques y colores. Del dataset de cinco mil imágenes (5150), en promedio, se etiquetaron 4 clases por cada imagen. Cada clase superó las 1800 etiquetas para un total de casi 20700 etiquetas en todo el conjunto de datos.

²<https://roboflow.com/>

Tabla 3.2: Código de colores para las etiquetas del dataset de imágenes.

Nombre Clase	Color	Etiquetas
ALKASELTZER	Negro	2127
ASPIRINA_100	Verde	2079
BUSCAPINA	Fucsia	2149
DOLEX_GRIPA	Azul oscuro	2037
GAVISCON	Azul	2049
IBUFLASH_MIGRAN	Amarillo	1975
IBUPROFENO_GENFAR	Rojo	2182
LEVONORGESTREL	Morado	2076
NORAVR_GARGANTA	Cian	1865
VITAC_MK	Naranja	2024

Tras el etiquetado, se logró un conjunto de datos relativamente balanceado como se muestra en la figura 3.5, que presenta el histograma generado por Roboflow de etiquetas por clase.



Figura 3.5: cantidad de etiquetas por cada clase. Fuente: Gráfico resultante etiquetado Roboflow

Finalmente, la figura 3.6 muestra la distribución porcentual entre la adquisición no controlada y controlada del total de etiquetas por clase. En general se evidencia que las etiquetas obtenidas por adquisición controlada no representan ni el diez por ciento del total de las etiquetas para cada clase.

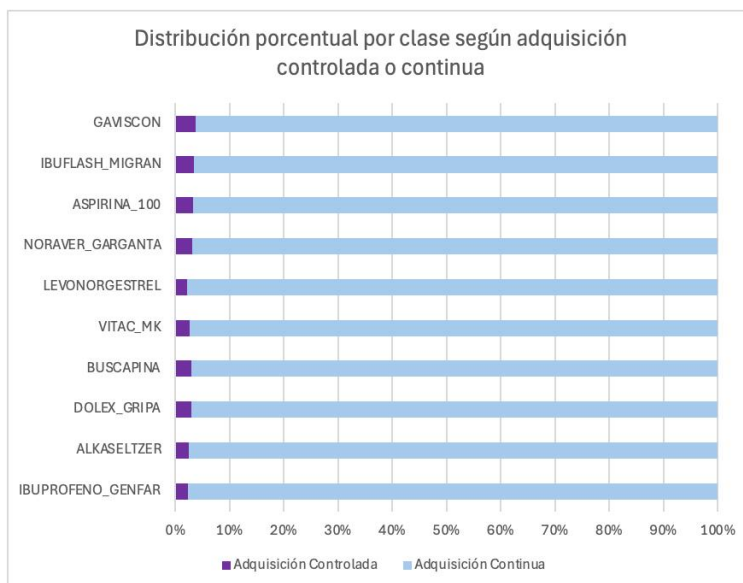


Figura 3.6: Distribución porcentual por clase según etiquetas de imágenes en adquisición controlada y no controlada. Fuente: Gráfico resultante etiquetado Roboflow

Adicionalmente, en el proceso de etiquetado se obtuvo información clave sobre la cantidad de medicamentos presentes por imagen. Aproximadamente el 20 % de las imágenes contienen un único medicamento objetivo, mientras que el 17 % presentan dos y tres anotaciones de medicamentos. El resto de las imágenes incluyen conjuntos con cuatro, cinco, seis y hasta siete medicamentos. Esta información se presenta en la figura 3.7. Esta variabilidad en la cantidad de medicamentos anotados por imagen permite capturar diferentes escenarios de uso, contribuyendo así a mejorar la precisión del modelo en situaciones reales.

El análisis de las anotaciones o etiquetas que arroja Roboflow³ reveló además características espaciales importantes evidenciadas en el mapa de calor que se muestra en la figura 3.8. Estos mapas de calor indican con que tanta frecuencia se repetía la posición del medicamento en las imágenes. En general se exhibe un equilibrio en la distribución espacial de las anotaciones de cada medicamento, lo cual facilita la identificación de ubicaciones específicas en las estanterías. Sin embargo, es necesario considerar que la distribución desigual en el conteo de elementos por imagen dificulta la obtención de una representación precisa de la distribución espacial de los medicamentos. Esta limitación en la homogeneidad de los datos plantea un desafío para la consistencia del modelo, que debe adaptarse a variaciones en la cantidad y disposición de los medicamentos para asegurar su eficacia en contextos reales.

³<https://roboflow.com/>

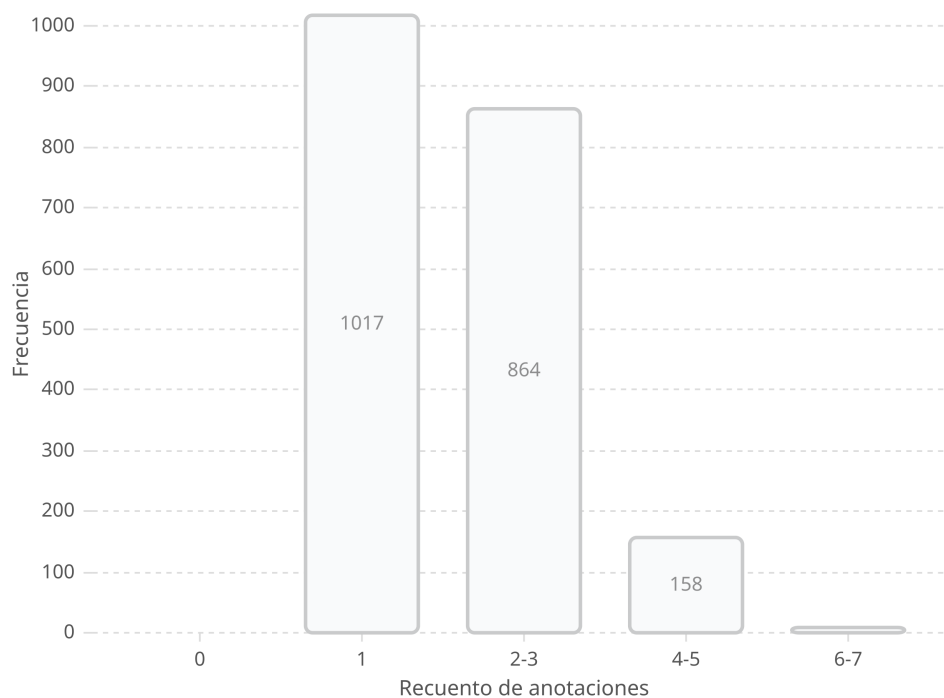


Figura 3.7: Gráfico de número de etiquetas vs. cantidad de imágenes. Fuente: Gráfico resultante etiquetado Roboflow

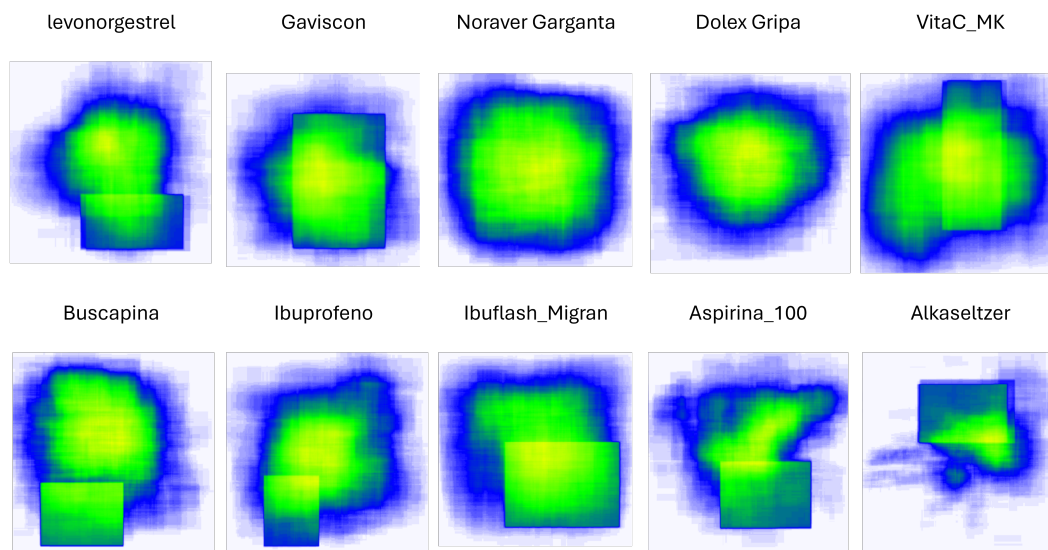


Figura 3.8: Gráfico mapa de calor, ocurrencias de la ubicación del objeto dentro de las estanterías. Fuente: Gráfico resultante etiquetado Roboflow

3.2.3. Separación del conjunto de datos de entrenamiento, validación y prueba

La preparación de datos para el flujo MLOps se basó en la subdivisión de la información en conjuntos de entrenamiento, evaluación y prueba, aprovechando la estructura consistente proporcionada por la plataforma Roboflow⁴. La plataforma suministró una subdivisión de los datos, con una distribución del 70 % para el conjunto de entrenamiento, 20 % para el conjunto de validación y 10 % para el conjunto de prueba, propuesta que es consistente con lo observado en la literatura -como se puede confirmar en la tabla 2.1 del capítulo de antecedentes -

El conjunto de entrenamiento proporciona al modelo la información necesaria para aprender los patrones de los datos, mientras que el conjunto de validación se utiliza para ajustar los hiperparámetros del modelo y evitar el sobreajuste; finalmente, el conjunto de prueba se utiliza para evaluar el rendimiento final del modelo con datos no vistos anteriormente. Al aprovechar la subdivisión de Roboflow⁴, se ahorró tiempo y esfuerzo en la preparación de datos, manteniendo al mismo tiempo una estructura consistente desde la fuente original. Esto facilitó el seguimiento del flujo MLOP y garantizó la calidad y confiabilidad de los datos utilizados.

3.2.3.1. Pre-procesamiento de imágenes

Ya que el conjunto de datos de adquisición no controlada incluye registros de múltiples fuentes, hay un gran número de registros que no presentan una estandarización en sus dimensiones, como se presenta en la figura 3.9, estos registros presentan dimensiones superiores a los 1080 píxeles y no mantienen la relación de aspecto 1:1, condición que genera problemas al intentar utilizar los registros con los modelos YOLO [40]. Por lo tanto, para garantizar la compatibilidad y consistencia de las imágenes con los modelos YOLO, se realizó un proceso de re-escalado utilizando la opción de creación de conjuntos de datos de Roboflow⁴ que de acuerdo con los requerimientos del modelo deben cumplir con una relación de aspecto 1:1 y una resolución 640 x 640 píxeles en formato jpg.

⁴<https://roboflow.com/>

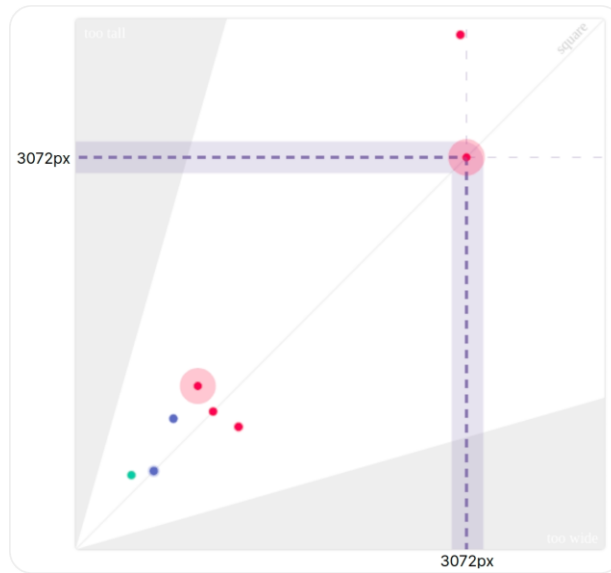


Figura 3.9: Distribución media del tamaño de las imágenes. Fuente: Gráfico resultante etiquetado Roboflow

En el caso de los datos adquiridos de manera controlada, este proceso de re-escalado no fue necesario, ya que se garantizó desde la toma de registros que todos los datos estuvieran en el formato adecuado para los modelos YOLO. Esto se logró mediante la implementación de un sistema de captura de imágenes que garantizaba que todas las imágenes tuvieran las mismas dimensiones y relación de aspecto. De esta manera, se evitó el problema de la falta de estandarización en las dimensiones y se aseguró que todos los datos estuvieran en el formato requerido.

3.2.3.2. Aumento de datos

Para enriquecer la variabilidad de los datos, se utilizó la herramienta de Roboflow⁴ para realizar un aumento de datos manteniendo las etiquetas de las imágenes originales. Para esta investigación, se emplearon operaciones de redimensionamiento con estiramiento (resize-stretch), desenfoque (blur), cizalladura (shear), rotación e inversión vertical y horizontal. Esto se hizo considerando que es probable encontrar todas estas condiciones en las imágenes que contienen las cajas de medicamentos. Al final de este paso se obtuvieron 19057 imágenes. El proceso de aumento incluyó varias transformaciones específicas:

- Inversión horizontal y vertical, lo que permitió al modelo aprender a reconocer medicamentos en diferentes orientaciones.

- Rotaciones de 90 grados, tanto en el sentido de las agujas del reloj como en sentido contrario, además de una rotación completa (180 grados), lo que facilitó al modelo el aprendizaje de la identificación de medicamentos desde distintos ángulos.
- Cortes horizontales de hasta ± 15 grados y verticales de hasta ± 10 grados, lo que ayudó al modelo a reconocer medicamentos en diversas posiciones dentro de la imagen.
- Desenfoque de hasta 1.5 píxeles, lo cual permitió que el modelo identificara medicamentos a diferentes niveles de resolución, simulando condiciones de imagen más difusas o de menor calidad.

3.2.4. Implementación del modelo YOLO

YOLO es un *framework* de aprendizaje profundo especializado en la detección de objetos, su arquitectura se basa en una serie de capas de cuello de botella CSP y un algoritmo piramidal SPP que realiza la transformación de espacio de las características y crea un único vector, el cual es recibido por capas de neuronas totalmente conectadas que se encargan de realizar la inferencia. Finalmente, aplica un algoritmo de supresión no máxima para el post procesamiento y realiza el dibujo de las cajas sobre la imagen de salida. Este algoritmo es ampliamente desarrollado y su implementación se resume en la configuración de los hiperparámetros de la arquitectura y la correcta instalación de las dependencias requeridas. Su arquitectura permite un procesamiento rápido de las imágenes, crucial para aplicaciones móviles o de baja latencia. Los entrenamientos de este estudio se realizaron en una GPU Nvidia NVIDIA L4 en la plataforma Google Colab, consumiendo las APIs de Roboflow y ultralytics.

Aunque ya se definió la versión de YOLO con la que se realizará el entrenamiento —en este caso, la versión 8—, aún es necesario seleccionar la variante del modelo preentrenado específico dentro de esa versión (n, s, m, l, x) que se deben seleccionar según la disponibilidad de recursos computacionales y el nivel de precisión requerido. Posteriormente, se identifican los hiperparámetros (batch size, learning rate, epochs, etc.) que se configurarán, con el fin de optimizar su desempeño.

En el repositorio de github de YOLOv8 se pueden encontrar distintos modelos pre entrenados en 100 épocas y las configuraciones por defecto de los hiperparámetros de entrenamiento mencionados anteriormente. En la figura 3.10 se observan los desempeños de los distintos modelos frente al conocido dataset COCO. Para este caso de estudio, se seleccionaron los modelos YOLOv8n (nano) y YOLOv8m (medium). Lo anterior, considerando que el objetivo final es desarrollar un aplicativo móvil cuyo entorno tiene recursos más bien limitados.

Modelo	tamaño (píxeles)	mAP ^{val} 50-95	Velocidad CPU ONNX (ms)	Velocidad A100 TensorRT (ms)	parámetros (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

Figura 3.10: Modelos pre-entrenados en YOLOv5. Fuente: repositorio Ultralytics

3.2.4.1. Espacio experimental uno: entrenamiento de los modelos YOLOv8

El primer paso consistió en entrenar los modelos utilizando su configuración por defecto, ya que de principio no se sabe si los hiperparámetros pre-establecidos son adecuados para el caso de estudio. En esta etapa se ejecutó el modelo YOLOv8n (nano) durante 100 épocas, lo que permitió obtener un total de 100 versiones entrenadas del modelo. Cabe resaltar que la validación se realizó de manera paralela al entrenamiento, de modo que al finalizar cada época se evaluaba el desempeño alcanzado. Entre los 100 modelos generados, se seleccionó aquel que presentó el mejor comportamiento en términos de métricas de rendimiento. Este procedimiento se replicó posteriormente con el modelo YOLOv8m (medium). Luego, se comparan los resultados de los dos mejores modelos y se selecciona uno de ellos, el mejor, de acuerdo con los siguientes criterios: Si las métricas evidenciaban una diferencia significativa en el desempeño, se seleccionaría el modelo con mejores resultados. En contraste, si la diferencia entre ambos era marginal, se elegiría el modelo nano dado que es este el que requiere un menor consumo de recursos computacionales.

Alguna información relevante de los modelos y sus hiperparametros que destacan en estos procesos de entrenamiento y validación son:

- Requisitos computacionales que para las arquitecturas nano y medium son de 8.7 y 78.7 billones de FLOPs, con 3.2 y 25.9 millones de parámetros respectivamente.
- Tamaño de la imagen de entrada (img), entre más grande la imagen se tendrá un costo computacional mayor puesto que cada pixel conformaría una característica a la entrada de la red. Ambos modelos requieren imágenes con una resolución de 640 x 640 píxeles y una relación de aspecto 1:1.
- Número de épocas (epochs) que tendrá la etapa de entrenamiento que inicialmente fue 100.

- Ubicación de los archivos (data) donde se encuentran el conjunto de datos de imágenes de entrenamiento y validación, así como también el número de clases y los nombres de las etiquetas. En este caso corresponde al repositorio de Roboflow
- Pesos (weights) pre-entrenados si se desea realizar transfer learning. Estos valores ya vienen definidos de modelo pre-entrenado utilizado.
- Cache que habilita el almacenamiento previo de las imágenes en el cache, así los entrenamientos no cargarán las imágenes en cada época y se disminuye su tiempo de ejecución.
- Tamaño del lote para cada época de entrenamiento (batch), es decir, cuantas imágenes se pasarán en cada iteración para completar una época, lo ideal sería que todas las imágenes conformaran el lote, sin embargo, el costo computacional es demasiado alto, por esto se acostumbra crear lotes o batch que sigan la relación de 2^n . Lotes más pequeños, conllevan a requerir más iteraciones por época y por ende un tiempo de entrenamiento mayor. Por otro lado, tamaños de lote más grandes incrementan el costo computacional (en este caso gráfico) que producen fallos por falta de memoria en el dispositivo de procesamiento dedicado a los gráficos (GPU).

En el primer entrenamiento que fue con la nano, el ajuste pre-determinado de los hiperparametros no permitió aprovechar las capacidades de la GPU. Por lo tanto, Con el fin de optimizar el entrenamiento, se llevó a cabo una búsqueda exhaustiva del tamaño de lote óptimo. Ya en el segundo entrenamiento que fue con el medium se pudo lograr un mejor aprovechamiento al configurar los hiperparametros preestablecido. Los resultados se presentan en la tabla 3.3.

Name	System CPU textbfAverage Perc.	GPU Total textbfMemory (GiB)	GPU Used textbfMemory (GiB)	GPU % textbfUsed Memory
10C-NANO- HiperParamsDefault	34.0 %	22.53	5.33	23.7 %
10C-MEDIUM- HiperParamsDefault	38.2 %	22.53	13.50	60.0 %

Tabla 3.3: Métricas de CPU y GPU con valores convertidos a GiB y porcentajes expresados claramente.

Se encontró que la configuración más adecuada para mantener una carga computacional estable del 60 % en una GPU NVIDIA L4 corresponde a un tamaño de lote de 108 imágenes a una resolución de 640 px a una relación de aspecto 1:1. Esta configuración consume 13.26 GB de VRAM y permite alcanzar velocidades de 154.4 mili-segundos en el encadenamiento hacia adelante (forward) y 241.2 mili-segundos en el encadenamiento hacia atrás (backward) como se evidencia en la tabla Table 3.4.

Tabla 3.4: Búsqueda exhaustiva del tamaño de lote óptimo

GFLOPs	GPU mem (GB)	Forward (ms)	Backward (ms)	Número de imágenes
8.204	0,201	30,63	159.5	1
16.41	0.308	19.08	27.13	2
32.82	0.535	20.47	26.19	4
65.63	1.017	20	27.71	8
131.3	2.007	22.87	35.73	16
886.3	13.26	154.4	241.2	108

La eficiencia de los modelos se monitorizan por medio de métricas clave como: **Precisión** que indica la proporción de objetos correctamente identificados. **Recuperación** que indica la proporción de objetos presentes en la imagen que fueron detectados correctamente. **mAP50** que corresponde a la precisión promedio en una intersección sobre el umbral de unión (IoU) del 50 %. Esta métrica evalúa la precisión del modelo en la localización de los objetos y **mAP50-95** que es la precisión promedio en umbrales de IoU entre el 50 % y el 95 %. Esta métrica proporciona una visión más completa del rendimiento del modelo en diferentes niveles de confianza. Además de las métricas mencionadas, se tiene en cuenta el tiempo de entrenamiento y el tiempo de inferencia en diferentes plataformas de despliegue. Esta información permite evaluar la eficiencia computacional de los modelos y seleccionar el más adecuado para el contexto de aplicación.

3.2.4.2. Espacio experimental dos: optimización del mejor modelo YOLO

Los resultados -que se verán en detalle en el capítulo 4- de las métricas de los modelos arrojaron resultados con diferencias equiparables, por lo tanto, se estableció el modelo nano como la alternativa a emplear en adelante para alcanzar los objetivos planteados en el desarrollo del aplicativo.

Por lo tanto, se hace un re-entrenamiento al modelo nano, esta vez ajustando los hiperparámetros partiendo de la experiencia ganada del entrenamiento del modelo medium que tuvo por poco mejor desempeño. En ese sentido, se configuró el tamaño de bloque que equilibrara la carga de trabajo al 60 % de la capacidad computacional, con esto, el modelo alcanzó un uso del 77 % del GPU como se evidencia en la tabla 3.5.

Name	System CPU textbfAverage Perc.	GPU Total textbfMemory (GiB)	GPU Used textbfMemory (GiB)	GPU % textbfUsed Memory
10C-NANO-HiperParamsBest	86.7 %	39.95	30.83	77.2 %

Tabla 3.5: Métricas de CPU y GPU con valores convertidos a GiB y porcentajes expresados claramente.

Adicional, se implementó una función de autoajuste de parámetros de Ultralytics, AdamW, una variante del optimizador Adam que incluye regularización mediante descomposición de peso (weight decay). Esta técnica penaliza los valores altos de los pesos, ayudando a prevenir el sobre-ajuste y mejorando así la capacidad de generalización del modelo durante un ciclo de 100 iteraciones. En otras palabras, el optimizador encuentra iterativa-mente la mejor configuración de hiperparametros para ajustar el modelo la fig 3.11 corresponde al resultado de la optimización con AdamW.

```
# Tune hyperparameters on data for 100 epochs
model.tune(data="/content/datasets/Recognize-medications-boxes-6/data.yaml", epochs=100, iterations=300, optimizer="AdamW")

# 10/10 iterations complete (6425.85s)
# Results saved to runs/detect/tune3
# Best fitness=0.93773 observed at iteration 3
# Best fitness metrics are {'metrics/precision(B)': 0.99262, 'metrics/recall(B)': 0.93423, 'metrics/mAP50(B)': 0.95133, 'metrics/mAP50-95(B)': 0.93621, 'val/box_loss': 0.30765, 'val/cls_loss': 0.30765}
# Best fitness model is runs/detect/train307
# Best fitness hyperparameters are printed below.

# lr0: 0.00025
# lr1: 0.01025
# momentum: 0.87414
# weight_decay: 0.00052
# warmup_epochs: 3.0
# warmup_momentum: 0.00581
# box: 7.07983
# cls: 0.456
# dfl: 1.36741
# hsv_h: 0.01342
# hsv_s: 0.6055
# hsv_v: 0.3556
# degrees: 0.0
# translate: 0.11443
# scale: 0.497
# shear: 0.0
# perspective: 0.0
# flipud: 0.0
# fliplr: 0.50065
# bgr: 0.0
# mosaic: 0.00471
# mixup: 0.0
# copy_paste: 0.0
```

Figura 3.11: captura de pantalla código de programación. Fuente: elaboración propia

Los hiperparametros arrojados por la optimización son los que se utilizan para el ajuste final del entrenamiento del modelo. La figura 3.11 presenta la captura de pantalla del modelo entrenando con dichos parámetros. finalmente, el modelo nano ajustado se ensaya con el dataset de prueba. En lo que sigue, se evalúa si el modelo ya está listo para ser exportado a un dispositivo de borde. Para eso, se realizaron pruebas de rendimiento utilizando la herramienta de referencia proporcionada por Ultralytics. Estas pruebas se llevaron a cabo en dos plataformas: CPU y GPU, con un enfoque en la inferencia, es decir, en la capacidad del modelo para procesar nuevas imágenes y generar predicciones de manera eficiente. Confirmada su disposición de exportación, se procede a compilar el modelo en TensorFlow Lite para su ejecución en dispositivos móviles. Los resultados de lo aquí dicho se muestran en el capítulo 4.

```

# results = model.train(
#     data="/content/datasets/Recognize-medications-boxes-5/data.yaml",
#     epochs=400,
#     batch=-1,
#     imgsz=640,
#     device=0,
#     lr0= 0.00825,
#     lrf= 0.01025,
#     momentum= 0.87414,
#     weight_decay= 0.00052,
#     warmup_epochs= 3.0,
#     warmup_momentum= 0.80581,
#     box= 7.07983,
#     cls= 0.456,
#     dfl= 1.36741,
#     hsv_h= 0.01342,
#     hsv_s= 0.6055,
#     hsv_v= 0.3556,
#     degrees= 0.0,
#     translate= 0.11443,
#     scale= 0.497,
#     shear= 0.0,
#     perspective= 0.0,
#     flipud= 0.0,
#     fliplr= 0.50865,
#     bgr= 0.0,
#     mosaic= 0.80471,
#     mixup= 0.0,
#     copy_paste= 0.0
# )

```

Figura 3.12: captura de pantalla código de programación. Fuente: elaboración propia

3.3. Desarrollo del aplicativo móvil

A continuación, se presenta una descripción detallada del diseño del aplicativo móvil que constituye una etapa fundamental en el desarrollo de la solución, ya que establece las bases técnicas y conceptuales que guiarán su implementación. Aquí se presenta la definición de los requerimientos funcionales y no funcionales, el diseño conceptual que describe la interacción entre los distintos componentes del sistema, así como las historias de usuario que permiten vincular las necesidades de los usuarios con la lógica del aplicativo. Posteriormente, se describe el desarrollo o implementación del mismo donde se explica el proceso para exportar el modelo entrenado a un aplicativo android y la programación como tal del resto de funcionalidades.

3.3.1. Requerimientos funcionales y no funcionales

En el desarrollo del aplicativo móvil se definieron los requerimientos funcionales y no funcionales que sirven de guía para definir tanto el diseño como la implementación del mismo. Los requerimientos funcionales especifican las acciones que el sistema debe realizar, es decir, las funcionalidades que aseguran que el aplicativo cumpla con su propósito de reconocer medicamentos. Por su parte, los requerimientos no funcionales establecen las condiciones y restricciones bajo las cuales el sistema debe operar,

incluyendo aspectos de software y de hardware. A continuación, se presenta en la tabla 3.6 y 3.7 los requerimientos funcionales y no funcionales respectivamente definidos para el aplicativo.

Tabla 3.6: Requerimientos funcionales del aplicativo

REF	FUNCIONES	TIPO
RF1	El aplicativo debe ser ejecutado desde un dispositivo móvil Android	Compatibilidad
RF2	El aplicativo debe permitir la captura de imágenes con la cámara del dispositivo	Captura
RF3	El aplicativo debe convertir el formato y ajustar la orientación de la imagen según las necesidades del modelo implementado	Pre-procesado
RF4	El aplicativo debe re-dimensionar el tamaño de la imagen según las necesidades del modelo implementado	Pre-procesado
RF5	El aplicativo debe permitir que se ejecute el modelo entrenado de inferencia	Inferencia
RF6	El aplicativo debe definir las coordenadas correctas donde se ubicaran las cajas de contorno de acuerdo con los resultados de inferencia	Post-procesado
RF7	El aplicativo debe definir la etiqueta de acuerdo con los resultados de inferencia	Post-procesado
RF8	El aplicativo debe renderizar la capa que muestra las cajas de contorno y las etiquetas	Visualización
RF9	El aplicativo debe mostrar el resumen con lista de nombres de medicamentos reconocidos y sus cantidades	Reporte

Los requerimientos funcionales se organizaron según el tipo de requisito dentro de las etapas de procesamiento de visión artificial, que incluyen la compatibilidad de la aplicación con dispositivos Android, la captura y preprocesamiento de imágenes, la ejecución del modelo de inferencia, el postprocesamiento de resultados y la visualización de etiquetas, hasta la generación de reportes con la lista y cantidad de medicamentos detectados.

Los requerimientos no funcionales definidos para el aplicativo se relacionan principalmente con los recursos tecnológicos necesarios para su implementación y ejecución. En cuanto al hardware, se establece la necesidad de contar con un dispositivo móvil Android que disponga de cámara integrada para la captura de imágenes. Respecto al software, se especifica el uso del sistema operativo Android como plataforma base de ejecución, junto con tecnologías clave como TensorFlow Lite, que permite la inferencia del modelo en dispositivos móviles que a su vez ejecuta el modelo YOLOv5, empleado para la detección y clasificación de medicamentos.

Tabla 3.7: Requerimientos no funcionales del aplicativo

REF	FUNCIONES	TIPO
RNF1	dispositivo móvil Android versión 13	Hardware
RNF2	dispositivo móvil Android con cámara	Hardware
RNF3	Kotlin	Software
RNF4	Framework Tensorflow Lite	Software
RNF5	YoloV8	Software
RNF6	Lenguaje Python 3.10 para el entrenamiento del modelo	Software

Finalmente, se señala el uso del lenguaje Python en su versión 3.10 para el entrenamiento del modelo y Kotlin⁵ como lenguaje de programación para aplicaciones en Android

3.3.2. Diagrama conceptual

A partir de los requerimientos funcionales mencionados en la tabla 3.6 se ha desarrollado un diagrama conceptual que se presenta en la figura 3.13. Este diagrama muestra como los distintos conceptos se relacionan y asocian entre sí, cada flecha se emplea un verbo que describe la acción o el efecto de un concepto sobre otro.

El sistema inicia cuando el usuario abre la aplicación móvil Android y la inicializa. A partir de allí, el usuario puede capturar imágenes de cajas de medicamentos mediante la cámara del dispositivo. Una vez obtenida la imagen, el aplicativo ejecuta la fase de pre-procesamiento, donde se ajusta el formato, orientación y tamaño de la imagen para que sea compatible con los requerimientos del modelo de inteligencia artificial.

En la etapa de inferencia, el sistema utiliza el framework TensorFlow Lite para cargar y ejecutar el modelo YOLO previamente entrenado. Este modelo analiza la imagen y detecta los medicamentos presentes. Posteriormente, en la fase de post-procesamiento, se calculan las coordenadas de las cajas de contorno que enmarcan los objetos identificados y se asocian a su respectiva clase.

Después, el sistema procede renderizar la caja de contorno y a etiquetar cada medicamento reconocido, asignándole el nombre correspondiente y su posición en la imagen. Finalmente, la aplicación realiza un resumen con la lista de medicamentos reconocidos y sus cantidades.

⁵<https://kotlinlang.org/>

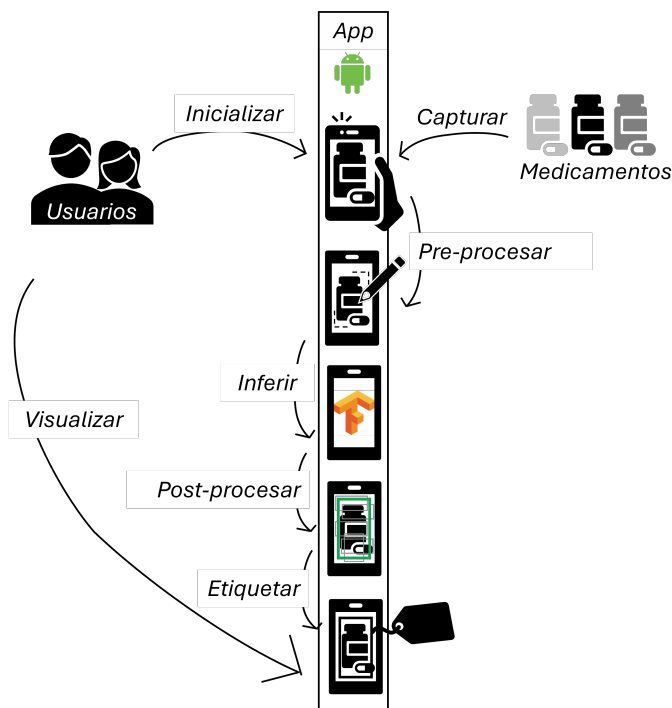


Figura 3.13: Diagrama conceptual. Fuente: elaboración propia

3.3.3. Historias de usuarios

Para el desarrollo del aplicativo se utilizó la metodología ágil Scrum, dado que esta permite gestionar el proyecto de manera iterativa e incremental, facilitando la adaptación a cambios y promoviendo la entrega temprana de resultados funcionales. Esta elección responde a la necesidad de mantener un proceso flexible y orientado a las prioridades del usuario, asegurando que cada avance del sistema aporte valor real al objetivo planteado [41].

En este marco, se emplearon historias de usuario como herramienta principal para la definición de funcionalidades. Estas historias permiten expresar de forma simple y clara, desde la perspectiva del usuario, las acciones que el sistema debe realizar y los beneficios esperados. Por lo tanto, a continuación, en la tabla 3.8 se presentan las historias de usuario definidas para el aplicativo, acompañadas de sus respectivos escenarios y criterios de aceptación, que sirven como guía para validar el cumplimiento de los requerimientos funcionales establecidos.

Tabla 3.8: Historias de usuario y criterios de aceptación

REF	Historia de Usuario	Escenarios y Criterios de Aceptación
RF1	Como usuario de farmacia, necesito que la aplicación se ejecute en un dispositivo Android con cámara	1. Ejecución en Android: En caso que el usuario tenga un dispositivo Android 10 o superior, cuando abre la aplicación, esta debe ejecutarse sin errores. 2. Compatibilidad mínima: En caso que el usuario tenga un dispositivo con 3 GB de RAM o más, cuando abre la aplicación. 3. Incompatibilidad controlada: En caso que el usuario tenga un dispositivo con versión Android no soportada, cuando intente instalar la aplicación, el sistema debe notificar la incompatibilidad.
RF2	Como usuario de farmacia, necesito capturar imágenes con la cámara para que el sistema pueda reconocer los medicamentos.	1. Captura correcta: En caso que el usuario habilite la cámara, cuando toma la foto, la imagen se guarda y se envía al módulo de preprocesamiento. 2. Permisos de cámara: En caso que el usuario no haya dado permiso, cuando intente tomar una foto, el sistema debe solicitar acceso a la cámara. 3. Reintento de captura: En caso que el usuario desee mejorar la foto, cuando presione “reintentar”, la aplicación debe permitir nueva captura.
RF3	Como sistema de reconocimiento, necesito convertir el formato y ajustar la orientación de la imagen para que el modelo pueda procesarla correctamente.	1. Formato compatible: En caso que la imagen sea JPG o PNG, cuando ingrese al sistema, debe convertirse al formato esperado por el modelo. 2. Orientación corregida: En caso que la imagen se haya tomado en vertical u horizontal, cuando se procese, debe ajustarse automáticamente. 3. Error controlado: En caso que la imagen esté corrupta, cuando se intente procesar, debe solicitar nueva captura.

Continúa en la siguiente página

Cuadro 3.8 – continuación de la página anterior

REF	Historia de Usuario	Escenarios y Criterios de Aceptación
RF4	Como sistema de reconocimiento, necesito redimensionar la imagen capturada para que cumpla con las dimensiones requeridas por el modelo de inferencia.	1. Tamaño esperado: En caso que la imagen sea de alta resolución, cuando pase al preprocesamiento, debe ajustarse al tamaño definido. 2. Preservación de calidad: En caso que se redimensione, cuando se guarde, la calidad debe mantenerse sin pixelado severo. 3. Optimización de peso: En caso que supere 5 MB, cuando se procese, debe comprimirse manteniendo calidad mínima.
RF5	Como sistema de inventario, necesito ejecutar el modelo YOLO entrenado para reconocer medicamentos en la imagen capturada.	1. Inferencia exitosa: En caso que el usuario haya capturado una imagen válida, cuando el modelo procese, debe obtener predicciones con confianza > 0.5. 2. Error en ejecución: En caso que falle la carga del modelo, cuando se intente ejecutar, debe mostrar un mensaje de error.
RF6	Como sistema de postprocesamiento, necesito calcular las coordenadas de las cajas de contorno para ubicar correctamente los medicamentos detectados.	1. Cajas válidas: En caso que se detecte un medicamento, cuando se calcule, las coordenadas deben estar dentro de la imagen. 2. Múltiples objetos: En caso que se detecten varios, cuando se dibujen, deben calcularse coordenadas independientes. 3. Supresión no máxima: En caso que haya detecciones superpuestas, cuando se postprocese, debe conservarse la de mayor confianza.
RF7	Como sistema de postprocesamiento, necesito asignar una etiqueta a cada detección para identificar el nombre del medicamento.	1. Etiqueta correcta: En caso que se detecte un objeto, cuando se asigne clase, debe corresponder al medicamento. 2. Varias detecciones: En caso que se detecten varios medicamentos, cuando se asignen etiquetas, cada uno debe tener su nombre. 3. Confianza baja: En caso que la predicción tenga confianza < 0.5, cuando se intente etiquetar, no debe asignarse etiqueta.

Continúa en la siguiente página

Cuadro 3.8 – continuación de la página anterior

REF	Historia de Usuario	Escenarios y Criterios de Aceptación
RF8	Como usuario de farmacia, necesito visualizar las cajas y etiquetas superpuestas en la imagen capturada para verificar los medicamentos detectados.	1. Renderizado de cajas: En caso que se detecten medicamentos, cuando se muestre la imagen, deben aparecer cajas de contorno. 2. Visualización de etiquetas: En caso que cada caja tenga nombre, cuando se renderice, debe mostrarse junto a la caja. 3. Imagen limpia: En caso que no se detecten objetos, cuando se muestre, la imagen debe estar sin cajas ni etiquetas.
RF9	Como usuario de farmacia, necesito visualizar un resumen con la lista de medicamentos detectados y sus cantidades para facilitar el inventario.	1. Lista de medicamentos: En caso que se detecten medicamentos, cuando se genere el reporte, debe incluir los nombres. 2. Cantidad de medicamentos: En caso que un medicamento se repita, cuando se genere, debe mostrarse el conteo. 3. Reporte vacío: En caso que no se detecte ningún medicamento, cuando se genere, debe mostrarse mensaje “No se detectaron medicamentos”.

3.3.4. Implementación

Una vez completado el diseño del aplicativo, se aborda la implementación, la cual se estructura en dos partes: la exportación del modelo dentro del aplicativo y el desarrollo del resto de funcionalidades del aplicativo.

3.3.4.1. Exportación del modelo

Una vez se encontró el mejor modelo con el comportamiento mas conveniente y habiendo comprobado el rendimiento de este mismo tanto en CPU como en GPU se inicia el proceso de exportación al ambiente móvil. Esto se hace con la herramienta de TensorFlow Lite de acuerdo con el flujo MLOps mencionado anteriormente. TensorFlow Lite permite optimizar el modelo Ultralytics YOLOv8 en aplicaciones basadas en dispositivos de borde, como teléfonos inteligentes y tabletas, es ampliamente compatible con una amplia gama de plataformas, incluyendo Linux embebido, Android, iOS y MCU. Por otro lado, al procesar los datos localmente en el dispositivo, TensorFlow Lite reduce significativamente la latencia en comparación con

las soluciones basadas en la nube. Esto se traduce en una experiencia de usuario más fluida y receptiva. Además, la exportación a TensorFlow Lite permite procesar los datos de imágenes de medicamentos en el dispositivo, sin necesidad de transmitirlos a un servidor remoto. Esto mejora la privacidad del usuario al minimizar la cantidad de datos personales que se comparten y lo más importante es que TensorFlow Lite optimiza el modelo YOLOv8, reduciendo su tamaño y haciéndolo más eficiente para su implementación en dispositivos móviles con recursos limitados.

El proceso de conversión a TensorFlow Lite se simplifica gracias al uso del modelo YOLOv8 proporcionado por Ultralytics. Este modelo incluye un atributo de exportación y parámetros para especificar el formato de salida deseado, lo que facilita la conversión a TensorFlow Lite con solo unas pocas líneas de código. Una vez exportados, los modelos se almacenan en un repositorio en la nube, desde donde pueden descargarse e implementarse en la aplicación móvil para la detección de medicamentos. Esta estrategia centralizada facilita la distribución y actualización de los modelos a medida que se mejora su rendimiento o se amplía su funcionalidad.

3.3.4.2. Desarrollo del aplicativo

Una vez cargado el modelo dentro del ambiente móvil, se procede a programar el resto de funcionalidades relacionadas con las entradas y las salidas del modelo que finalmente permitan al usuario visualizar las imágenes reconocidas. A continuación, se presentan las pantallas del aplicativo (ver figura 3.14 y 3.15) las cuales se explicarán con detalle en adelante. Cabe recordar que estas funcionalidades fueron desarrolladas con el lenguaje de programación Kotlin⁶.

La pantalla A de la figura 3.14 corresponde a la interfaz de inicio o bienvenida, en la cual se presentan el nombre de la aplicación, el logotipo y la versión. Su función principal es ofrecer una introducción visual al usuario al momento de abrir el aplicativo desde el menú de aplicaciones del dispositivo móvil, mientras se cargan los recursos y funcionalidades necesarias para su ejecución.

La pantalla B corresponde al menú principal, el cual organiza y ofrece al usuario acceso a las funciones esenciales de la aplicación mediante botones: tomar foto, galería, inferencia local, inferencia remota, configuración y ayuda. Cada una de estas opciones redirige a las diferentes pantallas que componen el sistema.

La pantalla C corresponde a la interfaz de captura de imágenes, donde el usuario emplea la cámara del dispositivo para registrar fotografías de los medicamentos en caja mediante el botón inferior de color rojo. Esta vista también permite visualizar el historial

⁶<https://kotlinlang.org/>

de imágenes capturadas en la sesión y el número total de fotografías almacenadas en la galería.

La pantalla D corresponde a la galería de imágenes. Desde esta vista, el usuario puede seleccionar y revisar las fotografías previamente tomadas. La subvista D.1 permite verificar el reconocimiento de una imagen individual; la subvista D.2 posibilita la selección múltiple de imágenes; y la subvista D.3 de la figura 3.15 habilita la inferencia conjunta sobre las imágenes seleccionadas. Adicionalmente, mediante la opción D.3.1, el usuario puede generar un reporte en formato de mensaje de texto que se envía directamente a WhatsApp con los resultados obtenidos.

La pantalla E corresponde al proceso de inferencia local. En esta vista, el modelo ejecutado en el dispositivo analiza la imagen seleccionada, dibuja las cajas de contorno sobre los medicamentos identificados y asigna sus etiquetas correspondientes. En la parte inferior se presenta un resumen con los nombres de los medicamentos reconocidos y la cantidad de unidades detectadas.

La pantalla F corresponde a la opción de inferencia remota, que conecta al usuario con una página web donde se encuentra cargado el modelo, de modo que la detección de imágenes se realice en un entorno externo al aplicativo.

Finalmente, la pantalla G corresponde al módulo de configuración, que brinda la posibilidad de ajustar parámetros técnicos del modelo de reconocimiento, como el confidence threshold y el IoU threshold, otorgando mayor control sobre la precisión de los resultados. A su vez, el aplicativo incluye una sección de ayuda, donde se presentan recomendaciones prácticas orientadas a mejorar la calidad de las capturas, tales como asegurar una iluminación adecuada y la nitidez de la imagen.

La adquisición de datos se registra a través de la funcionalidad de captura de datos directamente desde la cámara del dispositivo. El proceso de inferencia que se lleva a cabo en el aplicativo móvil consta de tres etapas clave: la carga del modelo, la ejecución local y análisis, y la visualización de resultados. Primero, la aplicación carga el modelo de aprendizaje automático más reciente, segundo, el modelo se ejecuta directamente en el dispositivo móvil, procesando la imagen capturada por el usuario y extrayendo características relevantes para determinar la identidad del medicamento. Tercero, la aplicación presenta el resultado de la clasificación de manera visual y clara, superponiendo la imagen original con un contorno que delimita el empaque del medicamento. Además, se muestra una etiqueta con el nombre del medicamento identificado

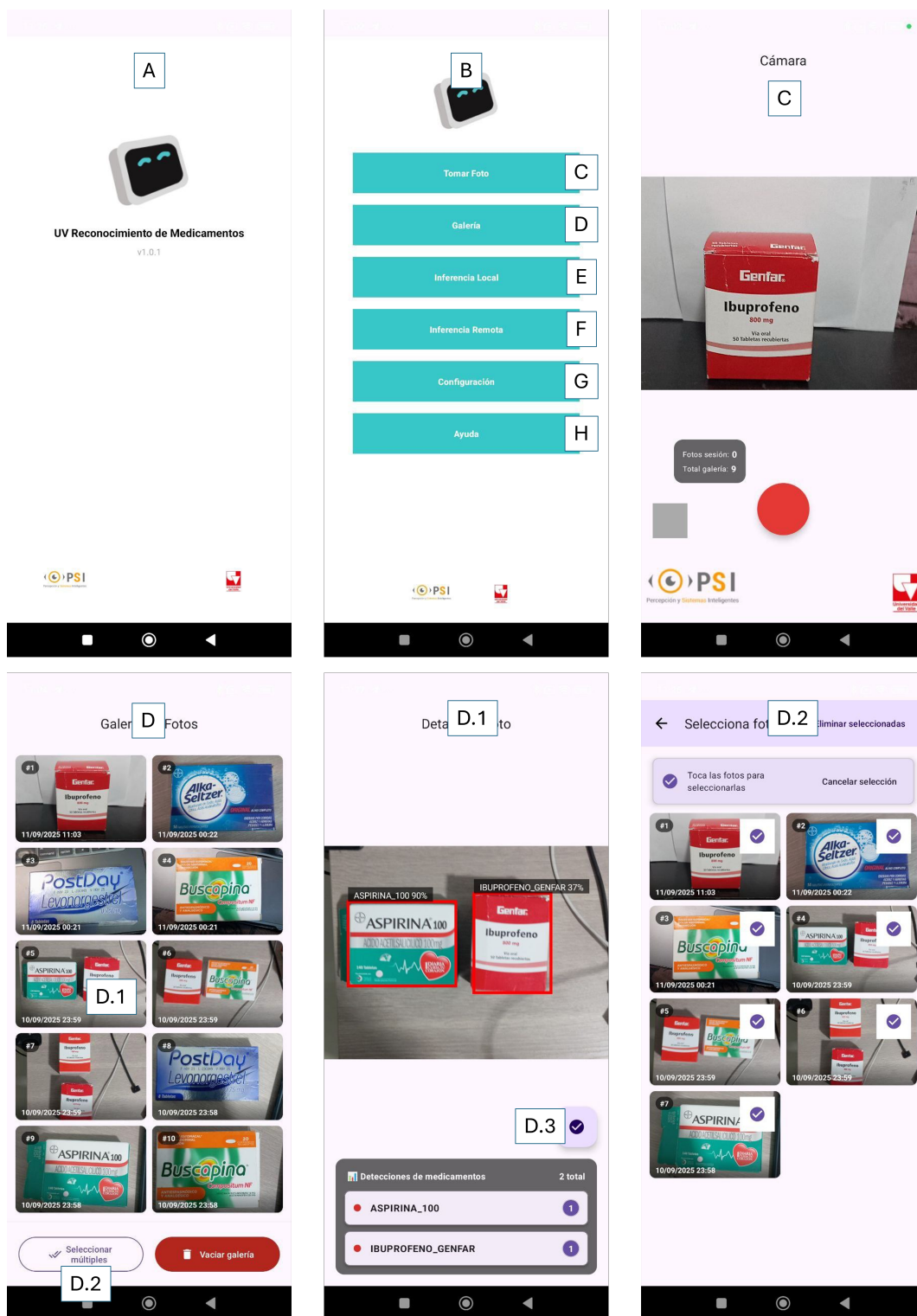


Figura 3.14: Captura de pantalla de las vistas de la aplicación móvil en teléfono Android.
Fuente: elaboración propia según diseño de App

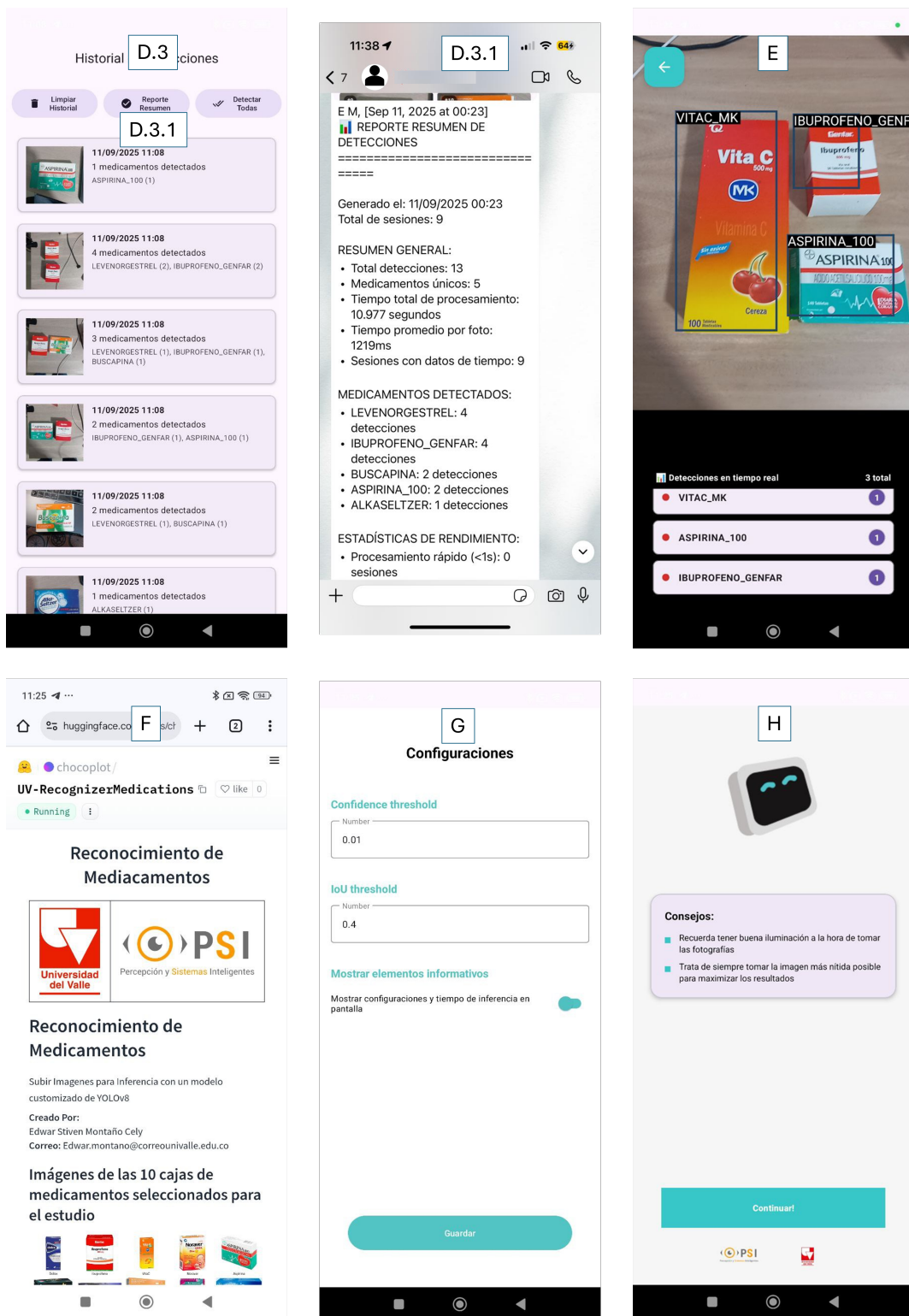


Figura 3.15: Captura de pantalla de las vistas de la aplicación móvil en teléfono Android.
Fuente: elaboración propia según diseño de App

4. PRUEBAS Y RESULTADOS

4.1. Introducción

A continuación, se describen los espacios experimentales y sus respectivos resultados evaluando las métricas que mejor describen el desempeño de los modelos, con base en ello se determina el modelo que se exportará al aplicativo. De todas formas, los resultados obtenidos demuestran que tanto YOLOv8n como YOLOv8m presentan un buen rendimiento para el reconocimiento de medicamentos. Las métricas de evaluación utilizadas, como la precisión, la recuperación, mAP50 y mAP50-95, indican que los modelos son capaces de identificar correctamente los medicamentos en las imágenes, localizarlos con precisión y asignar un nivel de confianza adecuado a las detecciones.

4.1.1. Espacio experimental uno y dos: evaluación de los modelos YOLOv8 y optimización del mejor modelo

El entrenamiento llevado a cabo durante 100 épocas permitió alcanzar una precisión aproximada del 94 % con el modelo YOLOv8n y del 94.5 % con el modelo YOLOv8m (ver figura 4.1). Estos resultados de desempeño se corroboran a través de la sensibilidad, que en promedio fue de 97.2 % entre ambos modelos evaluados (ver figura 4.2). Los resultados preliminares del rendimiento de los modelos YOLOv8n y YOLOv8m mostraron que la diferencia en sus métricas de desempeño fue mínima, con un promedio de solo 0.4 %.

También se evaluaron las métricas mAP50 y mAP50-95. Los resultados mostraron un mAP50 promedio de 97.5 % en ambos modelos evaluados (ver figura 4.3), mientras que el modelo YOLOv8n alcanzó un mAP50-95 de 86 % y el YOLOv8m un 87 % (ver figura 4.4). Estos resultados indican que los modelos logran identificar correctamente los medicamentos en las imágenes, incluso en condiciones de diversa complejidad. Además, la elevada capacidad de recuperación del modelo permite detectar la mayoría de los medicamentos presentes en cada imagen. Los valores de mAP50 y mAP50-95 confirman la capacidad del modelo para generalizar bien en distintos escenarios y condiciones de iluminación.

Ya que son rendimientos equiparables, se definió el modelo nano como la alternativa para usar en el aplicativo móvil ya que requiere menores demandas computacionales. Con base en la experiencia ganada se aplica un proceso de

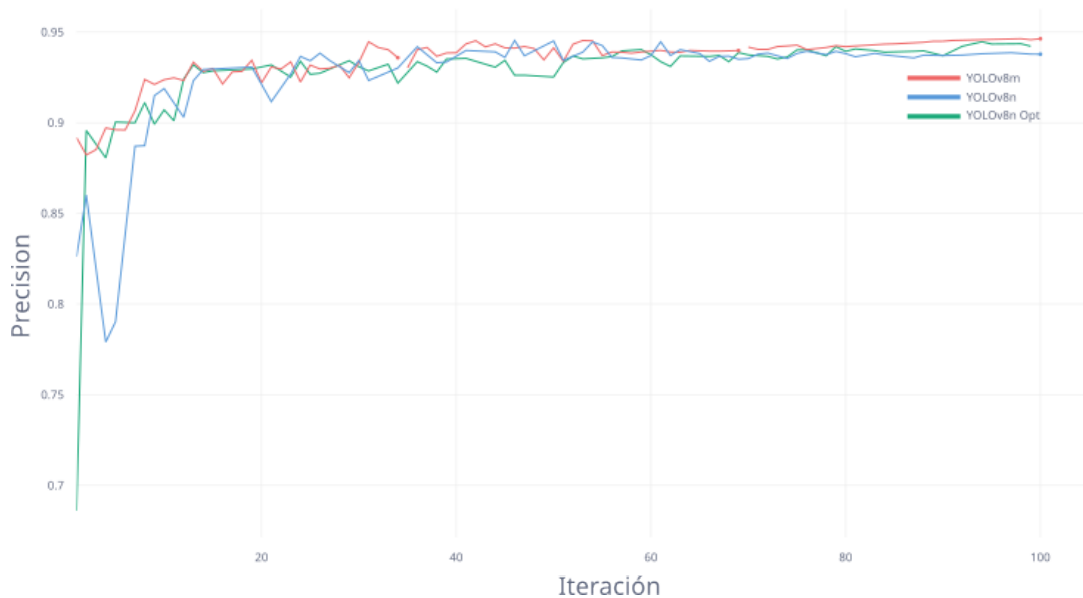


Figura 4.1: Precisión de los modelos YOLO sobre un conjunto de datos nunca antes visto. Fuente: Gráfico resultante de la evaluación del modelo en CometML

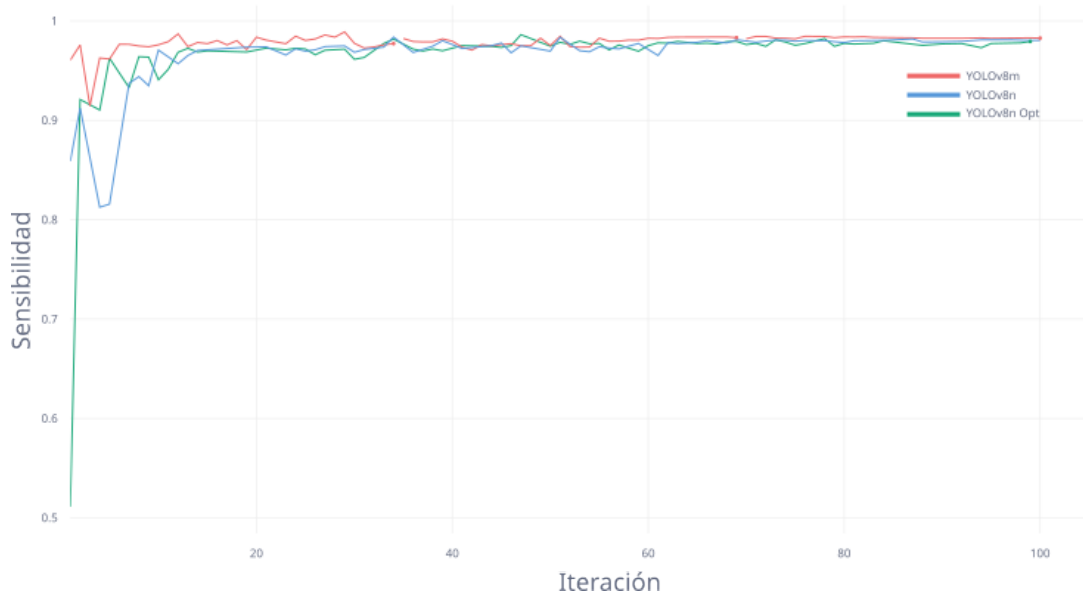


Figura 4.2: Sensibilidad de los modelos YOLO sobre un conjunto de datos nunca antes visto. Fuente: Gráfico resultante de la evaluación del modelo en CometML

optimización adicional al modelo nano en un segundo entreno para lograr un equilibrio adecuado entre la precisión en la localización y en la clasificación de los objetos. Los resultados de precisión, sensibilidad, mAP50 y mAP50-95 de este modelo también se grafican en las figuras 4.1, 4.2, 4.3, 4.4 respectivamente.

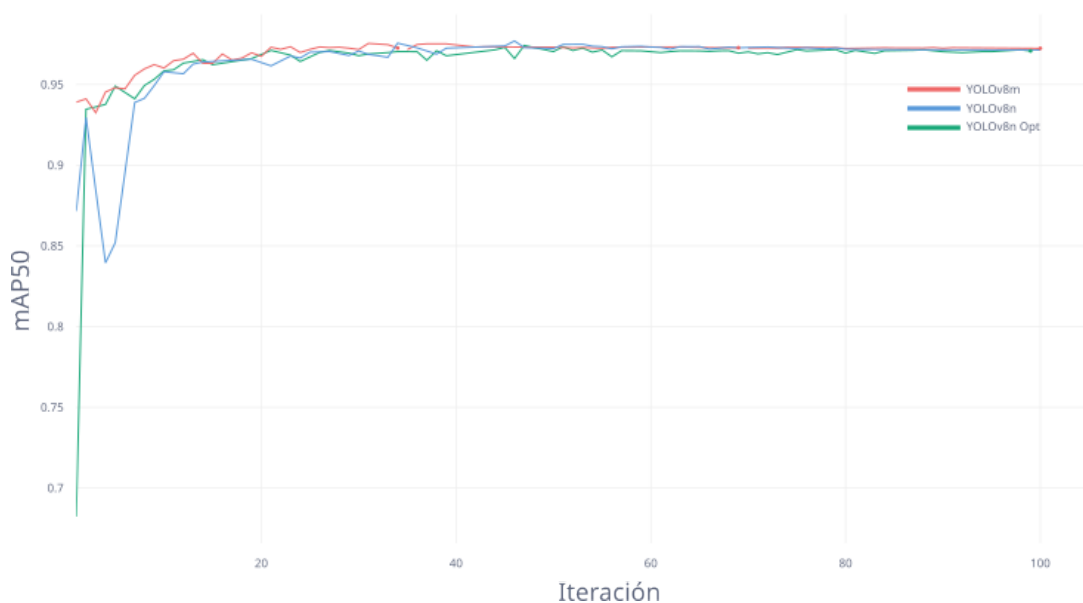


Figura 4.3: mAP50 de los modelos YOLO sobre un conjunto de datos nunca antes visito. Fuente: Gráfico resultante de la evaluación del modelo en CometML

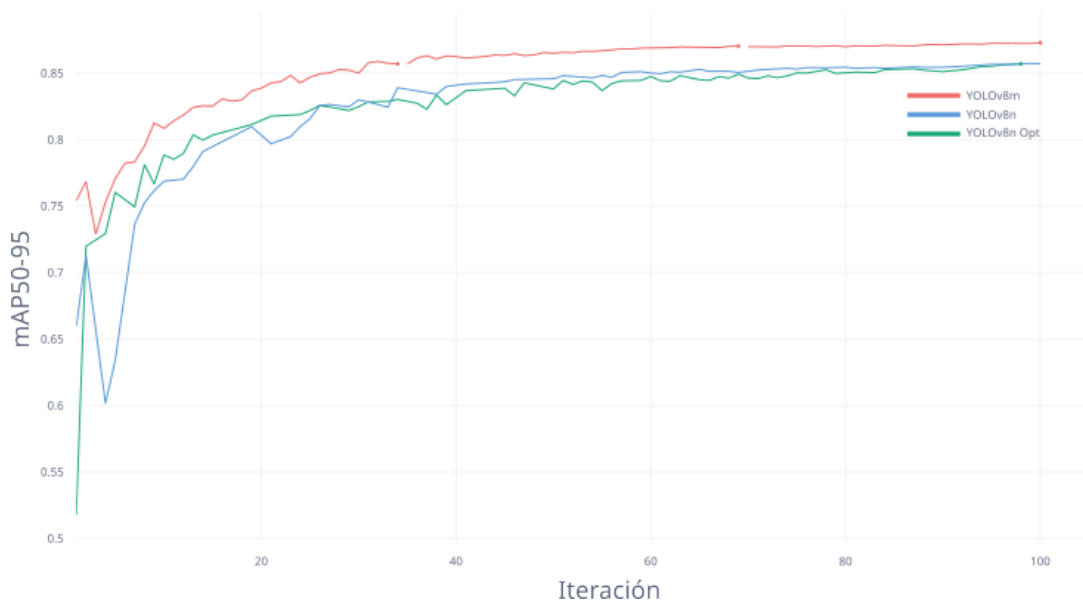


Figura 4.4: mAP50-95 de los modelos YOLO sobre un conjunto de datos nunca antes visito. Fuente: Gráfico resultante de la evaluación del modelo en CometML

Gracias a estos ajustes, el modelo mostró un rendimiento robusto en el reconocimiento y clasificación de medicamentos. Es importante destacar que todos los resultados obtenidos se lograron mediante la evaluación en un conjunto de datos completamente nuevo para los modelos, lo cual indica una sólida capacidad de

generalización y alta precisión en la detección de medicamentos bajo diversas condiciones. Finalmente en la figura 4.5 se muestran las graficas de validacion y test del entrenamiento del mejor modelo.

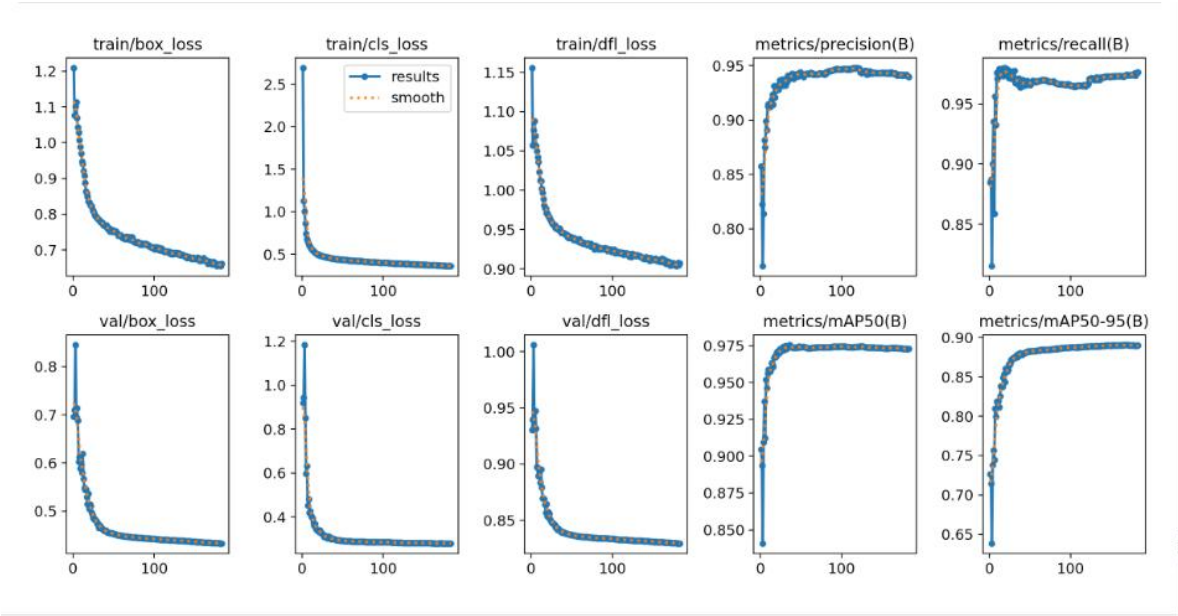


Figura 4.5: Gráficas de validación y test del entrenamiento del mejor modelo

4.1.2. Exportación modelo Yolov8n al ambiente móvil

Las pruebas de inferencia en GPU arrojaron resultados prometedores, que se presentan en la tabla 4.1. Los resultados destacan el rendimiento superior del modelo sobre la compilación TorchScript. En promedio, el modelo alcanzó un mAP50-95 del 85.14%, lo que refleja una alta precisión en la detección y clasificación de medicamentos. Además, la velocidad de inferencia fue de 3.97 milisegundos por imagen, lo cual permite un procesamiento rápido y eficiente. Esto se traduce en una capacidad de inferencia de 251 imágenes por segundo, lo que demuestra la escalabilidad del modelo para manejar grandes volúmenes de datos en tiempo real. Estos resultados validan al modelo YOLOv8n como una solución robusta y eficiente para aplicaciones prácticas en entornos de alta demanda, como el monitoreo en farmacias o la gestión de inventarios en tiempo real.

Tabla 4.1: Evaluación GPU

Plataforma	Disponibilidad	Dimencion (MB)	MPA50-95 (B)	Tiempo de inferencia (ms/im)	FPS
PyTorch	SI	6.0	0.8580	8.35	119
TorchScript	SI	11.9	0.8514	3.97	251
ONNX	SI	11.7	0.8513	9.03	110

En lo que respecta a las bibliotecas de compilación como ONNX, el modelo YOLOv8n también mostró un rendimiento satisfactorio, alcanzando un mAP50-95 de 85.13 %. Sin embargo, la velocidad de inferencia en esta plataforma fue de 9.03 milisegundos por imagen, lo cual se traduce en una capacidad de procesamiento de aproximadamente 110 imágenes por segundo. Esta diferencia de rendimiento, en comparación con otros entornos, se debe principalmente a la sobrecarga de procesamiento adicional asociada a la compilación del modelo en ONNX. A pesar de esta reducción en la velocidad de inferencia, el uso de ONNX sigue siendo una opción viable para entornos que prioricen la portabilidad y la compatibilidad en múltiples plataformas, aunque con un ligero compromiso en el tiempo de respuesta. Esto convierte a ONNX en una alternativa adecuada para aplicaciones en sistemas donde la interoperabilidad entre distintas bibliotecas y dispositivos es crucial.

Las pruebas realizadas en CPU (ver tabla 4.2), utilizando compilaciones en TorchScript y TensorFlow Lite, alcanzaron un mAP50-95 promedio de 85.13 %, lo cual refleja una alta precisión en la detección y clasificación de medicamentos. Sin embargo, la velocidad de inferencia fue considerablemente menor en comparación con los resultados obtenidos en GPU. En particular, TorchScript logró una velocidad de inferencia de 56.39 milisegundos por imagen, lo que equivale a una capacidad de procesamiento de aproximadamente 17 imágenes por segundo, mientras que TensorFlow Lite registró un tiempo de inferencia de 155.42 milisegundos por imagen, procesando apenas 6 imágenes por segundo.

Tabla 4.2: Evaluación CPU

Plataforma	Disponibilidad	Dimensión (MB)	MPA50-95 (B)	Tiempo de inferencia (ms/im)	FPS
PyTorch	SI	6.0	0.8580	70.87	14
TorchScript	SI	11.9	0.8513	56.39	17
ONNX	SI	11.7	0.8513	66.07	15
TensorFlow Lite	SI	11.7	0.8513	155.42	6

El uso de la biblioteca de compilación ONNX en CPU también impactó en la velocidad de inferencia, con un tiempo de 66.07 milisegundos por imagen, lo cual permite procesar alrededor de 15 imágenes por segundo. A pesar de esta disminución en la velocidad de procesamiento, ONNX mantuvo un rendimiento consistente en cuanto a precisión, con un mAP50-95 de 85.13 %. Estos resultados sugieren que, aunque el rendimiento en CPU es limitado en términos de velocidad, las compilaciones en TorchScript, TensorFlow Lite y ONNX siguen siendo opciones válidas para aplicaciones que prioricen la precisión y la portabilidad del modelo en dispositivos sin acceso a GPU. Sin embargo, en aplicaciones que requieren una alta capacidad de procesamiento en tiempo real, el uso de GPU sigue siendo la opción preferida.

Finalmente, la optimización del modelo YOLOv8n resultó en un incremento del 0.25 % en el desempeño final del modelo. Sin embargo, es importante destacar que el tiempo de inferencia experimentó una mejora significativa, reduciéndose de 1.8

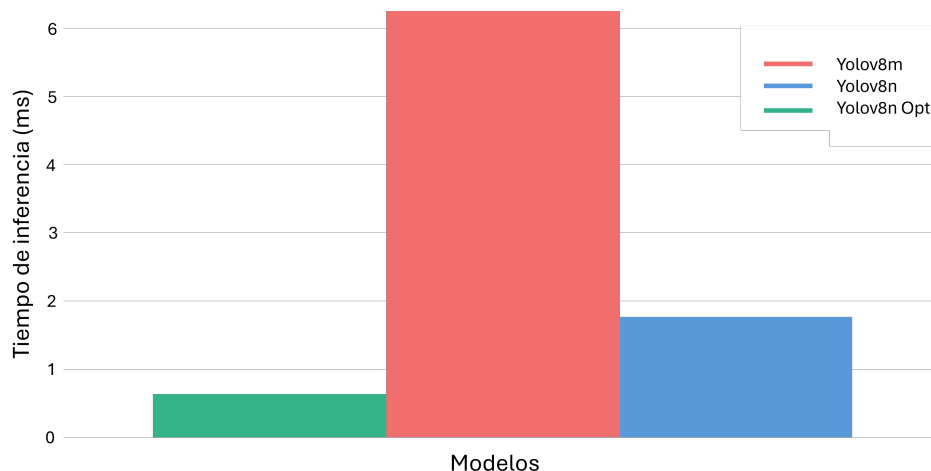


Figura 4.6: Tiempos de inferencia de los modelos YOLO. Fuente: Gráfico resultante de la evaluación del modelo en CometML

milisegundos a 0.6 milisegundos (ver figura 4.6). Esta reducción en el tiempo de inferencia representa una mejora clave para aplicaciones en tiempo real, especialmente en dispositivos con recursos limitados. Gracias a esta optimización, el modelo no solo mantiene su precisión en la detección de medicamentos, sino que también mejora su eficiencia, facilitando su implementación en contextos de uso rápido y continuo, como el monitoreo en farmacias o el reconocimiento en aplicaciones móviles de asistencia médica.

4.1.3. Pruebas de Campo

Una vez que el modelo se cargó con éxito en la aplicación y el sistema estuvo en funcionamiento, se procedió a realizar las pruebas de campo, cuyos resultados se sintetizaron posteriormente en una matriz de confusión. Dichas pruebas se llevaron a cabo en un entorno no controlado, específicamente en una droguería, y se estructuraron en seis escenarios:

- i) Captura de imágenes de un solo medicamento perteneciente a una clase, para un total de 10 fotografías.
- ii) Captura de imágenes de dos medicamentos de una misma clase, para un total de 10 fotografías.
- iii) Captura de imágenes de dos medicamentos, cada uno de una clase diferente, para un total de 45 fotografías.
- iv) Captura de imágenes de tres medicamentos, cada uno de una clase diferente, para un total de 120 fotografías.
- v) Captura de una única imagen de varios medicamentos que no pertenecen a ninguna de las clases de interés, para un total de 1 fotografía.
- vi) Captura de imágenes de uno o dos medicamentos de clases distintas acompañados por otro que no pertenece a ninguna

clase de interés (objeto extraño), con un total de 15 fotografías (10 en el primer caso y 5 en el segundo).

En total, se obtuvieron 201 imágenes directamente desde la aplicación, utilizando la opción “tomar foto”. Estas imágenes fueron almacenadas en la galería del aplicativo desde donde fueron posteriormente procesadas para su reconocimiento. El objetivo de la prueba fue evaluar la capacidad del aplicativo para identificar correctamente los medicamentos presentes en cada imagen. Los resultados de las pruebas se resumen en la figura 4.7, correspondiente a la matriz de confusión, una herramienta visual que permite analizar en detalle el desempeño del sistema en el reconocimiento de medicamentos a partir de imágenes. En dicha matriz, cada fila representa la clase real —el medicamento que debería ser identificado—, mientras que cada columna indica la predicción generada por el modelo. Los valores en la diagonal principal reflejan los aciertos, mientras que aquellos fuera de la diagonal corresponden a errores de clasificación.

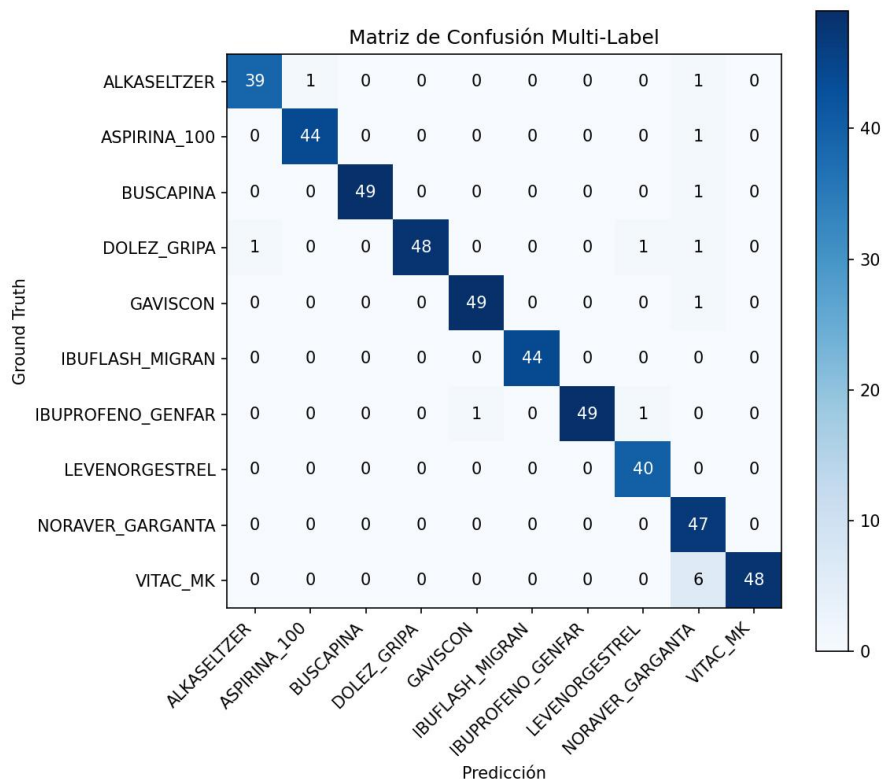


Figura 4.7: Matriz de confusión multi-label

Los resultados de la matriz de confusión muestran un desempeño sobresaliente del modelo en el reconocimiento de los medicamentos. En primer lugar, se identifican cuatro clases con un rendimiento perfecto: BUSCAPINA, DOLEZ-GRIPA, VITAC-MK e IBUPROFENO-GENFAR. Para estas clases, las métricas de precisión,

exhaustividad (recall) y F1 alcanzan el valor de 1.00, lo que indica que en todos los casos los medicamentos presentes en el conjunto de prueba fueron correctamente detectados, sin que se registraran falsos positivos.

En un segundo grupo se encuentran las clases ALKASELTZER, ASPIRINA-100, GAVISCON, IBUFLASH-MIGRAN, LEVENORGESTREL y NORAYER-GARGANTA. En este caso, la precisión se mantuvo siempre por encima de 0.88 y el recall osciló entre 0.81 y 1.00. Esto significa que el modelo detecta de manera muy confiable los medicamentos de estas clases, aunque en algunos casos se presentan falsos negativos, principalmente cuando el sistema no logra reconocer un medicamento presente o se produce solapamiento con otra clase. En particular, la clase NORAYER-GARGANTA presentó algunos falsos positivos, probablemente por confusión con imágenes de medicamentos visualmente similares o por la presencia de elementos de fondo y artefactos que dificultaron la clasificación.

El análisis detallado de la matriz de confusión confirma que la mayoría de los aciertos se concentran en la diagonal principal, lo que evidencia la alta capacidad del modelo para clasificar correctamente los medicamentos en sus respectivas clases. Los valores fuera de la diagonal fueron escasos. Por ejemplo, ALKASELTZER presentó un falso positivo como DOLEZ-GRIPA y otro como NORAYER-GARGANTA, aunque en ambos casos la incidencia fue mínima. Nuevamente, se resalta que la clase NORAYER-GARGANTA es la más susceptible a generar falsos positivos, con seis instancias en las que se predijo incorrectamente su presencia.

El análisis conjunto de la matriz de confusión y de las métricas obtenidas confirma el alto desempeño del sistema. Para la mayoría de los medicamentos, tanto la precisión como la sensibilidad superan el 90 %, e incluso se registran clases con reconocimiento perfecto. Los pocos errores identificados corresponden a casos puntuales, generalmente asociados a la similitud entre ciertos medicamentos o a condiciones visuales complejas presentes en las imágenes.

4.2. Discusión

El análisis de la distribución de medicamentos en el conjunto de datos reveló patrones relevantes, mostrando que aproximadamente el 50 % de las imágenes contenían solo un medicamento, mientras que el resto presentaba entre dos y siete medicamentos. Los mapas de calor generados reflejaron un equilibrio espacial en las anotaciones, lo cual permitió identificar ubicaciones específicas en las estanterías, proporcionando una representación detallada de la distribución de cada medicamento.

La evaluación de los modelos YOLOv8 arrojó resultados prometedores en la tarea de reconocimiento de medicamentos, con un desempeño notable. Se alcanzó una precisión del 94 %, una sensibilidad promedio del 97.2 %, un mAP50 de 97.5 % y un mAP50-95 de 86 %, lo que indica una alta capacidad del modelo para identificar con precisión los medicamentos en las imágenes.

Además, se realizaron pruebas de rendimiento en CPU y GPU para evaluar la capacidad de inferencia del modelo en distintas plataformas. En GPU, el modelo alcanzó un mAP50-95 de 85 % y una velocidad de inferencia promedio de 7 milisegundos por imagen, demostrando tanto un alto rendimiento como escalabilidad para manejar grandes volúmenes de datos. En CPU, aunque el modelo mantuvo un mAP50-95 similar al obtenido en GPU, la velocidad de inferencia promedio fue de 87 milisegundos por imagen, evidenciando una mayor carga de procesamiento en esta plataforma. Estas pruebas subrayan el potencial del modelo para funcionar eficientemente en dispositivos con recursos computacionales limitados, lo que extiende sus aplicaciones a escenarios prácticos, como sistemas de control en farmacias y aplicaciones móviles de asistencia sanitaria.

5. CONCLUSIONES

El estudio confirmó la viabilidad de diferenciar medicamentos a partir de las características visuales de sus empaques, incluso en condiciones de iluminación variables. El sistema logró reconocer múltiples medicamentos en una misma imagen, incluyendo escenarios con hasta siete productos simultáneamente. Esto demuestra que las diferencias gráficas presentes en los empaques, procesadas mediante técnicas de visión artificial, son suficientes para permitir una identificación precisa que reduce la dependencia de procesos manuales y susceptibles a errores humanos.

La construcción del dataset fue un aporte central de la investigación, dado que no existen bases públicas de este tipo en Colombia. La base de datos obtenida incorporó variaciones de orientación e iluminación, lo que incrementó la robustez del modelo entrenado. Este dataset constituye un recurso de valor académico y social, ya que puede ser reutilizado por futuros investigadores en proyectos orientados al sector salud, sin fines de lucro y con un enfoque en el beneficio público.

Se implementó la arquitectura YOLOv8, seleccionada a partir de su uso exitoso en contextos de detección de productos en entornos minoristas, un escenario análogo al de las farmacias. La integración con herramientas como Roboflow, Ultralytics y Comet permitió desarrollar un flujo de trabajo MLOps completo que facilitó el entrenamiento, la optimización de hiperparámetros y el monitoreo del modelo. Este enfoque garantizó la precisión, escalabilidad y mejora continua del sistema de reconocimiento.

Se desarrolló una aplicación móvil para dispositivos Android, que permite la captura de imágenes y la ejecución local del modelo entrenado. Esta aplicación constituye la interfaz práctica entre el usuario y el sistema de visión artificial, ofreciendo una herramienta funcional para el apoyo a procesos de inventario en farmacias.

La evaluación del modelo YOLOv8n optimizado y exportado a TensorFlow Lite mostró un desempeño sobresaliente en dispositivos móviles. En pruebas, se alcanzó una precisión del 94 % y una sensibilidad del 97,2 %, con velocidades de inferencia de 7 ms por imagen en GPU (95 imágenes/segundo) y 87 ms por imagen en CPU. Estos resultados evidencian que el modelo es lo suficientemente eficiente para operar en tiempo real y adaptable a dispositivos con recursos computacionales limitados, lo que amplía su aplicabilidad en entornos de alto tráfico y en sistemas de inventario automatizados.

5.1. Limitaciones

Esta investigación presenta varias limitaciones que afectan su aplicabilidad y generalización. En primer lugar, el proceso de entrenamiento y optimización del modelo YOLOv8 requiere una gran cantidad de datos de alta calidad, tanto en términos de volumen como de precisión en las anotaciones. La recopilación y etiquetado de estos datos representan un desafío importante, ya que requieren recursos significativos y son susceptibles a errores, especialmente en entornos con condiciones de iluminación y ángulos variables que pueden afectar la precisión del modelo en situaciones reales.

Además, el modelo, aunque fue optimizado para ejecutarse en dispositivos móviles mediante TensorFlow Lite, aún depende de recursos computacionales elevados para el entrenamiento inicial. Esto implica que los ajustes y mejoras continuas del modelo pueden ser limitados en entornos con baja capacidad computacional o en zonas con conectividad restringida, lo cual dificulta la actualización del modelo en tiempo real o su entrenamiento en dispositivos de baja potencia.

Otro aspecto limitante es la dependencia de una conexión a internet para el flujo de trabajo MLOps, necesario para la adquisición de datos, el etiquetado y el almacenamiento en la nube de modelos entrenados. Esto puede ser una barrera en entornos rurales o en zonas con poca infraestructura de conectividad, afectando la escalabilidad del sistema y limitando su adopción en lugares con acceso limitado a internet.

Finalmente, la implementación en dispositivos móviles presenta un rendimiento de inferencia más lento cuando se ejecuta en CPU en comparación con el rendimiento en GPU. Esto limita su capacidad para aplicaciones en tiempo real en sistemas que no disponen de hardware especializado, lo que puede restringir el uso del sistema en dispositivos de gama baja y en situaciones que requieran una detección rápida de múltiples medicamentos en secuencia.

Estas limitaciones sugieren que, aunque el modelo es efectivo en el contexto del estudio, se requerirán mejoras adicionales en la recopilación de datos, la optimización de recursos y la infraestructura de conectividad para extender su aplicabilidad y robustez en escenarios más diversos y exigentes.

6. TRABAJOS FUTUROS

Para mejorar el alcance y la precisión del modelo, así como su aplicabilidad en entornos diversos, es necesario incrementar el conjunto de datos de entrenamiento, incluyendo imágenes de medicamentos en diversas condiciones ambientales, como cambios de iluminación, ángulos y variaciones en el empaque. Además, sería útil incorporar más clases de medicamentos para ampliar la cobertura del sistema y su capacidad de generalización, permitiendo su adaptación a inventarios más amplios en farmacias y hospitales. Aunque se realizaron ajustes de hiperparámetros básicos, la implementación de métodos avanzados de optimización, como la búsqueda de hiperparámetros mediante algoritmos evolutivos o la optimización bayesiana, podría mejorar aún más el rendimiento del modelo, equilibrando precisión y eficiencia computacional.

El desarrollo de una infraestructura MLOps completa, garantizará la mejora continua del sistema, se sugiere desarrollar una infraestructura que permita la actualización automática del modelo en producción mediante la incorporación de nuevos datos etiquetados. Esta infraestructura facilitaría un aprendizaje continuo, permitiendo que el modelo se adapte a nuevos tipos de medicamentos. A la vez, es necesario ampliar las pruebas en distintos tipos de hardware, como dispositivos embebidos o hardware especializado en IA de bajo consumo, que permitirán evaluar el rendimiento y ajustar los modelos para dispositivos con limitaciones de procesamiento. Esto es especialmente importante para asegurar su implementación en dispositivos móviles de gama baja o en zonas rurales donde la infraestructura tecnológica es limitada.

Se sugiere desarrollar características adicionales, como un sistema de alerta para notificar el bajo inventario de medicamentos, una función de escaneo de múltiples estanterías para grandes farmacias, y la capacidad de llevar un registro automatizado de la caducidad de los medicamentos. Estas funciones enriquecerían la aplicación y la convertirían en una herramienta integral para la gestión farmacéutica. En colaboración con hospitales y farmacias para realizar pruebas en entornos reales, permitiendo obtener datos en tiempo real y recibir retroalimentación de los usuarios finales. Esto no solo mejoraría la validación del modelo en situaciones prácticas, sino que también ayudaría a adaptar el sistema a necesidades específicas del sector salud.

7. ANEXOS

A lo largo del documento, se han hecho referencias a varios documentos y materiales adicionales que están disponibles en el directorio virtual alojado en Google Drive. La estructura de este directorio se presenta en la siguiente figura 7.1.

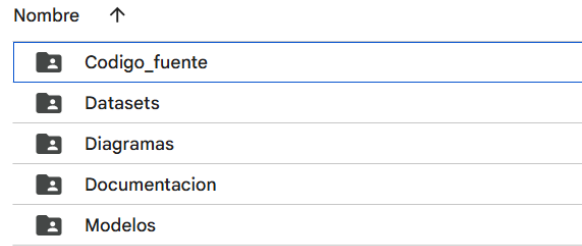


Figura 7.1: Anexos. Fuente: Elaboración propia

Enlace de los anexos: https://drive.google.com/drive/folders/1vIzBV1AegCp5x1N0cSkE3XvBmtAEk6U9?usp=drive_link

De acuerdo con el objetivo 1 *“Construir un conjuntos de datos de imágenes digitales de medicamentos comerciales distribuidos en Colombia, con variaciones en su orientación e iluminación”*, se ha generado un conjunto de datos público disponible en este enlace:

Click aquí para acceder al conjunto de datos de imágenes de medicamentos.

El código fuente del proyecto se encuentra alojado en dos repositorios públicos en GitHub. El primero está enfocado en el desarrollo de la aplicación Android, mientras que el segundo contiene los scripts en Python correspondientes al proceso de entrenamiento, ajuste y evaluación del modelo.

1. Aplicación móvil Android:
https://github.com/EdwarMontano/App_RecognizeMedications
2. Scripts de Python:
https://github.com/EdwarMontano/Trainer_RecognizeMedications

REFERENCIAS

- [1] M. S. Rodríguez Barrero, A. Buitrago Mejía, N. Varón Triana, and R. Quintero García, “La satisfacción de los usuarios afiliados al sistema de salud en la ciudad de Ibagué, Colombia,” *Revista Lebre*, 7 2020.
- [2] V. D. Hristov, D. N. Saliev, and D. V. Slavov, “Artificial Intelligence Systems for Warehouses Stocks Control,” *2022 8th International Conference on Energy Efficiency and Agricultural Engineering, EE and AE 2022 - Proceedings*, 2022.
- [3] Audifarma, “Por que es importante el inventario en la farmacia,” 12 2018.
- [4] J. Ávalos Juárez, *Aplicaciones de Visión por Computador en Farmacia Hospitalaria*. PhD thesis, Universidad de Sevilla, Sevilla, 3 2016.
- [5] Ministerio de Salud de la Republica de Colombia, “Manual Buenas Prácticas De Almacenamiento Bodega Zona Franca Del Ministerio De Salud Y Protección Social Ministerio De Salud Y Protección Social Bogotá, Marzo De 2021,” p. 7, 2021.
- [6] I. Romero-Zayas, F. Campos Añón, M. Santos Virosta, J. Cordón del Pozo, C. Santos Montero, A. Niñerola Baizán, and D. Fuster, “Implementation of the failure modes and effects analysis in a Hospital Radiopharmacy Unit,” *Revista Española de Medicina Nuclear e Imagen Molecular (English Edition)*, vol. 41, pp. 300–310, 9 2022.
- [7] M. Acosta Muñoz, *Estrategia de Nivelación de inventarios de medicamentos en droguerías Comfandi*. PhD thesis, Universidad del Valle, Santiago de Cali, 2021.
- [8] R. Solovyev, W. Wang, and T. Gabruseva, “Weighted boxes fusion: Ensembling boxes from different object detection models,” *Image and Vision Computing*, vol. 107, p. 104117, 3 2021.
- [9] B. Santra and D. P. Mukherjee, “A comprehensive survey on computer vision based approaches for automatic identification of products in retail store,” *Image and Vision Computing*, vol. 86, pp. 45–63, 6 2019.
- [10] A. K. Ali, “Inventory Management in Pharmacy Practice : A Review of Literature Abstarct,” *Archives of Pharmacy Practice*, vol. 2, no. 4, pp. 151–157, 2011.
- [11] J. C. Santa María Pinedo, C. A. Ríos López, C. Rodríguez Grández, and C. W. García Estrella, “Reconocimiento de patrones de imágenes a través de un sistema de visión artificial en MATLAB,” *Revista Científica de Sistemas e Informática*, vol. 1, pp. 15–26, 7 2021.

- [12] S. C. O. A. I. o. t. N. S. COUNCIL and TECHNOLOGY, “National Artificial Intelligence Research and Development Strategic Plan,” 2023.
- [13] R. Krashinsky, O. Giroux, S. Jones, N. Stam, and S. Ramaswamy, “NVIDIA Ampere Architecture In-Depth,” 2020.
- [14] I. A. Salazar Ortiz and H. M. Núñez Castro, *Lectura de Medidores de Energía por Medio de Sistemas de Reconocimiento de Caracteres con Machine Learning - Redes Neurales (OCR)*. PhD thesis, Universidad de los Andes, Bogota, 12 2020.
- [15] M. Rivero-Palacio, W. Alfonso-Morales, and E. Caicedo-Bravo, “Mobile Application for Anemia Detection through Ocular Conjunctiva Images,” in *2021 IEEE Colombian Conference on Applications of Computational Intelligence, ColCACI 2021 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., 5 2021.
- [16] P. Wajire, S. Angadi, and L. Nagar, “Image Classification for Retail,” in *2020 International Conference on Industry 4.0 Technology*, (Pune), pp. 23–28, 2 2020.
- [17] T. T. Mac, “Application of Improved YOLOv3 for Pill Manufacturing System,” *IFAC-PapersOnLine*, vol. 54, pp. 544–549, 1 2021.
- [18] S. Li, S. Zhou, Y. Huang, C. Liu, and X. Deng, “YOLO-CBD: A Recognition and Detection Method for Cigarette Box Based on YOLOv5,” in *Proceedings - 2023 9th International Symposium on System Security, Safety, and Reliability, ISSSR 2023*, pp. 243–250, Institute of Electrical and Electronics Engineers Inc., 2023.
- [19] K. Kumar, J. Simha, and R. Agarwal, “Retail Shelf Planning Using YOLO V8,” in *2023 Global Conference on Information Technologies and Communications, GCITC 2023*, Institute of Electrical and Electronics Engineers Inc., 2023.
- [20] H. Zhu, A. Dong, and J. Yu, “A Drug Box Recognition System Based on Deep Learning and Cloud-Edge Collaboration for Pharmacy Automation,” in *Proceedings of the International Conference on Parallel and Distributed Systems - ICPADS*, pp. 2579–2587, IEEE Computer Society, 2023.
- [21] P. Herrera-Toranz, J. Castro-Rivera, and W. Ugarte, “Detection and Verification of the Status of Products Using YOLOv5,” in *ICSBT International Conference on Smart Business Technologies*, vol. 2023-July, pp. 83–93, Science and Technology Publications, Lda, 2023.
- [22] C. G. Melek, E. B. Sonmez, and S. Albayrak, “Object Detection in Shelf Images with YOLO,” in *IEEE EUROCON 2019 -18th International Conference on Smart Technologies*, (Novi Sad), pp. 1–5, 2019.

- [23] W.-J. Chang, L.-B. Chen, C.-H. Hsu, C.-P. Lin, and T.-C. Yang, “A Deep Learning-Based Intelligent Medicine Recognition System for Chronic Patients,” *IEEE Access*, vol. 7, pp. 44441–44458, 2019.
- [24] Ministerio de Salud de la Republica de Colombia, “Definición de Estandares de datos de Medicamentos ,” 8 2018.
- [25] F. Chollet, *Deep Learning with Python*. Shelter Island: Manning Publications Co., second ed., 2021.
- [26] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *IEEE International Conference on Neural Networks - Conference Proceedings*, vol. 4, pp. 1942–1948, 1995.
- [27] P. Ponce Cruz, *Inteligencia Artificial Con Aplicaciones a la Ingeniería*, vol. 1. Ciudad de Mexico: Alfaomega, first ed., 7 2010.
- [28] P. Y. Simard, D. Steinkraus, and J. C. Platt, “Best practices for convolutional neural networks applied to visual document analysis,” *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, vol. 2003-Janua, pp. 958–963, 2003.
- [29] Megha Shroff, “Know your Neural Network architecture more by understanding these terms,” 5 2023.
- [30] R. Klette, *Concise Computer Vision - An Introduction into Theory and / Reinhard Klette / Springer*. 2014.
- [31] B. Chitradevi and P.Srimathi, “An Overview on Image Processing Techniques,” *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, pp. 6466–6472, 2014.
- [32] K. R. Chowdhary, “Machine Vision,” in *Fundamentals of Artificial Intelligence*, pp. 669–706, New Delhi: Springer India, 2020.
- [33] R. Szeliski, *Computer Vision*. Texts in Computer Science, Cham: Springer International Publishing, 2022.
- [34] Ioannis Pitas, *Digital Image Processing Algorithms and Applications*. Wiley-Interscience, 1 ed., 3 2000.
- [35] R. Sanginetti, “Procesamiento Digital de Imágenes,” 2017.
- [36] S. Krig, *Computer Vision Metrics*. Cham: Springer International Publishing, 1 ed., 9 2016.
- [37] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” *Arxiv*, 6 2015.

- [38] C.-H. Lee, C.-R. Lin, and M.-S. Chen, “Sliding-window filtering: an efficient algorithm for incremental mining,” in *Proceedings of the Tenth International Conference on Information and Knowledge Management, CIKM '01*, (New York, NY, USA), pp. 263–270, Association for Computing Machinery, 2001.
- [39] Ministerio de Salud y Protección Social, “Boletín Gasto de Bolsillo en Medicamentos,” no. 02, pp. 1–30, 2022.
- [40] Ultralytics, “Ultralytics YOLOv8 Documentation,” 3 2024.
- [41] Rajani Dixit and Brij Bhushan, “Scrum: An Agile Software Development Process and Metrics,” *Journal on Today’s Ideas - Tomorrow’s Technologies*, vol. 7, 6 2019.