



**UNIVERSIDAD POLITÉCNICA SALESIANA**  
**SEDE GUAYAQUIL**  
**CARRERA DE MECATRÓNICA**

**IMPLEMENTACIÓN DE UN SISTEMA BASADO EN VISIÓN  
ARTIFICIAL PARA LA VERIFICACIÓN DEL USO CORRECTO  
DE EQUIPOS DE PROTECCIÓN PERSONAL Y EL MONITOREO  
DE ACCESO A ZONAS DE RIESGOS EN ENTORNOS  
INDUSTRIALES**

Trabajo de titulación previo a la obtención del  
título de Ingeniero en Mecatrónica

**AUTORES:** Kevin Rolando Ochoa Barrera  
Franklin Israel Navarro Chavez  
**TUTOR:** Ing. Alberto Santiago Ramírez Farfán

Guayaquil - Ecuador  
2026

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA  
UNIVERSIDAD POLITÉCNICA SALESIANA**

Nosotros, **Franklin Israel Navarro Chavez** con documento de identificación N° **0955338314** y **Kevin Rolando Ochoa Barrera** con documento de identificación N° **0705001410**, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del **Dispositivo Tecnológico: IMPLEMENTACIÓN DE UN SISTEMA BASADO EN VISIÓN ARTIFICIAL PARA LA VERIFICACIÓN DEL USO CORRECTO DE EQUIPOS DE PROTECCIÓN PERSONAL Y EL MONITOREO DE ACCESO A ZONAS DE RIESGOS EN ENTORNOS INDUSTRIALES**, el cual ha sido desarrollado para optar por el título de: Ingeniero en Mecatrónica, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 18 de febrero del año 2026

Atentamente,



---

Franklin Israel Navarro Chavez  
0955338314



---

Kevin Rolando Ochoa Barrera  
0705001410

## CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, **Franklin Israel Navarro Chavez** con documento de identificación N° **0955338314** y **Kevin Rolando Ochoa Barrera** con documento de identificación N° **0705001410**; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo.

Guayaquil, 18 de febrero del año 2026

Atentamente,



---

Franklin Israel Navarro Chavez  
0955338314



---

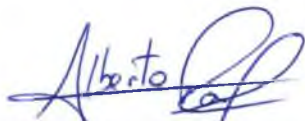
Kevin Rolando Ochoa Barrera  
0705001410

## CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, **Alberto Santiago Ramírez Farfán**, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **IMPLEMENTACIÓN DE UN SISTEMA BASADO EN VISIÓN ARTIFICIAL PARA LA VERIFICACIÓN DEL USO CORRECTO DE EQUIPOS DE PROTECCIÓN PERSONAL Y EL MONITOREO DE ACCESO A ZONAS DE RIESGOS EN ENTORNOS INDUSTRIALES**, realizado por **Franklin Israel Navarro Chavez** con documento de identificación N° **0955338314** y por **Kevin Rolando Ochoa Barrera** con documento de identificación N° **0705001410**, obteniendo como resultado final el trabajo de titulación bajo la opción **Dispositivo Tecnológico** que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 18 de febrero del año 2026

Atentamente,



---

Ing. Alberto Santiago Ramírez Farfán, Mg.

0923348890

## DEDICATORIA

Este trabajo de titulación está dedicado, en primer lugar, a mi familia. A mi padre, Franklin Navarro, por ser un pilar esencial en mi formación personal y académica, y por su constante disposición para brindarme apoyo en cada etapa de mi vida.

A mi madre, Sonia Chavez, quien ha sido mi compañera incondicional durante todo mi proceso universitario, ofreciéndome siempre su respaldo, orientación y motivación. Ella ha sido una influencia fundamental en mi crecimiento como persona, enseñándome a perseverar y a seguir adelante a pesar de los obstáculos y las dificultades.

Finalmente, dedico este proyecto de titulación a todos los estudiantes que continúan esforzándose por alcanzar sus metas profesionales, especialmente a los futuros Ingenieros en Mecatrónica. Espero que, en algún momento, pueda contribuir a su desarrollo, así como ellos podrán aportar significativamente al bienestar de los demás.

**Franklin Israel Navarro Chavez**

Este trabajo de titulación está dedicado, en primer lugar, a mi madre, Martha Barrera, quien ha sido el pilar fundamental de mi vida. Gracias a su esfuerzo constante, su apoyo incondicional y los sacrificios que ha realizado por mí, hoy estoy a punto de culminar esta etapa universitaria. Su ejemplo de fortaleza y amor ha sido la motivación que me ha impulsado a no rendirme.

A mi padre, Rolando Ochoa, por inspirarme a elegir esta carrera y enseñarme, con su ejemplo, el valor del trabajo duro, la responsabilidad y la perseverancia. Sus consejos y enseñanzas me han acompañado en cada desafío, recordándome que, aun en los momentos difíciles, siempre es posible salir adelante con determinación y fe.

Extiendo esta dedicatoria a toda mi familia, quienes me han inculcado que la educación es el mayor legado que se puede recibir y también el mejor regalo que se puede ofrecer. Gracias a su apoyo y confianza, he podido crecer personal y profesionalmente, entendiendo que el conocimiento es el puente que nos permite avanzar, innovar y convertir los sueños en realidad.

**Kevin Rolando Ochoa Barrera**

## AGRADECIMIENTO

Expreso mi más sincero agradecimiento a Dios por brindarme la fortaleza, la sabiduría y la constancia necesarias para culminar esta importante etapa de mi vida académica.

A mis padres, Franklin Navarro y Sonia Chavez, por su amor, apoyo incondicional y confianza permanente en cada uno de mis pasos. Gracias por sus consejos, sacrificios y por motivarme a superarme día tras día.

A mis docentes y a la institución que formó parte de mi preparación profesional, por compartir sus conocimientos, experiencias y enseñanzas, los cuales fueron fundamentales para mi formación como Ingeniero en Mecatrónica.

A mis compañeros y amigos, quienes fueron un apoyo constante durante este proceso, compartiendo experiencias, aprendizajes y momentos que contribuyeron a mi crecimiento personal y académico.

Finalmente, agradezco a todas las personas que, de una u otra manera, aportaron con su apoyo y contribuyeron al desarrollo y culminación de este trabajo de titulación.

**Franklin Israel Navarro Chavez**

Expreso mi sincero agradecimiento a los docentes de la carrera de Ingeniería en Mecatrónica, quienes fueron parte fundamental de mi formación académica y profesional. A través de sus conocimientos, experiencia y dedicación, me brindaron no solo las bases técnicas necesarias, sino también la orientación y motivación para proyectarme como un profesional comprometido y preparado para los retos del futuro.

Asimismo, agradezco a las amistades que la universidad puso en mi camino, tanto de manera directa como indirecta. Cada una de ellas aportó algo valioso a esta etapa de mi vida, demostrando compañerismo, apoyo y calidad humana. Gracias por acompañarme en los momentos de mayor exigencia, por las experiencias compartidas y por hacer de este proceso un recorrido más llevadero, significativo y enriquecedor.

**Kevin Rolando Ochoa Barrera**

## RESUMEN

La presente tesis desarrolla e implementa un sistema de visión artificial orientado a la detección automática de Equipos de Protección Personal (EPP) mediante técnicas de aprendizaje profundo, el objetivo principal es contribuir a la mejora de la seguridad laboral a través de un modelo capaz de identificar en tiempo real elementos como casco y chaleco de seguridad en entornos operativos.

Para ello, se construyó y etiquetó un dataset específico, el cual fue empleado para entrenar un modelo de detección basado en la arquitectura YOLO11s, seleccionada por su equilibrio entre velocidad de inferencia y precisión, lo que la hace adecuada para entornos con recursos computacionales limitados. El proceso de entrenamiento se realizó bajo una configuración estructurada mediante archivo data.yaml, incluyendo particiones de entrenamiento, validación y prueba.

El desempeño del modelo fue evaluado utilizando métricas estándar como precisión, exhaustividad, mAP@0.5 y matriz de confusión, los resultados evidencian una alta capacidad de detección, con valores elevados de precisión y recall, lo que confirma la convergencia estable del modelo y su potencial aplicabilidad en sistemas de monitoreo de seguridad industrial.

En conclusión, el sistema propuesto demuestra que la implementación de modelos de detección basados en deep learning puede integrarse eficazmente en soluciones de bajo costo, contribuyendo a la reducción de riesgos laborales y al fortalecimiento de la cultura preventiva en entornos de trabajo.

**Palabras claves:** Visión artificial, detección de objetos, aprendizaje profundo, YOLO11s, Equipos de Protección Personal (EPP), seguridad industrial, procesamiento de imágenes, redes neuronales convolucionales.

## ABSTRACT

This thesis presents the development and implementation of a computer vision-based system for the automatic detection of Personal Protective Equipment (PPE) using deep learning techniques. The main objective is to enhance workplace safety by identifying, in real time, the correct use of protective elements such as safety helmets and reflective vests in industrial environments.

A dedicated dataset was created and manually labeled to train an object detection model based on the YOLO11s architecture. This model was selected due to its balance between inference speed and detection accuracy, making it suitable for low-resource hardware platforms while maintaining near real-time performance. The training process was conducted using a structured configuration defined in a data.yaml file, including training, validation, and test data splits.

Model performance was evaluated using standard metrics such as Precision, Recall, mAP@0.5, and confusion matrix analysis. The results demonstrate high detection capability, with strong precision and recall values, indicating stable convergence and reliable generalization on validation data.

The proposed system proves that deep learning-based object detection models can be effectively integrated into cost-efficient safety monitoring solutions, contributing to accident prevention and the reinforcement of occupational safety culture in industrial settings.

**Keywords:** Computer vision, object detection, deep learning, YOLO11s, Personal Protective Equipment (PPE), industrial safety, image processing, convolutional neural networks.

## ÍNDICE

<b>I.</b>	<b>Introducción</b>	1
<b>II.</b>	<b>Problema</b>	2
<b>III.</b>	<b>Justificación</b>	3
<b>IV.</b>	<b>Objetivos</b>	4
IV-A.	Objetivo general . . . . .	4
IV-B.	Objetivos específicos . . . . .	4
<b>V.</b>	<b>Marco Teórico</b>	5
V-A.	Visión Artificial . . . . .	5
V-B.	Tipos de uso de visión artificial . . . . .	6
V-C.	Clasificación de la visión artificial . . . . .	6
V-D.	Detección de objetos . . . . .	6
V-E.	Segmentación de la visión artificial . . . . .	7
V-F.	Dataset . . . . .	9
V-G.	Tipos de dataset . . . . .	9
V-H.	Formato y estructura . . . . .	10
V-I.	Hardware Propuesto para el Sistema . . . . .	11
V-J.	Cámara Web . . . . .	12
V-K.	Cerradura electrica con solenoide 12v dc . . . . .	13
V-L.	Tecnologías Clave . . . . .	14
V-M.	Redes Neuronales Artificiales (RNA) . . . . .	15
V-N.	Estructura básica de RNA . . . . .	16
V-Ñ.	Tipos de redes neuronales . . . . .	17
V-O.	Equipos de Protección Personal (EPP) . . . . .	18
V-P.	Detección de EPP mediante Visión Artificial . . . . .	18
V-Q.	Delimitación de Zonas de Riesgo . . . . .	20
V-R.	Impacto del Proyecto . . . . .	21
<b>VI.</b>	<b>Marco Metodológico</b>	23
VI-A.	Funcionamiento de visual studio . . . . .	24
VI-B.	Preparación de entorno . . . . .	24
VI-C.	Elaboración de Dataset . . . . .	25
VI-C1.	Etiquetado de imágenes . . . . .	26
VI-C2.	Conformación del Dataset . . . . .	26
VI-D.	Entrenamiento del modelo . . . . .	28
VI-E.	Métricas de evaluación . . . . .	30
VI-E1.	Precisión general (Accuracy) . . . . .	30
VI-E2.	Precisión por clase . . . . .	30
VI-E3.	Recall o Exhaustividad . . . . .	30
VI-F.	Pruebas en modelo pre-entrenado . . . . .	30
VI-G.	Resultados del entrenamiento . . . . .	31
VI-G1.	Resultados cualitativos del entrenamiento: . . . . .	31

<b>VII. Resultados</b>	34
VII-A. Desempeño cuantitativo global . . . . .	34
VII-B. Matriz de confusión . . . . .	35
VII-C. Distribución de etiquetas . . . . .	35
VII-D. Umbral de confianza y compromiso Presión-Recall . . . . .	36
VII-D1. Precision–Confidence Curve . . . . .	37
VII-D2. Recall–Confidence Curve . . . . .	37
VII-D3. F1–Confidence Curve . . . . .	38
<b>VIII. Cronograma</b>	39
<b>IX. Presupuesto</b>	40
<b>X. Conclusiones</b>	41
<b>XI. Recomendaciones</b>	42
<b>Referencias</b>	43
<b>Anexo A: Guía</b>	45
A-A. Procedimiento de Etiquetado . . . . .	45
A-B. Reglas de Etiquetado . . . . .	48
A-C. División del Dataset . . . . .	48
A-D. Conclusión . . . . .	49
<b>Anexo B: Código</b>	50
<b>Anexo C: resultados del entrenamiento con 100 Epoch</b>	50

## ÍNDICE DE FIGURAS

1.	Esquema general de un sistema de visión artificial [21]. . . . .	5
2.	Interpretación de la visión artificial [22]. . . . .	6
3.	Detección de la visión artificial [21]. . . . .	7
4.	Detección de la visión artificial [23]. . . . .	8
5.	Flujo de datos [24]. . . . .	9
6.	Diagrama de representación del DataSet [21]. . . . .	10
7.	Diagrama de representación del DataSet [25]. . . . .	11
8.	Esquema de la una red neuronal artificial [26]. . . . .	12
9.	Tipos de cámara. [27]. . . . .	13
10.	Cerradura electrica 12V. [28]. . . . .	14
11.	Tecnología y su entorno. [26]. . . . .	14
12.	Red neuronal. [26]. . . . .	16
13.	Red neuronal. [29]. . . . .	17
14.	Trabajador industrial equipado correctamente con EPP [30]. . . . .	18
15.	Detección de EPP con modelo YOLO [31]. . . . .	20
16.	Representación se zonas seguras [23]. . . . .	21
17.	Impacto esperado en la reducción de accidentes [25]. . . . .	22
18.	Diagrama de flujo del funcionamiento. Elaborado por autores . . . . .	23
19.	Funcionamiento visual. Elaborado por autores . . . . .	24
20.	Entorno por visual studio code. Elaborado por autores . . . . .	24
21.	Entrenamiento del modelo. Elaborado por autores . . . . .	25
22.	Uso del Roboflow. Elaborado por autores . . . . .	25
23.	Etiquetado de EPP. Elaborado por autores . . . . .	26
24.	Etiquetado de EPP. Elaborado por autores . . . . .	26
25.	DataSet. Elaborado por autores . . . . .	27
26.	Imágenes del DataSet. Elaborado por autores . . . . .	27
27.	Versión Beta. Elaborado por autores . . . . .	28
28.	Entrenamiento del modelo. Elaborado por autores . . . . .	28
29.	Entrenamiento del modelo. Elaborado por autores . . . . .	29
30.	Entrenamiento del modelo. Elaborado por autores . . . . .	30
31.	Prueba visual. Elaborado por autores . . . . .	31
32.	Predicciones en validación. Elaborado por autores . . . . .	32
33.	Predicciones en validación. Elaborado por autores . . . . .	32
34.	Predicciones en validación. Elaborado por autores . . . . .	33
35.	Curva Precisión-Recall sobre el conjunto de prueba. Elaborado por autores . . . . .	34
36.	Evolución de métricas durante el entrenamiento de mAP. Elaborado por autores . . . . .	34
37.	Matriz de confusión normalizada sobre el conjunto de prueba. Elaborado por autores . . . . .	35
38.	Distribución de etiquetas sobre el conjunto de prueba. Elaborado por autores . . . . .	36
39.	Presión-Recall umbral de confianza. Elaborado por autores . . . . .	37
40.	Precision-Confidence umbral de confianza. Elaborado por autores . . . . .	37
41.	Recall-Confidence umbral de confianza. Elaborado por autores . . . . .	38
42.	F1-Confidence umbral de confianza. Elaborado por autores . . . . .	38
43.	Registro fotográfico del proyecto. Elaborado por autores . . . . .	45
44.	Registro fotográfico del proyecto. Elaborado por autores . . . . .	45
45.	Registro fotográfico del proyecto. Elaborado por autores . . . . .	46
46.	Registro fotográfico del proyecto. Elaborado por autores . . . . .	46
47.	Registro de casco y gafas. Elaborado por autores . . . . .	47
48.	Registro fotográfico del proyecto. Elaborado por autores . . . . .	47
49.	Registro fotográfico del proyecto. Elaborado por autores . . . . .	48

50.	Resultados visuales del modelo YOLO11s. Elaborado por autores . . . . .	48
51.	Resultados visuales del modelo YOLO11s. Elaborado por autores . . . . .	49

## ÍNDICE DE TABLAS

I.	Cronograma . . . . .	39
II.	Presupuesto para la implementación del trabajo de titulación (simulación y documentación) . . . . .	40

## I. INTRODUCCIÓN

La seguridad industrial constituye un pilar fundamental en la prevención de accidentes laborales y en la protección de la integridad física de los trabajadores, a pesar de la existencia de normativas y protocolos estrictos sobre el uso obligatorio de Equipos de Protección Personal (EPP), en muchos entornos industriales aún se presentan incumplimientos que incrementan el riesgo de incidentes. La supervisión manual tradicional resulta limitada debido a factores como la subjetividad humana, la falta de monitoreo continuo y la imposibilidad de cubrir múltiples áreas de manera simultánea.

En los últimos años, el avance de la visión artificial y el aprendizaje profundo ha permitido desarrollar sistemas automatizados capaces de analizar imágenes y videos en tiempo real con altos niveles de precisión, en particular los modelos de detección de objetos basados en Redes Neuronales Convolucionales (CNN) han demostrado un desempeño sobresaliente en tareas de identificación y localización de elementos específicos dentro de una escena. Entre estas arquitecturas, la familia YOLO se ha consolidado como una de las más eficientes, al combinar velocidad de inferencia y exactitud en la detección.

En este contexto, la presente tesis propone el diseño e implementación de un sistema basado en visión artificial para la verificación automática del uso correcto de EPP en entornos industriales. Para ello, se construyó y etiquetó un dataset específico orientado a la detección de elementos como casco y chaleco de seguridad, el cual fue utilizado para entrenar un modelo basado en la arquitectura YOLO11s, esta versión fue seleccionada por su capacidad de operar en plataformas de recursos limitados, manteniendo un rendimiento adecuado para aplicaciones en tiempo casi real.

El sistema desarrollado integra etapas de adquisición de imágenes, procesamiento, entrenamiento del modelo, evaluación mediante métricas estándar (Precision, Recall y mAP) y validación experimental. Los resultados obtenidos evidencian una alta capacidad de detección y estabilidad en el desempeño, demostrando la viabilidad de implementar soluciones automatizadas de bajo costo que contribuyan a fortalecer la cultura preventiva y optimizar los procesos de supervisión en seguridad industrial.

## II. PROBLEMA

En los entornos industriales y de construcción, el cumplimiento del uso de Equipos de Protección Personal (EPP) es una medida esencial para prevenir accidentes laborales. A pesar de las normativas existentes y los esfuerzos de supervisión, los índices de accidentes relacionados con la omisión o el uso incorrecto del EPP continúan siendo alarmantemente altos [1]. Según datos de la International Labour Organization (ILO), más de 2.3 millones de personas mueren cada año por accidentes o enfermedades laborales, muchas de ellas vinculadas a la falta de protección adecuada. Esta situación evidencia una brecha significativa entre la normativa de seguridad laboral y su aplicación efectiva en el campo [2].

Uno de los principales problemas radica en la dependencia de la supervisión no automatizada para verificar el cumplimiento del uso del EPP. Este método, tradicionalmente aceptado, presenta limitaciones claras como los supervisores no pueden estar presentes en todas las zonas de trabajo simultáneamente, la verificación visual es propensa a errores humanos y la vigilancia constante implica altos costos operativos [3]. El monitoreo humano no garantiza una respuesta inmediata ante infracciones, lo que aumenta la exposición al riesgo en tiempo real y disminuye la capacidad preventiva del sistema de seguridad ocupacional [4].

Otro aspecto relevante es la ausencia de un registro disponible y detallado de riesgos previamente identificados y mitigados. Sin este respaldo, las empresas no pueden aprender de experiencias pasadas ni implementar medidas preventivas basadas en datos históricos [5]. Este vacío informativo repercute fuertemente en la toma de decisiones y en la capacidad de anticiparse a posibles incidentes [6].

La falta de transparencia en la documentación puede derivar en problemas legales y regulatorios, especialmente en países donde las normativas de seguridad son estrictas [7]. Esta situación plantea la necesidad de un sistema integral, autónomo y preciso que no solo identifique el uso adecuado del EPP mediante visión artificial, sino que también actúe en consecuencia, limitando el acceso a zonas limitadas en caso de incumplimiento [8].

Un sistema de estas características permitiría reducir la dependencia de la supervisión humana, así como el número de infracciones y mejorar la eficiencia general en la prevención de riesgos laborales [9]. La implementación de soluciones basadas en inteligencia artificial y “edge computing” representa una alternativa viable y escalable para enfrentar este desafío de forma tecnológica, proactiva y sostenible [10].

### III. JUSTIFICACIÓN

La prevención de accidentes laborales es una prioridad en entornos industriales y de construcción, donde el incumplimiento en el uso de Equipos de Protección Personal (EPP) representa una de las principales causas de lesiones graves e incluso mortales [11].

Si bien existen normativas que obligan al uso de Equipos de Protección Personal, su cumplimiento efectivo continúa siendo un desafío operativo [12]. La vigilancia no automatizada presenta deficiencias como fatiga del personal, errores humanos, falta de cobertura en tiempo real y altos costos económicos, lo que evidencia la necesidad de soluciones más precisas, automatizadas y sostenibles [13].

La tecnología de visión artificial permite realizar un monitoreo continuo, no invasivo y con mínima intervención humana, lo que incrementa la eficiencia y reduce la posibilidad de omisiones en el control [14]. La aplicación de redes neuronales profundas, además, permite adaptarse a distintos entornos laborales con condiciones adversas como ruido visual, iluminación variable y múltiples trabajadores en escena [15]. No obstante, para que esta tecnología tenga un impacto real, es crucial integrarla con sistemas físicos de control de acceso [16]. Al vincular la detección automatizada de EPP con mecanismos como puertas inteligentes, torniquetes o barreras electrónicas, se garantiza que únicamente ingresen a zonas de riesgo aquellas personas que portan adecuadamente su equipo de seguridad [17].

Esta integración convierte al sistema no solo en un agente de monitoreo, sino también en una herramienta de acción preventiva. Además, la recopilación automática de datos permite generar reportes, emitir alertas y tomar decisiones estratégicas orientadas a la mejora continua de la seguridad. En comparación con los sistemas tradicionales basados en sensores o supervisión manual, la incorporación de algoritmos de visión artificial y aprendizaje profundo incrementa significativamente la precisión en la detección del uso correcto de los equipos de protección personal y en la validación del acceso a zonas de riesgo, esta mejora tecnológica permite un monitoreo más fiable, adaptable a diferentes condiciones lumínicas y contextos industriales. [18].

Por estas razones, el desarrollo de un sistema de visión artificial combinado con control de acceso automatizado se justifica técnica, social y económicamente [19]. No solo responde a una necesidad crítica en el ámbito de la seguridad laboral, sino que también aprovecha tecnologías emergentes como “edge computing”, inteligencia artificial y aprendizaje profundo para ofrecer una solución eficiente, escalable y alineada con los objetivos de la industria 4.0. Este tipo de sistema representa un paso concreto hacia la transformación digital de los procesos de seguridad ocupacional, generando entornos más seguros, inteligentes y responsables [20].

## IV. OBJETIVOS

### *IV-A. Objetivo general*

Implementar un sistema basado en visión artificial que permita la verificación del uso correcto de los equipos de protección personal (EPP) y monitoree el acceso a zonas de riesgo en entornos industriales, contribuyendo a la reducción de accidentes laborales.

### *IV-B. Objetivos específicos*

- Integrar un dataset orientado a la identificación y detección del uso correcto de equipos de protección personal en tiempo real.
- Entrenar una red de visión artificial, implementando un sistema que permita el acceso a zonas de riesgo de acuerdo al uso correcto de equipos de protección personal.
- Validar el sistema y la efectividad mediante pruebas de campo en entornos controlados, evaluando métricas de precisión, sensibilidad y especificidad de detección.

## V. MARCO TEÓRICO

### V-A. Visión Artificial

La visión artificial es una rama de la inteligencia artificial y del procesamiento digital de imágenes, cuyo objetivo es permitir a una máquina adquirir, procesar y analizar imágenes para realizar tareas específicas. Esta disciplina se basa en dotar a los sistemas informáticos de la capacidad de interpretar el entorno visual de forma similar a como lo haría un ser humano.

En el ámbito industrial, la visión artificial permite automatizar procesos como la inspección de calidad, el reconocimiento de objetos y, en este caso, la detección automática del uso correcto de Equipos de Protección Personal (EPP) por parte de los trabajadores [6].

Su objetivo principal es permitir que las computadoras extraigan información significativa de imágenes digitales o videos y luego tomen decisiones o actúen basándose en esa información.

La visión artificial sigue un flujo de procesamiento estructurado que inicia con la adquisición de la imagen, donde se capturan escenas estáticas o en movimiento mediante cámaras convencionales, sensores especializados como infrarrojos o LiDAR, o fuentes digitales previamente registradas. Posteriormente, la información visual obtenida pasa por un proceso de mejoramiento y acondicionamiento mediante técnicas de procesamiento digital orientadas a la reducción de ruido, corrección de color, ajuste de contraste, normalización o escalado, con el fin de optimizar la calidad de la imagen para su análisis. Una vez preprocesada, se procede a la fase de análisis y extracción de características, en la cual se identifican elementos relevantes como bordes, esquinas, texturas, formas y regiones de interés mediante algoritmos matemáticos y modelos de aprendizaje automático o profundo. Finalmente, con los datos extraídos se realiza la etapa de comprensión o interpretación de alto nivel, donde el sistema ejecuta tareas específicas tales como reconocimiento, clasificación, detección, segmentación, seguimiento o toma de decisiones automatizadas, dependiendo del objetivo aplicado.



Figura 1. Esquema general de un sistema de visión artificial [21].

### V-B. Tipos de uso de visión artificial

Es un campo de la inteligencia artificial orientado a la interpretación automatizada de información visual, permitiendo que un sistema tecnológico analice imágenes o videos para extraer conocimiento útil. Esta disciplina se aplica ampliamente en tareas como inspección y control de calidad industrial, reconocimiento de objetos y personas, medición sin contacto, gestión automatizada de procesos, monitoreo de seguridad, análisis de comportamiento y apoyo a la interacción humano-máquina en entornos robóticos. Dentro de los enfoques metodológicos más utilizados se encuentran la clasificación, la detección y la segmentación, cada uno con características y niveles de complejidad diferentes.

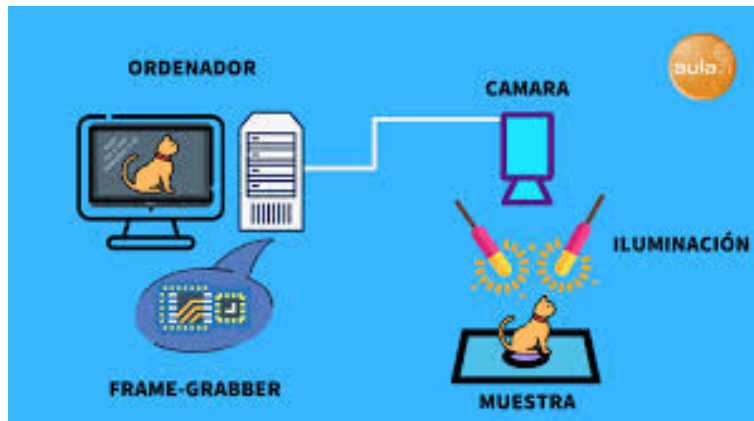


Figura 2. Interpretación de la visión artificial [22].

### V-C. Clasificación de la visión artificial

Es un proceso mediante el cual un sistema asigna una etiqueta o categoría a una imagen completa o a una sección relevante de ella, permitiendo identificar si el objeto de interés pertenece o no a una clase determinada en el contexto industrial, este enfoque puede emplearse para determinar si un trabajador utiliza un casco o chaleco de seguridad. Para esto se utilizan modelos basados en redes neuronales convencionales y arquitecturas avanzadas como ResNet, EfficientNet o modelos basados en transformadores visuales. Sin embargo la clasificación no proporciona información sobre la ubicación exacta del objeto dentro de la escena, lo que limita su aplicación cuando se requiere analizar posicionamiento o contexto espacial.

### V-D. Detección de objetos

En este apartado no solo identifica la presencia de un elemento específico sino que también determina su localización dentro del entorno visual mediante cuadros delimitadores, este enfoque permite reconocer si un casco se encuentra realmente en la región correspondiente a la cabeza del trabajador. Tecnologías como YOLO, Faster R-CNN o SSD se han convertido en referentes de alto rendimiento en aplicaciones en tiempo real, razón por la cual resultan especialmente útiles en la verificación del cumplimiento de normas de seguridad durante el acceso a zonas restringidas.

Desde un punto de vista técnico el problema de detección implica dos subprocesos simultáneos: la localización, que consiste en determinar las coordenadas espaciales ( $x$ ,  $y$ , ancho y alto) del objeto dentro de la imagen, y

la clasificación, que asigna una etiqueta de clase a cada región detectada. El modelo debe además estimar una probabilidad o nivel de confianza asociado a cada predicción.

En los enfoques tradicionales, la detección de objetos se basaba en técnicas de extracción manual de características como bordes, texturas o descriptores locales por ejemplo, HOG o SIFT, combinados con clasificadores como SVM, sin embargo, estos métodos presentaban limitaciones frente a variaciones de escala, iluminación y oclusión.

Con la llegada del aprendizaje profundo, la detección de objetos evolucionó significativamente gracias al uso de Redes Neuronales Convolucionales (CNN), que permiten aprender automáticamente representaciones jerárquicas de características directamente desde los datos, actualmente, los modelos se clasifican en dos grandes categorías: detectores de dos etapas, como Faster R-CNN, que primero generan propuestas de regiones y luego las clasifican detectores de una sola etapa, como la familia YOLO (You Only Look Once), que realizan la detección y clasificación en un único paso, logrando mayor velocidad de inferencia.

En aplicaciones industriales, como la verificación del uso de Equipos de Protección Personal (EPP), la detección de objetos permite identificar simultáneamente múltiples elementos dentro de una escena, incluso en presencia de varias personas, diferentes posturas y condiciones ambientales variables. La eficiencia del modelo se evalúa comúnmente mediante métricas como Precision, Recall y mAP (mean Average Precision), que cuantifican tanto la exactitud en la clasificación como la calidad en la localización espacial.

En el contexto de este trabajo, la detección de objetos se implementa mediante la arquitectura YOLO11s, la cual permite procesar imágenes en tiempo casi real, combinando alta precisión con eficiencia computacional, lo que la hace adecuada para sistemas de monitoreo en entornos industriales.

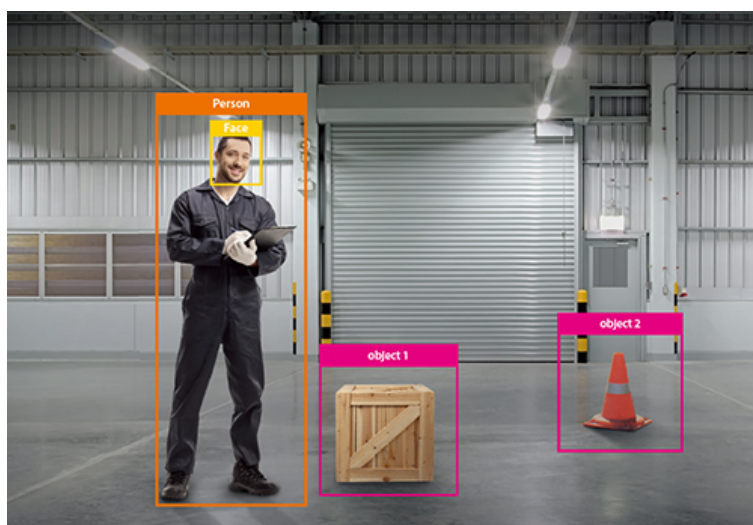


Figura 3. Detección de la visión artificial [21].

#### V-E. Segmentación de la visión artificial

La segmentación integra el nivel más avanzado dentro del análisis visual puesto que permite diferenciar regiones o píxeles específicos de la imagen, logrando una representación detallada de contornos y formas, a través de la segmentación semántica e instancia, modelos como Mask R-CNN, U-Net o DeepLab no solo identifican el

objeto, sino que analizan su forma, posición exacta y superficie, brindando una precisión significativamente mayor. En procesos de seguridad industrial esta técnica es fundamental para validar no solo la presencia del equipo de protección personal sino el uso correcto y la colocación adecuada, evitando falsos positivos que puedan darse cuando el trabajador sostiene el equipo pero no lo lleva correctamente colocado.

Desde una perspectiva técnica, la segmentación puede abordarse bajo diferentes enfoques. La segmentación semántica clasifica cada píxel de la imagen asignándole una etiqueta de clase, sin distinguir instancias individuales dentro de la misma categoría. La segmentación por instancias, en cambio, no solo identifica la clase del objeto, sino que diferencia cada objeto individual, incluso si pertenecen a la misma categoría, existe también la segmentación panóptica, que combina ambos enfoques al proporcionar información semántica y de instancia de manera simultánea.

Tradicionalmente, la segmentación se realizaba mediante métodos clásicos como umbralización, crecimiento de regiones, detección de bordes o algoritmos basados en clustering (por ejemplo, k-means). Sin embargo, estos métodos presentaban limitaciones frente a variaciones de iluminación, ruido o complejidad de fondo. Con el avance del aprendizaje profundo, surgieron arquitecturas especializadas como Fully Convolutional Networks (FCN), U-Net, Mask R-CNN y modelos derivados de la familia YOLO con capacidad de segmentación, que permiten aprender representaciones espaciales complejas directamente a partir de grandes volúmenes de datos.

La segmentación es especialmente relevante en aplicaciones donde se requiere alta precisión espacial, como en diagnóstico médico, conducción autónoma, inspección industrial y monitoreo de seguridad. En contextos industriales, por ejemplo permite analizar con mayor exactitud la correcta colocación de Equipos de Protección Personal, identificar zonas específicas de riesgo o evaluar condiciones del entorno con mayor nivel de detalle que una simple caja delimitadora.

En términos computacionales, la segmentación es más exigente que la detección de objetos, ya que implica procesar y clasificar cada píxel de la imagen, por lo tanto esto requiere mayor capacidad de memoria y procesamiento, aunque ofrece un nivel superior de información estructural sobre la escena analizada.

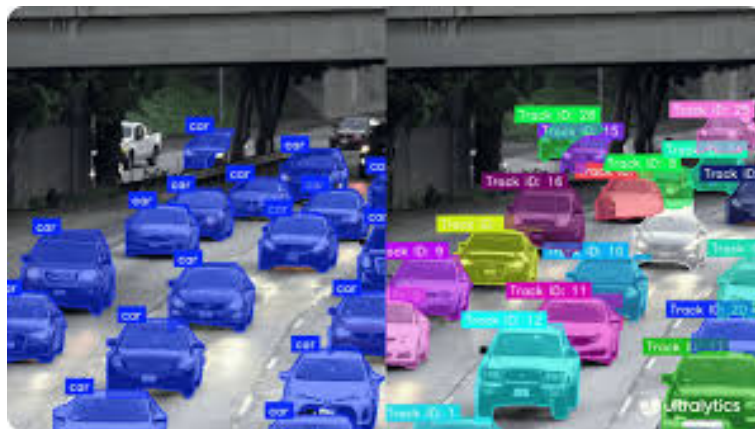


Figura 4. Detección de la visión artificial [23].

### V-F. Dataset

Un dataset o conjunto de datos es una colección organizada de información que se utiliza para análisis, investigación o entrenamiento de modelos de inteligencia artificial. En el contexto del aprendizaje automático y las redes neuronales, un dataset contiene ejemplos que el modelo usa para aprender patrones y relaciones entre variables.

Un dataset puede representarse como una matriz de datos, donde:

Cada fila de un conjunto de datos corresponde a un registro o instancia individual que representa un ejemplo dentro del problema a analizar, mientras que cada columna contiene una característica o atributo que actúa como variable independiente y describe alguna propiedad cuantitativa o cualitativa del registro. En el caso de los problemas supervisados, además de estas características, el conjunto de datos incorpora una etiqueta o valor objetivo como variable dependiente, la cual se desea predecir o clasificar mediante el modelo de aprendizaje automático o visión artificial, apoyándose frecuentemente en plataformas como Roboflow para la gestión, el etiquetado preciso y la estructuración de estos activos de datos de manera eficiente.



Figura 5. Flujo de datos [24].

### V-G. Tipos de dataset

El dataset de entrenamiento training set se utiliza para entrenar el modelo, permitiéndole ajustar sus parámetros internos y aprender los patrones presentes en los datos, además el dataset de validación validation set se emplea durante el entrenamiento para ajustar hiperparámetros y prevenir el sobreajuste overfitting, asegurando que el modelo no se limite únicamente a los ejemplos vistos. Finalmente el dataset de prueba test set constituye un conjunto de datos independiente que se reserva para evaluar el desempeño final del modelo, verificando que sea capaz de generalizar correctamente a información nueva y no utilizada previamente en el entrenamiento.

En visión artificial, predominan los datasets de imágenes, que consisten en colecciones organizadas de archivos en formatos como JPG o PNG. Estas imágenes pueden estar acompañadas de archivos de anotación que contienen información sobre las clases, coordenadas de cajas delimitadoras o máscaras de segmentación. Dependiendo de la tarea, el nivel de anotación puede variar desde etiquetas simples (clasificación) hasta anotaciones pixel a pixel de

la segmentación.

También existen datasets de texto, utilizados en procesamiento de lenguaje natural, que se almacenan en distintos formatos, estos pueden incluir etiquetas para tareas como análisis de sentimiento o clasificación semántica. De esta forma los datasets de audio y video contienen información temporal, lo que implica un tratamiento adicional relacionado con secuencias, frecuencias o fotogramas consecutivos.

Otra clasificación importante se basa en la finalidad dentro del proceso de entrenamiento. El dataset de entrenamiento se utiliza para que el modelo aprenda los patrones subyacentes ajustando sus parámetros internos. El dataset de validación permite ajustar hiperparámetros y monitorear el desempeño durante el entrenamiento para evitar sobreajuste, después el dataset de prueba se emplea para evaluar el rendimiento final del modelo en datos no vistos previamente, garantizando su capacidad de generalización.

En el ámbito de la visión artificial aplicada, como en sistemas de detección de objetos, los datasets deben cumplir criterios de calidad como diversidad de escenarios, variabilidad de iluminación, diferentes escalas, oclusiones y equilibrio entre clases, un dataset mal balanceado o poco representativo puede generar sesgos en el modelo afectando métricas como precisión y recall.

En síntesis, el dataset constituye el elemento fundamental sobre el cual se construye el modelo de aprendizaje automático, ya que la calidad, cantidad y representatividad de los datos influyen directamente en el desempeño, robustez y capacidad de generalización del sistema desarrollado.

Diagrama esquemático de un **dataset** en Dataverse 4.0

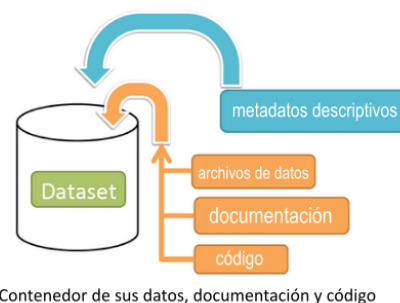


Figura 6. Diagrama de representación del DataSet [21].

#### V-H. Formato y estructura

Los datasets pueden almacenarse en diversos formatos según el tipo de información que contienen. En el caso de datos tabulares, se utilizan formatos como CSV, Excel o bases de datos SQL, organizados en filas y columnas. Para imágenes, los datos suelen almacenarse en carpetas con archivos en formatos como JPG, PNG o formatos especializados para visión artificial. Los datasets de texto pueden presentarse en archivos TXT, JSON, XML o en colecciones diseñadas para procesamiento de lenguaje natural. Finalmente, los datos de audio o video se almacenan en formatos como WAV, MP3, MP4, AVI, entre otros, según el tipo de análisis que se vaya a realizar.

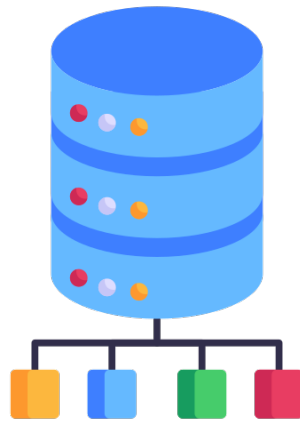


Figura 7. Diagrama de representación del DataSet [25].

#### V-I. Hardware Propuesto para el Sistema

Para la implementación del sistema de visión artificial se optó por ejecutar el modelo directamente en una computadora portátil, considerando su capacidad de procesamiento, flexibilidad de desarrollo y facilidad de integración con herramientas de entrenamiento y validación. Esta decisión permitió simplificar la arquitectura del sistema, eliminando la necesidad de dispositivos embebidos adicionales y facilitando la etapa de pruebas y ajustes del modelo.

La laptop utilizada dispone de un procesador multinúcleo que permite ejecutar tareas de preprocesamiento, entrenamiento e inferencia de manera eficiente. En caso de contar con unidad de procesamiento gráfico (GPU), el rendimiento del modelo mejora significativamente al acelerar las operaciones matriciales propias de las redes neuronales convolucionales. Esto resulta especialmente relevante en arquitecturas de detección de objetos como YOLO11s, que requieren múltiples operaciones de convolución y cálculo de mapas de características en tiempo reducido.

El uso de un entorno de ejecución local ofrece además ventajas en términos de depuración, monitoreo de métricas y control de recursos computacionales. A través de bibliotecas especializadas como PyTorch y el framework Ultralytics, es posible gestionar de forma directa el entrenamiento del modelo, ajustar hiperparámetros y visualizar en tiempo real indicadores como pérdida (loss), precisión, recall y mAP.

Desde el punto de vista operativo, esta configuración permite realizar inferencias en tiempo casi real, dependiendo de la resolución de entrada y la carga computacional. Asimismo, facilita la escalabilidad del sistema, ya que el modelo entrenado puede posteriormente migrarse a plataformas embebidas o sistemas de mayor capacidad si se requiere implementación en campo.

La elección de ejecutar el sistema en una laptop proporciona un entorno robusto, flexible y adecuado para el desarrollo, validación y optimización del modelo de visión artificial, garantizando un equilibrio entre capacidad de procesamiento, accesibilidad y eficiencia en el desarrollo experimental.

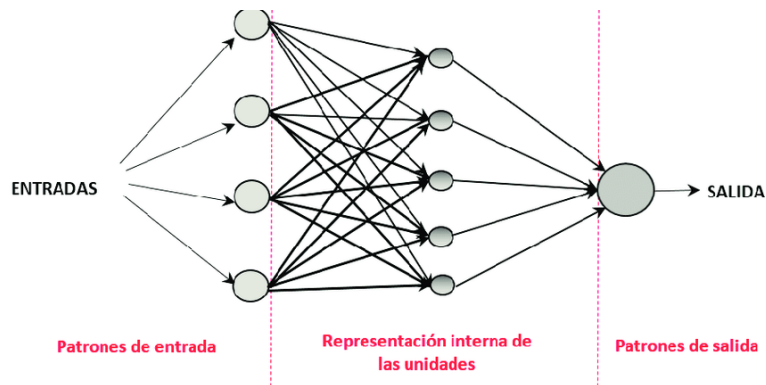


Figura 8. Esquema de la una red neuronal artificial [26].

### V-J. Cámara Web

Una cámara web es un dispositivo de captura de video digital diseñado para registrar imágenes en tiempo real y transmitir las a través de una computadora u otro equipo conectado a una red. Su funcionamiento se basa en un sensor de imagen, generalmente del tipo CCD (Charge-Coupled Device) que convierte la luz recibida en señales eléctricas. Estas señales son procesadas por un circuito interno que las transforma en datos digitales, los cuales pueden ser visualizados, almacenados o transmitidos mediante software especializado. En términos funcionales, la cámara web sirve para la comunicación audiovisual a distancia, siendo ampliamente utilizada en videoconferencias, educación virtual, telemedicina, monitoreo de seguridad y creación de contenidos multimedia. Desde una perspectiva académica y tecnológica, la cámara web representa un elemento fundamental en la interacción hombre-máquina y en el desarrollo de entornos virtuales colaborativos, al permitir la transmisión eficiente y en tiempo real de información visual a través de redes digitales.

Existen diversos tipos de cámaras web que se clasifican según su diseño, funcionalidad y aplicación tecnológica, cada una con características específicas que las hacen adecuadas para distintos entornos académicos e industriales. Las cámaras web integradas se encuentran incorporadas en dispositivos como laptops, monitores o tablets, y su principal ventaja radica en la conveniencia ya que no requieren instalación adicional y están optimizadas para videollamadas y educación virtual.

Las cámaras IP (Internet Protocol) operan de manera autónoma al conectarse directamente a una red local o a Internet lo que permite la transmisión de video sin necesidad de un ordenador, siendo ampliamente utilizadas en sistemas de videovigilancia y monitoreo remoto.

Las cámaras web HD y Full HD se distinguen por su alta resolución (720p o 1080p), proporcionando mayor nitidez y detalle en la imagen y resultan adecuadas para conferencias profesionales, grabaciones y transmisiones de alta calidad. En un nivel superior, las cámaras 4K y de alta gama ofrecen resoluciones superiores a 1080p, incorporando sensores avanzados y tecnologías de corrección de iluminación, enfoque automático y reducción de ruido, siendo empleadas en entornos profesionales como producción audiovisual e investigación científica.

Finalmente las cámaras web con micrófono o sensores integrados incluyen componentes adicionales, como micrófonos estéreo o sensores de movimiento, mejorando la interacción y la experiencia del usuario en contextos de comunicación y seguridad.



Figura 9. Tipos de cámara. [27].

#### V-K. Cerradura eléctrica con solenoide 12v dc

Una cerradura eléctrica con solenoide de 12V en corriente continua es un dispositivo electromecánico diseñado para controlar el acceso a un área específica mediante la activación eléctrica de un mecanismo de bloqueo. Su funcionamiento se basa en el principio del electromagnetismo: cuando se aplica una diferencia de potencial al solenoide, este genera un campo magnético que produce el desplazamiento lineal de un núcleo metálico interno, permitiendo liberar o bloquear el mecanismo de apertura.

El solenoide está compuesto por una bobina de alambre conductor enrollada alrededor de un núcleo ferromagnético, al circular corriente eléctrica a través de la bobina, se genera un campo magnético que atrae o empuja el núcleo móvil, provocando la retracción del pestillo. Cuando la alimentación se interrumpe, un resorte interno devuelve el núcleo a su posición original restableciendo el estado de bloqueo, este tipo de actuador es ampliamente utilizado en sistemas de control de acceso debido a su simplicidad, rapidez de respuesta y confiabilidad.

En aplicaciones de seguridad industrial, la cerradura eléctrica puede integrarse con sistemas de visión artificial para automatizar el control de ingreso a zonas de riesgo. En este contexto, el sistema de detección evalúa previamente condiciones como el uso adecuado de Equipos de Protección Personal. Solo cuando se cumplen los criterios establecidos, el sistema envía una señal eléctrica que activa el solenoide y permite el acceso al área restringida. De esta forma, el dispositivo físico se convierte en un mecanismo de ejecución que materializa la decisión tomada por el sistema inteligente.

Estos dispositivos operan típicamente a 12V DC y requieren una corriente suficiente para generar el campo magnético necesario para el desplazamiento del núcleo, debido a que el consumo de corriente puede superar la capacidad de ciertos microcontroladores o dispositivos de control, es común emplear módulos de relé o transistores como etapa de potencia para su activación segura.

La incorporación de una cerradura eléctrica con solenoide en el sistema no solo permite validar el cumplimiento de normas de seguridad antes del acceso, sino que también integra el componente físico con la lógica de control desarrollada mediante inteligencia artificial, fortaleciendo el enfoque de prevención en entornos industriales.



Figura 10. Cerradura electrica 12V. [28].

#### V-L. Tecnologías Clave

La revolución reciente en visión artificial se ha visto impulsada en gran medida por la aplicación de técnicas avanzadas de Deep Learning. El aprendizaje profundo (Deep Learning, DL) es una rama del aprendizaje automático que utiliza redes neuronales artificiales con múltiples capas, conocidas como redes profundas, para modelar abstracciones de alto nivel en los datos, lo que permite identificar patrones complejos y representaciones jerárquicas. Dentro de esta área, las redes neuronales convolucionales (CNNs, Convolutional Neural Networks) constituyen la columna vertebral de la visión artificial moderna; están diseñadas específicamente para procesar datos con una topología conocida, como imágenes, aplicando filtros convolucionales que permiten aprender jerarquías de características visuales, desde bordes simples hasta formas completas de objetos. Por su parte, las redes neuronales recurrentes (RNNs, Recurrent Neural Networks) y los transformers se utilizan para el procesamiento de secuencias de imágenes, como videos, y para tareas que requieren comprensión contextual y temporal, como el subtulado automático de imágenes (Image Captioning) o sistemas de preguntas y respuestas visuales, donde la relación entre los elementos en el tiempo o el contexto es fundamental.

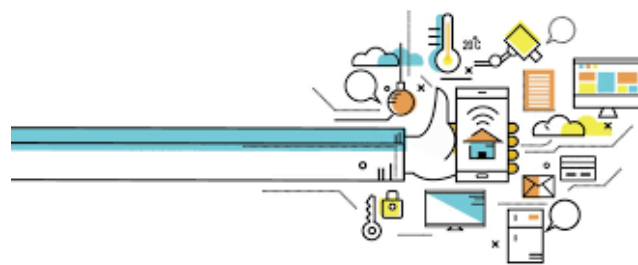


Figura 11. Tecnología y su entorno. [26].

### *V-M. Redes Neuronales Artificiales (RNA)*

Una red neuronal artificial es un modelo computacional inspirado en el funcionamiento del cerebro humano, diseñado para reconocer patrones, procesar información y aprender a partir de datos. Estas redes forman parte del campo del aprendizaje automático (machine learning) y constituyen la base del aprendizaje profundo (deep learning).

Desde una perspectiva matemática, una red neuronal puede entenderse como un sistema que transforma un conjunto de variables de entrada en una salida mediante combinaciones lineales ponderadas y funciones no lineales. Durante el proceso de entrenamiento, el modelo ajusta automáticamente sus parámetros internos con el objetivo de minimizar el error entre las predicciones generadas y los valores reales esperados. Este ajuste se realiza generalmente mediante algoritmos de optimización basados en gradiente, que permiten encontrar configuraciones de parámetros que mejor se adapten a los datos disponibles.

Una característica fundamental de las RNA es su capacidad de generalización, es decir, la habilidad de aprender patrones a partir de un conjunto de datos y aplicarlos posteriormente a información no vista previamente. Esta propiedad es especialmente relevante en problemas de visión artificial, donde las imágenes pueden presentar variaciones significativas en iluminación, escala, orientación y ruido. Gracias a su naturaleza adaptativa, las redes neuronales pueden identificar patrones subyacentes sin necesidad de definir manualmente reglas explícitas.

En términos de representación del conocimiento, las RNA permiten construir modelos altamente flexibles capaces de capturar dependencias complejas entre variables. Esto las convierte en herramientas poderosas para tareas como clasificación, regresión, reconocimiento de patrones y análisis de señales. Su desempeño mejora considerablemente cuando se dispone de grandes volúmenes de datos y capacidad computacional adecuada, lo que ha impulsado su aplicación en áreas como la visión por computadora, el procesamiento de lenguaje natural y la robótica.

En el contexto de la visión artificial aplicada, las redes neuronales artificiales permiten que el sistema aprenda automáticamente características relevantes de las imágenes, reduciendo la dependencia de técnicas de extracción manual y mejorando la precisión en tareas de detección y reconocimiento.

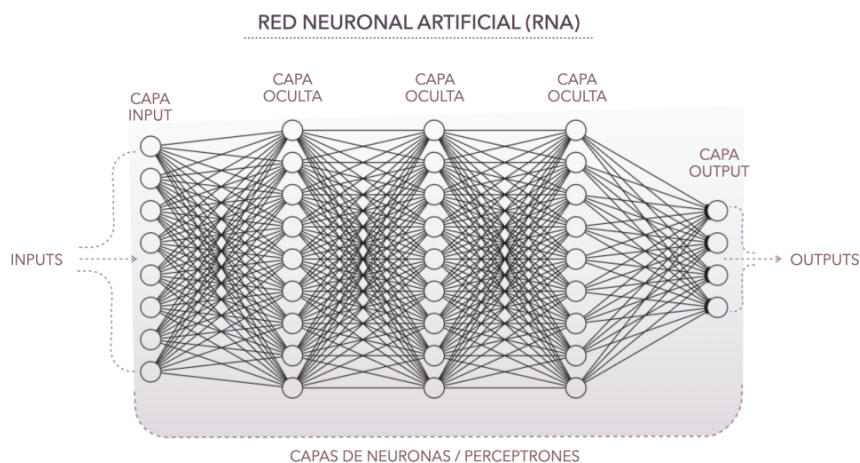


Figura 12. Red neuronal. [26].

### V-N. Estructura básica de RNA

Una red neuronal artificial se compone de unidades denominadas neuronas artificiales, organizadas en distintas capas con funciones específicas. La capa de entrada se encarga de recibir los datos iniciales, que pueden ser imágenes, texto o valores numéricos. Posteriormente, las capas ocultas procesan esta información mediante operaciones matemáticas, transformando los datos a través de funciones no lineales que permiten captar relaciones complejas.

La capa de salida produce el resultado final, ya sea una clasificación, predicción o decisión. Cada conexión entre neuronas está asociada a un peso ( $w$ ) que indica la relevancia de la señal transmitida, mientras que cada neurona aplica una función de activación, como la sigmoide, ReLU o tanh, que define su nivel de respuesta ante los estímulos recibidos.

la arquitectura se organiza en una capa de entrada, una o varias capas intermedias y una capa de salida. La capa de entrada recibe las variables o características del problema, las cuales pueden corresponder a valores numéricos, atributos estructurados o píxeles de una imagen. Las capas intermedias, también llamadas capas ocultas, realizan transformaciones sucesivas sobre la información recibida, generando representaciones internas que capturan relaciones cada vez más abstractas entre los datos. Finalmente, la capa de salida produce la predicción del modelo, ya sea una clase, una probabilidad o un valor continuo.

Cada conexión entre neuronas posee un parámetro ajustable denominado peso, el cual determina la influencia de una neurona sobre otra. Además, cada neurona suele incorporar un término adicional conocido como sesgo, que permite desplazar la función de activación y mejorar la flexibilidad del modelo. El conjunto total de pesos y sesgos constituye los parámetros entrenables de la red, cuyo ajuste es el objetivo principal durante el proceso de aprendizaje.

La profundidad de la red, definida por el número de capas intermedias, influye directamente en su capacidad de modelado. Redes más profundas pueden capturar patrones de mayor complejidad, ya que cada capa aprende representaciones jerárquicas basadas en la salida de la capa anterior. En aplicaciones de visión artificial, esta

característica resulta fundamental, pues las primeras capas suelen capturar características simples como bordes o texturas, mientras que las capas más profundas identifican formas y estructuras completas.

Asimismo, la estructura de la red define el flujo de información, que generalmente se realiza en sentido hacia adelante durante la inferencia, mientras que en el entrenamiento se aplica un proceso inverso para ajustar los parámetros mediante la propagación del error. Este mecanismo estructural permite que la red aprenda de manera iterativa hasta alcanzar un nivel de desempeño adecuado.

En síntesis, la estructura de una Red Neuronal Artificial determina su capacidad de representación, su complejidad computacional y su desempeño en tareas específicas. Una arquitectura bien diseñada permite equilibrar precisión, eficiencia y capacidad de generalización, aspectos fundamentales en sistemas de visión artificial como el desarrollado en este trabajo.

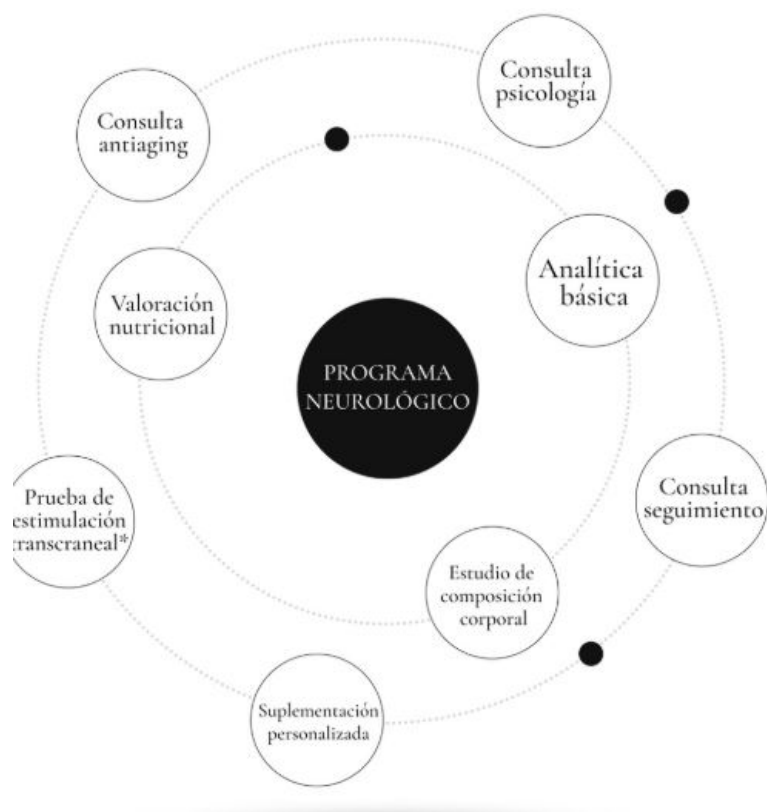


Figura 13. Red neuronal. [29].

#### V-Ñ. Tipos de redes neuronales

Existen diferentes arquitecturas de RNA, según la naturaleza de los datos y el objetivo del modelo:

1. Perceptrón Multicapa (MLP): utilizado para problemas de clasificación y regresión básicos.
2. Redes Convolucionales (CNN): especializadas en procesamiento de imágenes y video.

3. Redes Recurrentes (RNN y LSTM): diseñadas para datos secuenciales, como texto o series temporales.
4. Redes Generativas (GAN): empleadas para crear contenido nuevo como imágenes, música, etc.

La ecuación general es:

$$\text{Salida} = \sigma(Wx + b) \quad (1)$$

Donde:

1.  $W$  = pesos del modelo
2.  $x$  = entrada (imagen procesada)
3.  $b$  = sesgo
4.  $\sigma$  = función de activación (ejemplo: ReLU, Sigmoid)

#### V-O. Equipos de Protección Personal (EPP)

Los Equipos de Protección Personal (EPP) son dispositivos o elementos diseñados para proteger a los trabajadores de posibles riesgos que puedan amenazar su seguridad o salud durante sus actividades laborales. Su uso correcto es fundamental en entornos industriales para evitar accidentes graves o incluso mortales.

Entre los principales EPP utilizados se encuentran:

1. Cascos
2. Chalecos reflectivos
3. Guantes de seguridad
4. Botas con punta de acero
5. Lentes de seguridad
6. Protección auditiva



Figura 14. Trabajador industrial equipado correctamente con EPP [30].

#### V-P. Detección de EPP mediante Visión Artificial

La detección automática del uso de EPP se logra a través de técnicas avanzadas de procesamiento de imágenes e inteligencia artificial, como las redes neuronales convolucionales (CNN). Entre los algoritmos más utilizados para detección de objetos en imágenes se encuentra el modelo YOLO (You Only Look Once), que permite identificar

múltiples objetos en una sola pasada por la red [22].

Su objetivo principal es identificar en tiempo real la presencia o ausencia de elementos de protección como cascos, chalecos reflectivos, gafas de seguridad o guantes, a partir del análisis de imágenes o secuencias de video capturadas por cámaras instaladas en áreas industriales.

Este problema se modela como una tarea de detección multiclase donde el sistema debe localizar espacialmente cada objeto de interés mediante cajas delimitadoras y, simultáneamente, asignar una etiqueta correspondiente a la categoría del EPP, el modelo debe ser capaz de operar bajo condiciones variables de iluminación, ángulos de captura, oclusiones parciales y presencia de múltiples trabajadores en una misma escena. Estas variaciones introducen desafíos relacionados con la robustez y la capacidad de generalización del sistema.

La implementación de esta solución se basa en modelos de aprendizaje profundo entrenados con datasets previamente etiquetados, en los cuales cada instancia de EPP está anotada con coordenadas espaciales y su respectiva clase, durante el entrenamiento, el modelo aprende representaciones visuales discriminativas que le permiten diferenciar entre objetos similares y reducir la probabilidad de falsos positivos y falsos negativos, la calidad del dataset, la diversidad de escenarios y el equilibrio entre clases influyen directamente en el desempeño del sistema.

En diversas aplicaciones de detección de EPP no solo busca identificar la presencia del equipo, sino también garantizar su uso correcto en contextos dinámicos, esto implica que el sistema debe procesar imágenes en tiempo casi real, manteniendo un equilibrio entre precisión y velocidad de inferencia. En este sentido, arquitecturas optimizadas como YOLO11s resultan adecuadas, ya que permiten realizar detecciones rápidas sin comprometer significativamente la exactitud.

Desde una perspectiva operativa la detección automática de EPP contribuye a la reducción de riesgos laborales, optimiza los procesos de supervisión y fortalece la cultura preventiva dentro de las organizaciones, esto permite generar registros digitales que pueden utilizarse para auditorías de seguridad, análisis estadístico de cumplimiento y mejora continua de protocolos internos.

La detección de EPP mediante visión artificial integra técnicas avanzadas de aprendizaje profundo, procesamiento de imágenes y análisis en tiempo real para ofrecer una solución tecnológica orientada a la seguridad industrial, combinando eficiencia computacional con alta capacidad de reconocimiento visual.



Figura 15. Detección de EPP con modelo YOLO [31].

### V-Q. Delimitación de Zonas de Riesgo

Dentro del entorno industrial, es necesario delimitar zonas seguras y zonas de riesgo. La delimitación puede realizarse mediante el análisis del video captado por las cámaras, asociando regiones específicas del espacio a áreas peligrosas.

Desde una perspectiva técnica, la delimitación puede implementarse mediante la definición de regiones de interés (ROI, Regions of Interest) dentro del campo visual de la cámara, estas regiones se modelan a partir de coordenadas espaciales en la imagen y funcionan como áreas de supervisión activa. Cuando un objeto detectado por ejemplo una persona sin EPP— ingresa o permanece dentro de una zona previamente definida como crítica, el sistema puede generar alertas automáticas o registrar el evento para análisis posterior.

La integración de modelos de detección de objetos con la delimitación de zonas permite no solo identificar la presencia de trabajadores, sino también analizar su ubicación relativa dentro del entorno. Esto implica realizar cálculos espaciales basados en las coordenadas de las cajas delimitadoras y compararlas con los límites geométricos establecidos para cada área de riesgo. En sistemas más avanzados, pueden incorporarse técnicas de calibración de cámara o estimación de perspectiva para mejorar la precisión espacial y aproximar distancias reales.

La delimitación de zonas de riesgo aporta una capa adicional de inteligencia al sistema de visión artificial, ya que transforma la detección pasiva en una supervisión contextualizada, no se trata únicamente de identificar objetos, sino de comprender su relación con el entorno y evaluar si se están incumpliendo protocolos de seguridad. Esta capacidad resulta especialmente relevante en escenarios donde el riesgo no depende únicamente de la presencia de un objeto, sino de su ubicación dentro del espacio operativo.

En términos generales la implementación de zonas virtuales reduce la necesidad de supervisión humana constante y permite una respuesta inmediata ante comportamientos inseguros. Asimismo, facilita la generación de métricas relacionadas con la frecuencia de incumplimientos en áreas específicas, contribuyendo a la toma de decisiones

orientadas a la mejora de la seguridad industrial.

La delimitación de zonas de riesgo mediante visión artificial integra análisis espacial, detección automática y monitoreo contextual, permitiendo convertir un sistema de reconocimiento visual en una herramienta activa de prevención y control dentro del entorno laboral.



Figura 16. Representación se zonas seguras [23].

#### *V-R. Impacto del Proyecto*

La implementación del sistema propuesto tendrá un impacto positivo significativo en diversos aspectos de la seguridad laboral. En primer lugar, contribuirá a la reducción de accidentes, al garantizar que los trabajadores utilicen correctamente los equipos de protección personal antes de acceder a zonas de riesgo. Esto permitirá la optimización de los procesos de seguridad, automatizando la supervisión y reduciendo la dependencia de inspecciones manuales. Este enfoque también se traduce en un ahorro económico, al disminuir la ocurrencia de incidentes y los costos asociados a lesiones, seguros y pérdidas de productividad.

Desde el punto de vista económico, la disminución de accidentes laborales puede traducirse en una reducción de costos asociados a indemnizaciones, interrupciones de producción y sanciones regulatorias. Además la implementación de tecnologías basadas en inteligencia artificial posiciona a la organización dentro de un enfoque de innovación tecnológica, incrementando su competitividad y modernización en procesos industriales.

En el ámbito académico y tecnológico, el proyecto demuestra la viabilidad de integrar modelos de aprendizaje profundo, como YOLO11s, en aplicaciones reales de seguridad industrial, esto abre la posibilidad de futuras mejoras, como la incorporación de análisis predictivo, integración con sistemas de gestión empresarial o expansión hacia otros elementos de control dentro del entorno laboral.

El uso de herramientas de visión artificial promueve una cultura preventiva dentro de la organización, ya que refuerza la importancia del cumplimiento de normas de seguridad mediante mecanismos objetivos y constantes de evaluación. La presencia de un sistema automatizado no solo detecta incumplimientos, sino que también genera conciencia sobre la necesidad de mantener prácticas seguras de manera permanente.

Finalmente el impacto del proyecto trasciende la implementación técnica del modelo, ya que contribuye a mejorar la seguridad, la eficiencia operativa y la adopción de tecnologías emergentes en el contexto industrial.



Figura 17. Impacto esperado en la reducción de accidentes [25].

## VI. MARCO METODOLÓGICO

A continuación se presenta el funcionamiento del proyecto a través de un diagrama de flujo

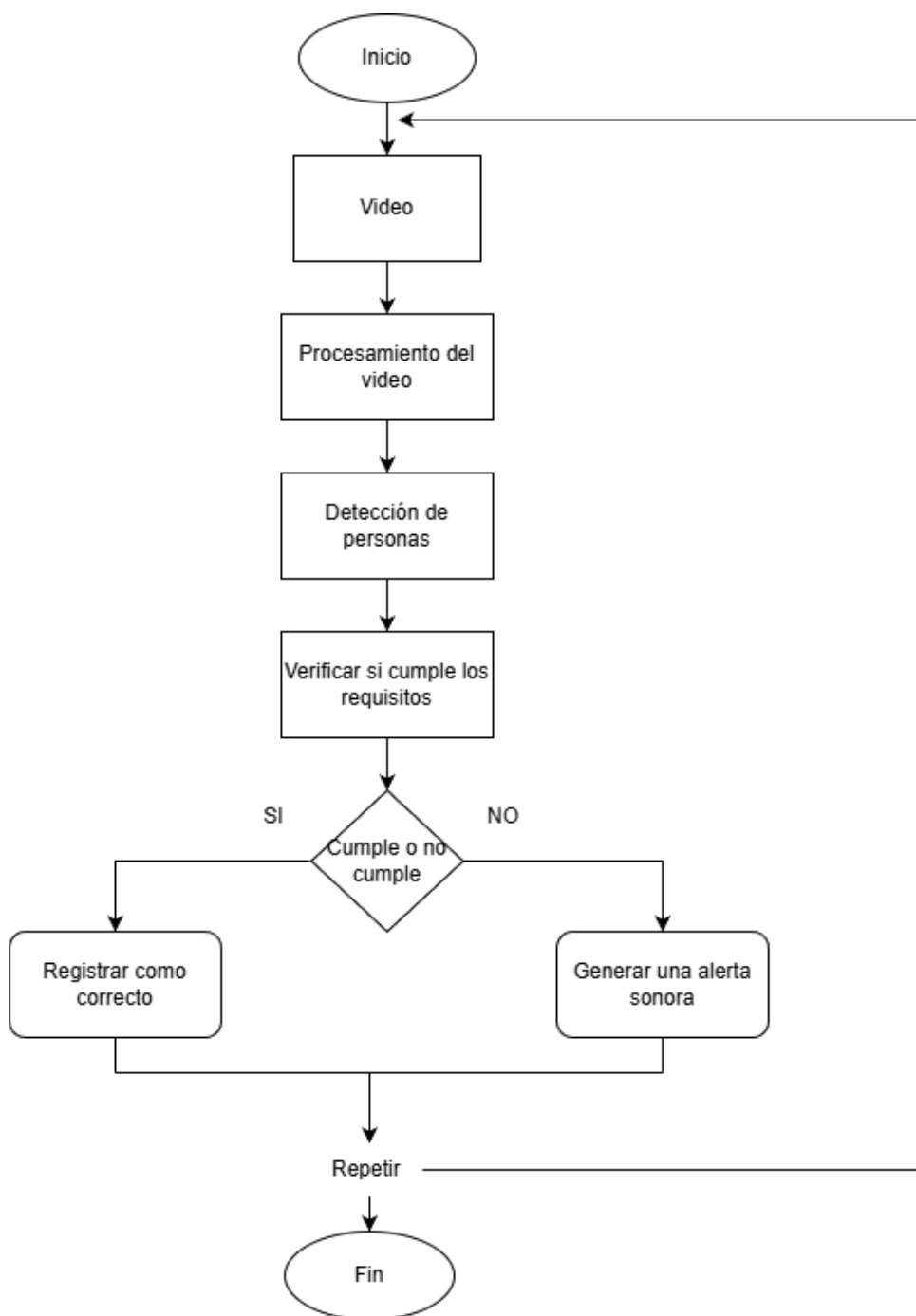


Figura 18. Diagrama de flujo del funcionamiento. Elaborado por autores

## VI-A. Funcionamiento de visual studio

En esta imagen se presenta el funcionamiento del sistema de visión, en primer lugar se captura la escena y el fotograma, donde se ejecuta el modelo de detección. El módulo de inferencia procesa la imagen, determina las coordenadas de las cajas delimitadoras y asigna la etiqueta correspondiente al EPP usado, finalmente estos resultados se envían a la interfaz de salida para su visualización en tiempo real.

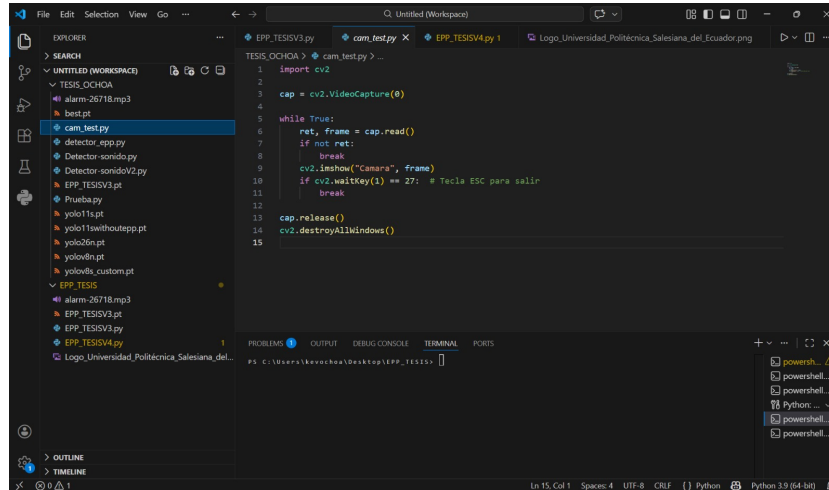


Figura 19. Funcionamiento visual. Elaborado por autores

## VI-B. Preparación de entorno

Se habilita un entorno mediante Visual Studio Code, donde se programa en Python un sistema de detección de EPP en tiempo real utilizando un modelo llamado YOLO. En este entorno se guardan los archivos y se aplica una lógica clave que define los elementos de seguridad, como casco, gafas y chaleco. Al detectar la ausencia de alguno de estos elementos, se activará una alarma sonora, todo esto en tiempo real para evaluar los resultados obtenidos.

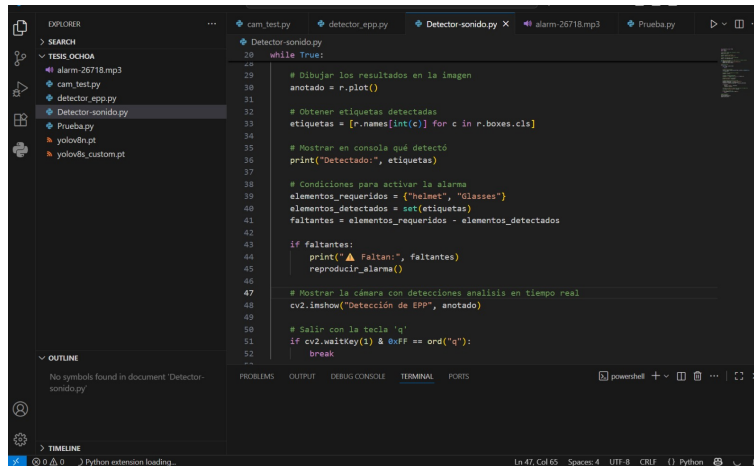
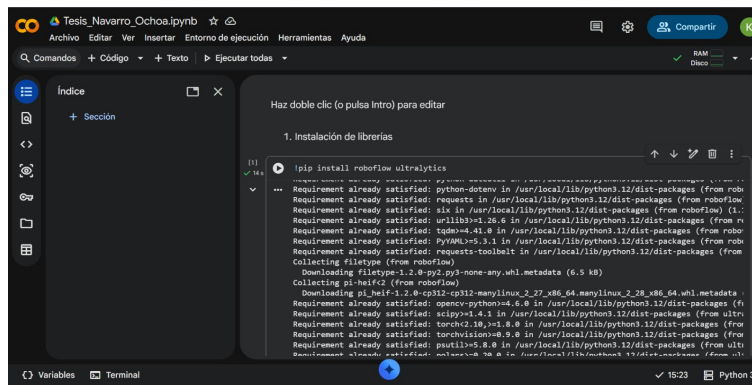


Figura 20. Entorno por visual studio code. Elaborado por autores

Para el desarrollo del sistema de visión artificial se empleó un entorno de programación basado en Python, utilizando notebooks interactivos que permiten la ejecución secuencial de código, la visualización de resultados y

la documentación del proceso. En la Figura se muestra la preparación del entorno de trabajo, donde se realiza la instalación de librerías especializadas mediante el gestor de paquetes pip.



```
pip install roboflow ultralytics
Requirement already satisfied: python-dotenv in /usr/local/lib/python3.12/dist-packages (from roboflow)
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from roboflow)
Requirement already satisfied: six in /usr/local/lib/python3.12/dist-packages (from roboflow)
Requirement already satisfied: urllib3<1.26.8 in /usr/local/lib/python3.12/dist-packages (from roboflow)
Requirement already satisfied: tqdm<4.41.8 in /usr/local/lib/python3.12/dist-packages (from roboflow)
Requirement already satisfied: PyYAML<5.3.1 in /usr/local/lib/python3.12/dist-packages (from roboflow)
Requirement already satisfied: requests-toolbelt in /usr/local/lib/python3.12/dist-packages (from roboflow)
Collecting filetype
  Downloading filetype-1.2.0-py2.py3-none-any.whl.metadata (6.5 kB)
Collecting pi-heif2
  Downloading pi_heif-1.2.0-cp312-cp312-manylinux_2_27_x86_64_manylinux_2_28_x86_64.whl.metadata (
Requirement already satisfied: opencv-python<4.6.0 in /usr/local/lib/python3.12/dist-packages (fr
Requirement already satisfied: scipy<1.4.1 in /usr/local/lib/python3.12/dist-packages (from ultr
Requirement already satisfied: torch<2.10.0 in /usr/local/lib/python3.12/dist-packages (from
Requirement already satisfied: torchvision<0.9.0 in /usr/local/lib/python3.12/dist-packages (from
Requirement already satisfied: pillow<5.8.0 in /usr/local/lib/python3.12/dist-packages (from ult
Requirement already satisfied: numpy<1.24.0 in /usr/local/lib/python3.12/dist-packages (from ult
```

Figura 21. Entrenamiento del modelo. Elaborado por autores

### VI-C. Elaboración de Dataset

La conformación del conjunto de datos representa el pilar fundamental del sistema, ya que es a partir de él que el modelo adquiere los patrones visuales de interés. Para este proyecto, se recopilamos imágenes provenientes de entornos controlados, buscando abarcar una amplia variedad de fondos, ángulos y condiciones de iluminación. Cada imagen fue etiquetada de manera coherente según la categoría correspondiente a los EPP, garantizando así que el conjunto de entrenamiento sea representativo y permita una adecuada generalización del modelo en aplicaciones reales.

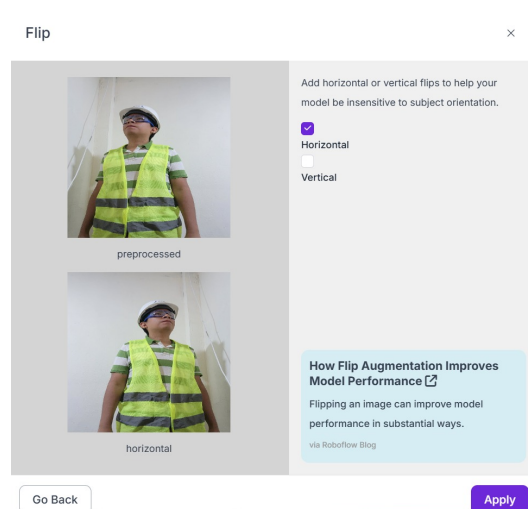


Figura 22. Uso del Roboflow. Elaborado por autores

Durante la elaboración del dataset se aplicaron técnicas de aumentación de datos con el fin de incrementar la diversidad del conjunto de entrenamiento y mejorar la capacidad de generalización del modelo. En la Figura se muestra la aplicación del volteo horizontal mediante la plataforma Roboflow, donde a partir de una imagen original se genera una nueva muestra con orientación invertida.

VI-C1. *Etiquetado de imágenes:* Se anotan manualmente las instancias de cada EPP que se use en cada caja correspondiente, se estableció un protocolo de consistencia como el tamaño mínimo de la caja y poder evitar solapamientos innecesarios, se realizó una validación por muestreo para mantener el control de calidad de las anotaciones.

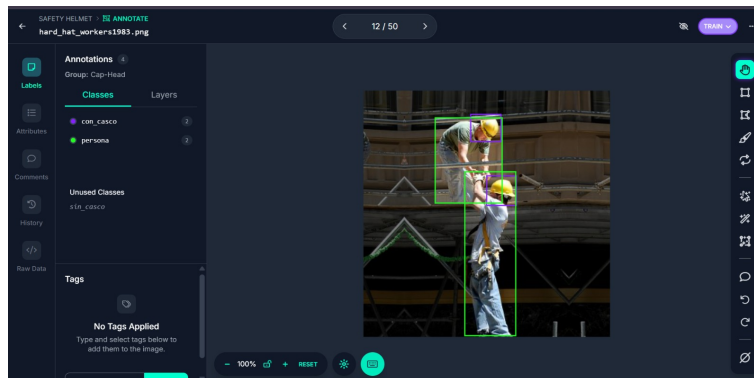


Figura 23. Etiquetado de EPP. Elaborado por autores

La correcta delimitación de cada instancia garantiza que el modelo aprenda a diferenciar adecuadamente los elementos de interés incluso cuando coexisten múltiples clases en una misma imagen. Asimismo, la calidad del etiquetado influye directamente en el desempeño del modelo, ya que anotaciones imprecisas pueden generar errores durante el proceso de entrenamiento y afectar negativamente métricas como la precisión y el recall.

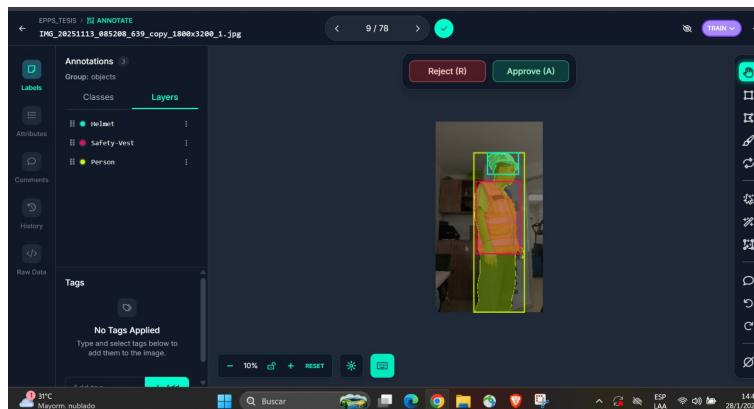


Figura 24. Etiquetado de EPP. Elaborado por autores

VI-C2. *Conformación del Dataset:* En la primera fase del proyecto se seleccionaron manualmente imágenes que contienen las tres categorías de interés: casco, chaleco, gafas industriales. Estas fotografías fueron tomadas en diversas condiciones de iluminación, ángulos y fondos para garantizar que el modelo de inteligencia artificial observe suficientes variaciones. Esta diversidad en el dataset inicial es clave para lograr una buena generalización en el modelo de detección.

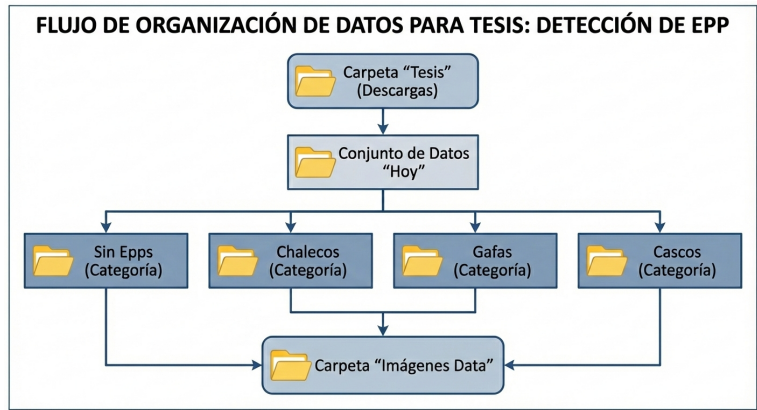


Figura 25. DataSet. Elaborado por autores

El Flujo de Organización de Datos detalla la estructura lógica y jerárquica empleada para gestionar el conjunto de imágenes, al segmentar la información en carpetas específicas para cada clase como Cascos, Chalecos y Gafas, se facilita la implementación de scripts de carga y el preprocesamiento de datos necesario para el entrenamiento. La organización de los archivos del dataset se estructuró de manera lógica y reproducible: cada carpeta indica el tipo de EPP (casco, chaleco y gafas.) y se subdivide en fases de “entrenamiento” o “prueba”. Esto permite automatizar los procesos de carga, etiquetado y validación, además de mantener claridad en el flujo de datos. Una estructura bien definida facilita la labor del investigador y reduce posibles errores de asignación de categoría o fase.

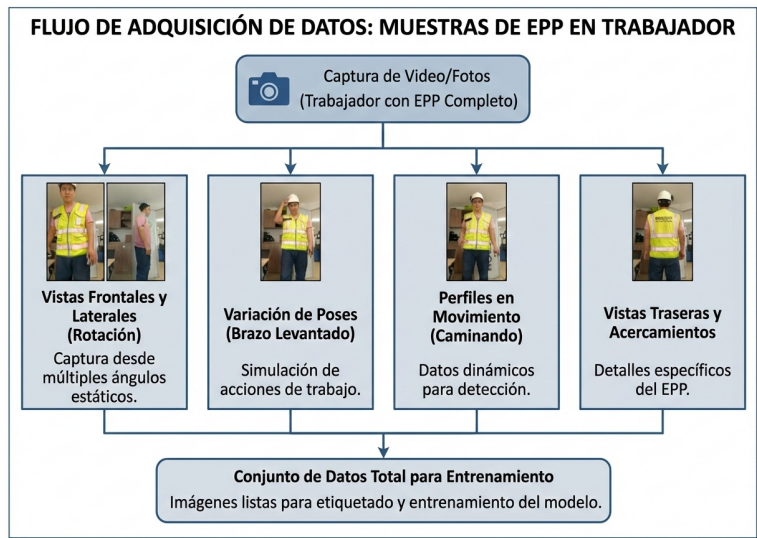


Figura 26. Imágenes del DataSet. Elaborado por autores

Una vez organizado el dataset, se visualizó internamente su contenido para comprobar que las anotaciones estuvieran correctas y que cada instancia correspondiera a la categoría adecuada. Esta revisión visual incluye comprobar caja delimitadora, etiquetas y la consistencia de los datos. Al preparar el dataset de esta manera, se garantiza que el modelo de detección realice inferencias con bases sólidas y que el sistema final desarrollado tenga mayor robustez operativa.

Este enfoque es fundamental para que el algoritmo aprenda a reconocer el equipo de protección personal EPP bajo

distintas perspectivas y poses, minimizando falsos negativos y asegurando que el sistema sea capaz de generalizar correctamente el reconocimiento en escenarios reales de supervisión industrial.

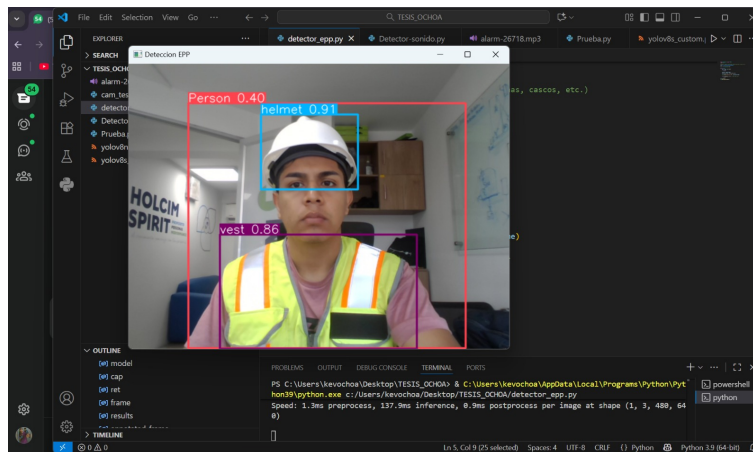


Figura 27. Versión Beta. Elaborado por autores

#### VI-D. Entrenamiento del modelo

Se opta por YOLOv11s debido a que ofrece un equilibrio eficiente entre velocidad de inferencia y precisión, lo que lo convierte en una alternativa adecuada para entornos con recursos computacionales limitados y necesidad de respuesta en tiempo casi real. Esta versión ligera permite mantener un buen desempeño sin comprometer significativamente el rendimiento del sistema. El proceso de entrenamiento se lleva a cabo utilizando el archivo de configuración data.yaml, donde se definen las rutas del dataset y las clases, junto con los hiperparámetros predeterminados proporcionados por la implementación de Ultralytics.

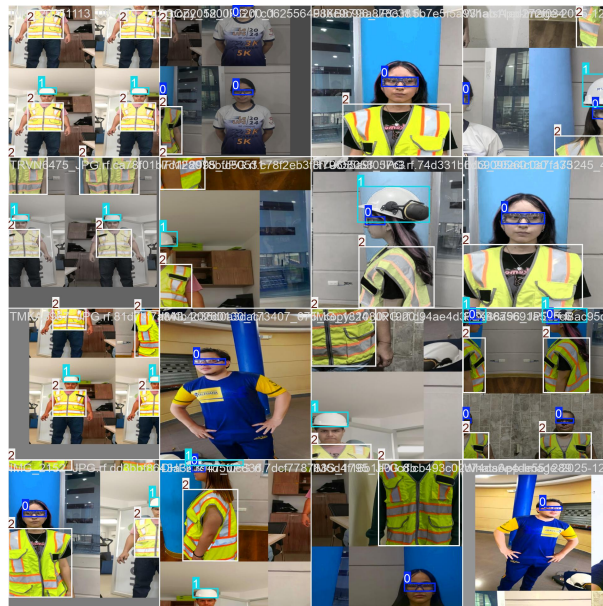


Figura 28. Entrenamiento del modelo. Elaborado por autores

En esta etapa se realizó la preparación del entorno de trabajo para el entrenamiento del modelo de detección de objetos, utilizando Google Colaboratory como plataforma de ejecución. Como se observa en la Figura, se cargó correctamente el conjunto de datos previamente etiquetado y exportado desde Roboflow, el cual fue organizado automáticamente en las carpetas de entrenamiento, validación y prueba, siguiendo la estructura requerida por el modelo YOLO11s.

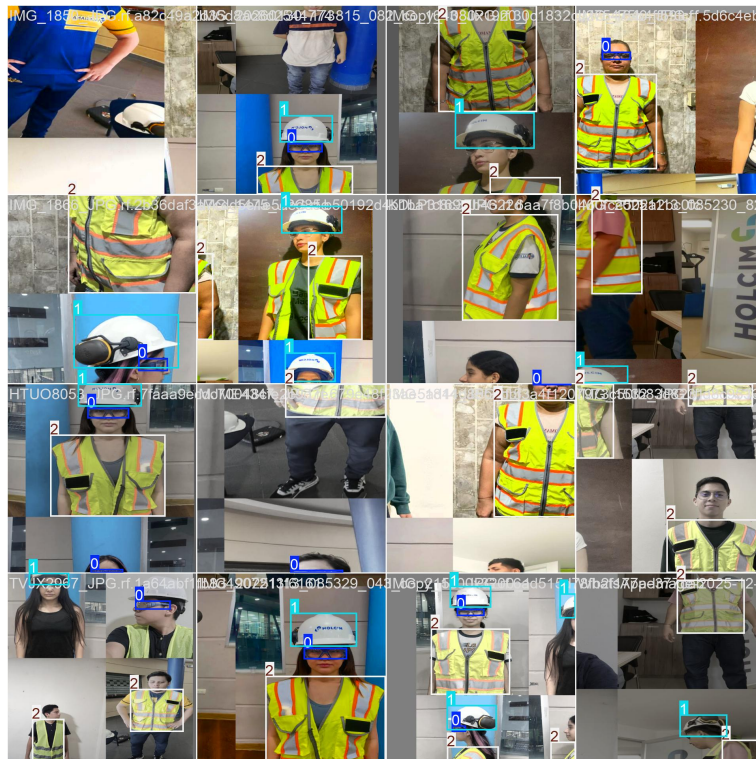


Figura 29. Entrenamiento del modelo. Elaborado por autores

Seguidamente se procedió a la carga del modelo base YOLO11s mediante la librería Ultralytics, utilizando pesos preentrenados. Esta estrategia permite aprovechar el aprendizaje previo del modelo sobre grandes conjuntos de datos, facilitando un proceso de transferencia de aprendizaje que mejora la eficiencia y reduce el tiempo de entrenamiento. La correcta integración del dataset y la carga del modelo base son pasos esenciales para garantizar un entrenamiento estable y un desempeño adecuado en la detección de los objetos de interés.

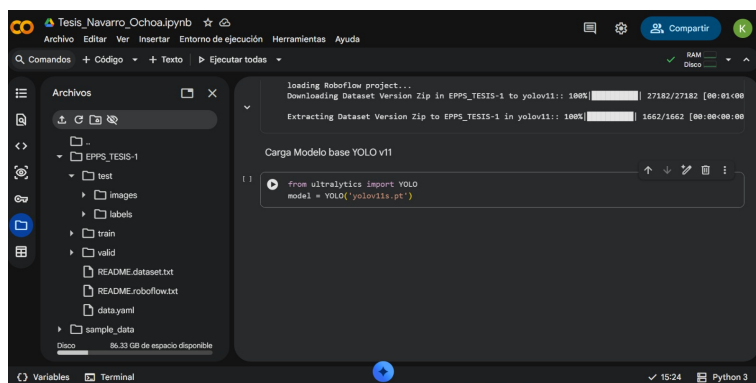


Figura 30. Entrenamiento del modelo. Elaborado por autores

## VI-E. Métricas de evaluación

*VI-E1. Precisión general (Accuracy):* La métrica de accuracy se emplea como primera medida de evaluación, determinando el porcentaje de predicciones correctas realizadas por el modelo en relación con el total de predicciones efectuadas. Por ejemplo, si el modelo clasifica correctamente 90 imágenes de un total de 100, la precisión alcanzaría un 90 %

$$\text{Exactitud} = \frac{\text{Clasificaciones Correctas}}{\text{Total de Clasificaciones}}$$

*VI-E2. Precisión por clase:* En esta métrica de evaluación se pretende medir la precisión del algoritmo específicamente para los dos EPP, centrándose en la identificación de falsos positivos. A menor cantidad de falsos positivos detectados por el algoritmo, mayor será su precisión

*VI-E3. Recall o Exhaustividad:* Esta métrica de evaluación se utiliza para medir el porcentaje de verdaderos positivos de una clase específica de EPP, poniendo énfasis en la identificación de falsos negativos. Un valor alto de recall indica que el modelo es capaz de detectar casi todos los casos positivos reales.

$$\text{Precisión} = \frac{\text{Verdaderos Positivos (TP)}}{\text{Verdaderos Positivos (TP) + Falsos Positivos (FP)}}$$

## VI-F. Pruebas en modelo pre-entrenado

Con el objetivo de entender el flujo completo del sistema y evaluar de manera temprana el funcionamiento del modelo, se llevaron a cabo pruebas exploratorias de la interfaz y de precisión sobre un prototipo inicial. Estas pruebas no tienen como finalidad determinar el desempeño final, sino identificar posibles patrones de error y guiar ajustes en el dataset, en los umbrales de decisión y en la configuración de la arquitectura del modelo.

Se analizaron imágenes que contenían múltiples EPP dentro de la misma escena, y se observó que la predicción del modelo se vuelve inconsistente frente a situaciones de solapamiento, cambios de escala y fondos heterogéneos. Este comportamiento es consistente con prototipos que utilizan clasificación de imagen completa, los cuales no identifican instancias de forma explícita y, como consecuencia, pueden confundir o mezclar señales provenientes de diferentes objetos.

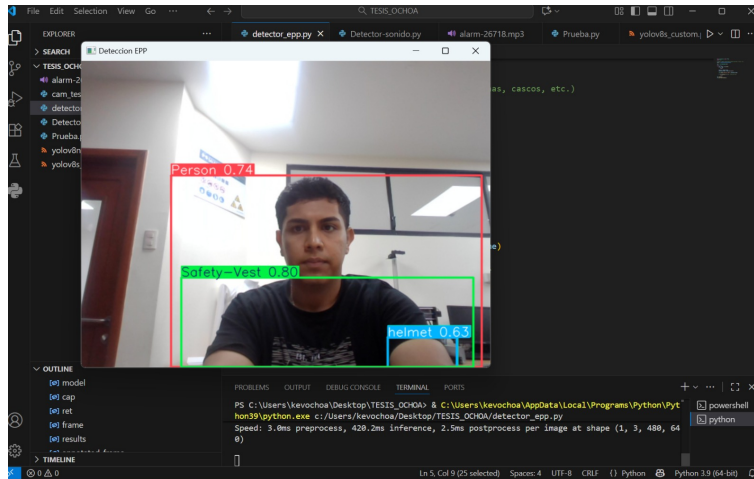


Figura 31. Prueba visual. Elaborado por autores

Se utilizó una imagen nítida que contenía únicamente un EPP, bajo estas condiciones, la salida del modelo mostró un alto nivel de confianza, demostrando que la claridad de los contornos, el contraste con el fondo y la ausencia de obstrucciones contribuyen significativamente a mejorar la precisión.

Se enfocó la evaluación en gafas bajo un escenario comparable, logrando también altos porcentajes de acierto. Estos resultados confirman que, cuando las imágenes están bien encuadradas y contienen un único sujeto, el prototipo ofrece un desempeño estable y confiable.

## VI-G. Resultados del entrenamiento

*VI-G1. Resultados cualitativos del entrenamiento:* Se muestran composiciones de imágenes con las predicciones generadas por el modelo sobre el conjunto de validación o prueba, en las cuales cada EPPS detectada está resaltada mediante cuadros delimitadores diferenciados por colores según la clase correspondiente. Las imágenes incluyen escenas variadas de muchos entornos diferentes, lo que permite observar las ventajas y el límite del modelo más allá de los indicadores numéricos.

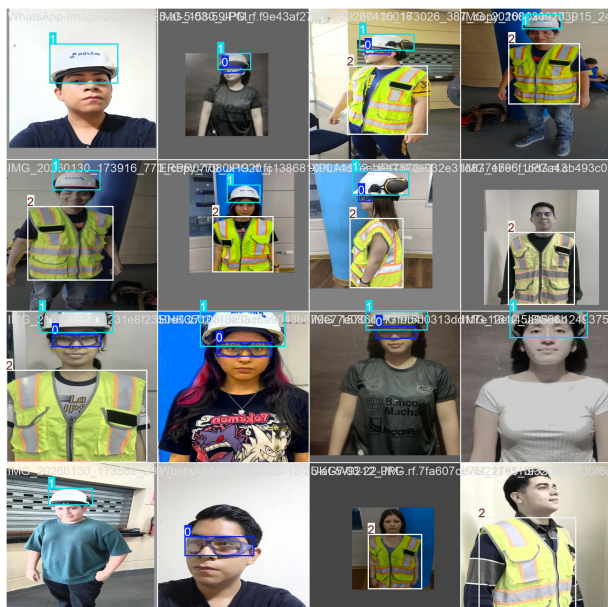


Figura 32. Predicciones en validación. Elaborado por autores

Se aprecia que las detecciones se ajustan adecuadamente a los contornos de los EPP, incluso en diferentes ángulos, escalas y condiciones de iluminación, lo que evidencia una adecuada etapa de etiquetado y una buena capacidad de aprendizaje durante el entrenamiento.

La consistencia en la localización y clasificación de los objetos dentro del batch refleja una correcta convergencia del modelo y una adecuada representación de las características visuales relevantes en el conjunto de entrenamiento.

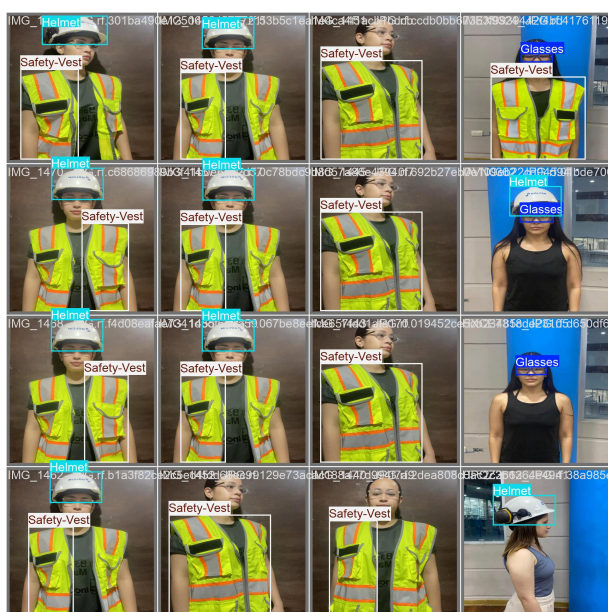


Figura 33. Predicciones en validación. Elaborado por autores

En este apartado se observa una adecuada correspondencia entre las regiones anotadas y los objetos reales, con cajas ajustadas a los contornos del casco, chaleco reflectivo y lentes de seguridad, la variabilidad en orientación

frontal y lateral, iluminación y presencia parcial de EPP contribuye a una mayor robustez del modelo, ya que el conjunto de validación presenta escenarios diversos.

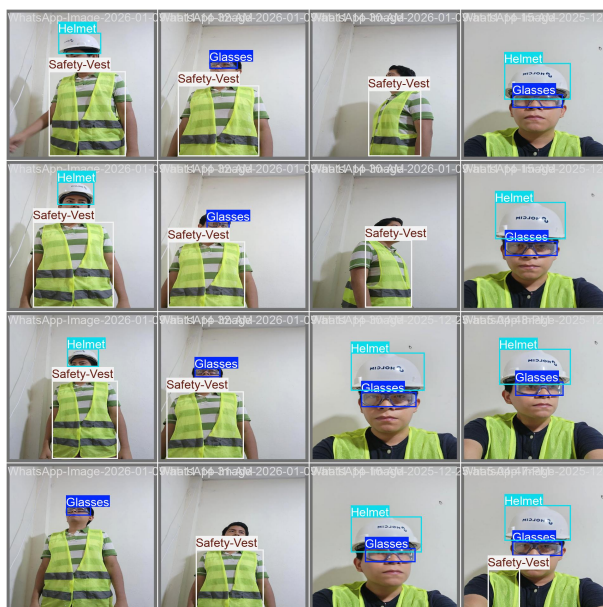


Figura 34. Predicciones en validación. Elaborado por autores

En esta imagen se observa consistencia en el etiquetado bajo distintas variaciones de pose frontal y lateral, encuadre y escala, incluyendo casos donde los objetos aparecen parcialmente visibles, las cajas presentan un ajuste preciso a los contornos del casco, chaleco y lentes, lo que indica una correcta definición de las coordenadas x, y, width, height normalizadas empleadas durante el entrenamiento.

## VII. RESULTADOS

### VII-A. Desempeño cuantitativo global

La curva Precision-Recall evidencia el desempeño global del modelo en cada clase. Se observan valores muy altos, con un mAP 0.5 aproximado de 0.99, lo que indica un excelente rendimiento de detección. Las curvas se mantienen cercanas al vértice superior derecho, reflejando que el modelo logra simultáneamente alta precisión y alto recall. Esto significa que no solo detecta correctamente la mayoría de los objetos reales, sino que además comete muy pocos errores al clasificarlos.

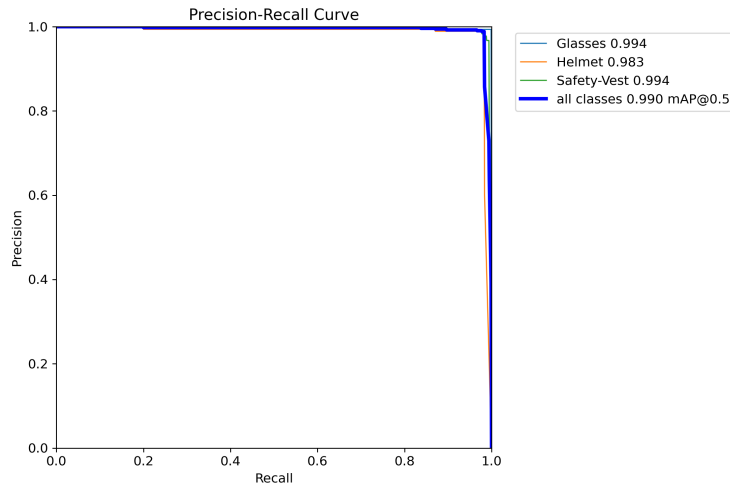


Figura 35. Curva Precisión-Recall sobre el conjunto de prueba. Elaborado por autores

Las gráficas muestran la evolución de las pérdidas (box-loss, cls-loss, dfl-loss) tanto en entrenamiento como en validación. Se observa una disminución progresiva y estable, lo que indica convergencia adecuada sin sobreajuste significativo.

Las métricas de precisión recall y mAP aumentan rápidamente en las primeras épocas y luego se estabilizan en valores cercanos a 0.98–0.99. Esto demuestra que el modelo YOLO11s logra aprender eficientemente las características del conjunto de datos y mantener un desempeño consistente.

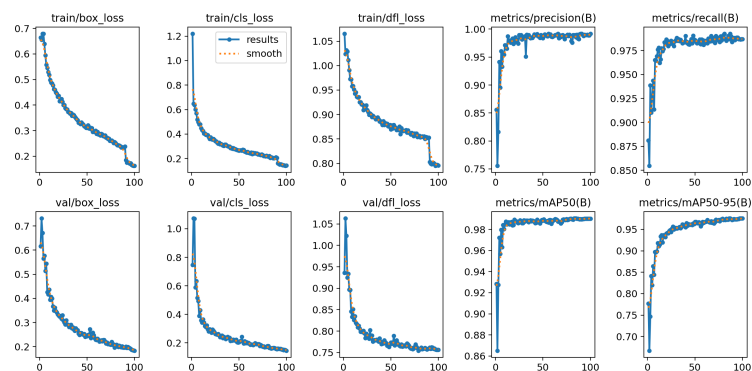


Figura 36. Evolución de métricas durante el entrenamiento de mAP. Elaborado por autores

### VII-B. Matriz de confusión

Al normalizar la matriz, se aprecia que las clases principales presentan valores cercanos a 0.98 – 1.00 de clasificación correcta lo que confirma la alta precisión del sistema, las pequeñas proporciones de error corresponden principalmente a detecciones clasificadas como fondo. Esto refuerza la estabilidad del modelo y su bajo nivel de confusión entre categorías.

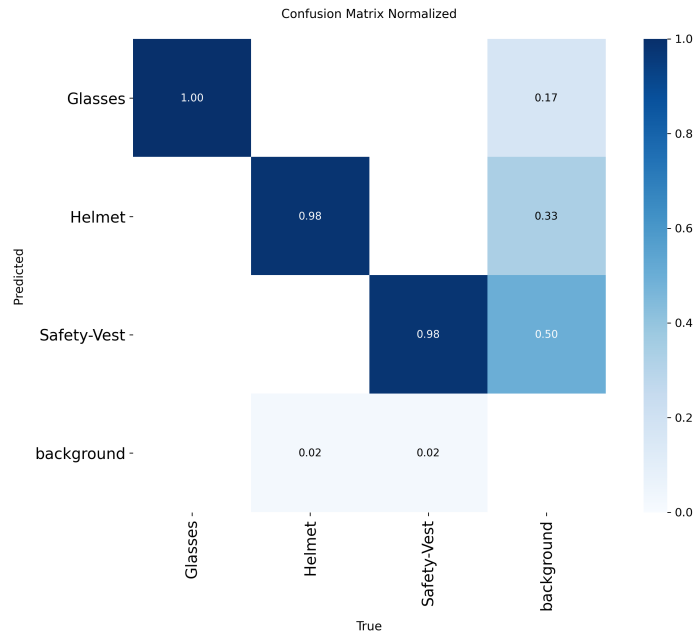


Figura 37. Matriz de confusión normalizada sobre el conjunto de prueba. Elaborado por autores

### VII-C. Distribución de etiquetas

Esta imagen presenta la distribución de instancias por clase y la ubicación espacial de las cajas delimitadoras. Se observa un número equilibrado de ejemplos:

1. Glasses: 933
2. Helmet: 1044
3. Safety-Vest: 1041

Además los mapas de calor indican que los objetos suelen concentrarse en regiones específicas de la imagen, y las dimensiones de las cajas muestran cierta variabilidad en tamaño. Esta distribución balanceada favorece un entrenamiento estable y reduce el sesgo entre clases.

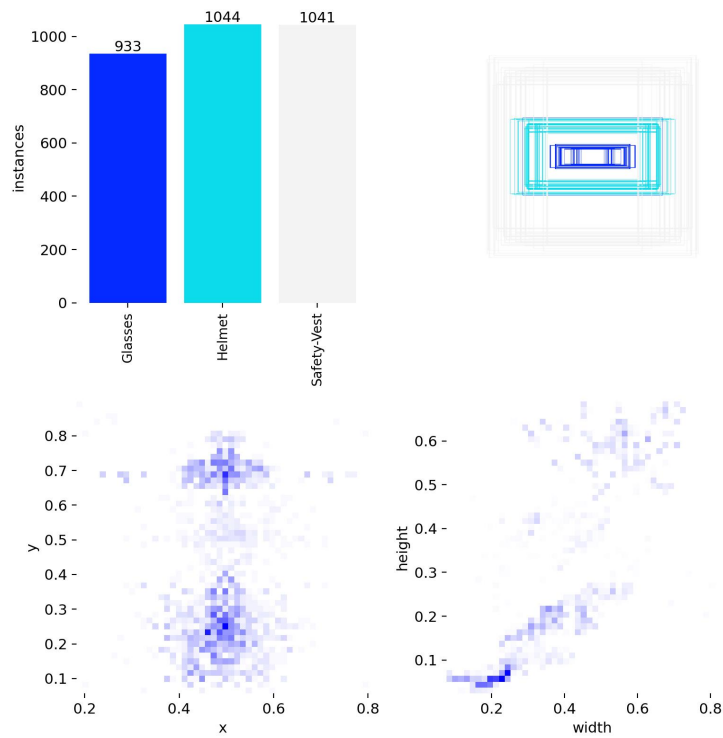


Figura 38. Distribución de etiquetas sobre el conjunto de prueba. Elaborado por autores

#### VII-D. Umbral de confianza y compromiso Precisión-Recall

La curva Precisión-Recall permite evaluar el rendimiento general del modelo para cada una de las clases. Se aprecian valores considerablemente elevados, con un  $mAP@0.5$  cercano a 0.99, lo que demuestra un desempeño sobresaliente en la tarea de detección. Las curvas se sitúan próximas al extremo superior derecho del gráfico, indicando que el modelo mantiene simultáneamente niveles altos de precisión y exhaustividad. En términos prácticos, esto implica que identifica la mayoría de los objetos reales presentes en la imagen y, al mismo tiempo, genera una cantidad mínima de clasificaciones incorrectas.

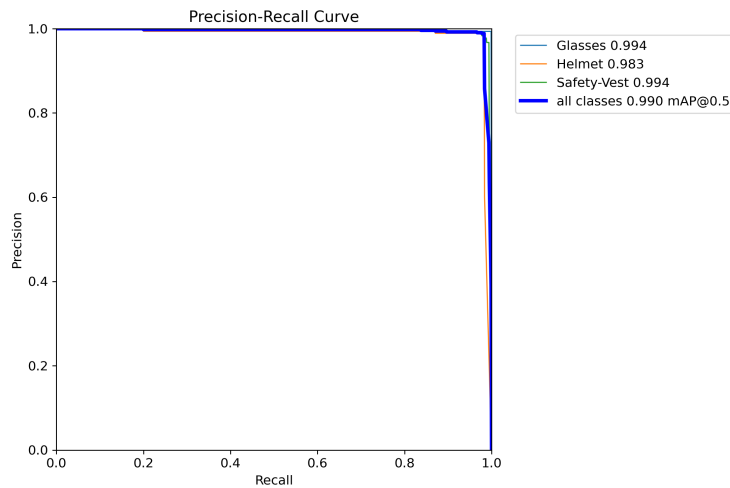


Figura 39. Presión-Recall umbral de confianza. Elaborado por autores

*VII-D1. Precision-Confidence Curve:* Esta gráfica muestra cómo varía la precisión (Precision) del modelo en función del umbral de confianza. Se observa que, a medida que aumenta el nivel de confianza requerido para aceptar una detección, la precisión se incrementa rápidamente hasta estabilizarse cerca de 1.0 en las tres clases (Glasses, Helmet y Safety-Vest). Esto indica que cuando el modelo realiza predicciones con alta confianza, la probabilidad de que sean correctas es muy elevada. En términos prácticos, demuestra que el modelo tiene baja tasa de falsos positivos cuando se trabaja con umbrales de confianza medios o altos.

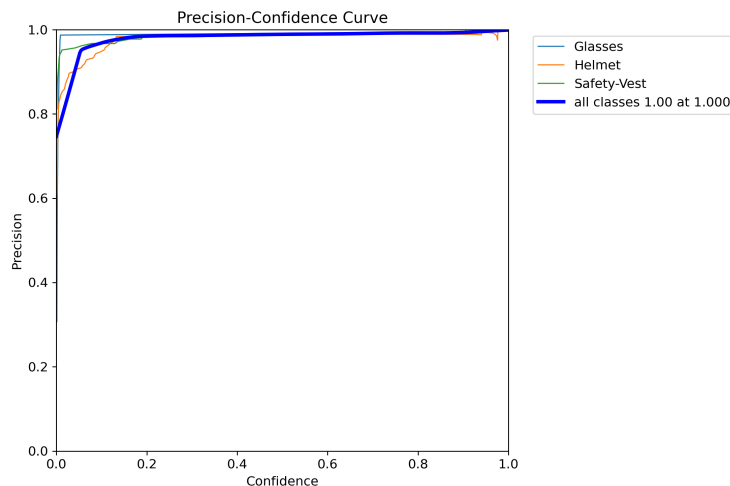


Figura 40. Precision-Confidence umbral de confianza. Elaborado por autores

*VII-D2. Recall-Confidence Curve:* En esta gráfica se analiza cómo cambia la exhaustividad (Recall) al modificar el umbral de confianza. Se aprecia que el recall se mantiene cercano a 1.0 en valores bajos y medios de confianza, pero disminuye cuando el umbral se acerca a 1.0. Esto es un comportamiento esperado, ya que al exigir mayor seguridad en las predicciones, el modelo descarta algunas detecciones verdaderas. Sin embargo, en un rango óptimo de confianza, el modelo mantiene una cobertura muy alta de verdaderos positivos.

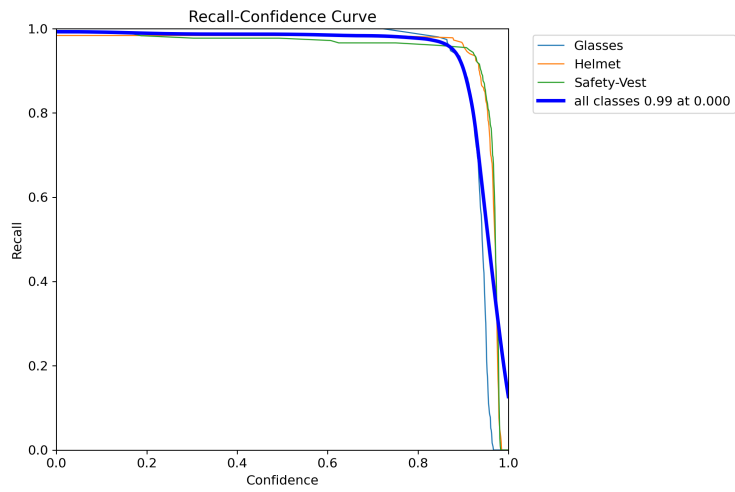


Figura 41. Recall-Confidence umbral de confianza. Elaborado por autores

*VII-D3. F1-Confidence Curve:* La curva F1 muestra el equilibrio entre precisión y recall en función del umbral de confianza. Se observa que el valor máximo de F1 ( 0.99) se alcanza alrededor de un umbral de confianza cercano a 0.5. Esto indica que ese punto representa el mejor balance entre detectar la mayor cantidad de objetos reales y minimizar los falsos positivos. Por encima de ese valor, el F1 comienza a disminuir debido a la caída del recall.

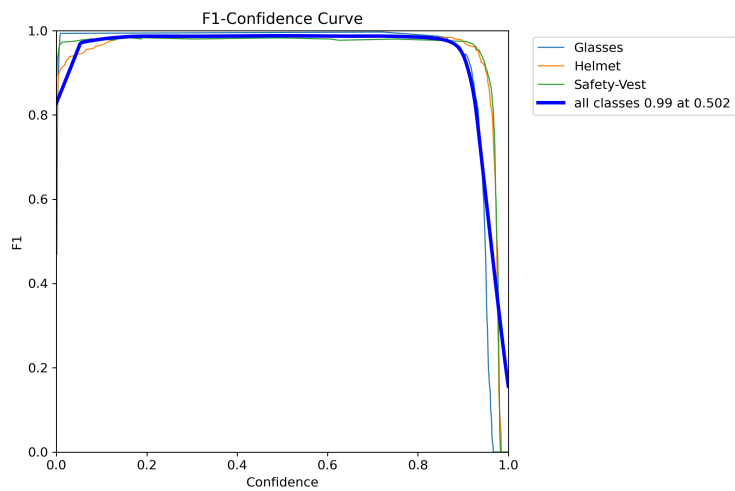
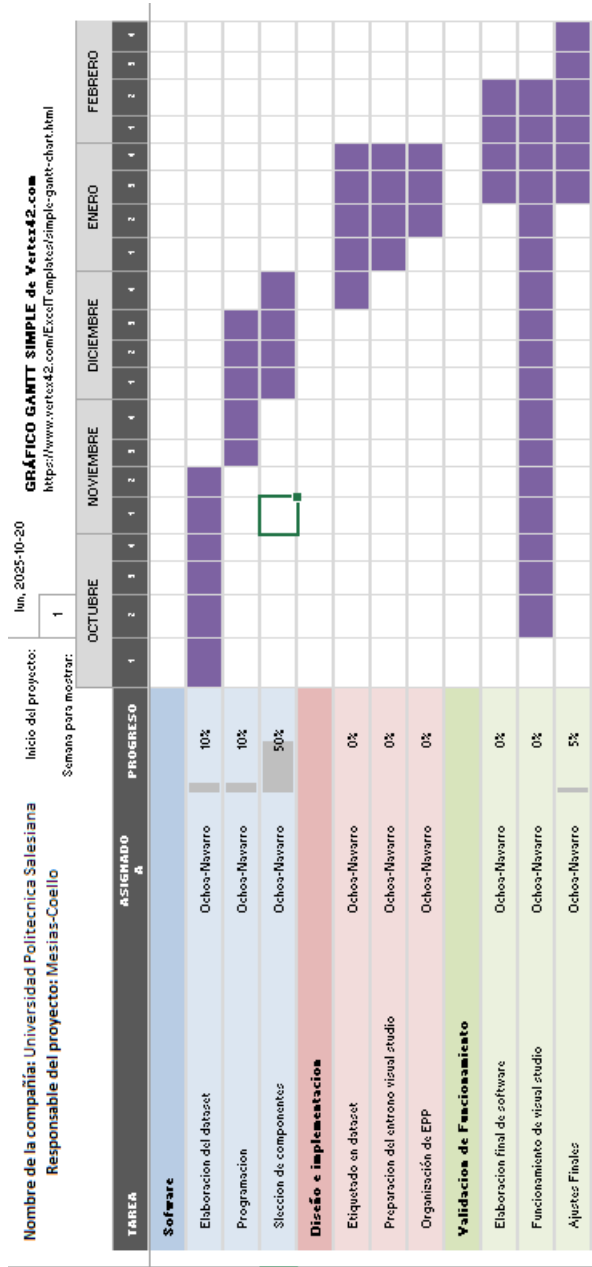


Figura 42. F1-Confidence umbral de confianza. Elaborado por autores

### VIII. CRONOGRAMA

A continuación se muestra el cronograma de trabajo en la tabla I.

Tabla I  
CRONOGRAMA



## IX. PRESUPUESTO

Tabla II  
PRESUPUESTO PARA LA IMPLEMENTACIÓN DEL TRABAJO DE TITULACIÓN (SIMULACIÓN Y DOCUMENTACIÓN)

<b>Ítem</b>	<b>Artículo</b>	<b>P. Unitario USD</b>	<b>Cantidad</b>	<b>P. Total USD</b>
1	Licencia	\$200.00	1	\$200.00
2	Laptop	\$800.00	1	\$800.00
3	Casco	\$50.00	1	\$50.00
4	Chaleco	\$30.00	1	\$30.00
5	Gafas	\$10.00	1	\$10.00
6	Cámara	\$80.00	1	\$80.00
7	Arduino	\$25.00	1	\$25.00
<b>SUBTOTAL</b>				\$1,145.00
<b>MANO DE OBRA</b>				\$800.00
<b>TOTAL</b>				<b>\$1,995.00</b>

## X. CONCLUSIONES

El desarrollo del sistema basado en visión artificial permitió demostrar que la detección automática del uso de Equipos de Protección Personal (EPP) es técnicamente viable mediante modelos de aprendizaje profundo. La implementación de la arquitectura YOLO11s evidenció un desempeño eficiente al mantener un equilibrio adecuado entre precisión y velocidad de inferencia, lo cual resulta fundamental para aplicaciones en tiempo casi real dentro de entornos industriales.

La construcción y etiquetado del dataset constituyó una etapa clave en el proceso, ya que la calidad, diversidad y correcta organización de las imágenes influyeron directamente en el rendimiento del modelo. La división estructurada en conjuntos de entrenamiento, validación y prueba permitió evaluar la capacidad de generalización del sistema y verificar su convergencia sin sobreajuste significativo.

Los resultados obtenidos a través de métricas como Precision, Recall y mAP@0.5 confirmaron un alto nivel de desempeño en la detección de los EPP considerados, mostrando estabilidad en las últimas épocas de entrenamiento. Asimismo, el análisis de la matriz de confusión evidenció una baja tasa de falsos positivos y falsos negativos, lo que valida la confiabilidad del modelo en condiciones controladas.

Finalmente, se concluye que el sistema propuesto representa una solución de bajo costo y alta aplicabilidad para la supervisión automatizada de seguridad industrial, contribuyendo a la reducción de riesgos laborales y fortaleciendo la cultura preventiva mediante monitoreo continuo y objetivo.

## XI. RECOMENDACIONES

Se recomienda ampliar el dataset incorporando mayor diversidad de escenarios, condiciones de iluminación, ángulos de captura y variaciones en los equipos de protección, con el fin de mejorar la robustez del modelo frente a entornos reales más complejos.

Es conveniente evaluar el desempeño del sistema en escenarios industriales reales y no solo en condiciones controladas, con el propósito de validar su comportamiento ante oclusiones, múltiples personas en escena y fondos heterogéneos.

Para futuras investigaciones, se sugiere integrar mecanismos adicionales como reconocimiento facial o sistemas de control de acceso automatizado, permitiendo vincular la detección del uso correcto de EPP con la autorización de ingreso a zonas de riesgo.

## REFERENCIAS

- [1] A. M. Vukićević, M. Petrović y P. Milošević, «A systematic review of computer vision-based personal protective equipment compliance in industry practice: advancements, challenges and future directions,» *Artificial Intelligence Review*, vol. 57, 2024. DOI: 10.1007/s10462-024-10978-x.
- [2] C. Reaño, J. V. Riera, V. Romero, P. Morillo y S. Casas-Yrurzum, «A cloud-edge computing architecture for monitoring protective equipment,» *Journal of Cloud Computing*, 2024. DOI: 10.1186/s13677-024-00649-1.
- [3] M. Imam, K. Baina, Y. Tabii et al., «Ensuring Miners' Safety in Underground Mines Through Edge Computing: Real-Time PPE Compliance Analysis Based on Pose Estimation,» *IEEE Access*, vol. 12, 2024. DOI: 10.1109/ACCESS.2024.3470558.
- [4] C. Liao, H. Chen, C. Liu, Y. Yu y P. Zhao, «Computer Vision-Based Monitoring Method of Non-Wearing Helmet Events Using Face Recognition,» *arXiv*, 2023. DOI: 10.21203/rs.3.rs-3290775/v1.
- [5] Y.-R. Lee, S.-H. Jung, K.-S. Kang, H.-C. Ryu y H.-G. Ryu, «Deep learning-based framework for monitoring wearing personal protective equipment on construction sites,» *Journal of Computational Design and Engineering*, vol. 10, n.º 2, 905–917, 2023. DOI: 10.1093/jcde/qwad019.
- [6] M. U. F. Kisaiezehra, M. A. Bhutto y A. K. Kazi, «Real-Time Safety Helmet Detection Using YOLOv5 at Construction Sites,» *Intelligent Automation & Soft Computing*, vol. 36, 911–927, 2023. DOI: 10.32604/iasc.2023.031359.
- [7] Z. Tang et al., «Optimized YOLOv5 model for safety helmet and flame detection in industrial environments,» *Signal, Image and Video Processing*, 2025. DOI: 10.1007/s11760-025-04073-z.
- [8] I. Yousif et al., «Safety 4.0: Harnessing computer vision for advanced safety in manufacturing through human and PPE detection,» *Journal of Manufacturing Systems*, 2024. DOI: 10.1016/j.jmsy.2024.04.015.
- [9] Y. Shan et al., «An Improved Lightweight Safety Helmet Detection Model: FD-YOLOv5 for Real-World Surveillance,» *Computers, Materials & Continua*, 2025. DOI: 10.32604/cm.2025.060557.
- [10] C. Y. Li et al., «SD-YOLOv5: A rapid detection method for personal protective equipment in construction environments,» *Frontiers in Built Environment*, 2025. DOI: 10.3389/fbuil.2025.1563483.
- [11] V. Isailovic, A. Peulic, M. DJapan, M. Savkovic y A. M. Vukicevic, «The compliance of head-mounted industrial PPE by using deep learning object detectors,» *Scientific Reports*, vol. 12, pág. 20 282, 2022. DOI: 10.1038/s41598-022-20282-9.
- [12] A. M. Vukicevic et al., «A systematic review of computer vision-based personal protective equipment compliance in industry practice: advancements, challenges and future directions,» *Artificial Intelligence Review*, 2024. DOI: 10.1007/s10462-024-10978-x.
- [13] G. Arana-Landín, I. Laskurain-Iturbe, M. Iturrate y B. Landeta-Manzano, «Assessing the influence of industry 4.0 technologies on occupational health and safety,» *Heliyon*, vol. 9, e13720, 2023. DOI: 10.1016/j.heliyon.2023.e13720.
- [14] Y.-R. Lee y S.-H. Jung, «Deep learning-based framework for monitoring wearing personal protective equipment on construction sites,» *Journal of Computational Design & Engineering*, vol. 10, n.º 2, págs. 905-917, 2023. DOI: 10.1093/jcde/qwad040.
- [15] M. I. B. Ahmed, L. Saraireh, A. Rahman, S. Al-Qarawi et al., «Personal Protective Equipment Detection: A Deep-Learning-Based Sustainable Approach,» *Sustainability*, vol. 15, n.º 18, pág. 13 990, 2023. DOI: 10.3390/su151813990.

- [16] M. Gugssa, L. Li, L. Pu, A. Gurbuz, Y. Luo y J. Wang, «Near-real-time safety glove detection through edge computing and transfer learning,» *Engineering, Construction and Architectural Management*, 2024. DOI: 10.1108/ECAM-07-2023-0763.
- [17] F. Kisaehzra, M. U. Bhutto y A. K. Kazi, «Real-Time Safety Helmet Detection Using YOLOv5 at Construction Sites,» *Intelligent Automation & Soft Computing*, vol. 36, n.º 5, págs. 911-927, 2023. DOI: 10.32604/iasc.2023.031359.
- [18] I. o. I. S. Beijing Jiaotong University et al., «FEFD-YOLOV5: A Helmet Detection Algorithm Combined with Feature Enhancement and Feature Denoising,» *Electronics*, vol. 12, n.º 13, pág. 2902, 2023. DOI: 10.3390/electronics12132902.
- [19] G. Hou, Q. Chen, Z. Yang, Y. Zhang, D. Zhang y H. Li, «Helmet detection method based on improved YOLOv5,» *Chinese Journal of Engineering*, vol. 46, n.º 2, págs. 329-342, 2024. DOI: 10.13374/j.issn2095-9389.2022.12.07.002.
- [20] W. Xiao, Z. Ye, J. Yuan, S. Liu et al., «MG-YOLOv5s: A Faster and Stronger Helmet Detection Algorithm,» *Computers, Systems Software Engineering (CSSE)*, 2024. DOI: 10.32604/csse.2023.040475.
- [21] Toshniwal”, “*ML Guide To Build PPE Detection Model For Construction Safety*”. 2023.
- [22] Journals”, “*Detección de componentes de pista faltantes en tiempo real orientada a la computación de borde*”.
- [23] H. Vision, *Pioneering the Future of Intelligent Security: Hanwha Vision*. 2024, Accedido: 3-junio-2025. dirección: [https://www.hanwhavision.com/en/news-center/1576834/?utm\\_source](https://www.hanwhavision.com/en/news-center/1576834/?utm_source).
- [24] W. Y. Q. Y. “Tang Y., “*Transportation Research Record Journal Of The Transportation Research Board*”. “Edge-Computing Oriented Real-Time Missing Track Components Detection”,
- [25] “Admin.”, “*Vision AI detection of PPE usage in the workplace.*” 2024, Accedido: 3-junio-2025. dirección: [https://ehs4safety.com/vision-ai/?utm\\_source](https://ehs4safety.com/vision-ai/?utm_source).
- [26] Management”, “*Edge Computing impulsa innovaciones en el control de acceso.*” 2023, Accedido: 20-mayo-2025. dirección: [https://www.asisonline.org/security-management-magazine/monthly-issues/security-technology/archive/2023/august/edge-computing-unlocks-innovations-in-access-control?utm\\_source](https://www.asisonline.org/security-management-magazine/monthly-issues/security-technology/archive/2023/august/edge-computing-unlocks-innovations-in-access-control?utm_source).
- [27] Ding y X. Luo”, “*Personal Protective Equipment Detection in Extreme Construction Conditions.*” Equipo de proteccion, 2023.
- [28] Z. D.-S. J. Z. Y. Z. C. ”Shi C., “*Protective Equipment Detection in Industrial Environments. Electronics*”. Equipment Detection in Industrial Environments, 2023.
- [29] L. L. H. C. “Lo J., “*Detection Based on Deep Learning Algorithm.*” Based on Deep Learning Algorithm. Sustainability, 2022.
- [30] el edge Computing”, “*Securizando el Edge Computing*”. 2024, Accedido: 1-junio-2025. dirección: <https://www.seidor.com/es-ec/blog/securizando-edge-computing>.
- [31] “Admin.”, “*Vision AI detection of PPE usage in the workplace.*”

## ANEXO A GUIA

El Objetivo de este trabajo es describir el procedimiento para la creación y etiquetado del dataset de imágenes en la plataforma de Roboflow, utilizado para el entrenamiento del modelo de visión artificial para detección de Equipos de Protección Personal (EPP).

Se empleó Roboflow debido a su interfaz intuitiva para anotación de imágenes, compatibilidad con modelos YOLO, control de versiones del dataset y facilidad para exportar datos en distintos formatos.

Tenemos el tipo del proyecto, clases (Helmet, Glasses, Safety-Vest). Con licencia CC BY 4.0

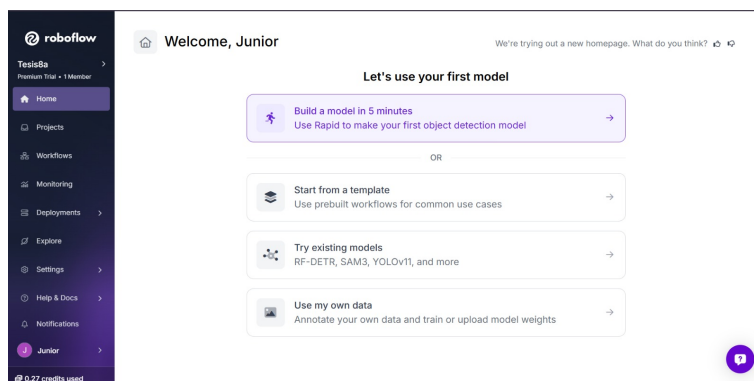


Figura 43. Registro fotográfico del proyecto. Elaborado por autores

### A-A. Procedimiento de Etiquetado

Paso 1: Crear el proyecto en Roboflow seleccionando tipo Object Detection.

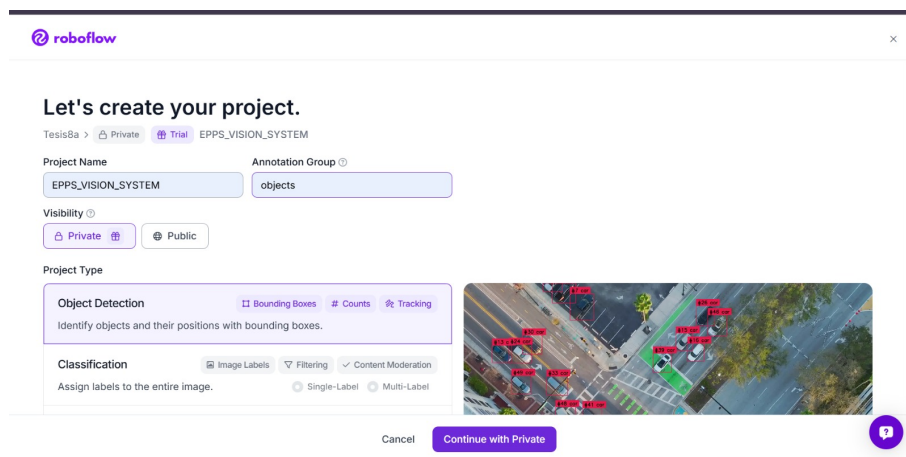


Figura 44. Registro fotográfico del proyecto. Elaborado por autores

Paso 2: Subir las imágenes en formato JPG o PNG.



Figura 45. Registro fotográfico del proyecto. Elaborado por autores



Figura 46. Registro fotográfico del proyecto. Elaborado por autores

Paso 3: Etiquetar cada objeto EPP usando Bounding Boxes.

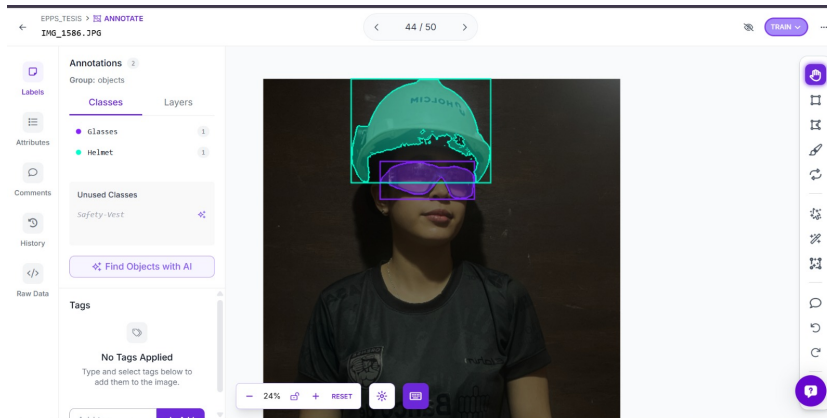


Figura 47. Registro de casco y gafas. Elaborado por autores

Paso 4: Verificar anotaciones para evitar errores.

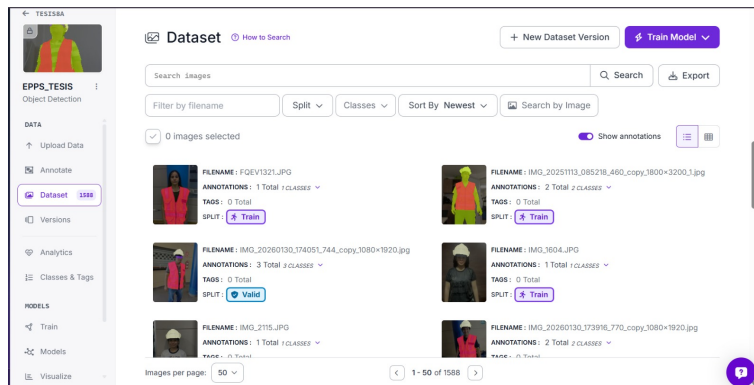


Figura 48. Registro fotográfico del proyecto. Elaborado por autores

Paso 5: Generar una nueva versión del dataset y exportar en formato YOLO.

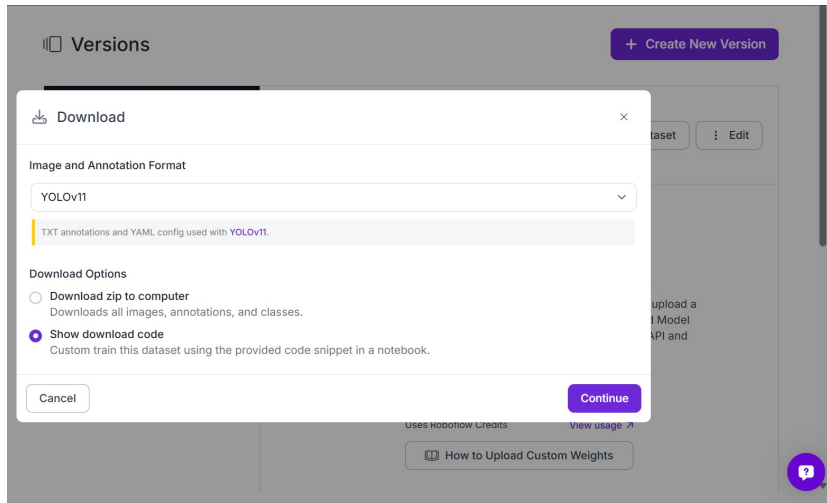


Figura 49. Registro fotográfico del proyecto. Elaborado por autores

### A-B. Reglas de Etiquetado

1. Etiquetar cada EPP por separado.
2. No etiquetar personas completas.
3. Etiquetar objetos parcialmente visibles.
4. Mantener precisión en los bordes.

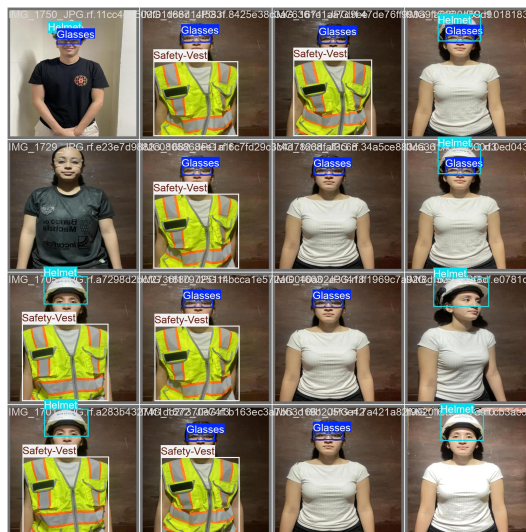


Figura 50. Resultados visuales del modelo YOLO11s. Elaborado por autores

### A-C. División del Dataset

1. Train: 70 %
2. Validation: 20 %
3. Test: 10 %

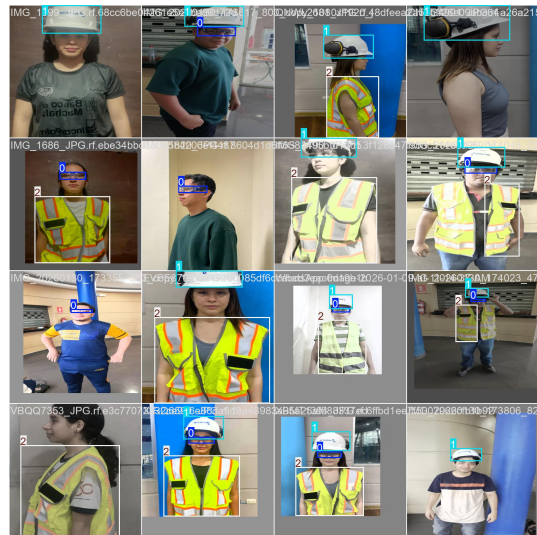


Figura 51. Resultados visuales del modelo YOLO11s. Elaborado por autores

#### A-D. Conclusión

El proceso de etiquetado permitió generar un dataset confiable para el entrenamiento del modelo de detección de EPP, garantizando precisión en el sistema automático de control de acceso industrial.

ANEXO B  
CODIGO

– coding: utf-8 –

```
”TesisNavarrochoa.ipynb
```

Automatically generated by Colab.

Original file is located at <https://colab.research.google.com/drive/1nDfbkSqEPk0dSjcVas6gG7lxhQJ9ykeH>

1. Instalación de librerías

```
”
```

```
!pip install roboflow ultralytics
```

```
”2. Descarga de dataset personalizado desde Roboflow”
```

```
from roboflow import Roboflow
```

```
rf = Roboflow(api_key = "b41xTfDhpyL7X98jxSTt")
```

```
project = rf.workspace("tesis8a").project("epps_tesis - umyib")version = project.version(1)dataset =  
version.download("yolov11"
```

```
, location="/content/drive/MyDrive/TesisYOLOPP3/dataset")
```

```
Çarga Modelo base YOLO v11
```

```
Entrenamiento Personalizado Modelo Yolov11s ”
```

```
from ultralytics import YOLO
```

```
model = YOLO("yolo11s.pt")
```

```
data_path = "/content/drive/MyDrive/TesisYOLOPP3/dataset/data.yaml" results = model.train(data =  
data_path, epochs = 100, imgsz = 640)
```

```
.^ctivar drive en Colab”
```

```
from google.colab import drive
```

```
drive.mount('/content/drive')
```

```
from google.colab import drive drive.mount('/content/drive')
```

```
”Guarda el proyecto en DRIVE
```

```
”
```

```
PROJECTDIR = "/content/drive/MyDrive/TesisYOLOPP3"
```

```
!cp -r /content/runs /content/drive/MyDrive/TesisYOLOPP3/
```

ANEXO C

RESULTADOS DEL ENTRENAMIENTO CON 100 EPOCH

Overriding model.yaml nc=80 with nc=3

```

from n params module arguments 0 -1 1 928 ultralytics.nn.modules.conv.Conv [3, 32, 3, 2]

1 -1 1 18560 ultralytics.nn.modules.conv.Conv [32, 64, 3, 2]

2 -1 1 26080 ultralytics.nn.modules.block.C3k2 [64, 128, 1, False, 0.25]
3 -1 1 147712 ultralytics.nn.modules.conv.Conv [128, 128, 3, 2]
4
-1 1 103360 ultralytics.nn.modules.block.C3k2 [128, 256, 1, False, 0.25]
5 -1 1 590336 ultralytics.nn.modules.conv.Conv [256, 256, 3, 2]

6 -1 1 346112 ultralytics.nn.modules.block.C3k2 [256, 256, 1, True]

7 -1 1 1180672 ultralytics.nn.modules.conv.Conv [256, 512, 3, 2]

8 -1 1 1380352 ultralytics.nn.modules.block.C3k2 [512, 512, 1, True]

9 -1 1 656896 ultralytics.nn.modules.block.SPPF [512, 512, 5]

10 -1 1 990976 ultralytics.nn.modules.block.C2PSA [512, 512, 1]

11 -1 1 0 torch.nn.modules.upsampling.Upsample [None, 2, 'nearest']

12 [-1, 6] 1 0 ultralytics.nn.modules.conv.Concat [1]
13 -1 1 443776 ultralytics.nn.modules.block.C3k2 [768, 256, 1, False]
14 -1 1 0 torch.nn.modules.upsampling.Upsample [None, 2, 'nearest']
15 [-1, 4] 1 0 ultralytics.nn.modules.conv.Concat [1]
16 -1 1 127680 ultralytics.nn.modules.block.C3k2 [512, 128, 1, False]
17 -1 1 147712 ultralytics.nn.modules.conv.Conv [128, 128, 3, 2] 18 [-1, 13] 1 0 ultralytics.nn.modules.conv.Concat
[1]
19 -1 1 345472 ultralytics.nn.modules.block.C3k2 [384, 256, 1, False] 20
-1 1 590336 ultralytics.nn.modules.conv.Conv [256, 256, 3, 2]
21 [-1, 10] 1 0 ultralytics.nn.modules.conv.Concat [1]
22 -1 1 1511424 ultralytics.nn.modules.block.C3k2 [768, 512, 1, True]
23 [16, 19, 22] 1 820569 ultralytics.nn.modules.head.Detect
16, None, [128, 256, 512
] YOLO11s summary: 182 layers, 9,428,953 parameters, 9,428,937 gradients, 21.6 GFLOPs

```

Transferred 493/499 items from pretrained weights

Freezing layer 'model.23.dfl.conv.weight'

AMP: running Automatic Mixed Precision (AMP) checks...

Downloading <https://github.com/ultralytics/assets/releases/download/v8.4.0/yolo26n.pt> to 'yolo26n.pt': 100AMP: checks

passed

train: Fast image access (ping: 0.4±0.1 ms, read: 0.1±0.0 MB/s, size: 41.3 KB)

train: Scanning /content/drive/MyDrive/Tesis<sub>Y</sub>OLO<sub>E</sub>PP3/dataset/train/labels.cache...,1776images,38backgrounds,0corrupt : 100WARNINGBoxandsegmentcountsshouldbeequal,butgotlen(segments) = 3013,len(boxes) = 3018.Toresolvethispleasesupplyeitheradetectorsegmentdataset,notadetect – segmentmixeddataset.

albumentations : Blur( $p = 0,01$ , blur<sub>limit</sub> = (3, 7)), MedianBlur( $p = 0,01$ , blur<sub>limit</sub> = (3, 7)),

ToGray( $p = 0,01$ , method = 'weighted<sub>a</sub>verage', num<sub>output</sub>channels = 3), CLAHE( $p = 0,01$ , clip<sub>limit</sub> = (1,0, 4,0), tile<sub>grid</sub>size = (8, 8))

val : Fastimageaccess(ping : 0,40,1ms, read : 0,10,0MB/s, size : 37,2KB)

val : Scanning/content/drive/MyDrive/Tesis<sub>Y</sub>OLO<sub>E</sub>PP3/dataset/valid/labels.cache...,300images,7

backgrounds,0corrupt : 100Plottinglabelsto/content/runs/detect/train/labels.jpg...optimizer : 'optimizer =

auto' found, ignoring'lr0 = 0,01'and'momentum = 0,937'anddeterminingbest'optimizer', 'lr0'and'momentum'auto

AdamW(lr = 0,001429, momentum = 0,9)withparametergroups81weight(decay = 0,0), 88weight(decay = 0,0005), 87bias(decay = 0,0)Imagesizes640train, 640val

Using2dataloaderworkers

Loggingresultsto/content/runs/detect/train

Startingtrainingfor100epochs...

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize1/1004,08G0,66441,2211,06644640 : 100ClassImagesInstances  
95) : 100all3005170,8560,8810,9280,777

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize2/1004,88G0,65510,64741,02441640 : 100ClassImagesInstance  
95) : 100all3005170,7550,8550,8650,667

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize3/1004,92G0,67970,63751,03152640 : 100ClassImagesInstance  
95) : 100all3005170,8160,9390,9270,747

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize4/1004,96G0,67860,60061,02941640 : 100ClassImagesInstance  
95) : 100all3005170,9410,910,9720,841

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize5/1004,96G0,63950,56971,01149640 : 100ClassImagesInstance  
95) : 100all3005170,8960,9250,9570,82

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize6/1004,96G0,59460,51960,990549640 : 100ClassImagesInstanc  
95) : 100all3005170,960,9440,980,864

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize7/1005G0,55840,49660,972439640 : 100ClassImagesInstances  
95) : 100all3005170,9330,9140,9630,844

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize8/1005G0,54490,48040,97262640 : 100ClassImagesInstancesB  
95) : 100all3005170,9550,9650,9840,898

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize9/1005G0,5290,48040,957839640 : 100ClassImagesInstancesE  
95) : 100all3005170,9540,9640,980,897

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize10/1005G0,51790,4490,958243640 : 100ClassImagesInstancesE  
95) : 100all3005170,9710,9690,9850,899

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize11/1005G0,49880,43380,951159640 : 100ClassImagesInstancesE  
95) : 100all3005170,970,9760,9880,918

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize12/1005G0,48840,42430,943147640 : 100ClassImagesInstancesE  
95) : 100all3005170,9650,9780,9850,911

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize13/1005G0,48470,40430,945350640 : 100ClassImagesInstancesE  
95) : 100all3005170,9870,9620,9870,914

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize14/1005G0,47710,40510,936635640 : 100ClassImagesInstancesE  
95) : 100all3005170,9790,9660,9870,934

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize15/1005,04G0,46190,3940,935147640 : 100ClassImagesInstancesE  
95) : 100all3005170,980,9770,9850,937

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize16/1005,04G0,45930,39440,935539640 : 100ClassImagesInstancesE  
95) : 100all3005170,9760,9740,9860,92

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize17/1005,04G0,45010,39120,925334640 : 100ClassImagesInstancesE  
95) : 100all3005170,9790,9770,9870,937

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize18/1005,04G0,44510,37590,925146640 : 100ClassImagesInstancesE  
95) : 100all3005170,9830,9840,9840,933

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize19/1005,04G0,43230,360,921158640 : 100ClassImagesInstancesE  
95) : 100all3005170,9740,9820,9840,932

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize20/1005,04G0,43670,37410,920236640 : 100ClassImagesInstancesE  
95) : 100all3005170,9760,9860,9890,94

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize21/1005,04G0,4150,35680,909258640 : 100ClassImagesInstancesE  
95) : 100all3005170,9850,9850,9880,942

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize22/1005,04G0,41620,35430,915348640 : 100ClassImagesInstan

95) : 100all3005170,9780,980,9880,941

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize23/1005,04G0,42360,36140,913657640 : 100ClassImagesInstan

95) : 100all3005170,9810,9830,9870,946

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize24/1005,04G0,41330,35410,918940640 : 100ClassImagesInstan

95) : 100all3005170,9740,9820,9860,948

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize25/1005,04G0,39980,34310,909358640 : 100ClassImagesInstan

95) : 100all3005170,990,9850,9870,944

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize26/1005,04G0,40010,33980,905145640 : 100ClassImagesInstan

95) : 100all3005170,9860,9840,9890,952

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize27/1005,04G0,38780,32950,899549640 : 100ClassImagesInstan

95) : 100all3005170,9890,9820,9880,956

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize28/1005,04G0,3840,32240,90244640 : 100ClassImagesInstan

95) : 100all3005170,9810,980,9860,95

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize29/1005,04G0,38340,32830,899851640 : 100ClassImagesInstan

95) : 100all3005170,9840,9850,9860,954

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize30/1005,04G0,37280,31620,89546640 : 100ClassImagesInstan

95) : 100all3005170,9850,9860,9890,953

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize31/1005,08G0,37170,31850,897745640 : 100ClassImagesInstan

95) : 100all3005170,9870,9880,9880,952

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize32/1005,08G0,36710,31360,89555640 : 100ClassImagesInstan

95) : 100all3005170,9510,9850,9880,953

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize33/1005,08G0,36980,30680,893551640 : 100ClassImagesInstan

95) : 100all3005170,990,9870,9890,955

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize34/1005,08G0,35890,30380,890555640 : 100ClassImagesInstan

95) : 100all3005170,9870,9850,9880,956

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize35/1005,08G0,36290,31530,89150640 : 100ClassImagesInstan

95) : 100all3005170,9890,9860,9870,953

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize36/1005,08G0,3610,31280,89442640 : 100ClassImagesInstances  
95) : 100all3005170,9860,9850,9880,956

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize37/1005,08G0,35470,3060,890669640 : 100ClassImagesInstances  
95) : 100all3005170,9840,9830,9890,962

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize38/1005,08G0,34910,30160,885744640 : 100ClassImagesInstances  
95) : 100all3005170,9850,9840,9880,96

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize39/1005,08G0,3450,29970,886663640 : 100ClassImagesInstances  
95) : 100all3005170,9850,9840,9880,959

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize40/1005,08G0,34140,29930,883251640 : 100ClassImagesInstances  
95) : 100all3005170,9870,990,9880,955

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize41/1005,08G0,33110,28680,878657640 : 100ClassImagesInstances  
95) : 100all3005170,9910,9840,9870,965

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize42/1005,08G0,32930,28720,886639640 : 100ClassImagesInstances  
95) : 100all3005170,990,9830,9860,959

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize43/1005,08G0,32590,28130,878854640 : 100ClassImagesInstances  
95) : 100all3005170,990,9820,9860,962

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize44/1005,08G0,33270,28640,882437640 : 100ClassImagesInstances  
95) : 100all3005170,9910,9810,990,961

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize45/1005,08G0,32760,27790,880649640 : 100ClassImagesInstances  
95) : 100all3005170,9870,9840,9880,961

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize46/1005,08G0,32320,27840,881363640 : 100ClassImagesInstances  
95) : 100all3005170,9870,9870,9870,962

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize47/1005,08G0,31690,27380,876455640 : 100ClassImagesInstances  
95) : 100all3005170,9850,9830,9860,961

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize48/1005,08G0,31450,26950,875537640 : 100ClassImagesInstances  
95) : 100all3005170,990,990,9880,964

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize49/1005,08G0,31080,2670,878772640 : 100ClassImagesInstan

95) : 100all3005170,9870,9870,9890,962

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize50/1005,08G0,31070,26580,869457640 : 100ClassImagesInstan

95) : 100all3005170,9840,9860,9890,968

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize51/1005,08G0,32140,26850,877839640 : 100ClassImagesInstan

95) : 100all3005170,9880,9840,990,967

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize52/1005,08G0,30850,2680,871659640 : 100ClassImagesInstan

95) : 100all3005170,9910,9830,9890,965

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize53/1005,08G0,31280,26580,87941640 : 100ClassImagesInstan

95) : 100all3005170,9890,9830,9880,965

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize54/1005,08G0,30580,27330,874733640 : 100ClassImagesInstan

95) : 100all3005170,9880,9850,9870,967

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize55/1005,08G0,31060,26950,873241640 : 100ClassImagesInstan

95) : 100all3005170,9870,9870,9880,964

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize56/1005,08G0,29550,2570,868761640 : 100ClassImagesInstan

95) : 100all3005170,9920,9820,9870,958

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize57/1005,08G0,29730,25560,870248640 : 100ClassImagesInstan

95) : 100all3005170,990,9870,9890,963

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize58/1005,08G0,29470,25030,868151640 : 100ClassImagesInstan

95) : 100all3005170,9910,9860,9890,968

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize59/1005,08G0,29990,25420,875351640 : 100ClassImagesInstan

95) : 100all3005170,9870,9810,9860,965

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize60/1005,08G0,28730,24730,8645640 : 100ClassImagesInstan

95) : 100all3005170,9860,9850,9880,967

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize61/1005,08G0,29530,24580,872246640 : 100ClassImagesInstan

95) : 100all3005170,9870,9830,9890,968

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize62/1005,08G0,28510,24130,866143640 : 100ClassImagesInstan

95) : 100all3005170,9890,9850,9890,968

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize63/1005,08G0,28740,24850,866341640 : 100ClassImagesInstan  
95) : 100all3005170,990,9830,990,972

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize64/1005,08G0,28170,23710,86741640 : 100ClassImagesInstan  
95) : 100all3005170,9890,9880,990,968

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize65/1005,08G0,28820,24620,866948640 : 100ClassImagesInstan  
95) : 100all3005170,9910,9870,9890,967

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize66/1005,08G0,28460,24150,868459640 : 100ClassImagesInstan  
95) : 100all3005170,9920,9850,9880,966

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize67/1005,08G0,27890,24470,862946640 : 100ClassImagesInstan  
95) : 100all3005170,9840,990,9880,97

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize68/1005,08G0,27750,24080,867850640 : 100ClassImagesInstan  
95) : 100all3005170,9890,9850,9870,966

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize69/1005,08G0,28160,24070,863259640 : 100ClassImagesInstan  
95) : 100all3005170,9860,9880,9860,969

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize70/1005,08G0,27270,23410,861663640 : 100ClassImagesInstan  
95) : 100all3005170,9890,9850,9870,964

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize71/1005,08G0,27240,2350,859454640 : 100ClassImagesInstan  
95) : 100all3005170,990,9830,9890,967

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize72/1005,08G0,26840,23180,862444640 : 100ClassImagesInstan  
95) : 100all3005170,9870,990,990,973

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize73/1005,08G0,26880,22880,861152640 : 100ClassImagesInstan  
95) : 100all3005170,9820,9890,9890,971

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize74/1005,08G0,26080,23350,856149640 : 100ClassImagesInstan  
95) : 100all3005170,9880,9860,9890,971

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize75/1005,08G0,26080,2290,861245640 : 100ClassImagesInstan  
95) : 100all3005170,9880,9860,990,972

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize76/1005,08G0,26160,22150,859933640 : 100ClassImagesInstan

95) : 100all3005170,990,9840,9890,969

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize77/1005,08G0,2580,22130,861151640 : 100ClassImagesInstan

95) : 100all3005170,9880,990,990,973

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize78/1005,08G0,25210,2190,85451640 : 100ClassImagesInstan

95) : 100all3005170,990,9860,990,974

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize79/1005,08G0,25530,22450,854555640 : 100ClassImagesInstan

95) : 100all3005170,9880,9880,990,975

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize80/1005,08G0,25130,2180,856840640 : 100ClassImagesInstan

95) : 100all3005170,9790,9910,990,972

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize81/1005,08G0,24650,21460,853552640 : 100ClassImagesInstan

95) : 100all3005170,9760,9930,990,973

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize82/1005,08G0,24260,22180,854543640 : 100ClassImagesInstan

95) : 100all3005170,990,9870,9890,972

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize83/1005,08G0,24260,20570,855155640 : 100ClassImagesInstan

95) : 100all3005170,9890,9850,9910,975

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize84/1005,08G0,24170,21120,853861640 : 100ClassImagesInstan

95) : 100all3005170,990,9870,9910,973

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize85/1005,08G0,2390,20680,855764640 : 100ClassImagesInstan

95) : 100all3005170,9850,990,9910,973

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize86/1005,08G0,23470,20230,850167640 : 100ClassImagesInstan

95) : 100all3005170,990,9870,990,976

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize87/1005,08G0,23520,20240,853838640 : 100ClassImagesInstan

95) : 100all3005170,9830,9930,990,976

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize88/1005,08G0,23420,2060,853252640 : 100ClassImagesInstan

95) : 100all3005170,9820,990,990,974

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize89/1005,08G0,23450,21010,851949640 : 100ClassImagesInstan

95) : 100all3005170,9910,9840,990,972

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize90/1005,08G0,23790,20580,852858640 : 100ClassImagesInstan  
95) : 100all3005170,9920,9870,990,973

Closingdataloadermosaicalbumentations : Blur( $p = 0,01$ , blur<sub>l</sub>imit = (3, 7)), MedianBlur( $p = 0,01$ , blur<sub>l</sub>imit = (3, 7)), ToGray( $p = 0,01$ , method = 'weighted<sub>a</sub>verage', num<sub>o</sub>utput<sub>c</sub>hannels = 3), CLAHE( $p = 0,01$ , clip<sub>l</sub>imit = (1,0, 4,0), tile<sub>g</sub>rid<sub>s</sub>ize = (8, 8))

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize91/1005,08G0,18490,16020,803228640 : 100ClassImagesInstan  
95) : 100all3005170,9920,9880,990,974

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize92/1005,08G0,17510,15720,799529640 : 100ClassImagesInstan  
95) : 100all3005170,990,9860,990,974

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize93/1005,08G0,17610,15160,798227640 : 100ClassImagesInstan  
95) : 100all3005170,9880,9880,990,973

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize94/1005,08G0,17510,15350,799118640 : 100ClassImagesInstan  
95) : 100all3005170,990,9890,990,974

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize95/1005,08G0,17170,15220,800727640 : 100ClassImagesInstan  
95) : 100all3005170,990,9890,990,975

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize96/1005,08G0,17190,14620,798829640 : 100ClassImagesInstan  
95) : 100all3005170,9890,9870,990,975

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize97/1005,08G0,16610,14670,79525640 : 100ClassImagesInstan  
95) : 100all3005170,990,9870,990,976

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize98/1005,08G0,16350,1430,797225640 : 100ClassImagesInstan  
95) : 100all3005170,9890,9870,990,976

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize99/1005,08G0,16230,14020,796827640 : 100ClassImagesInstan  
95) : 100all3005170,9890,9870,990,976

Epoch GPU<sub>m</sub>embox<sub>l</sub>osscls<sub>l</sub>ossdf<sub>l</sub>lossInstancesSize100/1005,08G0,16270,14190,795631640 : 100ClassImagesInsta  
95) : 100all3005170,9920,9870,990,976

100 epochs completed in 1.484 hours.

Optimizer stripped from /content/runs/detect/train/weights/last.pt, 19.2MB

Optimizer stripped from /content/runs/detect/train/weights/best.pt, 19.2MB

Validating /content/runs/detect/train/weights/best.pt...  
Ultralytics 8.4.12 Python-3.12.12 torch-2.9.0+cu126 CUDA:0 (Tesla T4, 15095MiB)  
YOLO11s summary (fused): 101 layers, 9,413,961 parameters, 0 gradients, 21.3 GFLOPs  
Class Images Instances Box(P R mAP50 mAP50-95): 100all 300 517 0.989 0.987 0.99 0.976  
Glasses 154 154 0.992 1 0.994 0.969  
Helmet 184 185 0.987 0.984 0.983 0.968  
Safety-Vest 178 178 0.989 0.977 0.994 0.99  
Speed: 0.2ms preprocess, 4.7ms inference, 0.0ms loss, 4.7ms postprocess per image  
Results saved to /content/runs/detect/train