Universidad del Valle de Guatemala

Facultad de ingeniería

Data Science

Catedrático: Luís Furlán

Proyecto 2.

Resultados Iniciales

Nelson Eduardo García Bravatti 22434

Joaquín André Puente Grajeda 22296

José Antonio Mérida Castejón 201105

Guatemala, noviembre de 2025

Selección de Modelos

Introduccion

El problema que buscamos resolver, al igual que la investigacion previa se pueden encontrar en este link. Para este proyecto, decidimos realizar una investigación sobre cinco algoritmos diferentes para resolver problemas de vision computadoras que involucren tanto identificación de objetos como clasificación. Optamos primordialmente por algoritmos / modelos que nos permitan identificar y clasificar objetos dentro de un pipeline completo ya que es nuestra primera experiencia con visión por computadora. Esto nos simplifica un poco la implementación de los modelos, y nos permite realizar un análisis adecuado sobre modelos más "simples". Posteriormente, elegimos tres modelos para implementar y realizar comparaciones.

1. YOLOv8

Tipo: Detector de una sola etapa (Ultralytics, 2023)
Razones de uso:

- Es uno de los modelos más rápidos y precisos actualmente, ideal para detección en tiempo real.
- Ofrece una API muy sencilla para entrenar y exportar el modelo a distintos formatos (ONNX, TensorRT, CoreML).
- Permite realizar no solo detección de objetos, sino también segmentación, estimación de pose y seguimiento.
- Cuenta con una comunidad activa y documentación extensa, lo que facilita su implementación.

SSD es clave en la historia del diseño eficiente de redes convolucionales para visión. Demostró que los detectores podían ser compactos, entrenarse con un único flujo y ejecutarse en tiempo real sin sacrificar demasiado la precisión. Su estructura inspiró arquitecturas posteriores como RetinaNet y EfficientDet.

Se elige por su velocidad y simplicidad, ideal para tareas que requieren detección en tiempo real o donde los recursos de hardware son limitados. Aun siendo más ligero que otros modelos, conserva una buena capacidad de generalización y se adapta bien a implementaciones en dispositivos con GPU moderada o incluso CPU.

2. Faster R-CNN

Tipo: Detector de dos etapas (con red de propuestas de regiones, RPN)
Razones de uso:

- Tiene alta precisión y excelente desempeño al localizar y clasificar objetos.
- Es un estándar de referencia en investigación y comparación académica.

- Funciona muy bien con objetos pequeños o solapados, donde otros modelos tienden a fallar.
- Se puede adaptar con distintas redes base (ResNet, Swin Transformer, etc.) para ajustar precisión y velocidad.

Faster R-CNN consolidó el paradigma de detección en dos etapas, donde una primera red propone regiones candidatas y una segunda las refina y clasifica. Este enfoque sigue siendo un estándar para tareas que requieren precisión por encima de la velocidad, como análisis médico, inspección industrial o imágenes satelitales. Además, su diseño modular permite sustituir los backbones (ResNet, VGG, Swin Transformer) y experimentar con distintas funciones de pérdida y estrategias de anclaje.

Se selecciona por su alta precisión y capacidad de detección en escenarios complejos, especialmente con objetos pequeños o parcialmente solapados. Es el modelo de referencia en la literatura científica y se prefiere cuando la velocidad no es la principal limitante (por ejemplo, en investigación biomédica, inspección industrial o análisis de laboratorio).

3. DETR / RT-DETR

Tipo: Detector basado en transformadores (Facebook AI, 2020; versión rápida RT-DETR, 2023)
Razones de uso:

- Sustituye los métodos tradicionales de propuesta y supresión de no-máximos por un enfoque de atención global.
- Entrena de forma más limpia y directa, sin necesidad de definir anclas ni parámetros manuales.
- La versión RT-DETR logra rendimiento en tiempo real manteniendo la arquitectura basada en transformadores.
- Es especialmente útil para escenas complejas con múltiples objetos o relaciones espaciales.

DETR no solo ofrece un nuevo modelo, sino un nuevo marco conceptual para la visión por computadora: la detección sin heurísticas. Aunque su entrenamiento es más exigente (convergencia lenta y gran demanda de datos), su simplicidad arquitectónica lo hace atractivo para investigaciones sobre interpretabilidad, atención visual y modelado de escenas. Sus derivados, como Deformable DETR o Conditional DETR, optimizan la convergencia y demuestran la flexibilidad del enfoque.

Se elige por su enfoque moderno y conceptual, que integra visión y atención global en una sola arquitectura. Es ideal en contextos donde interesa capturar relaciones espaciales complejas entre múltiples objetos o donde la interpretabilidad del proceso de atención es valiosa (por ejemplo, análisis de escenas o investigación cognitiva computacional).

4. RetinaNet

Tipo: Detector de una sola etapa
Razones de uso:

- Introduce la Focal Loss, que mejora el desempeño en conjuntos de datos con fuerte desequilibrio de clases (mucho fondo y pocos objetos).
- Ofrece un equilibrio entre simplicidad, precisión y velocidad.
- Permite usar distintas redes base y ajustarse a distintos niveles de complejidad.

Muestra cómo una función de pérdida bien diseñada puede igualar el desempeño de arquitecturas más complejas. Su arquitectura sigue siendo sencilla y elegante, combinando una red base (por ejemplo, ResNet) con una Feature Pyramid Network (FPN) que permite detectar objetos en múltiples escalas. Por ello, RetinaNet es un excelente punto medio entre el rendimiento de dos etapas y la simplicidad de una sola.

Se elige por su equilibrio entre precisión y eficiencia. Ofrece resultados comparables a Faster R-CNN pero con menor complejidad computacional. Es especialmente útil cuando el dataset presenta clases muy desbalanceadas (por ejemplo, detección de insectos u organismos poco frecuentes frente a mucho fondo vacío).

5. EfficientDet

Tipo: Detector híbrido (Google Brain, 2020)
Razones de uso:

- Utiliza una escala compuesta que optimiza simultáneamente la profundidad, el ancho y la resolución del modelo.
- Logra alta eficiencia computacional sin sacrificar precisión.
- Es ideal para dispositivos con recursos limitados, como sistemas embebidos o móviles.

EfficientDet ejemplifica una tendencia actual en la visión por computadora: el diseño de modelos escalables y sostenibles. En lugar de buscar únicamente mayor precisión, prioriza la eficiencia energética y la adaptabilidad, lo cual es fundamental para el despliegue en dispositivos móviles, drones o sistemas embebidos. Su arquitectura ha influido en muchos modelos ligeros posteriores.

Se elige por su alta eficiencia computacional y escalabilidad. Es una excelente opción cuando se busca implementar detección en dispositivos móviles, drones o sistemas embebidos, donde el consumo de energía, la memoria y el tiempo de inferencia son factores limitantes. Se elige para investigación,

Motivos de selección de los 3 seleccionados

Se eligieron los modelos Faster R-CNN, RetinaNet y SSD por representar tres enfoques complementarios dentro de la detección de objetos y permitir un análisis equilibrado entre precisión, eficiencia y complejidad. Faster R-CNN fue seleccionado por su alta exactitud y estabilidad, al ser un modelo de dos etapas ampliamente validado en la literatura, ideal como referencia base para comparar resultados. RetinaNet se eligió por su equilibrio entre precisión y velocidad, gracias a la introducción de la Focal Loss, que mejora el rendimiento en datasets con clases desbalanceadas o predominio de fondo. Finalmente, SSD fue escogido por su simplicidad y rapidez, al realizar detección en una sola pasada con un diseño multiescala eficiente, lo que lo convierte en una opción práctica para aplicaciones en tiempo real o con recursos computacionales limitados. En conjunto, estos tres modelos permiten evaluar el compromiso entre rendimiento, costo computacional y aplicabilidad en distintos contextos de detección.

Referencias

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *You Only Look Once: Unified, Real-Time Object Detection.* *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.* *Advances in Neural Information Processing Systems (NeurIPS)*, 28. DOI: 10.48550/arXiv.1506.01497

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). *End-to-End Object Detection with Transformers (DETR).* *European Conference on Computer Vision (ECCV)*. DOI: 10.1007/978-3-030-58452-8_13

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). *Focal Loss for Dense Object Detection. Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2980–2988. DOI: 10.1109/ICCV.2017.324

Tan, M., Pang, R., & Le, Q. V. (2020). *EfficientDet: Scalable and Efficient Object Detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10781–10790. DOI: 10.1109/CVPR42600.2020.01080