

---

This is the **published version** of the bachelor thesis:

Manresa Martínez, Roberto; Benavente Vidal, Robert, dir. Detección de piscinas ilegales en imágenes aéreas utilizando técnicas de Deep Learning. 2023. (Enginyeria de Dades)

---

This version is available at <https://ddd.uab.cat/record/281556>

under the terms of the  license

# Detección de piscinas ilegales en imágenes aéreas utilizando técnicas de Deep Learning

Roberto Manresa Martínez

**Resumen**— El objetivo de este trabajo es utilizar técnicas de Deep Learning para detectar piscinas ilegales en imágenes aéreas, mejorando la eficiencia y reduciendo los costos asociados a su identificación. Para ello, se ha realizado fine-tuning del modelo Transformer DETR utilizando un conjunto de datos de imágenes aéreas de Cataluña. Los resultados demuestran la efectividad del sistema en la detección precisa de piscinas, aunque se enfrenta a desafíos al detectar piscinas parcialmente así como piscinas de menor tamaño. Este trabajo contribuye al avance de la detección de piscinas ilegales ofreciendo una alternativa prometedora para optimizar los procesos de identificación y control de estas infraestructuras.

**Palabras clave**— Detección de Objetos, Detección de piscinas ilegales, Transformer, DETR, Piscinas, Deep Learning, Ortoimágenes, Cataluña

**Abstract**— The objective of this study is to utilize Deep Learning techniques to detect illegal swimming pools in aerial images, aiming to enhance efficiency and reduce associated identification costs. To achieve this, a fine-tuning process is performed on the Transformer DETR model using a dataset of aerial images from Catalonia. The results demonstrate the system's effectiveness in accurately detecting swimming pools, although it faces challenges when detecting partially as well as smaller-sized ones. This research contributes to the advancement of illegal swimming pool detection by offering a promising alternative for optimizing the identification and monitoring processes of such infrastructures.

**Keywords**— Object Detection, Illegal Pool Detection, Transformer, DETR, Pools, Deep Learning, Orthoimages, Catalonia

## 1 INTRODUCCIÓN

EN los últimos años, se ha producido un aumento del número de piscinas construidas sin licencia en España. Estas piscinas ilegales suponen un riesgo para la seguridad pública y el equilibrio ecológico, ya que si no se mantienen adecuadamente puede convertirse en un caldo de cultivo para bacterias y otros contaminantes que pueden afectar la calidad del agua y la salud pública. La detección manual de estas piscinas ilegales es una tarea costosa y lenta, que requiere una gran cantidad de recursos humanos y una revisión periódica de las imágenes aéreas obtenidas de todo el país, España tiene una superficie de 505.990 km<sup>2</sup> y

contiene 1.266.000 piscinas registradas [1], lo que convierte esta tarea repetitiva en monumental. Por ello, considero necesario desarrollar métodos automáticos que permitan identificar piscinas, ya que son mucho más rápido y baratos que las personas y se puede ejecutar en paralelo sobre distintas partes de España, ahorrando miles de euros en gastos y a su vez recaudando millones en multas y nuevos impuestos. Este método automático encontraría piscinas, comprobaría si están registradas y si no lo están, enviaría un aviso a la autoridad correspondiente para su revisión manual por un experto.

### 1.1. Estado del arte

La detección de objetos en imágenes es una tarea de visión por computador en la que el objetivo es detectar y localizar objetos de interés en una imagen. La tarea implica identificar la posición y los límites de los objetos en una imagen y clasificar los objetos en diferentes categorías.

---

• E-mail de contacto: roberto.manresa@autonoma.cat  
 • Trabajo tutorizado por: Robert Benavente Vidal (Departamento de Ciencias de la Computación)  
 • Curs 2022/23

Los métodos más avanzados del estado del arte se pueden categorizar en Convolutional Neural Network(CNN) de dos tipos principales: métodos de una etapa y métodos de dos etapas. Los métodos de una etapa priorizan la velocidad de inferencia y los modelos de ejemplo incluyen YOLO [2], SSD [3] y RetinaNet [4]. Los métodos de dos etapas priorizan la precisión de detección y los modelos de ejemplo incluyen Faster R-CNN [5], Mask R-CNN [6] y Cascade R-CNN [7].

Si nos centramos en el problema de la detección de piscinas, podemos ver que también sigue el mismo patrón y todos los trabajos encontrados utilizan CNNs. Por ejemplo, Biel Stela ha desarrollado un método para detectar piscinas ilegales en su Trabajo de Fin de Máster [8] utilizando Yolo. También, un software desarrollado por Capgemini ha permitido descubrir más de 20.000 piscinas privadas no declaradas en Francia [9]. Además, en Redlands, California, utilizando imágenes satélites, se extrajeron datos de entrenamiento etiquetando manualmente piscinas en Redlands y con ellos se entrenó un SSD que detectaba piscinas [10]. Estos trabajos demuestran la viabilidad y utilidad de los métodos automáticos para la detección de piscinas ilegales.

Como ya se ha mencionado, la detección de objetos se está llevando a cabo mediante el uso de CNNs, sin embargo, en otras áreas como el procesamiento del lenguaje natural (NLP), los Transformers están demostrando ser muy exitosos y están logrando mejores resultados. Una de las grandes ventajas de los Transformers es su capacidad para el aprendizaje por transferencia, esto permite que corporaciones inviertan grandes cantidades de dinero en entrenar a un modelo y que el resto de la comunidad pueda hacer fine-tuning y obtener excelentes resultados. Uno de los Transformers entrenados para la detección de objetos es DETR(DEtection TRansformer) de Facebook [11], un modelo entrenado con ImageNet [12] que ha obtenido muy buenos resultados y que en muchos casos superan a los otros métodos mencionados anteriormente como explican en su artículo. Viendo el potencial de los Transformers y su poco uso en el área de la detección de objetos, se ha decidido explorar su uso en este trabajo.

## 2 OBJETIVOS

El objetivo general que se quiere conseguir es, diseñar e implementar un sistema de detección de piscinas ilegales en imágenes aéreas mediante técnicas de deep learning, logrando alta precisión y eficiencia, ya que se planea usar el modelo para la identificación de piscinas en países o regiones enteras. Para ello, se plantean los siguientes objetivos principales:

- Encontrar y localizar piscinas ilegales en imágenes aéreas obtenidas por satélites, verificando su registro y notificar para su posterior revisión manual por un experto.
- Explorar el uso de Transformers como método para la detección de piscinas y compararlo con otros métodos basados en redes convolucionales, analizando sus ventajas e inconvenientes.

- Realizar una revisión bibliográfica de los modelos propuestos para la detección de piscinas ilegales o problemas similares, y compararlos con el modelo desarrollado en este trabajo para situar el trabajo en el estado del arte y valorar sus aportaciones y mejoras.

Además, se busca que el sistema de detección sea robusto ante diferentes condiciones climáticas e iluminación, y que cumpla con los objetivos principales planteados para lograr una detección precisa y eficiente de las piscinas.

También se plantean objetivos secundarios que sería interesante realizar:

- Desarrollar un sistema de notificación automático que envíe alertas por correo electrónico cuando se detecten nuevas piscinas ilegales.
- Explorar el uso de técnicas de aprendizaje activo para mejorar la eficiencia del sistema de detección de piscinas ilegales, seleccionando las imágenes más relevantes para su revisión manual.
- Experimentar con diferentes arquitecturas de redes neuronales y técnicas de preprocesamiento de imágenes para mejorar la precisión del modelo de detección de piscinas ilegales.

## 3 METODOLOGÍA

En esta sección se describe la metodología empleada en el trabajo. Se utiliza el método ágil para el desarrollo del proyecto, que se caracteriza por su flexibilidad y adaptación continua a los cambios y necesidades del proyecto. El trabajo se divide en sprints de dos semanas, durante los cuales se planifican, ejecutan y revisan las tareas. A continuación se presenta el plan de trabajo detallado:

1. Realizar una revisión bibliográfica sobre las técnicas de deep learning aplicadas a la detección de objetos en general, y más específicamente las usadas para la detección de piscinas.
2. Recolectar y preparar los datos necesarios para entrenar y evaluar el modelo. Esto incluye imágenes aéreas del territorio y las ubicaciones de las piscinas registradas en dicho territorio.
3. Adaptar el modelo DETR utilizando su artículo [11] y repositorio en GitHub [13], donde proporcionan información esencial del modelo para usarlo, prepararlo y entrenarlo. Se utilizará un modelo preentrenado y se preparará el entorno de trabajo en un servidor proporcionado por el CVC(Computer Vision Center) para el entrenamiento.
4. Realizar un proceso de re-entrenamiento, aprovechando el transfer learning, mediante la técnica de fine-tuning para optimizar la detección de objetos en el conjunto de datos utilizado. Se explorarán diferentes parámetros y combinaciones para optimizar el modelo para la tarea.
5. Evaluar el modelo utilizando diferentes métricas objetivas para comparar su rendimiento con otros modelos

existentes que utilicen otras técnicas de deep learning. Se analizarán los resultados obtenidos y se discutirán las ventajas y desventajas del método propuesto, identificando posibles retos y limitaciones del sistema y proponiendo posibles mejoras o líneas futuras de investigación.

6. Se redactarán varios informes a lo largo del proyecto, incluyendo el informe inicial, dos informes de progreso y el informe final. El informe final incluirá una descripción detallada de la metodología utilizada, los resultados obtenidos y su análisis y discusión. También se identificarán posibles mejoras y líneas futuras de investigación. Además, se preparará una presentación para exponer los aspectos más relevantes del trabajo.

Para la ejecución del proyecto, se utilizarán diversas herramientas y técnicas. El lenguaje de programación Python será utilizado en todo el proceso, incluyendo la recolección y preparación de datos, debido a su idoneidad para trabajar con deep learning. Además, se empleará la biblioteca PyTorch [14], por su compatibilidad con DETR y porque ofrece una amplia variedad de herramientas y módulos para simplificar la construcción de modelos de deep learning. Para el control de versiones, se utilizará la plataforma GitHub [15], que es reconocida y confiable.

## 4 PLANIFICACIÓN

Para planificar el proyecto se ha elaborado un diagrama de Gantt dónde se distribuyen los 4 meses disponibles entre 6 tareas del proyecto, que se han subdividido en 19 subtareas. Todo esto se puede observar en la Figura 5 en el apéndice. Cabe destacar que este es un plan tentativo y está sujeto a cambios según el progreso del proyecto.

En cuanto a la planificación del trabajo, se realizan sprints de dos semanas y se pueden establecer las siguientes subtareas detalladas con el tiempo dedicado en horas al lado de cada subtarea:

### Sprint 1 (30h)

- Subtarea 1 (10h): Realizar una revisión bibliográfica sobre las técnicas de deep learning aplicadas a la detección de objetos, especialmente las usadas en imágenes aéreas.
- Subtarea 2 (15h): Recolectar los datos necesarios para empezar a entrenar y evaluar el modelo, por un lado, las imágenes aéreas del territorio y por el otro las ubicaciones de cada una de las piscinas registradas en dicho territorio.
- Subtarea 3 (5h): Redacción del informe inicial.

### Sprint 2 (30h)

- Subtarea 4 (15h): Preprocesar los datos obtenidos en el sprint anterior para poder utilizarlos en el entrenamiento del modelo.
- Subtarea 5 (15h): Preparar el entorno de trabajo y las herramientas necesarias para el desarrollo del proyecto.

### Sprint 3 (50h)

- Subtarea 6 (20h): Adaptar el modelo preentrenado a al problema planteado.
- Subtarea 7 (30h): Entrenar el modelo con el conjunto de datos preparado utilizando de fine-tuning.

### Sprint 4 (40h)

- Subtarea 8 (30h): Realizar pruebas y ajustes en el modelo para conseguir un mejor rendimiento en la detección de piscinas.
- Subtarea 9 (10h): Redacción del informe de progreso 1.

### Sprint 5 (54h)

- Subtarea 10 (20h): Recolectar más datos para entrenar y evaluar el modelo.
- Subtarea 11 (34h): Entrenar el modelo con una mayor cantidad de datos y arreglar bugs.

### Sprint 6 (40h)

- Subtarea 12 (15h): Evaluar el modelo utilizando diferentes métricas objetivas para comparar el rendimiento con otros modelos existentes.
- Subtarea 13 (15h): Comparar el rendimiento del modelo con otros existentes en el mismo ámbito.
- Subtarea 14 (10h): Redacción del informe de progreso 2.

### Sprint 7 (35h)

- Subtarea 15 (20h): Analizar los resultados obtenidos y discutir las ventajas y desventajas del método propuesto.
- Subtarea 16 (15h): Redacción del informe final.

### Sprint 8 (21h)

- Subtarea 17 (10h): Preparar la presentación oral y ensayarla.
- Subtarea 18 (10h): Creación de póster y dossier.
- Subtarea 19 (1h): Presentar el TFG.

## 5 DESARROLLO

En este apartado de desarrollo se explica en profundidad el modelo DETR, el proceso de recolección y preparación de los datos y la preparación del modelo para resolver la detección de piscinas.

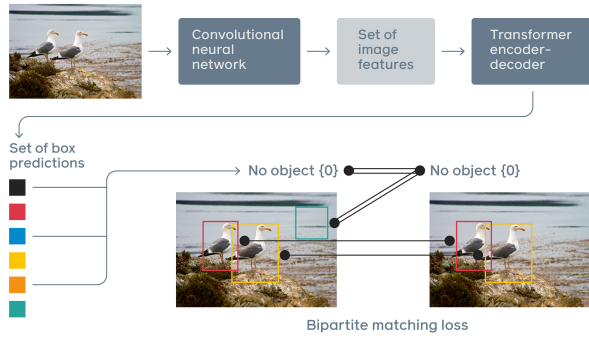


Fig. 1: Modelo DETR (imagen obtenida de [13])

## 5.1. Modelo DETR

DETR (DEtection TRansformer), como se muestra en la Figura 1, es un nuevo enfoque para la detección de objetos y la segmentación panóptica desarrollado por Facebook. Es uno de los primeros marco de detección de objetos que integra con éxito los Transformers como un bloque de construcción central en el pipeline de detección.

El modelo toma una imagen como entrada y la pasa a través de una CNN para extraer características visuales. Estas características se pasan a un codificador Transformer, que razona sobre la imagen en su conjunto y genera un conjunto fijo de consultas de objetos aprendidas. Estas consultas se pasan a un decodificador Transformer junto con las características visuales para generar el conjunto final de predicciones. Cada predicción consta de una caja delimitadora y una clasificación de objetos.

Durante el entrenamiento, las predicciones se asignan de manera única a las bounding box reales mediante el emparejamiento bipartito con el algoritmo húngaro [16] y se calcula una función de pérdida para actualizar los pesos del modelo.

## 5.2. Recolección y preparación de datos

Como se puede observar en el diagrama de la Figura 2, primero se adquieren las imágenes y la posición de las piscinas del Instituto Cartográfico y Geológico de Cataluña (ICGC) [17], las imágenes usadas son ortoimágenes ya facilitan el trabajo al eliminar variables. Para recolectar la información se ha creado un programa que descargue las imágenes de todas las zonas de Cataluña una a una y las recorra con una sliding window, solo guardando las windows que contengan piscinas con sus anotaciones correspondientes en el formato COCO para su posterior uso.

La sliding window utilizada tiene un tamaño de 700x700, siguiendo los límites establecidos por los investigadores de Facebook, quienes indican que todas las imágenes deben tener un tamaño entre 500 y 1300 píxeles. Para cubrir la imagen completa, la sliding window se desplaza 175 píxeles, lo que permite realizar data augmentation mediante la superposición. Cabe destacar que las imágenes utilizadas en este trabajo tienen una resolución espacial de 25 cm por píxel, lo que significa que cada píxel representa un área de

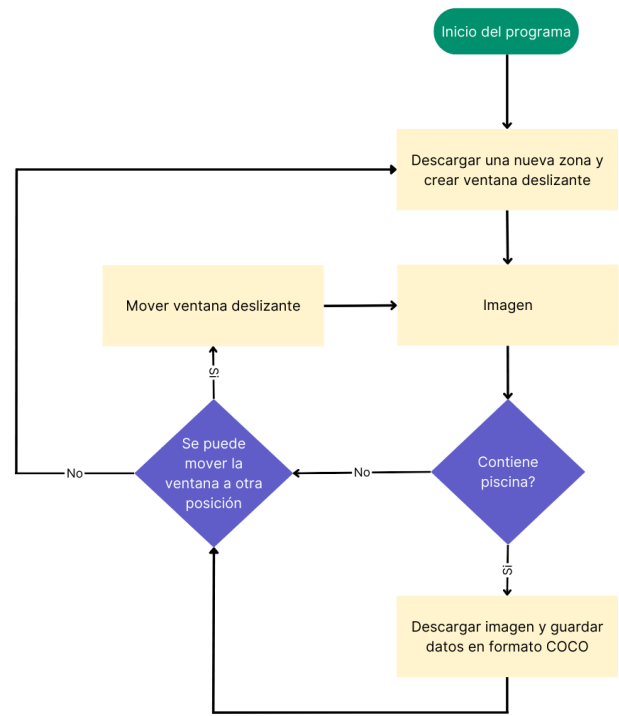


Fig. 2: Diagrama de como se han recolectado los datos.

25x25 cm.

Por cada window guardada se crean anotaciones en formato COCO, eso significa que se guardan dos tipos de anotaciones. Por un lado, las anotaciones de la window: el ID de la imagen, el nombre del archivo, anchura, altura y el origen de donde se ha extraído. Por otro lado, la información de todas las piscinas de la window: ID piscina, ID de la imagen a la que pertenece, categoría, área y bounding box. En este caso, solo existe una categoría que sería "piscina", y el formato de la bounding box sigue la estructura (x mínimo, y mínimo, anchura, altura), como se ilustra en la Figura 3.

Es importante tener en cuenta que en el ICGC no existe una categoría específica de "piscina", ya que la categoría abarca diferentes tipos de zonas hidráulicas, como piscinas, embalses y otros cuerpos de agua. Con la recolección de datos mencionada se han obtenido 80.205 imágenes con 218.213 zonas hidráulicas marcadas en total. Se observa una tendencia en las imágenes, donde aquellas con menos anotaciones suelen ser imágenes rurales con embalses de riego u otras características similares. Para filtrar los datos, se decidió utilizar las imágenes que contenían al menos 6 piscinas marcadas. Además, se realizó un recorrido manual de todas las imágenes para eliminar aquellas que claramente no tenían ninguna piscina y solo generaban errores en el entrenamiento. Con estos criterios, se construyó una base de datos final que consta de 9,550 imágenes que contienen 99,454 piscinas anotadas. Estos datos se dividieron en conjuntos de entrenamiento, prueba y validación, correspondiendo al 80 %, 10 % y 10 % respectivamente del total de imágenes.

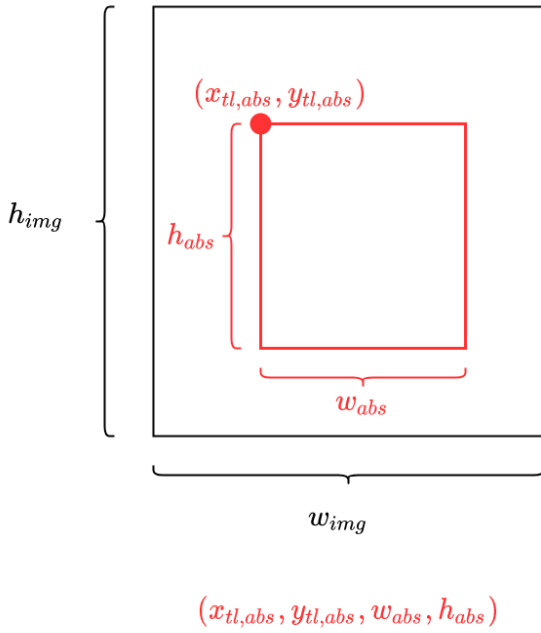


Fig. 3: Bounding box en formato COCO. (imagen obtenida de [18])

Es importante destacar que al verificar los datos finales, se ha observado una gran cantidad de piscinas en las imágenes que no estaban etiquetadas, así como etiquetas colocadas incorrectamente sobre objetos que no son piscinas. Esto puede ser resultado de piscinas no declaradas o de una declaración incorrecta de su posición. Además, la falta de datos actualizados con todas las ubicaciones de las piscinas y las imágenes correspondientes al mismo año también puede limitar los resultados potenciales.

Para realizar una evaluación válida y precisa, se ha llevado a cabo un repaso manual de todas las imágenes de test. Durante este proceso, se han eliminado las piscinas no visibles, como aquellas totalmente cubiertas por árboles o bajo estructuras, y se han agregado las bounding boxes faltantes en las piscinas que no estaban marcadas inicialmente.

Por último, la base de datos se ha publicado en Roboflow [19] para que esté disponible para cualquier persona interesada, y el código utilizado en este trabajo se ha compartido en GitHub [15].

### 5.3. Pipeline

Posteriormente, se ha diseñado un pipeline para entrenar al modelo que ha sido adaptado y ajustado para adecuarlo a la tarea mediante el ajuste de las clases que predice, se ha hecho para que solo prediga la clase "piscina". El modelo predice para cada imagen 100 bounding boxes y la probabilidad de que cada una de las bounding box sea de la clase "piscina".

Como se mencionó anteriormente, el modelo DETR utiliza el algoritmo húngaro para seleccionar las bounding boxes y clases más similares durante el entrenamiento. Una vez que se determinan, se calculan tres funciones de loss:

dos de ellas se centran en las bounding boxes, el l1 loss y el generalized IoU [20], mientras que la tercera se basa en la probabilidad de clase y se calcula utilizando el binary cross entropy loss, dado que solo se trabaja con una única clase.

En las imágenes de entrenamiento se ha aplicado data augmentation, donde cada imagen tiene una probabilidad del 50 % de ser volteada horizontalmente, otro 50 % de ser volteada verticalmente y un 50 % de ser rotada 90 grados en cualquier dirección. Además, se han normalizado las bounding boxes para que estén dentro del rango de 0 a 1, y se ha realizado una normalización de la imagen para mejorar los resultados durante el entrenamiento.

Por último, debido a la naturaleza de los datos, se ha decidido aplicar non-maximum suppression [21] a las bounding box predichas para evitar superposición, ya que no puede haber dos piscinas una encima de otra.

## 6 RESULTADOS

En esta sección se presentan los resultados obtenidos del trabajo realizado, empezando por las métricas de test, pasando por ver las predicciones en las imágenes y por último la comparación con el estado del arte.

El mejor modelo entrenado se obtuvo mediante el proceso de fine-tuning de un modelo DETR, sin congelar ninguna de sus partes. Este modelo fue entrenado durante 241 épocas utilizando los siguientes parámetros: un learning rate de 0.0001, un weight decay de 0.0001 y un batch size de 8. El entrenamiento se realizó utilizando el optimizador AdamW y se asignaron pesos de loss de 5 para el l1, 2 para el generalized IoU y 2 para el binary cross entropy loss. Además, se aplicó la técnica de gradient norm clipping [22] para abordar los problemas de explosión y desvanecimiento del gradiente.

El proceso de entrenamiento de este modelo se llevó a cabo durante un total de 21 horas utilizando una tarjeta gráfica NVIDIA TITAN Xp con 12 GB de RAM. Por otro lado, la inferencia del modelo en el mismo hardware se realiza en tan solo 0.13 segundos por cada conjunto de 8 imágenes, lo que se traduce en un tiempo de aproximadamente 16.25 ms por imagen. Estos resultados demuestran la eficiencia y capacidad de procesamiento del modelo entrenado en términos de velocidad de inferencia.

### 6.1. Métricas

Para evaluar y comprender el rendimiento del modelo, se decidió calcular la Precisión, el Recall y el F1 score. La Precisión indica el porcentaje de predicciones correctas, el Recall representa el porcentaje de objetos reales que han sido detectados, y el F1 score es una medida que combina ambos valores y permite evaluar el rendimiento global del modelo.

Estas métricas se calcularon utilizando matrices de confusión, tal como se muestra en la Tabla 1. Las matrices se generaron utilizando las predicciones del modelo y las bounding boxes reales, aplicando un umbral de IoU de 0.5

y ajustando la confianza de las predicciones en un rango de 0.1 a 0.9.

En la matriz de confusión, los valores True Positive representan las bounding boxes predichas correctamente, es decir, aquellas que se superponen con las reales con un IoU superior al umbral establecido. Los False Positives corresponden a las predicciones incorrectas, por lo tanto, son aquellas bounding boxes predichas que no cumplen con esta condición. Por último, los False Negatives son las bounding boxes reales que no han sido detectadas por ninguna predicción que supere el umbral de IoU establecido. Es importante destacar que la matriz de confusión no incluye valores True Negatives, ya que este concepto no es relevante en la detección de objetos.

TABLA 1: Matriz de confusión con de 0.2 confianza y 0.5 de umbral de IoU sobre los datos de test

|          | Positive prediction | Negative prediction |
|----------|---------------------|---------------------|
| Positive | 8981                | 2570                |
| Negative | 1691                | -                   |

Como se ha mencionado se han calculado la Precisión, el Recall y el F1 score de cada uno de las confianzas indicadas anteriormente, como se puede observar en la tabla 2.

TABLA 2: Métricas obtenidas para el umbral IoU de 0.5 según la confianza

| Confianza | Precisión | Recall | F1 score |
|-----------|-----------|--------|----------|
| 0.1       | 0.7820    | 0.8099 | 0.7957   |
| 0.2       | 0.8415    | 0.7775 | 0.8083   |
| 0.3       | 0.8701    | 0.7547 | 0.8083   |
| 0.4       | 0.8885    | 0.7366 | 0.8054   |
| 0.5       | 0.9008    | 0.7194 | 0.7999   |
| 0.6       | 0.9121    | 0.7010 | 0.7927   |
| 0.7       | 0.9237    | 0.6779 | 0.7819   |
| 0.8       | 0.9334    | 0.6427 | 0.7612   |
| 0.9       | 0.9505    | 0.5620 | 0.7064   |

Al analizar las métricas calculadas, se observa que a medida que aumenta la confianza de las predicciones, la Precisión también aumenta pero el Recall disminuye. Además, se nota que el F1 score aumenta y alcanza su máximo valor con una confianza de 0.2 y 0.3, y luego comienza a disminuir nuevamente.

Teniendo en cuenta el escenario actual y los resultados obtenidos, se considera que el Recall es más importante en este caso, esto se debe a que se planea que los resultados sean revisados por un humano, quien finalizará el filtrado de las piscinas. Por lo tanto, se seleccionará una confianza de 0.2, ya que proporciona el F1 score más alto y, de las dos opciones consideradas, tiene el mayor Recall. Con una Precisión del 84 % y un Recall del 77 %, se reducirá significativamente el trabajo humano necesario y, como resultado, se aumentará la eficiencia del proceso.

## 6.2. Resultados visibles

A continuación, se presentarán los resultados del modelo utilizando las imágenes procesadas, con el objetivo de

analizar su desempeño y limitaciones. Las predicciones se han realizado utilizando un umbral de confianza de 0.2 sobre los datos de test.

En la imagen 1, representada en las Figuras 4a y 4b, el modelo muestra una detección precisa de todas las piscinas. Se evidencia que ha aprendido a identificar piscinas sin importar su orientación, color, forma o tamaño. En la imagen se pueden observar piscinas vacías, piscinas verdes, piscinas con formas peculiares, piscinas de diversos tamaños, y el modelo ha logrado detectarlas correctamente. Además, el modelo ha sido capaz de identificar piscinas parcialmente cubiertas por árboles o con sombras pronunciadas, lo que demuestra su habilidad para superar esos desafíos y realizar detecciones precisas.

Al examinar la imagen 2, representada en las Figuras 4c y 4d, se puede apreciar que el modelo ha realizado predicciones precisas para varias piscinas. Sin embargo, se han dejado sin detectar las más difíciles de identificar, como cuatro piscinas muy pequeñas, una piscina vacía y una piscina parcialmente cubierta por un árbol. Además, se observa que una de las bounding boxes reales no está ubicada correctamente, pero el modelo ha predicho su posición de manera acertada. Por último, se identifica una bounding box real que indica una zona como piscina, pero aparentemente es errónea, ya que contiene matorrales y hierba en su interior. Esto sugiere que podría tratarse de una piscina en construcción o en estado deteriorado.

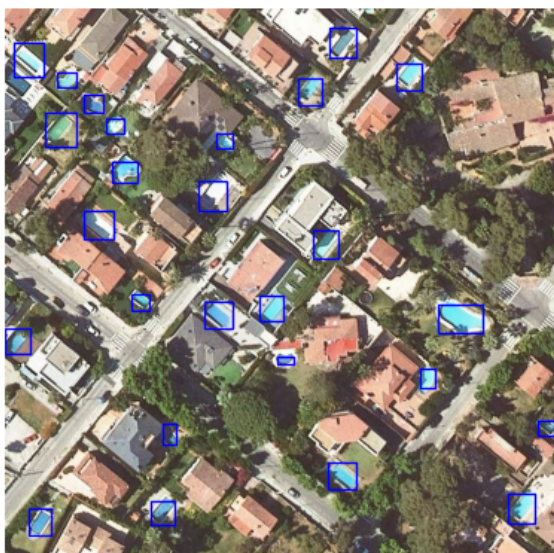
En la imagen 3, representada en las Figuras 4e y 4f, se puede apreciar que el modelo ha logrado detectar correctamente las piscinas más grandes. Sin embargo, ha presentado dificultades al detectar piscinas más pequeñas. Incluso en el caso de una piscina grande, el modelo la ha identificado erróneamente como dos piscinas debido a la presencia de una sombra. Por otro lado, el modelo ha detectado una de las piscinas pequeñas, pero la ha estimado como significativamente más grande de lo esperado.

Como se ha podido observar, existen casos en los que el modelo no logra detectar correctamente las piscinas. En primer lugar, está el desafío de las piscinas cubiertas o parcialmente cubiertas. El modelo puede detectar algunas piscinas parcialmente cubiertas, dependiendo de su visibilidad. Sin embargo, en el caso de piscinas completamente cubiertas, no es posible detectarlas mediante imágenes aéreas, lo que resulta en errores en los datos, ya que la bounding box indica la presencia de una piscina que no es visible.

Otro desafío se presenta con las piscinas muy pequeñas. El modelo enfrenta dificultades para detectar piscinas por debajo de cierto tamaño. Es importante considerar que estas piscinas pequeñas pueden ser desmontables o inflables, lo que podría haber generado errores en el etiquetado durante la revisión del conjunto de test. Es posible que en los datos de entrenamiento haya poca presencia de piscinas tan pequeñas, lo que dificulta el aprendizaje del modelo en este aspecto.

En resumen, aunque el modelo ha mostrado un rendi-





(a) Imagen 1 con bounding box reales.



(b) Imagen 1 con predicciones.



(c) Imagen 2 con bounding box reales.



(d) Imagen 2 con predicciones.



(e) Imagen 3 con bounding box reales.



(f) Imagen 3 con predicciones.

Fig. 4: Imágenes de test con bounding boxes con un 70 % de confianza



miento generalmente preciso en la detección de piscinas, hay situaciones particulares, como piscinas cubiertas, piscinas muy pequeñas y casos de sombras, donde su desempeño se ve limitado. Estos desafíos específicos deben tenerse en cuenta al interpretar los resultados y considerar posibles mejoras para futuras aplicaciones.

### 6.3. Comparación con el estado del arte

Con el fin de situar el trabajo dentro del estado del arte, se decidió realizar una comparación de los resultados obtenidos con los presentados en el trabajo de Biel Stela [8], ya que es el único trabajo encontrado que proporciona sus resultados. Es importante destacar que no se utilizó la misma base de datos, ya que no se tenía acceso a la utilizada en el otro trabajo. Además, se emplearon diferentes áreas geográficas, siendo este trabajo realizado en Cataluña, mientras que el otro se enfocó en la isla de Mallorca. También se observó una diferencia en la resolución espacial de las imágenes, siendo 25 cm por píxel en este trabajo y 15 cm por píxel en el otro. Estas diferencias hacen que las comparaciones directas no sean completamente precisas. Sin embargo, dado que es el único trabajo que proporciona datos de sus resultados, se decidió realizar una comparación básica.

TABLA 3: Datos obtenido de [8]

|                  | Confianza |      |
|------------------|-----------|------|
|                  | 0.4       | 0.8  |
| <b>Precisión</b> | 0.72      | 0.94 |
| <b>Recall</b>    | 0.90      | 0.74 |
| <b>F1 score</b>  | 0.8       | 0.83 |

Al comparar los resultados presentados en la tabla 2 con los datos obtenidos del trabajo de Biel 3, se puede observar una similitud entre ambos. Sin embargo, es importante destacar que para realizar comparaciones más precisas sería necesario utilizar la misma base de datos tanto para el entrenamiento como para el test. En este sentido, es relevante mencionar que en este trabajo se utilizó una base de datos que se ha hecho pública [19], la cual está disponible para su acceso por parte de cualquier interesado. Esto se ha realizado con el objetivo fomentar la reproducibilidad y brinda la oportunidad de realizar evaluaciones comparativas más exhaustivas en el futuro.

## 7 CONCLUSIONES

En este estudio se ha evaluado la viabilidad de utilizar Transformers como detectores de piscinas, los resultados obtenidos con el modelo DETR demuestran que el uso de Transformers en la detección de objetos es viable, y se vislumbra su potencial como uno de los métodos más utilizados en un futuro cercano.

Los resultados obtenidos son considerados buenos y podrían aplicarse para reducir la carga de trabajo humana en la detección de piscinas ilegales, eliminando gran parte de las tareas repetitivas. No obstante, se reconoce la importancia del factor humano en la etapa final de verificación de las piscinas detectadas y en la toma de

decisiones correspondientes.

El método empleado en este estudio se considera igualmente válido que el estado del arte actual. Se plantean futuras líneas de investigación, como la mejora de los datos, ya que se identifica como el principal factor limitante del estudio. Una posible solución sería utilizar técnicas de aprendizaje activo para mejorar la calidad de los datos, tal como se mencionó en los objetivos secundarios. Además, se propone explorar la técnica del mosaico como otra línea de investigación, con el fin de mejorar la detección de piscinas pequeñas. Esta técnica consiste en dividir la imagen en fragmentos más pequeños, entrenar y realizar inferencias con ellos.

En resumen, se considera que este estudio ha logrado cumplir los objetivos principales establecidos. Sin embargo, no se han explorado otras arquitecturas y técnicas, ni se ha implementado un sistema de notificación por correo electrónico para piscinas ilegales, los cuales eran objetivos secundarios. Esto se debe a que se ha utilizado todo el tiempo disponible para alcanzar el objetivo principal del estudio.

## REFERENCIAS

- [1] El Periódico. “En España hay más de 1.266.000 piscinas, una por cada 37 habitantes.” <https://www.elperiodico.com/es/sociedad/20220826/cuantas-piscinas-hay-espana-14350587> (accessed Apr. 16, 2023).
- [2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” arXiv.org, <https://arxiv.org/abs/1506.02640> (accessed Apr. 20, 2023).
- [3] W. Liu et al., “SSD: Single shot multibox detector,” arXiv.org, <https://arxiv.org/abs/1512.02325> (accessed May 28, 2023).
- [4] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” arXiv.org, <https://arxiv.org/abs/1708.02002> (accessed May 28, 2023).
- [5] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” arXiv.org, <https://arxiv.org/abs/1506.01497> (accessed May 28, 2023).
- [6] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” arXiv.org, <https://arxiv.org/abs/1703.06870> (accessed May 28, 2023).
- [7] Z. Cai and N. Vasconcelos, “Cascade R-CNN: Delving into high quality object detection,” arXiv.org, <https://arxiv.org/abs/1712.00726> (accessed May 28, 2023).
- [8] B. Stela, “Supervised detection of swimming pools from orthoimages in Mallorca,” TFM, Computer Vision Center, Universitat Autònoma de Barcelona, Barcelona, Spain, 2021.

- [9] J. Pastor, “Francia llevaba años buscando piscinas privadas no declaradas hasta que se topó con la inteligencia artificial,” *Gadgets y tecnología: últimas tecnologías en electrónica de consumo*, <https://www.xataka.com/robotica-e-ia/francia-ha-descubierto-20-000-piscinas-privadas-no-declaradas-gracias-a-inteligencia-artificial> (accessed Apr. 21, 2023).
- [10] R. Singh, “How we did it: Integrating arcgis and Deep Learning at UC 2018,” *ArcGIS Blog*, <https://www.esri.com/arcgis-blog/products/api-python/analytics/how-we-did-it-integrating-arcgis-and-machine-learning-at-uc-2018/> (accessed Jun. 14, 2023).
- [11] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko. End-to-End Object Detection with Transformers. In 16th European Conference on Computer Vision – ECCV 2020, pp.213–229, 2020
- [12] ImageNet, <https://www.image-net.org/> (accessed Feb. 5, 2023).
- [13] Facebookresearch, “Facebookresearch/detr: End-to-end object detection with Transformers,” *GitHub*, <https://github.com/facebookresearch/detr> (accessed Feb. 3, 2023).
- [14] “Pytorch,” *PYTORCH DOCUMENTATION*. Open Source, Mar. 15, 2023
- [15] R. Manresa, “Roberto-uni/pool-detection,” *GitHub*, <https://github.com/Roberto-Uni/Pool-Detection> (accessed Apr. 16, 2023).
- [16] H. W. Kuhn, “The Hungarian method for the assignment problem,” *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83-97, 1955.
- [17] Descargar La cartografia de Catalunya ICGC, <http://www.icc.cat/appdownloads/> (accessed Feb. 1, 2023).
- [18] “Annotation conversion: Coco JSON to Yolo Txt,” Haobin Tan, <https://haobin-tan.netlify.app/ai/computer-vision/object-detection/coco-json-to-yolo-txt/> (accessed Jul. 1, 2023).
- [19] R. Manresa, “Pools Detector Object Detection Dataset (V4, 2023-06-28 3:07am) by student of the Universitat autonoma de barcelona,” *Roboflow*, <https://universe.roboflow.com/student-of-the-universitat-autonoma-de-barcelona/pools-detector/dataset/4> (accessed Jun. 30, 2023).
- [20] H. Rezatofighi et al., “Generalized intersection over union: A metric and a loss for bounding box regression,” *arXiv.org*, <https://arxiv.org/abs/1902.09630> (accessed Jun. 30, 2023).
- [21] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, “Soft-NMS – improving object detection with one line of code,” *arXiv.org*, <https://arxiv.org/abs/1704.04503> (accessed Jun. 30, 2023).
- [22] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” *arXiv.org*, <https://arxiv.org/abs/1211.5063> (accessed Jul. 2, 2023).

## APÉNDICE

| TAREAS                                        | MARZO      |            | ABRIL      |            | MAYO        |                            | JUNIO       |                                           |
|-----------------------------------------------|------------|------------|------------|------------|-------------|----------------------------|-------------|-------------------------------------------|
|                                               | SPRINT 1   | SPRINT 2   | SPRINT 3   | SPRINT 4   | SPRINT 5    | SPRINT 6                   | SPRINT 7    | SPRINT 8                                  |
| Investigación                                 | Subtarea 1 |            |            |            |             |                            |             |                                           |
| Recolección de datos y procesamiento de datos | Subtarea 2 | Subtarea 4 |            |            | Subtarea 10 |                            |             |                                           |
| Preparación del entorno y del modelo          |            | Subtarea 5 | Subtarea 6 |            |             |                            |             |                                           |
| Entrenamiento                                 |            |            | Subtarea 7 | Subtarea 8 | Subtarea 11 |                            |             |                                           |
| Evaluación                                    |            |            |            |            |             | Subtarea 12<br>Subtarea 13 | Subtarea 15 |                                           |
| Documentación                                 | Subtarea 3 |            |            | Subtarea 9 |             | Subtarea 14                | Subtarea 16 | Subtarea 17<br>Subtarea 18<br>Subtarea 19 |

Fig. 5: Diagrama de Gantt