
Samsung Innovation Campus (2024)

Curso - Inteligencia artificial

Red Neuronal Convolucional para la detección y localización de humanos en entornos rurales y accidentados

Docentes:

Mtro. Eduardo De Avila Armenta

Dr. Alberto Luque Chang

Integrantes:

José Emmanuel De La Cerdá Lara

Donovan Said Santoyo Meza

Hugo Cano Ruiz

4 de abril de 2025

Índice

Resumen	2
Introducción	2
Antecedentes	3
Objetivos	3
Objetivo general	3
Objetivos específicos	3
Metodología	4
Elección del framework	4
Implementación de YOLO	6
Búsqueda y elección de bases de datos	9
Unificación de datasets	11
Desarrollo	11
Ajuste de parámetros	12
Entrenamiento del algoritmo	12
Resultados	13
Discusión	15
Conclusiones y trabajos futuros	16
Referencias	17

Resumen

En desastres naturales, la localización rápida de personas en riesgo resulta crítica para la supervivencia. Este trabajo busca reducir ese tiempo mediante la implementación de un modelo automatizado de detección de personas usando imágenes aéreas captadas por vehículos aéreos no tripulados (drones). Se utilizan los dataset de "C2A", el cual ofrece imágenes de desastres tomadas con UAV, con personas insertadas digitalmente para simular escenarios. Como complemento, se utilizó el dataset "VisDrone", con fotos aéreas tomadas mediante UAVs en situaciones reales y a diferentes escalas y posiciones. Se emplea el modelo YOLOv8s de Ultralytics, conocido por su eficiencia en tareas de detección de objetos. El proyecto utiliza una Red Neuronal Convolutacional (CNN), ideales para el procesamiento de imágenes, y por ende, para el desarrollo del proyecto, obteniendo resultados favorables aunque con ciertas limitaciones.

Palabras clave: Detección de personas, YOLOv8, Drone, C2A, VisDrone, búsqueda y rescate, siniestros humanos, desastres naturales.

Introducción

Los desastres naturales y siniestros provocados por humanos han ido en aumento por muchos factores, de los cuales destacan el calentamiento global, el cambio climático, la densidad de población, entre otros. Todos estos cosechan año tras año una gran cantidad de bajas civiles a lo largo del mundo, tan solo en América Latina estos han afectado a más de 20 países, de los cuales en total suman alrededor de 800 fallecimientos en 2024, informa la ONU, que junto a ello, añade una advertencia ya que esta región tiende a ser vulnerable ante dichos siniestros. En todos estos acontecimientos, la búsqueda y rescate nunca ha sido una tarea sencilla en ningún contexto, ya sea por la difícil accesibilidad de los integrantes de los cuerpos de Protección Civil en terrenos tan accidentados o de alto riesgo, así como la falta de información para lograr identificar a la población afectada y en estado de peligro. En casos de emergencia, la velocidad de localización es crítica para la supervivencia humana, lo cual exige métodos eficientes de detección y localización. Con este proyecto se busca apoyar la cobertura de la necesidad de recabar información vital sobre la cantidad de personas en estado de riesgo para los cuerpos de emergencia, que también, pueden pasar por alto a personas debido a fatiga, percepción, entornos accidentados, entre otros factores [1].

A su vez, la tecnología a dado grandes pasos en el campo de la visión computacional [2], más en específico en la detección de objetos a través de algoritmos complejos, dentro de los que destacan, existen las Redes Neuronales (Neural Network), las cuales según IBM son “*un subconjunto del aprendizaje automático, y están en el centro de los algoritmos de aprendizaje profundo. Están compuestas de capas de nodos, que contienen una capa de entrada, una o más capas ocultas y una capa de salida. Cada nodo se conecta a otro, y tiene un peso y umbral asociados [3]*“. Estas mismas poseen varias vertientes, entre las cuales se encuentran

las Redes Neuronales Convolucionales (CNN), las cuales son de gran utilidad en el procesamiento de imágenes.

El proyecto, en esta ocasión, estará trabajando en el desarrollo y entrenamiento de una CNN, la cual a través de técnicas de visión computacional, logre hacer una detección de personas a distancias considerables en entornos caóticos y rurales, con el fin de que en futuros trabajos, sea implementado a vehículos aéreos no tripulados (Drones convencionales, o bien UAV's).

Antecedentes

El desarrollo de software para la detección de personas no es algo nuevo, es algo que lleva bajo estudio desde la década de los 70's con los proyectos como los de reconocimiento facial por parte de un equipo de investigación dirigido por Woodrow W Bledsoe [\[4\]](#), que con el paso del tiempo han ido evolucionando por una gran comunidad de investigadores a lo largo del mundo. Teniendo como resultado proyectos los cuales se asemejan a nuestro proyectos, tales como Dronefinder [\[5\]](#), el cual busca a través de drones identificar a las personas por medio de cámaras y cabinas móviles de control. También existen más proyectos, como el dirigido por Aerocamaras [\[6\]](#) que busca implementar drones que a su vez transporten material de utilidad para emergencias, o el titulado “*5G: Rescate de personas con drones*” con sede en Andalucía, España, el cual tiene como objetivo de la misma forma, implementar tecnologías enfocadas en drones para el apoyo a grupos de rescate, haciendo énfasis en su colaboración con la Cruz Roja [\[7\]](#).

Todos estos tienen como objetivo facilitar las tareas de búsqueda y rescate de personas en situaciones de emergencia, aunque claro, todos con una metodología distinta, en nuestro caso, nuestro objetivo es darle al dron la capacidad de identificar personas de forma autónoma, sin necesidad de la manipulación de un elemento, ya que claro, no todas las personas cuentan con la capacitación necesaria para el uso de los mismos, así como también se sabe que no siempre se cuenta con una gran cantidad de recursos humanos en situaciones de emergencia, por lo que automatizar estas labores de búsqueda serán de gran ayuda para este sector.

Objetivos

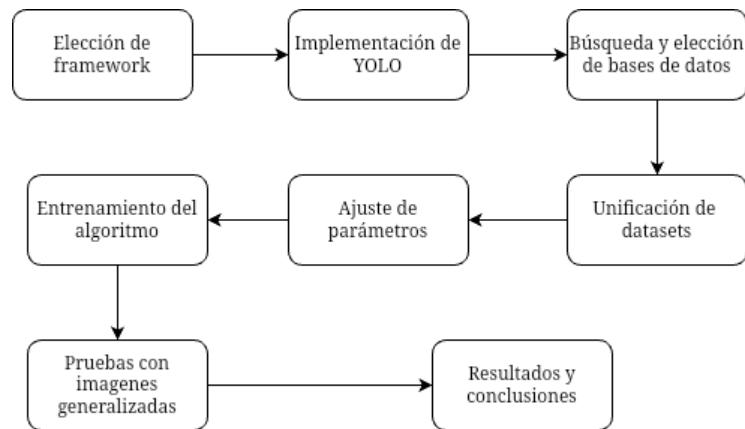
Objetivo general

Desarrollar un software especializado en la detección de personas en situaciones de riesgo, haciendo uso de una CNN, así como bases de datos las cuales nos serán de gran utilidad para este proyecto. Todo ello con el fin de en el futuro, brindar una herramienta a los cuerpos de emergencias en siniestros humanos y naturales para la identificación de la población en puntos de difícil acceso, esto de manera autónoma y buscando la mayor sencillez de uso.

Objetivos específicos

- Encontrar una base de datos ideal para nuestra CNN.
- Desarrollar una CNN para el procesamiento de imágenes con apoyo del framework YOLO, en su versión Yolov8-nano.
- Entrenar la CNN para la detección de humanos con los datasets previamente seleccionados.
- Comprobar y procesar los resultados, analizando el umbral y la precisión con la que la CNN identifica a las personas en imágenes referentes al tema.

Metodología



Elección del framework

Tras una búsqueda exhaustiva de un framework ideal para nuestros objetivos, encontramos Yolo de Ultralytics (You Only Look Once). YOLO utiliza una arquitectura de red neuronal convolucional optimizada para detectar objetos en imágenes en una sola pasada (one-shot detection), lo que lo hace extremadamente rápido y eficiente. Su funcionamiento se basa en los siguientes pasos:

- Entrada de imagen: Se proporciona una imagen o video al modelo.
- Extracción de características: Utiliza una red neuronal convolucional (CNN) para extraer características relevantes de la imagen.
- Predicciones en celdas de la imagen: Divide la imagen en una cuadrícula y predice las cajas delimitadoras (bounding boxes), categorías y confianza para cada objeto.
- Filtrado de predicciones: Usa técnicas como Non-Maximum Suppression (NMS) para eliminar detecciones redundantes y mejorar la precisión.
- Salida de resultados: Devuelve las coordenadas de los objetos detectados, su categoría y la confianza en la predicción.

También se tuvieron en cuenta otros frameworks con los cuales trabajar tales como OpenCV, TensorFlow-Keras, PyTorch, Scikit-Image, entre otros. Aunque la elección de Yolo en nuestro proyecto fue dada por los datos de la siguiente tabla (Tabla 1):

Característica	YOLO	OpenCV / Scikit-Image	TensorFlow / PyTorch
Enfoque	Modelo de detección de objetos.	Procesamiento de imágenes (filtros, bordes, transformaciones).	Aprendizaje profundo (Deep Learning) para visión por computadora.
Tipo de tarea	Detección, segmentación y clasificación de objetos e imágenes y videos.	Manipulacion de imagenes (escalado,conversion de color, deteccion de borde, etc.).	Entrenamiento de modelos personalizados para clasificación, segmentación y detección.
Velocidad	Rápido en línea (tiempo real)	Depende de la tarea, por lo general es rápido.	Tiende a ser más lento dependiendo del modelo y su arquitectura.
Precisión	Alta precisión en detección, aunque depende de la versión y el entrenamiento realizado.	No realiza detección de objetos por sí solo.	Alta precisión, aunque requiere de un entrenamiento intensivo.
Uso de GPU	Puede usar CUDA para acelerar sus procesos.	No requiere GPU (a excepción de OpenCV con CUDA).	Poseen soporte para aceleración por GPU.
Facilidad de uso	Fácil de usar con modelos preentrenados.	Muy simple, ideal para procesamiento básico.	Requiere conocimientos de Deep Learning para una customización profunda.
Entrenamiento	Preentrenado o personalizado.	N / A	Permite entrenar modelos personalizados desde cero.
Aplicaciones generales	Vigilancia, automóviles autónomos, salud, drones.	Edición de imágenes, visión por computadora en general	Modelos personalizados para clasificación, segmentación, generalización de imágenes.

Tabla 1. Comparación entre distintos frameworks

Como podemos apreciar, YOLO en esta ocasión representa una buena elección para satisfacer nuestros objetivos, por sus enfoques, su practicidad, y su relativamente sencilla implementación en drones.

Implementación de YOLO

YOLO es un framework que trabaja por defecto con 24 capas convolucionales, de las cuales cuatro capas de agrupamiento máximo y dos capas totalmente conectadas.

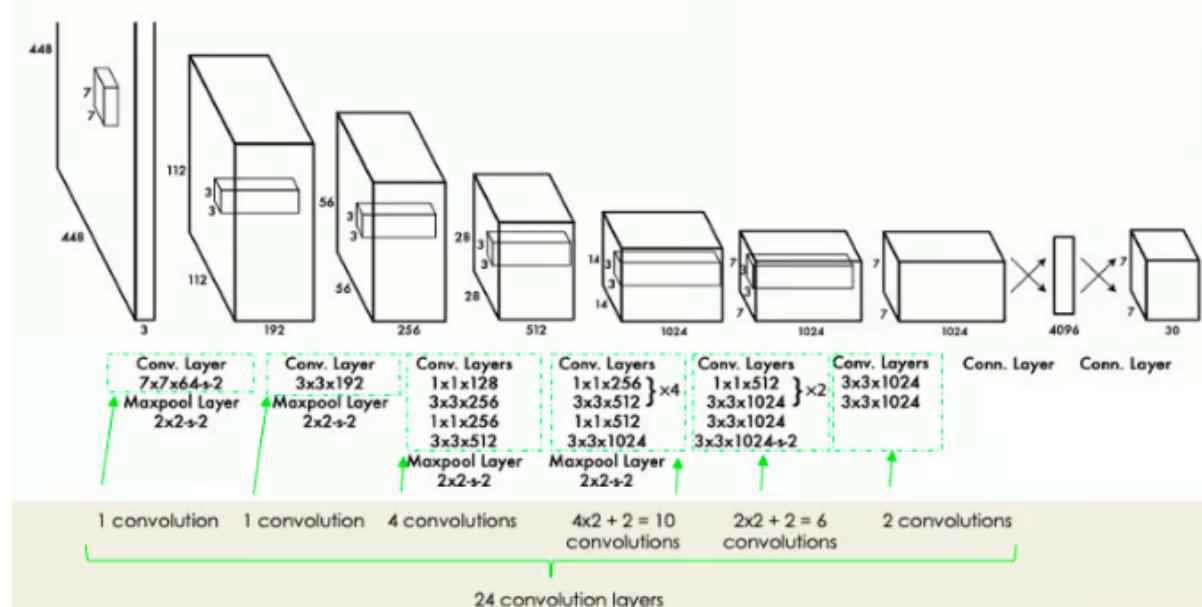


Figura 1. Imagen representativa del summary de YOLO, obtenida de [datacamp.com \[8\]](https://www.datacamp.com)

La cual realiza los siguientes pasos en el procesado de imágenes:

- Cambia el tamaño de la imagen de entrada a 448x448 píxeles antes de pasar por la red convolucional.
- Se aplica una convolución 1x1 para reducir el número de canales, a la que sigue una convolución 3x3 para generar una salida cuboidal.
- Posteriormente pasa por una función de activación (ReLU), excepto en la capa final, que utiliza una función de activación lineal.

El algoritmo funciona principalmente basándose en cuatro enfoques:

1- Bloques residuales: Este primer paso comienza dividiendo la imagen original (A) en celdas de cuadrícula NxN de igual forma. Cada celda de la cuadrícula es responsable de localizar y predecir la clase del objeto que abarca, junto con el valor de probabilidad/confianza, tal y como se muestra en la Figura 2.

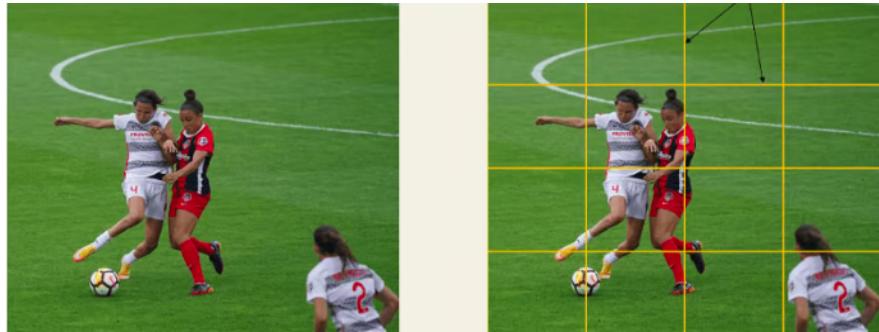


Figura 2. Segmentacion de la imagen, obtenida de datacamp.com [8]

2- Regresión de caja delimitadora : El siguiente paso es determinar las cajas delimitadoras, que corresponden a rectángulos que resaltan todos los objetos de la imagen. Podemos tener tantas cajas delimitadoras como objetos haya en una imagen determinada. YOLO determina los atributos de estos cuadros delimitadores utilizando un único módulo de regresión con el siguiente formato, donde Y es la representación vectorial final de cada recuadro delimitador.

$$Y = [pc, bx, by, bh, bw, c1, c2] \quad (1)$$

pc corresponde a la puntuación de probabilidad de la cuadrícula que contiene un objeto. Por ejemplo, todas las cuadrículas en rojo tendrán una puntuación de probabilidad superior a cero.

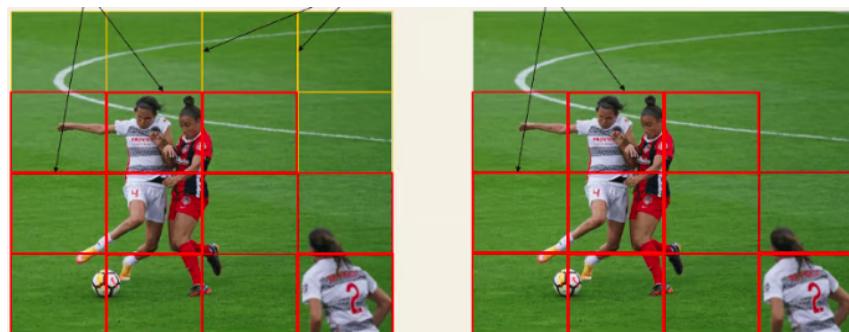


Figura 3. obtenida de datacamp.com [8]

bx , by son las coordenadas x e y del cuadro delimitador con respecto a la celda de la malla envolvente.

bh , bw corresponden a la altura y la anchura del cuadro delimitador con respecto a la celda de la rejilla envolvente.

c1 y c2 corresponden a las dos clases Jugador y Pelota. Podemos tener tantas clases como requiera tu caso de uso.

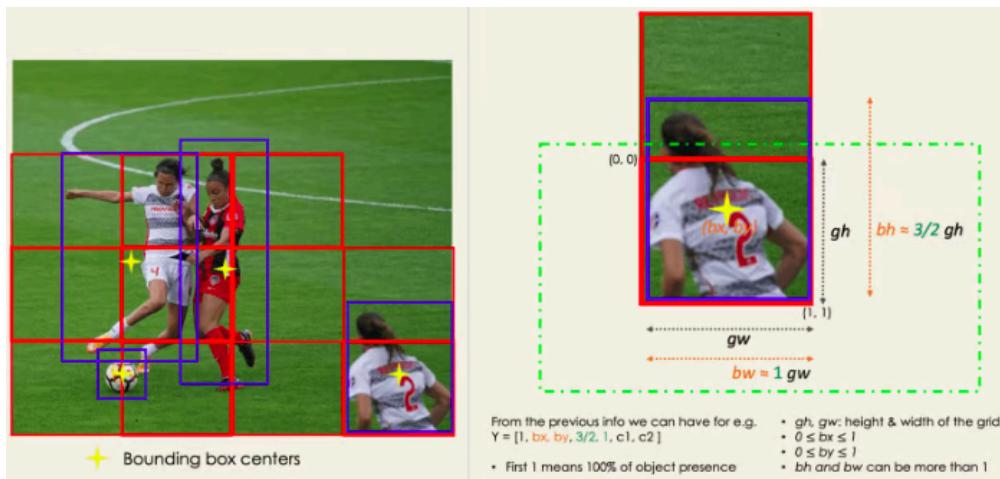


Figura 4. obtenida de datacamp.com [8]

4- Intersección sobre IOU: La mayoría de las veces, un solo objeto de una imagen puede tener varias cajas cuadriculadas candidatas a la predicción, aunque no todas sean relevantes. El objetivo del IOU (un valor entre 0 y 1) es descartar esas casillas de la cuadrícula para conservar sólo las que sean relevantes. Ésta es la lógica que hay detrás:

- El usuario define su umbral de selección IOU, que puede ser, por ejemplo, 0,5.
- A continuación, YOLO calcula el IOU de cada celda de la cuadrícula, que es el área de Intersección dividida por el Área de Unión.
- Por último, ignora la predicción de las celdas de la cuadrícula que tengan un $\text{IOU} \leq$ umbral y considera las que tengan un $\text{IOU} >$ umbral (Figura 5).

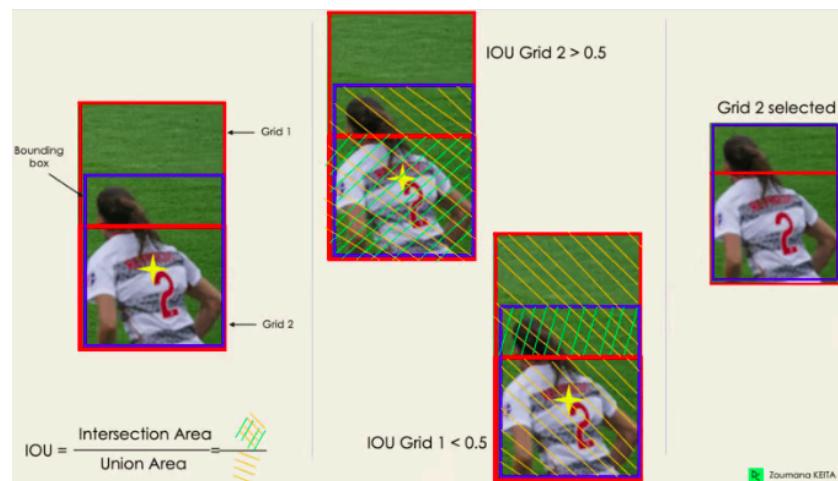


Figura 5. Seleccion de area a trabajar con base al umbral, obtenida de datacamp.com [8]

Establecer un umbral para el IOU no siempre es suficiente, porque un objeto puede tener varias casillas con IOU por encima del umbral, y dejar todas esas casillas podría incluir ruido. Aquí es donde podemos utilizar el NMS (Non-Max Supressor) para conservar sólo las casillas con mayor puntuación de probabilidad de detección.

En esta ocasión, se optó por la implementación de la versión Yolov8-nano, ya que esta posee mayor compatibilidad con otros frameworks, el consumo de recursos es relativamente bajo, y también posee una mayor cantidad de documentación frente a otras versiones.

Búsqueda y elección de bases de datos

En un principio, se hizo el hallazgo de la base de datos C2A, la cual es un recurso diseñado para avanzar en la detección humana en escenarios de desastres utilizando imágenes UAV. Este conjunto de datos aborda una brecha crítica en el campo de la visión por computadora y la respuesta a desastres al proporcionar una colección diversa a gran escala de imágenes sintéticas que combinan escenas de desastres reales con poses humanas [9]. Esta base de datos recaba un aproximado de 10,000 imágenes de resoluciones variadas, que contienen como características principales:

- La toma de imágenes aéreas reales desde alturas considerables
- La colocación de un grupo de humanos de manera aleatoria en las imágenes
- Existen imágenes generadas por IA
- Contiene una sección de etiquetas (labels) en formato COCO por cada imagen, con la información de la ubicación de las personas en las imágenes.



Figura 6. Ejemplo de la base de datos C2A

La base de datos C2A posee una licencia MIT, la cual ofrece a los desarrolladores y organizaciones una opción de licencia permisiva y no restrictiva. Permite a individuos, organizaciones y usuarios utilizar, modificar, distribuir y sublicenciar libremente el software [10]. Y cabe mencionar que, esta base de datos no posee imágenes de personas heridas o en un mal estado de salud. También todas las personas las cuales aparecen en las imágenes mantienen su identidad en oculto, por lo que no se hace uso de su información, manteniéndose en el anonimato.

Posteriormente, se añadió al proyecto la base de datos VisDrone [11], la cual es una base de datos diseñada para la detección, seguimiento y segmentación de objetos en imágenes y videos capturados por drones. Fue desarrollada por el equipo Object Detection in Drone-captured Scenes (ODAI) en la Universidad de Tianjin, China. De la misma forma, esta incluye un aproximado de 10,000 imágenes, las cuales fueron tomadas en diferentes escenarios, tales como zonas urbanas, zonas rurales, carreteras e intersecciones.



Figura 7. Ejemplo de la base de datos VisDrone

Unificación de datasets

Se integraron ambos datasets en formato YOLOv8 para poder tener mayor diversidad de situaciones, por lo tanto, tener un mejor entrenamiento. Para ello, se tomaron las etiquetas de las imágenes de VisDrone, para así poder filtrar solo las imágenes que cuenten con las etiquetas de personas.

Se revisó el número de etiquetas que se han obtenido con personas, más el de etiquetas que no, y se verificó que la suma de imágenes con y sin personas, coincidiera con el número total del dataset de VisDrone para garantizar que no hubiera pérdida en el filtramiento. Corroborando que no ha habido ningún fallo, se trasladaron las imágenes con sus etiquetas correspondientes a la carpeta de C2A respetando la estructura correspondiente de las carpetas tanto de /val como de /train, usando las mismas imágenes separadas como vienen en VisDrone.

Posteriormente, se realizó el conteo final de imágenes para saber el número total después de la unificación de ambos datasets para propósitos de documentación. Obteniendo un aproximado de 16,000 imágenes.

Desarrollo

Cabe mencionar que estos procedimientos se están realizando en servicios de sistemas en la nube, más específico, en Google Colab, en el entorno de ejecución con GPU T4, para una mayor eficiencia en el tiempo de procesamiento de imágenes, también haciendo uso de la compatibilidad que posee con Google Drive, ya que en este mismo se almacenan las carpetas

con las bases de datos, y que a su vez, realiza el almacenamiento de versiones del código de la CNN, por respaldo de la misma, que a su vez, también nos sirvió para almacenar los checkpoints de cada ejecución del entrenamiento.

Ajuste de parámetros

Comenzamos realizando un tune (modificación) del modelo que se utilizó, esto con el fin de lograr obtener los mejores hiperparametros para poder entrenar el modelo con el dataset correspondiente, siendo los siguientes hiperparametros los seleccionados para el entrenamiento:

- Gradiente por minilotes de 16 datos
- Cambio de entrada de la imagen a 640x640 pixeles
- Un total de 600 épocas, de las cuales, por cada una generará un checkpoint, de los cuales, tendrá una marca en la última época con la menor pérdida.
- La condición de paro fue: si posterior a 20 épocas la pérdida de información no presenta mejoras, el entrenamiento se detiene.

El resultado final del entrenamiento será la última época con la menor pérdida de información.

En consecuencia, se modificaron los parámetros de detección del modelo, eliminando la detección de objetos los cuales no sean humanos, esto en un documento de formato “.yaml”, que después, generará boxes (recuadros) los cuales en la parte superior mostrará el umbral del algoritmo sobre la certeza de que sea un humano lo que está señalando, este umbral maneja parámetros de 0 a 1.

Entrenamiento del algoritmo

Consistió en procesar esta nueva base de datos a través de la CNN con los hiperparametros previamente ajustados, para lo cual, se obtuvo la Tabla 2, la cual corresponde a la respuesta del proceso de entrenamiento.

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances
182/600	4.52G Class all	1.105 Images 2024	0.6862	0.9328	63 R 0.72
183/600	4.57G Class all	1.108 Images 2024	0.6875	0.9337	33 R 0.72
184/600	3.73G	1.102	0.6849	0.9316	31

	Class all	Images 2024			R 0.72
185/600	4.58G Class all	1.105 Images 2024	0.6894	0.9331	12 R 0.72
186/600	5.03G Class all	1.111 Images 2024	0.6877	0.933	37 R 0.72
187/600	4.73G Class all	1.112 Images 2024	0.6902	0.9332	33 R 0.72

Tabla 2. Toma de información sobre el entrenamiento entre las épocas 182 a 187.

Resultados

A continuación se muestran los resultados de imágenes procesadas posterior al entrenamiento, las mismas muestran en un cuadro a las personas detectadas con su calificación sobre la seguridad que tiene el algoritmo de que estas mismas sean humanos.



Figura 8. Resultado en la detección de personas de la base de datos C2A



Figura 9. Resultado en entornos montañosos, una sola persona

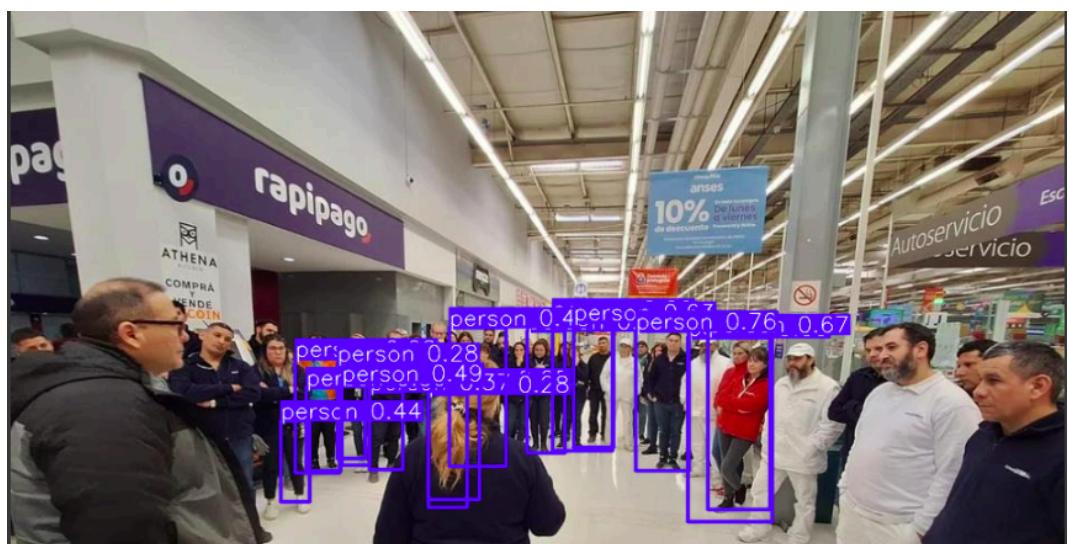


Figura 10. Resultado: en aglomeraciones cercanas a la capturadora, es posible detectar errores por cuestiones del entrenamiento, ya que el mismo fue hecho desde posiciones relativamente lejanas.



Figura 11. Resultado en entornos boscosos, con un umbral aceptable.

Discusión

Como se puede apreciar, el algoritmo cumple con los objetivos establecidos en un inicio, ya que logra detectar a las personas en las condiciones previstas para el modelo. Funciona de buena manera para los entornos esperados, ya sea desastres naturales o tomas aéreas de personas y los resultados iniciales indican que el modelo logra desempeñarse correctamente en los escenarios para los cuales fue diseñado

Una de las limitaciones más importantes que se pueden apreciar al modelo, es la detección de personas muy cerca del lente, ya que no logra hacer una detección precisa de las personas, esto se debe a que dentro de las imágenes del dataset con el cual ha sido entrenado el modelo, no incluye ese tipo de escenarios. Cabe aclarar que esto no es un fallo del modelo, sino que no es un propósito del mismo, solo busca que detecte personas en espacios abiertos desde vistas elevadas, no personas a muy corta distancia, ya que el enfoque está en maximizar la zona visible en cada imagen

También el modelo tiene limitaciones al procesar imágenes de calidad extremadamente alta o baja, porque al tener un nivel extremadamente alto o muy bajo, el modelo tiende a perder precisión en la interpretación de la información visual. Los espacios oscuros son otra debilidad que presenta el modelo, ya que no dispone de suficiente información visual para emitir una detección con alta confianza.

En ciertos casos, se ha notado que llega a necesitar identificar el cuerpo completo para detectar en un umbral de confianza alto a las personas, aunque esto no ocurre en todos los casos, si es una tendencia observable en detecciones parciales

Se plantea la posibilidad de que al incluir cierto tipo de imágenes como escenarios nocturnos o de diversas calidades, podría contribuir a una mayor precisión en la detección de personas, ya que esto, cubriría los "vacíos" que puede llegar a tener el dataset y generalizarían más el modelo. El impacto esperado sería mejorar el recall y una posible reducción en la cantidad de falsos negativos

Se evaluó la posibilidad de sobreentrenamiento pero no se identificaron indicios claros, esto sugiere que el modelo generaliza razonablemente bien dentro de su dominio. Sin embargo, no se descarta del todo y podría considerarse en futuros ajustes. Otra posible causa de fallas es la distribución desigual de ejemplos.

Este resultado permite centrar los esfuerzos de mejora en otros factores más vinculados a la calidad y variedad de los datos, más que en el ajuste del modelo base.

Aunque el modelo cumple con su objetivo, existen oportunidades para optimizar su desempeño. Una posible línea de mejora consiste en ampliar el dataset con imágenes que presenten distintos niveles de iluminación y calidad. Además, se propone experimentar con otras versiones del modelo, como YOLOv8x, con el fin de evaluar si un modelo más complejo puede mejorar la capacidad de detección en condiciones variables. Estas mejoras podrían implementarse en versiones posteriores del sistema, para robustecer su aplicación en contextos reales de emergencia.

Conclusiones y trabajos futuros

El objetivo principal de este trabajo fue desarrollar un sistema capaz de detectar personas en contextos de desastres naturales a partir de imágenes aéreas capturadas por drones, con el propósito de contribuir a una respuesta más eficiente en situaciones donde el tiempo resulta crítico para la supervivencia. Para ello, se entrenó un modelo basado en YOLOv8 utilizando datasets sintéticos y reales (C2A y VisDrone), priorizando su capacidad de generalización frente a escenarios complejos.

Los resultados obtenidos indican que el modelo logra identificar personas en imágenes aéreas bajo distintas condiciones de iluminación y distancia, alcanzando un desempeño técnico adecuado para su propósito. Asimismo, se observaron tiempos de inferencia reducidos, lo que sugiere que el sistema podría ser útil en situaciones que demanden un análisis rápido de imágenes.

Si bien el sistema aún no ha sido implementado en escenarios reales, los experimentos realizados demuestran la viabilidad técnica del enfoque. Este trabajo representa un primer paso hacia el desarrollo de herramientas automatizadas que puedan asistir en tareas de búsqueda y rescate, con el potencial de optimizar recursos y contribuir a salvar vidas en contextos de emergencia.

Como líneas futuras de trabajo, se propone seguir fortaleciendo el modelo mediante la incorporación de datos aún más variados, incluyendo escenarios nocturnos, presencia de obstáculos, cuerpos parcialmente visibles y entornos urbanos complejos. También se plantea explorar versiones más robustas del modelo, como YOLOv8x, y aplicar técnicas de aumento de datos que puedan mejorar su capacidad de generalización.

Adicionalmente, sería valioso validar el sistema en contextos más cercanos a situaciones reales, ya sea a través de simulaciones con drones en entornos controlados o mediante pruebas en secuencias de video en tiempo real. Estas acciones permitirían acercar el desarrollo actual a una solución práctica y confiable, con potencial de integrarse en protocolos de respuesta ante desastres naturales.

Referencias

[1] V. Dannemann. “América Latina, azotada por desastres naturales – DW – 20/02/2025”. dw.com. Accedido el 28 de marzo de 2025. [En línea]. Disponible: <https://www.dw.com/es/américa-latina-azotada-por-desastres-naturales/a-71688827>

[2] “Avances en Visión por Computadora”. Martes Tecnológico. Accedido el 30 de marzo de 2025. [En línea]. Disponible: <https://www.martestecnologico.com/avances-en-vision-por-computadora/>

[3] IBM. “¿Qué son las redes neuronales convolucionales? | IBM”. IBM - United States. Accedido el 30 de marzo de 2025. [En línea]. Disponible: <https://www.ibm.com/mx-es/think/topics/convolutional-neural-networks>

[4] “Woodrow Bledsoe Originates Automated Facial Recognition : History of Information”. History of Information. Accedido el 2 de abril de 2025. [En línea]. Disponible: <https://www.historyofinformation.com/detail.php?entryid=2495>

[5] “YOLO: Real-Time Object Detection”. Survival Strategies for the Robot Rebellion. Accedido el 29 de marzo de 2025. [En línea]. Disponible: <https://pjreddie.com/darknet/yolo/>

[6] “Dronefinder – Sistema de búsqueda y localización de desaparecidos”. Dronefinder. Accedido el 2 de abril de 2025. [En línea]. Disponible: <https://dronefinder.es/>

- [7] “Drones en emergencias y rescates | Aerocamaras”. Aerocamaras Especialistas en Drones. Accedido el 2 de abril de 2025. [En línea]. Disponible:
<https://aerocamaras.es/servicios-drones-profesionales/emergencias-y-rescates-con-drones/>
- [8] Z. Keita. “YOLO Object Detection Explained”. datacamp.com. Accedido el 31 de marzo de 2025. [En línea]. Disponible:
<https://www.datacamp.com/blog/yolo-object-detection-explained>
- [9] R. NIhal, Y. Benjamin, K. Itoyama y K. Nakadai. “UAV-Enhanced Combination to Application: Comprehensive Analysis and Benchmarking of a Human Detection Dataset for Disaster Scenarios”. kaggle.com. Accedido el 24 de marzo de 2025. [En línea]. Disponible:
<https://www.kaggle.com/datasets/rbgnihal/c2a-dataset/data>
- [10] “Licencia MIT | AppMaster”. AppMaster - The No-Code platform for building web & mobile apps. Accedido el 1 de abril de 2025. [En línea]. Disponible:
<https://appmaster.io/es/glossary/licencia-mit>
- [11] “VISDRONE”. VISDRONE. Accedido el 30 de marzo de 2025. [En línea]. Disponible:
<http://aiskyeye.com/>