

Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica
IE0499 - Proyecto Eléctrico - Grupo 01
Profesor: Dr. Jaime Cascante Vindas

Proyecto Eléctrico I-2025

Profesor Encargado: Dr. Marvin Coto Jiménez

Contabilización de Partículas Pequeñas en Imágenes utilizando Técnicas de Aprendizaje Automático

—
DSM-Contabilización de partículas

Realizado por:

Fecha de asignación: 14/03/2025

Danny Solórzano Mayorga

Fecha de entrega: 30/06/2025

Carné: C27667

Dirección: San Isidro, Alajuela, 20106

Tel: +506 61616444

Correo: danny.solorzano@ucr.ac.cr

Resumen

Este proyecto plantea desarrollar un sistema automatizado capaz de contabilizar partículas pequeñas en fotografías, basándose en un conjunto de datos previamente recopilado. Se explorarán diferentes métodos para el procesamiento de imágenes y algoritmos de aprendizaje automático para realizar el conteo en las imágenes. El desafío principal consiste en establecer las estrategias de contabilización, la cual se puede abordar con la cuantización de la cantidad que es posible detectar.

Índice

1	Objetivos del Proyecto	1
1.1	Objetivo General	1
1.2	Objetivos Específicos	1
2	Metodología	1
2.1	Investigación de literatura	1
2.2	Selección y Análisis del Dataset	2
2.3	Preprocesamiento de Imágenes	2
2.4	Entrenamiento del Modelo	2
2.5	Establecimiento de los Límites de Detección	2
2.6	Evaluación y Ajuste del Modelo	2
2.7	Generación de Resultados y Recomendaciones	3
3	Introducción	4
4	Estado del Arte	8
4.1	Evolución de la Detección de Objetos	8
4.2	Detección de Objetos Pequeños: Desafíos y Soluciones	9
4.3	Evaluación Escalable y Métricas de Precisión	10
4.4	Aplicaciones Especializadas: Imágenes Satelitales y R2-CNN	10
4.5	Contribuciones Fundamentales para el Proyecto	11
5	Descarga y análisis del dataset disponible	12
6	Creación del conjunto de datos de entrenamiento	13
7	Exploración de repositorios con modelos de clasificación de imágenes	14
8	Seleccionar el repositorio a utilizar con el conjunto de imágenes	16
8.1	Elección del modelo	20
9	Experimentación y Enfoque Utilizado	20
10	Resultados Obtenidos	21
11	Estudio de Límites de Detección	23
12	Conclusiones	24
13	Recomendaciones Futuras	25
	Referencias	27

1. Objetivos del Proyecto

1.1. Objetivo General

Desarrollar un sistema de contabilización de partículas pequeñas en imágenes, capaz de cuantificar partículas utilizando técnicas de aprendizaje automático.

1.2. Objetivos Específicos

1. Realizar una investigación en la literatura acerca del problema de detección de elementos pequeños en imágenes para identificar tendencias actuales y los modelos más convenientes.
2. Realizar un análisis del conjunto de datos disponible y preparar los datos para el entrenamiento del modelo.
3. Implementar un modelo de aprendizaje automático, para detectar partículas en imágenes.
4. Definir de manera preliminar los parámetros que determinan cuántas partículas pueden ser detectadas.
5. Evaluar la precisión del modelo utilizando métricas estándar (precisión, recall, F1-score) y analizar cómo la variabilidad de tamaño, cantidad y contraste afecta la precisión.

2. Metodología

2.1. Investigación de literatura

1. Se propone la lectura, análisis y resumen de los siguientes artículos:
 - a) Ma, Z., Yu, L., & Chan, A. B. (2015). Small instance detection by integer programming on object density maps. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3689-3697) [Ma, Yu y Chan 2015].
 - b) Chen, C., Liu, M. Y., Tuzel, O., & Xiao, J. (2017). R-CNN for small object detection. In Computer Vision—ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part V 13 (pp. 214-230). Springer International Publishing [Chen et al. 2017].

- c) Yu, X., Han, Z., Gong, Y., Jan, N., Zhao, J., Ye, Q., ... & Shi, H. (2020). The 1st tiny object detection challenge: Methods and results. In Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16 (pp. 315-323). Springer International Publishing [Yu et al. 2020].
2. Después de la lectura de estos tres artículos iniciales, se deben consultar al menos tres artículos adicionales relacionados con el tema para escribir un pequeño estado del arte y una descripción de los problemas principales analizados hasta el momento en cuanto a detección de objetos pequeños en imágenes. Un punto de partida puede ser: <https://github.com/kuanhungchen/awesome-tiny-object-detection>

2.2. Selección y Análisis del Dataset

1. Se utilizarán los datasets contenidos en: <https://www.kaggle.com/datasets/daenys2000/small-object-dataset/data> para entrenar y evaluar el modelo. Estos datasets se deben revisar y documentar en cuanto a cantidad de imágenes disponibles, su resolución y la selección de algoritmos disponibles para su aprovechamiento.

2.3. Preprocesamiento de Imágenes

1. Normalización de las imágenes para mejorar el contraste.
2. Uso de técnicas de aumento de datos para incrementar la diversidad de las imágenes y la construcción del groundtruth.

2.4. Entrenamiento del Modelo

1. Implementación de un modelo de red neuronal convolucional (CNN) o redes neuronales más avanzadas como YOLO o Faster R-CNN. <https://github.com/hamzagorgulu/Small-Object-Detection-with-YOLO>
2. Ajuste de hiperparámetros y validación cruzada para optimizar el modelo.

2.5. Establecimiento de los Límites de Detección

1. Experimentación con diferentes tamaños de objetos en las imágenes y la cuantización de las cantidades que es posible detectar.

2.6. Evaluación y Ajuste del Modelo

1. Evaluación utilizando métricas de rendimiento estándar como precisión, recall y F1-score.

2. Análisis del impacto de diferentes configuraciones sobre la precisión del modelo.

2.7. Generación de Resultados y Recomendaciones

1. Propuesta de umbrales y estrategias para la detección de partículas en función de los resultados obtenidos.
2. Análisis de la viabilidad de aplicar este sistema en entornos reales.

3. Introducción

La detección y contabilización de partículas pequeñas en imágenes es un problema muy importante en el campo de la visión por computadora. Tiene aplicaciones en áreas como la biología microscópica, la seguridad, la robótica, el monitoreo ambiental y la manufactura de precisión. En todos estos casos se presentan desafíos particulares por causa del tamaño reducido de los objetos de estudio, la baja resolución relativa que tienen, la variabilidad en las condiciones de captura y la presencia de ruido o presencia de un fondo extraño que se puede confundir con las partículas de interés.

En este contexto, el presente proyecto busca desarrollar un sistema automatizado para contabilizar partículas pequeñas en imágenes, al usar técnicas de aprendizaje automático. Todo esto con el fin de superar las limitaciones de los enfoques de conteo manual o heurístico.

En el campo de la visión por computadora, uno de los problemas que son más persistentes y complejos es la detección precisa y eficiente de objetos pequeños en imágenes. A pesar de que se ha logrado un gran avance en tareas de detección de objetos de tamaño mediano o grande por medio al desarrollo de modelos de aprendizaje profundo, la detección y contabilización de objetos pequeños continúa representando un reto técnico importante, en especial cuando cuando estos objetos están agrupados densamente o distribuidos de forma muy irregular.

Este problema es especialmente importante en aplicaciones científicas, médicas e industriales, donde es necesario identificar y contar partículas de tamaño muy reducido como células, insectos, defectos en materiales o elementos microscópicos, los cuales en muchas ocasiones solo representan unos cuantos píxeles en una imagen de alta resolución.

Motivación

Como ya se mencionó, en muchas disciplinas, como la biología celular, la microbiología, la química analítica y los procesos de manufactura, el análisis de imágenes que tienen partículas de tamaño reducido es una necesidad bastante común. A pesar de esto, el análisis manual de estas imágenes es un proceso muy laborioso y que está bastante propenso al error humano. Además, es casi inviable recurrir a otros métodos cuando se presentan casos donde se requiere un procesamiento masivo y en tiempo real.

Los métodos clásicos que se utilizan para esto, están basados en umbrales de binarización, detección de bordes o filtrado morfológico. Estos son herramientas muy útiles, pero suelen fallar cuando cambian algunas condiciones como la iluminación, cuando las partículas son casi imperceptibles o cuando se traslapan entre ellas.

La dificultad se basa en el hecho de que estos objetos tienen una cantidad limitada de información y eso hace que sea difícil diferenciarlos del fondo de la imagen. Además, por

esta misma razón, las redes convolucionales tradicionales no son capaces (en general) de identificarlos de forma eficiente.

La literatura científica reciente ha notado esta problemática como un problema un poco más específico dentro de la detección de objetos, al que se le ha denominado **detección de objetos pequeños** o **tiny object detection**. La categoría se refiere a objetos que representan menos del 2 % del área de la imagen o los que tienen un tamaño menor a 32×32 píxeles.

Revisión de la Literatura Relevante

Tres trabajos muy importantes son los que orientan y justifican el enfoque que se toma en este proyecto. El primero apunta a lo que se explica a continuación.

El reconocimiento de objetos pequeños ha estado históricamente retrasado en comparación con la detección de objetos grandes, lo cual es claro debido a la falta de datasets adecuados y a la complejidad que implica dicho trabajo. Las partículas pequeñas ocupan un área muy pequeña dentro de la imagen, lo cual reduce mucho la cantidad de información discriminativa que pueden usar los modelos.

Esto da un bajo desempeño de los detectores tradicionales, en particular los que están basados en regiones propuestas o clasificadores como los enfoques R-CNN en su forma original. Así lo destaca el artículo [Chen et al. 2017]. En este primer trabajo se afirma que los detectores como R-CNN tienen una caída bastante clara en el rendimiento cuando se aplican a objetos que ocupan menos del 1 % del área de la imagen, porque falla la generación de regiones propuestas y eso da una representación insuficiente en las capas intermedias del modelo.

Para resolver estas limitaciones, se han visto enfoques innovadores que incorporan información contextual, modelos de densidad y formulaciones de optimización más robustas. Esto es lo que propone el segundo artículo que guía este trabajo [Ma, Yu y Chan 2015]. En este se propone un marco basado en mapas de densidad de objetos y programación entera. Esto permitiría resolver de forma simultánea las tareas de detección y conteo.

Este enfoque es muy útil cuando las partículas se superponen o están agrupadas. Lo importante es que permite recuperar la ubicación y la cantidad de objetos y no depende de la segmentación precisa de cada instancia. Por medio de la segmentación del mapa de densidad de forma local y al usar restricciones globales de conteo, el método es bastante mejor a otros métodos puramente discriminativos, en particular en datasets que pueden ser complejos, como células microscópicas, enjambres de insectos y multitudes densas de personas.

El tercer artículo que inspira el rumbo de este trabajo describe a continuación [Yu et al. 2020]. En este se han empujado los límites de la detección de objetos pequeños en imágenes de alta resolución. Aquí se presenta un enfoque con un amplio campo de visión, como las imágenes captadas por cámaras de vigilancia o drones.

Esto es un nuevo reto, debido a que, al aumentar el campo de visión, los objetos de estudio (de tamaño pequeño) ahora son proporcionalmente más pequeños y, por tanto, se vuelven mucho más difíciles de detectar. En el trabajo se exploran arquitecturas avanzadas como Faster R-CNN, Cascade R-CNN y FPN y estrategias de entrenamiento multiescala, técnicas de aumento de datos específicas para distribución de tamaños, y métodos de ensamble jerárquico.

Entonces, todos estos avances demostraron que, si bien los modelos convolucionales pueden ser muy eficaces para tareas generales de detección, es posible que se requieran ajustes importantes y conocimiento experto adicional para lograr extrapolar todo esto al dominio de objetos de tamaño pequeño. Esta es la línea sobre la cual se encamina este proyecto.

La metodología general para trabajarlo parte de entender que una sola implementación de una red neuronal convolucional no es suficiente para resolver el problema de manera precisa y eficiente, y por esto se utilizará una forma de trabajo más integral, empezando por el análisis de literatura, pasando con la exploración y adaptación de los datos, siendo luego al preprocesamiento inteligente de imágenes, luego realizando la selección y entrenamiento de modelos de aprendizaje automático y, finalmente, evaluando el éxito por medio de métricas estándar como precisión, recall y F1-score.

Además, es importante recordar que se abordará el problema desde una perspectiva cuantitativa y no cualitativa, lo cual permitirá ver no solo la capacidad del sistema para detectar partículas individuales, sino que también se podrán ver aspectos como la variabilidad en tamaño, densidad y contraste de las partículas pequeñas. Así, se puede decir que un aspecto central del proyecto está en el hecho de establecer umbrales que permitan definir hasta qué punto el sistema puede discernir partículas sin que ocurran algunos falsos positivos o negativos.

Rumbo del Proyecto

Por su parte, es necesario mencionar algunos de los principales aportes que se esperan de este proyecto. El proyecto busca contribuir en varios aspectos como los siguientes. En primer lugar, se quiere proveer una herramienta que sea efectiva para la detección de partículas pequeñas en contextos donde los métodos clásicos normalmente fallarían. En segundo, se busca validar el uso de los mapas de densidad junto con la optimización al utilizarlos como una alternativa que es bastante útil con respecto a los detectores puramente discriminativos. En tercero, se desea generar conocimiento que sea potencialmente aplicable a diversas áreas donde puede ser posible que se necesite contar elementos pequeños, como la citometría, el conteo de defectos en materiales o el monitoreo de fauna silvestre en diferentes partes del país.

Así, se puede decir que el presente trabajo es un punto medio entre la ingeniería de datos, la visión por computadora y el aprendizaje automático, que tiene como objetivo principal

generar un sistema que sea confiable, escalable y adaptable para la detección y conteo de partículas de tamaño reducido.

Los desafíos que esto puede presentar son diversos, como se explicó anteriormente, pero, así mismo, las oportunidades de impacto que esto puede tener en diversos campos científicos y tecnológicos son muy grandes. Una vez planteada una base sólida teórica y experimental, el proyecto buscar avanzar en el estado del arte y tiene el fin de contribuir con una herramienta práctica para problemas reales en los que la detección de partículas de tamaño reducido es una tarea crítica.

Por tanto, con todo lo mencionado anteriormente, se puede afirmar que este trabajo se enmarca dentro de la necesidad y búsqueda de soluciones inteligentes para problemas que pueden resultar complejos en el análisis de imágenes, lo cual es un área creciente en la actualidad, donde el aprendizaje automático es una herramienta fundamental. Aprovechando los avances más recientes realizados en este sentido y analizarlos dentro de una solución específica para detección de objetos pequeños, este proyecto es, en primera instancia, un ejercicio técnico y, posteriormente, se plantea como un aporte práctico al campo de la visión computacional aplicada.

4. Estado del Arte

La detección de objetos de tamaño reducido es uno de los problemas más persistentes y uno de los retos más complejos que hay actualmente en el campo de la visión por computadora. A pesar de los avances varios que se han hecho en clasificación y localización de objetos de tamaño regular, al utilizar redes neuronales profundas, el problema de identificar y contabilizar partículas diminutas sigue siendo un reto significativo. El interés en ello radica en el hecho de que es aplicable a problemas reales, ya que esas partículas podrían ser partículas microscópicas, defectos materiales, células o vehículos lejanos. Este capítulo presenta una revisión de algunas investigaciones y desarrollos tecnológicos relacionados con el área en el estudio y su relación con el presente proyecto, haciendo principal énfasis en las soluciones que se presentan a la detección de objetos pequeños y su relación con el conteo automatizado.

4.1. Evolución de la Detección de Objetos

En las últimas dos décadas, la detección de objetos ha evolucionado mucho. Se han implementado métodos basados en descriptores manuales como HOG o Haar y complejas arquitecturas de redes neuronales convolucionales y transformers [Zou et al. 2023]. En esta parte es importante destacar que el progreso se puede dividir en dos partes: la era de los métodos tradicionales (antes del 2014) y la era del aprendizaje profundo (después del 2014), donde aparecen modelos como R-CNN, Faster R-CNN, YOLO y RetinaNet. Esto permitió mejorar mucho en la precisión y velocidad, aunque los objetos de tamaño reducidos aún eran un punto débil, por su casi despreciable representación en los mapas de características y también debido a los problemas que había en la escala y contexto.

Los primeros métodos que se presentaron estaban basados en características hechas manualmente como SIFT (Scale-Invariant Feature Transform), HOG (Histogram of Oriented Gradients) y clasificadores como SVMs. Estos eran muy eficientes en entornos controlados, pero no veían variaciones contextuales o patrones complejos en imágenes reales. Luego, cuando apareció AlexNet y el uso masivo de GPU la visión por computadora dio un salto importante y eso marca el comienzo de la era del aprendizaje profundo en la detección de objetos.

Algunos modelos como Faster R-CNN, YOLO (You Only Look Once) y RetinaNet eran soluciones estándar para estos problemas normales de detección. No obstante, su rendimiento para objetos pequeños era muy limitado, debido a que las capas profundas de las CNN tienden a perder información espacial precisa, la cual es sumamente necesaria para detectar objetos pequeños. Esto motivó el desarrollo de nuevas estrategias.

4.2. Detección de Objetos Pequeños: Desafíos y Soluciones

Un trabajo muy importante e ilustrativo sobre esto es el de [Ma, Yu y Chan 2015]. Aquí se muestra un enfoque híbrido que combina mapas de densidad con programación entera para detectar objetos de tamaño reducido en entornos densos. El modelo permite identificar regiones que tienen una alta densidad de objetos y no requiere exclusivamente de segmentarlos. Esto es muy útil cuando las partículas se traslapan o están en algunos conglomerados densos. De hecho, esto es lo que ocurre en imágenes de microscopía, multitudes urbanas o cúmulos de polvo que puede haber en ciertos sectores industriales. Este enfoque no está basado en propuestas tradicionales de regiones ni en la existencia de una segmentación explícita, lo cual es una gran ventaja, ya que lo vuelve escalable y robusto. En este proyecto, esta idea es muy útil para contextos donde las partículas no se encuentran separadas por completo o donde se desea obtener un conteo sin tener que localizar con precisión cada contorno individual de cada una de estas.

Por su parte [Chen et al. 2017] se enfrenta con el problema desde la arquitectura de los modelos. Así, adapta el clásico R-CNN y mejora su capacidad de detección de objetos pequeños al hacer ajustes en la propuesta de regiones y añadir contexto espacial. En lugar de confiar solo en propuestas generadas por *selective search*, se mejora la jerarquía de características para conservar información relevante en niveles bajos de la red, donde los objetos pequeños son todavía bastante distinguibles.

Este trabajo resalta por el uso del contexto como la relación de un objeto con su entorno inmediato. Al trabajar con objetos pequeños, esta relación es muy importante, ya que su apariencia puede ser ambigua o a veces es complicado diferenciarlos del fondo. Así, incorporar información contextual hace que mejore mucho la discriminación entre partículas verdaderas y falsos positivos o ruido.

Luego, el artículo [Yu et al. 2020] tiene una visión más contemporánea y muestra que los mejores modelos para este problema de identificación son los que combinan redes de dos etapas con estructuras piramidales (FPN), estrategias multiescala y aumentos de datos especializados. Entre lo más importante de todo esto se encuentra la representación jerárquica de características (feature pyramids), la cual es sumamente relevante para mantener la información visual relevante de objetos que ocupan pocos píxeles.

Los resultados de este desafío dejan ver que ningún modelo por sí solo es suficiente: los equipos que llegaron más lejos usaron una combinación de técnicas de entrenamiento con imágenes aumentadas sintéticamente, uso de FPN, inferencia multiescala y segmentación semántica con detección. Este enfoque tiene sentido y se relaciona con el planteamiento del presente proyecto, ya que se busca incorporar diferentes etapas (preprocesamiento, detección, ajuste de parámetros y evaluación cuantitativa) como un sistema integral.

4.3. Evaluación Escalable y Métricas de Precisión

El artículo de [Shinya 2023] muestra un análisis de las métricas tradicionales para la detección de objetos de tamaño reducido. Aquí se critican las divisiones de tamaño que normalmente se usan en benchmarks como COCO, ya que se afirma que son demasiado gruesas y poco informativas. En el artículo se muestra el BandASAP, un conjunto de métricas de evaluación que se basan en filtros pasa-banda triangulares y trapezoidales, para analizar la precisión por rangos de tamaño de forma confiable.

Esto es particularmente relevante para la detección de partículas, ya que los métodos tradicionales (COCO APsmall, APmedium, APlarge) no diferencia el rendimiento del modelo entre partículas de 8×8 píxeles y de 32×32 . Este tipo de análisis fino es sumamente necesario cuando se quiere establecer umbrales de detectabilidad, como se propone en la metodología de este proyecto antes explicada.

El BandASAP también se destaca porque brinda una correspondencia natural con la estructura de los detectores modernos. Algunas redes usan arquitecturas piramidales donde las resoluciones cambian de forma exponencial entre niveles. Al usar filtros triangulares y trapezoidales se captura con mayor precisión el rendimiento del modelo en estos niveles. Esto permite tomar decisiones de rediseño o ajuste.

4.4. Aplicaciones Especializadas: Imágenes Satelitales y R2-CNN

El trabajo de [Pang et al. 2019] presenta el modelo R2-CNN para abordar la detección de objetos pequeños en imágenes satelitales de muy alta resolución, lo cual tiene muchas similitudes técnicas con el presente proyecto. En imágenes de 18000×18000 píxeles, existe un reto mayor al de la detección de objetos de 8×8 píxeles, que es la eficiencia computacional y la eliminación de falsos positivos.

La arquitectura R2-CNN usa tres bloques principales: Tiny-Net, un backbone liviano diseñado que busca la eficiencia y que es entrenable desde cero; un bloque de atención global, que se usa para agregar contexto espacial; y un sistema de clasificación y detección autoreforzado, así el clasificador puede descartar regiones vacías y evitar el procesamiento que no sea necesario.

Aquí se presenta el enfoque de procesamiento por parches, lo cual es una característica sumamente útil. Ya que las imágenes grandes son subdivididas en bloques de 640×640 píxeles con traslape, se puede reducir la carga computacional sin perder continuidad espacial. Este se podría utilizar en el presente proyecto cuando se utilicen imágenes de alta resolución que contienen miles de partículas distribuidas.

4.5. Contribuciones Fundamentales para el Proyecto

Al analizar estos seis artículos, se identificaron las siguientes contribuciones (las más importantes) que fundamentan e inspiran el presente proyecto:

- De [Ma, Yu y Chan 2015]: Se muestra la idea de usar mapas de densidad y cuantificación espacial al aplicarlas atareas de conteo. Esto es útil cuando las partículas se presentan en aglomeraciones densas. Esto se puede combinar con una capa de refinamiento o postprocesamiento.
- De [Chen et al. 2017]: Se observa la propuesta de ajustar los generadores de regiones propuestas y usar contexto para mejorar las arquitectónicas en modelos como R-CNN o YOLO. El uso de redes multiescala y estrategias de mejora de contexto pueden servir para mejorar el rendimiento en partículas que se distinguen visualmente.
- De [Yu et al. 2020]: Se muestra el uso de detección jerárquica multiescala, estructuras como FPN y el hecho de usar de forma conjunta modelos de dos etapas con técnicas de aumento de datos especializados para objetos de tamaño reducido. Además, podría ser útil la generación de partículas adicionales mediante data augmentation controlado.
- De [Shinya 2023]: Se observa métricas avanzadas como BandASAP para evaluar el modelo según el rango de escala para ver los límites prácticos del sistema. Esto dice qué tan confiable es el conteo cuando se excede cierto nivel de aglomeración o cuando se reduce el tamaño.
- De [Pang et al. 2019]: Se muestra una arquitectura modular como R2-CNN, que se pueden utilizar en problemas reales con imágenes de gran tamaño, al aplicar recortes por parches, backbones livianos y bloques de atención global. La estructura clasificador- detector se puede usar en imágenes biológicas o industriales. En estos casos la mayor parte de las regiones no contienen partículas.
- De [Zou et al. 2023]: Se observa una visión panorámica de la evolución de la detección de objetos, lo cual es sumamente útil para ubicar este proyecto en el contexto histórico de avances en esta área y decidir entre arquitecturas de dos etapas, una sola etapa o transformadores como DETR, que se podrían usar en etapas posteriores del desarrollo del proyecto.

5. Descarga y análisis del dataset disponible

Se procedió a descargar y analizar los datasets contenidos en el siguiente enlace: [daenys2000/small-object-dataset](https://daenys2000.github.io/small-object-dataset/). En particular, se utilizó el que tiene como conjunto base las imágenes de puntos, como la siguiente:



Figura 1: Imagen de muestra del dataset.

Con respecto a esto se puede decir que la base seleccionada es altamente adecuada para el proyecto, tanto en términos técnicos como prácticos. En primer lugar, tiene una gran coincidencia con el objetivo del proyecto: detección y conteo de partículas pequeñas en imágenes. Esta base contiene diferentes ejemplos reales de fotografías de objetos pequeños. En la base de datos están representados de dos formas:

- **Puntos (gt-dots)** como representación del centroide o presencia del objeto.
- **Cajas delimitadoras (gt-bbox)** que podrían ser muy útiles en el proyecto para entrenar detectores basados en YOLO (esta herramienta se explicará en la siguiente sección) o algunos modelos similares.

Cada subdataset tiene dos carpetas:

- **gt-dots/** Mapas binarios con puntos blancos sobre fondo negro, lo cual podría ser muy útil para segmentación o regresión de densidad.
- **gt-bbox/** Archivos `.mat` con coordenadas `[x, y, w, h]`, que podrían ser ideales para detección con redes tipo YOLO.

En este sentido, se puede decir que la base de datos seleccionada se alinea perfectamente con los objetivos específicos del proyecto, ya que: permitirá preparar los datos para la detección y evaluar métodos de contabilización de partículas en imágenes.

Además de todo esto, es importante destacar la diversidad de los subdatasets con objetos pequeños contenidos en esta base de datos. Esto se muestra en la siguiente tabla:

Subdataset	Prom. Objetos	Resolución	Densidad	Relevancia
Fly	86 ± 39	648×72	Alta	Ideal para detección y segmentación en imágenes comprimidas.
Honeybee	28 ± 6	640×480	Media	Tamaño y densidad similares a partículas medianas.
Fish	56 ± 9	300×410	Alta	Fondo ruidoso (agua). Simula condiciones desafiantes.
Seagull	866 ± 107	624×964	Muy Alta	Benchmark para detección en masas densas.

Se van seleccionar imágenes para el proyecto a partir de los sets *Fly*, *Fish* y *Honeybee*, que contienen imágenes con conteos variados. Además, otros aspectos muy positivos son que: las imágenes están todas en formatos comunes (.jpg, .png). Las imágenes de puntos vienen en .png binarios y las de cajas en .mat que facilitaría la conversión YOLO. En general, todo esto da una estructura amigable para preprocesamiento y visualización y, debido a este buen formato, sería posible usar Python con `scipy.io`, `OpenCV`, `NumPy` o alguna otra herramienta para manipular eficientemente y de forma sencilla el dataset, como se verá más adelante.

Por su parte, se debe mencionar que este dataset puede ser muy útil para evaluar la sensibilidad del modelo porque las imágenes varían significativamente en cantidad de objetos por imagen, distribución espacial, contraste con el fondo y se pueden aplicar filtros o seleccionar subconjuntos específicos para pruebas controladas.

Además, a diferencia de otros datasets, este posee un tamaño bastante manejable. Es bastante liviano. También es modular: se puede trabajar solo con una categoría y es manejable en local o en Google Colab, lo cual se alinea con el objetivo de probar entrenamientos pequeños antes de escalar.

Finalmente, cabe mencionar que el dataset seleccionado, posee referencias académicas y respaldo en la literatura. Tiene como fuente al artículo ya conocido *Small instance detection by integer programming on object density maps* [Ma, Yu y Chan 2015].

6. Creación del conjunto de datos de entrenamiento

Para empezar con las pruebas, se construyó un dataset de 200 imágenes a partir del dataset de referencia (con puntos blancos y negros), con una cantidad variable y pequeña de puntos (se buscaba que tuvieran entre 0 y 50 puntos) en diferentes posiciones.

Este dataset se encuentra en el repositorio del proyecto glaurung112/tinyobjectsdetection como *dataset_espaldo* ya que no se utiliza realmente, si no que se optó por realizar un script que permita extraer las imágenes directamente desde Kaggle.

Además, un aspecto que se debe mencionar es que, como ya se vio anteriormente, las imágenes provienen de ambas carpetas, tanto `train/` como `test/` del dataset de Kaggle, ya que ambas contienen menos de 200 imágenes. También, se debe resaltar que realmente las imágenes guardadas no contienen entre 0 y 50 puntos (como se quería inicialmente), sino entre 0 y 60 puntos, ya que solo había disponibles 166 imágenes en el dataset con menos de 50 puntos, por lo cual se aumentó un poco la cantidad máxima de puntos permitida.

7. Exploración de repositorios con modelos de clasificación de imágenes

Se realizó una revisión de implementaciones disponibles en Github para clasificación de imágenes, que tuvieran relación con detección de objetivos pequeños. Los repositorios se exploraron a partir de las publicaciones generadas a partir de ellos, la posibilidad de trabajar con dataset pequeños, y de trabajar con pequeños elementos en las imágenes.

Se encontró es que en la mayoría de artículos y, en general, en la literatura académica, utilizaban *YOLOv* (como ya se mencionado anteriormente). En particular, *YOLOv5* es una versión reciente de la familia de modelos YOLO (*You Only Look Once*), desarrollada por *Ultralytics*. Esta versión incluye mejoras significativas en la detección de objetos pequeños, incluyendo una arquitectura optimizada, integración de transformadores y una mejor detección multiescala. Estas mejoras permiten lograr una mayor precisión y mejor velocidad, en especial en aplicaciones en tiempo real.

Así, las principales implementaciones en GitHub para clasificación de imágenes relacionadas con el problema de la detección de objetos pequeños:

1. Yolov11-Easy-Use-Object-Detection-Segmentation-Classification

- **Repositorio:** AISoltani/Yolov11-Easy-Use-Object-Detection
- **Descripción:** Esta implementación tiene una interfaz sencilla para utilizar YOLOv11 en tareas de detección, segmentación y clasificación. Incluye mejoras en la arquitectura para una mejor detección de objetos pequeños y permite entrenar modelos personalizados de forma eficiente.

2. Yolo-v11-Object-Detection-Tracking

- **Repositorio:** Stormynova/Yolo-v11-Object-Detection-Tracking
- **Descripción:** Este proyecto combina la detección y el seguimiento de objetos en tiempo real utilizando YOLOv11. Está optimizado para detectar objetos domésticos pequeños. Así, da estimaciones de distancia y ángulo, además de una interfaz web para visualización en vivo.

3. YOLOs-CPP

- **Repositorio:** YOLOs-CPP
- **Descripción:** Es una implementación en C++ que permite integrar modelos desde YOLOv5 hasta YOLOv11 en aplicaciones utilizando *ONNX Runtime* y *OpenCV*. Es ideal para desarrolladores que buscan rendimiento en tiempo real en entornos de producción.

4. small-object-detection

- **Repositorio:** [erbayat/small-object-detection](#)
- **Descripción:** Este repositorio se enfoca en la evaluación de diferentes variantes de YOLOv11 para la detección de objetos pequeños. Incluye varias comparativas de latencia y rendimiento, así como la integración de técnicas como SAHI (*Sliced Aided Hyper Inference*) con el fin de mejorar la detección de objetos diminutos.

5. QueryDet

- **Repositorio:** [ChenhongyiYang/QueryDet-PyTorch](#)
- **Descripción:** QueryDet introduce un mecanismo de consulta en cascada. Esto permite acelerar la detección de objetos pequeños en imágenes de alta resolución. Este repositorio usa una estrategia de dos etapas para localizar objetos pequeños de manera eficiente, lo que mejora la precisión y reduce el costo computacional.

6. Small-Object-Detection-with-YOLO

- **Repositorio:** <https://github.com/hamzagorgulu/Small-Object-Detection-with-YOLO>
- **Descripción:** Propone una adaptación de YOLOv5 para detección de objetos pequeños. Además, incluye varias modificaciones y añadidos que permiten visualizar los datos y evaluar el rendimiento del modelo.

7. yolov5-cbam

- **Repositorio:** <https://github.com/aash1999/yolov5-cbam>
- **Descripción:** Es una modificación de yolov5 para trabajar con objetos muy pequeños. Las adaptaciones que se realizan permiten optimizar el entrenamiento de los datos con un modelo modificado obtenido de la versión original de yolov5 y que está optimizado de tal forma que hace viable trabajar tanto en CPU como en GPU.

Así, luego de revisar esto, se procedió a buscar algunos recursos adicionales que estuvieran relacionados con el tema, ya que en la mayoría de la literatura consultada se hace referencia a herramientas extra, a parte de los repositorios de Github mencionados anteriormente. Algunas de las principales fueron las siguientes:

- **SAHI (Slicing Aided Hyper Inference)**: Es una biblioteca ligera para mejorar la detección de objetos pequeños por medio del procesamiento por segmentos. Es compatible con modelos YOLOv5 y posteriores.
- **TinyDet**: Es un modelo diseñado específicamente para detectar objetos pequeños con alta precisión y eficiencia computacional.

8. Seleccionar el repositorio a utilizar con el conjunto de imágenes

Se procedió a analizar los repositorios disponibles con la intención de elegir uno que se adapte de la mejor forma a la base de datos generada y, luego, establecer qué tan bien clasifica el problema planteado en el proyecto.

Así, primero se estudiaron las publicaciones recientes destacadas sobre *Detección de Objetos Pequeños* generadas a partir de los repositorios disponibles mencionados en la sección anterior. Estas, en su mayoría, han utilizado implementaciones de YOLOv, pero, en general, todas utilizan técnicas relacionadas con la detección de objetos pequeños. Estas investigaciones abarcan diversas aplicaciones, desde imágenes aéreas, hasta la agricultura de precisión o el análisis de entornos submarinos.

MASF-YOLO: Mejora de YOLOv11 para Visión desde Drones

- **Título**: *MASF-YOLO: An Improved YOLOv11 Network for Small Object Detection on Drone View*
- **Autores**: Liugang Lu, Dabin He, Congxiang Liu, Zhixiang Deng
- **Fecha**: 2025
- **Resumen**: Esta investigación propone MASF-YOLO, que es una versión mejorada de YOLOv11 diseñada para detectar objetos pequeños en imágenes capturadas por drones. Este proyecto introduce módulos como MFAM para agregar características multiescala, IEMA para atención eficiente y DASI para poder generar una integración adaptativa de dimensiones. En las pruebas realizadas con el conjunto de datos seleccionados, VisDrone2019, MASF-YOLO-s superó a YOLOv11-s en un 4.6 % en

mAP@0.5 y un 3.5 % en mAP@0.5:0.95, manteniendo así una eficiencia computacional notable.

Se puede consultar en: arxiv.org

YOLOv11-SDC: Detección en Imágenes de Sonar de Barrido Lateral

- **Título:** *Side-Scan Sonar Small Objects Detection Based on Improved YOLOv11*
- **Autor:** Xu, X.
- **Fecha:** 2025
- **Resumen:** Este estudio presenta YOLOv11-SDC, que es una adaptación de YOLOv11 para detectar objetos pequeños en imágenes de sonar de barrido lateral (SSS). Este proyecto introduce módulos como SF para extracción de características relevantes y CGAF para atención conjunta en dimensiones espaciales, de canal y píxel. En el conjunto de datos SIMD, se pudo observar que YOLOv11-SDC superó a versiones anteriores de YOLO en precisión, *recall* y mAP, lo cual comprueba su eficacia en entornos submarinos complejos.

Se puede consultar en: mdpi.com

PC-YOLO11s: Extracción Eficiente de Características para Objetos Pequeños

- **Título:** *PC-YOLO11s: A Lightweight and Effective Feature Extraction Method for Small Target Image Detection*
- **Autores:** No encontrados
- **Fecha:** 2025
- **Resumen:** Esta investigación introduce PC-YOLO11s, el cual incorpora un mecanismo de atención espacial y de coordenadas (CSA) para mejorar la detección de objetos pequeños. Se observó que, al usar este mecanismo con módulos como SPPF, el modelo logra una mejor fusión de información contextual y espacial, lo cual aumenta considerablemente la precisión sin incrementar significativamente la complejidad computacional.

Se puede consultar en: mdpi.com

YOLO-MECD: Detección de Cítricos Basada en YOLOv11

- **Título:** *YOLO-MECD: Citrus Detection Algorithm Based on YOLOv11*
- **Autores:** Yin, H.; Wei, Q.; Gao, Y.; Hu, H.; Wang, Y.
- **Fecha:** 2025

- **Resumen:** Este estudio presenta YOLO-MECD, que es una adaptación de YOLOv11 con la intención de mejorar la detección de cítricos en entornos agrícolas. Aunque el enfoque principal de la investigación no era exclusivamente la detección de objetos pequeños, sí es cierto que las técnicas implementadas mejoran la precisión en la identificación de frutas parcialmente ocultas o en condiciones de iluminación que pueden ser variables.

Se puede consultar en: mdpi.com

SAHI: Inferencia Hipersegmentada para Objetos Pequeños

- **Título:** *Slicing Aided Hyper Inference and Fine-tuning for Small Object Detection*
- **Autores:** Fatih Cagatay Akyon, Sinan Onur Altinuc, Alptekin Temizel
- **Fecha:** 2022
- **Resumen:** SAHI (que ya ha sido mencionado anteriormente) es un marco de inferencia que tiene la intención de mejorar la detección de objetos pequeños utilizando la segmentación de imágenes en partes más pequeñas para así poder realizar un análisis individual. Esta técnica ha demostrado aumentar la precisión promedio (AP) en modelos como FCOS, VFNet y TOOD. SAHI es compatible con Detectron2, MMDetection y YOLOv5, y su mayor ventaja es que está disponible públicamente en GitHub.

Se puede consultar en: arxiv.org

Small-Object-Detection-with-YOLO

- **Título:** *Modifying the neck of YOLO architectures with input partitioning for improving small object detection*
- **Autores:** Hamza Gorgulu, Onur Caki
- **Fecha:** 2022
- **Resumen:** Este proyecto trata los desafíos de la detección de objetos pequeños, como la pérdida de información espacial y la escasez de datos, mediante una versión modificada de YOLO optimizada para usar en dispositivos móviles y en tiempo real. Las mejoras incluyen capas adicionales, recortes aleatorios y división de la imagen de entrada.

Se puede consultar en: https://github.com/hamzagorgulu/Small-Object-Detection-with-YOLO/blob/main/project_report.pdf

HIC-YOLOv5: Improved YOLOv5 for Small Object Detection

- **Título:** *HIC-YOLOv5: Improved YOLOv5 For Small Object Detection*
 - **Autores:** Shiyi Tang, Yini Fang, Shu Zhang
 - **Fecha:** 2023
 - **Resumen:** Aquí se explica que la detección de objetos pequeños es un reto debido al alto costo computacional. Para resolverlo, se propone HIC-YOLOv5, una versión mejorada de YOLOv5 que incluye predicción adicional, un bloque de involution y el módulo de atención CBAM. Estas mejoras aumentan la precisión sin elevar significativamente el costo computacional que se debe utilizar.
- Se puede consultar en:** <https://arxiv.org/pdf/2309.16393v1.pdf>

Es claro que las publicaciones mencionadas anteriormente demuestran avances significativos en la detección de objetos pequeños utilizando YOLOv y técnicas complementarias. Ahora bien, estas investigaciones han abordado desafíos específicos en diversos entornos, desde imágenes aéreas hasta aplicaciones agrícolas y submarinas, todas con el mismo fin de mejorar la precisión y eficiencia de los modelos de detección, todas utilizan herramientas similares, entre las cuales se debe elegir la que mejor se aparte al conjunto de prueba generado en este proyecto.

En esta dirección, es necesario mencionar que, aparte de las publicaciones realizadas, se deben considerar otros aspectos en la selección del repositorio a utilizar. Para esto, es necesario volver a los objetivos del proyecto y recordar que la tarea de contabilizar partículas pequeñas en imágenes presenta un conjunto de retos técnicos muy específicos que, en su mayoría, ya se consideran en los objetivos y metodología planteada para este proyecto. Con esto en mente, se puede realizar un análisis de las ventajas, desventajas y la aplicabilidad específica al caso de cada uno de los repositorios mencionados con anterioridad.

Es importante establecer algunos criterios clave para seleccionar el modelo. Dado que el proyecto requiere contabilizar partículas pequeñas en imágenes, se deben priorizar modelos con las siguientes características:

- Alta capacidad de detección de **objetos pequeños y densamente ubicados**.
- Posibilidad de trabajar con **datasets limitados**.
- Flexibilidad para entrenar o ajustar modelos personalizados.
- Métricas de precisión alta a pesar del tamaño reducido que pueden tener los objetos.

8.1. Elección del modelo

Ahora bien, de todo lo anterior, se concluyó que la mejor combinación para este proyecto (basado en todo lo mencionado anteriormente, junto con la lectura de las publicaciones realizadas a partir de estos modelos) es utilizar **YOLOv5** haciendo uso de las implementaciones propuestas en Small-Object-Detection-with-YOLO [Görgülü y Çakı 2022] y yolov5-cbam [Tang, Fang y Zhang 2023]. Las razones son varias, entre ellas están las siguientes:

- YOLOv5 resulta ser muy ligero, lo cual hace que sea manejable con pocos datos y optimizado para objetos pequeños.
- Se espera optimizar el modelo para no requerir una gran infraestructura de cómputo y hacerlo compatible con datasets pequeños es decir, adaptarlo a las condiciones y recursos disponibles.
- Las implementaciones que se encuentran en dichos repositorios permitirán adaptar yolov5 a la baja capacidad computacional con la que se cuenta y realizar técnicas de particionado de datos y visualización de los mismos. Además, brindarán una mayor facilidad para visualizar los datos.

9. Experimentación y Enfoque Utilizado

El enfoque experimental utilizado parte de dos necesidades concretas: utilizar un bajo costo computacional y optimizar la detección de objetos muy pequeños (3 píxeles). Para abordar estas necesidades, se integró y adaptó una combinación de dos trabajos destacados en el ámbito de detección de objetos pequeños: el proyecto Small-Object-Detection-with-YOLO, disponible en: <https://github.com/hamzagorgulu/Small-Object-Detection-with-YOLO> y el modelo HIC-YOLOv5, disponible en: <https://github.com/aash1999/yolov5-cbam>. Ambos enfoques parten de la arquitectura YOLOv5, conocida por su eficiencia y velocidad, y proponen mejoras para abordar las limitaciones de detección de objetos diminutos. El sistema desarrollado se centró en optimizar el procesamiento completo, desde la preparación del dataset hasta la evaluación cuantitativa del modelo entrenado. Se generaron scripts personalizados que automatizan todo el ciclo experimental. La implementación completa se encuentra en el siguiente repositorio, que es un fork del repositorio original de Marvin Coto Jiménez: <https://github.com/glaurung112/tinyobjects-detection>. El proceso comienza con la instalación automatizada de dependencias, lo que garantiza un entorno homogéneo de ejecución. A continuación, se descarga el dataset (mencionado anteriormente) desde Kaggle, con posibilidad de filtrar las imágenes por cantidad mínima y máxima de partículas, generando un conjunto de entrenamiento representativo.

Luego, se realiza la generación de etiquetas en formato YOLO. Se producen archivos `.txt` compatibles con las estructuras de entrada del modelo. Posteriormente, se implementa un sistema de organización de datos que asegura que cada imagen cuente con su correspondiente etiqueta en una misma carpeta. Y luego, se validó que no existan imágenes sin etiquetas.

Además, se realizó una adaptación de lo propuesto en Small-Object-Detection-with-YOLO para realizar *input partitioning*, haciendo recortes aleatorios de las imágenes. Este se adaptó para generar archivos en formato `.png`. Este proceso aumenta la diversidad del dataset y mejora la capacidad del modelo para detectar partículas pequeñas, en particular aquellas que podrían perderse en resoluciones completas. También, se realiza un ajuste automático de las etiquetas a cada recorte.

Luego se dividió el conjunto en entrenamiento y validación en la proporción deseada (80/20 por defecto), y se automatizó la generación del archivo YAML necesario para indicar al modelo las rutas de los datos.

Además, para inspecciones cualitativas, se adaptó lo propuesto en el repositorio mencionado, para visualizar imágenes aleatorias con sus etiquetas usando `matplotlib`, evitando el uso de bibliotecas como `seaborn`. Este paso fue importante para garantizar la calidad del dataset antes del entrenamiento.

En cuanto al modelo, se empleó una versión modificada de YOLOv5 conocida como HIC-YOLOv5, la cual incluye tres mejoras fundamentales según el artículo *HIC-YOLOv5: Improved YOLOv5 For Small Object Detection*: un cuarto cabezal de predicción especializado en objetos pequeños, un bloque de involution entre el backbone y el neck que permite enriquecer la información de canal, y la integración del módulo de atención CBAM al final del backbone para destacar características espaciales y de canal relevantes. Estas modificaciones se utilizaron para mejorar la detección sin sacrificar la eficiencia computacional. Finalmente, para la etapa de evaluación, se adaptó la propuesta de Small-Object-Detection-with-YOLO para realizar detección, calcular métricas como precisión, recall y F1-score, lo cual resulta esencial para analizar el comportamiento del modelo en la detección de partículas pequeñas.

Este enfoque experimental no solo replica los hallazgos de los trabajos previos, sino que los adapta a un nuevo contexto y los automatiza, permitiendo llevar a cabo experimentos más rápidos, reproducibles y flexibles. Con esta metodología, se logró entrenar el modelo, evaluarlo de forma cuantitativa y cualitativa, y extraer conclusiones relevantes sobre los límites y su potencial en la detección de partículas.

10. Resultados Obtenidos

La evaluación del sistema se llevó a cabo mediante un análisis cuantitativo exhaustivo, basado en métricas estándar en tareas de detección de objetos: precisión, exhaustividad

(recall), F1-score y precisión. Para ello, se utilizó una versión modificada de lo que se propone en Small-Object-Detection-with-YOLO que automatiza la ejecución de la inferencia sobre el conjunto de validación y genera un conjunto de salidas que incluyen las imágenes anotadas y reportes en formato de texto. Esto realmente es una versión que también fue modificada a partir del modelo original propuesto por *YOLOv5* de *Ultralytics*.

La información que se obtiene a partir de esto resulta fundamental, dado que uno de los mayores retos en la detección de objetos pequeños es precisamente la baja respuesta de modelos de detección de objetos ante regiones de muy baja resolución espacial.

Durante las pruebas experimentales, primero se entrenó el modelo por 5 épocas con una resolución de entrada de 640x640 píxeles, un **batch size** de 4, y usando la arquitectura original de yolov5. Este primer entrenamiento no funcionó correctamente, ya que la unidad en la que se estaba realizando el proceso no contaba con la capacidad computacional suficiente para ejecutar el entrenamiento y se estaba apagando y reiniciando constantemente. Por tanto, se procedió a ejecutar el entrenamiento desde Google Colab, para poder acceder a una unidad de mayor capacidad. Sin embargo, al realizar el entrenamiento, las tres métricas (Precisión, Recall y F1-Score) estaban en 0. La conclusión es que el entrenamiento se realizó durante muy pocas épocas y el modelo no estaba detectando ningún punto en las imágenes.

Posteriormente, se realizó el entrenamiento durante 20 épocas, con los mismo parámetros, 640x640 píxeles y **batch size** de 4. Sin embargo, los valores de las métricas seguían siendo muy bajos, aunque ahora el modelo sí identificaba algunos puntos. No obstante, todos eran falsos positivos. De igual forma, el modelo no estaba siendo entrenado con suficientes épocas. A pesar de entender que la cantidad de épocas era el factor determinante, no era tan sencillo resolver el problema, ya que el entrenamiento con estas 20 épocas ya había tardado varias horas en realizarse (alrededor de 8 horas) y un entrenamiento con más épocas iba a ser posiblemente una tarea de varios días.

Antes de continuar con esto, se decidió alterar los demás parámetros y se notó que al cambiar a 320x320 píxeles y **batch size** de 16 y volver a entrenar el modelo con 20 épocas (otras 8 horas aproximadamente) se empezaron a obtener *True Positives*. Luego, se empezó a utilizar la arquitectura modificada *yolov5m-cbam-involution.yaml* propuesta en HIC-YOLOv5 y, finalmente, se empezó a obtener una cantidad significativa de *True Positives*. Estas fueron excelentes noticias, ya que confirmaban la necesidad de realizar el entrenamiento grande para obtener finalmente buenos resultados de una forma que valiera la pena el esfuerzo que se tendría que realizar para entrenar el modelo durante varios días. Finalmente, se entrenó el modelo por 80 épocas con una resolución de entrada de 320x320 píxeles, un **batch size** de 16, y usando la arquitectura modificada propuesta en HIC-YOLOv5.

Las métricas obtenidas fueron:

- **Precisión:** 0.21 %
- **Recall:** 0.43 %
- **F1-Score:** 0.31 %
- **Promedio IoU:** 0.65 %

Estos resultados reflejan un equilibrio entre la capacidad del modelo para detectar las partículas presentes (recall) y su habilidad para evitar falsas detecciones (precisión). En particular, el F1-Score obtenido indica que el modelo logra un cierto grado de balance, todo esto (por supuesto) considerando que el modelo solo se entrenó durante 80 épocas. Ahora bien, cabe destacar que, por esto, se propone en el repositorio del proyecto entrenar el modelo con 150 épocas como punto de partida. Para esa cantidad de épocas se precide un comportamiento mucho mejor y una mayor aproximación a la detección correcta de objetos pequeños con mayores valores de las métricas. Esto se puede predecir ya que, como se observó durante el proceso de experimentación, la detección de objetos por parte del modelo mejoró mucho al aumentar la cantidad de épocas de 20 a 80, y 150 es casi el doble de la cantidad utilizada en este proyecto hasta el momento.

Para finalizar, un aspecto destacado fue la robustez del modelo frente a escenarios extremos. Se implementaron controles para evitar errores comunes como la división por cero, especialmente críticos en imágenes con pocas o ninguna detección cuando el modelo no estaba bien entrenado, pero aún así se querían observar los resultados obtenidos. Además, se incluyó una salida visual para cada imagen procesada, con las predicciones anotadas (esto se logró a partir de lo que se propone en los repositorios mencionados) lo cual facilita la edición y revisión manual. En general, esta fase de evaluación demostró que el sistema es capaz de mantener un rendimiento relativamente elevado (considerando las circunstancias) y estable, que es su principal ventaja, ya que se puede continuar el entrenamiento y las pruebas con más épocas si se cuenta con una mayor cantidad de tiempo. Sin embargo, no se puede concluir que los resultados obtenidos sean reproducibles realmente, debido al gran problema encontrado con el tiempo de ejecución del entrenamiento conforme aumenta el número de épocas, como se explica a continuación. Sin embargo, incluso con una cantidad bastante reducida de épocas en el entrenamiento el modelo presentó resultados no desfavorables.

11. Estudio de Límites de Detección

Para caracterizar los límites prácticos del sistema desarrollado, se puede recurrir a las pruebas que se realizaron, las cuales permiten observar el comportamiento ante diferentes

condiciones de detección. Se consideraron dos factores importantes: tamaño de los objetos y épocas de entrenamiento.

Con respecto al impacto del **tamaño**, se puede mencionar que en todos los experimentos realizados se lanzaron los *warnings* incorporados por *YOLOv5*, en los cuales se indica que los objetos son demasiado pequeños. Incluso en el modelo modificado especialmente para la detección de objetos pequeños se pudieron observar estas advertencias. Esto es bastante claro y tiene mucho sentido, ya que en este proyecto se estaba trabajando con *puntos* de no más de 3 píxeles en la imagen. Probablemente cuando se aborda el problema de la detección de objetos pequeños en imágenes no se suelen considerar este tipo de dimensiones tan extremas. No obstante, el modelo logró una respuesta bastante adecuada y se espera continuar con la evaluación del mismo durante más épocas.

Sobre esto último, la cantidad de épocas de entrenamiento se reconoce como la principal limitación de este proyecto y como una de las mayores barreras que se tienen al trabajar con este tipo de modelo de detección en imágenes. La capacidad computacional requerida para adaptar el modelo es muy alta, a pesar de que se realizaron diversas optimizaciones y se recurrió a diferentes adaptaciones del modelo realizadas por diferentes personas y proyectos. Incluso con todo este esfuerzo, el tiempo requerido para entrenar el modelo fue demasiado alto, en escala de días, por lo cual no fue posible realizar varios experimentos y queda la pregunta abierta de si los resultados obtenidos son reproducibles realmente o fueron un evento más ocasional.

Adicionalmente, se puede mencionar que hay una gran limitación por parte de la versión original de *YOLOv5* para detectar objetos pequeños sin ningún tipo de modificación (su capacidad es prácticamente nula). Se puede afirmar con certeza que, al menos en los experimentos realizados en este proyecto, jamás se habrían obtenidos métricas mayores a 0 sin utilizar el modelo adaptado propuesto en los repositorios mencionados. En general, se concluye que:

- El modelo original de *YOLOv5* presenta una pérdida significativa de rendimiento cuando el radio de las partículas es muy pequeño
- La arquitectura mejorada *HIC-YOLOv5* ofrece muchas ventajas claras frente al modelo base, especialmente en escenarios críticos, menores a 3 píxeles. Sin embargo, sigue teniendo la limitación de la capacidad computacional, ya que requiere de mucho tiempo para ejecutar un entrenamiento con un número elevado de épocas.

12. Conclusiones

El presente proyecto permitió evidenciar las dificultades de la detección de objetos extremadamente pequeños en imágenes, con un tamaño de apenas 3 píxeles. A lo largo del

proceso, se logró constatar que la arquitectura original de YOLOv5 no posee la capacidad necesaria para detectar este tipo de objetos sin realizar modificaciones al modelo.

La integración de la arquitectura modificada propuesta por HIC-YOLOv5 (que incluye un cabezal adicional para objetos pequeños, un bloque de involution y la atención CBAM) demostró ser muy útil y efectiva. Esto (entre otros aspectos de preprocesamiento de los datos) fue lo que permitió alcanzar métricas aceptables (precisión de 0.21 %, recall de 0.43 %, F1-score de 0.31 % y una IoU promedio de 0.65 %) incluso con una cantidad limitada de épocas de entrenamiento (hasta 80).

Se entendió que la combinación de estrategias propuestas en los repositorios *Small-Object-Detection-with-YOLO* y *YOLOv5-CBAM*, especialmente la partición de imágenes (input partitioning) y la visualización de etiquetas, fue esencial para construir un sistema robusto y adaptado. Así, el sistema desarrollado automatiza estas tareas, permitiendo no solo preparar los datos, sino también entrenar y evaluar el modelo de forma reproducible, simplificando mucho la experimentación futura.

No obstante, una limitación muy clara durante el desarrollo fue el tiempo de entrenamiento requerido. Aunque se lograron resultados aceptables con 80 épocas, este entrenamiento tomó varios días y aumentarlo implicaría mucho tiempo, lo cual complica una exploración más profunda con los recursos que se cuentan actualmente. Aun así, el desempeño logrado demuestra que el enfoque es prometedor y que con mayor capacidad computacional podrían alcanzarse métricas superiores.

13. Recomendaciones Futuras

- **Incrementar la cantidad de épocas:** Se recomienda realizar entrenamientos con al menos 150 épocas. Dado que el desempeño mejora notablemente con el número de épocas, esto podría aumentar la precisión y recall del modelo de forma muy significativa.
- **Evitar sobreajuste:** Aunque es importante aumentar las épocas, no se recomienda incrementar demasiado ninguno de los parámetros (por ejemplo, el tamaño del batch), ya que esto podría llevar al sobreajuste, dada la naturaleza del conjunto de datos.
- **Evaluar otras variantes de YOLO:** Existen versiones más recientes de la familia YOLO (como YOLOv11), así como modelos ligeros basados en transformers. Sería útil comparar su desempeño con el de HIC-YOLOv5 para tareas de detección de objetos diminutos.

- **Ampliar la automatización:** El sistema actual ya automatiza desde la creación del dataset hasta la evaluación. Se podría extender para empaquetarlo como una herramienta usable por terceros que facilite su integración en otros sistemas.
- **Ampliar pruebas sobre otras clases de objetos pequeños:** Si bien el proyecto se enfocó en puntos blancos sobre fondo negro, sería muy interesante evaluar el desempeño del sistema ante otros tipos de partículas u objetos pequeños con características distintas (colores o formas).
- **Conseguir mayor capacidad computacional:** En algún momento del futuro se podría intentar conseguir mayor capacidad computacional para seguir entrenando el modelo con parámetros mucho mayores (por medio de apoyo externo o comprando acceso a alguna unidad de Google Colab), ya que este es un proyecto prometedor y podría generar resultados muy interesantes si se involucra una mayor cantidad de recursos.

Referencias

- Chen, C., M. Y. Liu, O. Tuzel y J. Xiao. 2017. «R-CNN for small object detection». En *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20–24, 2016, Revised Selected Papers, Part V 13*, 214–230. Springer International Publishing.
- Görgülü, Hamza y Onur Çakı. 2022. «Modifying the neck of YOLO architectures with input partitioning for improving small object detection». *COMP541Deep Learning, Fall 2022*.
- Ma, Z., L. Yu y A. B. Chan. 2015. «Small instance detection by integer programming on object density maps». En *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3689–3697.
- Pang, Jiangmiao, Cong Li, Jianping Shi, Zhihai Xu y Huajun Feng. 2019. «R2-CNN: fast Tiny object detection in large-scale remote sensing images». *IEEE Transactions on Geoscience and Remote Sensing* 57 (8): 5512–5524.
- Shinya, Yosuke. 2023. «BandRe: Rethinking band-pass filters for scale-wise object detection evaluation». En *2023 18th International Conference on Machine Vision and Applications (MVA)*, 1–5. IEEE.
- Tang, Shiyi, Yini Fang y Shu Zhang. 2023. «HIC-YOLOv5: Improved YOLOv5 For Small Object Detection». *Heriot-Watt University, Hong Kong University of Science and Technology, Ocean University of China*.
- Yu, X., Z. Han, Y. Gong, N. Jan, J. Zhao, Q. Ye y H. Shi. 2020. «The 1st tiny object detection challenge: Methods and results». En *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, 315–323. Springer International Publishing.
- Zou, Zhengxia, Keyan Chen, Zhenwei Shi, Yuhong Guo y Jieping Ye. 2023. «Object detection in 20 years: A survey». *Proceedings of the IEEE* 111 (3): 257–276.