



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

TÍTULO DEL TRABAJO DE TITULACIÓN

**DESARROLLO DE UN ALGORITMO BASADO EN PYTHON
QUE PERMITA ENTRENAR UNA RED NEURONAL, PARA CONTEO
AUTOMÁTICO DEL FLUJO DE PERSONAS QUE CIRCULAN**

AUTOR

Muñoz Alejandro Lesly Melanie

MODALIDAD DE TITULACIÓN

Proyecto UIC

**Previo a la obtención del grado académico en
INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN**

TUTOR

Ing. Rosero Vásquez, Shendry Balmore, Msc.

Santa Elena, Ecuador

Año 2023



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

TRIBUNAL DE SUSTENTACIÓN

Ing. José Sánchez Aquino Mgt.
DIRECTOR DE LA CARRERA

Ing. Shendry Rosero Vásquez, Msc
TUTOR

Ing. Walter Orozco Iguasnia. Mgt.
DOCENTE ESPECIALISTA

Ing. Mariorie Coronel S. Mgt.
DOCENTE GUÍA UIC



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por Muñoz Alejandro Lesly Melanie, como requerimiento para la obtención del título de Ingeniero en Tecnologías de la Información.

La Libertad, a los 08 días del mes de agosto del año 2023

TUTOR

A handwritten signature in blue ink, appearing to be "Rosero Vásquez Shendry", written over a horizontal line.

Ing. Rosero Vásquez Shendry, Msc.



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

DECLARACIÓN DE RESPONSABILIDAD

Yo, MUÑOZ ALEJANDRO LESLY MELANIE

DECLARO QUE:

El trabajo de Titulación, “Desarrollo De Un Algoritmo Basado En Python Que Permita Entrenar Una Red Neuronal, Para Conteo Automático Del Flujo De Personas Que Circulan” previo a la obtención del título en Ingeniero en Tecnologías de la Información, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

La Libertad, a los 08 días del mes de agosto del año 2023

EL AUTOR

A handwritten signature in blue ink, appearing to read "Lesly Melanie Muñoz Alejandro", is written over a horizontal line.

Lesly Melanie Muñoz Alejandro



UPSE
UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CERTIFICACIÓN DE ANTIPLAGIO

En calidad de tutor del trabajo de titulación denominado **“Desarrollo De Un Algoritmo Basado En Python Que Permita Entrenar Una Red Neuronal, Para Conteo Automático Del Flujo De Personas”**, elaborado por la estudiante, **Muñoz Alejandro Lesly Melanie**, estudiante de la Carrera de Tecnologías de la Información de la Facultad de Sistemas y Telecomunicaciones de la Universidad Estatal Península de Santa Elena, previo a la obtención del título de Ingeniera en Tecnologías de la Información y la Comunicación, me permito declarar que una vez analizado en el sistema anti plagio **“COMPILATIO MAGISTER”**, y luego de haber cumplido los requerimientos exigidos de valoración el presente proyecto ejecutado, se encuentra con 4% de similitud con respecto a otros documentos, por consiguiente se procede a emitir el presente informe. Adjunto reporte de similitud.



CERTIFICADO DE ANÁLISIS
magister

Proyecto UIC - Lesly Melanie Muñoz Alejandro

4%
Similitudes



3% Texto entre comillas
< 1% similitudes entre comillas
< 1% Idioma no reconocido

Nombre del documento: Proyecto UIC - Lesly Melanie Muñoz Alejandro.docx
ID del documento: 3b3af093dbe9d1b38744a5ae30ff1d5192c76d5c

Depositante: SHENDRY BALMORE ROSERO VASQUEZ
Fecha de depósito: 2/8/2023
Tipo de carga: interface
fecha de fin de análisis: 2/8/2023

Número de palabras: 21.913
Número de caracteres: 151.724

TUTOR

Ing. Rosero Vásquez Shendry, Msc.



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

AUTORIZACIÓN

Yo, MUÑOZ ALEJANDRO LESLY MELANIE

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales de artículo profesional de alto nivel con fines de difusión pública, además apruebo la reproducción de este artículo académico dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor

Santa Elena, a los 08 días del mes de agosto del año 2023

EL AUTOR

A handwritten signature in blue ink, appearing to read "Muñoz Alejandro Lesly Melanie", is written over a horizontal line.

MUÑOZ ALEJANDRO LESLY MELANIE

AGRADECIMIENTO

Quiero expresar mi profunda gratitud a todas las personas fundamentales en este proceso de investigación. Su apoyo incondicional y aliento me han impulsado a superar obstáculos y perseverar.

En primer lugar, quiero agradecer de manera especial a mis hermanos. Su constante apoyo emocional, palabras de aliento y paciencia han sido una fuente invaluable de inspiración para mí. Gracias por creer en mí y en mi capacidad para llevar a cabo este proyecto.

A mi estimado tutor, quiero expresar mi sincero agradecimiento por su guía experta y su dedicación. Su orientación me ha ayudado a dar forma y fortalecer esta tesis.

Por último, pero no menos importante, quiero agradecer a mis padres. Su confianza en mí han sido el pilar que ha sostenido mi trayectoria académica. Gracias por brindarme todas las oportunidades y para crecer como persona y estudiante. Su apoyo que me ha impulsado a perseguir mis sueños.

A todos ustedes, les debo el éxito de esta tesis. Sin su presencia y respaldo, este logro no habría sido posible. Espero que estas palabras sean un reflejo sincero de mi agradecimiento hacia cada uno de ustedes.

¡Gracias de corazón!

Lesly Melanie, Muñoz Alejandro

ÍNDICE GENERAL

TITULO DEL TRABAJO DE TITULACIÓN	I
TRIBUNAL DE SUSTENTACIÓN	II
CERTIFICACIÓN.....	III
DECLARACIÓN DE RESPONSABILIDAD	IV
DECLARO QUE:	IV
CERTIFICACIÓN DE ANTIPLAGIO	V
AUTORIZACIÓN	VI
AGRADECIMIENTO	VII
DEDICATORIA.....	VII
ÍNDICE GENERAL	VIII
ÍNDICE DE TABLAS	XII
ÍNDICE DE FIGURAS	XIII
RESUMEN	XVI
ABSTRACT.....	XVII
INTRODUCCIÓN	2
CAPÍTULO I: 1. FUNDAMENTACIÓN	3
1.1 ANTECEDENTES	3
1.2 DESCRIPCIÓN DEL PROCESO.....	5
1.3 OBJETIVO.....	9
1.3.1 OBJETIVO GENERAL	9
1.3.2 OBJETIVOS ESPECIFICOS	9
1.4 JUSTIFICACIÓN	10

1.5 ALCANCE DEL PROYECTO	13
1.6 METODOLOGÍA.....	14
1.6.1 METODOLOGÍA DE INVESTIGACIÓN	14
1.6.2 VARIABLES	15
1.6.3 TECNICAS DE RECOLECCIÓN DE INFORMACIÓN.....	15
1.6.4 METODOLOGIA DE DESARROLLO.....	16
CAPITULO II: 2. MARCOS DEL PROYECTO	17
2.1 MARCO CONTEXTUAL.....	17
2.1.1 El GAD (Gobierno Autónomo Descentralizado del cantón salinas).....	17
2.1.2 MISIÓN	17
2.1.3 VISIÓN	18
2.1.4 LEY ORGÁNICA DE TRANSPARENCIA Y ACCESO A LA INFORMACIÓN PÚBLICA.....	18
2.1.5 LEY ORGÁNICA DE PARTICIPACIÓN CIUDADANA Y LEY ORGÁNICA DEL CONSEJO DE PARTICIPACIÓN CIUDADANA Y CONTROL SOCIAL	19
2.2 MARCO CONCEPTUAL.....	20
2.2.1 Inteligencia artificial	20
2.2.1.1 Redes neuronales.....	20
2.2.1.2 Redes neuronales convolucionales.....	21
2.2.1.3 Machine Learning o Aprendizaje automatico	22
2.2.1.4 Deep Learning o Aprendizaje profundo.....	22
2.2.1.5 Algoritmos de vision para deteccion de objetos	22
2.2.1.6 Algoritmo de vision para seguimiento de objetos.....	23
2.2.1.7 Algoritmo de vision para conteo de objetos.....	24
2.2.2 Vision artificial.....	25
2.2.2.1 Imágenes y espacio de color	25
2.2.2.2 OpenCV.....	25
2.2.3 Procesamiento de imagen.....	26

2.2.3.1	Seguimiento o tracking	26
2.2.3.2	Pesos entrenados	26
2.2.3.3	Cajas delimitadoras (bounding boxes)	27
2.2.4	Modelo YOLOv5	27
2.2.4.1	Mejor precisión	27
2.2.4.2	Mayor velocidad.....	27
2.2.4.3	Tamaños de modelo escalables	28
2.2.4.4	Entrenamiento y personalización sencillos	28
2.2.4.5	Soporte para diferentes aplicaciones	28
2.2.4.6	Entrenamiento personalizado con Yolov5	28
2.2.4.7	Preparación del conjunto de datos personalizados.....	28
2.2.5	Lenguajes de programación	29
2.2.5.1	Python	29
2.2.5.2	Google Colab	29
2.2.5.3	Jupyter Notebook	29
2.2.6	Bibliotecas e herramientas	30
2.2.6.1	Keras	30
2.2.6.2	TensorFlow	30
2.2.6.3	Yolov5.....	30
2.2.6.4	Pytorch	30
2.2.6.5	Make Sense	31
2.2.6.6	Tensorboard.....	31
2.2.6.7	Numpy.....	31
2.3	MARCO TEORICO.....	31
2.3.1	La inteligencia artificial como herramienta de estrategia y seguridad	31

2.3.2	Reconocimiento de patrones de imágenes a través de un sistema de visión artificial	32
2.3.3	Estrategia y avances recientes en detección y clasificación de objetos por medio de algoritmos de aprendizaje automático.....	32
CAPITULO III: 3. PROPUESTA		33
3.1 REQUERIMIENTO		33
3.1.1	Requerimientos funcionales	33
3.1.2	Requerimientos no funcionales	37
3.2 COMPONENTES DE LA PROPUESTA		38
3.2.1	Diagrama de caso de uso.....	38
3.2.1	Diagrama de bloques.....	43
3.3 DESARROLLO DE LA PROPUESTA.....		46
3.3.1	Módulo de imágenes	47
3.3.1.1	Repositorio de imágenes	47
3.3.1.2	Datos de imágenes.....	48
3.3.1.3	Etiquetado de personas.....	49
3.3.1.4	Generar archivo en formato YOLO	50
3.3.2	Módulo de creación de la red	51
3.3.2.1	Evaluar métrica y set de validación	52
3.3.2.2	Alteraciones para mejorar la red	55
3.3.2.3	Procesamiento de épocas.....	57
3.3.2.4	Análisis de datos resultante de predicción	58
3.3.3	Módulo de entrenamiento	64
3.3.3.1	Conjunto de datos.....	64
3.3.3.2	Entrada de imágenes	65
3.3.3.3	Pre-procesamiento.....	65
3.3.3.4	Extracción de características	66

3.3.3.5 Entrenar la red.....	68
3.4 PRUEBA Y RESULTADOS DEL SISTEMA	72
3.4.1 Evaluación del rendimiento de tiempo de respuesta	72
3.4.1.1 Prueba de imágenes.....	72
3.4.1.2 Prueba de video	74
3.4.1.3 Prueba en tiempo real.....	76
3.4.2 Evaluación de precisión para distintas cantidades de personas detectadas	77
3.4.2.1 Pruebas de precisión para un número específico de personas.....	78
CONCLUSIONES	81
RECOMENDACIONES	82
BIBLIOGRAFÍA	83

ÍNDICE DE TABLAS

Tabla 1 : Información revelada por el Ministerio de Gobierno, acerca de robo a personas por mes en Ecuador durante los años de 2019 hasta 2022: El universo	11
Tabla 2. Artículos de Ley Orgánica De Transparencia Y Acceso A La Información Pública: CPCCS	19
Tabla 3: Artículos De Participación Ciudadana y Ley Orgánica: CPCCS	20
Tabla 4: Requerimiento funcionales: Elaboración propia.	37
Tabla 5: Requerimientos no funcionales: Elaboración propia.	38
Tabla 6: Caso de Uso: Entrenar Modelo: Elaboración propia.	40
Tabla 7: Caso de Uso: Capturar y Procesar Datos: Elaboración propia.	41
Tabla 8: Caso de Uso: Contar Personas: Elaboración propia.	42
Tabla 9: Caso de Uso: Mantenimiento: Elaboración propia.	43

Tabla 10: Descripción del módulo de imágenes: Elaboración propia.	44
Tabla 11: Descripción del módulo de entrenamiento: Elaboración propia.	45
Tabla 12: Descripción del módulo de creación de la red: Elaboración propia.	46
Tabla 13: Resultado de prueba de imágenes: Elaboración propia.	74
Tabla 14: Resultado prueba de video: Elaboración propia.	76
Tabla 15: Resultado de prueba en tiempo real: Elaboración propia.	77

ÍNDICE DE FIGURAS

Figura 1: Metodología para el descubrimiento de conocimiento en bases de datos: J. M. Molina López y J. García Herrero	15
Figura 2: Metodología de desarrollo incremental para el entrenamiento de una red neuronal. Elaboración propia.	17
Figura 3: Ubicación Geográfica de Salinas: Google maps	18
Figura 4: Forma de la operatividad del funcionamiento: Innovation, Atria.	21
Figura 5: Funcionamiento de las redes neuronales convolucionales: Datademia.	22
Figura 6: Reconocimiento de objetos en imagen: Darkcrizt.	23
Figura 7: Visión artificial seguimiento: Sandra.	24
Figura 8: Elementos de un sistema de conteo: Luna, Andrés; Rodríguez, Nicolás.	24
Figura 9: Diagrama de Caso de Uso: Entrenar Modelo de Conteo Automático de Personas: Elaboración propia.	39
Figura 10: Diagrama de Caso de Uso: Capturar y Procesar Datos: Elaboración propia.	40
Figura 11: Diagrama de Caso de Uso: Contar Personas: Elaboración propia.	41
Figura 12: Diagrama de Caso de Uso: Mantenimiento: Elaboración propia.	42
Figura 13: Diagrama de bloque: Elaboración propia.	43
Figura 14: Repositorio de imágenes: Elaboración propia.	47

Figura 15: Esquema lumínico: Toomanyflash.	49
Figura 16: Etiquetado de las imágenes: Elaboración propia.	50
Figura 17: Archivos: Formato Yolo: Elaboración propia.	51
Figura 18: Contenido del archivo de configuración customdata.yaml: Elaboración propia.	52
Figura 19: Script de entrenamiento: Elaboración propia.	53
Figura 20: Archivos del Modelo Entrenado y Modelo Final: Elaboración propia.	53
Figura 21: Script Prueba de Inferencia: Elaboración propia.	53
Figura 22: Resultado de la Prueba de Inferencia en Imágenes: Elaboración propia.	54
Figura 23: Evaluación del Desempeño del Sistema en una Imagen Específica: Elaboración propia.	54
Figura 24: Escaneo de conjunto de entrenamiento (Train): Elaboración propia.	56
Figura 25: Escaneo de conjunto de entrenamiento (Val): Elaboración propia.	57
Figura 26: Avance en cada época y las métricas obtenidas: Elaboración propia.	58
Figura 27: Resultado de métrica mAP_0.5: Elaboración propia.	59
Figura 28: Resultado de métrica mAP_0.5:0.95: Elaboración propia.	59
Figura 29: Resultado de métrica / precisión: Elaboración propia.	60
Figura 30: Resultado de métrica / recall: Elaboración propia.	60
Figura 31: Resultado de pérdida de bounding boxes: Elaboración propia.	60
Figura 32: Resultado de pérdida de clasificación: Elaboración propia.	61
Figura 33: Resultado de Pérdida de objetos: Elaboración propia.	61
Figura 34: Resultado de pérdida de bounding boxes en la validación: Elaboración propia.	62
Figura 35: Resultado de pérdida de clasificación en el conjunto de validación: Elaboración propia.	62

Figura 36: Resultado de pérdida de objetos en el conjunto de validación: Elaboración propia.	62
Figura 37: Algoritmo de seguimiento: Elaboración propia.	67
Figura 38: Script de prueba: Elaboración propia.	68
Figura 39: Resultado obtenido imagen de prueba: Elaboración propia.	68
Figura 40: Comparación de detecciones con umbrales de confianza: Elaboración propia.	69
Figura 41: Carga del modelo con DetectMultiBackend: Elaboración propia.	70
Figura 42: Inferencia en la imagen de entrada utilizando el modelo cargado: Elaboración propia.	70
Figura 43: Supresión de No Máximos (NMS): Elaboración propia.	70
Figura 44: Resultado por pantalla de detección de personas: Elaboración propia.	71
Figura 45: Visualización de resultados de detección de personas: Elaboración propia.	71
Figura 46: Resultado por pantalla de contabilización: Elaboración propia.	72
Figura 47: Precisión del Sistema vs. Número de Persona: Elaboración propia.	78
Figura 48: Prueba de precisión con 7 personas: Elaboración propia.	79
Figura 49: Prueba de precisión con 8 personas: Elaboración propia.	80
Figura 50: Prueba de precisión con 10 personas: Elaboración propia.	81

RESUMEN

Este trabajo se enfoca en desarrollar un algoritmo de visión artificial para reconocer y clasificar personas. Se utilizan el aprendizaje profundo y redes neuronales convolucionales para lograr una detección precisa y eficiente de personas en distintos contextos visuales. El objetivo principal es mejorar la identificación de personas en planos verticales, lo que representa un desafío en el ámbito del reconocimiento de objetos. Durante la prueba en tiempo real, se llevaron a cabo distintas secuencias para la detección, y los resultados mostraron un tiempo promedio de 784.64 milisegundos, evidenciando la capacidad del sistema para adaptarse a diferentes situaciones. En la evaluación de precisión, se analizó la respuesta del sistema en función del número de personas presentes. Cumple con el requerimiento del 80% hasta diez personas, disminuyendo su precisión a partir de esa cantidad debido al solapamiento entre ellas.

Palabras claves: visión artificial, Redes neuronales convolucionales, Detección de personas

ABSTRACT

This work focuses on developing a computer vision algorithm to recognize and classify people. Deep learning and convolutional neural networks are used to achieve accurate and efficient detection of people in different visual contexts. The main objective is to improve the identification of people in vertical planes, which is a challenge in the field of object recognition. During the real-time test, different sequences for detection were carried out, and the results showed an average time of 784.64 milliseconds, evidencing the system's ability to adapt to different situations. In the accuracy evaluation, the system response was analyzed as a function of the number of people present. It complies with the 80% requirement up to ten people, decreasing its accuracy after that number due to the overlap between them.

Keywords: Artificial Visión, Convolutional Neural Networks, People Detection

INTRODUCCIÓN

El reconocimiento de objetos representa una técnica esencial en el campo de la visión artificial, cuyo propósito radica en identificar y clasificar elementos presentes en imágenes o videos. Este avance ha marcado un hito significativo en los algoritmos de aprendizaje profundo y aprendizaje automático. Mientras que los seres humanos poseen la capacidad innata de reconocer personas, observar detalladamente y comprender información visual en fotografías y videos, el objetivo es capacitar a las computadoras para realizar esta tarea de manera natural y adquirir un nivel de comprensión similar del contenido visual presente en las imágenes.

La detección de personas en imágenes y videos plantea desafíos computacionales debido a la gran diversidad en términos de forma, tamaño, color, textura y orientación que pueden presentar los objetos. Para abordar esta complejidad, se han desarrollado algoritmos especializados en el reconocimiento de formas y en el análisis de similitud entre imágenes.

Estos algoritmos utilizan técnicas avanzadas de visión artificial y aprendizaje automático para lograr una detección precisa y eficiente de personas en diferentes contextos visuales. Algunos de estos métodos incluyen el uso de redes neuronales convolucionales, clasificadores basados en características y enfoques de aprendizaje profundo, que permiten a las ocasiones identificar patrones y características distintivas para reconocer y localizar personas con alta precisión.

Hasta la fecha, no se ha desarrollado una solución genérica que permita la detección precisa de objetos en un plano vertical, lo que representa un desafío significativo en el ámbito del reconocimiento de objetos. La detección de objetos en este tipo de configuración requiere investigaciones adicionales y enfoques innovadores para lograr resultados confiables y precisos en estas situaciones.

El diseño de algoritmos efectivos para la detección en planos verticales implica considerar las peculiaridades geométricas y de iluminación que influyen en la apariencia de los objetos desde diferentes ángulos. Además, las técnicas tradicionales de detección de objetos en planos horizontales no pueden aplicarse directamente a esta situación, lo que implica la necesidad de adaptar o desarrollar nuevos enfoques.

El presente trabajo se enfoca en desarrollar un algoritmo que mejore la detección, con especial énfasis en la identificación y conteo de personas utilizando redes neuronales convolucionales y técnicas de aprendizaje profundo. La meta principal es implementar un análisis que permita identificar y contar personas, lo que podría tener aplicaciones en diversas áreas, como videovigilancia, control de aforo y planificación de eventos.

La investigación se ha realizado mediante una revisión exhaustiva de trabajos anteriores relacionados con el reconocimiento de objetos, el conteo de personas y el uso de redes neuronales convolucionales para tareas de visión artificial. Se han analizado proyectos de titulación y trabajos de investigación que abordaron temas similares, destacando aquellos que han logrado avances significativos en la detección y reconocimiento de personas en imágenes y videos.

Además, se ha realizado una búsqueda de literatura académica, artículos científicos y publicaciones relevantes en el campo de la visión artificial y el aprendizaje profundo. Se ha consultado bibliografía especializada y fuentes confiables para obtener información actualizada sobre las técnicas, herramientas y algoritmos más utilizados en el reconocimiento de objetos y la detección de personas.

CAPÍTULO I: 1. FUNDAMENTACIÓN

1.1 ANTECEDENTES

El reconocimiento de objetos es una técnica fundamental en el campo de la visión artificial, que se emplea para identificar y categorizar elementos presentes en imágenes o videos. Este avance se ha convertido en un hito importante dentro de los algoritmos de aprendizaje profundo y aprendizaje automático. Mientras los seres humanos somos capaces de reconocer de manera rápida y precisa personas, ubicaciones y detalles visuales en fotografías y videos, el objetivo es enseñar a las computadoras a realizar esta tarea de forma natural, permitiendo así adquirir un nivel de comprensión del contenido visual presente en las imágenes. [1]

La tarea de reconocimiento de personas resulta compleja para las computadoras debido a diversas razones. Pueden presentar una amplia variabilidad en términos de forma, tamaño,

color, textura y orientación. Esto implica que la computadora debe ser capaz de generalizar y adaptarse a diferentes características visuales para identificar correctamente.

Con el fin de abordar esta complejidad, se han desarrollado algoritmos especializados en el reconocimiento de formas y el análisis de similitud de imágenes. Estos algoritmos buscan identificar patrones y características específicas en las imágenes para inferir la presencia y la clase de los objetos detectados. Sin embargo, hasta el momento no existe una implementación genérica que permita la detección de objetos en un plano vertical.

Esta falta de una solución genérica representa un desafío en el ámbito del reconocimiento de objetos y requiere investigaciones adicionales para lograr un reconocimiento preciso y confiable en dichas situaciones.

Existen diferentes trabajos de investigación que aportan al desarrollo de algoritmo para entrenar una red neuronal del reconocimiento de objetos y se citan a continuación:

En referencia al proyecto de titulación titulado “IMPLEMENTACIÓN DE UN PROTOTIPO PARA CONTAR PERSONAS EN VIDEO CAPTURADO EN TIEMPO REAL UTILIZANDO TÉCNICAS DE APRENDIZAJE PROFUNDO” realizado en el año 2020 [2], se destaca la importancia del desarrollo de dispositivos de visión artificial para la detección y seguimiento de objetos en videos. Estos dispositivos requieren la implementación de algoritmos complejos en sistemas computacionales de alto rendimiento. Aunque existen diversas técnicas holísticas y de aprendizaje automático utilizadas para la detección de objetos, actualmente se observa una tendencia hacia el uso de nuevas técnicas de procesamiento de imágenes y videos basadas en redes neuronales convolucionales. Estas técnicas han demostrado obtener resultados más confiables en cuanto a la detección de objetos y también han mejorado el tiempo de procesamiento. Esto implica avances significativos en la capacidad de contar personas en tiempo real y abre nuevas posibilidades para aplicaciones en diversas áreas.

En el proyecto de titulación “APLICACIÓN DE REDES NEURONALES PARA LA IDENTIFICACIÓN DE OBJETOS EN TIEMPO REAL EN IMÁGENES TOMADAS POR UN QUADROTOR” realizado en el año 2014 [3], se ha abordado la problemática de la identificación de objetos en tiempo real utilizando herramientas como los drones de tipo quadrotor para tareas de vigilancia. En este proyecto, se han propuesto e implementado diversas técnicas que permiten realizar esta identificación de manera

eficiente. Se han realizado comparaciones entre diferentes clasificadores estadísticos y redes neuronales para evaluar su desempeño en la identificación de objetos en tiempo real. Estos avances tienen importantes implicaciones en el campo de la vigilancia, ya que permiten utilizar herramientas autónomas como los drones para realizar tareas de monitoreo y detección de objetos de manera eficaz.

En el proyecto de titulación titulado “USO DEL CONTADOR DE PERSONAS OBJETO EN TIEMPO REAL DETECCIÓN Y UN PEQUEÑO RED NEURONA” realizado en el año 2018 [4], el autor ha explorado el uso de redes neuronales profundas y diferentes entornos de desarrollo para la creación de un contador personal y el reconocimiento de personas utilizando redes neuronales. El objetivo del proyecto fue probar diferentes entornos de desarrollo y optimizar el modelo de reconocimiento y seguimiento de objetos. Como resultado, se logró desarrollar un programa funcional que utiliza técnicas de aprendizaje profundo y redes neuronales para realizar tareas de computación personal. El autor destacó que la optimización del programa podría conducir a una mejor actualización de pantalla y mayor precisión en los cálculos realizados. Estos avances son relevantes en el campo de la detección y reconocimiento de personas en tiempo real, y pueden tener aplicaciones en áreas como la videovigilancia y la seguridad.

Con base en las consideraciones expuestas anteriormente, se plantea desarrollar un algoritmo que mejore la detección de objetos. El objetivo principal es implementar un análisis que permita identificar y contar personas utilizando el modelo desarrollado. Esto implica ajustar los pesos del algoritmo para asegurar una detección precisa de los objetos y desarrollar un método que permita a la red neuronal reconocer objetos de una clase específica, en este caso, personas. Como resultado, se espera obtener un sistema capaz de contabilizar de manera automática el número de personas. Esta propuesta tiene el potencial de aplicarse en diversas áreas, como la videovigilancia, el control de aforo y la planificación de eventos, entre otros.

1.2 DESCRIPCIÓN DEL PROCESO

Dado que el algoritmo requiere entrenar una red neuronal en un plano vertical, se propone desarrollar un sistema que permita el conteo automático del flujo de personas. Este desarrollo se llevará a cabo a través de módulos o pasos que podrán adaptarse del artículo [5] titulado “DETECCIÓN DE OBJETOS CON PYTHON”. El objetivo principal de este

proyecto consiste en capturar aéreamente personas y contarlas utilizando técnicas y herramientas específicas. A continuación, se describirán los diferentes módulos que se seguirán para lograr este propósito, ejecutándolos de manera secuencial para garantizar una detección precisa y confiable del flujo de personas en el plano vertical.

Módulo de instalación:

- Se realizará la instalación de las versiones adecuadas de las librerías y paquetes necesarios para utilizar Machine Learning en Python.
- Se empleará la herramienta Make Sense para generar las etiquetas correspondientes a cada imagen y así facilitar la creación de la red neuronal.
- Se procederá a descargar el código de detección de imágenes con el fin de mejorar el entrenamiento de la red y evaluar los pesos obtenidos.

Módulo de imágenes:

- Se procede a crear el repositorio de imágenes que serán utilizadas para entrenar la red de detección.
- Se asegura la inclusión de fotografías con diferentes condiciones de iluminación, así como imágenes que presenten variaciones en la orientación y tamaño del objeto.
- Se utiliza Make Sense como editor de imágenes, permitiendo la creación de cajas delimitadoras (bounding-box) alrededor de cada objeto de interés.
- Las imágenes son almacenadas en un directorio específico y se genera un archivo en formato YOLO que registra las anotaciones de cada objeto, incluyendo las coordenadas de posición (x, y), altura y ancho.

Módulo de creación de la red:

- Se reserva un 20% de las imágenes del conjunto de datos para ser utilizadas como conjunto de validación y evaluar las métricas del modelo.
- Se emplea el código de detección de imágenes para realizar pequeñas modificaciones o ajustes, con el fin de mejorar la capacidad de la red neuronal.
- Se entrena la red convolucional utilizando un conjunto específico de imágenes en cada época de entrenamiento. Al finalizar el proceso, se genera un archivo que contiene los pesos de la nueva red creada.

- Se recomienda llevar a cabo el entrenamiento durante 50 a 60 épocas para lograr un mejor ajuste del modelo.
- Se evalúan las métricas considerando la cantidad de datos disponibles, se analiza la relación entre las variables y se obtiene un resultado mediante la predicción del modelo.

Módulo de entrenamiento de la red neuronal:

- Se realiza una prueba de la red utilizando capturas en tiempo real o imágenes nuevas que no ha sido expuesta previamente.
- Se utiliza el archivo de pesos generado durante el entrenamiento para recrear la red y realizar predicciones sin necesidad de volver a entrenarla.
- Se obtiene la información de salida que incluye la detección realizada en una imagen nueva.

En resumen, se ha logrado implementar con éxito la detección de objetos, permitiendo contar el número de personas presentes en un determinado lugar. Durante el desarrollo de este proyecto, se han utilizado diversas herramientas de código abierto, las cuales se describen a continuación:

Entorno de desarrollo

- ✓ **Jupyter Notebook:** es un entorno de desarrollo interactivo basado en el web diseñado para la creación y ejecución de códigos y análisis de datos. Es ampliamente utilizado en disciplinas como ciencia de datos, computación científica, periodismo computacional y aprendizaje automático. La interfaz flexible de Jupyter Notebook permite a los usuarios configurar y organizar sus flujos de trabajo de manera eficiente. Además, su diseño modular facilita la incorporación de extensiones que amplían y enriquecen sus capacidades funcionales. [6]

Lenguaje de programación

- ✓ **Python:** Es un lenguaje de programación de alto nivel ampliamente utilizado en el desarrollo de software. Se destaca por su enfoque en la legibilidad del código y su capacidad para facilitar el desarrollo rápido de aplicaciones. [7]

Framework

- ✓ **Yolo:** Es un sistema de detección de objetos en tiempo real que se caracteriza por su velocidad y precisión. Utiliza una red neuronal convolucional para procesar imágenes a una velocidad de 30 fotogramas por segundo. La red divide la imagen en diferentes regiones y predice las cajas delimitadoras y las probabilidades asociadas a cada región, lo que permite detectar objetos de manera eficiente. [8]
- ✓ **Keras:** Es una biblioteca de redes neuronales de alto nivel escrita en Python. Construida sobre TensorFlow 2.0, Keras es compatible con otras bibliotecas como CNTK y Theano. Proporciona una interfaz sencilla y modular que facilita la experimentación con redes neuronales profundas. Es ampliamente utilizada debido a su facilidad de uso y su capacidad para crear modelos de aprendizaje automático de forma rápida y eficiente. [9]
- ✓ **Tensorflow:** Es una biblioteca de código abierto para cálculos numéricos de alto rendimiento, especialmente diseñada para el desarrollo de modelos de aprendizaje automático. Proporciona una amplia gama de herramientas y funciones que permiten construir y entrenar modelos de manera eficiente. TensorFlow es ampliamente utilizado en la comunidad de aprendizaje automático debido a su flexibilidad, escalabilidad y rendimiento. [10]

Librerías

- ✓ **Open CV:** Es una biblioteca de código abierto optimizada para aplicaciones en tiempo real. Está diseñada específicamente para mejorar la eficiencia computacional en el procesamiento de imágenes, aprovechando al máximo los recursos de los procesadores multinúcleo. OpenCV ofrece una amplia gama de funciones y algoritmos para el procesamiento de imágenes, detección de objetos, seguimiento de objetos, visión por computadora y otras tareas relacionadas. [11]
- ✓ **NumPy:** Es un paquete fundamental en Python que proporciona un soporte eficiente para el manejo de matrices y operaciones matemáticas en arrays multidimensionales. Se utiliza ampliamente en el procesamiento numérico y científico debido a su capacidad para realizar cálculos eficientes en matrices de datos. NumPy ofrece una amplia gama de funciones y métodos para manipular y

operar con matrices, lo que lo convierte en una herramienta poderosa para el procesamiento y análisis de datos. [12]

- ✓ **Matplotlib:** Es una biblioteca en Python que proporciona herramientas para la creación de gráficos y visualizaciones estadísticas. Permite crear una amplia variedad de gráficos, desde simples trazados de líneas y barras hasta diagramas de errores, histogramas y figuras interactivas y animadas. Matplotlib es ampliamente utilizado en el análisis de datos y la visualización debido a su versatilidad y facilidad de uso. [13]
- ✓ **Pandas:** Es una biblioteca de código abierto diseñada para el análisis de datos en Python. Proporciona estructuras de datos flexibles y eficientes, como el DataFrame, que permite manipular y analizar datos de manera rápida y eficiente. Pandas ofrece numerosas funciones y herramientas para la limpieza de datos, la manipulación de columnas, el filtrado de datos, el manejo de fechas y muchas otras tareas comunes en el análisis de datos. Es ampliamente utilizado en proyectos de ciencia de datos y análisis estadístico debido a su potencia y facilidad de uso. [14]

Este proyecto de investigación se centra en la línea de investigación de Tecnología y Sistemas de la Información, específicamente en el campo de Inteligencia Computacional. Se relaciona con aspectos de gestión de desarrollo de software y utiliza técnicas de Machine Learning para la preparación de datos, entrenamiento, detección y desarrollo de algoritmos de visión artificial. [15]

1.3 OBJETIVO

1.3.1 OBJETIVO GENERAL

Desarrollar un sistema de detección de personas mediante visión artificial utilizando redes neuronales convolucionales y algoritmos para la contabilización del número de personas.

1.3.2 OBJETIVOS ESPECIFICOS

- Analizar el proceso de detección de personas en un ángulo vertical para identificar deficiencias en la aglomeración de personas.
- Construir el datasets mediante búsquedas para la clasificación y reconocimiento de objetos.

- Categorizar el pre-procesamiento de imágenes con la captura en tiempo real para la limpieza y preparación de fotografías.
- Evaluar la precisión del 80% o superior en las pruebas de detección y reconocimiento de objetos del algoritmo, con especial énfasis en la identificación de personas.

1.4 JUSTIFICACIÓN

La inseguridad representa una amenaza directa a los derechos fundamentales del ser humano. Según los análisis presentados en el estudio [16], revelan que los delincuentes aprovechan las concentraciones masivas de personas en la comunidad para llevar a cabo delitos, en particular, aquellos conocidos como "delitos de oportunidad". Esta problemática genera un incremento significativo en el riesgo de inseguridad en el entorno en cuestión.

Se fundamenta en la demanda creciente de seguridad por parte de la ciudadanía, motivada por el incremento de incidentes violentos y delictivos en la región.

La provincia de Santa Elena, reconocida por su tranquilidad y hospitalidad, se destaca principalmente por su actividad turística. Sin embargo, se ha observado un preocupante incremento en la incidencia de muertes violentas en la región durante el año 2022 en comparación con el año anterior. Según datos recopilados [16], se ha registrado un total de 53 muertes violentas en lo que va del año, lo cual representa un aumento del 165% en comparación con el mismo período de 2021.

Es importante resaltar que la mayoría de estas muertes, específicamente 46, se concentran en el distrito La Libertad-Salinas. En contraste, en el mismo periodo del año anterior, solo se habían reportado 11 casos en esta zona, lo que evidencia un alarmante incremento del 318% en la incidencia de muertes violentas en dicha área.

En la Tabla 1 se presenta el índice delictivo de robos a personas por mes en Ecuador. Este estudio se realizó a partir de la pregunta "¿Los ciudadanos han sido víctimas de robo o asalto en los últimos cuatro meses?". Los datos muestran que desde enero hasta mayo de 2022 se han reportado un total de 12,548 casos, lo que representa casi la mitad del año en tan solo cinco meses, según el informe del Ministerio de Gobierno [17].

MES	2019	2020	2021	2022
ENERO	2.425	2.994	1.836	2.247
FEBRERO	2.392	2.614	1.901	2.334
MARZO	2.623	1.237	2.042	2.744
ABRIL	2.518	331	1.836	2.661
MAYO	2.513	738	1.871	2.562
JUNIO	2.496	1.274	2.139	
JULIO	2.591	1.700	2.272	
AGOSTO	2.562	1.586	2.196	
SEPTIEMBRE	2.506	1.765	2.186	
OCTUBRE	2.766	1.977	2.167	
NOVIEMBRE	2.726	1.956	2.222	
DICIEMBRE	2.883	1.954	2.721	

Tabla 1 : Información revelada por el Ministerio de Gobierno, acerca de robo a personas por mes en Ecuador durante los años de 2019 hasta 2022: El universo

A partir de estas conclusiones, se hace imprescindible la implementación de un sistema, que sea factible detectar la formación de aglomeraciones de personas, generando alarmas en tiempo real con el objetivo de prevenir situaciones de inseguridad.

El sistema de reconocimiento de personas es una herramienta crucial para mantener el control y la seguridad en áreas con alta densidad poblacional. Al implementar un sistema automatizado de conteo y monitoreo constante de individuos, se facilita la adopción de medidas preventivas, como políticas de control de acceso, asignación eficiente de recursos de seguridad y acciones adecuadas para evitar situaciones de riesgo. Esta solución optimiza la seguridad en entornos con gran afluencia de personas

La implementación de un sistema de conteo y monitoreo automático del flujo de personas ofrece una solución eficiente para garantizar la seguridad y prevenir situaciones de inseguridad en entornos con alta densidad poblacional. La capacidad de contar y monitorear de manera automatizada el movimiento de personas proporciona una herramienta efectiva para tomar medidas preventivas y mantener un entorno seguro.

Representa una solución efectiva para mitigar los riesgos y asegurar la seguridad en dichas áreas.

El reconocimiento de objetos, en particular de personas, plantea desafíos significativos en el ámbito de la informática. Aunque se han diseñado algoritmos especializados en el análisis de formas y la comparación de imágenes, hasta la fecha no se ha logrado desarrollar una solución generalizada que permita una detección precisa de objetos en un plano vertical.

El problema radica en la complejidad derivada de la variabilidad de formas y características de los objetos, junto con los desafíos asociados a la representación y procesamiento de imágenes en un entorno tridimensional. La ausencia de una solución genérica y eficiente en este contexto impulsa la necesidad de investigar y desarrollar nuevas técnicas y enfoques para mejorar la capacidad

En el contexto actual, el objetivo principal de este proyecto es el desarrollo de un algoritmo que mejore la capacidad de detección y reconocimiento de objetos, con un enfoque específico en la identificación de personas. El propósito es asegurar una alta precisión en las pruebas de detección y reconocimiento, con el fin de reducir al mínimo los falsos positivos y lograr una identificación confiable de las personas objetivo.

Para alcanzar este objetivo, se explorarán diversas técnicas y enfoques de vanguardia en el campo del reconocimiento de objetos, como el análisis de características, el aprendizaje automático y la visión por computadora. Además, se llevarán a cabo rigurosas evaluaciones y pruebas para validar la eficacia y la precisión del algoritmo propuesto.

Este proyecto tiene como objetivo proporcionar una solución efectiva y confiable para mejorar la identificación de personas en diversos escenarios y contribuir a la seguridad y el rendimiento de aplicaciones relacionadas con el reconocimiento de objetos.

Por lo tanto, este proyecto se justifica por la necesidad de abordar la ineficiencia actual en la identificación de personas, especialmente en lugares de monitoreo, y por la importancia de contar con un sistema preciso y confiable que permita identificar y contar personas de manera automática.

El tema propuesto [18], está alineado a los objetivos del “Plan Nacional de Desarrollo Creación de Oportunidades” específicamente en siguiente eje:

Eje 1.- Social

Objetivo 5.- Proteger a las familias, garantizar sus derechos y servicios, erradicar la pobreza y promover la inclusión social [18].

Política 5.3.- Consolidar un sistema de seguridad social universal, eficiente, transparente y sostenible, en corresponsabilidad entre el estado, el sector privado y la ciudadanía [18].

Eje 2.- Seguridad Integral

Objetivo 9.- Garantizar la seguridad ciudadana, orden público y gestión de riesgo [18].

Política 9.1.- Fortalecer la protección interna, el mantenimiento y control del orden público, que permita prevenir y erradicar los delitos conexos y la violencia en todas sus formas, en convivencia con la ciudadanía en el territorio nacional y áreas jurisdiccionales [18].

Política 9.3.- Impulsar la reducción de riesgos de desastres y atención oportuna a emergencias ante amenazas naturales o antrópicas en todos los sectores y niveles territoriales [18].

1.5 ALCANCE DEL PROYECTO

Se procederá a implementar un modelo de reconocimiento de objetos utilizando herramientas y módulos de desarrollo especializados. El objetivo principal será analizar y medir la confiabilidad mediante la detección de personas en un plano vertical. Para alcanzar este objetivo, se tomarán en cuenta los siguientes aspectos clave:

- **Módulo de instalación:** Se llevará a cabo una investigación exhaustiva de lenguajes de programación compatibles con machine learning y redes neuronales. Se seleccionará el lenguaje más apropiado y se empleará la Suite Anaconda como entorno de desarrollo integrado (IDE) para realizar la instalación y configuración del entorno de trabajo. Además, se integrará la herramienta Jupyter Notebooks para facilitar la programación interactiva y se descargarán los archivos requeridos para el desarrollo del proyecto.
- **Módulo de imágenes:** Será necesario obtener un conjunto de imágenes para el proceso de entrenamiento del algoritmo de detección de objetos. Cada imagen deberá ir acompañada de un archivo XML en un formato específico que contenga

información detallada sobre la clase y la posición de cada objeto presente en la imagen.

- **Módulo de creación de la red:** En esta etapa, además de la validación del conjunto de datos y la aplicación del aumento de datos, se llevará a cabo la configuración y ajuste de los hiperparámetros de la red neuronal. Se realizarán pruebas exhaustivas para optimizar su desempeño y garantizar una detección precisa de los objetos objetivo. Se utilizarán técnicas de aprendizaje profundo y algoritmos de optimización para lograr un rendimiento óptimo en la identificación y propuesta de predicciones.
- **Módulo de entrenamiento de la red neuronal:** En esta fase, se procesarán las imágenes de entrenamiento para generar los pesos de la red neuronal convolucional. Con el entrenamiento completo, se podrán realizar pruebas con nuevas imágenes sin necesidad de reentrenar. Además, se evaluará el rendimiento de la red mediante métricas específicas para medir su precisión y eficacia en la detección y reconocimiento de objetos.

1.6 METODOLOGÍA

1.6.1 METODOLOGÍA DE INVESTIGACIÓN

Para abordar el proyecto propuesto, se utilizará la metodología de investigación exploratoria. Se llevará a cabo una búsqueda exhaustiva y un análisis de la investigación relacionada, incluyendo diversos modelos utilizados actualmente [19]. Se compararán las estructuras de las obras relacionadas para identificar similitudes y diferencias con las propuestas presentadas, lo que permitirá obtener una comprensión más profunda y generar nuevas ideas y soluciones.

Con base en el estudio de diagnóstico [19], se utilizará para lograr el objetivo de comprender el modelo a desarrollar para la implementación genérica de detección de personas en un plano vertical, donde el servidor pueda reconocer a una persona. Este estudio proporcionará información clave y guiará el diseño del modelo, permitiendo la creación de un sistema efectivo y preciso para la detección de personas en dicho contexto.

Las técnicas de KDD (Knowledge Discovery in Databases) [20], ilustradas en la figura 1, son un proceso interactivo e iterativo utilizado para extraer conocimiento a partir de la recopilación de datos. Este proceso consta de varias etapas interrelacionadas, que

incluyen la selección de datos, preprocesamiento, transformación, minería de datos y evaluación de resultados. La naturaleza iterativa del enfoque permite ajustar y refinar el proceso en función de los resultados obtenidos, lo que mejora la calidad y relevancia de los conocimientos descubiertos a lo largo del tiempo.

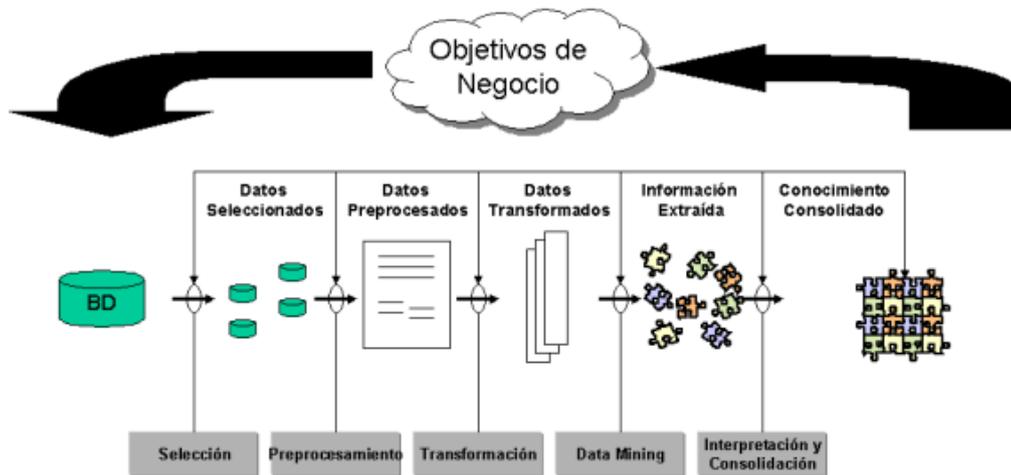


Figura 1: Metodología para el descubrimiento de conocimiento en bases de datos: J. M. Molina López y J. García Herrero

Dentro del alcance de este proyecto, se ha desarrollado un método de observación para recolectar información pertinente en el lugar donde se llevará a cabo la actividad. Este método ha sido diseñado con el propósito de obtener una muestra representativa de la captura de datos que nos permitirá identificar situaciones de inseguridad y prevenir el incremento de los índices de robo. La información recolectada será utilizada para implementar medidas preventivas y asegurar la seguridad.

1.6.2 VARIABLES

- Tiempo que tarda el algoritmo identificando en un entorno vertical para generar respuesta conforme al conteo de flujo de persona.

1.6.3 TECNICAS DE RECOLECCIÓN DE INFORMACIÓN

Para la recopilación de datos, se utilizarán técnicas como la extracción de información de sitios web oficiales en Internet, la realización de experimentos y el aprovechamiento de proyectos anteriores de recopilación de datos. Estas estrategias se implementarán con el fin de obtener los datos necesarios para garantizar el correcto funcionamiento del sistema.

1.6.4 METODOLOGIA DE DESARROLLO

El método por desarrollar implica la creación de un conjunto de datos que servirá para determinar el procedimiento necesario para el entrenamiento y validación del modelo propuesto. Este proceso se llevará a cabo mediante una serie de pasos lógicos, que incluyen la búsqueda, selección, entrenamiento y validación de los datos. Estas etapas son fundamentales para lograr una implementación exitosa de la metodología del modelo incremental de las redes neuronales convolucionales [21].

A continuación, se proporciona una descripción detallada de cada una de las etapas mencionadas, que se representan en la figura 2.

- 1) **Investigación y búsqueda:** Se recopila la información necesaria para el desarrollo del sistema. Se exploran diferentes fuentes de datos y se realiza un análisis detallado para identificar los requisitos y objetivos del proyecto.
- 2) **Preparación y selección:** Se crea un diagrama de flujo que representa el modelo de relación de casos de uso e interfaz del sistema. Se realiza una cuidadosa selección de los componentes y funcionalidades que serán implementados, teniendo en cuenta los requisitos identificados en la etapa anterior.
- 3) **Implementación y entrenamiento:** Se lleva a cabo la codificación de los módulos y componentes del sistema utilizando un lenguaje de programación adecuado. Se aplican las técnicas y algoritmos necesarios para lograr el funcionamiento deseado. Además, se realiza el entrenamiento del modelo, ajustando sus parámetros y optimizando su rendimiento.
- 4) **Prueba y validación:** Se realiza una exhaustiva prueba del sistema para detectar errores y fallas. Se llevan a cabo pruebas funcionales, de rendimiento y de seguridad para garantizar que el producto final sea funcional y cumpla con los requisitos establecidos. Los errores encontrados se corrigen y se validan los resultados obtenidos, asegurando que el sistema funcione de manera satisfactoria.

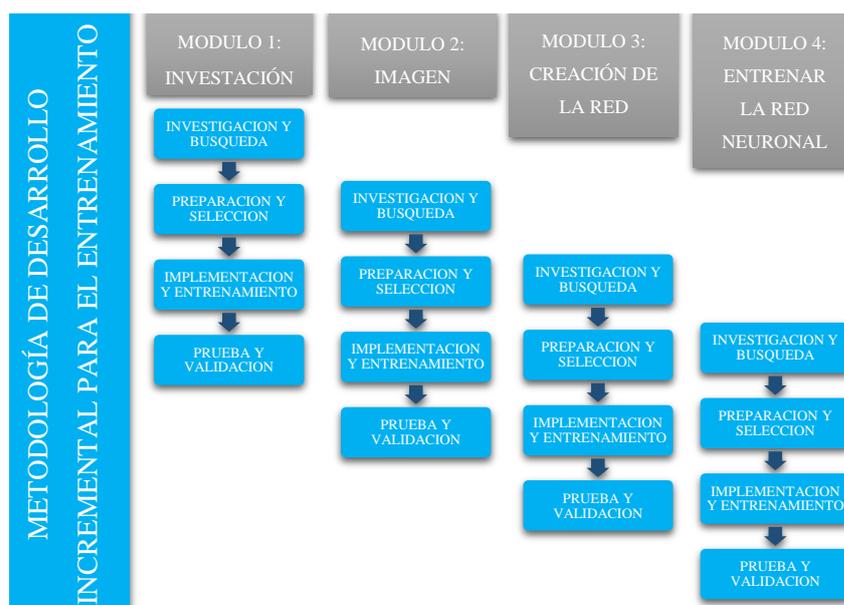


Figura 2: Metodología de desarrollo incremental para el entrenamiento de una red neuronal. Elaboración propia.

CAPITULO II: 2. MARCOS DEL PROYECTO

2.1 MARCO CONTEXTUAL

2.1.1 El GAD (Gobierno Autónomo Descentralizado del cantón salinas)

El Gobierno Municipal del cantón Salinas tiene como compromiso trabajar de manera transparente en beneficio de los ciudadanos de Salinas [22]. Su enfoque se centra en fortalecer las políticas públicas con el objetivo de alcanzar metas comunes en concordancia con las competencias constitucionales y legales establecidas. Según los resultados del Censo del 2010, se determinó que la población en la provincia se concentra principalmente en jóvenes, y específicamente en Salinas se estima una población total de 68.675 personas.

2.1.2 MISIÓN

Promover el desarrollo sustentable garantizando el derecho al Buen Vivir a través de proyectos de equidad, inclusión, ordenamiento territorial, participación ciudadana, medio ambiente, seguridad ciudadana, cultura, actividades deportivas en beneficio de sus habitantes. [23]

2.1.3 VISIÓN

Consolidar la institución en el desarrollo socio económico y productivo, ofreciendo mejores servicios públicos con calidad y eficacia, impulsando microproyectos dentro de las áreas de competencia a beneficio de los habitantes del Cantón. [23]

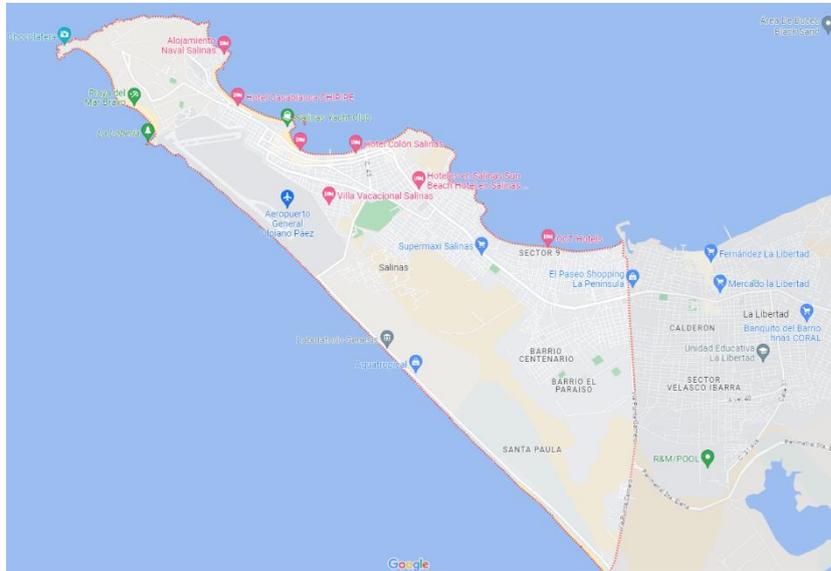


Figura 3: Ubicación Geográfica de Salinas: Google maps

2.1.4 LEY ORGÁNICA DE TRANSPARENCIA Y ACCESO A LA INFORMACIÓN PÚBLICA

La Ley Orgánica De Transparencia Y Acceso A La Información Pública (LOTAIP) garantiza el derecho acceder a las fuentes de información con el fin de ejercer la participación democrática respecto de la utilización de archivos públicos a excepción de aquellas que por seguridad nacional no deben ser conocidas [24]:

ARTÍCULOS	CONTENIDO
Art. 1	Principios de Publicidad de la Información Pública.
Art. 3	Ámbito de aplicación de la ley.
Art. 4	Principios de aplicación de la ley.
Art. 5	Información pública.

Art. 7	Difusión de la información pública.
Art. 8	Promoción del derecho de acceso a la información.
Art. 9	Responsabilidad sobre la entrega de la información pública.
Art. 23	Sanción a funcionarios y /o empleados públicos y privados.

Tabla 2. Artículos de Ley Orgánica De Transparencia Y Acceso A La Información Pública: CPCCS

2.1.5 LEY ORGÁNICA DE PARTICIPACIÓN CIUDADANA Y LEY ORGÁNICA DEL CONSEJO DE PARTICIPACIÓN CIUDADANA Y CONTROL SOCIAL

La Ley Orgánica de Participación Ciudadana y Control Social (LOPC) y la Ley Orgánica de Consejos Provinciales y Municipales de Planificación y Coordinación de la Competencia Sectorial (LOCPCCS) tiene el fin de garantizar su efectiva implementación e integrar los procesos diarios de la participación ciudadana y gestión institucional con los siguientes artículos [24]:

LOPC	LOCPCCS	CONTENIDOS
	Art. 5 N. 2	CPCCS establece mecanismos de Rendición de Cuentas.
Art. 88		Derecho ciudadano a la Rendición de Cuentas.
Art. 89	Art. 9	Definición de Rendición de Cuentas.
Art. 90	Art. 11	Sujetos obligados a rendir cuentas.
Art. 91		Objetivos de la Rendición de Cuentas.
Art. 92	Art.10	Contenidos de la Rendición de Cuentas que corresponde al nivel político.

Art. 93	Art.10	Contenidos de la Rendición de Cuentas que corresponde al nivel programático y operativo.
Art. 94		Mecanismos para rendir cuentas.
Art. 95		Periodicidad de la Rendición de Cuentas.
	Art. 11	Sanciones.
	Art. 12	Monitoreo.
Art. 96		El libre acceso a la información pública.
Art. 97		Principios generales del acceso a la información.
Art. 98		Transparencia de la administración pública.
Art. 60, N. 4		Funciones de las asambleas locales: organizar Rendición de Cuentas.

Tabla 3: Artículos De Participación Ciudadana y Ley Orgánica: CPCCS

2.2 MARCO CONCEPTUAL

2.2.1 Inteligencia artificial

La inteligencia artificial (IA) es un campo de estudio de la informática que se enfoca en el desarrollo de sistemas y programas capaces de realizar tareas que requieren inteligencia humana, como percepción, razonamiento, aprendizaje y toma de decisiones autónomas. Su objetivo es crear máquinas que imiten algunas funciones cognitivas humanas y brinden soluciones eficientes en diferentes áreas [25].

2.2.1.1 Redes neuronales

Son modelos computacionales que se basan en el funcionamiento del cerebro humano. Son una herramienta clave en el aprendizaje automático y la inteligencia artificial. Estas redes están compuestas por un conjunto de nodos interconectados, conocidos como neuronas artificiales o unidades de procesamiento. Cada neurona realiza cálculos y

transmite la información a través de conexiones ponderadas, permitiendo el procesamiento y la interpretación de datos complejos [26].

La figura representa la arquitectura de una red neuronal, mostrando la interconexión de las neuronas artificiales y las conexiones ponderadas que permiten el procesamiento de datos en el campo de la inteligencia artificial [27].

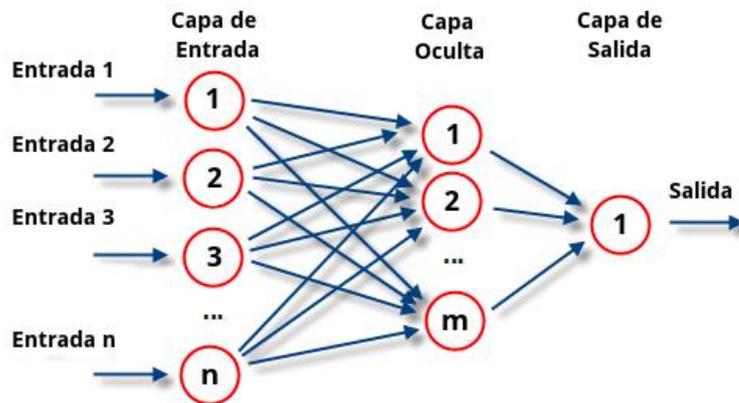


Figura 4: Forma de la operatividad del funcionamiento: Innovation, Atria.

2.2.1.2 Redes neuronales convolucionales

Las (CNN) son arquitecturas de redes neuronales profundas utilizadas para procesar datos gráficos, como imágenes y videos. Han logrado avances significativos en tareas de visión por computadora, como clasificación de imágenes, detección de objetos, segmentación semántica y reconocimiento facial. Su diseño se basa en operaciones convolucionales, y capas totalmente conectadas para extraer características y analizar datos visuales. Son fundamentales en el campo de la inteligencia artificial [28].

Tienen una capacidad jerárquica de aprendizaje que les permite capturar características abstractas en datos visuales. Son efectivas en el procesamiento de imágenes y han impulsado avances en la visión por computadora. Se entrenan utilizando algoritmos supervisados y han mejorado la clasificación de imágenes y la detección de objetos [28].

La figura 6 Representación del funcionamiento de las (CNN) en el procesamiento de datos visuales [29].

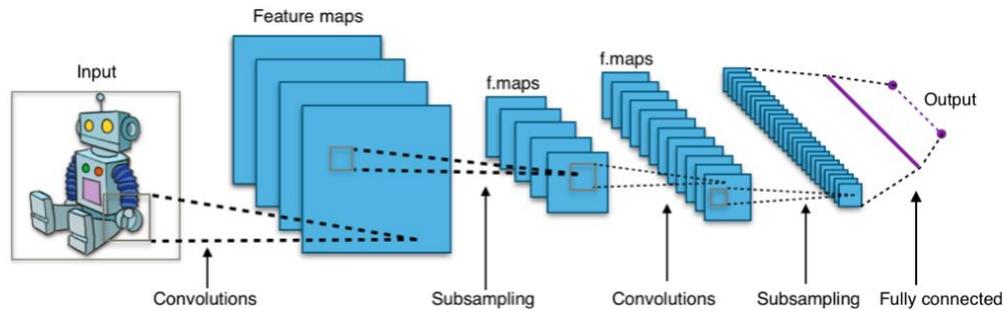


Figura 5: Funcionamiento de las redes neuronales convolucionales: Datademia.

2.2.1.3 Machine Learning o Aprendizaje automático

El aprendizaje automático, como subcampo de la inteligencia artificial (IA), se enfoca en el desarrollo de algoritmos y modelos que permiten a las máquinas aprender y mejorar automáticamente a partir de los datos. En lugar de ser programadas explícitamente para tareas específicas, las máquinas utilizan análisis de patrones y construcción de modelos matemáticos para realizar predicciones, tomar decisiones y extraer conocimientos de nuevos datos no vistos anteriormente [30].

2.2.1.4 Deep Learning o Aprendizaje profundo

Es una disciplina perteneciente al campo del aprendizaje automático que se centra en la creación de modelos de inteligencia artificial basados en la estructura y funcionamiento del cerebro humano, particularmente en las redes neuronales artificiales [31].

En contraste con los modelos convencionales de aprendizaje automático, que emplean una o pocas capas de neuronas artificiales, los modelos de aprendizaje profundo poseen la capacidad de adquirir y representar de forma automática patrones y características complejas a través de redes neuronales profundas, las cuales se componen de múltiples capas de neuronas interconectadas [31]

2.2.1.5 Algoritmos de visión para detección de objetos

Se ha consolidado como una herramienta de amplio uso en diversas aplicaciones del mundo real, entre las que se incluyen la detección de incendios en vídeos, la conducción autónoma, el monitoreo de seguridad y el análisis de escenas por medio de drones. Los algoritmos de detección de objetos se clasifican en dos categorías principales: los algoritmos tradicionales basados en el procesamiento de imágenes y los algoritmos basados en redes neuronales convolucionales [32].

El aprendizaje profundo ha avanzado gracias al mejor rendimiento del hardware y conjuntos de datos de anotación de imágenes a gran escala. Algoritmos basados en redes neuronales convolucionales han demostrado éxito en tareas de visión artificial como clasificación, detección y segmentación de imágenes. Estos algoritmos ofrecen mayor precisión y tiempo de prueba reducido, siendo la elección predominante en esta área [32]. La Figura 7 ilustra el proceso de reconocimiento de objetos en una imagen, mostrando cómo los algoritmos de aprendizaje automático y las técnicas de visión artificial se aplican para identificar y clasificar objetos presentes en una imagen determinada [33].



Figura 6: Reconocimiento de objetos en imagen: Darkcrizt.

2.2.1.6 Algoritmo de vision para seguimiento de objetos

El seguimiento de objetos es una técnica ampliamente utilizada en el ámbito de la visión por computadora, con aplicaciones diversas que incluyen vigilancia, interacción humano-computadora, obtención de imágenes médicas, monitoreo del flujo de tráfico y reconocimiento de actividades humanas. Un ejemplo relevante es su aplicación en seguridad nacional, donde el seguimiento de objetos puede ser utilizado para rastrear a un delincuente en fuga a través de cámaras de vigilancia urbanas [34].

Dentro de la API de seguimiento de OpenCV se incluyen varios algoritmos tradicionales de seguimiento que no se basan en el aprendizaje profundo. Estos rastreadores, en comparación con aquellos basados en el aprendizaje profundo, suelen presentar una precisión limitada en sus resultados [34].

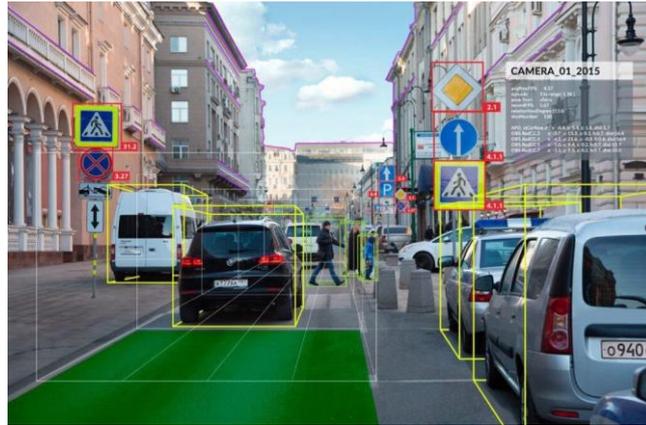


Figura 7: Visión artificial seguimiento: Sandra.

La Figura 8 muestra un ejemplo visual de la aplicación de técnicas de visión artificial en el contexto del seguimiento de objetos, destacando la capacidad de los algoritmos y métodos utilizados para rastrear y seguir objetos en una escena específica [35].

2.2.1.7 Algoritmo de vision para conteo de objetos

Los algoritmos de visión para el conteo de objetos son técnicas empleadas en el ámbito de la visión por computadora para llevar a cabo el conteo y estimación del número de objetos presentes en una imagen o secuencia de imágenes. Estos algoritmos encuentran una amplia gama de aplicaciones, como el monitoreo de tráfico, el seguimiento de personas, el conteo de células en imágenes médicas y el control de inventario, entre otros casos de uso [36].

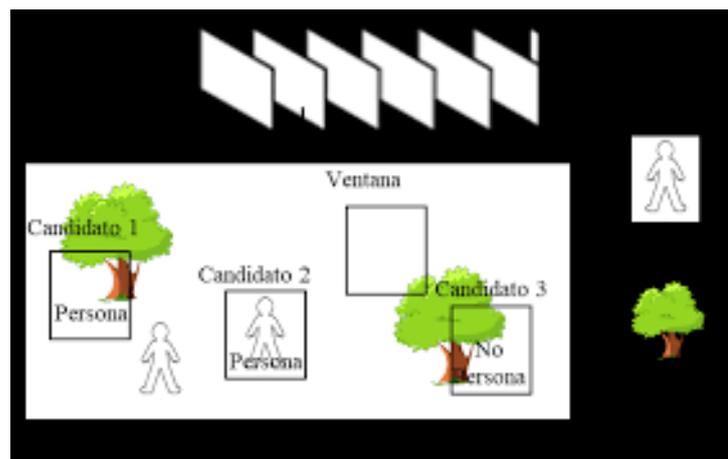


Figura 8: Elementos de un sistema de conteo: Luna, Andrés; Rodríguez, Nicolás.

La Figura 9 muestra los elementos clave que conforman un sistema de conteo, resaltando los componentes esenciales necesarios para llevar a cabo el proceso de conteo de manera efectiva y precisa [37].

En el conteo de objetos, se emplean diversos enfoques y algoritmos. Algunos métodos tradicionales se fundamentan en características visuales, como la detección de bordes, la segmentación de objetos y la extracción de características específicas. Estos enfoques a menudo implican la necesidad de una calibración y configuración manual para adaptarse a distintos tipos de objetos y entornos [36].

2.2.2 Vision artificial

La visión por computadora se dedica al procesamiento y análisis de imágenes y videos, posibilitando que las máquinas comprendan y extraigan información visual de manera automática. Esta disciplina se aplica en una amplia gama de aplicaciones, tales como el reconocimiento facial, la detección de objetos y la interpretación de escenas visuales [38].

2.2.2.1 Imágenes y espacio de color

Una imagen se puede entender como una composición de elementos ubicados en posiciones específicas y con valores asignados. En el contexto de las imágenes digitales, estos elementos se denominan píxeles, los cuales son responsables de definir la resolución de la imagen. La resolución de una imagen depende de la cantidad de píxeles que la conforman. Cada píxel se representa mediante un número que varía en un rango de 0 a 255, lo cual indica la intensidad luminosa en una escala de grises. En este sentido, una imagen se presenta en una estructura bidimensional compuesta por filas y columnas [39].

Para la codificación de un píxel, se emplea un espacio de color, como el modelo RGB (rojo, verde y azul). En este sistema, la imagen se compone de tres matrices superpuestas y aditivas que representan los colores. Cada matriz corresponde a la intensidad de uno de los colores primarios (rojo, verde y azul). Al variar la intensidad en cada una de las matrices, se logra representar una amplia gama de colores en la imagen [39].

2.2.2.2 OpenCV

OpenCV es una biblioteca de visión por computadora que ofrece un conjunto de funciones y algoritmos optimizados para llevar a cabo diversas tareas relacionadas con el procesamiento de imágenes y videos. Su funcionalidad abarca desde operaciones básicas

de procesamiento de imágenes, como filtrado, transformaciones geométricas, detección de bordes y umbralización, hasta algoritmos más avanzados, como reconocimiento facial, detección de características, correspondencia de imágenes, análisis de movimiento y estimación de profundidad [40].

2.2.3 Procesamiento de imagen

Es un campo de estudio enfocado en la manipulación y análisis de imágenes digitales. Este campo se basa en la aplicación de técnicas y algoritmos para mejorar, modificar o extraer información útil de las imágenes. El procesamiento de imágenes digitales permite la extracción de información valiosa y mejora la interpretación visual de los datos contenidos en las imágenes [41].

2.2.3.1 Seguimiento o tracking

El seguimiento de objetos es una técnica ampliamente empleada para rastrear y seguir objetos en secuencias de imágenes o videos. Su objetivo principal consiste en determinar la trayectoria y el comportamiento de un objeto en movimiento a lo largo del tiempo. Para el seguimiento de objetos, se emplean diversos enfoques que se pueden clasificar en base a características, apariencia, modelos y aprendizaje automático [42].

2.2.3.2 Pesos entrenados

Los pesos de un modelo son los parámetros que se actualizan durante el proceso de entrenamiento de una red neuronal. Estos pesos son valores numéricos que determinan la influencia de cada neurona en la red y se utilizan para realizar predicciones o inferencias en la etapa de prueba. Durante el entrenamiento, los pesos se ajustan de forma iterativa con el objetivo de minimizar la discrepancia entre las predicciones generadas por el modelo y las salidas deseadas [43].

Representan los parámetros que se modifican durante el proceso. Estos pesos son valores numéricos asignados a las conexiones entre las neuronas y determinan la influencia de cada neurona en la red. Durante la fase de inferencia o prueba, los pesos se utilizan para realizar predicciones o inferencias. Durante el entrenamiento, los pesos se ajustan de manera iterativa mediante técnicas como el descenso de gradiente, con el objetivo de minimizar la discrepancia entre las predicciones generadas por el modelo y las salidas deseadas, lo cual permite mejorar el rendimiento y precisión del modelo [43].

2.2.3.3 Cajas delimitadoras (bounding boxes)

Una Bounding Box es una herramienta utilizada para delimitar y detectar con precisión los caracteres que forman un objeto, incluyendo su ubicación, forma y tamaño. Se compone de valores de medición (x_{min} , y_{min} , x_{max} , y_{max}) que representan las coordenadas de los puntos de esquina de la caja delimitadora. Su función principal es proporcionar una visualización mejorada del objeto en un espacio multidimensional. [44]

2.2.4 Modelo YOLOv5

Es un modelo de detección de objetos en tiempo real que se fundamenta en la arquitectura YOLO (You Only Look Once). A diferencia de otros métodos que descomponen la detección en múltiples etapas, YOLOv5 adopta un enfoque de una sola etapa, realizando la detección y clasificación de objetos en una sola pasada. Esta metodología presenta mejoras significativas en términos de precisión y velocidad en comparación con sus versiones anteriores. Implementado en PyTorch, aprovecha las características y funcionalidades de esta biblioteca de aprendizaje automático para lograr un rendimiento óptimo [45].

2.2.4.1 Mejor precisión

Las últimas versiones y avances en la detección y clasificación de objetos han logrado una mejora notable en la precisión. Estos avances se basan en mejoras en la arquitectura de la red neuronal, así como en la incorporación de técnicas avanzadas, como la atención espacial y las convoluciones de múltiples escalas. Estas mejoras permiten capturar con mayor precisión y detalle las características de los objetos en las imágenes, lo que resulta en una mejora significativa en la precisión de detección y clasificación [46].

2.2.4.2 Mayor velocidad

A pesar de su mayor precisión, el modelo YOLOv5 logra tiempos de inferencia más rápidos en comparación con su predecesor, YOLOv4. Esta mejora en la eficiencia se debe a la optimización de la arquitectura de red y a la implementación de técnicas como la eliminación de capas redundantes y la compresión de modelos. Estas técnicas permiten reducir la complejidad computacional y mejorar la eficiencia en los cálculos durante el proceso de inferencia, lo que se traduce en tiempos de ejecución más rápidos sin comprometer la precisión en la detección y clasificación de objetos [46].

2.2.4.3 Tamaños de modelo escalables

El modelo ofrece diferentes tamaños (pequeño, mediano, grande y extragrande) adaptados a distintos requisitos de rendimiento y recursos computacionales. Esto permite a los usuarios elegir el equilibrio adecuado entre velocidad y precisión según sus necesidades específicas [46].

2.2.4.4 Entrenamiento y personalización sencillos

Facilita el entrenamiento y la personalización del modelo. Proporciona herramientas y scripts predefinidos para tareas comunes, como el etiquetado de datos y la configuración del entrenamiento. Esto facilita a los desarrolladores adaptar para diferentes conjuntos de datos y aplicaciones específicas. [46]

2.2.4.5 Soporte para diferentes aplicaciones

La herramienta puede ser empleada en una amplia gama de aplicaciones que requieren detección de objetos en tiempo real, tales como vehículos autónomos, vigilancia por video, reconocimiento de peatones y detección de objetos en imágenes médicas, entre otros. Su versatilidad permite adaptarse a diferentes escenarios y requisitos, proporcionando una solución efectiva y eficiente para la detección de objetos en tiempo real en diversas aplicaciones [46].

2.2.4.6 Entrenamiento personalizado con Yolov5

Requiere un conjunto de datos representativo y recursos computacionales adecuados para un entrenamiento efectivo. Esto permite adaptar el modelo a tareas específicas de detección de objetos y mejorar su capacidad de generalización en dominios de aplicación particulares, lo que lo convierte en una opción para implementaciones personalizadas de detección de objetos [47].

2.2.4.7 Preparación del conjunto de datos personalizados

La construcción del conjunto de datos implica la recopilación de imágenes y anotaciones pertinentes al dominio de aplicación específico. Posteriormente, se procede a la etiquetación de los objetos de interés presentes en las imágenes. Este conjunto de datos anotado se utiliza para entrenar el modelo de detección de objetos. A través de este proceso, es posible adaptar el modelo a un dominio específico y mejorar su capacidad para detectar los objetos de interés particulares [48].

Al crear y entrenar un conjunto de datos personalizado con Yolov5, se obtiene un modelo preciso y efectivo para las necesidades específicas. Requiere tiempo y atención para asegurar la calidad y representatividad de los datos recopilados [48].

2.2.5 Lenguajes de programación

Son sistemas formales diseñados para transmitir instrucciones a una computadora y escribir programas informáticos. Estos lenguajes permiten a los programadores expresar lógica y algoritmos de manera estructurada y comprensible para las máquinas. A través de ellos, se pueden realizar diversas tareas, como cálculos matemáticos, manipulación de datos, interacción con dispositivos y creación de software. Los lenguajes de programación proporcionan un medio eficiente para la comunicación entre humanos y máquinas, permitiendo la implementación de soluciones informáticas [49].

2.2.5.1 Python

La flexibilidad de Python es un lenguaje flexible que permite la interoperabilidad con múltiples lenguajes, como C/C++ y .NET. Es especialmente adecuado para la construcción de herramientas de inteligencia artificial y aprendizaje automático [50].

2.2.5.2 Google Colab

La herramienta permite la creación y ejecución de código a través de un navegador web. Es especialmente útil para actividades de aprendizaje automático, observación e investigación de proyectos. Además, ofrece la conveniencia de guardar los trabajos en la nube, lo que facilita el acceso y la colaboración en proyectos desde diferentes dispositivos [51].

2.2.5.3 Jupyter Notebook

Su característica distintiva radica en la capacidad de ejecutar flujos de trabajo completos de ciencia de datos. Esta herramienta permite llevar a cabo tareas como la limpieza y preparación de datos, la creación y entrenamiento de modelos de aprendizaje automático, la visualización de datos y muchas otras funcionalidades. Proporciona un entorno integrado que facilita la realización de múltiples etapas del proceso de ciencia de datos de manera eficiente y con un alto grado de automatización [52].

2.2.6 Bibliotecas e herramientas

Las bibliotecas son conjuntos de código predefinido que ofrecen funciones y herramientas específicas para realizar tareas comunes de programación. Están diseñadas para extender la funcionalidad básica del lenguaje y mejorar la eficiencia y conveniencia en la realización de diversas tareas [53].

2.2.6.1 Keras

Es una biblioteca de redes neuronales de código abierto escrita en Python que admite redes recurrentes y convolucionales. Proporciona flexibilidad a los investigadores para probar y experimentar con diferentes arquitecturas de red con cambios mínimos. Su enfoque modular y secuencial permite crear redes neuronales complejas con mayor legibilidad del código [54].

2.2.6.2 TensorFlow

Es una plataforma de aprendizaje automático de código abierto que ofrece un conjunto completo y flexible de herramientas, bibliotecas y recursos comunitarios. Está diseñada para permitir a los investigadores desarrollar y aplicar las últimas técnicas en aprendizaje automático (ML), así como a los desarrolladores crear e implementar de manera sencilla aplicaciones basadas en ML. TensorFlow proporciona una infraestructura sólida y eficiente que facilita la construcción, entrenamiento y despliegue de modelos de ML en una variedad de entornos y dispositivos [55].

2.2.6.3 YoloV5

Este sistema de reconocimiento de objetos en tiempo real es una implementación de código abierto que ha sido previamente entrenada utilizando conjuntos de datos diversos. El modelo resultante ofrece una mayor precisión en la localización y el reconocimiento de objetos en comparación con otros algoritmos, incluyendo versiones anteriores de Yolo. Se considera un algoritmo robusto y ha demostrado un buen desempeño en investigaciones recientes relacionadas con la detección de objetos [56].

2.2.6.4 Pytorch

PyTorch es un framework basado en la biblioteca Torch que permite acelerar operaciones matemáticas utilizando hardware especializado. Es especialmente adecuado para el diseño y entrenamiento de arquitecturas de redes neuronales en entornos paralelos [57].

2.2.6.5 Make Sense

Es una herramienta de código abierto que permite etiquetar varios tipos de objetos, como rectas, líneas, puntos y polígonos. Facilita y agiliza el proceso de preparación de conjuntos de datos y se puede utilizar directamente en un navegador web. Admite diversos formatos de archivos de salida, como Yolo, Voc Xml, Vgg Json y CSV [58].

2.2.6.6 Tensorboard

Es una herramienta de visualización y seguimiento que ofrece una interfaz gráfica interactiva para visualizar y comprender el rendimiento y comportamiento de los modelos de aprendizaje automático. Permite analizar métricas clave, como precisión, pérdida y otras métricas de evaluación, a lo largo del proceso de entrenamiento de un modelo. Esto permite a los usuarios evaluar el rendimiento del modelo en diferentes etapas y realizar ajustes si es necesario para mejorar su desempeño [59].

2.2.6.7 Numpy

NumPy es una biblioteca de Python ampliamente utilizada en ciencia de datos y computación científica. Permite manejar eficientemente arreglos multidimensionales de datos (ndarrays), almacenando y manipulando grandes conjuntos numéricos de manera eficiente. Con funciones y operaciones matemáticas rápidas y sencillas, es una herramienta esencial para el procesamiento y análisis de datos en Python [60].

2.3 MARCO TEORICO

2.3.1 La inteligencia artificial como herramienta de estrategia y seguridad

La tecnología avanza constantemente y la inteligencia humana se integra cada vez más con las máquinas para facilitar la toma de decisiones. Esto genera mejoras en la sociedad y es fundamental para implementar estrategias en distintos ámbitos. La sinergia entre el conocimiento humano y las capacidades de las máquinas abre nuevas posibilidades para optimizar procesos y crear soluciones más efectivas [61].

La IA abarca diversas definiciones, como la minería de datos, que se refiere a un conjunto de técnicas y herramientas aplicadas para descubrir conocimiento implícito previamente desconocido. Por otro lado, el Big Data recopila información y mediante algoritmos genera una variedad de conocimientos que pueden ser aprovechados para obtener ventajas. Las Ciencias de datos, en cambio, extraen conocimiento a partir de grandes

volúmenes de información, tanto estructurada como no estructurada. La Visión por computadora es una herramienta que procesa y analiza imágenes del mundo real con el fin de que puedan ser interpretadas por un sistema computacional avanzado, entre otras aplicaciones [61].

2.3.2 Reconocimiento de patrones de imágenes a través de un sistema de visión artificial

El reconocimiento de patrones es un proceso mediante el cual los seres humanos interpretan el mundo y aprenden a distinguir patrones visuales, auditivos y sensoriales. No obstante, para automatizar este proceso, es necesario resolver problemas relacionados con la medición, validación, procesamiento e interpretación de información en el entorno de estudio. Algunos problemas comunes en el reconocimiento de patrones son la selección de atributos y prototipos. El objetivo principal es mejorar el reconocimiento de patrones, medir el proceso de reconocimiento y desarrollar un sistema de visión artificial para su implementación [62].

El reconocimiento de patrones es un campo de estudio multidisciplinario que involucra conceptos de ingeniería, computación y matemáticas. Su objetivo es el análisis de objetos físicos y/o abstractos con el fin de extraer información relevante para establecer propiedades entre conjuntos de dichos objetos. Estas propiedades nos permiten interpretar el mundo que nos rodea y comprender mejor su estructura y comportamiento [62].

2.3.3 Estrategia y avances recientes en detección y clasificación de objetos por medio de algoritmos de aprendizaje automático

El enfoque principal de esta investigación se centra en las aplicaciones del aprendizaje automático en el campo de la visión por computadora. Se evalúan las estrategias y algoritmos de aprendizaje automático utilizados en esta área, destacando las aplicaciones más recientes en la detección y clasificación de objetos, así como la extracción de información de imágenes, documentos gráficos y videos. El objetivo es evolucionar el aprendizaje automático y la visión por computadora, dotando a las computadoras de capacidades similares a las humanas, como la comprensión de datos y la toma de decisiones basadas en información pasada y presente [63].

Se emplean métodos establecidos de aprendizaje supervisado, no supervisado y semisupervisado. Las soluciones propuestas involucran la recopilación de datos, el entrenamiento de un modelo y el uso del modelo capacitado para realizar predicciones. Los modelos de detección y análisis de objetos encuentran aplicaciones en áreas como la prevención de colisiones de tráfico y el reconocimiento basado en posturas humanas. No obstante, uno de los desafíos principales en la aplicación del aprendizaje automático y la visión por computadora es la anotación o etiquetado de datos. El estudio se enfoca en investigar y evaluar analíticamente las aplicaciones del aprendizaje automático en la visión por computadora [63].

CAPITULO III: 3. PROPUESTA

3.1 REQUERIMIENTO

En esta sección, se detallan los requisitos funcionales y no funcionales que han sido identificados utilizando técnicas de recopilación de información. El propósito es proporcionar soluciones óptimas para cada uno de estos requisitos.

3.1.1 Requerimientos funcionales

ID	REQUERIMIENTO	DESCRIPCIÓN DEL REQUERIMIENTO
RF-001	Detección de personas	El algoritmo debe ser capaz de detectar la presencia de personas en cada fotograma del video.
RF-002	Conteo automático	El algoritmo debe contar automáticamente el número de personas que ingresan y salen de la escena, manteniendo un registro preciso del flujo de personas en tiempo real.
RF-003	Registro y visualización de resultados	El algoritmo debe proporcionar un mecanismo para registrar y almacenar los resultados del conteo, así como también mostrar visualmente la información en tiempo real.

RF-004	Ajuste de parámetros	El algoritmo debe permitir ajustar parámetros como sensibilidad de detección, umbral de seguimiento y otras configuraciones relacionadas para adaptarse a diferentes escenarios y condiciones de iluminación.
RF-005	Adquisición de datos	El algoritmo debe ser capaz de recibir y procesar datos de entrada que representan imágenes o videos de personas en movimiento.
RF-006	Captura de video	El algoritmo debe ser capaz de capturar el video en tiempo real de una fuente de video o de una cámara.
RF-007	Seguimiento de personas	El algoritmo debe realizar un seguimiento preciso de las personas detectadas en el video a lo largo del tiempo, para contar correctamente el flujo de personas que entran y salen de la escena.
RF-008	Interfaz de usuario	Se debe desarrollar una interfaz de usuario intuitiva que permita a los usuarios interactuar con el algoritmo, iniciar y visualizar los resultados.
RF-009	Exportación de datos	El algoritmo debe permitir exportar los datos de conteo para su posterior análisis.
RF-0010	Filtrado de detecciones falsas	El algoritmo debe implementar técnicas de filtrado para minimizar las detecciones falsas
RF-0011	Clasificación de direcciones de movimiento	El algoritmo debe ser capaz de clasificar la dirección del movimiento de las personas detectadas, ya sea entrando o saliendo de la escena, con el fin de realizar un conteo correcto.

RF-0012	Adaptabilidad a diferentes escenarios	El algoritmo debe ser adaptable a diferentes escenarios y condiciones de iluminación, lo que implica la capacidad de ajustar automáticamente los parámetros de detección y seguimiento según las condiciones.
RF-0013	Entrenamiento de la red neuronal	El algoritmo debe permitir el entrenamiento de una red neuronal utilizando un conjunto de datos etiquetados, con el fin de mejorar la precisión y la capacidad de generalización del modelo.
RF-0014	Evaluación del rendimiento	El algoritmo debe incluir métricas de evaluación para medir el rendimiento del modelo, como la precisión, con el fin de analizar y mejorar el desempeño del sistema.
RF-0015	Actualización y reentrenamiento del modelo	El algoritmo debe proporcionar la capacidad de actualizar y reentrenar el modelo de manera periódica, utilizando nuevos datos etiquetados, para adaptarse a cambios en la escena o mejorar la precisión con el tiempo.
RF-0016	Optimización del rendimiento	El algoritmo debe estar diseñado de manera eficiente para optimizar el rendimiento y la velocidad de procesamiento, especialmente cuando se trabaja con videos en tiempo real.
RF-0017	Seguridad y privacidad	El algoritmo debe cumplir con medidas de seguridad y privacidad adecuadas, como el manejo seguro de datos sensibles y la protección de la información personal de las personas detectadas.

RF-0018	Compatibilidad con diferentes formatos de datos	El algoritmo debe ser compatible con una variedad de formatos de datos, como imágenes en diferentes resoluciones, videos en distintos formatos y archivos con metadatos, para facilitar la integración con diversas fuentes de datos.
RF-0019	Visualización en tiempo real	El algoritmo debe proporcionar una interfaz de visualización en tiempo real que muestre el conteo de personas actualizado a medida que se detectan y se registran.
RF-0020	Re-identificación de personas	El algoritmo puede implementar técnicas de Re-identificación para distinguir personas individuales en la escena y realizar un conteo más preciso, incluso en situaciones donde ocurren cruces o superposiciones.
RF-0021	Tolerancia a variaciones de apariencia	El algoritmo debe ser capaz de manejar variaciones en la apariencia de las personas, como cambios de vestimenta o diferentes ángulos de visión, para mantener la precisión del conteo.
RF-0022	Optimización del consumo de recursos	El algoritmo debe estar diseñado para optimizar el uso de recursos computacionales, como memoria y procesamiento, para garantizar un rendimiento eficiente incluso en sistemas con recursos limitados.
RF-0024	Escalabilidad	El algoritmo debe ser escalable y capaz de manejar escenarios con grandes volúmenes de personas, ya sea en tiempo real o en el análisis retrospectivo de videos.

RF-0025 Pruebas y validación

El algoritmo debe ser sometido a pruebas y validaciones exhaustivas para asegurar su precisión y confiabilidad en diferentes escenarios y condiciones, utilizando conjuntos de datos de prueba representativos.

Tabla 4: Requerimiento funcionales: Elaboración propia.

3.1.2 Requerimientos no funcionales

ID	REQUERIMIENTO	DESCRIPCIÓN DEL REQUERIMIENTO
RNF-001	Rendimiento	El algoritmo debe tener un rendimiento óptimo, con tiempos de procesamiento rápidos y eficientes, para garantizar una detección y conteo en tiempo real sin retrasos significativos.
RNF-002	Precisión	El algoritmo debe ser altamente preciso en la detección y conteo de personas, minimizando los errores y falsos positivos/negativos, para proporcionar resultados confiables y precisos.
RNF-003	Mantenibilidad	El código del algoritmo debe estar bien estructurado, modularizado y documentado, facilitando su mantenimiento y permitiendo futuras mejoras o actualizaciones.
RNF-004	Eficiencia en el consumo de recursos	El algoritmo debe utilizar eficientemente los recursos del sistema, como memoria y capacidad de procesamiento, minimizando el consumo innecesario y optimizando el rendimiento global.
RNF-005	Tolerancia a fallos	El algoritmo debe ser capaz de manejar situaciones de fallos de forma adecuada, recuperándose de errores inesperados y

		minimizando la pérdida de datos o la interrupción del servicio.
RNF-006	Mantenimiento de modelo	El algoritmo debe proporcionar funcionalidades para mantener y actualizar el modelo de la red neuronal de manera eficiente, permitiendo su adaptación a cambios en los patrones de movimiento de las personas.
RNF-007	Internacionalización	El algoritmo debe ser compatible con diferentes idiomas y configuraciones regionales, permitiendo la adaptación de la interfaz y los mensajes en función de las necesidades del usuario.
RNF-008	Tiempo de respuesta	El algoritmo debe tener un tiempo de respuesta rápido, brindando resultados y actualizaciones en tiempo real, para proporcionar una experiencia de usuario fluida y sin demoras significativas.

Tabla 5: Requerimientos no funcionales: Elaboración propia.

3.2 COMPONENTES DE LA PROPUESTA

3.2.1 Diagrama de caso de uso

El diagrama de caso de uso es una representación visual que muestra las interacciones entre el usuario (desarrollador) y el sistema en diversos escenarios. Cada caso de uso enfatiza las actividades principales realizadas en cada fase del proyecto, desde el entrenamiento inicial del modelo hasta el conteo en tiempo real y el mantenimiento continuo del sistema.

Caso de Uso: Entrenar Modelo	
USUARIO	Desarrollador

FUNCIÓN Realizar las actividades necesarias para entrenar la red neuronal y prepararla adecuadamente.

DESCRIPCIÓN El desarrollador lleva a cabo una serie de pasos para entrenar la red neuronal y prepararla adecuadamente para el conteo de personas. El proceso incluye la recolección de un conjunto de datos, la etapa de preprocesamiento de imágenes para asegurar la calidad de los datos, la extracción de características relevantes de las imágenes, la configuración y entrenamiento de la red neuronal con el conjunto de datos, y finalmente, la evaluación del rendimiento del modelo para asegurar su efectividad en el conteo preciso de personas.

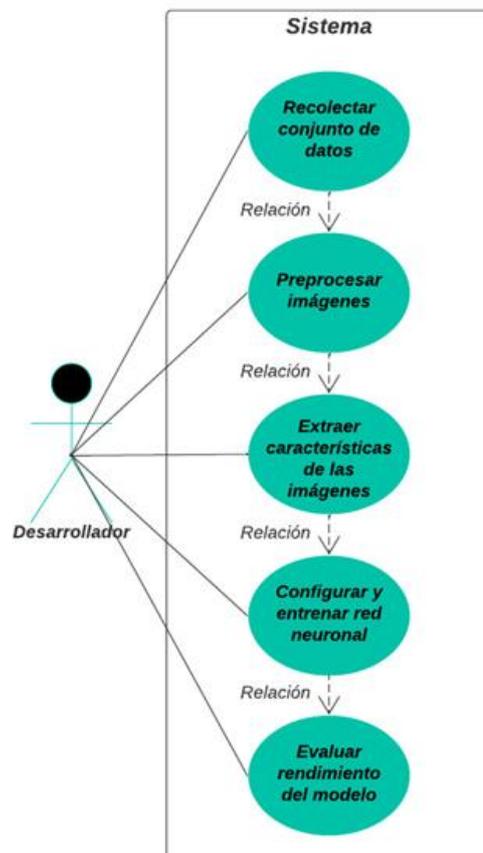


Figura 9: Diagrama de Caso de Uso: Entrenar Modelo de Conteo Automático de Personas: Elaboración propia.

PASOS

- Recolectar conjunto de datos

- Preprocesar imágenes
- Extraer características de las imágenes
- Configurar y entrenar red neuronal
- Evaluar rendimiento del modelo

Tabla 6: Caso de Uso: Entrenar Modelo: Elaboración propia.

Caso de Uso: Capturar y Procesar Datos

USUARIO	Desarrollador
FUNCIÓN	Se enfoca en las actividades relacionadas con la captura y procesamiento de los datos necesarios para el entrenamiento y la detección de personas.
DESCRIPCIÓN	Se centra en la obtención de datos en tiempo real, su preprocesamiento y su posterior procesamiento para la detección y el conteo de personas. Estas actividades son esenciales para garantizar que el sistema cuente con los datos adecuados

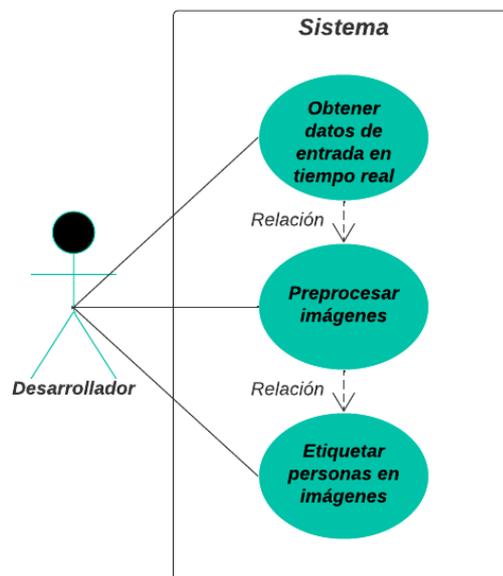


Figura 10: Diagrama de Caso de Uso: Capturar y Procesar Datos: Elaboración propia.

PASOS

- Obtener datos de entrada en tiempo real
- Preprocesar imágenes
- Etiquetar personas en imágenes

Tabla 7: Caso de Uso: Capturar y Procesar Datos: Elaboración propia.

Caso de Uso: Contar Personas

USUARIO

Desarrollador

FUNCIÓN

Automatiza la detección y conteo de personas en tiempo real mediante el uso del modelo entrenado en imágenes o videos capturados, agilizando el proceso de conteo.

DESCRIPCIÓN

El sistema realiza la captura en tiempo real de imágenes o videos y, mediante el modelo entrenado, lleva a cabo la detección y conteo de personas. Los resultados se registran y se muestran al usuario.

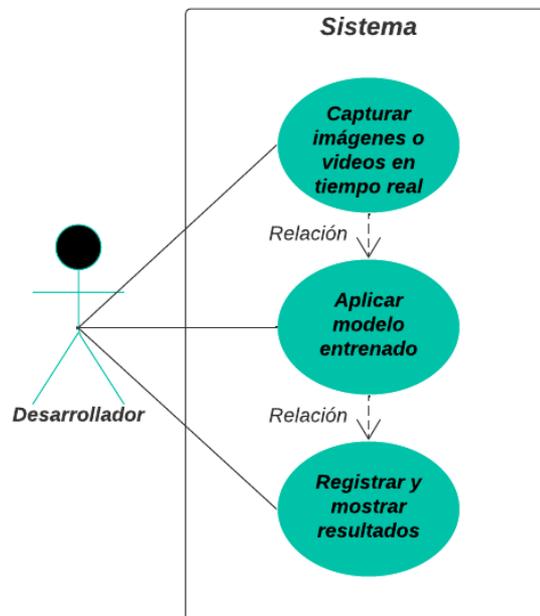


Figura 11: Diagrama de Caso de Uso: Contar Personas: Elaboración propia.

PASOS

- Capturar imágenes o videos en tiempo real
- Aplicar modelo entrenado para detección y conteo de personas
- Registrar y mostrar resultados

Tabla 8: Caso de Uso: Contar Personas: Elaboración propia.

Caso de Uso: Mantenimiento

USUARIO	Desarrollador
FUNCIÓN	Permite mantenerlo actualizado, optimizado y libre de errores. Garantiza que el sistema de conteo de personas se mantenga eficaz y capaz de adaptarse a nuevas situaciones y desafíos.
DESCRIPCIÓN	Se centra en el mantenimiento continuo del sistema. El desarrollador actualiza el modelo con nuevos datos, busca mejorar el rendimiento y realiza pruebas y ajustes periódicos para garantizar la precisión y efectividad del conteo.

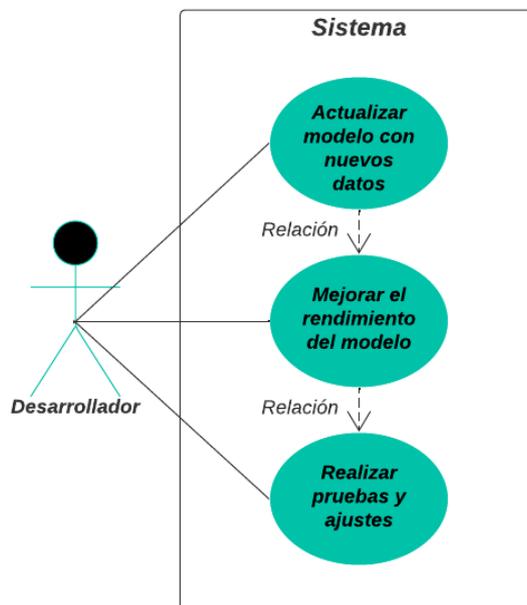


Figura 12: Diagrama de Caso de Uso: Mantenimiento: Elaboración propia.

PASOS

- Actualizar modelo con nuevos datos
- Mejorar el rendimiento del modelo
- Realizar pruebas y ajustes

Tabla 9: Caso de Uso: Mantenimiento: Elaboración propia.

3.2.1 Diagrama de bloques

El diagrama de bloques es una representación visual que muestra las diferentes etapas y componentes involucrados en el proyecto, proporcionando una visión clara de cómo se organizan y se relacionan entre sí [64]. En la figura 13, se pueden apreciar cada uno de los módulos del proyecto, destacando su estructura y la interconexión entre ellos.

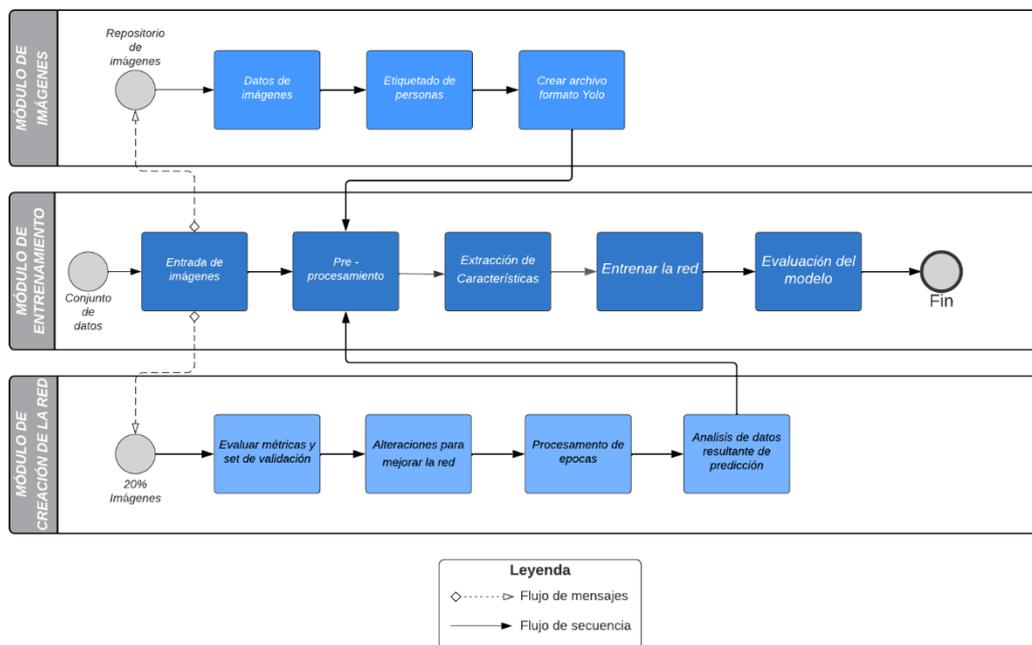


Figura 13: Diagrama de bloque: Elaboración propia.

Proceso	Descripción	Observaciones
Repositorio de imágenes	Almacenamiento centralizado de las imágenes utilizadas en el proceso.	<ul style="list-style-type: none"> - Organizar y estructurar adecuadamente el repositorio. - Garantizar acceso y seguridad.

Datos de imágenes	Extracción y recopilación de los datos de las imágenes necesarios para el modelo.	<ul style="list-style-type: none"> - Considerar calidad, cantidad y diversidad de los datos. - Realizar limpieza de datos si es necesario.
Etiquetado de personas	Anotación de las personas presentes en las imágenes para su posterior entrenamiento.	<ul style="list-style-type: none"> - Utilizar herramientas o métodos de etiquetado precisos. - Verificar calidad y coherencia de las etiquetas.
Crear archivo formato YOLO	Creación del archivo en el formato YOLO que contiene la información de las etiquetas.	<ul style="list-style-type: none"> - Seguir la estructura y el estándar del formato YOLO. - Verificar la correspondencia entre las etiquetas y las imágenes.

Tabla 10: Descripción del módulo de imágenes: Elaboración propia.

Proceso	Descripción	Observaciones
Inicio: Conjunto de datos	Estudio inicial del conjunto de datos que se utilizará en el proceso de entrenamiento y evaluación del modelo.	<ul style="list-style-type: none"> - Revisar calidad y disponibilidad de los datos. - Identificar posibles desequilibrios o sesgos.
Entrada de imágenes	Adquisición de las imágenes que serán utilizadas como datos de entrada para el modelo.	<ul style="list-style-type: none"> - Verificar formato y resolución de las imágenes. - Evaluar representatividad y diversidad.
Pre-procesamiento	Adquisición de las imágenes que serán utilizadas como datos de entrada para el modelo.	<ul style="list-style-type: none"> - Ajustes de tamaño, brillo, contraste, etc.

		- Eliminación de ruido o artefactos indeseados.
Extracción de características	Identificación de características relevantes de las imágenes.	- Utilizar métodos adecuados. - Considerar dimensionalidad y representatividad.
Entrenar la red	Ajuste de pesos y parámetros del modelo mediante el uso de los datos de entrada.	- Determinar número de iteraciones y tasa de aprendizaje. - Establecer hiperparámetros relevantes.
Evaluación del modelo	Medición del rendimiento y efectividad del modelo entrenado.	- Utilizar métricas de evaluación apropiadas. - Realizar validación cruzada o división adecuada de conjuntos.

Tabla 11: Descripción del módulo de entrenamiento: Elaboración propia.

Proceso	Descripción	Observaciones
20% Imágenes	Selección aleatoria del 20% de las imágenes del conjunto de datos para validación.	- Asegurar una representación adecuada del conjunto de datos de validación. - Evitar sesgos en la selección.
Evaluar métricas y set de validación	Cálculo de métricas de desempeño del modelo utilizando el conjunto de datos de validación.	- Utilizar métricas adecuadas para evaluar la precisión y el rendimiento del modelo. - Ajustar parámetros según los resultados obtenidos.

Alteraciones para mejorar la red	Realizar modificaciones en la arquitectura o hiperparámetros de la red para mejoras.	<ul style="list-style-type: none"> - Experimentar con diferentes configuraciones para encontrar mejoras significativas. - Evaluar el impacto de las alteraciones en el rendimiento del modelo.
Procesamiento de épocas	Entrenamiento del modelo a través de múltiples épocas para mejorar la precisión.	<ul style="list-style-type: none"> - Establecer el número adecuado de épocas para evitar sobreajuste o subajuste. - Supervisar el proceso de entrenamiento para controlar el progreso y la convergencia.
Análisis de datos de resultantes de predicción	Examinar los resultados de las predicciones del modelo para identificar patrones o errores.	<ul style="list-style-type: none"> - Analizar y comprender las tendencias y los errores en las predicciones. - Utilizar los hallazgos para realizar mejoras adicionales en el modelo.

Tabla 12: Descripción del módulo de creación de la red: Elaboración propia.

3.3 DESARROLLO DE LA PROPUESTA

Se enfocará en la detección y seguimiento de personas en imágenes o videos en tiempo real, lo que permitirá realizar un conteo preciso y automatizado del número de personas que ingresan y salen de la escena. Para ello, se recopilan y preprocesan conjuntos de datos que contienen imágenes o videos de personas en movimiento, los cuales servirán para entrenar la red neuronal. La red se configura y entrena utilizando técnicas de aprendizaje automático y visión por computadora, buscando mejorar la precisión del conteo.

Una vez entrenado el modelo, se evalúa su rendimiento para asegurar su efectividad en el conteo de personas. Posteriormente, se desarrolla una interfaz de usuario que permita a

los usuarios iniciar el conteo y visualizar los resultados en tiempo real. Además, se incorpora la opción de exportar los datos para análisis posterior.

El algoritmo resultante tiene diversas aplicaciones prácticas, como monitoreo de áreas con alto flujo de personas y tareas de seguridad, entre otras. Su implementación en Python proporciona flexibilidad y adaptabilidad, permitiendo su uso en diferentes escenarios y contextos. El desarrollo de esta herramienta contribuirá a automatizar y agilizar el proceso de conteo de personas, brindando una solución efectiva y de alto rendimiento.

3.3.1 Módulo de imágenes

El módulo se encarga de preparar y organizar el repositorio de imágenes para la red de detección. Aborda tareas como adquisición, preparación, etiquetado y generación de datos de entrenamiento. Es esencial para el funcionamiento del sistema.

3.3.1.1 Repositorio de imágenes

Se encarga de recopilar y almacenar un conjunto diverso de imágenes que servirán como datos de entrada para el entrenamiento de la red de detección. Estas imágenes deben representar diversas condiciones de luz, diferentes tamaños y orientaciones del objeto de interés. Se deben obtener imágenes que representen una variedad de condiciones de iluminación, ángulos de cámara y tamaño de objeto.

La Figura 14 representa el Repositorio de Imágenes, un componente esencial en el sistema propuesto. Aquí se almacenan y gestionan las imágenes para el entrenamiento y detección de personas.

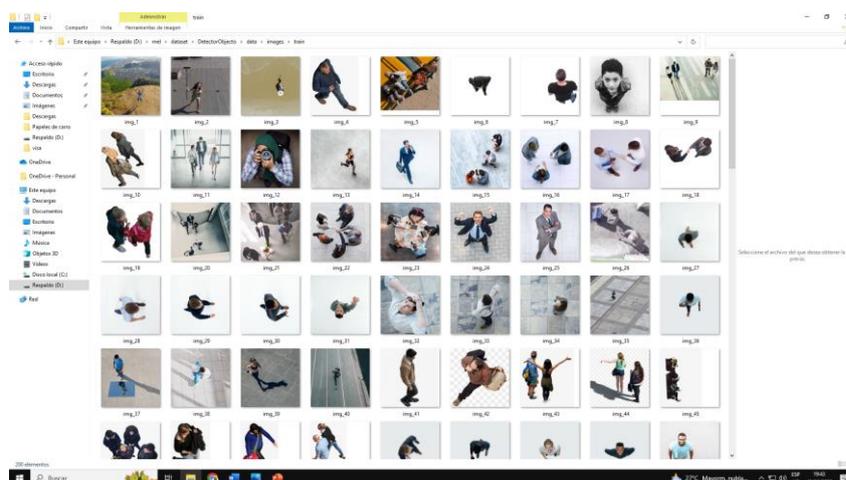


Figura 14: Repositorio de imágenes: Elaboración propia.

Considerando la siguiente información adicional: La selección y adquisición de imágenes para el repositorio es un aspecto fundamental en el desarrollo del proyecto. Se deben considerar y mencionar las fuentes de las cuales se obtuvieron las imágenes, ya sean cámaras de vigilancia, videos grabados en tiempo real o conjuntos de datos públicos.

Es fundamental recopilar un volumen adecuado de imágenes y garantizar la diversidad en condiciones de iluminación, ángulos de cámara y tamaño de objetos. Esto permitirá que el modelo de la red neuronal pueda generalizar y adaptarse eficazmente a diversos escenarios.

El proceso de adquisición y almacenamiento debe describirse detalladamente, incluyendo el uso de técnicas automatizadas [65], la gestión de metadatos asociados a cada imagen y la organización eficiente para facilitar el acceso y recuperación durante el entrenamiento.

Además, es esencial tener en cuenta las consideraciones legales, como obtener el consentimiento adecuado y cumplir con las leyes de privacidad y derechos de autor [66]. Si se realiza algún procesamiento adicional de las imágenes, como eliminación de distorsiones, reducción de ruido o mejora de la calidad, es importante mencionarlo para comprender cómo se manejan y mejoran las imágenes en esta etapa del proyecto.

3.3.1.2 Datos de imágenes

Es importante incluir imágenes con distintas condiciones de luz, ya que la red de detección debe ser capaz de reconocer y contar personas bajo diferentes niveles de iluminación. Además, se deben considerar imágenes con objetos torcidos y variados en tamaño para garantizar que la red pueda generalizar y detectar personas en situaciones diversas.

Antes de utilizar las imágenes en el entrenamiento de la red neuronal, es común aplicar técnicas de preprocesamiento para mejorar la calidad y la consistencia de las imágenes. Esto puede incluir el ajuste del tamaño de las imágenes, la normalización de los valores de píxeles, la corrección de distorsiones, la eliminación de ruido u otras operaciones que sean necesarias para optimizar el rendimiento del modelo [67].

Estas acciones contribuirán a obtener mejores resultados durante el entrenamiento de la red. Se considera iniciar con el conocimiento del esquema lumínico de cuatro puntos como muestra en la figura 15. Este esquema se refiere a una disposición de luces

estratégicamente ubicadas para iluminar el área de interés donde se capturan las imágenes para el conteo de personas. Conocer y aplicar este esquema lumínico te permitirá asegurar una iluminación adecuada en diferentes partes del área de detección, lo que facilitará la precisión del algoritmo de detección y conteo de personas [68].

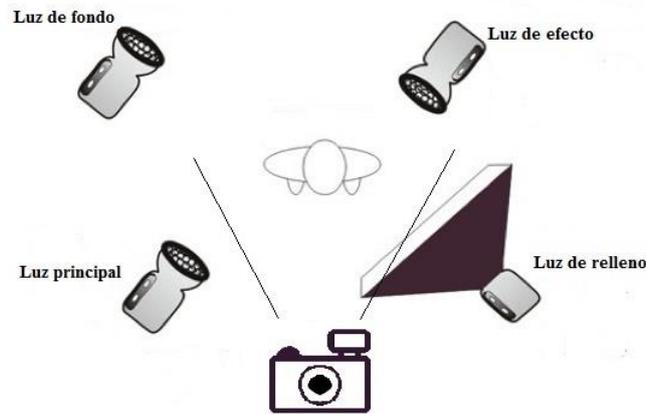


Figura 15: Esquema lumínico: Toomanyflash.

3.3.1.3 Etiquetado de personas

Se emplea una herramienta llamada Make Sense, un editor de imágenes, para realizar el etiquetado manual de cada objeto de interés. El siguiente paso consiste en asignar a cada objeto etiquetado una clase específica y crear una caja delimitadora alrededor de cada persona en las imágenes. Estas anotaciones son de suma importancia para el entrenamiento y la evaluación del modelo, ya que proporcionan las coordenadas (x, y) y las dimensiones (alto y ancho) de cada objeto presente en la imagen.

El etiquetado manual de personas en imágenes es esencial para entrenar la red neuronal en tareas de detección. Consiste en crear cajas delimitadoras alrededor de cada persona para indicar su posición y tamaño [69]. Esto proporciona las anotaciones necesarias para que la red aprenda a reconocer y contar personas con precisión.

La Figura 16 representa el proceso de etiquetado de las imágenes, una etapa crítica en el desarrollo del proyecto. Aquí se muestra cómo se utiliza la herramienta de etiquetado.

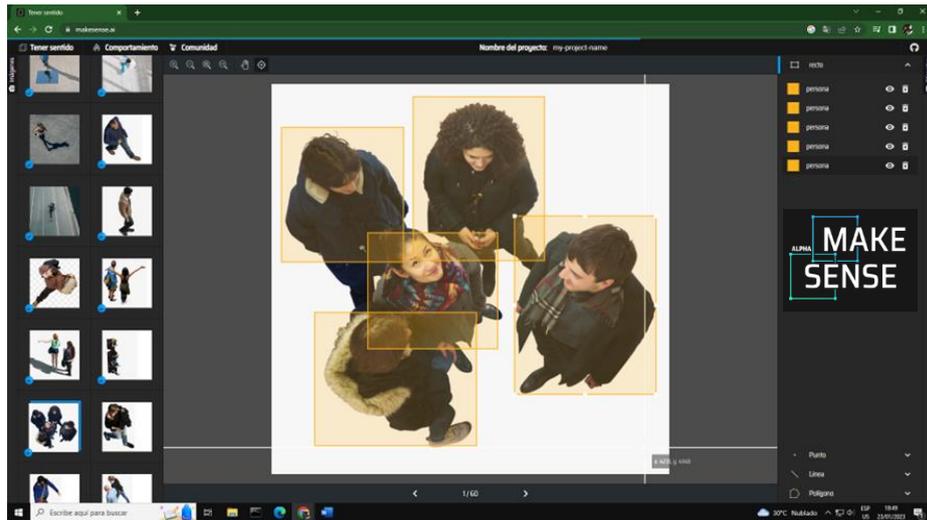


Figura 16: Etiquetado de las imágenes: Elaboración propia.

El método de etiquetado se personalizó con el propósito de seleccionar las personas en cada imagen y trazar las cajas delimitadoras alrededor de ellas. El etiquetar atributos específicos de personas, como la dirección del movimiento, la pose la vestimenta es como se va registrando para poder evaluar la precisión, manejando gran volumen de imágenes y exploran técnicas de aprendizaje activo para optimizar el etiquetado y reducir el esfuerzo humano necesario [70].

3.3.1.4 Generar archivo en formato YOLO

Tras la etiquetación de las imágenes, estas se almacenan en un directorio designado. Asimismo, se crea un archivo YOLO que contiene información pormenorizada sobre cada objeto anotado en las imágenes. Dicho archivo YOLO se emplea durante el entrenamiento de la red neuronal para suministrar los datos de entrada junto con sus respectivas anotaciones.

Se debe generar un conjunto de datos de entrenamiento. Esto implica organizar las imágenes junto con las anotaciones en un formato específico, que incluye la información de posición y tamaño de cada persona en las imágenes.

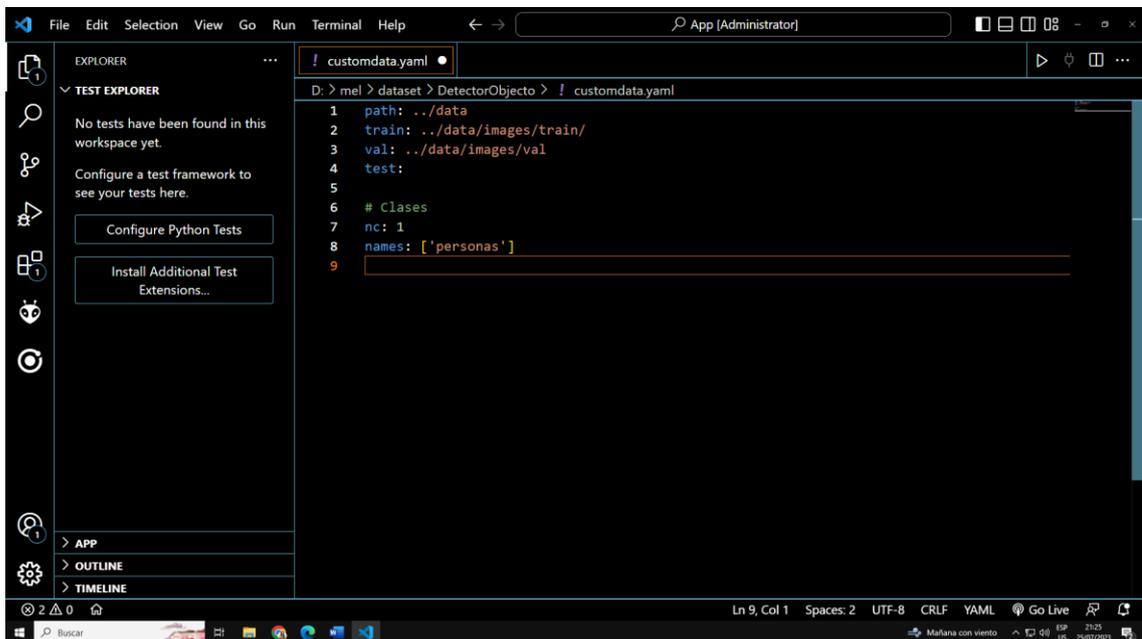
Una vez que las imágenes han sido etiquetadas y almacenadas en una ubicación específica, se genera un archivo en formato YOLO que contiene información detallada sobre cada objeto anotado en las imágenes. Este archivo YOLO, junto con las imágenes etiquetadas, conforma el conjunto de datos de entrenamiento para el proceso de entrenamiento de la red neuronal.

3.3.2.1 Evaluar métrica y set de validación

Los conjuntos de entrenamiento, prueba y validación son fundamentales en la evaluación de modelos de aprendizaje automático. El conjunto de entrenamiento se utiliza para entrenar el modelo, mientras que el conjunto de prueba se emplea para evaluar su rendimiento en datos no vistos. Por otro lado, el conjunto de validación se utiliza para ajustar los hiperparámetros y tomar decisiones durante el desarrollo del modelo, garantizando su eficacia y precisión en la detección de personas [71].

La evaluación exhaustiva del rendimiento de la red neuronal mediante el uso de métricas específicas y un conjunto de datos de validación. Se emplean métricas para medir la calidad de las predicciones realizadas por la red, especialmente en lo que respecta a la identificación precisa de personas. Además, se utiliza un conjunto de datos de validación separado del conjunto de entrenamiento para evaluar la capacidad de generalización de la red en datos no utilizados previamente, lo que permite realizar ajustes en los parámetros y mejorar tanto la precisión como la confiabilidad en la detección de objetos.

El archivo de configuración "customdata.yaml" es creado para establecer parámetros específicos en el proceso de entrenamiento de la red neuronal.



```
1 path: ../data
2 train: ../data/images/train/
3 val: ../data/images/val
4 test:
5
6 # Clases
7 nc: 1
8 names: ['personas']
9
```

Figura 18: Contenido del archivo de configuración customdata.yaml: Elaboración propia.

En la Figura 18 se puede observar el contenido del archivo, donde se muestra el "path" que apunta hacia el directorio "../data" donde se encuentra la carpeta de entrenamiento ("train") y validación ("val") con las imágenes. También se define que el número de clases

(nc) es igual a 1, y las etiquetas son ['personas'], indicando que el modelo será entrenado específicamente para la detección de personas.

En la figura 19, se emplea una instrucción de código que consiste en la ejecución del script de entrenamiento de un modelo de detección de objetos utilizando la biblioteca YOLOv5 en el lenguaje de programación Python.

```
!python train.py --img 640 --batch 4 --epochs 70 --data  
/content/yolov5/data/coco128.yaml --weights personas.pt --cache
```

Figura 19: Script de entrenamiento: Elaboración propia.

Una vez iniciado el proceso de entrenamiento, el sistema procede a entrenar la red neuronal utilizando el conjunto de datos de entrenamiento. Al finalizar el entrenamiento, se muestran dos archivos: uno corresponde al modelo de la última época y el otro al modelo de la mejor época alcanzada durante el entrenamiento. Estos archivos contienen los pesos y parámetros óptimos del modelo que ha sido entrenado figura 20.

```
70 epochs completed in 0.439 hours.  
Optimizer stripped from runs/train/exp2/weights/last.pt, 173.0MB  
Optimizer stripped from runs/train/exp2/weights/best.pt, 173.0MB
```

Figura 20: Archivos del Modelo Entrenado y Modelo Final: Elaboración propia.

Una vez copiada la dirección del mejor entrenamiento "best.pt", se procede a realizar la prueba de inferencia en imágenes específicas utilizando la siguiente línea de código, tal como se muestra en la figura 21.

```
!python detect.py --weights runs/train/exp2/weights/best.pt  
--img 416 --conf 0.1 --source /content/data/images/train/
```

Figura 21: Script Prueba de Inferencia: Elaboración propia.

Durante esta fase del procedimiento, la figura 22 presenta la información resultante de la detección en un conjunto de imágenes de ejemplo. Cada línea de la figura representa una imagen en la que se ha identificado la presencia de una persona. El propósito de esta etapa es evaluar la capacidad del modelo de red neuronal para detectar de manera efectiva el objeto de interés, en este caso, personas, en las imágenes proporcionadas.

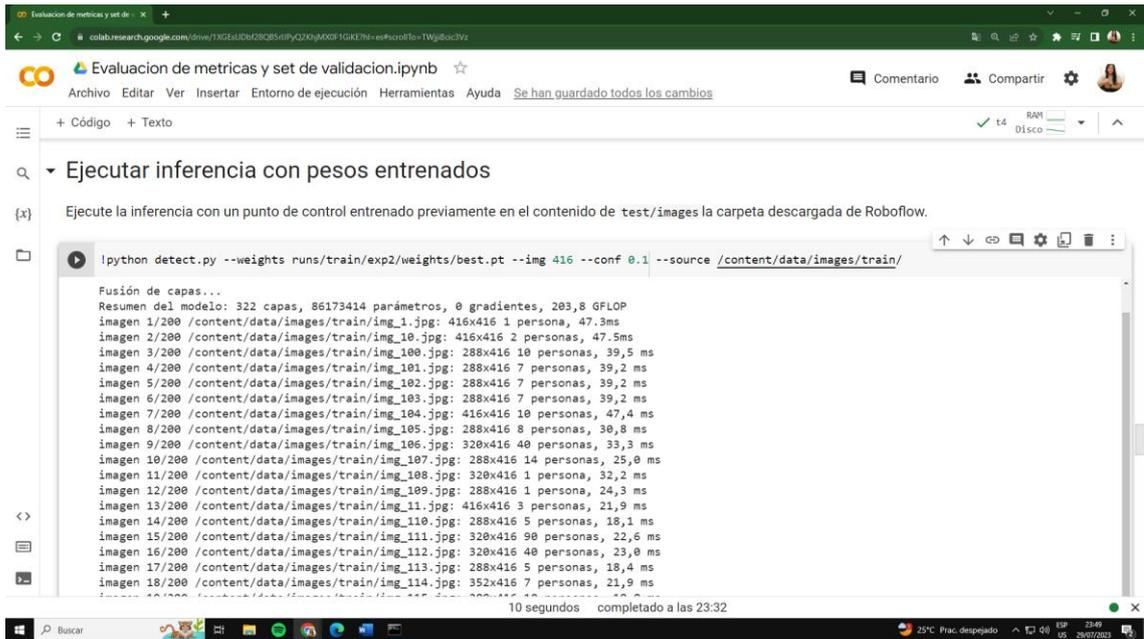


Figura 22: Resultado de la Prueba de Inferencia en Imágenes: Elaboración propia.

En la figura 23 se presenta una prueba realizada en una imagen específica para evaluar el funcionamiento del sistema. El objetivo de esta prueba es determinar si el sistema se ejecuta correctamente o si es necesario mejorar el entrenamiento de la red neuronal para lograr un mejor rendimiento.

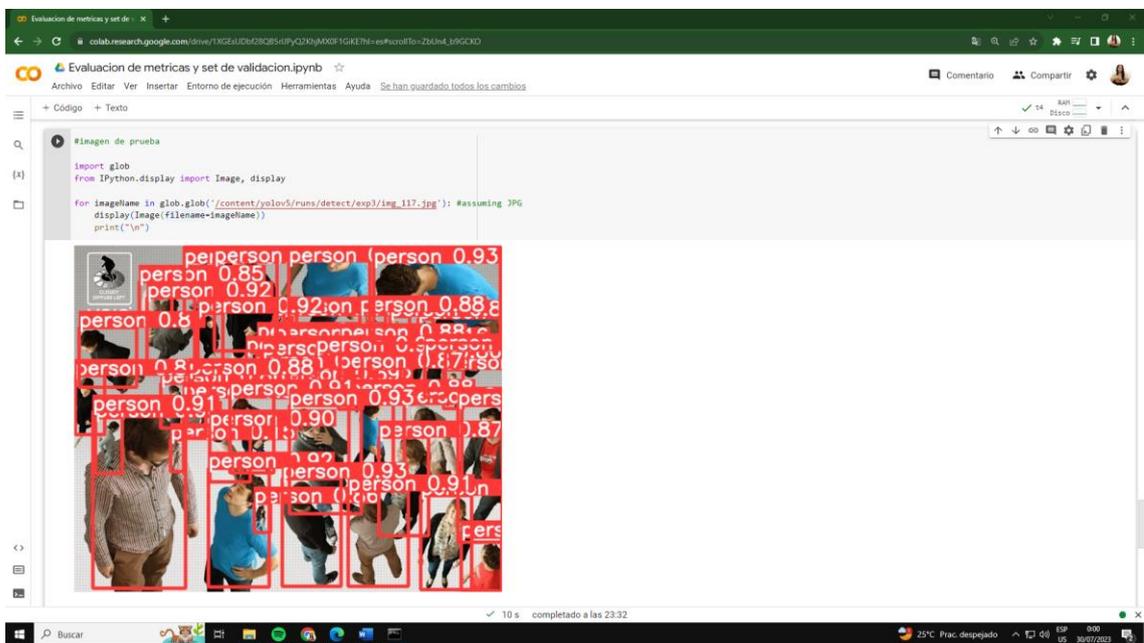


Figura 23: Evaluación del Desempeño del Sistema en una Imagen Específica: Elaboración propia.

La adición de la línea de código mencionada en el proyecto debe realizarse de forma coherente y personalizada según los requisitos específicos. Es crucial asegurarse de tener instalada correctamente la biblioteca YOLOv5 y disponer de los archivos de datos y pesos necesarios antes de ejecutar el proceso. Esto garantizará un entrenamiento y evaluación adecuados del modelo de detección de personas. La correcta preparación y configuración previa son fundamentales para obtener resultados precisos y confiables.

3.3.2.2 Alteraciones para mejorar la red

Las alteraciones realizadas tienen como objetivo mejorar la red y su rendimiento durante el proceso de entrenamiento. A continuación, se describen las modificaciones realizadas y sus implicaciones:

- **Aumento del tamaño de las imágenes de entrada:** Al redimensionar las imágenes a un tamaño más grande, por ejemplo, 640x640 píxeles, se mejora la capacidad del modelo para capturar detalles y características más finas, lo que puede resultar en una mayor precisión en la detección de personas.
- **Ajuste del tamaño del lote (batch size):** Utilizar un tamaño de lote más grande, como 4 imágenes, puede mejorar la eficiencia del entrenamiento al procesar más imágenes en paralelo, lo que acelera el proceso de optimización de los pesos del modelo.
- **Incremento del número de épocas de entrenamiento:** Realizar más épocas de entrenamiento, como 70, permite al modelo aprender de manera más exhaustiva y ajustar sus parámetros para obtener un mejor desempeño en el conjunto de datos de entrenamiento.
- **Utilización de pesos pre-entrenados:** Iniciar el entrenamiento con pesos pre-entrenados, como el archivo `personas.pt`, permite al modelo aprovechar el conocimiento previo y acelerar el proceso de convergencia hacia una solución óptima.
- **Uso de la precisión mixta (Automatic Mixed Precision - AMP):** La precisión mixta combina la utilización de valores de punto flotante de 16 bits y 32 bits para mejorar la eficiencia computacional del entrenamiento, lo que puede acelerar el proceso sin sacrificar la precisión.

- **Configuración del optimizador:** La elección del algoritmo de descenso de gradiente estocástico (SGD) y la tasa de aprendizaje (learning rate) influyen en la velocidad y estabilidad del entrenamiento, permitiendo controlar la convergencia del modelo.
- **Aplicación de transformaciones de aumento de datos (data augmentation):** Las técnicas de aumento de datos, como desenfoque y realce de contraste, aumentan la diversidad de los datos de entrenamiento, ayudando al modelo a generalizar mejor en situaciones nuevas o inéditas.
- **Gestión y restauración de imágenes corruptas:** La identificación y corrección de imágenes corruptas o con problemas de formato garantiza que el conjunto de datos de entrenamiento sea coherente y libre de datos inconsistentes.

Las modificaciones y observaciones mencionadas forman parte del proceso de mejora y ajuste del modelo durante el entrenamiento. La Figura 23 y 24 muestran visualmente los resultados, con el objetivo de lograr un desempeño óptimo y mejorar la calidad de los resultados obtenidos. Estos ajustes son fundamentales para garantizar que el modelo sea capaz de generalizar correctamente en diferentes situaciones y escenarios.

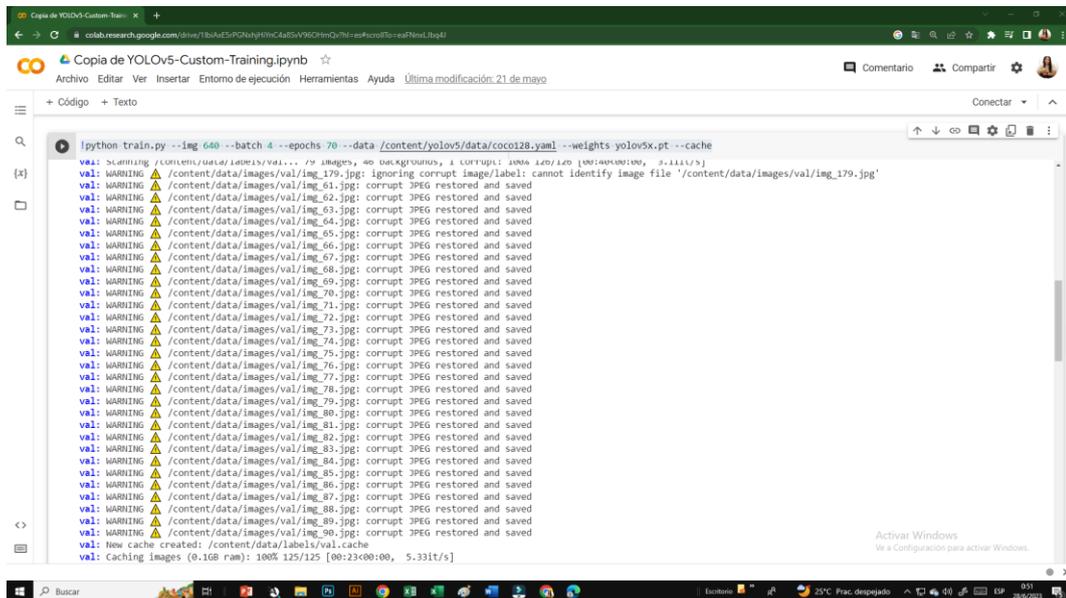
```

python train.py --img 640 --batch 4 --epochs 70 --data /content/yolov5/data/coco128.yaml --weights yolov5x.pt --cache
train: WARNING /content/data/images/train/img_179.jpg: ignoring corrupt image/label: cannot identify image file '/content/data/images/train/img_179.jpg'
train: corrupt JPEG restored and saved
train: WARNING /content/data/images/train/img_61.jpg: corrupt JPEG restored and saved
train: WARNING /content/data/images/train/img_62.jpg: corrupt JPEG restored and saved
train: WARNING /content/data/images/train/img_64.jpg: corrupt JPEG restored and saved
train: WARNING /content/data/images/train/img_65.jpg: corrupt JPEG restored and saved
train: WARNING /content/data/images/train/img_66.jpg: corrupt JPEG restored and saved
train: WARNING /content/data/images/train/img_67.jpg: corrupt JPEG restored and saved
train: WARNING /content/data/images/train/img_68.jpg: corrupt JPEG restored and saved
train: WARNING /content/data/images/train/img_69.jpg: corrupt JPEG restored and saved
train: WARNING /content/data/images/train/img_70.jpg: corrupt JPEG restored and saved
train: WARNING /content/data/images/train/img_71.jpg: corrupt JPEG restored and saved
train: WARNING /content/data/images/train/img_72.jpg: corrupt JPEG restored and saved
train: WARNING /content/data/images/train/img_73.jpg: corrupt JPEG restored and saved
train: WARNING /content/data/images/train/img_74.jpg: corrupt JPEG restored and saved
train: WARNING /content/data/images/train/img_75.jpg: corrupt JPEG restored and saved
train: WARNING /content/data/images/train/img_76.jpg: corrupt JPEG restored and saved
train: WARNING /content/data/images/train/img_77.jpg: corrupt JPEG restored and saved
train: WARNING /content/data/images/train/img_78.jpg: corrupt JPEG restored and saved
train: WARNING /content/data/images/train/img_79.jpg: corrupt JPEG restored and saved
train: WARNING /content/data/images/train/img_80.jpg: corrupt JPEG restored and saved
train: WARNING /content/data/images/train/img_81.jpg: corrupt JPEG restored and saved
train: WARNING /content/data/images/train/img_82.jpg: corrupt JPEG restored and saved
train: WARNING /content/data/images/train/img_83.jpg: corrupt JPEG restored and saved
train: WARNING /content/data/images/train/img_84.jpg: corrupt JPEG restored and saved
train: WARNING /content/data/images/train/img_85.jpg: corrupt JPEG restored and saved
train: WARNING /content/data/images/train/img_86.jpg: corrupt JPEG restored and saved
train: WARNING /content/data/images/train/img_87.jpg: corrupt JPEG restored and saved
train: WARNING /content/data/images/train/img_88.jpg: corrupt JPEG restored and saved
train: WARNING /content/data/images/train/img_89.jpg: corrupt JPEG restored and saved
train: WARNING /content/data/images/train/img_90.jpg: corrupt JPEG restored and saved
train: New cache created: /content/data/labels/train.cache
train: Caching images (0.26B ram): 100% 199/199 [00:22<00:00, 8.78it/s]
  
```

Figura 24: Escaneo de conjunto de entrenamiento (Train): Elaboración propia.

La iteración y mejora continua del modelo son esenciales para lograr un sistema de detección de personas eficiente y efectivo. Al realizar estos ajustes y optimizaciones, se

busca maximizar la capacidad del modelo para realizar conteos precisos en tiempo real y mejorar su utilidad en aplicaciones prácticas.



```
python train.py --img 640 --batch 4 --epochs 70 --data /content/yolov5/data/coco128.yaml --weights yolov5x.pt --cache
val: Scanning /content/data/labels/val... 79 images, 40 corrupted, 1 corrupt: 100% 1.0/1.0 [00:00:00, 5.1111/s]
val: WARNING /content/data/images/val/img_179.jpg: ignoring corrupt image/label: cannot identify image file '/content/data/images/val/img_179.jpg'
val: WARNING /content/data/images/val/img_61.jpg: corrupt JPEG restored and saved
val: WARNING /content/data/images/val/img_62.jpg: corrupt JPEG restored and saved
val: WARNING /content/data/images/val/img_63.jpg: corrupt JPEG restored and saved
val: WARNING /content/data/images/val/img_64.jpg: corrupt JPEG restored and saved
val: WARNING /content/data/images/val/img_65.jpg: corrupt JPEG restored and saved
val: WARNING /content/data/images/val/img_66.jpg: corrupt JPEG restored and saved
val: WARNING /content/data/images/val/img_67.jpg: corrupt JPEG restored and saved
val: WARNING /content/data/images/val/img_68.jpg: corrupt JPEG restored and saved
val: WARNING /content/data/images/val/img_69.jpg: corrupt JPEG restored and saved
val: WARNING /content/data/images/val/img_70.jpg: corrupt JPEG restored and saved
val: WARNING /content/data/images/val/img_71.jpg: corrupt JPEG restored and saved
val: WARNING /content/data/images/val/img_72.jpg: corrupt JPEG restored and saved
val: WARNING /content/data/images/val/img_73.jpg: corrupt JPEG restored and saved
val: WARNING /content/data/images/val/img_74.jpg: corrupt JPEG restored and saved
val: WARNING /content/data/images/val/img_75.jpg: corrupt JPEG restored and saved
val: WARNING /content/data/images/val/img_76.jpg: corrupt JPEG restored and saved
val: WARNING /content/data/images/val/img_77.jpg: corrupt JPEG restored and saved
val: WARNING /content/data/images/val/img_78.jpg: corrupt JPEG restored and saved
val: WARNING /content/data/images/val/img_79.jpg: corrupt JPEG restored and saved
val: WARNING /content/data/images/val/img_80.jpg: corrupt JPEG restored and saved
val: WARNING /content/data/images/val/img_81.jpg: corrupt JPEG restored and saved
val: WARNING /content/data/images/val/img_82.jpg: corrupt JPEG restored and saved
val: WARNING /content/data/images/val/img_83.jpg: corrupt JPEG restored and saved
val: WARNING /content/data/images/val/img_84.jpg: corrupt JPEG restored and saved
val: WARNING /content/data/images/val/img_85.jpg: corrupt JPEG restored and saved
val: WARNING /content/data/images/val/img_86.jpg: corrupt JPEG restored and saved
val: WARNING /content/data/images/val/img_87.jpg: corrupt JPEG restored and saved
val: WARNING /content/data/images/val/img_88.jpg: corrupt JPEG restored and saved
val: WARNING /content/data/images/val/img_89.jpg: corrupt JPEG restored and saved
val: WARNING /content/data/images/val/img_90.jpg: corrupt JPEG restored and saved
val: New cache created: /content/data/labels/val.cache
val: Caching images (0.168 ram): 100% 125/125 [00:23:00:00, 5.33it/s]
```

Figura 25: Escaneo de conjunto de entrenamiento (Val): Elaboración propia.

3.3.2.3 Procesamiento de épocas

El procesamiento de épocas es esencial en el entrenamiento de modelos de aprendizaje automático. En cada época, el modelo ajusta sus pesos y parámetros para mejorar la precisión y capacidad de generalización. Este proceso iterativo permite aprender patrones relevantes en los datos de entrenamiento, mejorando la detección y conteo de personas. La elección adecuada del número de épocas es clave para lograr un modelo óptimo [72]. El entrenamiento del modelo se llevó a cabo a través de 70 épocas, durante las cuales se supervisaron múltiples aspectos del proceso. Se monitoreó la memoria utilizada en la GPU, así como las pérdidas asociadas a las cajas detectadas (`box_loss`), las pérdidas relacionadas con los objetos identificados (`obj_loss`), las pérdidas de las clases asignadas (`cls_loss`) y el número de instancias presentes en cada iteración. Estos registros proporcionaron una visión detallada del rendimiento del modelo y su progreso durante el entrenamiento.

A medida que avanzan las épocas, se recopila información sobre las métricas de precisión (P), recall (R) y mAP (mean Average Precision) tanto a nivel individual para cada clase, como en el conjunto de imágenes. Estos resultados se presentan en forma de tabla, que detalla el número de imágenes y el número de instancias consideradas en cada época.

Estas métricas proporcionan una evaluación detallada del desempeño del modelo a lo largo del entrenamiento y ayudan a identificar posibles mejoras y áreas de enfoque para optimizar el rendimiento del sistema.

Se proporciona un resumen del proceso de entrenamiento, donde se indica que se han completado satisfactoriamente las 70 épocas en un tiempo total de 0.439 horas. Además, se destaca la ubicación de almacenamiento de los pesos del modelo, identificados como "runs/train/exp2/weights/last.pt" y "runs/train/exp2/weights/best.pt". Esta información se muestra en la figura 25, que representa visualmente el proceso de entrenamiento y los resultados obtenidos en el entrenamiento del modelo.

```

Copia de YOLOv5-Custom-Training.ipynb
Copia de YOLOv5-Custom-Training.ipynb
Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Última modificación: 21 de mayo
+ Código + Texto
all 125 646 0.674 0.992 0.788 0.539
Epoch GPU_mem box_loss obj_loss cls_loss Instances Size
65/69 5.436 0.02443 0.03867 0 18 640: 100% 50/50 [00:15:00:00, 3.311t/s]
Class Images Instances P R mAP50 mAP50-95: 100% 16/16 [00:03:00:00, 5.251t/s]
all 125 646 0.675 0.995 0.787 0.535
Epoch GPU_mem box_loss obj_loss cls_loss Instances Size
66/69 5.436 0.02503 0.03965 0 20 640: 100% 50/50 [00:15:00:00, 3.291t/s]
Class Images Instances P R mAP50 mAP50-95: 100% 16/16 [00:03:00:00, 5.231t/s]
all 125 646 0.675 0.995 0.781 0.533
Epoch GPU_mem box_loss obj_loss cls_loss Instances Size
67/69 5.436 0.02436 0.04088 0 35 640: 100% 50/50 [00:15:00:00, 3.311t/s]
Class Images Instances P R mAP50 mAP50-95: 100% 16/16 [00:03:00:00, 5.261t/s]
all 125 646 0.675 0.995 0.698 0.53
Epoch GPU_mem box_loss obj_loss cls_loss Instances Size
68/69 5.436 0.02508 0.04683 0 30 640: 100% 50/50 [00:15:00:00, 3.301t/s]
Class Images Instances P R mAP50 mAP50-95: 100% 16/16 [00:03:00:00, 5.241t/s]
all 125 646 0.675 0.995 0.698 0.529
Epoch GPU_mem box_loss obj_loss cls_loss Instances Size
69/69 5.436 0.02517 0.04106 0 19 640: 100% 50/50 [00:15:00:00, 3.311t/s]
Class Images Instances P R mAP50 mAP50-95: 100% 16/16 [00:03:00:00, 5.281t/s]
all 125 646 0.677 0.995 0.697 0.531

70 epochs completed in 0.439 hours.
Optimizer stripped from runs/train/exp2/weights/last.pt, 173.0MB
Optimizer stripped from runs/train/exp2/weights/best.pt, 173.0MB

Validating runs/train/exp2/weights/best.pt...
Fusing layers...
Model summary: 322 layers, 86173414 parameters, 0 gradients, 203.8 GFLOPs
Class Images Instances P R mAP50 mAP50-95: 100% 16/16 [00:04:00:00, 3.911t/s]
all 125 646 0.674 0.992 0.787 0.539
Results saved to runs/train/exp2
  
```

Figura 26: Avance en cada época y las métricas obtenidas: Elaboración propia.

Se lleva a cabo una evaluación del modelo utilizando los mejores pesos obtenidos, mediante la fusión de capas y el cálculo de métricas de precisión, recall y mAP, tanto para cada clase de objeto como para el conjunto global de imágenes. Este proceso proporciona una visión detallada del entrenamiento y validación del modelo de detección de objetos, presentando información relevante sobre el progreso en cada época y las métricas obtenidas en cada etapa.

3.3.2.4 Análisis de datos resultante de predicción

La sección de análisis de datos resultante de la predicción se ha llevado a cabo con el objetivo de evaluar el desempeño del modelo de detección de objetos. A continuación, se presentan los resultados obtenidos:

- Métrica mAP@0.5: Se obtuvo un valor suavizado de 0.6971 para el conjunto de entrenamiento (Train/exp2) en el paso 69.

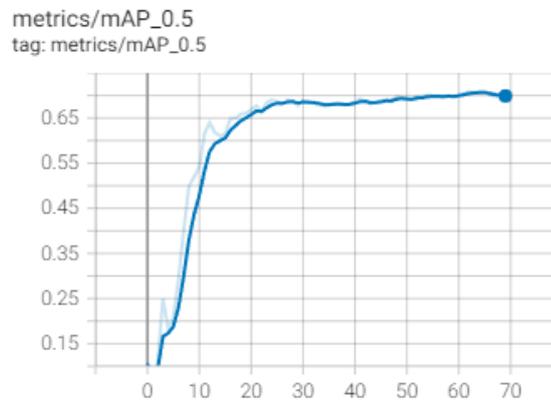


Figura 27: Resultado de métrica mAP_0.5: Elaboración propia.

- Métrica mAP@0.5:0.95: Se obtuvo un valor suavizado de 0.5312 para el conjunto de entrenamiento (Train/exp2) en el paso 69.

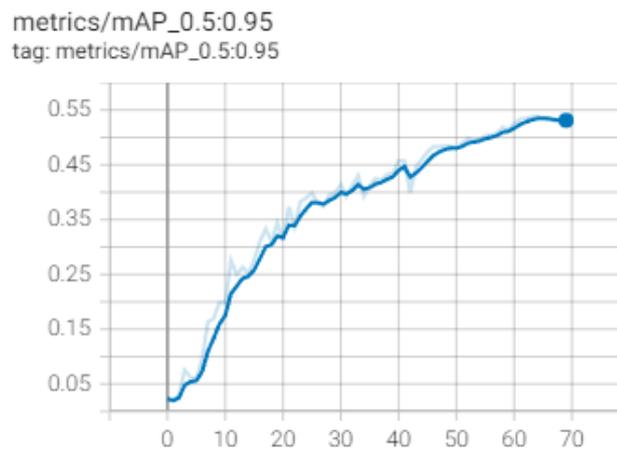


Figura 28: Resultado de métrica mAP_0.5:0.95: Elaboración propia.

- Precisión: Se obtuvo un valor suavizado de 0.6768 para el conjunto de entrenamiento (Train/exp2) en el paso 69.

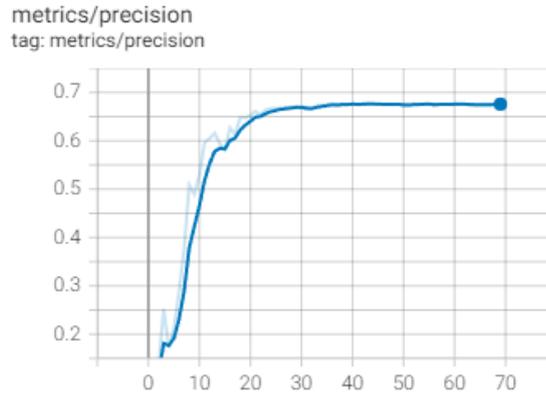


Figura 29: Resultado de métrica / precisión: Elaboración propia.

- Recall: Se obtuvo un valor suavizado de 0.9952 para el conjunto de entrenamiento (Train/exp2) en el paso 69.

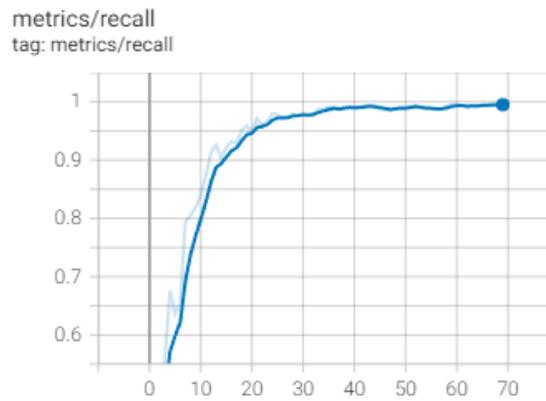


Figura 30: Resultado de métrica / recall: Elaboración propia.

- Pérdida de bounding boxes (box_loss): Se obtuvo un valor suavizado de 0.02517 para el conjunto de entrenamiento (Train/exp2) en el paso 69.

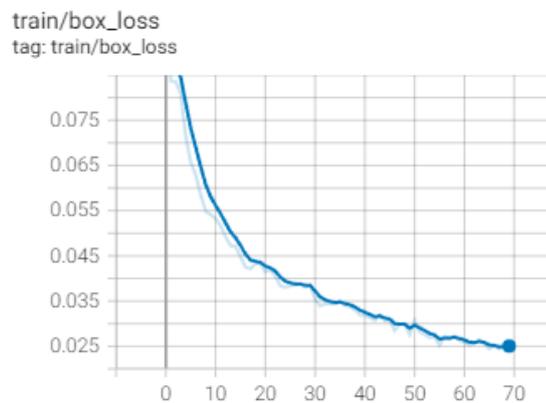


Figura 31: Resultado de pérdida de bounding boxes: Elaboración propia.

- Pérdida de clasificación (cls_loss): Se obtuvo un valor de 0 para el conjunto de entrenamiento (Train/exp2) en el paso 69.

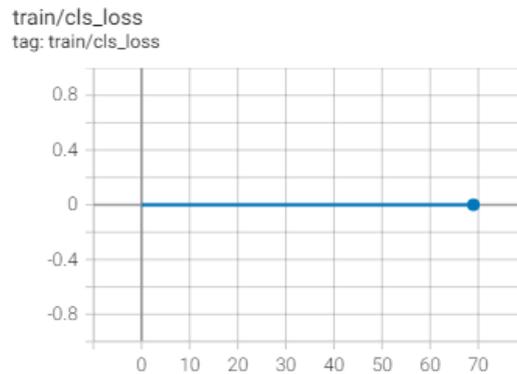


Figura 32: Resultado de pérdida de clasificación: Elaboración propia.

- Pérdida de objetos (obj_loss): Se obtuvo un valor suavizado de 0.04106 para el conjunto de entrenamiento (Train/exp2) en el paso 69.

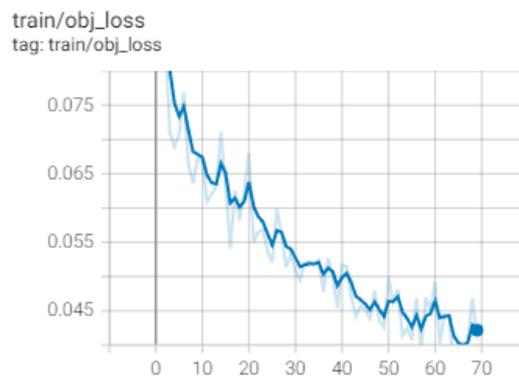


Figura 33: Resultado de Pérdida de objetos: Elaboración propia.

- Pérdida de bounding boxes en la validación (val/box_loss): Se obtuvo un valor suavizado de 0.01813 para el conjunto de entrenamiento (Train/exp2) en el paso 69.

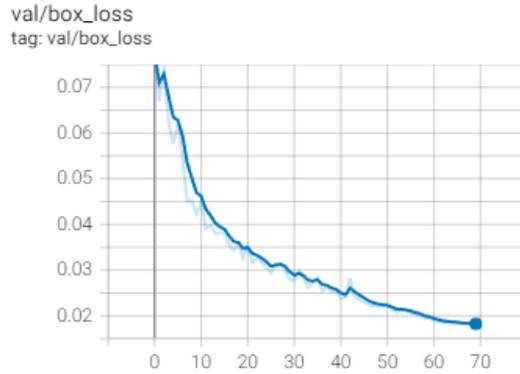


Figura 34: Resultado de pérdida de bounding boxes en la validación: Elaboración propia.

- Pérdida de clasificación en el conjunto de validación (val/cls_loss): Se obtuvo un valor de 0 para el conjunto de entrenamiento (Train/exp2) en el paso 69.

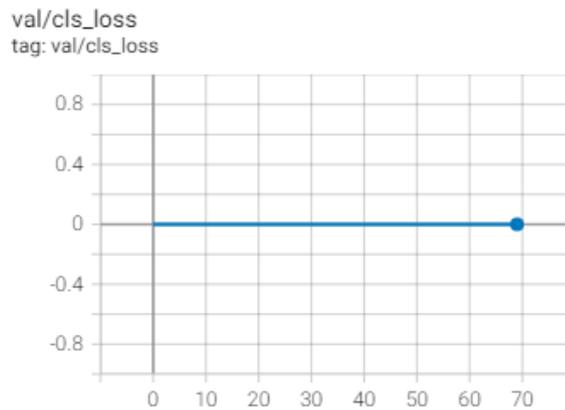


Figura 35: Resultado de pérdida de clasificación en el conjunto de validación: Elaboración propia.

- Pérdida de objetos en el conjunto de validación (val/obj_loss): Se obtuvo un valor suavizado de 0.03865 para el conjunto de entrenamiento (Train/exp2) en el paso 69.

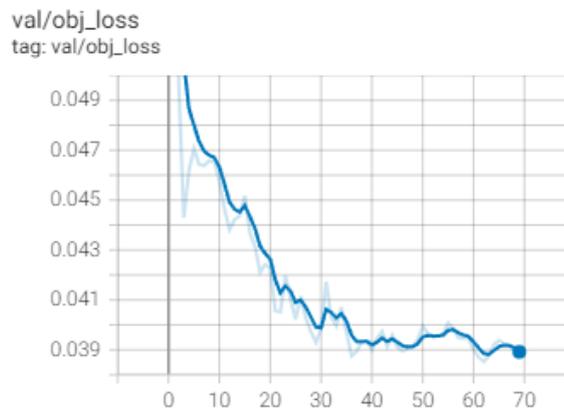


Figura 36: Resultado de pérdida de objetos en el conjunto de validación: Elaboración propia.

Los resultados obtenidos son fundamentales para evaluar el rendimiento del modelo durante las etapas de entrenamiento y validación. Proporcionan información esencial para realizar análisis detallados y realizar ajustes necesarios con el fin de mejorar la precisión y eficacia del modelo de detección de objetos. Estos análisis y ajustes son cruciales para garantizar que el modelo sea capaz de cumplir con los objetivos del proyecto de manera óptima.

Basado en los resultados obtenidos, se puede concluir que el rendimiento del modelo de detección de objetos fue en general positivo. La métrica mAP@0.5 registrada en el conjunto de entrenamiento alcanzó un valor de 0.6971, lo que indica una capacidad considerable para detectar objetos con un umbral de confianza del 70%.

Además, se obtuvieron altos valores de precisión (0.6768) y recall (0.9952), lo que sugiere que el modelo fue capaz de identificar con precisión la mayoría de las instancias de objetos presentes en las imágenes. Estos resultados demuestran el éxito del modelo en la detección y clasificación de objetos, lo cual valida la efectividad de la implementación y el proceso de entrenamiento llevado a cabo.

No obstante, es relevante considerar que existen áreas que requieren mejoras en el modelo de detección de objetos. Por ejemplo, la métrica mAP@0.5:0.95 obtuvo un valor de 0.5312, lo que indica que el modelo podría enfrentar dificultades en la detección de objetos con distintos niveles de confianza.

Asimismo, las pérdidas asociadas a las bounding boxes y objetos en el conjunto de validación mostraron valores susceptibles de mejora. Estos resultados sugieren la necesidad de revisar y ajustar el modelo con el fin de lograr una detección más precisa y confiable, lo que contribuirá a mejorar su rendimiento y eficacia en tareas futuras.

Si bien el desempeño del modelo de detección de objetos resultó prometedor, aún existen oportunidades para mejorarlo y perfeccionar su capacidad de predicción. Mediante la identificación de áreas de mejora y la implementación de ajustes adecuados, se puede aspirar a obtener resultados aún más satisfactorios en futuras iteraciones del modelo.

El modelo de detección demostró ser prometedor, pero aún hay oportunidades para mejorarlo. Identificar áreas de mejora y ajustar adecuadamente el modelo permitirá obtener resultados más satisfactorios en iteraciones. Este enfoque proactivo garantizará

un modelo más preciso y eficiente en diversas aplicaciones de detección y reconocimiento.

3.3.3 Módulo de entrenamiento

El entrenamiento del modelo es esencial en el desarrollo de sistemas de aprendizaje automático, mejorando su capacidad para realizar predicciones precisas y coherentes. Durante este proceso, el modelo ajusta parámetros internos y optimiza su capacidad para reconocer patrones y tomar decisiones basadas en los datos de entrada.

Se emplean técnicas como la retropropagación del error, el descenso del gradiente estocástico y la optimización de funciones de costo. Este módulo desempeña un papel crítico en el éxito del sistema, asegurando un modelo optimizado y capaz de generalizar en datos no vistos previamente.

3.3.3.1 Conjunto de datos

Se resalta la capacidad de generalización de una red neuronal cuando existe una estrecha relación entre la salida deseada y la salida calculada por la red para muestras no utilizadas durante el entrenamiento. Este proceso implica el uso de un conjunto de entrenamiento, que consiste en ejemplos o patrones utilizados para ajustar los pesos de la red mediante un algoritmo de aprendizaje, así como un conjunto de generalización que consta de ejemplos que la red nunca ha encontrado.

Características:

- La red neuronal tiene la capacidad de extraer información relevante de los datos de entrenamiento y aplicar ese conocimiento a situaciones desconocidas.
- La generalización permite a la red adaptarse y responder de manera adecuada a entradas que no ha encontrado previamente.
- La capacidad de generalización es un indicador de la habilidad de la red para capturar patrones y tendencias subyacentes en los datos.

La generalización en el contexto de una red neuronal hace referencia a su capacidad para extraer información de los datos de entrenamiento y aplicar ese conocimiento a situaciones desconocidas. En términos conceptuales, el proceso de aprendizaje de la red neuronal puede interpretarse como un problema de ajuste de una curva, donde la red actúa

como un mapeador no lineal que transforma los datos de entrada en una salida correspondiente.

Es fundamental considerar el riesgo de sobreajuste durante el entrenamiento de la red, ya que esto puede llevar a que la red memorice los datos de entrenamiento y presente dificultades para generalizar a nuevos datos de prueba. En consecuencia, la capacidad de generalización no está siempre garantizada, a pesar de lo afirmado por algunos autores. Es esencial aplicar técnicas de regularización y validación cruzada para mitigar el sobreajuste y asegurar que el modelo pueda adaptarse de manera efectiva a datos no vistos previamente.

3.3.3.2 Entrada de imágenes

Este componente se encarga del proceso de adquisición y preparación de imágenes con el objetivo de que puedan ser procesadas por un modelo o algoritmo posteriormente. En esta etapa, se determina la fuente de las imágenes y se realiza la carga de los archivos de imagen en el sistema. Es esencial asegurarse de que las imágenes sean de alta calidad y estén en el formato adecuado para su correcto procesamiento.

Además, se pueden aplicar diversas técnicas con el objetivo de normalizar las imágenes, ajustar su tamaño y corregir el brillo y contraste, entre otros ajustes necesarios para mejorar la calidad y uniformidad de las imágenes. Es crucial realizar una correcta entrada de imágenes para asegurar resultados precisos y coherentes en las etapas siguientes del proceso de análisis o clasificación de imágenes.

Las técnicas tienen como objetivo mejorar la calidad de las imágenes, realizar la eliminación de ruido o artefactos, y asegurar que todas las imágenes compartan características similares con el fin de facilitar tanto el entrenamiento como la inferencia de un modelo o algoritmo posterior.

3.3.3.3 Pre-procesamiento

En el ámbito de los algoritmos de aprendizaje supervisado, uno de los desafíos más importantes es el manejo de conjuntos de datos en bruto. En la etapa de recolección de datos, que suele ser un proceso no supervisado, es común encontrar conjuntos de datos con información redundante, ruido o características irrelevantes.

Por tanto, el preprocesamiento de datos se convierte en un paso crucial en el proceso de extracción de datos, ya que permite mitigar este tipo de problemas y generar conjuntos de datos mejorados. El preprocesamiento de datos implica la aplicación de técnicas, métodos para limpiar y, transformar los datos, eliminando ruido, reduciendo la dimensionalidad y seleccionando características relevantes.

Al mejorar la calidad de los datos y reducir la complejidad, el preprocesamiento de datos contribuye a mejorar la eficiencia y precisión de los algoritmos de aprendizaje supervisado.

3.3.3.4 Extracción de características

En el proceso de extracción de características, se llevan a cabo distintos procedimientos para cada una de ellas, debido a la naturaleza única que posee. Cada peculiaridad presenta propiedades y elementos únicos que requieren enfoques y algoritmos específicos para su comparación.

Debido a su variabilidad en representaciones, dimensiones, estructuras y distribuciones en los datos, como se muestra en la figura 37, se hace necesario aplicar métodos especializados. Estos enfoques y algoritmos permiten obtener representaciones más precisas y significativas, facilitando un análisis más detallado y una mejor comprensión de los datos.

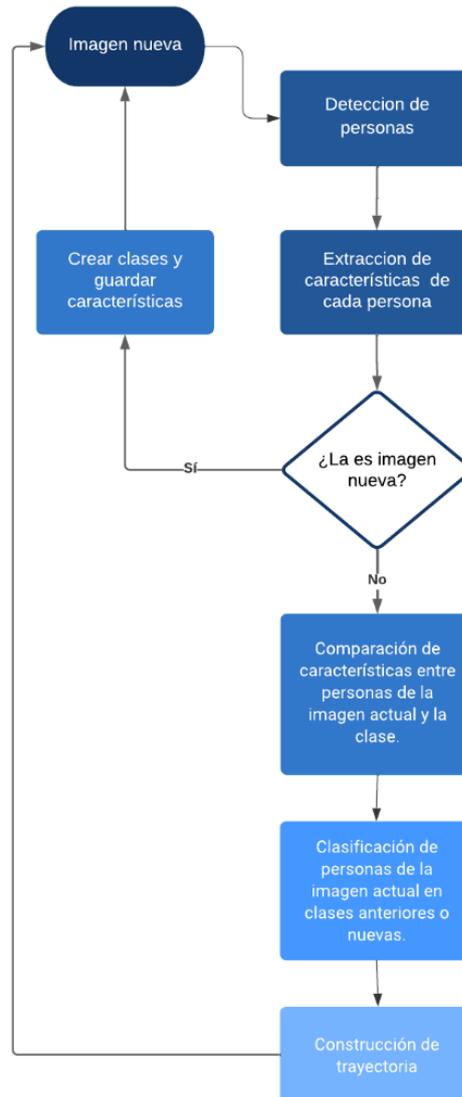


Figura 37: Algoritmo de seguimiento: Elaboración propia.

El script de código presentado en la figura 38, proporcionado implementa un proceso de detección de objetos en una imagen mediante el uso de un modelo de detección previamente entrenado. En concreto, el script utiliza un modelo denominado "personas.pt" para detectar la presencia de personas en la imagen "img_88.jpg".

La imagen de entrada se redimensiona a un tamaño de 640 píxeles para su procesamiento. Además, se establece un umbral de confianza de 0.25, lo que implica que solo se considerarán aquellas detecciones de objetos con una confianza igual o superior a dicho valor. En síntesis, el script permite identificar y localizar personas en la imagen de manera precisa, utilizando un modelo y estableciendo criterios de confianza para la detección.

```
!python detect.py --weights personas.pt --img 640 --conf 0.25 --source "img_88.jpg"

detect: weights=['personas.pt'], source=img_88.jpg, data=data\coco128.yaml, imgsz=[640, 640], conf_thres=0.25, iou_thres=0.45,
max_det=1000, device=, view_img=False, save_txt=False, save_conf=False, save_crop=False, nosave=False, classes=None, agnostic_n
ms=False, augment=False, visualize=False, update=False, project=runs\detect, name=exp, exist_ok=False, line_thickness=3, hide_l
abels=False, hide_conf=False, half=False, dnn=False, vid_stride=1
YOLOv5 v7.0-72-g064365d Python-3.10.6 torch-1.13.1+cpu CPU

Fusing layers...
Model summary: 322 layers, 86173414 parameters, 0 gradients, 203.8 GFLOPs
image 1/1 D:\yolov5\img_88.jpg: 480x640 20 persons, 764.0ms
Speed: 1.0ms pre-process, 764.0ms inference, 1.0ms NMS per image at shape (1, 3, 640, 640)
Results saved to runs\detect\exp46
```

Figura 38: Script de prueba: Elaboración propia.

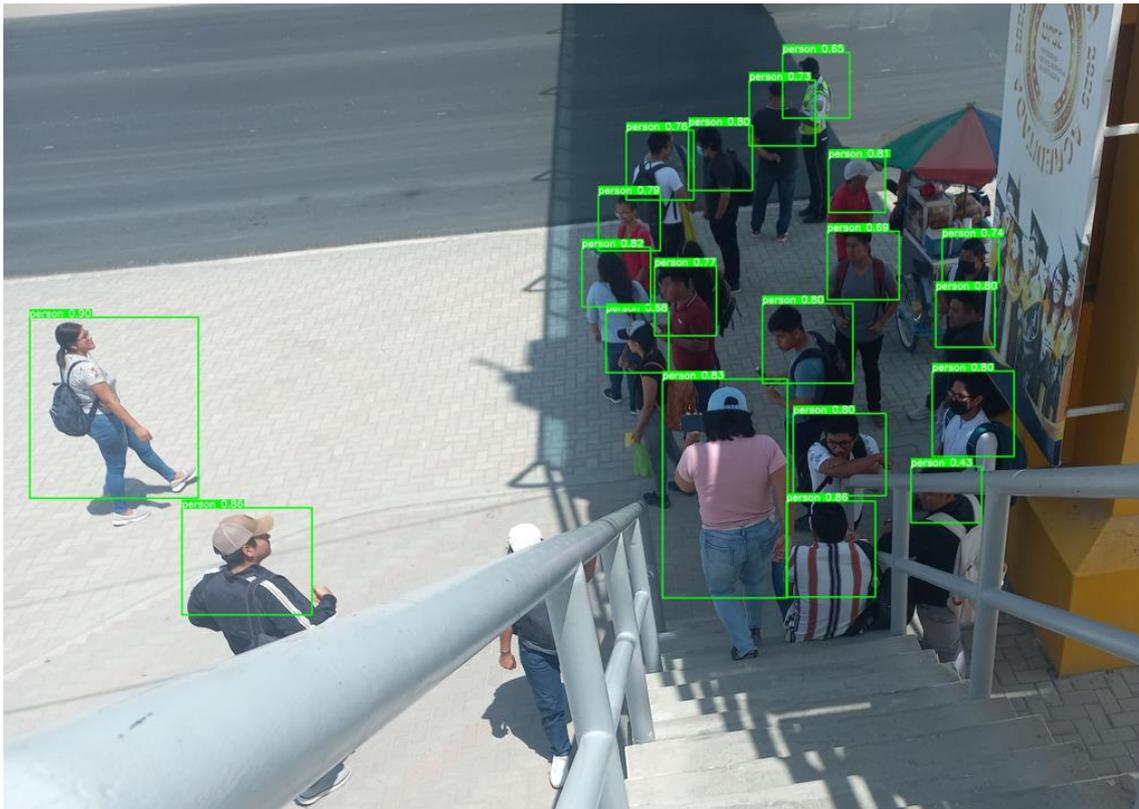


Figura 39: Resultado obtenido imagen de prueba: Elaboración propia.

A través de un bucle, se recorren todas las imágenes encontradas en la ruta especificada y se visualizan en el entorno de trabajo. Esta capacidad resulta útil en tareas de análisis y exploración de conjuntos de imágenes, permitiendo una rápida inspección y examen de los resultados obtenidos. La figura 39 ilustra un ejemplo del proceso de extracción.

3.3.3.5 Entrenar la red

El entrenamiento de una red neuronal como YOLOv5 implica ajustar los parámetros del modelo utilizando un conjunto de datos etiquetados. Durante el proceso de entrenamiento, se optimizan los pesos y bias de la red mediante algoritmos como el Descenso de

Gradiente Estocástico (SGD), minimizando una función de pérdida que compara las predicciones de la red con las etiquetas conocidas.

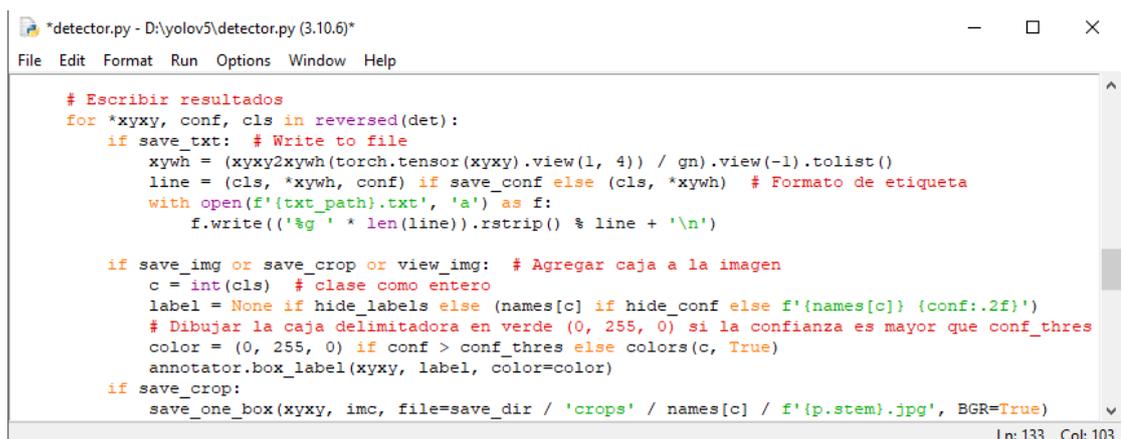
El entrenamiento se realiza en varias épocas, donde se pasa por todo el conjunto de datos varias veces, y se requiere un conjunto de datos de alta calidad y representatividad para obtener resultados precisos.

Una vez completado el entrenamiento, la red neuronal entrenada puede utilizarse para realizar detecciones en nuevas imágenes o videos. La red es capaz de reconocer y localizar objetos de interés, basándose en los patrones aprendidos durante el entrenamiento.

Esto permite identificar y ubicar objetos similares a los que se encontraban en el conjunto de datos de entrenamiento. El entrenamiento de redes neuronales como YOLOv5 es un proceso fundamental para obtener modelos precisos y efectivos en tareas de detección de objetos.

- **Ajuste del umbral de confianza para una detección personalizada**

El código de la figura 40, realiza una comprobación de confianza (umbral) y, en función del resultado, asigna un color específico a la caja delimitadora. Si la confianza de la detección supera el umbral establecido, la caja delimitadora se dibuja en color verde. De esta manera, las detecciones con una alta confianza en la presencia de personas se destacan visualmente, lo que proporciona una comprensión más rápida y clara de los resultados del modelo de detección.



```
*detector.py - D:\yolov5\detector.py (3.10.6)*
File Edit Format Run Options Window Help

# Escribir resultados
for *xyxy, conf, cls in reversed(det):
    if save_txt: # Write to file
        xywh = (xyxy2xywh(torch.tensor(xyxy).view(1, 4)) / gn).view(-1).tolist()
        line = (cls, *xywh, conf) if save_conf else (cls, *xywh) # Formato de etiqueta
        with open(f'{txt_path}.txt', 'a') as f:
            f.write('%g ' * len(line).rstrip() % line + '\n')

    if save_img or save_crop or view_img: # Agregar caja a la imagen
        c = int(cls) # clase como entero
        label = None if hide_labels else (names[c] if hide_conf else f'{names[c]} {conf:.2f}')
        # Dibujar la caja delimitadora en verde (0, 255, 0) si la confianza es mayor que conf_thres
        color = (0, 255, 0) if conf > conf_thres else colors(c, True)
        annotator.box_label(xyxy, label, color=color)

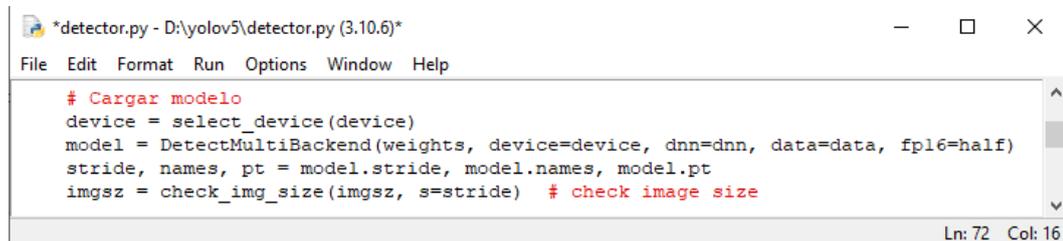
    if save_crop:
        save_one_box(xyxy, imc, file=save_dir / 'crops' / names[c] / f'{p.stem}.jpg', BGR=True)

Ln: 133 Col: 103
```

Figura 40: Comparación de detecciones con umbrales de confianza: Elaboración propia.

- **Detección de personas**

Para la detección de personas se utilizó el modelo Yolov5. En esta sección del código que muestra la figura 41, se carga el modelo utilizando la clase **DetectMultiBackend**:

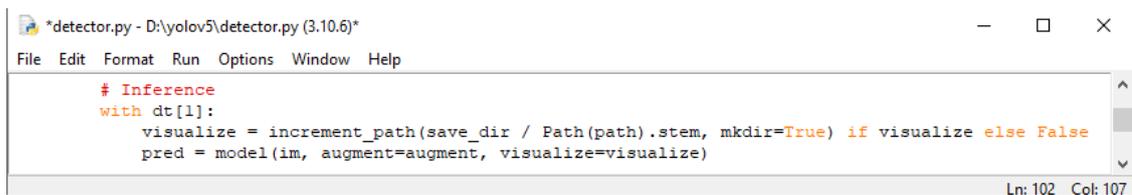


```
*detector.py - D:\yolov5\detector.py (3.10.6)*
File Edit Format Run Options Window Help
# Cargar modelo
device = select_device(device)
model = DetectMultiBackend(weights, device=device, dnn=dnn, data=data, fp16=half)
stride, names, pt = model.stride, model.names, model.pt
imgsz = check_img_size(imgsz, s=stride) # check image size
Ln: 72 Col: 16
```

Figura 41: Carga del modelo con DetectMultiBackend: Elaboración propia.

La clase **DetectMultiBackend** tiene la responsabilidad de cargar y configurar la inferencia. El parámetro "**weights**" especifica la ruta del archivo de pesos del modelo, que por defecto es "**Personas.pt**". El modelo es cargado en el dispositivo de cómputo designado mediante el parámetro "**device**".

Una vez que el modelo ha sido cargado exitosamente, se procede a realizar la inferencia en la imagen de entrada mediante la implementación del siguiente código de la figura 42.

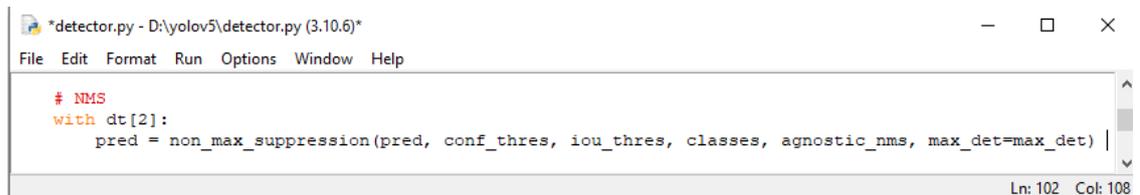


```
*detector.py - D:\yolov5\detector.py (3.10.6)*
File Edit Format Run Options Window Help
# Inference
with dt[1]:
    visualize = increment_path(save_dir / Path(path).stem, mkdir=True) if visualize else False
    pred = model(im, augment=augment, visualize=visualize)
Ln: 102 Col: 107
```

Figura 42: Inferencia en la imagen de entrada utilizando el modelo cargado: Elaboración propia.

La variable **im** representa la imagen de entrada y se pasa al modelo YOLOv5 para obtener las predicciones. La función **model()** realiza la inferencia y devuelve las predicciones en la variable **pred**.

Posteriormente, se aplica la supresión de no máximos (NMS) para filtrar las detecciones redundantes figura 43.



```
*detector.py - D:\yolov5\detector.py (3.10.6)*
File Edit Format Run Options Window Help
# NMS
with dt[2]:
    pred = non_max_suppression(pred, conf_thres, iou_thres, classes, agnostic_nms, max_det=max_det) |
Ln: 102 Col: 108
```

Figura 43: Supresión de No Máximos (NMS): Elaboración propia.

La función `non_max_suppression()` toma las predicciones, junto con los umbrales de confianza (`conf_thres`) y solapamiento IoU (`iou_thres`), y realiza la supresión de no máximos para seleccionar las detecciones más relevantes.



Figura 44: Resultado por pantalla de detección de personas: Elaboración propia.

- **Contabilización de personas**

El siguiente fragmento de código que se muestra en la figura 43, es responsable de mostrar los resultados de la detección de personas.

```
*detector.py - D:\yolov5\detector.py (3.10.6)*
File Edit Format Run Options Window Help
# Transmitir resultados
im0 = annotator.result()
person_count = len(det) # Obtener la cantidad de personas detectadas en el cuadro actual
cv2.putText(im0, f'Personas: {person_count}', (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
Ln: 147 Col: 121
```

Figura 45: Visualización de resultados de detección de personas: Elaboración propia.

Al realizar la detección y el procesamiento de las imágenes, se genera una imagen de salida final donde se agregan las etiquetas de las cajas delimitadoras y las clases detectadas. Además, se calcula el número de personas detectadas en el cuadro actual y se muestra este conteo en la imagen resultante utilizando texto. Este enfoque proporciona una representación visual de la cantidad de personas presentes en la imagen procesada.



Figura 46: Resultado por pantalla de contabilización: Elaboración propia.

3.4 PRUEBA Y RESULTADOS DEL SISTEMA

3.4.1 Evaluación del rendimiento de tiempo de respuesta

Para la evaluación del modelo de detección de personas, se seleccionó un conjunto diverso de imágenes, abarcando distintas situaciones y escenarios, a fin de obtener resultados representativos y significativos.

Cada imagen fue procesada individualmente, y se produjo el tiempo exacto que el sistema tomó para llevar a cabo dicha tarea. El tiempo se expresó en milisegundos (ms) para obtener mediciones precisas y detalladas. Asimismo, se contabilizó la cantidad de personas presentes en cada fotografía como parte del análisis de desempeño.

3.4.1.1 Prueba de imágenes

La prueba de imágenes consistió en evaluar el rendimiento en diversas fotografías. Cada imagen fue identificada por un nombre único y tenía un tamaño específico, variando entre 256x416 y 416x416 píxeles. Se produjo la cantidad de personas presentes en cada

fotografía, y se midió el tiempo que el modelo tardó en detectar las personas en milisegundos (ms).

Los resultados revelaron que el modelo presentado diferentes tiempos de detección y precisión en función del tamaño de la imagen y la cantidad de personas presentes. Permitiendo tener una visión detallada del desempeño del modelo e información valiosa suficiente para mejorar su precisión y eficiencia en aplicaciones futuras.

NOMBRE DE IMAGEN	TAMAÑO DE IMAGEN	CANTIDAD DE PERSONAS	TIEMPO DE DETECCIÓN
Img_100.jpg	288x416	10 personas	39.5ms
Img_162.jpg	320x416	34 personas	22.6ms
Img_102.jpg	288x416	7 personas	39.2ms
Img_105.jpg	288x416	8 personas	30.8 ms
Img_106.jpg	320x416	40 personas	33.3ms
Img_107.jpg	288x416	14 personas	25.0ms
Img_164.jpg	352x416	25 personas	21.8ms
Img_110.jpg	288x416	5 personas	18.1ms
Img_111.jpg	320x416	90 personas	22.6ms
Img_116.jpg	416x416	12 personas	22.0ms
Img_117.jpg	352x416	52 personas	22.4ms
Img_118.jpg	256x416	8 personas	23.1ms
Img_119.jpg	288x416	9 personas	18.5ms
Img_129.jpg	288x416	4 personas	19.9ms

Img_161.jpg	320x416	56 personas	22.8ms
Img_146.jpg	288x416	6 personas	19.9ms
Img_151.jpg	320x416	13 personas	22.8ms
Img_152.jpg	288x416	15 personas	19.5ms
Img_155.jpg	384x416	18 personas	24.2ms
Img_160.jpg	416x416	67 personas	22.4ms

Tabla 13: Resultado de prueba de imágenes: Elaboración propia.

El tiempo promedio es de 23.0 ms, que el sistema tardó en detectar a las personas que se desarrollaron en milisegundos y se representó mediante una distribución gráfica. No obstante, el modelo mantuvo una eficiencia considerable en general, demostrando su capacidad para abordar distintos tamaños de imágenes.

La prueba de imágenes pudo evaluar la eficacia y su capacidad para adaptarse a diversas condiciones fotográficas. La consistencia en el tiempo de detección en imágenes de menor resolución sugiere que el modelo es particularmente adecuado para aplicaciones en escenarios con limitaciones de procesamiento o recursos computacionales.

Es importante destacar que, si bien el tiempo de detección aumentó ligeramente en imágenes de mayor resolución, este incremento fue aceptable y no afectó significativamente la eficacia del modelo.

3.4.1.2 Prueba de video

Se realizó una prueba de referencia para evaluar el rendimiento del modelo en un video de dimensiones 640x384 píxeles. Durante la prueba, se llevaron a cabo detecciones de personas en cada fotograma del video. A continuación, se exponen los resultados obtenidos en la prueba de video:

NOMBRE DE VIDEO	TAMAÑO DE VIDEO	CANTIDAD DE PERSONAS	TIEMPO DE DETECCIÓN
------------------------	------------------------	-----------------------------	----------------------------

video.mp4	640x384	5 personas	668.0ms
	640x384	6 personas	616.0ms
	640x384	4 personas	623.0ms
	640x384	3 personas	669.0ms
	640x384	2 personas	657.0ms
	640x384	1 personas	662.0ms
	640x384	7 personas	681.0ms
	640x384	8 personas	700.0ms
	640x384	11 personas	668.0ms
	640x384	10 personas	742.6ms
	640x384	9 personas	669.0ms
	640x384	8 personas	662.1ms
	640x384	9 personas	685.0ms
	640x384	4 personas	663.0ms
	640x384	5 personas	659.0ms
	640x384	6 personas	668.0ms
	640x384	7 personas	661.0ms
	640x384	12 personas	665.0ms
	640x384	14 personas	726.5ms
	640x384	17 personas	722.0ms

Tabla 14: Resultado prueba de video: Elaboración propia.

En este análisis se produjo la cantidad de personas detectadas en cada cuadro del video, así como el tiempo de detección correspondiente para cada individuo. Cabe mencionar que los tiempos de detección pueden variar en función del número de personas presentes en cada fotograma, siendo el tiempo promedio de detección aproximadamente de 667.51 ms.

3.4.1.3 Prueba en tiempo real

Durante la prueba en tiempo real utilizando una cámara de video, se llevaron a cabo distintas secuencias de imágenes para la detección de personas. A continuación, se presentan los resultados obtenidos durante la prueba en tiempo real:

NOMBRE DE CAMARA	TAMAÑO	CANTIDAD DE PERSONAS	TIEMPO DE DETECCIÓN
0	480x640	58 personas	800.0ms
	480x640	77 personas	855.0ms
	480x640	69 personas	926.0ms
	480x640	3 personas	773.0ms
	480x640	60 personas	757.0ms
	480x640	7 personas	786.0ms
	480x640	85 personas	803.0ms
	480x640	9 personas	757.0ms
	480x640	12 personas	828.0ms
	480x640	10 personas	828.0ms
	480x640	11 personas	781.0ms

	480x640	13 personas	730.0ms
	480x640	6 personas	787.0ms
	480x640	15 personas	812.0ms
	480x640	36 personas	788.0ms
	480x640	59 personas	779.7ms
	480x640	8 personas	777.0ms
	480x640	37 personas	752.0ms
	480x640	65 personas	779.0ms
	480x640	57 personas	777.1ms

Tabla 15: Resultado de prueba en tiempo real: Elaboración propia.

Los resultados obtenidos muestran que la producción entre la complejidad de la escena y el tiempo de detección evidencia la capacidad del sistema para adaptarse a diferentes situaciones. Estos resultados son alentadores y respaldan la aplicabilidad del algoritmo. El tiempo promedio de detección en estas pruebas en tiempo real fue de aproximadamente 784.64 milisegundos.

3.4.2 Evaluación de precisión para distintas cantidades de personas detectadas

El propósito de esta evaluación consiste en analizar la variación de la precisión del sistema ante la presencia de un número específico de personas en el área de detección, con el objetivo de identificar el punto crítico de error en el cual el sistema cumple con el requisito de precisión del 80%. A continuación, se detallan los procedimientos a seguir para llevar a cabo este protocolo:

1. Iniciar la prueba con el área de detección libre de personas.
2. Poner en funcionamiento el sistema de detección.
3. Seleccionar el área específica donde se realizará la detección.
4. Establecer el aforo máximo permitido para la detección.

5. Tomar muestras de la cantidad de personas presentes en el área de detección.
6. Utilizar un software de conteo de datos para verificar la precisión del algoritmo. Se considerará una muestra como correcta si el número de personas detectadas por el sistema coincide con la cantidad real de personas en el área de detección.
7. Calcular el porcentaje de precisión dividiendo la cantidad de muestras correctas entre el total de muestras y multiplicando el resultado por 100.

3.4.2.1 Pruebas de precisión para un número específico de personas

En esta evaluación, se manipula y controla la cantidad de personas que se encuentran presentes dentro del área delimitada por el usuario. Los resultados de precisión del sistema, en relación con la detección del número de personas, se resumen en la Figura 8.

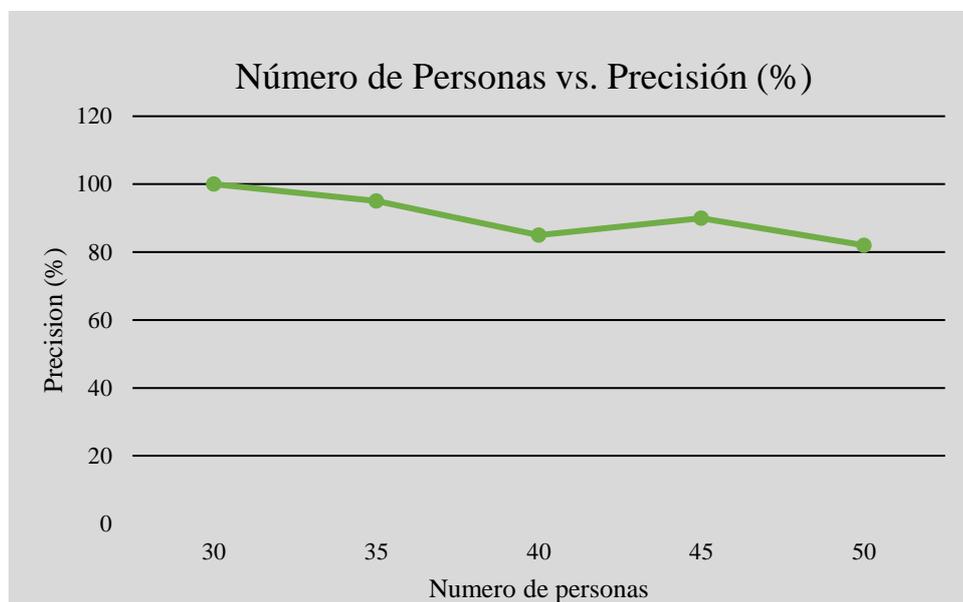


Figura 47: Precisión del Sistema vs. Número de Persona: Elaboración propia.

Presentar los resultados de la visualización de la cámara en el escenario: con persona detectada. La precisión obtenida para ambos casos fue del 100 % en las muestras tomadas, lo que demuestra la fiabilidad del sistema en esta área.

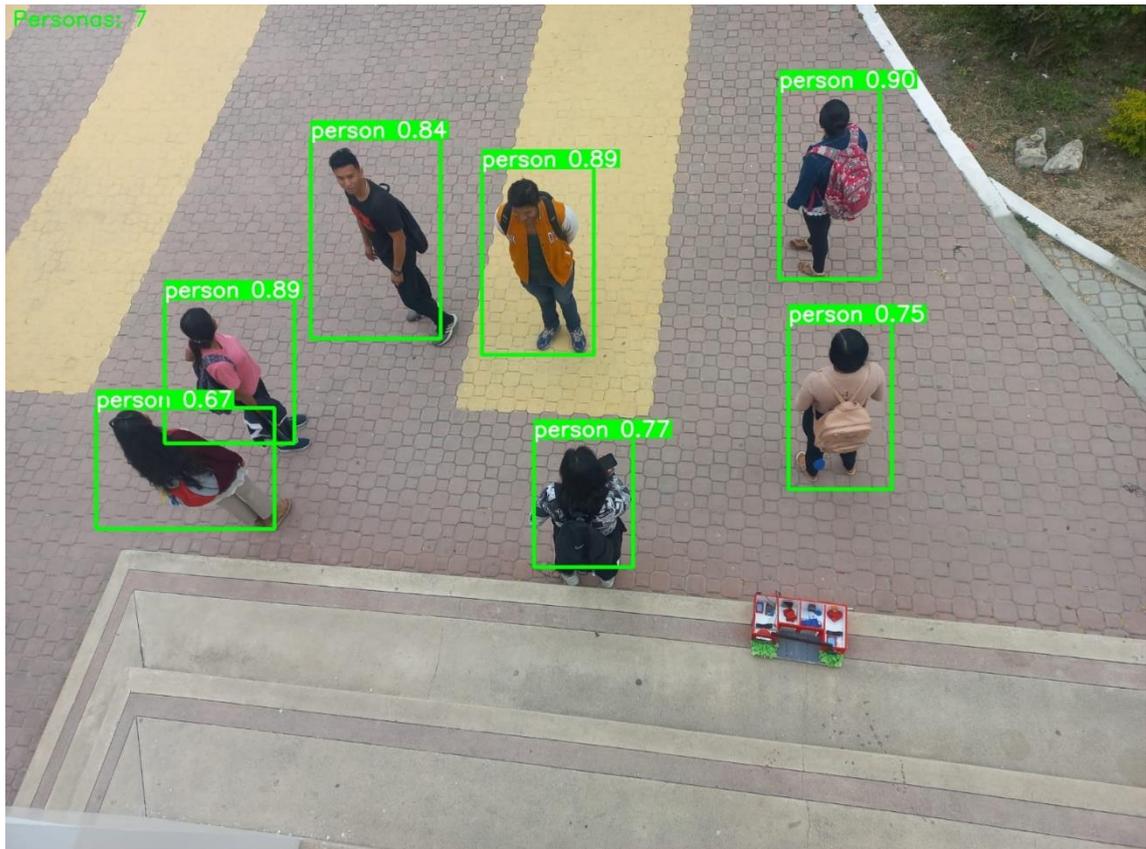


Figura 48: Prueba de precisión con 7 personas: Elaboración propia.

A partir de treinta y cinco personas, la precisión del sistema comienza a disminuir, alcanzando un 98.7 % de eficacia en este caso. Aunque no es absolutamente confiable, se encuentra por encima del requerimiento establecido (recordemos que el requerimiento mínimo es del 80 % para considerar el sistema sea verídico).

Para cuarenta personas, la precisión del sistema se reduce a 90.1 %, lo que indica que los errores de precisión se deben principalmente al solapamiento entre treinta personas desde la perspectiva de la cámara.

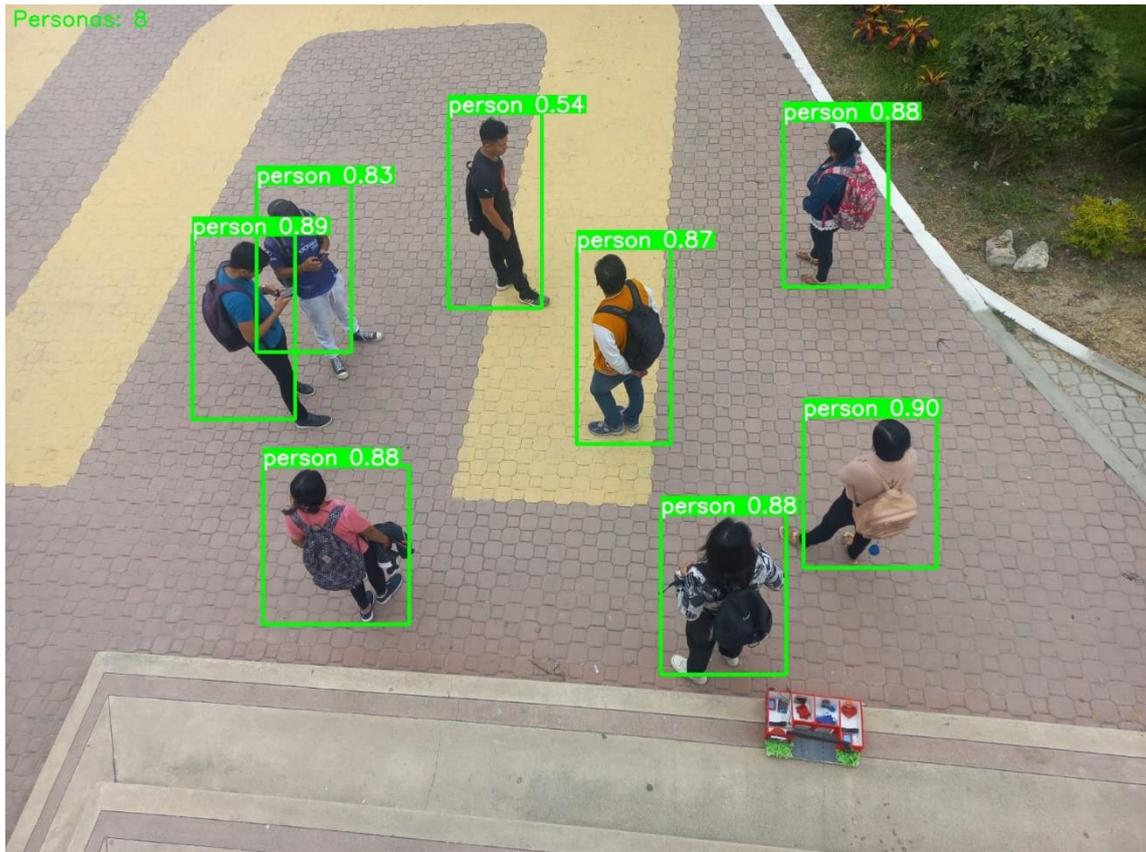


Figura 49: Prueba de precisión con 8 personas: Elaboración propia.

El descenso crítico en la precisión se observa al someter el sistema a la prueba con cincuenta personas en el escenario. En esta situación, se produce con mayor frecuencia el solapamiento entre personas debido a la limitación de espacio, lo que aumenta la probabilidad de que el sistema no detecte a una persona, obteniendo así una precisión del 80.2% para esta prueba.

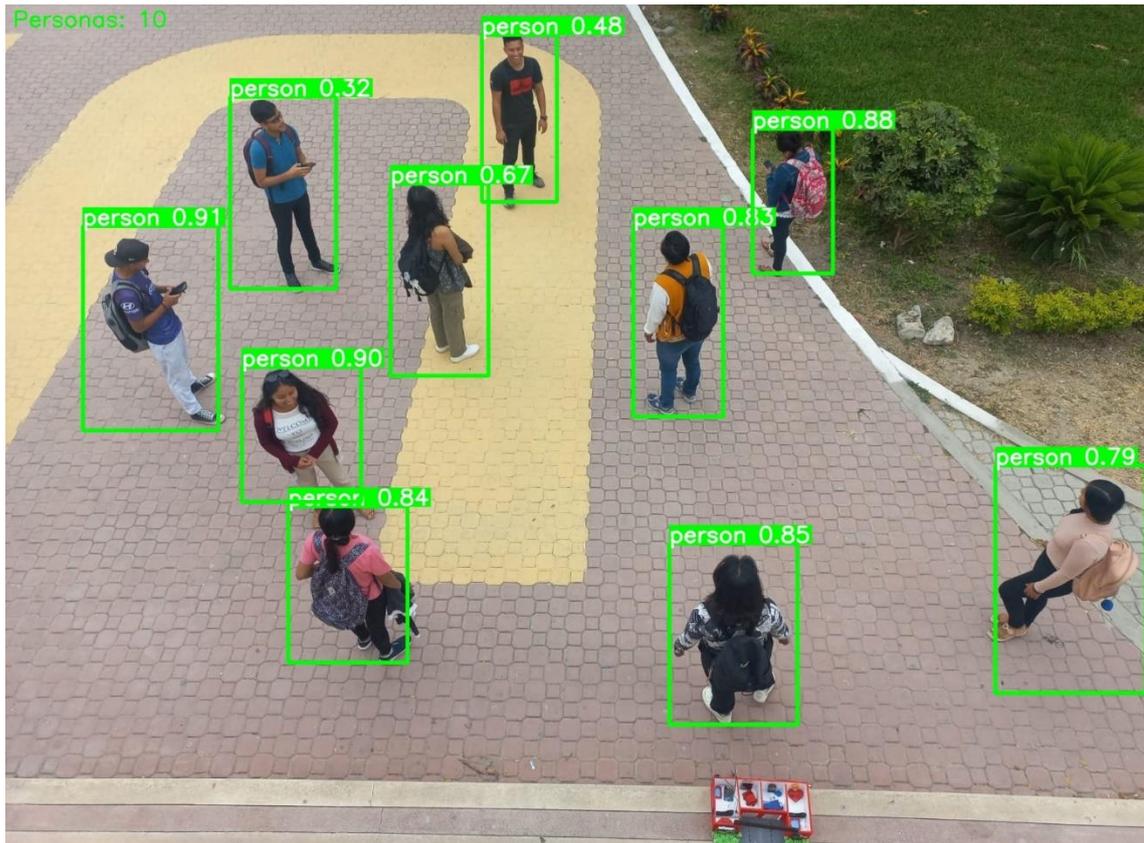


Figura 50: Prueba de precisión con 10 personas: Elaboración propia.

Finalmente, basándonos en los resultados presentados en la Figura 47 y considerando el tamaño del espacio monitorizado, el punto crítico en el cual el sistema cumple con el requerimiento del 80% de precisión se da cuando hay diez personas en el área seleccionada en todos los casos (logrando una precisión del 80.2% para esa cantidad de personas).

CONCLUSIONES

En términos de la evaluación del modelo de detección de personas, llevamos a cabo un análisis exhaustivo para medir su rendimiento mediante pruebas de imágenes y videos con diversas características. El objetivo principal fue evaluar la precisión del sistema al detectar un número específico de personas en el área de interés y determinar el punto crítico en el cual cumple con el requisito mínimo del 80% de precisión.

Para las pruebas de imágenes, se variaron los tamaños y escenarios, midiendo el tiempo de detección en milisegundos (ms) y registrando la cantidad de personas presentes en cada imagen. Los resultados demostraron una eficiencia destacada en imágenes de menor

resolución, con un tiempo promedio de detección de 23.0 ms. En el caso del video, el tiempo promedio fue de 667.51 ms, y en la prueba en tiempo real con una cámara de video, se obtuvo un tiempo promedio de 784.64 ms, evidenciando la capacidad adaptativa del sistema.

En la evaluación de precisión para un número específico de personas, se aplicaron procedimientos controlados para manipular la cantidad de personas en el área de detección. El análisis reveló que el sistema cumplía con el requisito del 80% de precisión cuando había hasta diez personas presentes. Sin embargo, la precisión disminuyó significativamente más allá de esta cantidad. Estos resultados son fundamentales para establecer los límites operativos del sistema y optimizar su desempeño en escenarios con diferentes cantidades de personas.

En conclusión, las pruebas realizadas permitieron obtener información valiosa sobre la eficiencia, adaptabilidad y precisión del modelo de detección de personas. Los datos recopilados son relevantes para mejorar el sistema y respaldar su aplicabilidad en diversos contextos y tamaños de imágenes, contribuyendo al avance de la tecnología en detección de personas.

RECOMENDACIONES

- Se sugiere continuar con la actualización y mejora del modelo de detección de personas basándonos en los resultados obtenidos en la evaluación. Identificar áreas de oportunidad y realizar ajustes en el algoritmo ayudará a aumentar la precisión y eficiencia del sistema en diferentes escenarios.
- Para mejorar la capacidad de generalización del modelo, es recomendable ampliar y diversificar el conjunto de datos utilizado en las pruebas. Incorporar imágenes y videos de distintos entornos y situaciones permitirá un entrenamiento más completo y robusto.
- Se sugiere llevar a cabo pruebas en situaciones más desafiantes, como condiciones climáticas adversas o áreas con alta densidad de personas, para evaluar el rendimiento del sistema en escenarios más realistas y exigentes.
- Continuar investigando y optimizando el algoritmo de detección utilizado en el sistema es esencial. El objetivo es lograr un equilibrio entre precisión y velocidad para que el sistema pueda realizar detecciones precisas en tiempo real.

- Si es posible, incorporar técnicas de aprendizaje profundo en el modelo puede mejorar aún más su capacidad de detección y precisión, dado su éxito en tareas similares.
- Realizar pruebas con diversas cámaras y configuraciones es recomendable para evaluar la compatibilidad del sistema con distintos equipos y condiciones de grabación.
- Se recomienda realizar un monitoreo constante del sistema en producción para detectar posibles errores o deficiencias. Actualizar regularmente el modelo con nuevos datos y mejoras garantizará su rendimiento óptimo a lo largo del tiempo.
- Al implementar sistemas de detección de personas, es crucial considerar las implicaciones éticas y de privacidad. Cumplir con las regulaciones y proteger la privacidad de las personas es fundamental para una implementación responsable.
- Además de las pruebas de precisión, se sugiere realizar pruebas de robustez para evaluar la capacidad del sistema para manejar situaciones inusuales o condiciones adversas, identificando posibles vulnerabilidades y áreas de mejora.
- Es fundamental documentar todas las pruebas realizadas, los resultados obtenidos y las conclusiones alcanzadas. Compartir esta información con la comunidad científica y la industria contribuirá al avance de la investigación en detección de personas y al desarrollo de soluciones más efectivas y confiables.

BIBLIOGRAFÍA

- [1] «MathWorks,» [En línea]. Available: <https://la.mathworks.com/solutions/image-video-processing/object-recognition.html>.
- [2] J. F. Parreño Escobar, «Implementación de un prototipo para contar personas en video capturado en tiempo real utilizando técnicas de aprendizaje profundo,» Universidad de las Fuerzas Armadas ESPE. Carrera de Ingeniería en Electrónica y Telecomunicaciones., Quito, 2020.

- [3] G. Hernández Hernández, «Aplicación de redes neuronales para la identificación de objetos en el tiempo real en imágenes tomadas por quadrotor.» Instituto Politécnico Nacional Centro de Investigación en Computación, México, 2014.
- [4] J. Jokela, «Person counter using real-time object detection and a small neural network,» Information and Communications Technology, Finlandia, 2018.
- [5] Na8, «Detección de Objetos con Python,» 2020. [En línea]. Available: <https://www.aprendemachinelearning.com/deteccion-de-objetos-con-python-yolo-keras-tutorial/>.
- [6] «Jupyter.org,» Jupyter, 2022. [En línea]. Available: <https://jupyter.org/>.
- [7] «Python,» Python Software Foundation, 2001-2022. [En línea]. Available: <https://www.python.org/>.
- [8] J. a. F. A. Redmon, «YOLOv3: An Incremental Improvement,» *arXiv*, 2018.
- [9] N. Aakash, P. Sayak y M. Maynard-Reid, «Keras Simple. Flexible. Poderoso.,» Keras, [En línea]. Available: <https://keras.io/>.
- [10] «{TensorFlow}: Large-Scale Machine Learning on Heterogeneous Systems,» tensorflow2015-whitepaper, 2015. [En línea]. Available: <https://www.tensorflow.org/>.
- [11] «OpenCV,» equipo de OpenCV, 2022. [En línea]. Available: <https://opencv.org/>.
- [12] «NumPy,» NumPy. Reservados todos los derechos., 2022. [En línea]. Available: <https://numpy.org/>.
- [13] «Matplotlib: visualización con Python,» El equipo de desarrollo de Matplotlib, 2012 – 2022. [En línea]. Available: <https://matplotlib.org/>.

- [14] «Pandas,» Pandas vía NumFOCUS, Inc. Hospedado por OVHcloud., 2022. [En línea]. Available: <https://pandas.pydata.org/>.
- [15] «Upse,» El Consejo Superior Universitario, 22 07 1998. [En línea]. Available: https://www.upse.edu.ec/secretariageneral/images/archivospdfsecretaria/4.REGLAMENTOS/1.%20NORMATIVAS%20ACAD%C3%89MICAS/REGLAMENTO_2019/RCS-SE-16-03-2019_REGLAMENTO_DEL_CENTRO_DE_INVESTIGACION_DE_SISTEMA_Y_TELECOMUNICACION.pdf.
- [16] R. Primicias, «Choneros, Lagartos y otras bandas trasladan sus crímenes a Santa Elena,» PRIMICIAS, 08 agosto 2022. [En línea]. Available: <https://www.primicias.ec/noticias/lo-ultimo/choneros-lagartos-tiguerones-bandas-trasladan-crimenes-santa-elena-ecuador/>.
- [17] eluniversocom, «Así están las cifras de la inseguridad en el primer semestre del 2022 comparadas con los tres años previos,» EL UNIVERSO, 5 agosto 2022. [En línea]. Available: <https://www.eluniverso.com/noticias/seguridad/asi-estan-las-cifras-de-la-inseguridad-en-el-primer-semester-del-2022-comparadas-con-los-tres-anos-previos-nota/>.
- [18] «Plan de Creacion de Oportunidades,» Secretaria Nacional de Planificacion, 2021-2025.
- [19] R. Hernández Sampieri, C. Fernández Collado y M. d. P. Baptista Lucio, Metodología de la Informacion, Mexico: McGraw Hill, 2014.
- [20] J. M. Molina López y J. García Herrero, TÉCNICAS DE ANÁLISIS DE DATOS, 2006.
- [21] R. Karim y P. Pujari, Practical Convolutional Neural Networks: implemente modelos avanzados de aprendizaje profundo utilizando Python, Packt Publishing, 2018.

- [22] I. I. n. d. e. y. censo, «Fascículo Provincial Santa Elena,» Santa Elena, 2010.
- [23] G. M. d. Salinas, «Alcaldía de Salinas,» Jefatura de RRPP, 2021. [En línea]. Available: <https://www.salinas.gob.ec/>.
- [24] C. D. P. C. Y. C. SOCIAL, «GUÍA ESPECIALIZADA DE RENDICIÓN DE CUENTAS PARA LOS GOBIERNOS AUTÓNOMOS DESCENTRALIZADOS,» CONSEJO DE PARTICIPACIÓN CIUDADANA Y CONTROL SOCIAL.
- [25] G. d. españa, «Que es la inteligencia artificial,» Plan de recuperacion, 19 04 2023. [En línea]. Available: <https://planderecuperacion.gob.es/noticias/que-es-inteligencia-artificial-ia-prtr>.
- [26] O. Garcia, «Redes Neuronales artificiales: Qué son y cómo se entrenan,» Xeridia, 16 septiembre 2019. [En línea]. Available: <https://www.xeridia.com/blog/redes-neuronales-artificiales-que-son-y-como-se-entrenan-parte-i>.
- [27] A. Innovation, «Qué son las redes neuronales y sus funciones,» ATRIA Innovation, 22 Octubre 2019. [En línea]. Available: <https://www.atriainnovation.com/que-son-las-redes-neuronales-y-sus-funciones/>.
- [28] «Redes neuronales convolucionales (CNN), aprendizaje profundo y visión informática,» intel, 2023. [En línea]. Available: <https://www.intel.es/content/www/es/es/internet-of-things/computer-vision/convolutional-neural-networks.html>.
- [29] Datademia, «¿Qué es Deep Learning y qué es una red neuronal?,» Datademia, 8 Junio 2022. [En línea]. Available: <https://datademia.es/blog/que-es-deep-learning-y-que-es-una-red-neuronal>.
- [30] Aprendizaje automático., Colombia: Ediciones de la U, 2022.

- [31] . A. L. Sotelo, Deep Learning: Teoría y aplicaciones, Colombia: Alpha Editorial, 2021.
- [32] Y. Wang y J. Ren, «Descripción general de los algoritmos de detección de objetos mediante redes neuronales convolucionales,» *Revista de Informática y Comunicaciones*, 2022.
- [33] Darkcrist, «OpenCV una biblioteca para el reconocimiento de objetos en imágenes y cámaras,» Desde Linux, 7 Enero 2020. [En línea]. Available: <https://blog.desdelinux.net/opencv-una-biblioteca-para-el-reconocimiento-de-objetos-en-imagenes-y-camaras/>.
- [34] D. Ameijeiras , H. González y Y. Hernández , «Revisión de algoritmos de detección y seguimiento de objetos con redes profundas para videovigilancia inteligente,» *Revista Cubana de Ciencias Informáticas*, 2020.
- [35] Sandra, «Deep Learning aplicado a la Visión Artificial,» ATRIA Innovation, 22 Febrero 2022. [En línea]. Available: <https://www.atriainnovation.com/deep-learning-aplicado-a-la-vision-artificial/>.
- [36] A. GUTIERREZ BONILLA y J. CARDONA ARISTIZABAL, «SISTEMA DE DETECCIÓN DE AGLOMERACIÓN Y CONTEO DE PERSONAS BASADO EN PROCESAMIENTO DE IMÁGENES,» UNIVERSIDAD EL BOSQUE PROGRAMA DE INGENIERÍA DE SISTEMAS, Bogotá, 2022.
- [37] A. F. Luna Camacho y N. E. Rodríguez Menjura, «Detección Y Conteo De Personas En Espacios Cerrados Utilizando Estrategias Basadas En Visión Artificial,» Pontificia Universidad Javeriana, Bogota.
- [38] «Que es la vision artificial,» Intel, 2023. [En línea]. Available: <https://www.intel.la/content/www/xl/es/internet-of-things/computer-vision/overview.html>.

- [39] J. Barros, «Fotografía: tamaño de imagen, resolución y tamaño físico,» 2022. [En línea]. Available: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://perio.unlp.edu.ar/catedras/iddi/wp-content/uploads/sites/125/2022/04/Fotografia.-Tamano-de-imagen-resolucion-y-tamano-fisico.pdf.
- [40] R. Marin, «¿Qué es OpenCV?,» INESEM, 12 febrero 2020. [En línea]. Available: <https://www.inesem.es/revistadigital/informatica-y-tics/opencv/>.
- [41] Perez, «Fundamentos básicos del procesamiento de imágenes,» famaf, 2018. [En línea]. Available: <https://www.famaf.unc.edu.ar/~pperez1/manuales/cim/cap2.html>.
- [42] «Seguimiento,» 28 septiembre 2018. [En línea]. Available: <http://what-when-how.com/introduction-to-video-and-image-processing/tracking-introduction-to-video-and-image-processing-part-1/>.
- [43] S. Raschka, *A Gentle Introduction to Neural Networks and Deep Learning*, Towards Data Science, 2020.
- [44] «¿Qué es Bounding Box 3D?,» Campamentos de entrenamiento KeepCoding, 4 Octubre 2022. [En línea]. Available: https://keepcoding.io/blog/que-es-bounding-box-3d/#Que_es_Bounding_Box_3D.
- [45] I. Candan Bengi , «YOLOv5: revolucionando la detección de objetos en tiempo real,» HashDork, 2023. [En línea]. Available: <https://hashdork.com/es/yolov5/>.
- [46] J. Nelson y J. Solawetz, «Respondiendo a la controversia sobre YOLOv5,» roboflow, 12 Junio 2020. [En línea]. Available: <https://blog.roboflow.com/yolov4-versus-yolov5/>.
- [47] I. Martínez, «Entrene su detector de objetos personalizado con YOLO,» morioh, 2022. [En línea]. Available: <https://morioh.com/p/f04ccdbc3381>.

- [48] S. Rath, «YOLOv5 – Entrenamiento personalizado de detección de objetos,» learnopencv, 19 Abril 2022. [En línea]. Available: <https://learnopencv.com/custom-object-detection-training-using-yolov5/>.
- [49] C. Villalba, A. Moraleda y M. Rubio, Lenguajes de programación, España: UNED, 2021.
- [50] «Los mejores entornos de desarrollo de Python para utilizar en el 2023,» Rootstack, 20 Octubre 2022. [En línea]. Available: <https://rootstack.com/es/blog/los-mejores-entornos-de-desarrollo-de-python-para-utilizar-en-el-2023>.
- [51] B. Bhagat, «¿Qué es Google Colaboratory 2023?,» ultragamerzpro, 1 enero 2023. [En línea]. Available: <https://ultragamerzpro.com/what-is-google-colaboratory-2023/>.
- [52] CEUPE, «¿Conoces Jupyter Notebook?,» Centro Europeo de Postgrado Mexico, 2023. [En línea]. Available: <https://www.ceupe.mx/blog/conoces-jupyter-notebook.html>.
- [53] Bottega, «Qué es una librería en programación,» Devcamp, 2020. [En línea]. Available: <https://devcamp.es/que-es-libreria-programacion/>.
- [54] «Keras,» Keras.io, 2023. [En línea]. Available: <https://keras.io/>.
- [55] D. d. TensorFlow, «TensorFlow,» Zenodo, 13 mayo 2022. [En línea]. Available: <https://zenodo.org/record/4758419#.ZGK8snbMKUk>.
- [56] M. Salinas, «Publicaciones etiquetadas: YoloV5 Inteligencia artificial para la ayuda a personas con discapacidad visual,» Saturday.ai, 9 marzo 2022. [En línea]. Available: <https://saturdays.ai/tag/yolov5/#:~:text=Este%20es%20un%20sistema%20de,para%20detectar%20objetos%20en%20im%C3%A1genes..>

- [57] «PyTorch,» La Fundacion PyTorch, 2022. [En línea]. Available: <https://pytorch.org/>.
- [58] «Make Sense,» [En línea]. Available: <https://www.makesense.ai/>.
- [59] «Tutorial de Tensorboard: Visualización de gráficos con ejemplo,» guru99, 14 Mayo 2023. [En línea]. Available: <https://guru99.es/tensorboard-tutorial/>.
- [60] L. Gonzalez, «Librería NumPy,» aprendeia, 20 Mayo 2023. [En línea]. Available: <https://aprendeia.com/libreria-de-python-numpy-machine-learning/>.
- [61] S. G. Romero Mier, «Inteligencia artificial como herramienta de estrategia y seguridad para defensa de los Estados,» *Revista de la Escuela Superior de Guerra Naval*, 2019.
- [62] J. Santa-Maria, C. Ríos y C. Garcia, «Reconocimiento de patrones de imágenes a través de un sistema de visión artificial,» *Revista Científica De Sistemas E Informática*, nº 15-26, 2021.
- [63] I. K. Asharul y A.-H. Salim , «Machine Learning in Computer Vision,» *Procedia Computer Science*, vol. 167, 2020.
- [64] «Diagrama de bloques,» Smartdraw.com, [En línea]. Available: <https://www.smartdraw.com/block-diagram/>. [Último acceso: 12 junio 2023].
- [65] «La automatización en el análisis de Imágenes para la caracterización por forma y tamaño de partículas.,» Instrumental CAS, 4 mayo 2020. [En línea]. Available: <https://www.cas-instrumental.com.ar/la-automatizacion-en-el-analisis-de-imagenes-para-la-caracterizacion-por-forma-y-tamano-de-particulas/>.
- [66] «Guia para tratamiento de datos personales en administracion publica,» *Acuerdo ministerial*, 2019.

- [67] O. Reyes, M. Mejia y J. Useche, «Técnicas de inteligencia artificial utilizadas en el procesamiento de imágenes,» *Revista EIA*, 19 Diciembre 2019.
- [68] «Iluminación en Fotografía: Tipos y Teoría de la Luz.,» Toomanyflash, 12 enero 2021. [En línea]. Available: <https://toomanyflash.com/iluminacion-en-fotografia/>.
- [69] L. R. Barba Guaman, «Uso de tecnicas deep learning para reconocimiento de objetos en areas rurales,» Universidad Politecnica de madrid, Madrid, 2021.
- [70] «¿Qué es la anotación de datos 2023? (Mejores herramientas, tipos, desafíos, tendencias),» Shaip, 5 mayo 2023. [En línea]. Available: <https://es.shaip.com/blog/the-a-to-z-of-data-annotation/>.
- [71] «Sets de Entrenamiento, Test y Validación,» Aprende aprendizaje automático., 3 marzo 2020. [En línea]. Available: <https://www.aprendemachinelearning.com/sets-de-entrenamiento-test-validacion-cruzada/>.
- [72] «Epoch Machine Learning: La Guía Más Completa,» CiberseguridadMAX., 11 Marzo 2023. [En línea]. Available: https://ciberseguridadmax.com/epoch/#Que_es_epoch_en_Machine_learning.