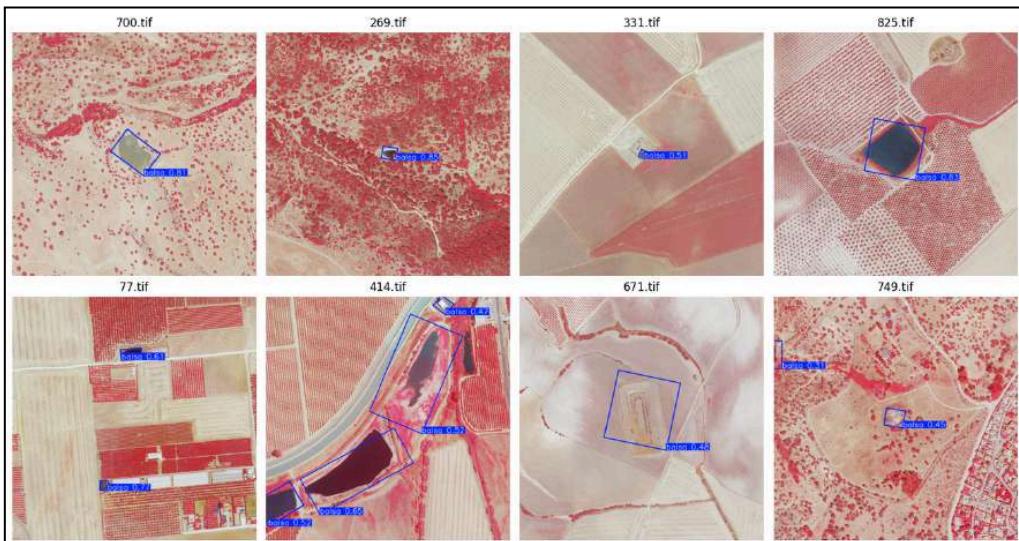


MÁSTER EN DATA SCIENCE Y BUSINESS ANALYTICS

DETECCIÓN DE BALSAS DE AGUA EN IMÁGENES AÉREAS PNOA UTILIZANDO DEEP LEARNING CON YOLOv8n-OBB



TFM elaborado por: Carlos Agulla Villar
Tutor/a de TFM: Juan Manuel Moreno Lamparero

Índice

Resumen.....	1
Abstract.....	1
Introducción y antecedentes.....	3
Objetivos del proyecto.....	4
Objetivo General.....	4
Objetivos Específicos.....	4
Material y métodos.....	5
Área de Estudio.....	5
Datos Utilizados.....	7
Metodología.....	8
Preprocesado de los datos.....	9
Preparación del Dataset.....	9
Entrenamiento del Modelo.....	10
Validación durante el Entrenamiento.....	11
Resultados.....	12
Adquisición de fuentes de datos.....	12
Datos vectoriales.....	12
Datos vectoriales.....	12
Descripción de fuentes de datos.....	12
Análisis exploratorio de datos o EDA.....	13
Preprocesado de datos.....	14
Preprocesado de Datos Vectoriales.....	14
Preprocesado de Datos Ráster.....	15
Preparación de Datos de Entrenamiento.....	15
Selección de variables.....	17
Partición de los datos en conjuntos train, val y test.....	17
Aplicación del modelo.....	18
Validación e inferencia del modelo.....	20
Validación.....	20
Inferencia.....	21
Comparativa de resultados.....	26
Conclusiones.....	27
Glosario.....	29
Referencias bibliográficas.....	35
Anexos.....	39
Anexo 1: Cookiecutter.....	40
Anexo 2: Archivo YALM.....	41
Anexo 3: Visualizaciones del Entrenamiento.....	42
Anexo 4: Resultado de la Inferencia Sobre Imágenes de Test.....	53
Anexo 5: Notebooks y Modelo entrenado.....	81

Resumen

Este Trabajo de Fin de Máster (TFM) se centra en el desarrollo de una metodología automatizada para la detección de balsas de agua, utilizando imágenes Ortofotos PNOA de Falso Color Infrarrojo de alta resolución y técnicas avanzadas de deep learning. El objetivo principal es establecer un flujo de trabajo robusto que abarque desde el preprocesamiento de datos vectoriales y ráster hasta el entrenamiento, la validación y la inferencia de un modelo de deep learning. El proceso comienza con el preprocesamiento de datos vectoriales (cartografía de balsas) y ráster (Ortofotos PNOA de Falso Color Infrarrojo), seguido de un análisis exploratorio de datos (EDA) para comprender la distribución y características de las balsas, hasta la preparación específica de los datos para el entrenamiento del modelo. Se calculan los Oriented Bounding Boxes (OBB), se generan recortes de imágenes para el entrenamiento y se crean etiquetas en formato YOLOv8-obb. La fase de entrenamiento se realiza utilizando la arquitectura ligera YOLOv8n-obb, aprovechando un modelo preentrenado (transfer learning) y la capacidad de procesamiento de GPUs en Google Colab. Finalmente, se lleva a cabo la inferencia sobre nuevas imágenes, empleando técnicas avanzadas de deep learning y SAHI para optimizar el procesamiento de imágenes y evaluar predicciones georreferenciadas. Los resultados de validación muestran un desempeño prometedor del modelo.

Este proyecto demuestra la viabilidad de utilizar Ortofotos PNOA de Falso Color Infrarrojo y la arquitectura YOLOv8n-obb para la detección automatizada de balsas, proporcionando una herramienta potencialmente útil para la gestión y el monitoreo de recursos hídricos.

Abstract

This Master's Thesis (TFM) focuses on developing an automated methodology for detecting water bodies using high-resolution False Color Infrared Orthophotos from PNOA and advanced deep learning techniques. The primary objective is to establish a robust workflow that covers data preprocessing, both vectorial and raster, as well as the training, validation, and inference of a deep learning model. The process begins with the preprocessing of vectorial data (water body mapping) and raster data (False Color Infrared Orthophotos from PNOA), followed by an exploratory data analysis (EDA) to understand the distribution and characteristics of the water bodies, leading to the specific preparation of data for model training. Oriented Bounding Boxes (OBB) are

calculated, image crops are generated for training, and labels are created in YOLOv8-obb format. The training phase is conducted using the lightweight YOLOv8n-obb architecture, leveraging a pre-trained model (transfer learning) and the processing power of GPUs in Google Colab. Finally, inference is performed on new images using advanced deep learning techniques and SAHI to optimize image processing and evaluate georeferenced predictions. The validation results show promising performance of the model.

This project demonstrates the feasibility of using False Color Infrared Orthophotos from PNOA and the YOLOv8n-obb architecture for the automated detection of water bodies, providing a potentially useful tool for the management and monitoring of water resources.

Introducción y antecedentes

Andalucía, una región de gran importancia agrícola en España, se enfrenta a desafíos considerables en la gestión de sus recursos hídricos. Su clima mediterráneo, caracterizado por una marcada irregularidad en las precipitaciones y la ocurrencia frecuente de sequías, se ve agravado por una elevada demanda de agua, especialmente en el sector agrícola.

La detección y el análisis espacial de las balsas de agua son cruciales para entender su impacto en el territorio y gestionar los recursos hidrológicos. Estas estructuras hidráulicas son ampliamente utilizadas en actividades agrícolas y de gestión hídrica. Las Ortofotos PNOA de Falso Color Infrarrojo proporcionan una fuente de datos detallada para este propósito. La Base Topográfica Nacional (BTN) del CNIG y la Red de Información Ambiental de Andalucía (REDIAM) complementan esta información con capas vectoriales de superficies de agua para usos agrícolas y ganaderos. El proyecto busca integrar estas fuentes de datos para desarrollar un método automatizado de detección de balsas utilizando *deep learning*.

El uso de modelos de detección de objetos, como los basados en la familia YOLO, se ha extendido en tareas de análisis de imágenes. Dada la naturaleza y la orientación arbitraria que pueden presentar las balsas en las imágenes aéreas, la variante YOLOv8 con *Oriented Bounding Boxes* (OBB) es especialmente adecuada (Bochkovskiy, et al. (2020), Khan et al. (2022)). Este enfoque permite capturar la orientación de los objetos detectados, mejorando la precisión respecto a los *bounding boxes* tradicionales alineados con los ejes.

Como señalan López-Tapia et al. (2021), "el uso de técnicas de aprendizaje automático con imágenes aéreas de alta resolución permite mejorar significativamente la detección automatizada de humedales". Esta afirmación refuerza la viabilidad del enfoque adoptado en este trabajo, aplicado ahora al caso específico de balsas en entornos agrícolas andaluces mediante imágenes PNOA.

Objetivos del proyecto

Objetivo General

Desarrollar un pipeline completo para la detección automatizada de balsas de agua en Ortofotos PNOA Falso Color Infrarrojo utilizando un modelo de deep learning basado en YOLOv8n-obb.

Objetivos Específicos

- Preprocesar datos vectoriales de marcos de imágenes PNOA, cartografía BTN y datos REDIAM para definir el ámbito de estudio e identificar potenciales ubicaciones de balsas.
- Preprocesar y optimizar imágenes ráster PNOA IRG, incluyendo la reducción de resolución y la creación de un mosaico virtual (VRT).
- Realizar un Análisis Exploratorio de Datos (EDA) sobre las capas vectoriales resultantes para comprender la distribución, tamaño y forma de las balsas, y su relación con los marcos de imágenes.
- Preparar un dataset estructurado para el entrenamiento del modelo YOLOv8n-OBB, calculando OBBs para las balsas, generando recortes de imágenes y creando archivos de etiquetas normalizadas.
- Entrenar un modelo YOLOv8n-obb utilizando el dataset preparado, aprovechando técnicas de transfer learning y recursos de computación en la nube (Google Colab).
- Evaluar el rendimiento del modelo entrenado utilizando métricas estándar de detección de objetos.
- Implementar un proceso de inferencia, utilizando SAHI, para aplicar el modelo a imágenes completas y convertir las predicciones en geometrías georreferenciadas.

Material y métodos

Área de Estudio

El área de estudio se localiza en la Comunidad Autónoma de Andalucía, España, abarcando una extensión total aproximada de 5800 km². Geográficamente, se sitúa mayoritariamente en la provincia de Sevilla, extendiéndose también hacia una porción de la provincia de Córdoba. Esta selección se realizó estratégicamente para conformar una zona compacta donde los usos del suelo predominantes son agrícolas, con el objetivo de asegurar una alta densidad de los elementos de interés para el proyecto: balsas de agua para uso agrícola o ganadero.

La delimitación del área de estudio se define por la extensión del conjunto de las 42 imágenes PNOA seleccionadas específicamente para este propósito. Se priorizó la continuidad espacial de estas imágenes, garantizando que todas se encontraran dentro del mismo huso geográfico. Asimismo, se verificó que la fecha de adquisición de todas las imágenes fuera la misma, lo que sugiere una captura durante un mismo vuelo, aportando una consistencia a los datos.

Características Geográficas y Topográficas

El área de estudio se sitúa en una zona de transición entre la Depresión del Guadalquivir y las primeras estribaciones de Sierra Morena. La parte sur, dentro de la provincia de Sevilla, se caracteriza por ser una llanura aluvial extensa y de baja altitud, modelada por el curso del río Guadalquivir y sus afluentes. Esta zona presenta una topografía predominantemente llana o suavemente ondulada, con suelos fértiles propicios para la agricultura.

A medida que se avanza hacia el norte, la topografía experimenta un ligero ascenso, marcando el inicio de las estribaciones de Sierra Morena. En esta zona, el relieve se vuelve ligeramente más accidentado, con la presencia de colinas suaves.

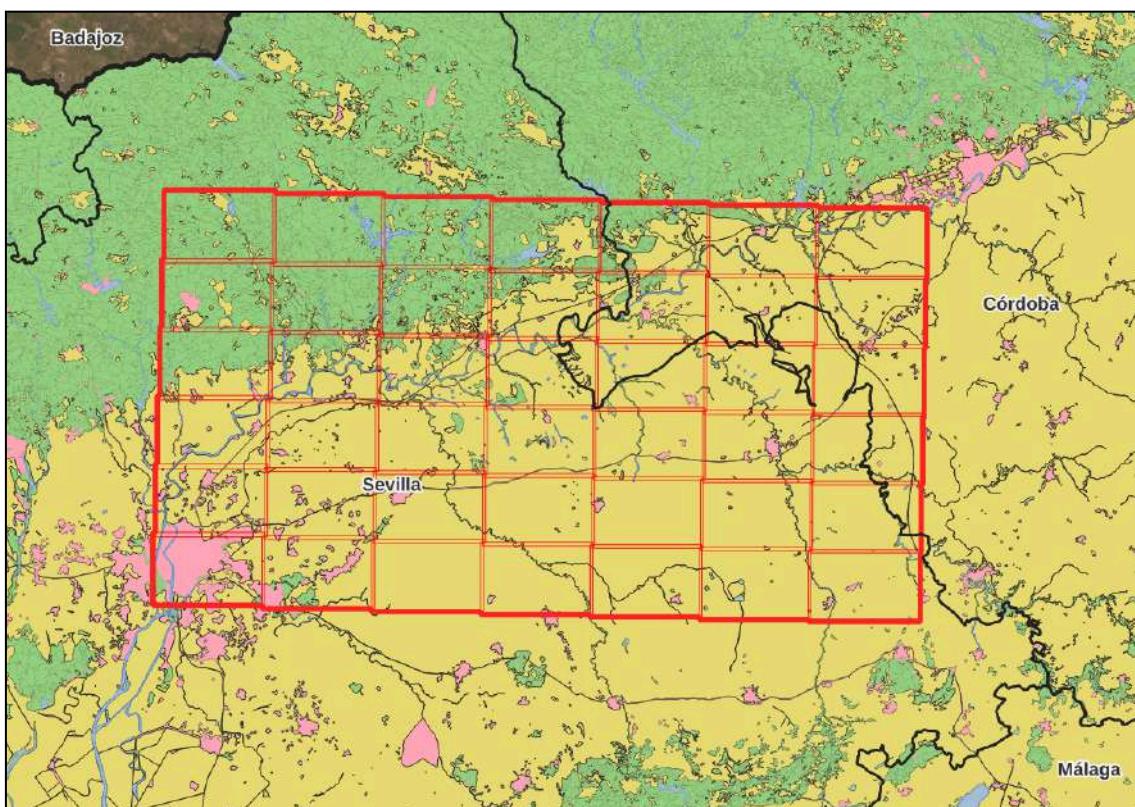
Usos del Suelo

El uso del suelo predominante en el área de estudio es agrícola. La llanura del Guadalquivir (mayormente en la provincia de Sevilla) se dedica principalmente a la agricultura intensiva de regadío, destacando cultivos como el olivar, los cereales de invierno y diversos cultivos hortícolas.

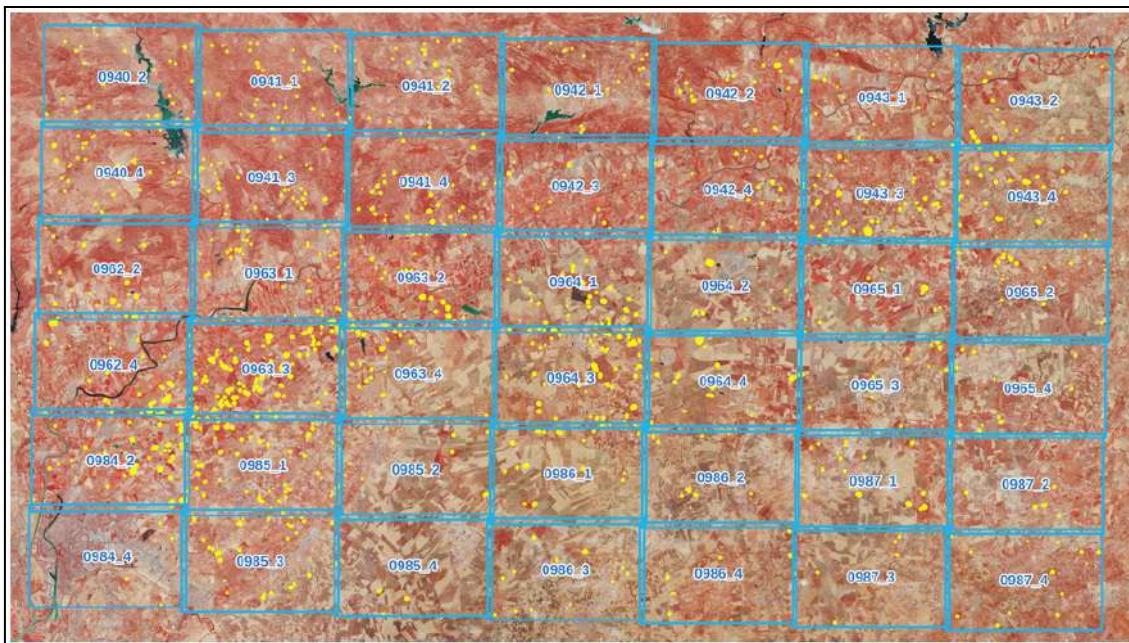
En la zona de transición y la porción perteneciente a la provincia de Córdoba, aunque la agricultura sigue siendo importante, se observa una mayor presencia de agricultura de secano (olivar, viñedo, cereales). A medida que el terreno se acerca a

las primeras elevaciones de Sierra Morena, puede haber una mayor representación de pastizales y vegetación natural asociada al monte mediterráneo.

El área de estudio se configura como una zona eminentemente agrícola, estratégicamente seleccionada por su homogeneidad en usos del suelo y condiciones de adquisición de las imágenes, abarcando principalmente la fértil llanura sevillana y extendiéndose hacia provincia de Córdoba.



Área de estudio (en rojo) sobre el Mapa Forestal de Andalucía. Representación de los marcos de las 42 Ortofotos PNOA. Zonas agrícolas y de cultivos en color amarillo y terrenos de 'monte' en tonos verdes.



Nomenclatura de las Ortofotos PNOA Falso Color con sus marcos (en azul) y la distribución de las balsas (en amarillo) dentro del ámbito de estudio. Se observan áreas con mayor densidad que otras.

Datos Utilizados

Datos Vectoriales:

La base principal de geometrías de balsas se obtuvo de la capa BTN0328S_ALM_AGU.shp (Almacenamiento de Agua) de la Base Topográfica Nacional (BTN), disponible en Centro de Descargas del CNIG (Centro Nacional de información Geográfica). Esta capa fue filtrada espacialmente utilizando la capa "Cartografía de Balsas año 2020" de la Red de Información Ambiental de Andalucía (REDIAM), específicamente seleccionando las entidades clasificadas como "*Balsa de riego o ganadera*" de la REDIAM. Las geometrías de la BTN0328S_ALM_AGU.shp fueron seleccionadas por definir de manera más precisa los perímetros de las balsas de agua (aunque en la misma categoría mezcla balsas con albercas y estanques), mientras que las geometrías de la capa de la REDIAM son más imprecisas e irregulares (pero clasifican las balsas según sus usos concretos). Los datos vectoriales fueron reproyectados al sistema de coordenadas EPSG:25830 (el mismo que el de las ortofotos PNOA) y recortados al ámbito de estudio definido por los marcos de las Ortofotos PNOA. Adicionalmente, se aplicó un filtrado por superficie mínima, eliminando balsas menores de 200 m², y se realizaron filtros manuales de geometrías no relevantes (erróneas o inexistentes).

Datos Ráster: Ortofotos PNOA Falso Color Infrarrojo

Se utilizaron ortofotos PNOA Falso Color Infrarrojo descargadas del Centro de Descargas del CNIG. Estas imágenes originales, distribuidas en formato COG (Cloud Optimized GeoTIFF), tienen una resolución de píxel de 0,25 metros. Sin embargo, esta resolución original genera un volumen de datos excesivamente grande para ser manejado eficientemente, resultando en tiempos de procesamiento prolongados y un uso elevado de recursos computacionales sin un beneficio significativo para la detección de balsas.

Para optimizar el procesamiento y reducir la carga computacional, se aprovechó la estructura de niveles de pirámide (overviews) inherente a los archivos COG. Se extrajo y utilizó el nivel 2 de la pirámide, que aplica un factor de reducción de 4, obteniéndose imágenes donde cada píxel representa 1 metro, resolución suficiente para identificar balsas. Este proceso redujo el volumen de píxeles en un 93.75%, logrando un balance óptimo entre la cantidad de datos y la resolución necesaria para la detección.

Todas las imágenes seleccionadas en el Área de estudio fueron capturadas en la misma fecha, julio de 2022. Se utilizaron un total de 42 imágenes individuales asociadas a los marcos de las hojas del Mapa Topográfico Nacional 1:25.000 (MTN25). Las versiones reducidas de estas imágenes se guardaron en formato GeoTIFF. La mayoría de imágenes procesadas varían en el rango de 250-350 MB. Posteriormente, se creó un Virtual Raster (VRT) combinando virtualmente todas las imágenes, lo que facilitó el acceso y procesamiento unificado, optimizando los recortes y reduciendo la complejidad del manejo de archivos. Todas las imágenes utilizan el mismo sistema de coordenadas (EPSG:25830).

Metodología

Para garantizar una organización eficiente y estructurada del proyecto desde sus inicios, se empleó la librería **cookiecutter**. Esta herramienta permitió generar una plantilla de carpetas predefinida (Anexo 1: Cookiecutter), facilitando la separación lógica de los datos brutos y procesados, los notebooks de análisis y experimentación, así como los modelos entrenados. El uso de cookiecutter sentó una base sólida para la gestión del flujo de trabajo y la reproducibilidad de los resultados a lo largo del desarrollo del TFM.

La Metodología básica empleada para la preparación del dataset y el entrenamiento del modelo se puede definir en los siguientes apartados:

Preprocesado de los datos

El objetivo principal de esta fase fue preparar los datos vectoriales de las balsas y las imágenes ráster PNOA para el entrenamiento de un modelo de detección de objetos, específicamente YOLOv8n-OBB. Esto implicó transformar los datos crudos en formatos adecuados y optimizados para el entrenamiento del modelo.

Preparación del Dataset

La preparación del dataset se realizó utilizando los datos vectoriales de las balsas previamente filtrados y las imágenes ráster preprocesadas (unidas virtualmente en un VRT). Los pasos clave fueron:

- **Cálculo de Bounding Boxes (BB) y Oriented Bounding Boxes (OBB):** Se calcularon los BBs y los OBBs para cada polígono de balsa. Los OBBs se prefirieron por ofrecer una delimitación más precisa que los BBs tradicionales, dada la orientación aleatoria de las balsas. Para garantizar precisión en la detección, se desarrolló una función personalizada para calcular Oriented Bounding Boxes (OBB) y ordenar sus vértices según convenciones estándar.
- **Generación de Cuadrados de Recorte:** Se generaron teselas cuadradas de tamaño fijo (640x640 píxeles) que contenían las geometrías de las balsas. Este tamaño fue elegido por ser óptimo para arquitecturas de CNN como YOLOv8. Se priorizó minimizar la cantidad de cuadrados necesarios para cubrir toda la distribución de las balsas.
- **Recorte de Imágenes Ráster:** Se extrajeron las porciones correspondientes de las imágenes PNOA (provenientes del VRT) utilizando las geometrías de los cuadrados de recorte definidos. Cada recorte se guardó como un archivo GeoTIFF individual. Se utilizó una función llamada `recortar_vrt` para esta tarea.
- **Creación de Etiquetas ('labels') en Formato YOLOv8-OBB:** Se generaron los archivos de texto con las coordenadas normalizadas de los vértices de los OBBs de las balsas dentro de cada recorte de imagen. Estas etiquetas son esenciales para los modelos de detección de objetos, proporcionando información precisa sobre la posición y categoría de los objetos. Se siguió el formato específico de YOLOv8n-obb, ordenando los vértices en sentido horario empezando por la esquina superior izquierda. Se utilizó una función llamada `crear_etiquetas_yolo_obb`.
- **Estructuración del Dataset:** Los recortes de imágenes y sus correspondientes archivos de etiquetas se organizaron en una estructura de directorios para entrenamiento (train), validación (val) y prueba (test). Se utilizó una división

estándar de 70% para entrenamiento, 20% para validación y 10% para prueba. Esta división asegura datos suficientes para el aprendizaje del modelo, la monitorización de su rendimiento durante el entrenamiento y una evaluación final independiente. El dataset final incluyó 593 imágenes GeoTiff de 640x640 píxeles que contenían las 1115 balsas, distribuidas al azar en los conjuntos.

- **Configuración del Archivo YAML:** Se creó un archivo YAML (data.yaml) para definir el conjunto de datos a utilizar por YOLOv8. Este archivo incluye las rutas a los directorios de entrenamiento, validación y prueba, así como la lista de clases (en este caso, solo 'balsa').

Entrenamiento del Modelo

El entrenamiento del modelo YOLOv8n-obb se llevó a cabo después de preparar el dataset.

- **Modelo Utilizado:** Se seleccionó la arquitectura **YOLOv8n-obb**. Se optó por la variante 'nano' (n) por ser ligera y eficiente en cómputo, y por la variante OBB para detectar objetos con orientación arbitraria, algo relevante para las balsas en imágenes aéreas.
- **Plataforma de Entrenamiento:** El entrenamiento se realizó en Google Colab. Esto fue necesario debido a la limitación de recursos de hardware local, específicamente la falta de una GPU potente. Google Colab proporcionó acceso gratuito a GPUs (como la Tesla T4), esenciales para acelerar significativamente el entrenamiento de modelos de deep learning, ya que entrenar YOLOv8 en una CPU local habría llevado tiempos prohibitivamente largos.
- **Técnica de Entrenamiento:** Se utilizó transfer learning, partiendo de un modelo YOLOv8n-obb pre-entrenado. Aprovechar el conocimiento adquirido por un modelo en un dataset masivo permite una convergencia más rápida y reduce la cantidad de datos y tiempo de entrenamiento necesarios en comparación con entrenar desde cero.
- **Parámetros de Entrenamiento:** La configuración del entrenamiento se definió en el archivo data.yaml y mediante el método model.train() de la librería Ultralytics. Algunos parámetros utilizados incluyeron: 100 épocas, una paciencia de 15, tamaño de lote (batch size) de 8 (en Colab, reducido para evitar problemas de memoria en T4) (vs 2 en un ejemplo local), tamaño de imagen (imgsz) de 640 píxeles, optimizador AdamW, y pesos específicos para

las pérdidas (box=7.5, cls=0.5, dfl=1.5). Se activó la generación de gráficos (plots=True) y el guardado de checkpoints (save=True).

- **Duración:** El entrenamiento se llevó a cabo durante aproximadamente 47 minutos.
- **Monitorización:** Durante el entrenamiento, se monitorearon las pérdidas (box_loss, cls_loss, dfl_loss) y las métricas de validación en el conjunto de validación.

Validación durante el Entrenamiento

El conjunto de validación (20% de los datos) se utilizó durante el entrenamiento para monitorizar el rendimiento del modelo y ajustar hiperparámetros. Tras finalizar el entrenamiento, se evaluó el modelo (específicamente el mejor checkpoint guardado, best.pt) en el conjunto de validación utilizando el método model.val(). Las métricas clave obtenidas fueron mAP@0.5 de 0.897 y mAP@0.5:0.95 de 0.574. Estos resultados indican una alta precisión del modelo, especialmente cuando la intersección sobre unión (IoU) entre predicciones y cajas reales es de al menos el 50%. El entrenamiento y la validación en Google Colab fueron etapas cruciales para superar las limitaciones de hardware y obtener un modelo con un rendimiento prometedor.

Resultados

A continuación describiremos los resultados obtenidos en las siguientes categorías:

Adquisición de fuentes de datos

Todas las fuentes de datos utilizadas para el proyecto son de fuentes públicas disponibles para su descarga en portales web.

Datos vectoriales

Los datos vectoriales de la BTN se descargaron los archivos ZIP por provincias del ámbito de estudio, que contenían shapefiles para diversas categorías geográficas.

Base Topográfica Nacional (CNIG)

[https://centrodedescargas.cnig.es/CentroDescargas\(btn](https://centrodedescargas.cnig.es/CentroDescargas(btn)

<https://www.idee.es/csw-inspire-idee/srv/spa/catalog.search?#/metadata/spaignBTN>

La cartografía de balsas de REDIAM se obtuvo como un archivo en formato GeoParquet.

REDIAM (Junta de Andalucía)

<https://www.juntadeandalucia.es/medioambiente/portal/acceso-rediam>

https://portalrediam.cica.es/descargas/index.php/s/descargas?dir=/10_SISTEMAS_PRODUCTIVOS/06_RECURSOS_HIDROLOGICOS/Balsas_2020_SIPNA20_2024_02/InfGeografica/InfVectorial/Geoparquet

Datos vectoriales

Las imágenes aéreas, Ortofotos PNOA Falso Color Infrarrojo, y sus marcos asociados se obtuvieron del Centro de Descarga del Instituto Geográfico Nacional en formato GeoTiff COG para las imágenes y shapefile para los marcos.

Ortofotos PNOA Falso Color Infrarrojo (CNIG)

<https://centrodedescargas.cnig.es/CentroDescargas/ortofoto-pnoa-falso-color-infrarrojo>

Descripción de fuentes de datos

Los marcos de las imágenes PNOA se cargaron en un GeoDataFrame, conteniendo y se almacenaron como marcos independientes y como un marco general uniendolos todos, lo que configuró el Área de estudio. Un total de 42 marcos que corresponden con las 42 imágenes aéreas de PNOA.

De la BTN, se seleccionaron shapefiles específicos con diversa información geográfica vinculada con categorías relacionadas con el agua, o con entidades con posible confusión con balsas de agua: Embalses, Lagunas, Almacenamiento de Agua (Piscinas, Balsas/Albercas/Estanques, Vaso de Salina), Construcciones Hidrográficas (ETAP/EDAR/DESALINIZADORAS), Ríos, Canales, Depósitos, Explotaciones mineras (muchas tienen balsas), Almacenes de residuos e Instalaciones deportivas. Estas capas contienen geometrías de tipo polígono o línea.

De la cartografía de balsas de REDIAM se extrajeron las identificadas como "*Balsa de riego o ganadera*".

De las características de las ortofotos ya se ha hablado en el apartado de Materiales y métodos > Datos utilizados > Datos Raster.

Análisis exploratorio de datos o EDA

El EDA se realizó sobre los datos vectoriales preprocesados de balsas y marcos de imágenes. Se analizó el conjunto de datos de balsas filtradas, con un total de 1115 polígonos correspondientes con perímetros de las balsas dentro del área de estudio, con sistema de coordenadas EPSG:25830.

Los marcos de imágenes (42 en total) también están en sistema de coordenadas EPSG:25830 y cubren una extensión de 5800 km². Se calcularon las métricas de características geométricas para las balsas: área, perímetro, compacidad y relación área/perímetro. El análisis de distribución de las áreas de los polígonos de las balsas, demostró un sesgo hacia valores bajos (1055-10600 m²), indicando predominancia de balsas pequeñas, con algunos outliers de áreas significativamente mayores. La relación perímetro-área reveló una heterogeneidad significativa en las formas de las balsas. El análisis de correspondencia demostró que los 42 marcos cubren todas las balsas, con una distribución desigual del número de balsas por marco (entre 37 y 100 en los top 10). La densidad espacial de las balsas varía, mostrando agrupaciones intensas en algunas áreas, que coinciden con zonas agrícolas, reforzando la hipótesis de su uso.

En cuanto a las imágenes PNOA, se confirmaron las características de las versiones reducidas: sistema de coordenadas EPSG:25830, resolución de 1x1 m, consistencia en la fecha de captura (julio de 2022), dimensiones consistentes, tamaño 250-350 MB y 3 bandas cada una (Infrarrojo, rojo, verde).

El EDA ha puesto de manifiesto que aunque los datos están bien estructurados la variabilidad en tamaño y forma de las balsas es un factor clave a considerar, ya que podría plantear desafíos para el entrenamiento del modelo YOLOv8.

Preprocesado de datos

Preprocesado de Datos Vectoriales

El objetivo principal de este proceso fué preparar y filtrar los datos vectoriales para su uso posterior. Esto implica varias etapas:

- **Delimitación del Ámbito de Estudio:** Preprocesado de los marcos de las imágenes PNOA. Estos marcos se unieron para definir el ámbito de estudio del proyecto.
- **Integración y Filtrado de Datos BTN:** Se abordó la integración y el filtrado de datos de la Base Topográfica Nacional (BTN) del CNIG, centrándose en capas relacionadas con superficies de agua. Estas se agrupan en GeoDataFrames únicos por categoría y se exportaron a archivos `.gpkg`, para su visualización en software GIS (Qgis).
- **Filtrado de Balsas:** Se aplicaron varios filtros para seleccionar los polígonos más representativos de balsas para el proyecto.
 - Se filtro la capa de Almacén de Agua de la BTN según atributos específicos como tipo y estado para aislar almacenamientos de agua en uso, incluyendo balsas. Se eliminaron las balsas en parcelas urbanas (no representativas o erróneas).
 - Se realizó un filtrado espacial utilizando datos de balsas de la REDIAM. Las geometrías de la BTN (Balsas/Albercas/Estanques) que intersectan con estas balsas de la REDIAM son seleccionadas, utilizando la mejor definición geométrica de la BTN y el filtro semántico de la REDIAM (selección de balsas de uso agrícola o ganadero).
 - Se aplicó un filtrado por superficie mínima (200 m²) para eliminar balsas pequeñas que podrían introducir ruido en el entrenamiento.
 - Se realizó un filtrado visual, eliminando manualmente balsas consideradas inapropiadas por diversas razones (inexistentes, abandonadas, etc...)
 - Finalmente, se asignó a cada balsa la referencia (HMTN25) de la ortofoto PNOA que le ofrece mejor cobertura.

- **Guardado y Visualización:** Los GeoDataFrames resultantes de cada etapa de filtrado se guardan en el formato vectorial `.gpkg` para su análisis visual en software GIS.

Preprocesado de Datos Ráster

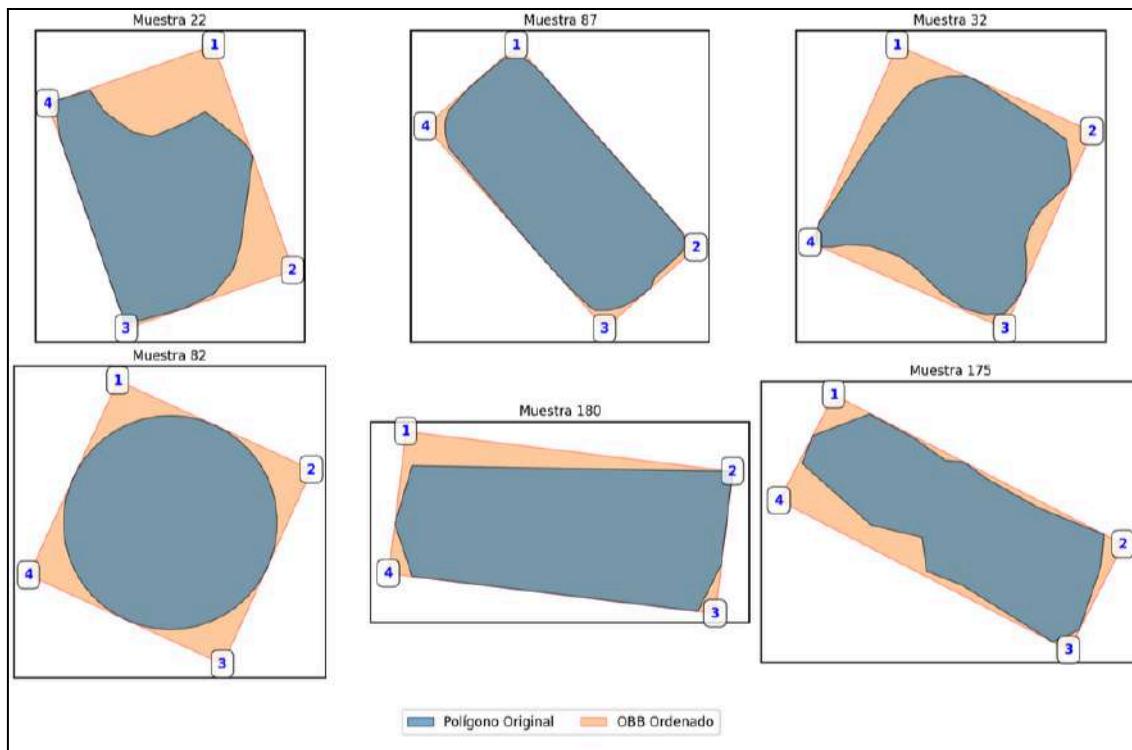
El objetivo principal es preparar y transformar las imágenes ráster en bruto para facilitar su procesamiento posterior.

- **Extracción de Niveles de Menor Resolución:** Para optimizar el procesamiento y reducir la carga computacional, se extrajo un nivel de menor resolución de las imágenes GeoTIFF COG (Cloud Optimized GeoTIFF) originales. Se decide utilizar el nivel 2 de la pirámide, con un factor de reducción de 4, lo que resulta en una resolución de 1 metro por píxel, considerado un buen compromiso entre detalle y eficiencia para la detección de balsas.
- **Generación de Versiones Reducidas:** Se generó una versión reducida de cada Imágenes GeoTIFF COG original.
- **Creación de un Virtual Raster (VRT):** Se creó un Virtual Raster (VRT) que combina todas estas imágenes reducidas en un único archivo virtual. Esto facilitó el acceso y el procesamiento unificado de la cobertura ráster completa del área de estudio.

Preparación de Datos de Entrenamiento

En esta etapa se utilizaron los datos vectoriales de las balsas filtrados y las imágenes ráster preprocesadas, unidas virtualmente en el VRT, para preparar el dataset de entrenamiento para YOLOv8n-obb.

- **Cálculo de Bounding Boxes (BB) y Oriented Bounding Boxes (OBB):** Se calcularon las envolventes rectangulares mínimas orientadas (OBB) para cada polígono de balsas. Se creó una función para obtener los OBB y ordenar sus vértices. Los OBB proporcionan una delimitación más precisa que los BB, especialmente para objetos con orientaciones aleatorias.



Ejemplo del ajuste de los OBB (Oriented Bounding Box) a las geometrías de las balsas de entrenamiento.

- **Generación de Cuadrados de Recorte:** Se creó una función para generar los cuadrados de recorte de tamaño fijo (640x640 píxeles, como recomienda Ultralytics para Yolov8). Se priorizó minimizar la cantidad de cuadros necesarios para cubrir toda la distribución de las balsas.
- **Recorte de Imágenes Ráster:** Se extrajeron las porciones de las imágenes PNOA (a partir del VRT preprocesado) utilizando las geometrías de los cuadrados de recorte. Cada recorte se guardó como un archivo GeoTIFF para utilizarlo posteriormente en el entrenamiento (junto con las '*labels*').
- **Creación de Etiquetas ('*labels*') en Formato YOLOv8n-obb:** Se generaron las etiquetas en archivos de texto ('.txt') con las coordenadas normalizadas de los vértices de los OBB de las balsas dentro de cada recorte de imagen, siguiendo el formato específico de YOLOv8n-obb. Los vértices se ordenaron en sentido horario empezando por la esquina superior izquierda (Como recomienda Ultralytics para Yolov8).

```
431.txt x
0 0.434799 0.875495 0.467055 0.912359 0.430786 0.944094 0.398530 0.907230
0 0.780326 0.059682 0.832400 0.119257 0.794935 0.152005 0.742861 0.092430
0 0.842287 0.055359 0.869765 0.087568 0.813178 0.135842 0.785701 0.103633
0 0.228617 0.410318 0.242894 0.485131 0.143977 0.504008 0.129700 0.429196
```

Ejemplo del contenido de un archivo de etiquetas OBB (Oriented Bounding Box) para el entrenamiento de YOLOv8.

Cada fila representa una detección. El primer número indica el índice de la clase detectada (en este caso, 0 corresponde a la categoría 'balsas'). Los ocho números siguientes son las coordenadas normalizadas (entre 0 y 1) de los cuatro vértices del OBB, listados en sentido horario o antihorario.

- **Estructuración del Dataset:** Finalmente, los recortes de imágenes y sus correspondientes archivos de etiquetas se organizaron en una estructura de directorios para entrenamiento (`train`), validación (`val`) y prueba (`test`), con una división típica de 70%, 20% y 10% respectivamente. Se creó el archivo YAML (`data.yaml`) que define las rutas a estos directorios y la lista de clases (en nuestro caso solo una: balsa) para que YOLOv8 pueda utilizar el dataset.

El preprocessado de datos se construyó como una secuencia de pasos para la transformación de los datos vectoriales y ráster brutos en formatos adecuados y optimizados para el entrenamiento del modelo YOLOv8n-obb.

Selección de variables

Las "variables" de entrada para el modelo de deep learning Yolov8n-obb son las imágenes aéreas (datos ráster) preprocessadas y recortadas por las zonas con presencia de balsas y las etiquetas o 'labels' con las coordenadas de los OBB de las balsas que contiene cada imagen. La "variable" de salida es la representación de la balsa detectada mediante el Oriented Bounding Box (OBB).

Partición de los datos en conjuntos train, val y test

El dataset (recortes de imagen + etiquetas) se dividió en tres conjuntos: entrenamiento, validación y test. Se utilizaron las proporciones estándar: 70% para entrenamiento, 20% para validación y 10% para prueba. Estas divisiones se organizaron en una estructura específica de directorios que conformaron el dataset de entrenamiento, y debe ir acompañada de un archivo YAML (`data.yaml`) que define las rutas a estos directorios y la lista de clases para que YOLOv8 pueda utilizar el dataset.

Esta división asegura datos suficientes para el aprendizaje, la monitorización del rendimiento durante el entrenamiento y una evaluación final independiente.

Se utilizaron un total de 593 imágenes GeoTiff de 640x640 píxeles, que contenían las 1115 balsas, y sus correspondientes archivos de etiquetas que se distribuyeron al azar en los conjuntos de entrenamiento, validación y test en las proporciones indicadas anteriormente.

Estructura del Dataset

El dataset debe estar organizado en una estructura jerárquica como la siguiente:

```
dataset/
  └── images/
      ├── train/      # Imágenes para entrenamiento
      ├── val/        # Imágenes para validación
      └── test/       # Imágenes para prueba
  └── labels/
      ├── train/      # Etiquetas para entrenamiento
      ├── val/        # Etiquetas para validación
      └── test/       # Etiquetas para prueba
```

Cada imagen en las carpetas `images` debe tener un archivo de texto correspondiente en las carpetas `labels`, con el mismo nombre pero extensión `.txt`. Estos archivos contienen las anotaciones (etiquetas) en formato YOLO-obb, que describen las posiciones y clases de los objetos presentes en la imagen.

Estructura del dataset para entrenamiento de modelos Yolo

Aplicación del modelo

Se utilizó un modelo **YOLOv8n-obb** para la detección de balsas. Se optó por la variante 'nano' (n) por ser ligera y eficiente en cómputo, y por la variante OBB para detectar objetos con orientación aleatoria, lo cual es relevante para las balsas en imágenes aéreas.

Para acelerar el entrenamiento y mejorar la convergencia con un dataset específico, se empleó transfer learning partiendo de un modelo YOLOv8n-obb preentrenado. Este enfoque permitió aprovechar características aprendidas previamente en grandes conjuntos de datos, reduciendo considerablemente el tiempo de entrenamiento y mejorando la capacidad de generalización del modelo.

El modelo YOLOv8n-obb combina la capacidad de detectar objetos rotados (fundamental para balsas) con una arquitectura optimizada para recursos mínimos, convirtiéndolo en una buena elección para proyectos sin acceso a GPU. Además, su integración en el ecosistema Ultralytics garantiza soporte técnico y flexibilidad para adaptarse a diferentes escenarios de detección. A diferencia de versiones anteriores, YOLOv8 adopta un diseño anchor-free, lo que simplifica el proceso de detección y reduce la carga computacional. Esto beneficia directamente al modelo YOLOv8n-obb,

haciendo que sea más rápido y adecuado para aplicaciones prácticas con limitaciones técnicas.

Es importante señalar que YOLOv8 está disponible en cinco variantes principales: Nano (n), Pequeño (s), Mediano (m), Grande (l) y Extra Grande (x). Estos modelos ofrecen un equilibrio entre velocidad y precisión: mientras que YOLOv8n es el más rápido y ligero, los modelos superiores como YOLOv8l o YOLOv8x proporcionan mayor precisión, especialmente en condiciones complejas o con objetos poco visibles, aunque a costa de requerir más recursos computacionales.

El entrenamiento se llevó a cabo en Google Colab debido a la necesidad de una GPU potente, no disponible localmente. Algunos de los parámetros utilizados en el entrenamiento fueron: 100 epoch, tamaño de lote (batch size) de 8, tamaño de imagen de 640x640 píxeles, optimizador AdamW, y pesos específicos para las pérdidas de box, clasificación y distribución (box=7.5, cls=0.5, dfl=1.5). La configuración del dataset se definió en un archivo YAML (Anexo 2: Archivo YALM).

El resto de parámetros de entrenamiento se detalla en el anexo del notebook "*06 Entrenamiento y Validación del Modelo (Google Colab)*". Se utilizó un modelo **YOLOv8n-obb** preentrenado cargado desde una ruta en Google Drive. La configuración del entrenamiento se realizó mediante el método `model.train()` de la librería Ultralytics, especificando varios parámetros:

```

# Entrenar el modelo
results = model.train(
    data="dataset/data.yaml",           # Ruta al archivo YAML
    epochs=100,                         # Aumentado para mejor convergencia
    patience=15,                        # Mayor paciencia para evitar paradas prematuras
    imgsz=640,                          # Tamaño estándar para YOLOv8
    batch=8,                            # Reducido para evitar problemas de memoria en T4
    optimizer="AdamW",                  # Optimizer más efectivo para detección de objetos
    lr0=0.01,                           # Learning rate inicial
    lrf=0.001,                          # Learning rate final
    momentum=0.937,                     # Momentum para el optimizador
    weight_decay=0.0005,                 # Weight decay para reducir overfitting
    warmup_epochs=3,                     # Epochs de calentamiento
    warmup_momentum=0.8,                # Momentum durante el calentamiento
    warmup_bias_lr=0.1,                 # Learning rate para bias durante calentamiento
    box=7.5,                            # Peso para pérdida de bounding box
    cls=0.5,                            # Peso para pérdida de clasificación
    dfl=1.5,                            # Peso para pérdida de distribución
    plots=True,                         # Generar gráficos
    save=True,                          # Guardar checkpoints
    cache=False,                        # No usar cache (para evitar problemas de memoria)
    device=0,                            # Usar GPU
    workers=4,                           # Aumentado ligeramente para mejor paralelismo
    project="deteccion_balsas",         # Nombre del proyecto
    name="yolov8n_obb_run1",            # Nombre del experimento
    exist_ok=True)                      # Continuar entrenamientos interrumpidos
)

```

Parámetros utilizados para el entrenamiento del modelo Yolov8n-obb

Tras aproximadamente 47 minutos de entrenamiento, se monitorizaron las pérdidas (box_loss, cls_loss, dfl_loss) y métricas del conjunto de validación. Al finalizar, se generaron automáticamente una serie de gráficos que muestran el desempeño del modelo YOLOv8n-obb durante el entrenamiento. Estas visualizaciones incluyen la matriz de confusión, las curvas F1-Confianza, Precisión-Recall y Recall-Confianza, así como representaciones sobre la distribución de los OBBs detectados. Además, se obtuvieron gráficos de evolución de las pérdidas y métricas a lo largo de los epochs. Estas imágenes, creadas con bibliotecas como matplotlib y seaborn, permiten evaluar la precisión, eficiencia y capacidad de generalización del modelo. El detalle específico de cada figura se explica en el Anexo 3: Visualizaciones del Entrenamiento.

Validación e inferencia del modelo

Validación

La validación del modelo se realizó utilizando el conjunto de validación (val), que representa el 20% de los datos totales. Este conjunto se utiliza para monitorear el

rendimiento del modelo durante el entrenamiento y ajustar los hiperparámetros. Después del entrenamiento, se evaluó el modelo con el mejor checkpoint guardado (`best.pt`) en el conjunto de validación utilizando el método `model.val()`.

Las **métricas clave** obtenidas en el conjunto de validación fueron:

- **mAP@0.5 (mean Average Precision a IoU=0.5): 0.897 (89.7%)**, que indica una alta precisión del modelo cuando una predicción tiene una Intersección sobre Unión (IoU) con la caja real de al menos el 50%.
- **mAP@0.5:0.95 (mean Average Precision a IoU entre 0.5 y 0.95): 0.574 (57.4%)**, es una métrica más estricta que evalúa la precisión promedio en diferentes umbrales de IoU.
- **Precisión (P): 0.883 (88.3%)**, es la proporción de predicciones correctas entre todas las predicciones realizadas.
- **Recall (R): 0.879 (87.9%)**, es la proporción de objetos reales detectados correctamente.

Estos resultados en el conjunto de validación demuestran un **muy buen desempeño del modelo YOLOv8n-obb para la tarea de detección de balsas con bounding boxes orientados**. La alta precisión y el buen recall sugieren que el modelo es capaz de detectar y localizar las balsas de manera efectiva. La rapidez de inferencia (~20 ms por imagen) también es destacada.

El entrenamiento y la validación en Google Colab fueron etapas cruciales para superar las limitaciones de hardware local y obtener un modelo de detección de balsas con un rendimiento tan prometedor. Los resultados de la validación proporcionan una buena referencia del rendimiento esperado del modelo en datos 'no vistos'. Aún así, consideramos que con una limpieza adicional de los datos y posibles modificaciones y ajustes en el entrenamiento se podrían alcanzar métricas aún mejores.

Inferencia

1. Contextualización y Objetivos de la Inferencia

La inferencia, implica la aplicación del modelo entrenado a nuevas imágenes para generar predicciones sobre la ubicación y extensión de las balsas. Además de la detección, se busca la conversión de estas predicciones en datos espaciales georreferenciados, lo que potencia su utilidad para el análisis geográfico y la gestión de recursos hídricos.

El objetivo principal de esta fase es la validación del modelo. Esto implica evaluar su capacidad para generalizar datos no vistos durante el entrenamiento,

asegurando que el modelo no sufre de sobreajuste (overfitting) y que sus predicciones son robustas y confiables.

2. Optimización de la Inferencia con SAHI

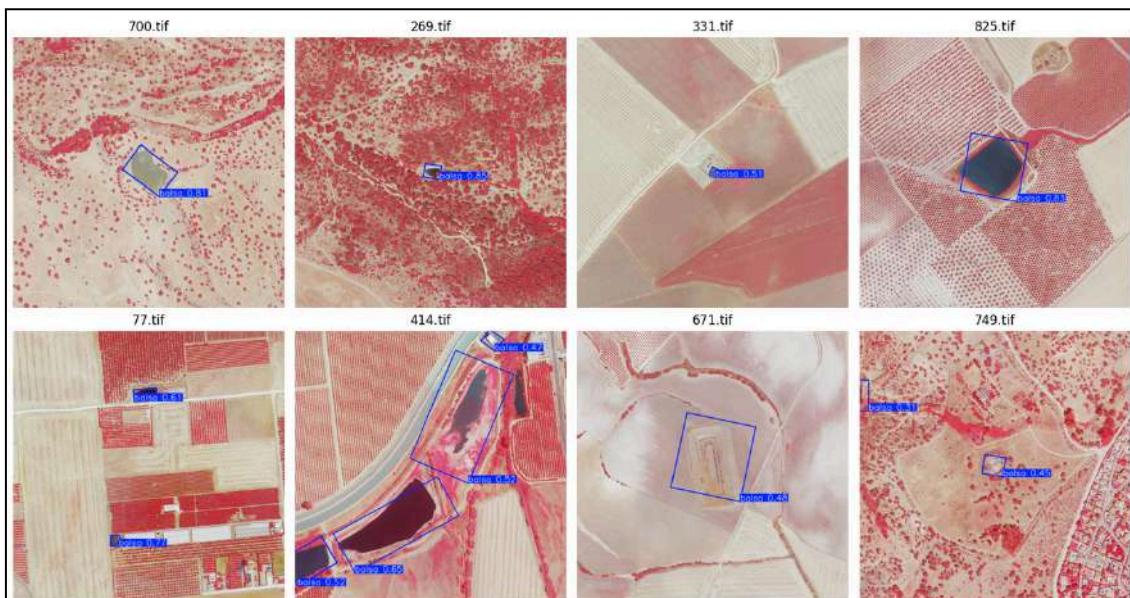
Un componente crítico de la metodología utilizada es el uso de la librería **SAHI** (Slicing Aided Hyper Inference). Esta herramienta se emplea para optimizar el proceso, abordando desafíos específicos asociados con el procesamiento de imágenes aéreas de alta resolución.

- **Manejo de Imágenes de Gran Tamaño:** Las imágenes aéreas suelen presentar grandes dimensiones y cubren extensas superficies, lo que puede sobrecargar la memoria del sistema durante el procesamiento. SAHI solventa este problema mediante la división de las imágenes en porciones más pequeñas, denominadas *slices*, permitiendo un procesamiento secuencial y una reducción significativa del consumo de memoria.
- **Mejora en la Detección de Objetos Pequeños:** La inferencia basada en *slices* incrementa la probabilidad de que los objetos pequeños ocupen una porción sustancial de la ventana de entrada del modelo. Este enfoque puede mejorar la sensibilidad del modelo a objetos que podrían pasar desapercibidos o ser detectados con un bajo umbral de confianza al procesar la imagen completa a una escala reducida.
- **Reducción de Artefactos en los Bordes:** Para mitigar los efectos de borde al dividir la imagen, SAHI permite el solapamiento entre *slices* adyacentes. Este solapamiento asegura que los objetos situados en los límites de las *slices* sean detectados en su totalidad en al menos una de ellas. Posteriormente, técnicas de post-procesamiento como GREEDYNMM (con la métrica IOS - Intersection Over Smallest) se aplican para fusionar las detecciones redundantes y generar un conjunto de predicciones coherente para la imagen completa.
- **Flexibilidad de Configuración:** SAHI ofrece la posibilidad de ajustar parámetros clave, como las dimensiones de las *slices* y el grado de solapamiento. Esta flexibilidad permite adaptar el proceso de inferencia a las características específicas de las imágenes y a los recursos computacionales disponibles.

3. Implementación de la Inferencia

La implementación de la inferencia se lleva a cabo mediante una serie de pasos automáticos, utilizando librerías de Python como Ultralytics y SAHI.

- **Carga del Modelo Entrenado:** Se carga el modelo YOLOv8n-obb previamente entrenado, utilizando la ruta al archivo que contiene los pesos del modelo.
- **Inferencia en Imágenes de Prueba:** Se realiza la inferencia sobre un conjunto de imágenes de prueba para obtener las predicciones del modelo. En el notebook *07 Inferencia y Evaluación del Modelo*, se pueden hacer predicciones sobre una muestra aleatoria de 8 imágenes de test, lo que permite una evaluación cualitativa rápida del rendimiento del modelo.



Inferencia sobre imágenes no vistas por el modelo en el entrenamiento. 8 recortes de imágenes aéreas del grupo de test con los resultados de la inferencia.

Como parte del proceso de evaluación del modelo entrenado, se llevó a cabo una inferencia completa sobre el conjunto de 160 imágenes no utilizadas durante el entrenamiento, pertenecientes a la carpeta test. Los resultados de esta inferencia se han incluido en el Anexo 4: Resultado de la Inferencia Sobre Imágenes de Test, donde se muestran las imágenes procesadas junto con las detecciones realizadas por el modelo. Este anexo sirve como referencia visual para analizar tanto los aciertos como las limitaciones del modelo en distintas condiciones ambientales, texturales y de ocultación.

Revisando los resultados de la inferencia, es posible identificar patrones comunes en las detecciones, como casos de balsas no detectadas debido a reflejos o vegetación, detecciones parciales, falsos positivos o detecciones dobles de una misma.

- **Inferencia sobre Imágenes Completas:** Para procesar imágenes de mayor tamaño, se implementa un flujo de trabajo interactivo que permite al usuario

seleccionar una imagen específica, de entre las utilizadas en el entrenamiento, para la inferencia. Se utiliza la librería ipywidgets para crear un selector que facilita la elección de la imagen por parte del usuario.

- **Configuración de los Parámetros de Inferencia:** Durante la fase de configuración de los parámetros clave para la inferencia con SAHI, se definió el umbral de confianza para las predicciones y el porcentaje de solapamiento entre *slices*. Estos parámetros son esenciales para equilibrar la precisión, el recall y el tiempo computacional en la detección de objetos, especialmente en escenarios complejos como la identificación de balsas.

Los resultados de validación del modelo YOLOv8n-obb indicaron que:

- El F1-score máximo (0,88) se alcanza con un umbral de confianza del 48,8%, lo que representa el mejor equilibrio teórico entre precisión y recall.
- La máxima precisión (1,0) ocurre alrededor del 88,2%, pero esto implica una reducción significativa del recall.
- Por otro lado, el recall comienza a caer drásticamente a partir del 55%, lo que indica que umbrales más altos pueden dejar de detectar instancias reales.

Aunque el umbral óptimo teórico según el F1-score es ~48,8%, se optó por elevarlo hasta el 60% tras un análisis visual de los resultados preliminares. Esta decisión se basó en las siguientes observaciones:

1. Reducción de falsos positivos: Umbrales muy bajos (como el 48,8%) generaban un número elevado de detecciones erróneas, muchas de las cuales no se correspondían con balsas reales.
2. Evitar pérdida de detecciones: Con umbrales superiores al 70%, se observó una disminución de la capacidad del modelo para detectar balsas.
3. Valor intermedio razonable: El 60% se consideró un compromiso adecuado, ya que:
 - Evita perder balsas relevantes.
 - Reduce el número de falsos positivos respecto a umbrales más bajos.

Para reducir aún más los posibles errores cometidos por el modelo a este umbral, se implementó un postprocesado adicional mediante capas de filtrado derivadas de la Base de Datos Nacional (BTN), como se comenta en el apartado:4. Post-Procesamiento y Análisis Espacial.

- **Ejecución de la Inferencia con SAHI:** Se realiza la inferencia utilizando la función `get_sliced_prediction` de la librería SAHI. Esta función implementa el proceso de dividir la imagen en *slices*, realizar la inferencia en cada *slice* y ensamblar las predicciones para obtener el resultado final.
- **Análisis de los Resultados de la Inferencia:** Los resultados de la inferencia se almacenan en una lista de objetos de predicción (*ObjectPrediction*), donde cada objeto contiene información sobre la ubicación, la máscara, la confianza y la categoría del objeto detectado. En el notebook, se proporciona una explicación detallada de la estructura de estos objetos (*ObjectPrediction*), destacando la importancia del atributo "*mask*" para la obtención de Oriented Bounding Boxes (OBB) de las predicciones.

4. Post-Procesamiento y Análisis Espacial

Una vez obtenidas las predicciones, se llevan a cabo operaciones de post-procesamiento para refinar los resultados y prepararlos para el análisis espacial.

- **Conversión a Geometrías Georreferenciadas:** Las predicciones, que inicialmente se expresan en coordenadas de píxeles, se transforman a coordenadas en sistema de coordenadas EPSG:25830 utilizando la información de la imagen original. Esto permite representar las ubicaciones de las balsas en el mismo sistema de coordenadas que las Ortofotos PNOA y las balsas originales, facilitando la comparación de los resultados.
- **Filtrado de Predicciones:** Se aplica un proceso de filtrado espacial para eliminar las predicciones que se solapan con entidades geográficas que se sabe que no son balsas (embalses, ríos, canales...). Para ello, se utilizan capas de información geográfica vectorial (extraídas de la BTN) y operaciones de intersección espacial.
- **Visualización de los Resultados:** Los resultados finales se visualizan en un mapa interactivo, lo que facilita la interpretación y el análisis de las detecciones. El mapa muestra el ámbito de la imagen seleccionada para la inferencia junto con los OBBs de las balsas detectadas y los OBBs de las

balsas utilizadas para el entrenamiento sobre un WMS de las Ortofotos PNOA en Falso Color.

La inferencia del modelo se ejecutó con una metodología robusta que combina técnicas de procesamiento de imágenes, análisis geoespacial y visualización interactiva. SAHI demuestra ser una herramienta clave para superar las limitaciones de memoria y mejorar la detección de objetos pequeños, aunque requiere ajustes en escenarios complejos.

Comparativa de resultados

Aunque no se realizó una comparación explícita con otros modelos o enfoques, los resultados obtenidos con el modelo YOLOv8n-obb entrenado son muy buenos en el conjunto de validación. El mAP@0.5 cercano al 90% sugiere que el modelo es capaz de identificar correctamente la gran mayoría de las balsas con alta confianza. El mAP@0.5:0.95 superior al 50% indica que la localización espacial de las predicciones también es buena. Sin embargo, durante la inferencia sobre imágenes completas, se observó que el modelo puede tener dificultades con balsas muy grandes (poco frecuentes y poco representadas en el entrenamiento), a veces prediciéndolas como múltiples balsas más pequeñas y en ocasiones solapadas, mientras que funciona bastante bien con balsas medianas y pequeñas. Esto sugiere que la heterogeneidad en el tamaño de las balsas identificada en el EDA presenta un desafío que el modelo no maneja completamente en su configuración actual. La presencia de outliers en el área de las balsas podría ser un factor contribuyente. Una limpieza adicional de datos o ajustes en el entrenamiento podrían mejorar el manejo de estas variaciones.

Conclusiones

Este trabajo ha demostrado la viabilidad del uso de técnicas avanzadas de deep learning, específicamente el modelo YOLOv8n-obb, para la detección automatizada de balsas de agua en imágenes aéreas de alta resolución obtenidas del Programa Nacional de Ortofotografía Aérea (PNOA) en Falso Color Infrarrojo. El desarrollo e implementación de un pipeline completo, desde el preprocesamiento de datos vectoriales y ráster hasta el entrenamiento, validación e inferencia del modelo, han permitido obtener resultados prometedores en términos de precisión, eficiencia computacional y aplicabilidad práctica.

Se realizó un preprocesamiento exhaustivo de datos vectoriales y ráster, integrando información de la BTN, REDIAM y los marcos de las Ortofotos PNOA, y optimizando las imágenes para el procesamiento. El EDA proporcionó una visión detallada de las características espaciales y geométricas de las balsas, resaltando la variabilidad en su tamaño y forma. La preparación del dataset, incluyendo el cálculo de OBBs y la generación de recortes etiquetados, fue fundamental para el entrenamiento del modelo YOLOv8n-OBB. El entrenamiento en Google Colab, utilizando un modelo ligero y transfer learning, demostró ser eficiente y necesario dadas las limitaciones de hardware local. Los resultados de validación del modelo YOLOv8n-obb son muy prometedores, con métricas mAP elevadas que confirman su capacidad para detectar y localizar balsas con alta precisión. No obstante, se observó que el modelo entrenado predice balsas grandes como entidades más pequeñas y solapadas, indicando un área de mejora potencial. El proyecto valida el uso de YOLOv8n-OBB y SAHI para la detección de objetos orientados en imágenes aéreas de alta resolución.

Para futuras mejoras, se podría explorar una mayor limpieza de datos, especialmente en relación con las balsas grandes y la heterogeneidad de la categoría "*Almacenamiento de Agua*", establecer distintas categorías de balsas según sus características, así como el ajuste de hiperparámetros de entrenamiento o la exploración de otras arquitecturas de modelos. También se debería explorar el uso de modelos más avanzados dentro de la familia YOLOv8, que ofrecen una mayor precisión en la detección de objetos, especialmente en condiciones difíciles, aunque requieren más recursos computacionales.

En conclusión, este TFM ha demostrado cómo la combinación de imágenes aéreas de alta resolución, modelos de deep learning avanzados y herramientas de procesamiento geoespacial puede servir como una solución eficaz y escalable para la detección automatizada de balsas de agua. Este enfoque tiene un gran potencial para

aplicarse en tareas relacionadas con la gestión sostenible de recursos hídricos, el monitoreo ambiental y la planificación territorial, especialmente en regiones con alta dependencia agrícola del riego.

La metodología desarrollada puede ser fácilmente adaptada para la detección de otros elementos cartográficos visibles en las ortofotos PNOA y con representación cartográfica en la Base Topográfica Nacional (BTN), extendiendo así su utilidad más allá del caso específico de las balsas. Por tanto, este trabajo representa un primer paso hacia la automatización de tareas cartográficas y ambientales mediante inteligencia artificial.

Glosario

AdamW Optimizer

Algoritmo de optimización utilizado durante el entrenamiento para ajustar los pesos del modelo, combinando descenso de gradiente adaptativo con decaimiento de peso.

Análisis Exploratorio de Datos (EDA)

Proceso inicial de inspección estadística y visualización de datos para comprender distribuciones, variabilidad y patrones en las balsas y marcos de imágenes.

Anchor-free

Enfoque en YOLOv8 que elimina la necesidad de "anclas" predefinidas para detectar objetos, simplificando el proceso y reduciendo carga computacional.

Archivo YAML

Fichero de configuración que define rutas y parámetros para entrenamiento en YOLOv8, incluyendo divisiones de datos y clases.

Base Topográfica Nacional (BTN)

Conjunto de datos cartográficos oficiales de España, utilizado para obtener geometrías de balsas y otras entidades hidrográficas (embalses, ríos, etc.).

Batch Size

Número de muestras procesadas antes de actualizar los pesos del modelo durante el entrenamiento.

Bounding Box (BB)

Rectángulo alineado con ejes que delimita un objeto en una imagen. Se utiliza como base para detección de objetos, aunque su precisión es menor que la de OBBs.

Cloud Optimized GeoTIFF (COG)

Versión de GeoTIFF diseñada para acceso eficiente en la nube, con niveles de pirámide integrados para visualización a diferentes escalas.

Cookiecutter

Librería de código abierto utilizada para crear proyectos a partir de plantillas predefinidas. Permite generar estructuras de directorios y archivos personalizadas a partir de un modelo, facilitando la creación consistente y rápida de nuevos proyectos.

EPSG:25830

Sistema de coordenadas UTM (Zona 30N) utilizado para georreferenciar ortofotos PNOA, balsas y predicciones.

Epoch

Una pasada completa sobre todo el conjunto de datos durante el entrenamiento.

Folium

Biblioteca de Python para crear mapas interactivos basados en Leaflet, empleada para visualizar predicciones georreferenciadas.

GeoDataFrame

Estructura de datos de la biblioteca geopandas que almacena geometrías (puntos, líneas, polígonos) junto con atributos, empleada para manejar capas vectoriales georreferenciadas.

Geoparquet

Formato de almacenamiento columnar optimizado para datos geoespaciales, utilizado para cargar la cartografía de balsas de REDIAM.

Geotransformación

Matriz que define la relación entre coordenadas de píxeles en una imagen y coordenadas geográficas (UTM), esencial para georreferenciar predicciones.

Georeferenciación

Proceso de asignar coordenadas geográficas a datos digitales, permitiendo su integración en sistemas SIG.

GeoTIFF

Formato de imagen rasterizada con metadatos geoespaciales integrados, utilizado para almacenar ortofotos PNOA y predicciones.

GPU (Graphics Processing Unit)

Hardware acelerador utilizado en Google Colab para entrenar modelos de deep learning, superando limitaciones de recursos locales.

GREEDYNMM (con la métrica IOS - Intersection Over Smallest)

Método de postprocesamiento en SAHI que fusiona predicciones redundantes usando la métrica de solapamiento "Intersección sobre el más pequeño" (IOS).

HMTN25

Código de la hoja cartográfica 1:25.000 del Mapa Topográfico Nacional (MTN25), asociado a cada ortofoto PNOA. Se usó para seleccionar imágenes específicas durante la inferencia.

Inferencia

Aplicación del modelo entrenado a nuevas imágenes para generar predicciones sobre la ubicación y extensión de las balsas.

Intersection Over Union (IoU)

Métrica que cuantifica el solapamiento entre predicciones y anotaciones reales, usada para evaluar precisión.

Learning Rate (Tasa de Aprendizaje)

Parámetro que controla la magnitud de los ajustes en los pesos del modelo durante el entrenamiento.

Loss Functions (box_loss, cls_loss, dfl_loss)

Funciones de pérdida que miden errores en la detección de cajas delimitadoras, clasificación y distribución de coordenadas, respectivamente.

mAP@0.5 (mean Average Precision at IoU=0.5)

Métrica que evalúa la precisión promedio del modelo cuando la intersección sobre unión (IoU) entre predicciones y anotaciones reales es $\geq 50\%$.

mAP@0.5 :0.95

Versión más estricta de mAP, evaluada en umbrales de IoU entre 0.5 y 0.95.

Ortofotos PNOA de Falso Color Infrarrojo

Imágenes aéreas de alta resolución (0,25 m) que combinan las bandas infrarroja, roja y verde para resaltar características hidrológicas. Se emplearon como fuente de datos principal para la detección de balsas.

Overlap Ratio (Ratio de Solapamiento)

Proporción de solapamiento entre *slices* adyacentes en SAHI.

Oriented Bounding Boxes (OBB)

Rectángulos rotados que delimitan objetos en imágenes, adaptándose a su orientación real. Mejoran la precisión en la detección de balsas respecto a bounding boxes tradicionales alineados con ejes.

Overfitting (Sobreajuste)

Fenómeno donde el modelo aprende patrones específicos del conjunto de entrenamiento, perdiendo capacidad de generalización. Se mitigó con validación cruzada y ajuste de hiperparámetros.

Precision (Precisión)

Proporción de predicciones correctas entre todas las detecciones realizadas.

Preprocesamiento de Datos

Transformación de datos brutos (ej.: imágenes y capas vectoriales) en formatos adecuados para el entrenamiento, incluyendo reducción de resolución y cálculo de OBBs.

Postprocesamiento

Pasos posteriores a la inferencia para refinar resultados, como filtrado de predicciones y conversión a geometrías georreferenciadas.

Recall (Recuperación)

Proporción de objetos reales detectados correctamente.

REDIAM (Red de Información Ambiental de Andalucía)

Base de datos pública que proporciona información sobre recursos hídricos en Andalucía, incluyendo balsas de uso agrícola y ganadero. Se usó para filtrar geometrías de la BTN.

Rasterio

Biblioteca para leer y escribir datos ráster geoespaciales, utilizada para obtener la geotransformación de ortofotos.

SAHI (Slicing Aided Hyper Inference)

Biblioteca que divide imágenes grandes en "slices" para inferencia eficiente, reduciendo consumo de memoria y mejorando la detección de objetos pequeños.

Shapely

Biblioteca para manipular geometrías planas (puntos, líneas, polígonos), integrada con GeoPandas para crear polígonos desde máscaras segmentadas.

Shapefile

Formato vectorial tradicional para almacenar datos geográficos, compuesto por múltiples archivos relacionados (.shp, .shx, .dbf). Se utilizó para cargar marcos de imágenes PNOA.

Slice Height/Width

Dimensiones de los fragmentos en los que se divide una imagen para inferencia con SAHI, configurados en 640×640 píxeles.

Software GIS (Sistemas de Información Geográfica)

Herramientas como QGIS o ArcGIS que permiten visualizar, analizar y gestionar datos espaciales. Se emplearon para validar resultados georreferenciados.

Sistema de Coordenadas de Referencia (CRS)

Marco de referencia que define cómo se proyectan las coordenadas geográficas. Se usó EPSG:25830 para mantener la consistencia espacial.

Transfer Learning

Técnica que utiliza un modelo preentrenado (ej.: YOLOv8n-obb) como base para adaptarlo a un nuevo conjunto de datos específico, acelerando el entrenamiento.

Ultralytics

Framework de código abierto ampliamente utilizado para el entrenamiento y la implementación de modelos YOLO (You Only Look Once) de detección de objetos.

Vectorización

Proceso de convertir datos rasterizados en representaciones vectoriales (polígonos, puntos, líneas), esencial para análisis espacial y visualización.

VRT (Virtual Raster)

Archivo virtual que combina múltiples imágenes ráster en una única capa sin duplicar datos físicos, facilitando el acceso y procesamiento unificado.

Web Mercator (EPSG:3857)

Sistema de coordenadas proyectado compatible con servicios web de mapas (ej.: OpenStreetMap), usado para visualizar datos en Folium.

YOLOv8n-obb

Arquitectura ligera de detección de objetos con soporte para OBBs, entrenada mediante transfer learning para identificar balsas en ortofotos.

Zoom Start

Parámetro en Folium que define el nivel inicial de zoom en un mapa interactivo.

Referencias bibliográficas.

Libros

- Barber, D., Bishop, C. M., & Frey, B. J. (2017). *Machine Learning and Pattern Recognition*. Cambridge University Press.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- Lillesand, T., Kiefer, R. W., & Chipman, J. (2015). *Remote Sensing and Image Interpretation*. Wiley.
- Zhang, Y., & Zhou, G. (2020). *Remote Sensing Image Processing Using Deep Learning: A Practical Guide*. CRC Press.
- Cantz M.J. (2010). *Image Analysis, Classifications, and Change Detection in Remote Sensing*. CRC Press, Taylor and Francis.
- Lakshmanan, V., Görner, M., & Gillard, R. (2021). *Practical Machine Learning for Computer Vision*. O'Reilly Media, Inc

Capítulos de Libros

- Chen, L., & Wang, X. (2021). *Deep learning for remote sensing image analysis*. En F. Li & H. Zhang (Eds.), *Advances in Artificial Intelligence and Remote Sensing* (pp. 45–68). Academic Press.
- Khan, S., Hayat, M., & Shah, M. (2022). *Object detection with oriented bounding boxes*. En M. Singh & A. Kumar (Eds.), *Computer Vision and Deep Learning Applications* (pp. 112–130). Springer.

Artículos de Revista

- Bochkovskiy, A., Wang, C.Y., & Liao, H.Y.M. (2020). *YOLOv4: Optimal Speed and Accuracy of Object Detection*. <https://arxiv.org/pdf/2004.10934>
- Liu, X., Yang, Z., Li, M., & Yan, J. (2021). *Arbitrary-Oriented Object Detection with Transformer*. *Pattern Recognition*, 120, 108183. <https://arxiv.org/pdf/2205.12785>
- Redmon, J., & Farhadi, A. (2018). *YOLOv3: An Incremental Improvement*. <https://arxiv.org/pdf/1804.02767>
- Yang, B., Yan, J., Lei, Z., & Li, S. Z. (2019). *Automatic Ship Detection in Remote Sensing Images from Google Earth of Complex Scenes Based on Multiscale Rotation Dense Feature Pyramid Networks*. *Remote Sensing*, 11(1), 1–18. <https://arxiv.org/pdf/1806.04331>

- Zhou, X., et al. (2018). *Towards Arbitrary-Oriented Text Detection with Oriented Bounding Box Attention Network*. IEEE Transactions on Multimedia, 21(8), 2035–2045. <https://arxiv.org/pdf/1711.04226>
- Congalton R.G. (1991). *A review of assessing the accuracy of classifications of remotely sensed data*. *Remote Sensing of Environment*, vol. 37, no. 1, pp. 35–46.
- Samed Ozdemir, et al (2024). *Extraction of Water Bodies from High-Resolution Aerial and Satellite Images Using Visual Foundation Models*. <https://www.mdpi.com/2071-1050/16/7/2995>
- López-Tapia, et al (2021). *Machine learning with high-resolution aerial imagery and data fusion to improve and automate the detection of wetlands*. International Journal of Applied Earth Observation and Geoinformation, 105, 102581. <https://doi.org/10.1016/j.jag.2021.102581>
- Ren, S., et al. (2015). *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. *Advances in Neural Information Processing Systems*. <https://arxiv.org/pdf/1506.01497>
- Hinton GE, Osindero S, Teh YW. (2006) A fast learning algorithm for deep belief nets. *Neural Comput*. 18(7):1527-54. <https://www.cs.toronto.edu/~hinton/absps/fastnc.pdf>
- LALIBERTE A.S., RANGO A., HAVSTAD K.M, et al. (2004). Object-oriented image analysis for mapping shrub encroachment from 1937 to 2003 in southern new mexico. *Remote Sensing of Environment*, vol. 93, nos. 1–2, pp. 198–210.
- Schölkopf, B., Herbrich, R., and Smola, A. J. (2001). A generalized representer theorem. in *International conference on computational learning theory*. Springer, 2001, pp. 416–426.
- Njathi, Y., Kabi, J. N., Wanjiku, L., Kiarie, G., Mugambi, L., & wa Maina, C. (2023). *Efficient Camera Trap Image Annotation Using YOLOv5*. https://repository.dkut.ac.ke:8080/xmlui/bitstream/handle/123456789/8526/Efficient_Camera_Trap_Image_annotation_Using_YOLOv5.pdf?sequence=1

Material Consultado en Internet

- Jane Torres (2024). *YOLOv8 Documentation: A Deep Dive into the Documentation*. <https://yolov8.org/yolov8-documentation/>
- Centro Nacional de Información Geográfica (CNIG). <https://centrodedescargas.cnig.es>
- Congalton, R. G., & Green, K. (2009). *Assessing the Accuracy of Remotely Sensed Data: Principles and Practices*. CRC Press.

<https://www.crcpress.com/Assessing-the-Accuracy-of-Remotely-Sensed-Data-Principles-and-Practices/Congalton-Green/p/book/9781420055122>

- Folium. (2024). Python Library for Interactive Maps. <https://python-visualization.github.io/folium/>
- GDAL/OGR. (2024). Geospatial Data Abstraction Library. <https://gdal.org/>
- Geopandas. (2024). GeoDataFrame Documentation. <https://geopandas.org/>
- QGIS. (2024). Quantum GIS - Open Source Geographic Information System. <https://qgis.org/es/site/>
- Rasterio. (2024). Reading and Writing Geospatial Raster Data. <https://rasterio.readthedocs.io/en/latest/>
- REDIAM – Red de Información Ambiental de Andalucía. (2024). Datos geográficos de balsas agrícolas y ganaderas. https://portalrediam.cica.es/descargas/index.php/s/descargas?dir=/10_SISTEMAS_PRODUCTIVOS/06_RECURSOS_HIDROLOGICOS/Balsas_2020_SIPNA20_2024_02/InfGeografica/InfVectorial/Geoparquet
- SAHI (Slicing Aided Hyper Inference). (2024). Ultralytics Documentos: Utilización de YOLO11 con SAHI para la inferencia fragmentada. <https://docs.ultralytics.com/es/guides/sahi-tiled-inference/>
- Shapely. (2024). <https://shapely.readthedocs.io/en/stable/>
- Ultralytics. (2024). Framework YOLOv8. <https://docs.ultralytics.com/es/models/yolov8/>
- Junta de Andalucía. (2023). Estrategia Andaluza de Adaptación al Cambio Climático (EAACC). <https://www.juntadeandalucia.es/medioambiente/portal/web/cambio-climatico/estrategia-andalucia/plan-andaluz-accion-clima-2021-2030>
- Instituto Geográfico Nacional (IGN). (2024). Marcos cartográficos HMTN25. Recuperado el 18 de abril de 2025, de <https://www.ign.es/web/ign/es>
- EPSG:25830 – Sistema de Coordenadas UTM Zona 30N. (2024). <https://epsg.io/25830>
- Cloud Optimized GeoTIFF (COG). (2024). Especificaciones técnicas. <https://www.cogeo.org/>
- PNOA - IGN (s.f.). Imagen, Especificaciones Técnicas. Instituto Geográfico Nacional. <https://pnoa.ign.es/pnoa-imagen/especificaciones-tecnicas>
- PNOA - IGN (s.f.). Ortofotos PNOA Falso Color Infrarrojo. Instituto Geográfico Nacional. <https://pnoa.ign.es/pnoa-imagen/ortofotos-pnoa-falso-color-infrarrojo>

- PNOA - IGN (s.f.). PNOA Imagen, Catálogo y Casos de Uso. Instituto Geográfico Nacional. <https://pnoa.ign.es/pnoa-imagen/catalogo-y-casos-de-uso>
- Innovatiana. (s.f.). YOLO: un detector de objetos en tiempo real. <https://es.innovatiana.com/post/what-is-yolo-in-ai>
- VisionPlatform.ai. (s.f.). La guía completa para la detección de objetos: Una introducción a la detección en 2024. <https://visionplatform.ai/es/la-guia-completa-para-la-deteccion-de-objetos-una-introduccion-a-la-deteccion-en-2024/>
- Detección de objetos. (s.f.). Wikipedia. https://es.wikipedia.org/wiki/Detecci%C3%B3n_de_objetos
- Roboflow. (s.f.). What is the YOLOv8 Oriented Bounding Boxes Annotation Format?. <https://roboflow.com/formats/yolov8-oriented-bounding-boxes>
- Roboflow. (s.f.). How to Train an Ultralytics YOLOv8 Oriented Bounding Box (OBB). <https://blog.roboflow.com/train-YOLOv8-obb-model/>
- Ultralytics. (s.f.). Intersection over Union (IoU) Explained. <https://www.ultralytics.com/glossary/intersection-over-union-iou>
- Especificaciones de la Base Topográfica Nacional (BTN). <https://www.ign.es/resources/docs/IGNCnig/BTN/ESPBTN.pdf>
- Practical Machine Learning for Computer Vision by Valliappa Lakshmanan, Martin Görner, Ryan Gillard. Chapter 4. Object Detection and Image Segmentation. <https://www.oreilly.com/library/view/practical-machine-learning/9781098102357/ch04.html>

Anexos

Anexo 1: Cookiecutter

Estructura de carpetas generada para el desarrollo del proyecto por la librería cookiecutter

BALSAS



clasificacion de balsas de riego en la cuenca del guadalquivir

Project Organization

```

├── LICENSE           <- Open-source license if one is chosen
├── Makefile          <- Makefile with convenience commands like `make data` or `make train`
├── README.md         <- The top-level README for developers using this project.
├── data              <- Data from third party sources.
│   ├── external       <- Intermediate data that has been transformed.
│   ├── interim        <- The final, canonical data sets for modeling.
│   ├── processed      <- The original, immutable data dump.
│   └── raw             <- A default mkdocs project; see www.mkdocs.org for details
├── docs              <- Trained and serialized models, model predictions, or model summaries
├── models             <- Jupyter notebooks. Naming convention is a number (for ordering),
                           the creator's initials, and a short '-' delimited description, e.g.
                           '1.0-jqp-initial-data-exploration'.
├── pyproject.toml    <- Project configuration file with package metadata for
                           balsas and configuration for tools like black
├── references         <- Data dictionaries, manuals, and all other explanatory materials.
├── reports            <- Generated analysis as HTML, PDF, LaTeX, etc.
│   └── figures         <- Generated graphics and figures to be used in reporting
├── requirements.txt   <- The requirements file for reproducing the analysis environment, e.g.
                           generated with `pip freeze > requirements.txt`
├── setup.cfg          <- Configuration file for flake8
└── balsas             <- Source code for use in this project.
    ├── __init__.py      <- Makes balsas a Python module
    ├── config.py         <- Store useful variables and configuration
    ├── dataset.py        <- Scripts to download or generate data
    ├── features.py       <- Code to create features for modeling
    ├── modeling           <- Code to run model inference with trained models
    │   ├── __init__.py
    │   ├── predict.py
    │   └── train.py        <- Code to train models
    └── plots.py          <- Code to create visualizations

```

Anexo 2: Archivo YALM

Contenido del Archivo YALM necesario para el entrenamiento de Yolov8n-obb

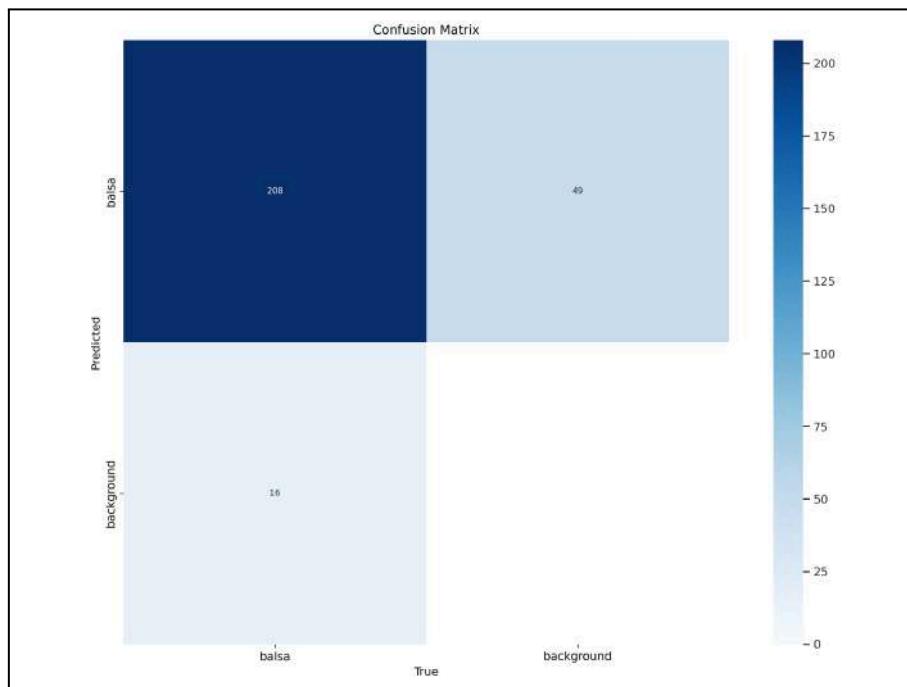
```
# data.yaml

# Rutas de entrenamiento/validación/prueba (ajusta según tu estructura de directorios)
path: /content/drive/My Drive/TFM_BALSAS/dataset # ruta base del dataset
train: images/train # ruta relativa a las imágenes de entrenamiento
val: images/val # ruta relativa a las imágenes de validación
test: images/test # ruta relativa a las imágenes de prueba (opcional)

# Nombres de las clases
names:
  0: balsa

# Información adicional
nc: 1 # número de clases
task: detect # tipo de tarea: detect, segment, classify, etc.
```

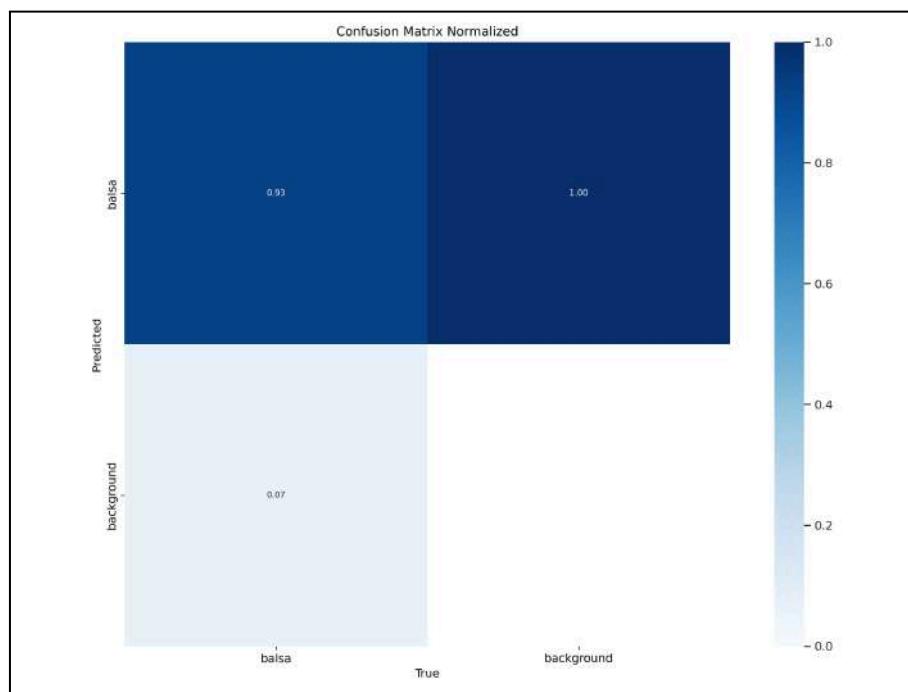
Anexo 3: Visualizaciones del Entrenamiento



Matriz de confusión: Esta matriz compara las predicciones del modelo con las etiquetas reales para dos clases: balsa y background. Los valores numéricos en cada celda representan el número de instancias clasificadas como verdaderas positivas, falsos positivos, falsos negativos y verdaderos negativos.

- **Verdadero positivo (balsa):** 208 instancias fueron correctamente identificadas como balsa.
- **Falsos positivos:** 49 instancias fueron clasificadas incorrectamente como balsa, cuando en realidad eran background.
- **Falsos negativos:** 16 instancias de balsa fueron clasificadas incorrectamente como background.
- **Verdaderos negativos:** Todas las instancias de background fueron correctamente identificadas como tales.

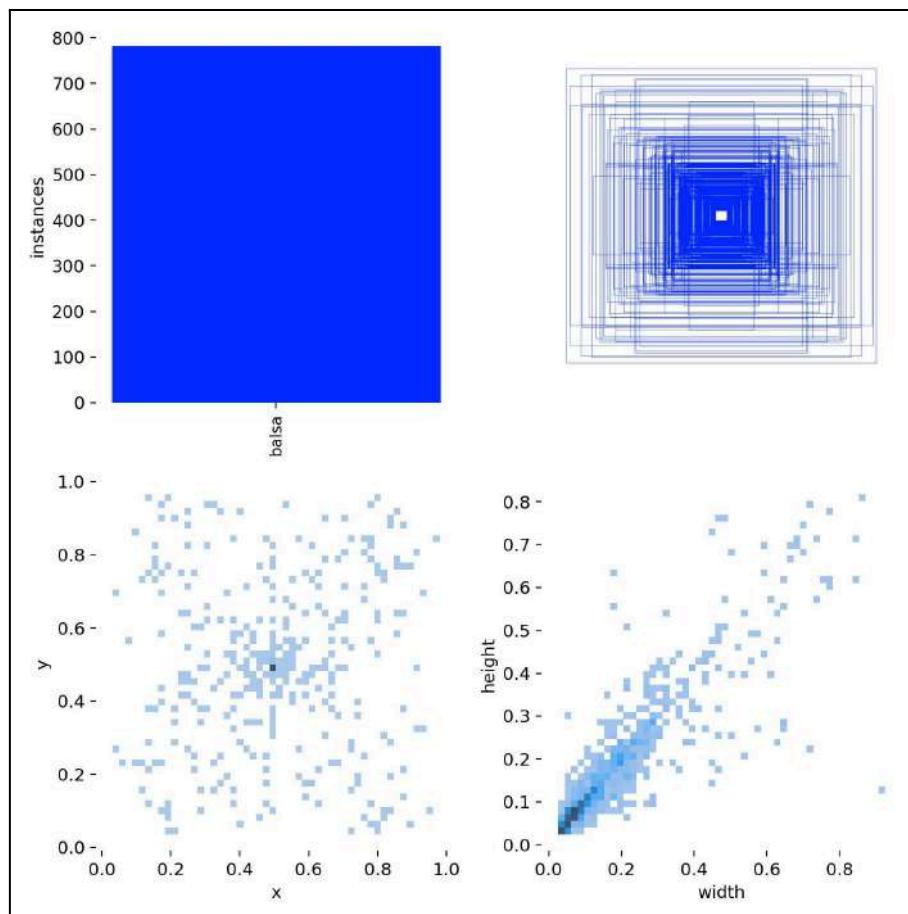
La **Matriz de Confusión** es una herramienta fundamental para evaluar el rendimiento del modelo de detección de objetos. En esta figura, se observa cómo el modelo clasifica las instancias en las clases balsa y background. Los valores en la diagonal principal indican las predicciones correctas, mientras que los valores fuera de la diagonal muestran errores de clasificación. Este análisis permite identificar patrones de fallos específicos del modelo.



Matriz de confusión normalizada: Los valores se expresan como proporciones en lugar de conteos absolutos. Esto facilita comparar el desempeño del modelo independientemente del tamaño del conjunto de datos.

- **Verdadero positivo (balsa):** El 93% de las instancias de balsa fueron clasificadas correctamente.
- **Falsos positivos:** El 7% de las instancias de background fueron clasificadas incorrectamente como balsa.
- **Falsos negativos:** El 7% de las instancias de balsa fueron clasificadas incorrectamente como background.
- **Verdaderos negativos:** El 100% de las instancias de background fueron correctamente identificadas.

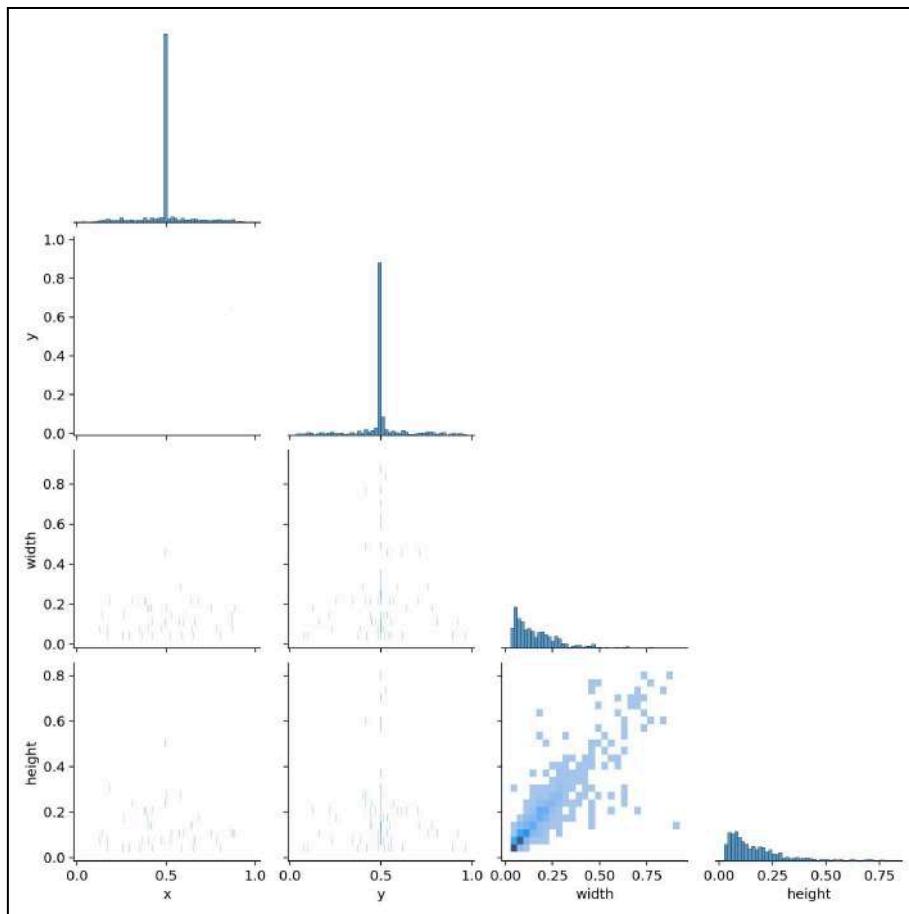
La **Matriz de Confusión Normalizada** proporciona una visión más clara del desempeño relativo del modelo al expresar los resultados como proporciones. En esta figura, se observa que el modelo tiene una alta precisión en la clasificación de balsa (93%), pero también presenta un margen de error significativo en términos de falsos positivos y falsos negativos. La normalización ayuda a interpretar estos resultados de manera más objetiva.



Distribución de **Oriented Bounding Boxes (OBBs)** detectadas por el modelo:

1. **Histograma de instancias:** Muestra la cantidad total de instancias detectadas para la clase balsa.
2. **Gráficos de dispersión:** Representan la distribución de las coordenadas (x, y) y las dimensiones (width, height) de las OBBs.
3. **Diagrama de cajas superpuestas:** Visualiza la variabilidad en las dimensiones de las OBBs.

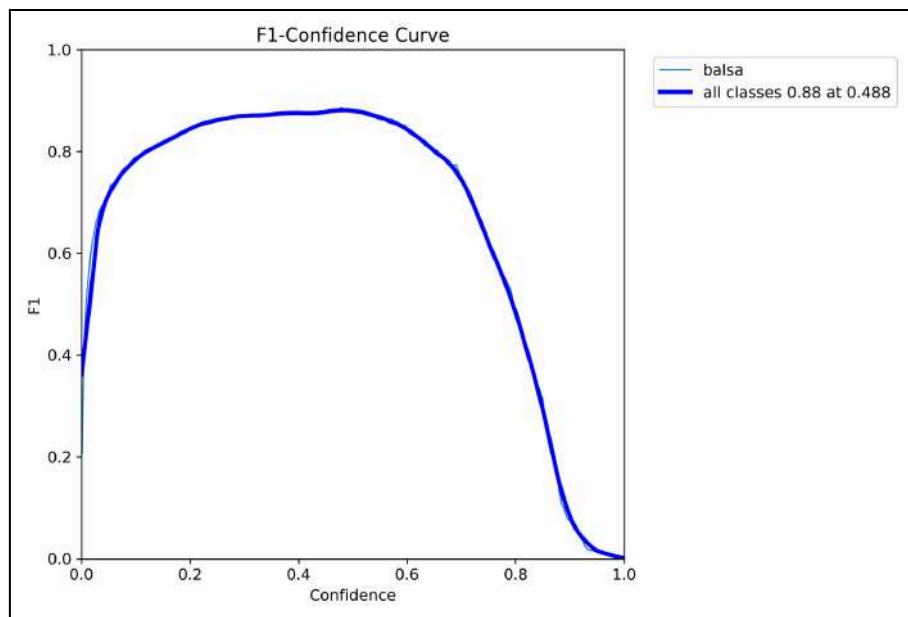
Estas visualizaciones proporcionan pistas sobre la distribución espacial y dimensional de los OBBs generados por el modelo. El histograma indica el número total de instancias detectadas, mientras que los gráficos de dispersión muestran cómo se distribuyen las posiciones y tamaños de los OBBs. El diagrama de cajas superpuestas ofrece una vista rápida de la variabilidad en las dimensiones de las cajas, lo que es útil para entender la consistencia del modelo en la detección de objetos.



Gráficos de Densidad y Histogramas: gráficos que muestran la distribución de características clave de los OBBs:

1. **Histogramas marginales:** Mostrando la distribución de las coordenadas (x, y), anchura (width) y altura (height).
2. **Gráficos de densidad:** Representando la relación entre pares de variables, como (x, y), (y, width) y (width, height).

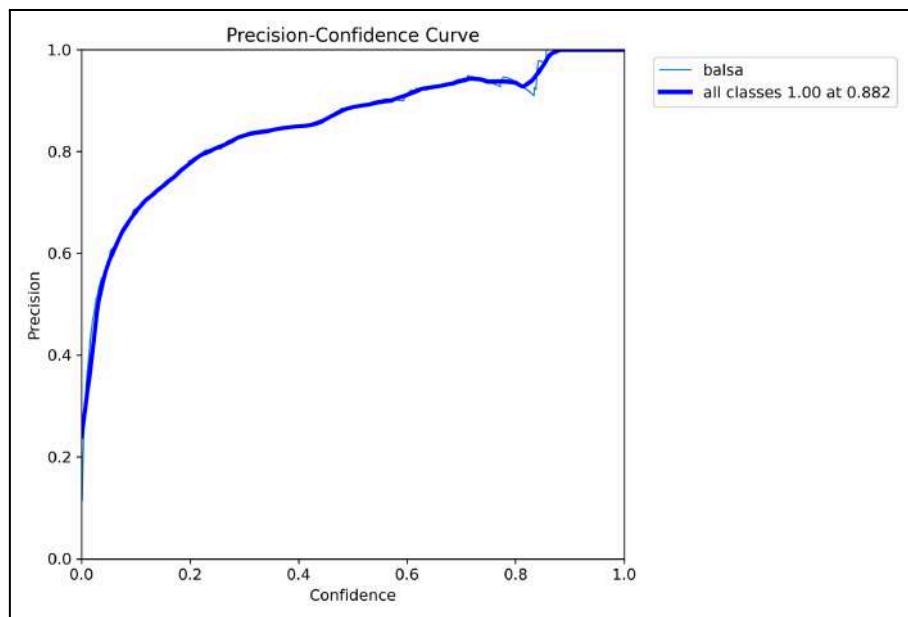
Estos gráficos combinados ofrecen una visión integral de cómo se distribuyen las características de los OBBs generadas por el modelo. Los histogramas marginales permiten identificar patrones en las posiciones y dimensiones de las cajas, mientras que los gráficos de densidad revelan correlaciones entre diferentes características. Estas visualizaciones son importantes para comprender el comportamiento del modelo y optimizar su rendimiento.



F1-Confidence Curve (Curva F1-Confianza), que representa la relación entre la métrica F1-score y el umbral de confianza del modelo. El F1-score combina precisión y recall en una sola métrica equilibrada.

- **Pico máximo:** El F1-score alcanza su valor máximo de **0,88** cuando el umbral de confianza está configurado en **0,488**.
- **Tendencia general:** A medida que aumenta el umbral de confianza, el F1-score disminuye debido a la reducción en el número de predicciones realizadas por el modelo.

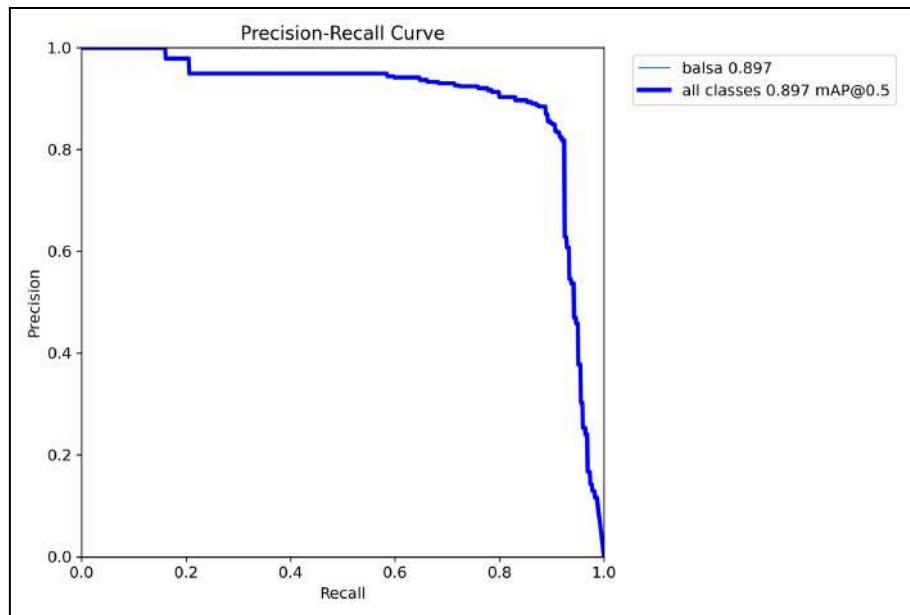
La Curva F1-Confianza es útil para determinar el umbral óptimo de confianza que maximiza el desempeño del modelo. En esta figura, se observa que el modelo alcanza su mejor rendimiento ($F1\text{-score} = 0,88$) cuando el umbral de confianza se establece en 0,488. Esta información es crucial para ajustar el comportamiento del modelo en aplicaciones prácticas, asegurando un equilibrio adecuado entre precisión y recall.



Precision-Confidence Curve (Curva Precisión-Confianza): representa cómo varía la precisión del modelo en función del umbral de confianza. La precisión es la proporción de predicciones correctas entre todas las predicciones realizadas.

- **Pico máximo:** La precisión alcanza su valor máximo de **1** cuando el umbral de confianza está configurado en **0,882**.
- **Tendencia general:** A medida que aumenta el umbral de confianza, la precisión tiende a mejorar, ya que el modelo se vuelve más selectivo al realizar predicciones.

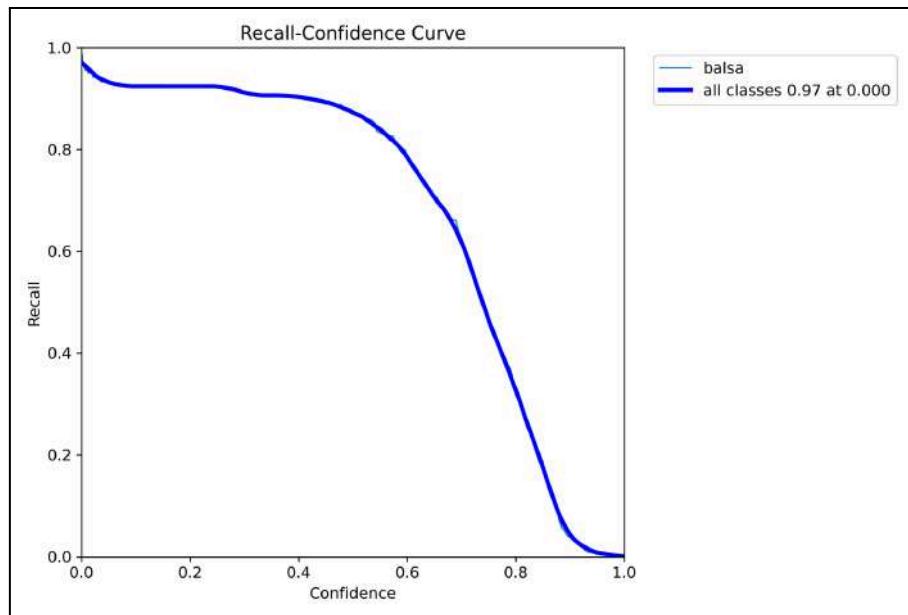
La **Curva Precisión-Confianza** es una herramienta clave para evaluar cómo el modelo ajusta su rendimiento en función del umbral de confianza. En esta figura, se observa que el modelo logra una precisión perfecta (1) cuando el umbral de confianza se establece en 0,882. Esto indica que el modelo puede ser muy preciso al filtrar predicciones con alta confianza, aunque esto podría reducir el número total de predicciones realizadas.



Precision-Recall Curve (Curva Precisión-Recall): representa la relación entre la precisión y el recall (sensibilidad) del modelo. El recall mide la proporción de instancias reales correctamente identificadas.

- **Punto inicial:** La curva comienza con una precisión cercana a 1 y un recall bajo, lo que indica que el modelo detecta pocas instancias pero con alta precisión.
- **Evolución:** A medida que el recall aumenta, la precisión disminuye gradualmente, mostrando un equilibrio entre capturar más instancias reales y mantener una alta precisión.
- **Valor final:** El modelo alcanza un mAP (Average Precision) de **0,897** para todas las clases, indicando un buen desempeño global.

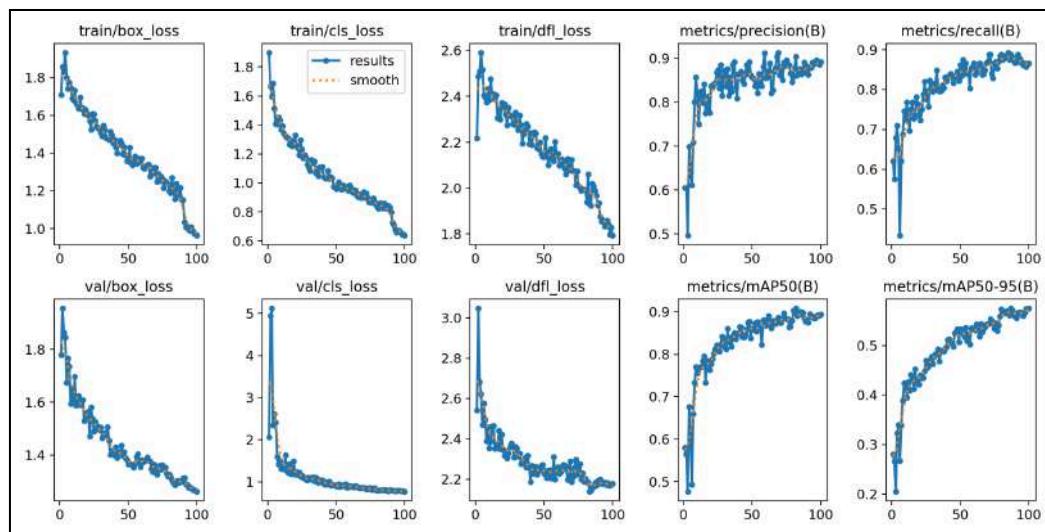
La Curva Precisión-Recall es fundamental para comprender el equilibrio entre la precisión y el recall del modelo. En esta figura, se observa que el modelo logra un alto nivel de precisión inicialmente, pero a medida que aumenta el recall, la precisión disminuye ligeramente. El mAP de 0,897 indica que el modelo tiene un desempeño sólido en términos de precisión promedio a lo largo de diferentes umbrales de confianza.



Recall-Confidence Curve (Curva Recall-Confianza): representa cómo varía el recall del modelo en función del umbral de confianza.

- **Punto inicial:** El recall comienza cerca de 1 con un umbral de confianza bajo, lo que indica que el modelo detecta casi todas las instancias reales.
- **Tendencia general:** A medida que aumenta el umbral de confianza, el recall disminuye, ya que el modelo se vuelve más selectivo y deja de detectar algunas instancias.
- **Valor mínimo:** El recall cae drásticamente cuando el umbral de confianza (en torno al 55%) supera cierto punto crítico.

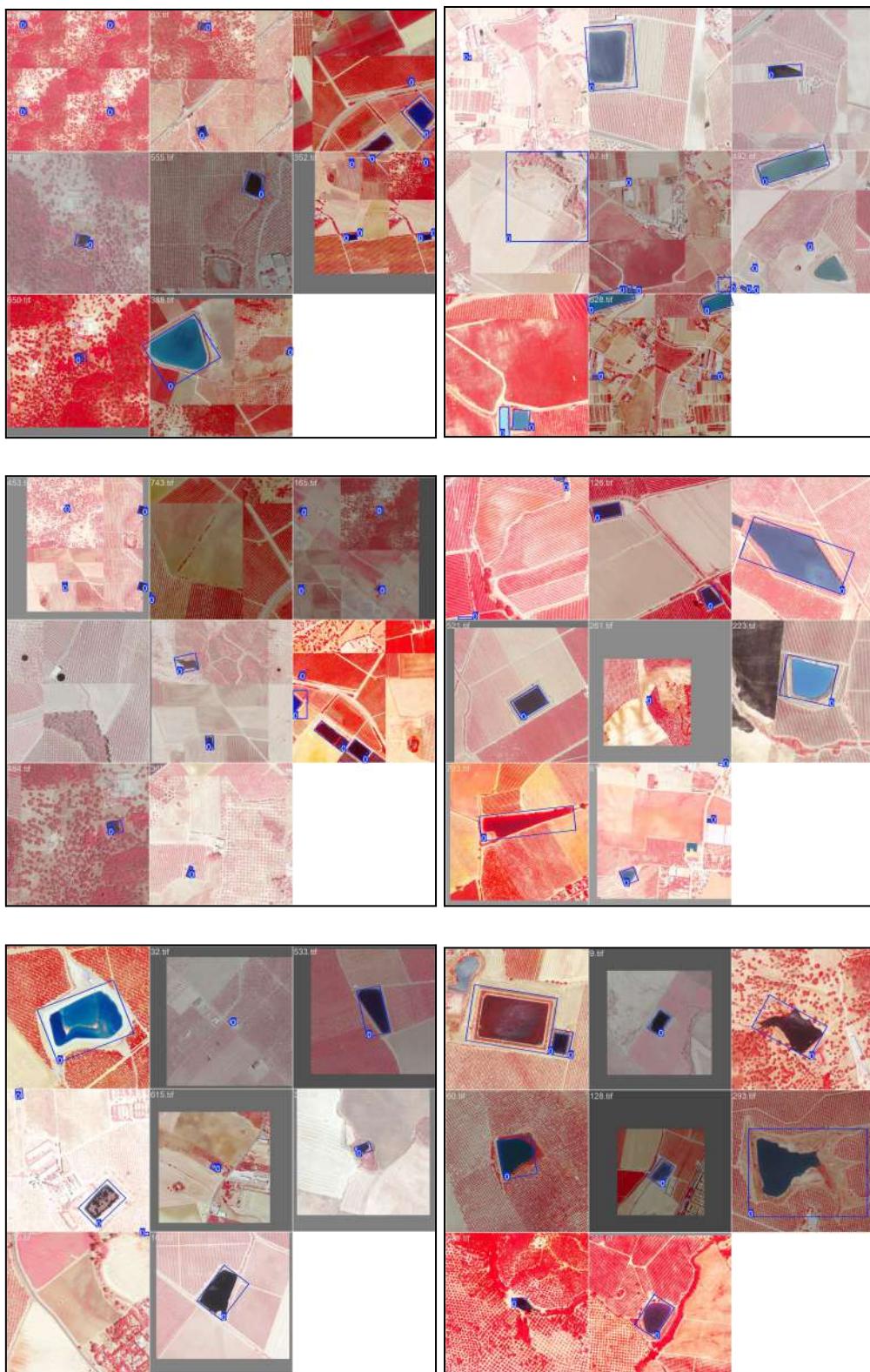
La **Curva Recall-Confianza** permite evaluar cómo el modelo ajusta su capacidad para detectar instancias reales en función del umbral de confianza. En esta figura, se observa que el modelo mantiene un alto recall inicialmente, pero este disminuye significativamente a medida que se incrementa el umbral de confianza. Este análisis es importante para determinar el punto óptimo donde se equilibran el recall y la confianza.



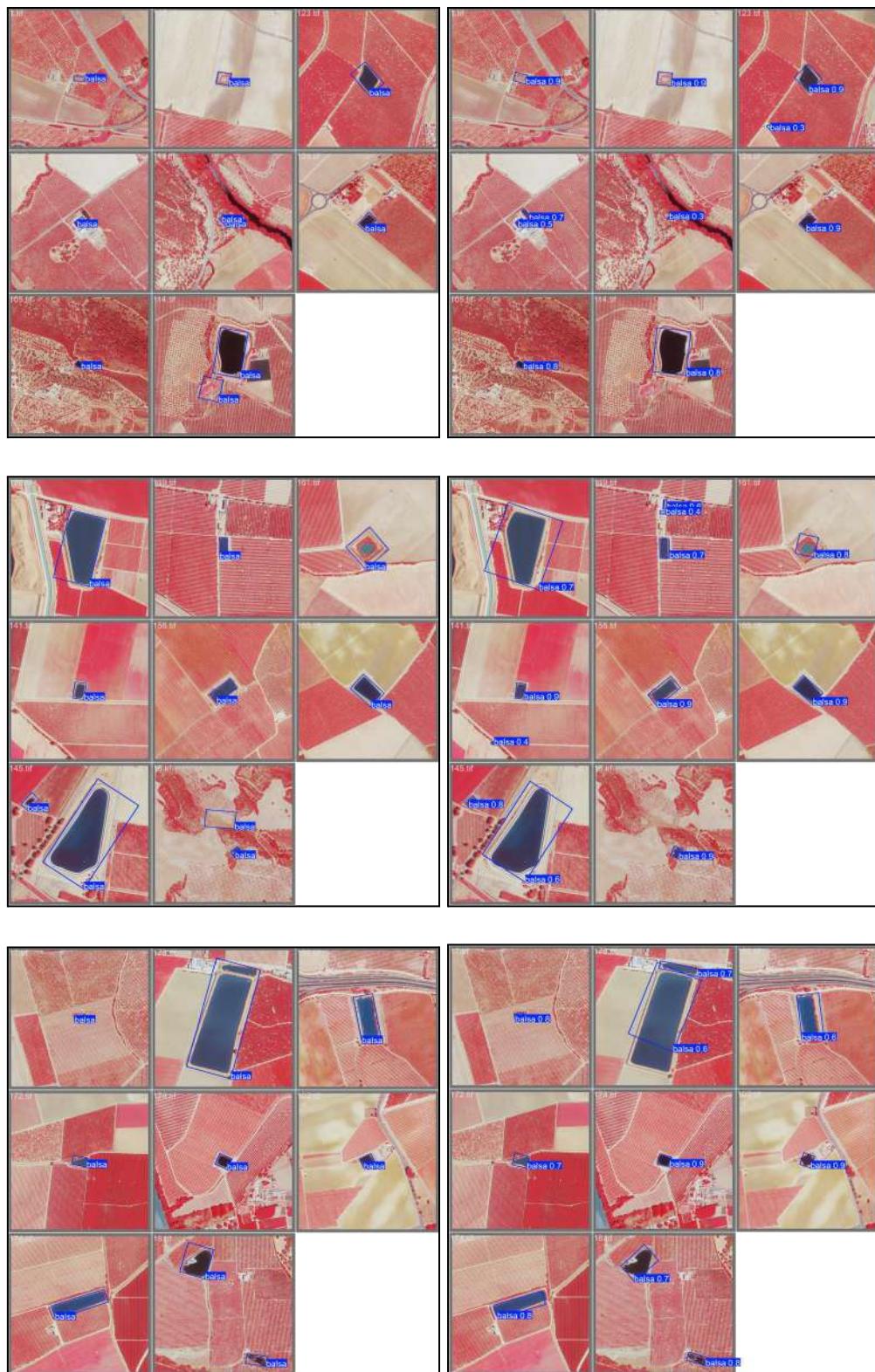
Entrenamiento y Validación (Gráficos de Pérdidas y Métricas): gráficos relacionados con el proceso de entrenamiento y validación del modelo:

- **Pérdidas de entrenamiento y validación:**
 - train/box_loss: Pérdida asociada a la localización de los OBBs durante el entrenamiento.
 - train/cls_loss: Pérdida asociada a la clasificación durante el entrenamiento.
 - train/dfl_loss: Pérdida diferencial focal (Differential Focal Loss) durante el entrenamiento.
 - val/box_loss, val/cls_loss, val/dfl_loss: Correspondientes pérdidas en el conjunto de validación.
- **Métricas de precisión y recall:**
 - metrics/precision(B): Precisión en función de los epochs.
 - metrics/recall(B): Recall en función de los epochs.
- **mAP (Mean Average Precision):**
 - metrics/mAP50(B): mAP calculado a un IoU de 0,5.
 - metrics/mAP50-95(B): mAP calculado a IoUs entre 0,5 y 0,95.

Estos gráficos resumen el progreso del modelo durante el entrenamiento y validación. Las pérdidas (box_loss, cls_loss, dfl_loss) muestran cómo el modelo mejora su capacidad para localizar objetos y clasificarlos correctamente. Las métricas de precisión y recall (precision(B) y recall(B)) muestran el desempeño del modelo en términos de exactitud y capacidad de detección. Finalmente, las métricas de mAP (mAP50(B) y mAP50-95(B)) proporcionan una evaluación integral del rendimiento del modelo en diferentes niveles de IoU.



Muestra de imágenes utilizadas durante el entrenamiento con el modelo YOLOv8n-obb. Las anotaciones visuales (cuadrados azules) guían el proceso de aprendizaje, permitiendo al modelo identificar patrones consistentes en diferentes contextos. La inclusión de escenarios variados y la rotación de los OBBs garantiza que el modelo sea capaz de generalizar y realizar predicciones precisas en situaciones reales.



Pares de mosaicos de imágenes generadas durante la validación del modelo en las que se muestra, en las imágenes de la izq. los OBBs de los datos de partida y a la dcha. las mismas imágenes con los OBBs predichos por el modelo con el valor del grado de confianza

Anexo 4: Resultado de la Inferencia Sobre Imágenes de Test

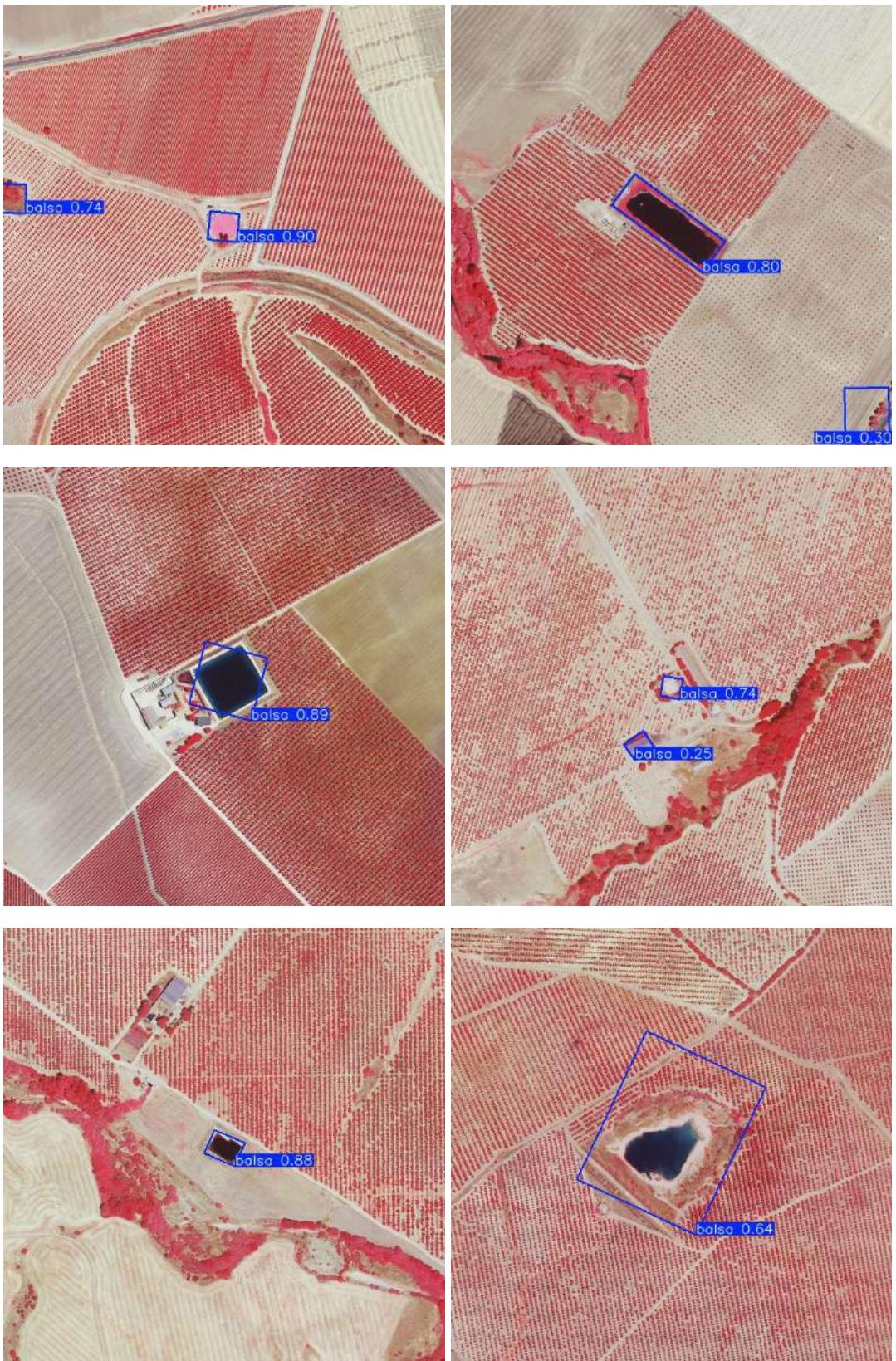
Se presentan, a continuación los resultados de la inferencia completa realizada sobre las 160 imágenes no vistas durante el entrenamiento del modelo, correspondientes a la carpeta de imágenes de test. Estas imágenes han sido utilizadas exclusivamente para evaluar el desempeño real del modelo en escenarios desconocidos, proporcionando una visión general de su capacidad de detección en condiciones diversas.

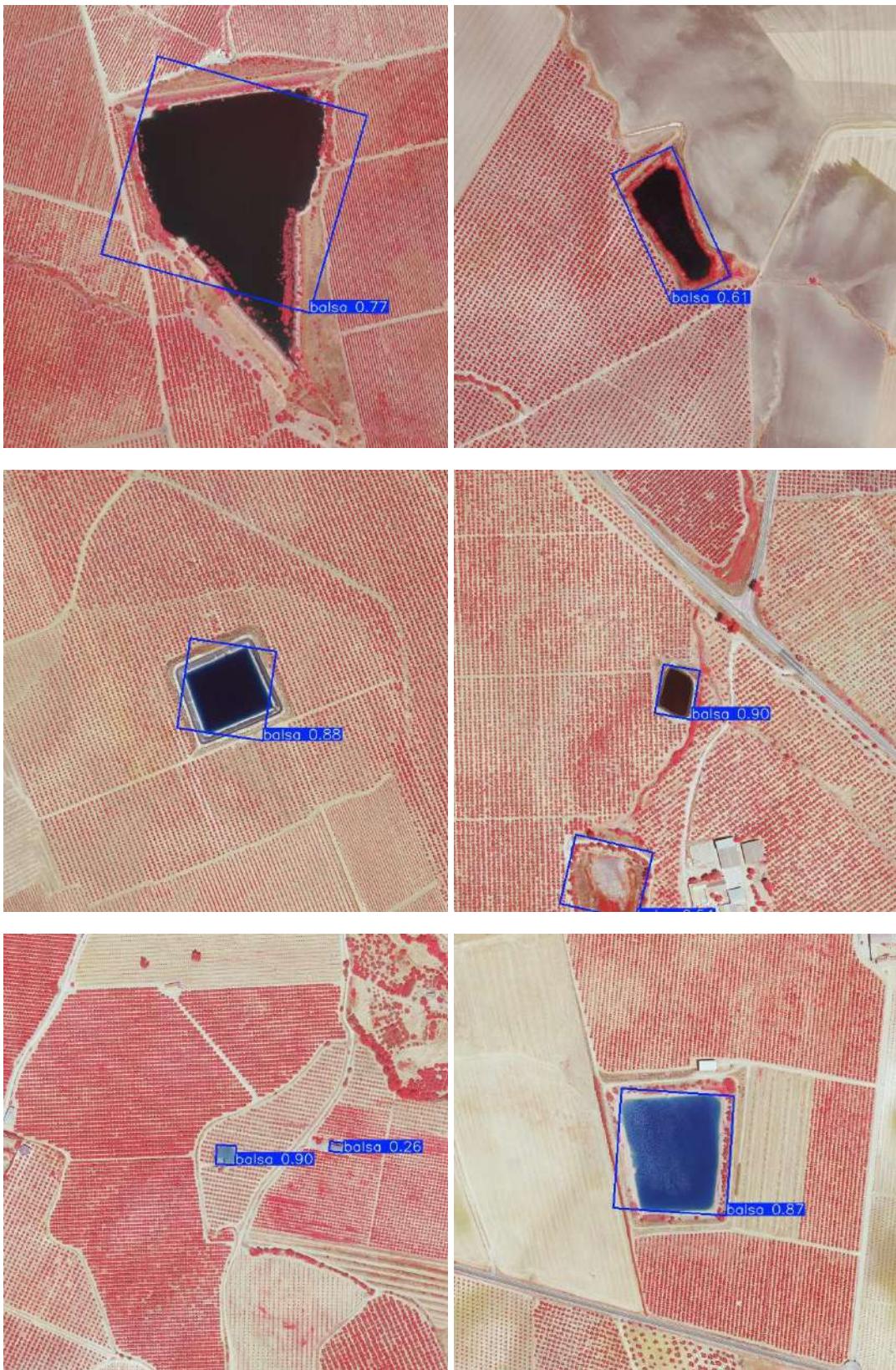
En términos generales, el modelo ha mostrado un buen rendimiento, con un alto porcentaje de detecciones correctas y valores significativos de confianza. No obstante, también se han observado algunas limitaciones en ciertos casos puntuales:

- En situaciones con reflejos en la superficie del agua, el modelo ha tenido dificultades para identificar balsas que son claramente visibles a simple vista.
- Algunas balsas cubiertas parcial o totalmente por vegetación no han sido detectadas, aunque en otros casos similares sí se ha logrado una detección exitosa.
- Respecto a balsas secas, el comportamiento ha sido heterogéneo: mientras que en muchos casos se han identificado correctamente, en algunos otros pocos casos no se han detectado.
- También se han registrado casos de falsas detecciones, principalmente en elementos artificiales como tejados rectangulares de naves industriales, cuya forma y textura pueden asemejarse a la de una balsa o en fragmentos cortados en los bordes de las imágenes pertenecientes a otras entidades.
- Se han observado casos, también de detecciones dobles, identificando la misma balsa con dos OBBs de distintas dimensiones u orientación.
- En algunas circunstancias particulares, se han observado detecciones parciales de balsas, especialmente en aquellas con diferencias de textura interna o bordes poco definidos, lo que sugiere que el modelo puede tener cierta sensibilidad a estas características visuales.
- Como comentario final, cabe destacar que, en ocasiones, las orientaciones de los OBB no son completamente precisas. Sin embargo, teniendo en cuenta que el objetivo principal es la identificación de la presencia y localización de balsas, y dada la gran heterogeneidad morfológica observada en las mismas, esta imprecisión en la orientación no resulta crítica ni afecta al propósito fundamental del modelo.

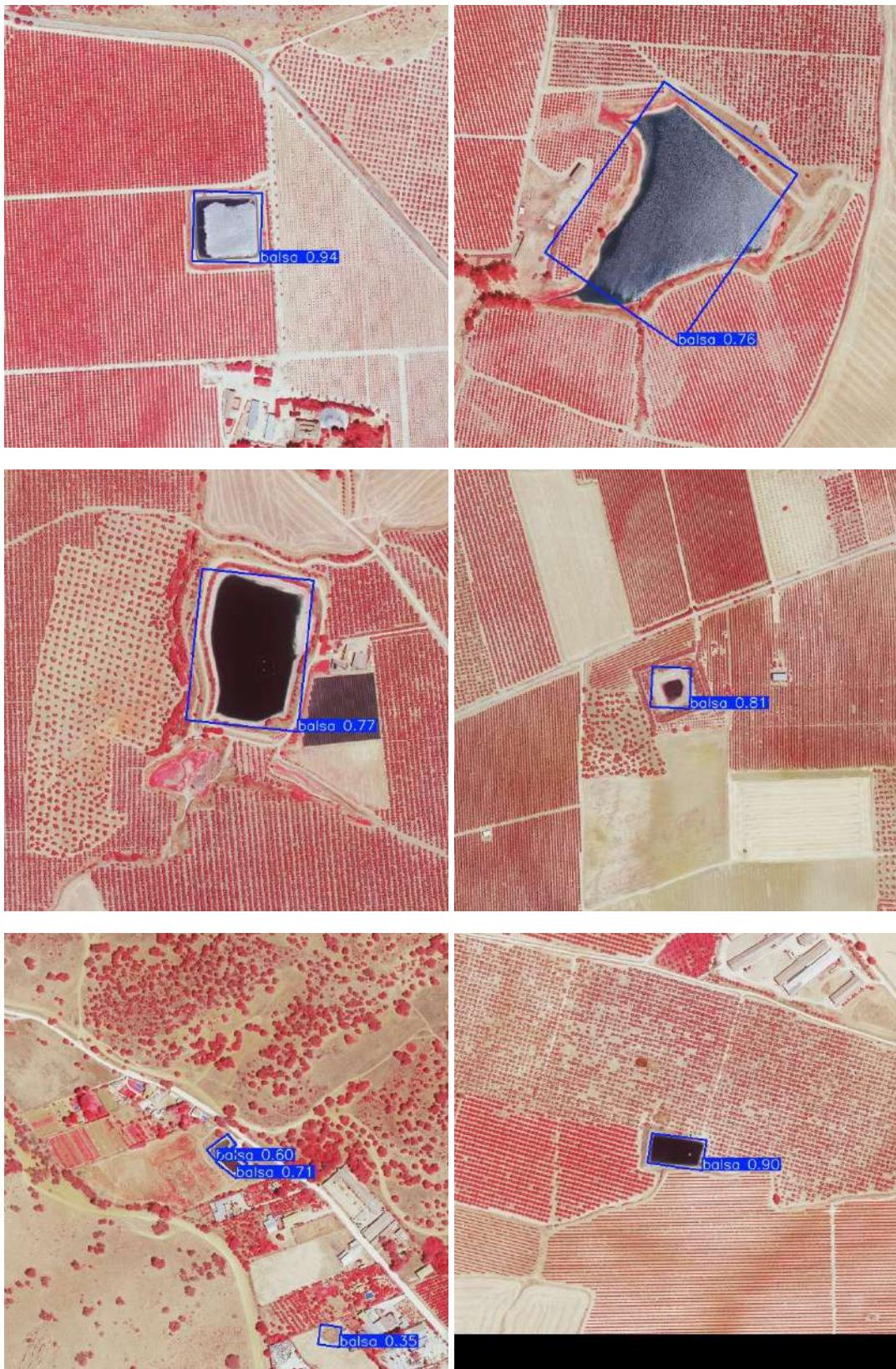
Estas observaciones permiten tener una visión más precisa del comportamiento del modelo en entornos reales y ayudan a identificar posibles áreas de mejora para futuros ajustes o iteraciones del modelo.

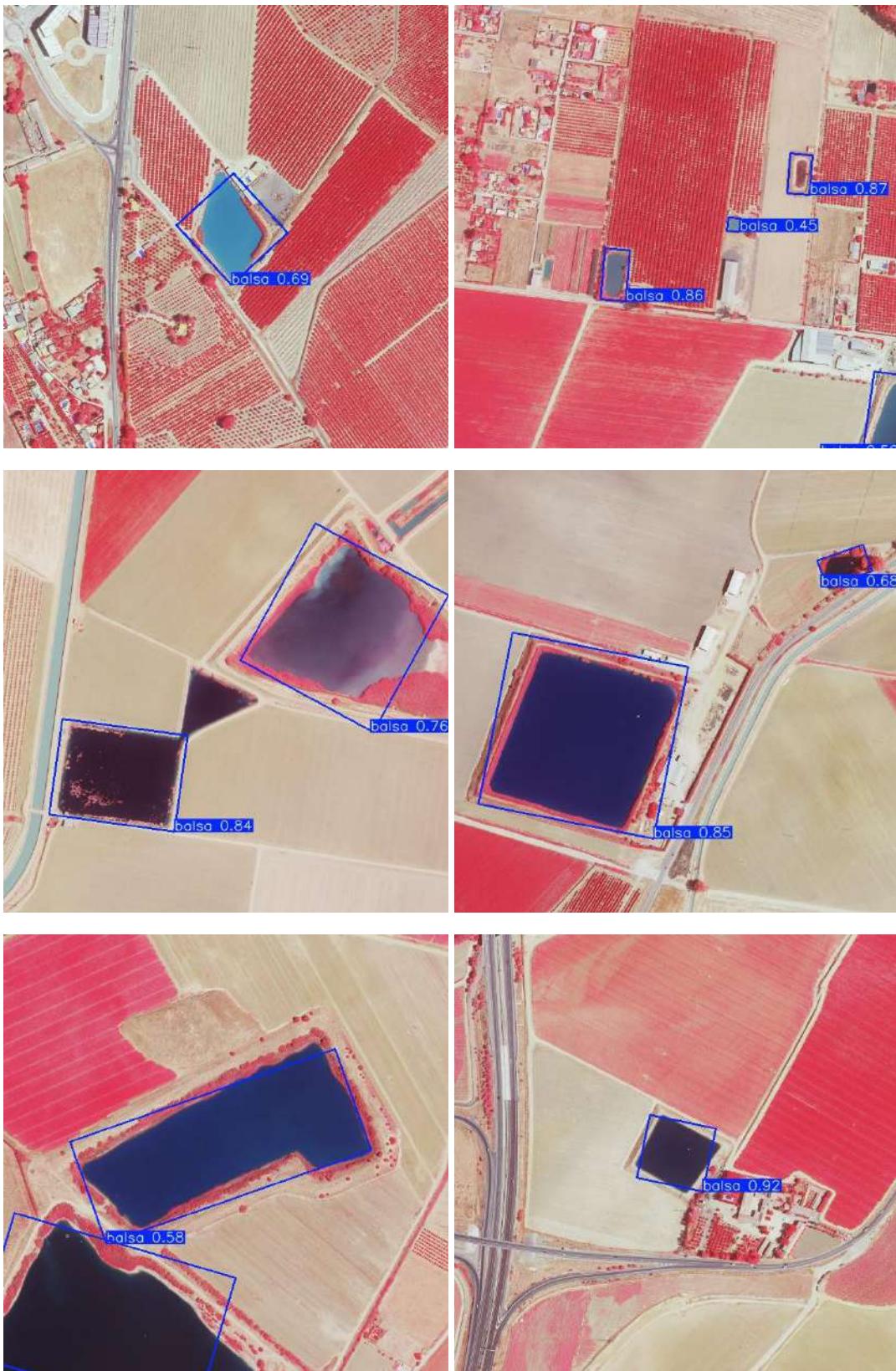




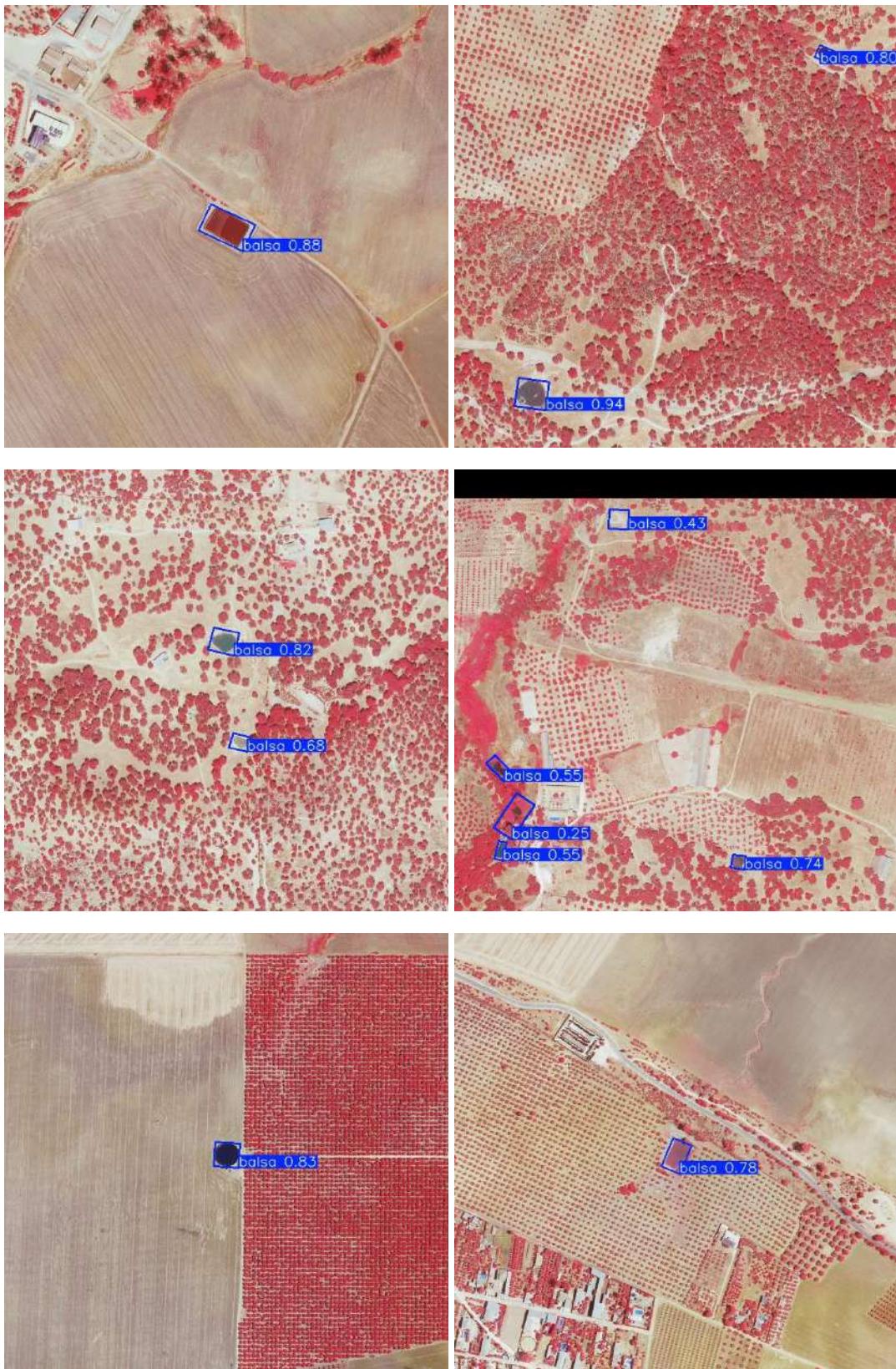


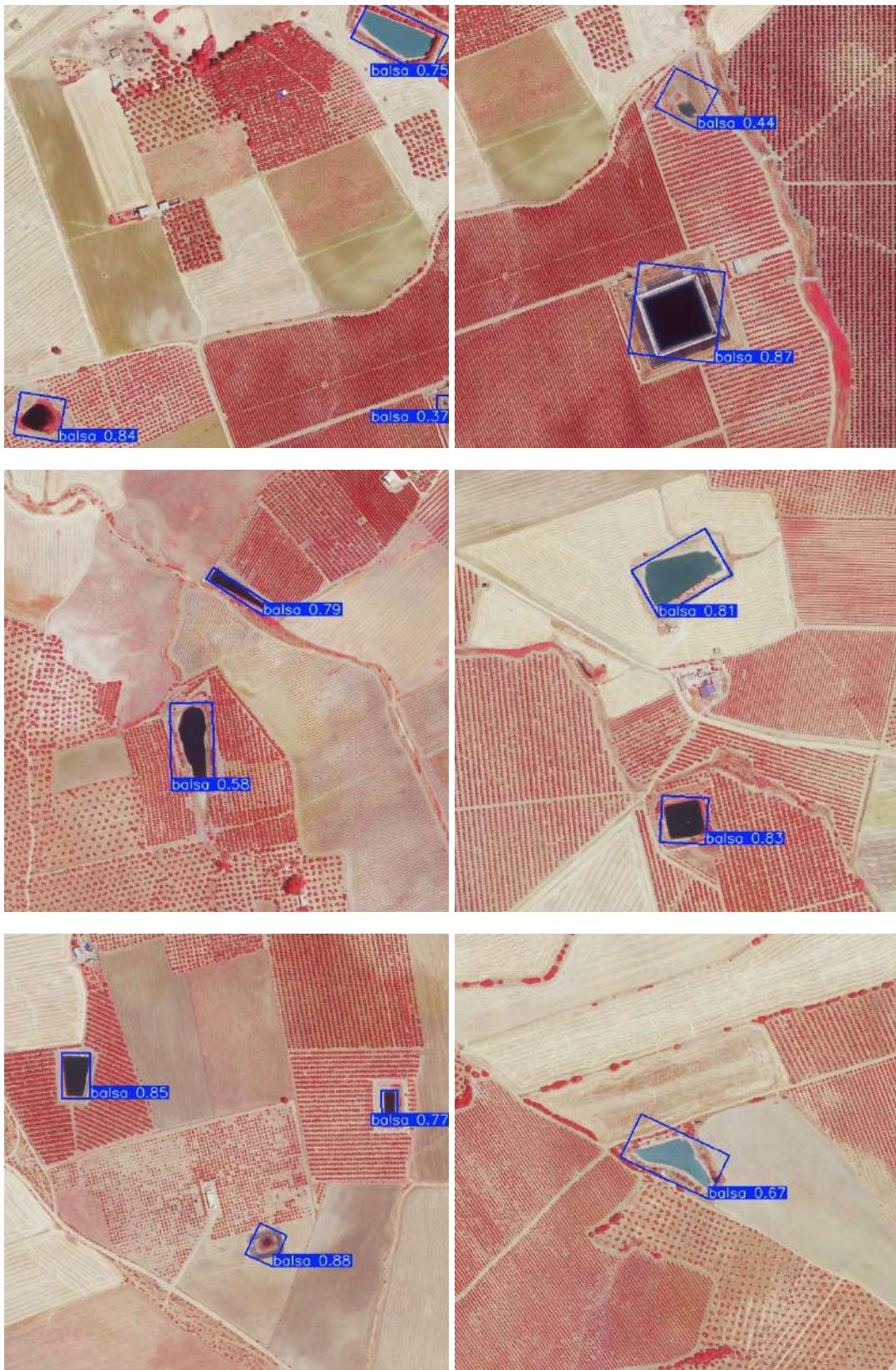


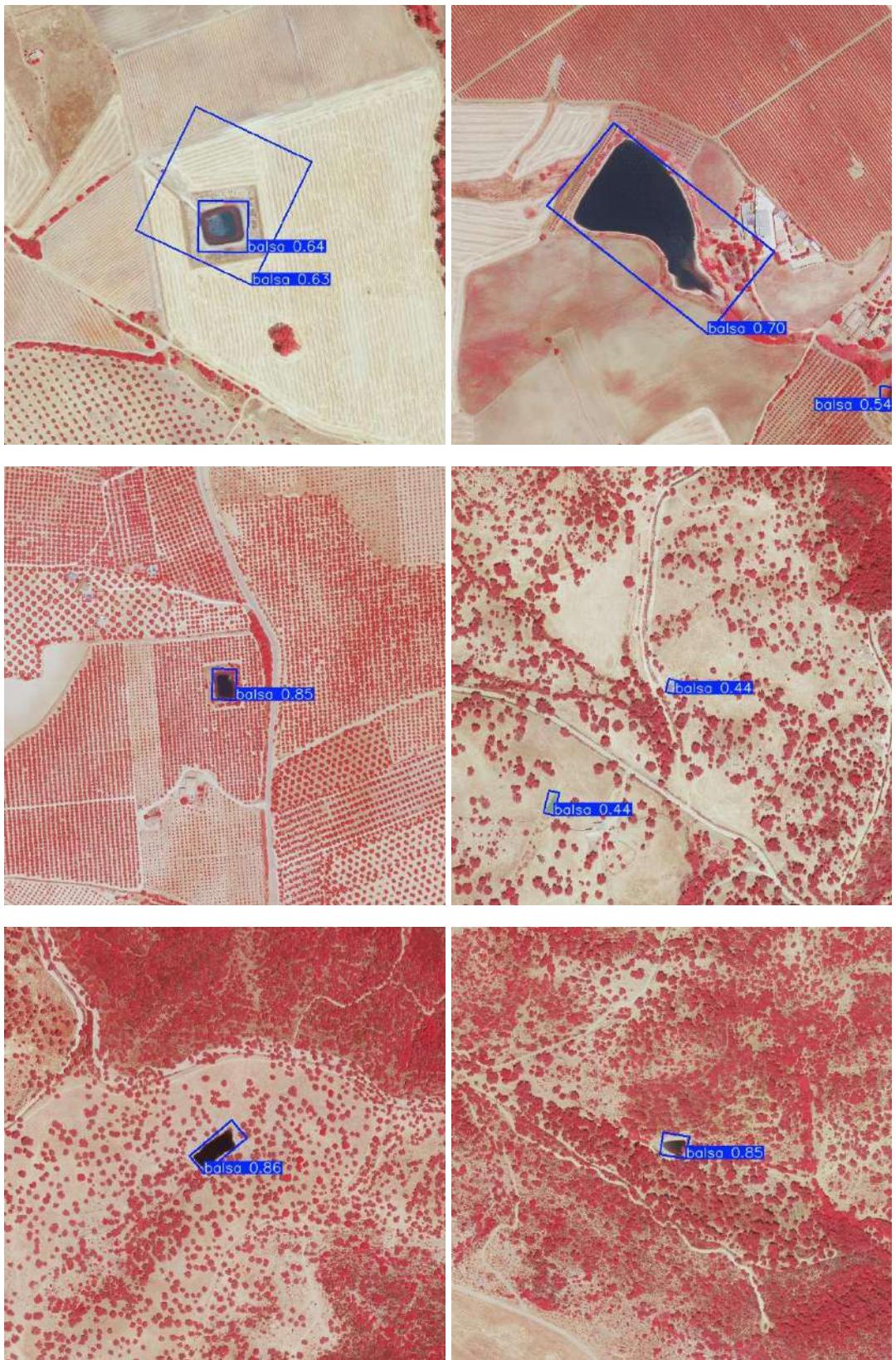


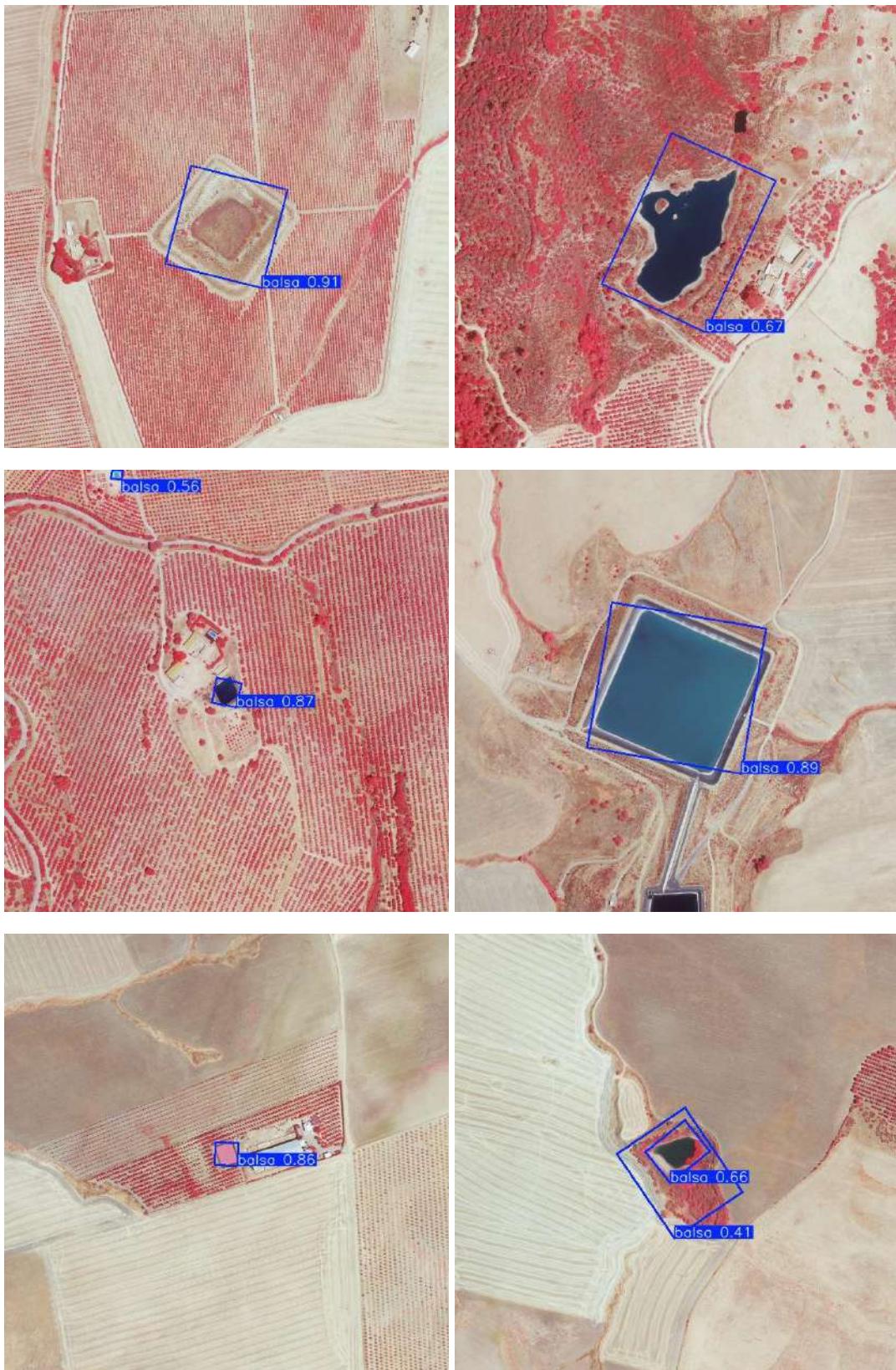




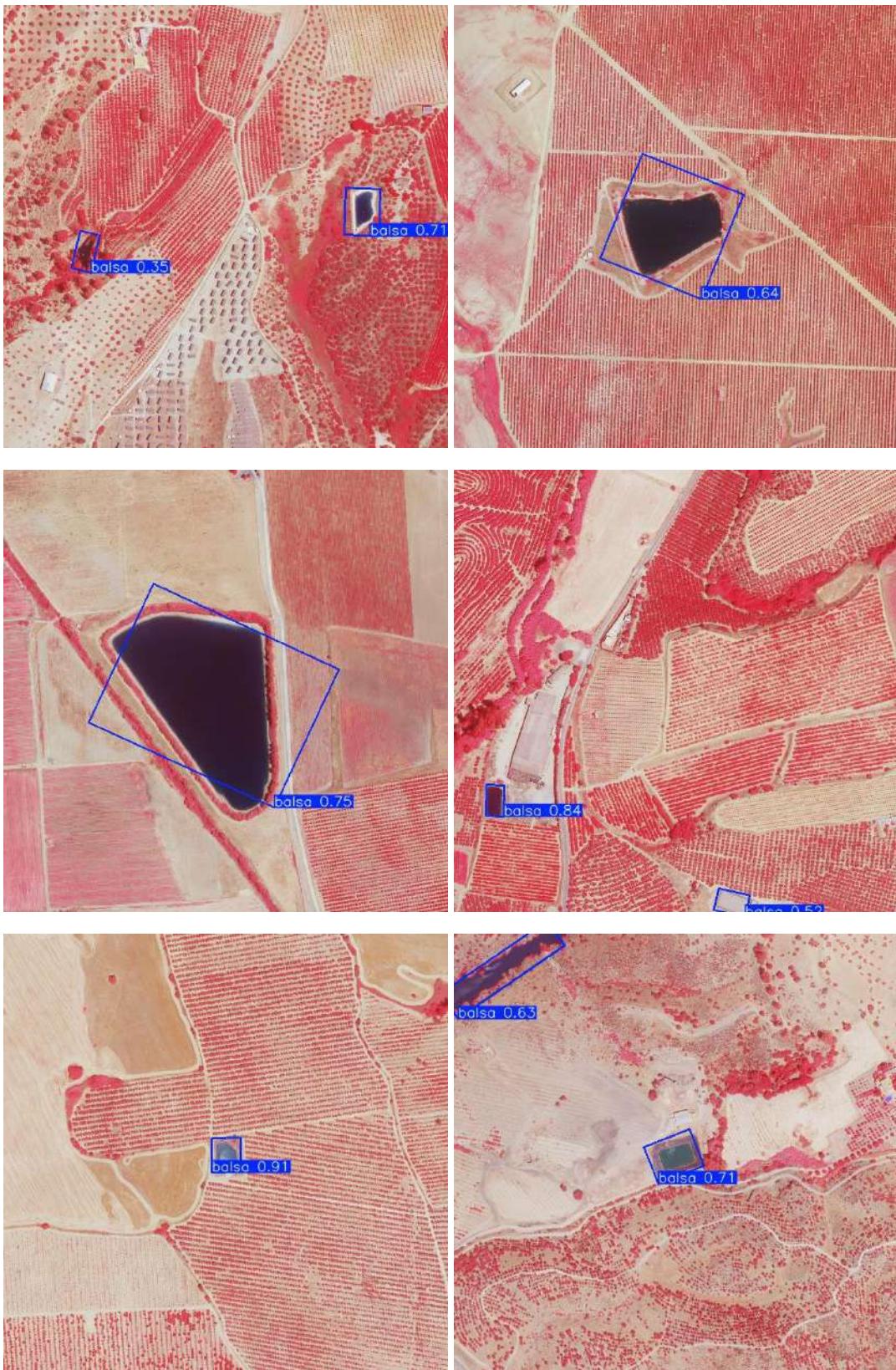




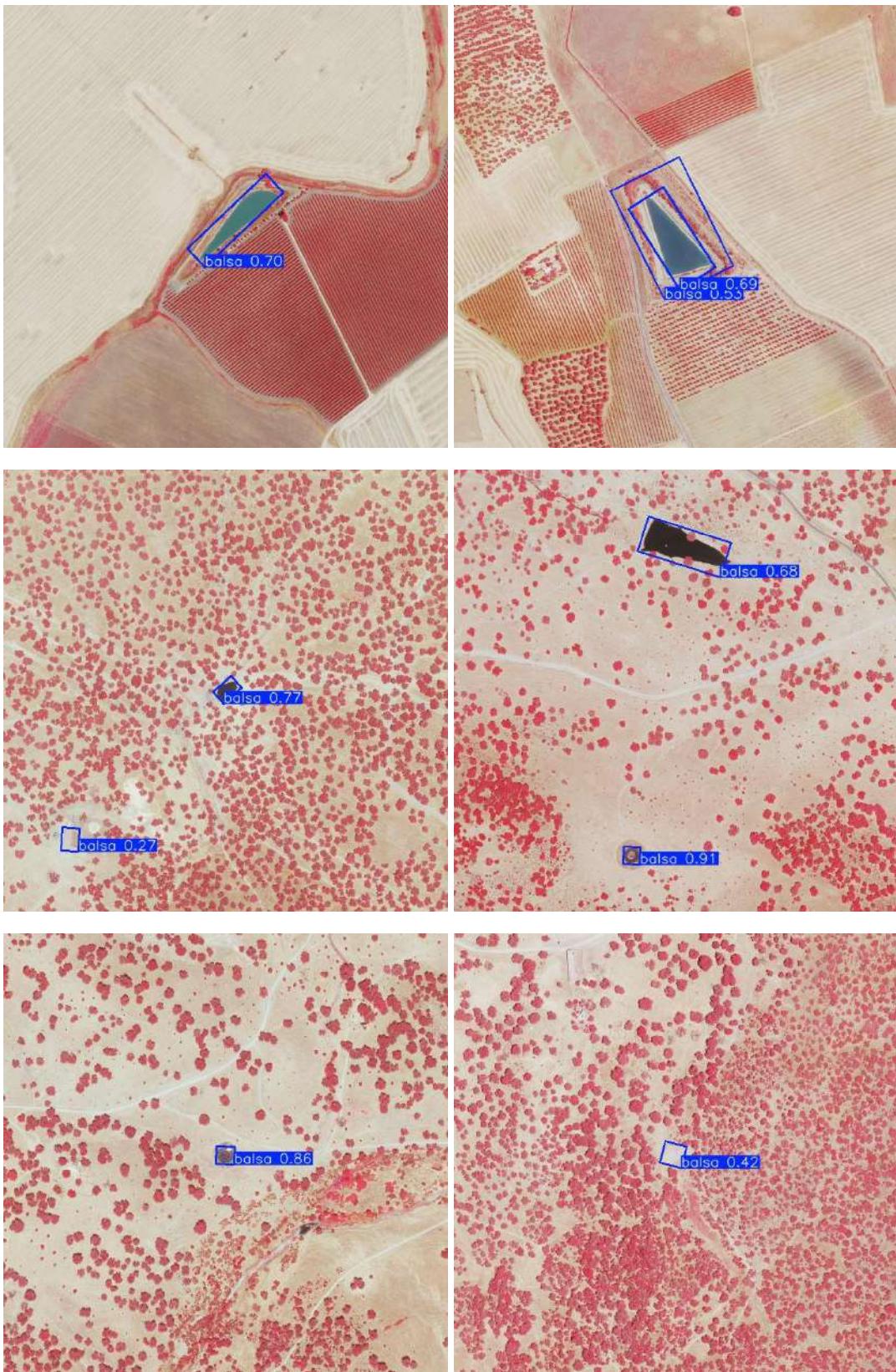


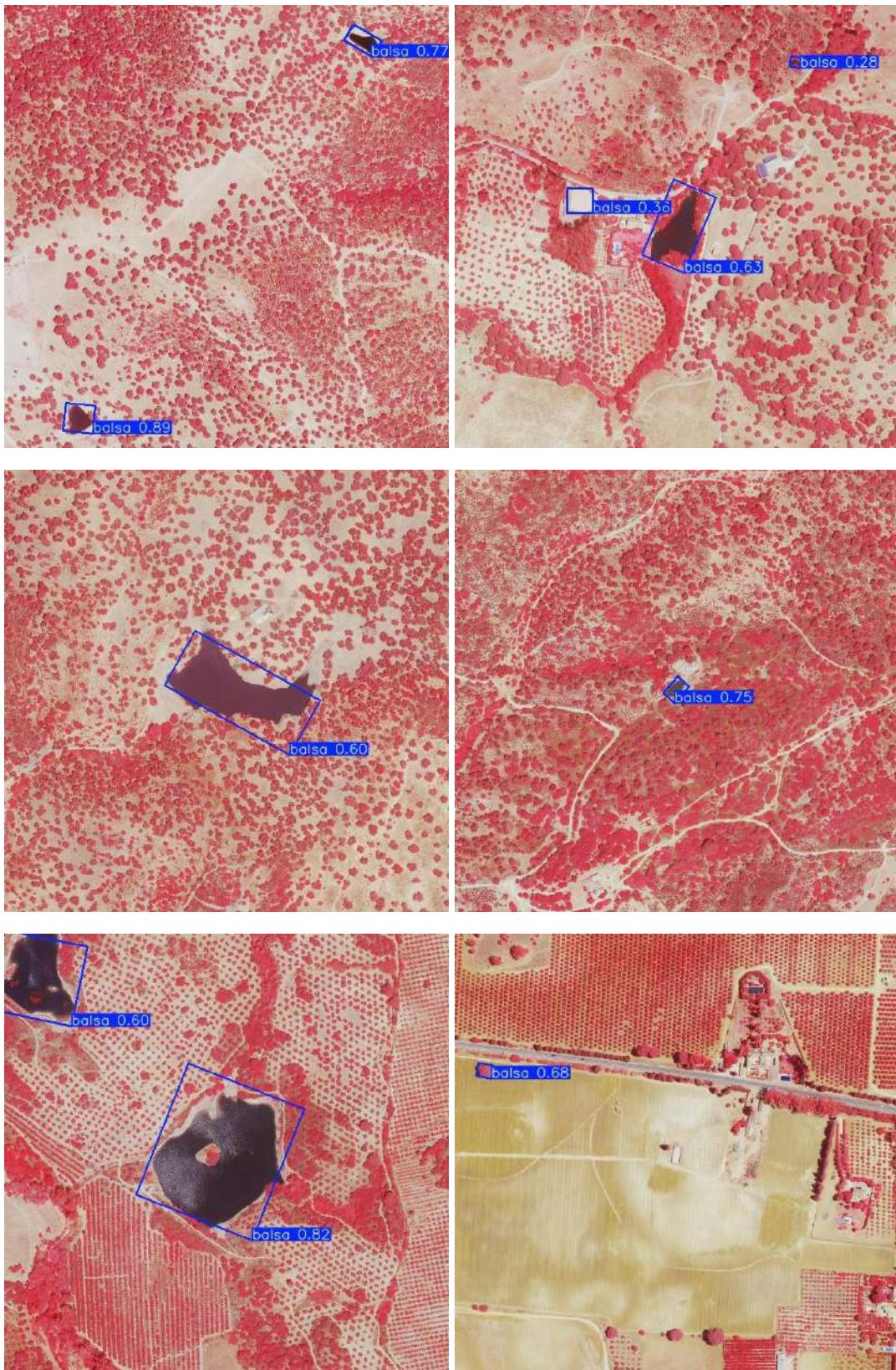


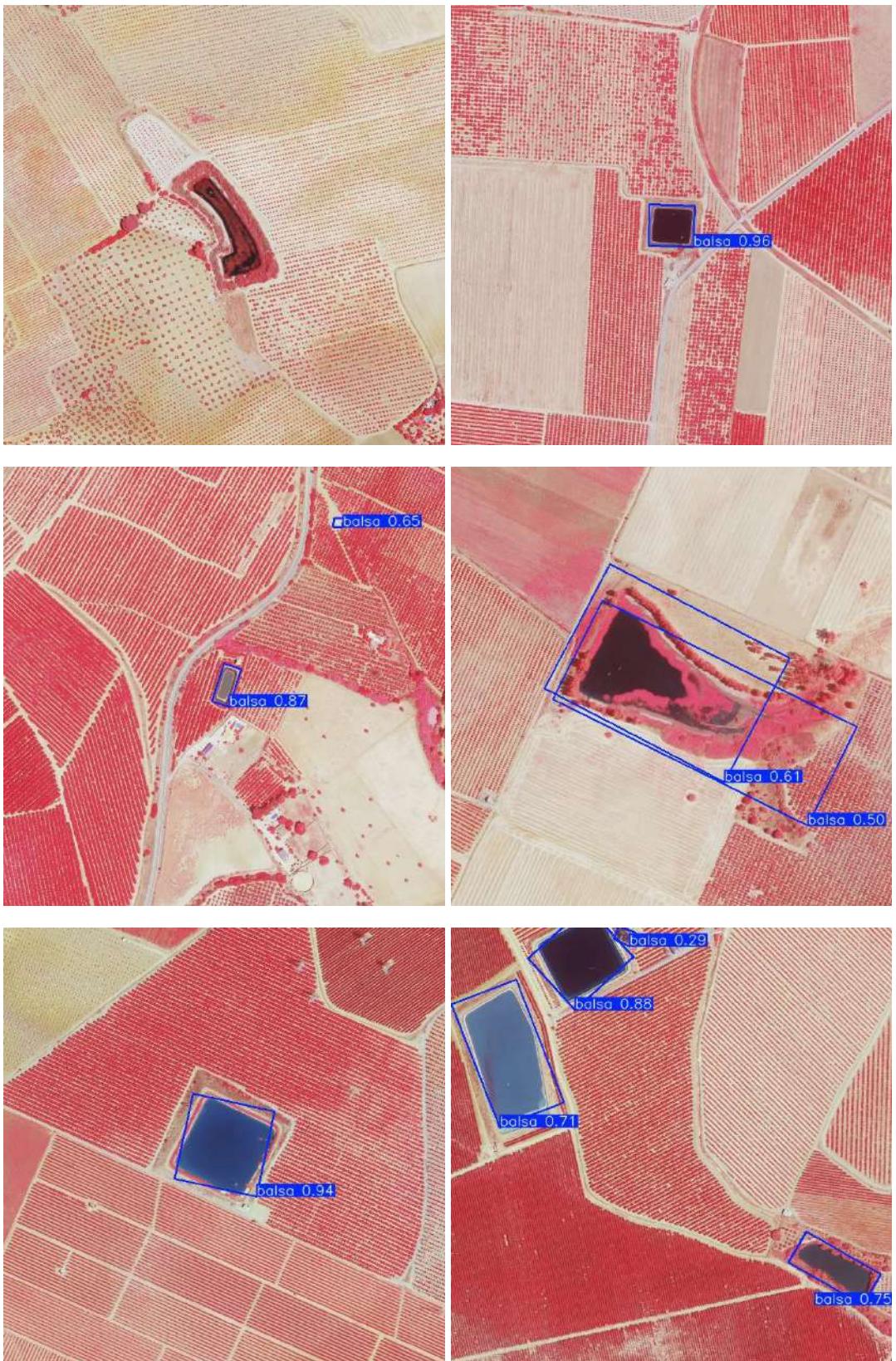


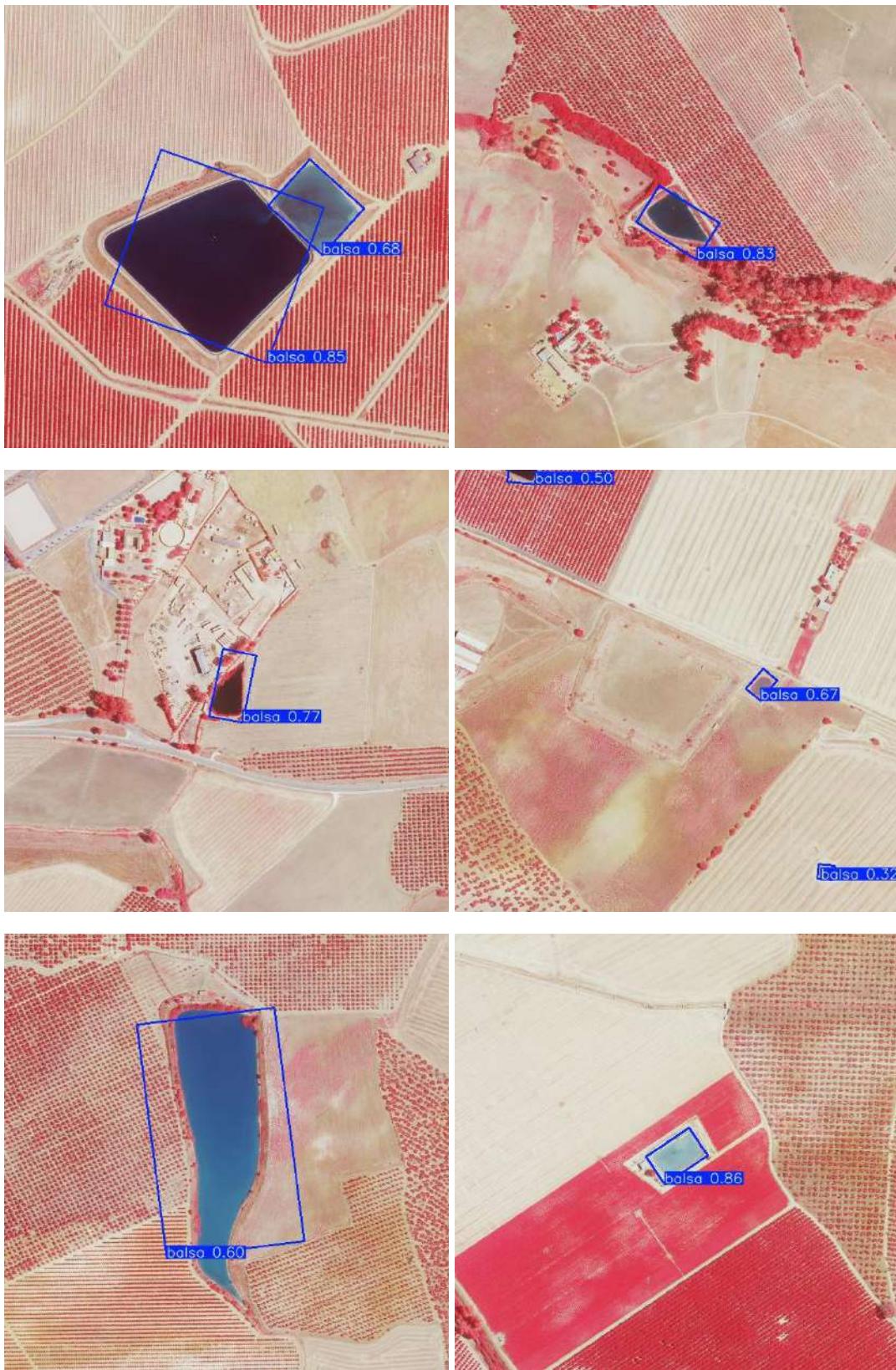


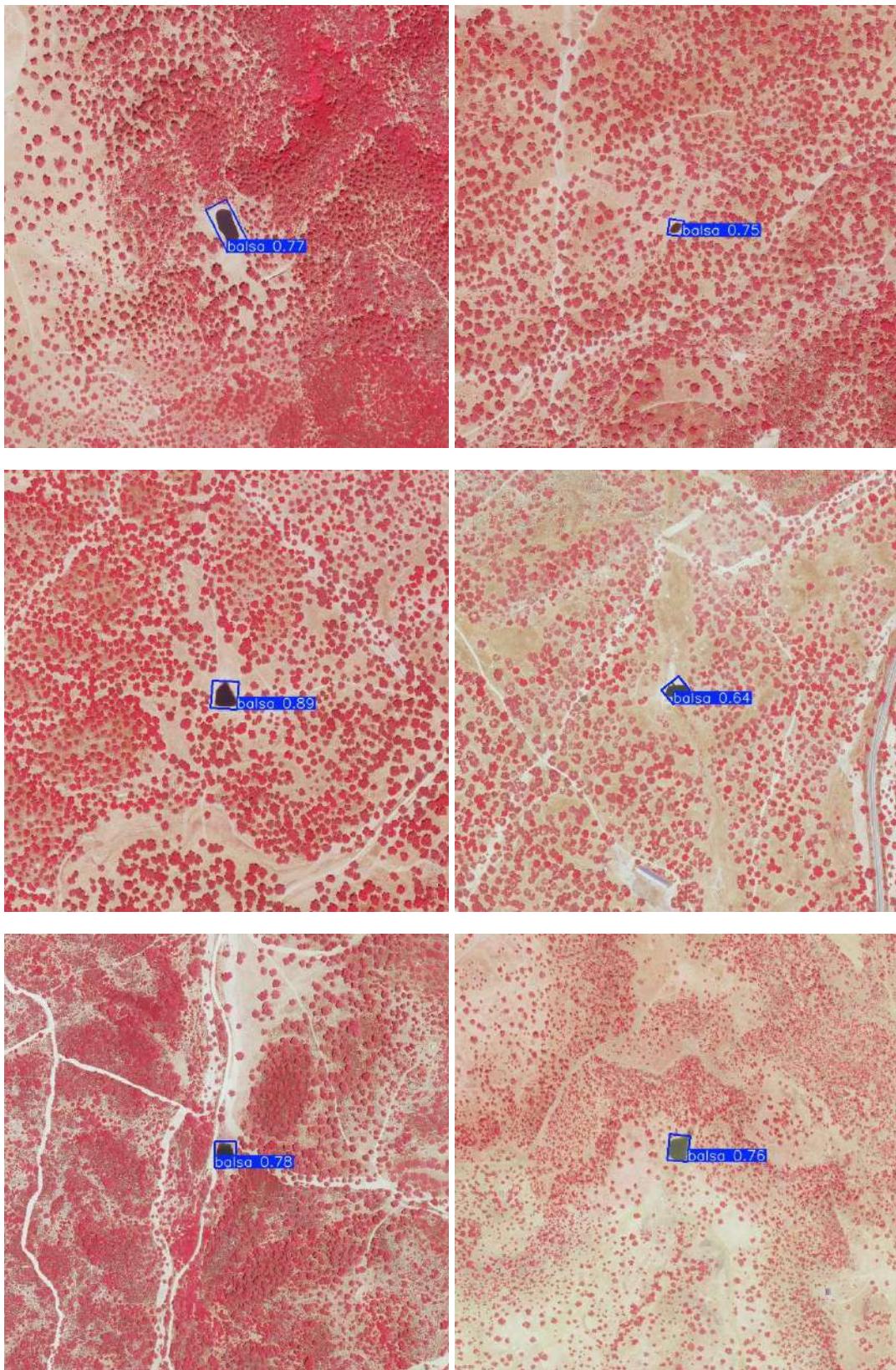


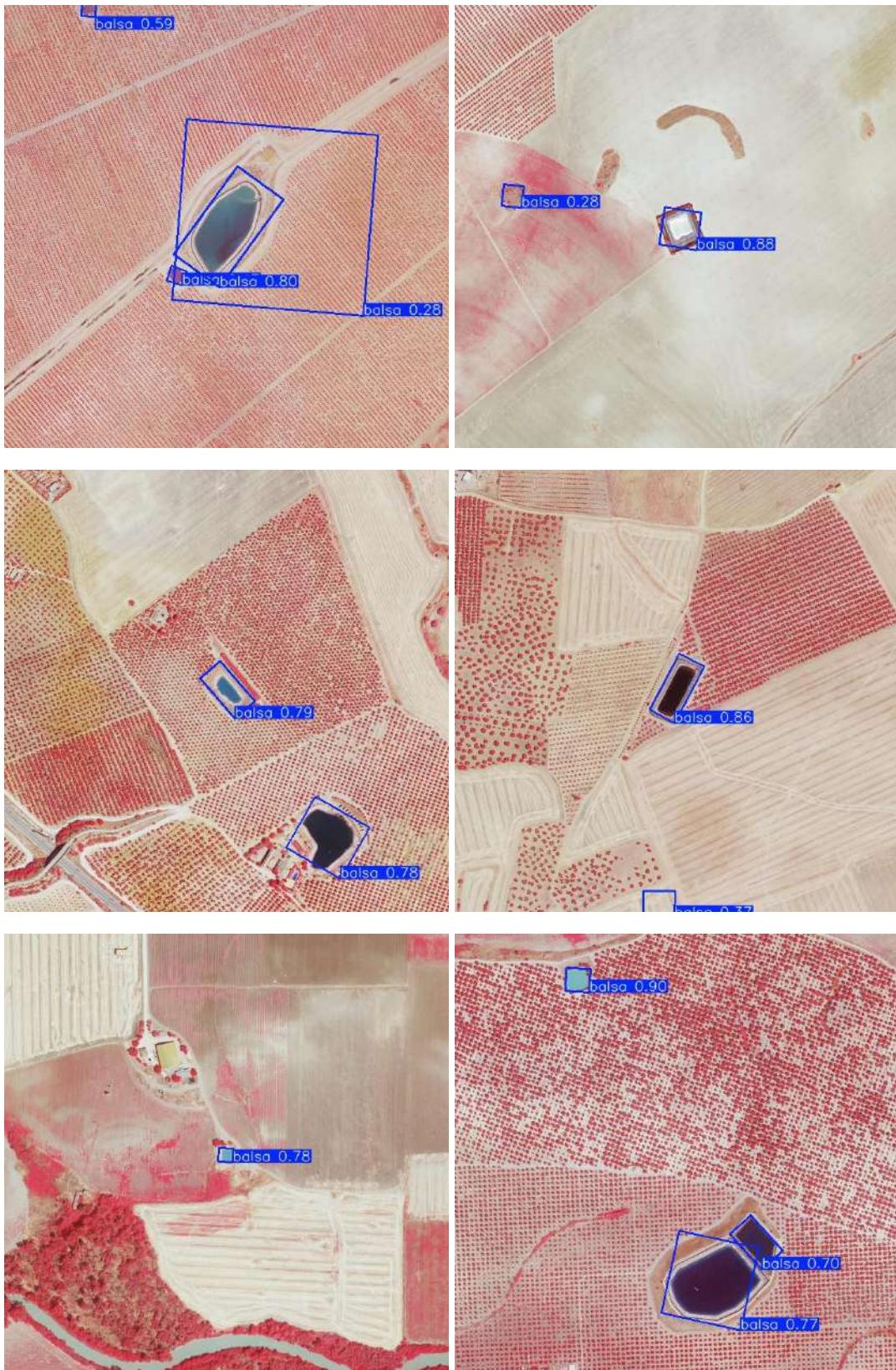


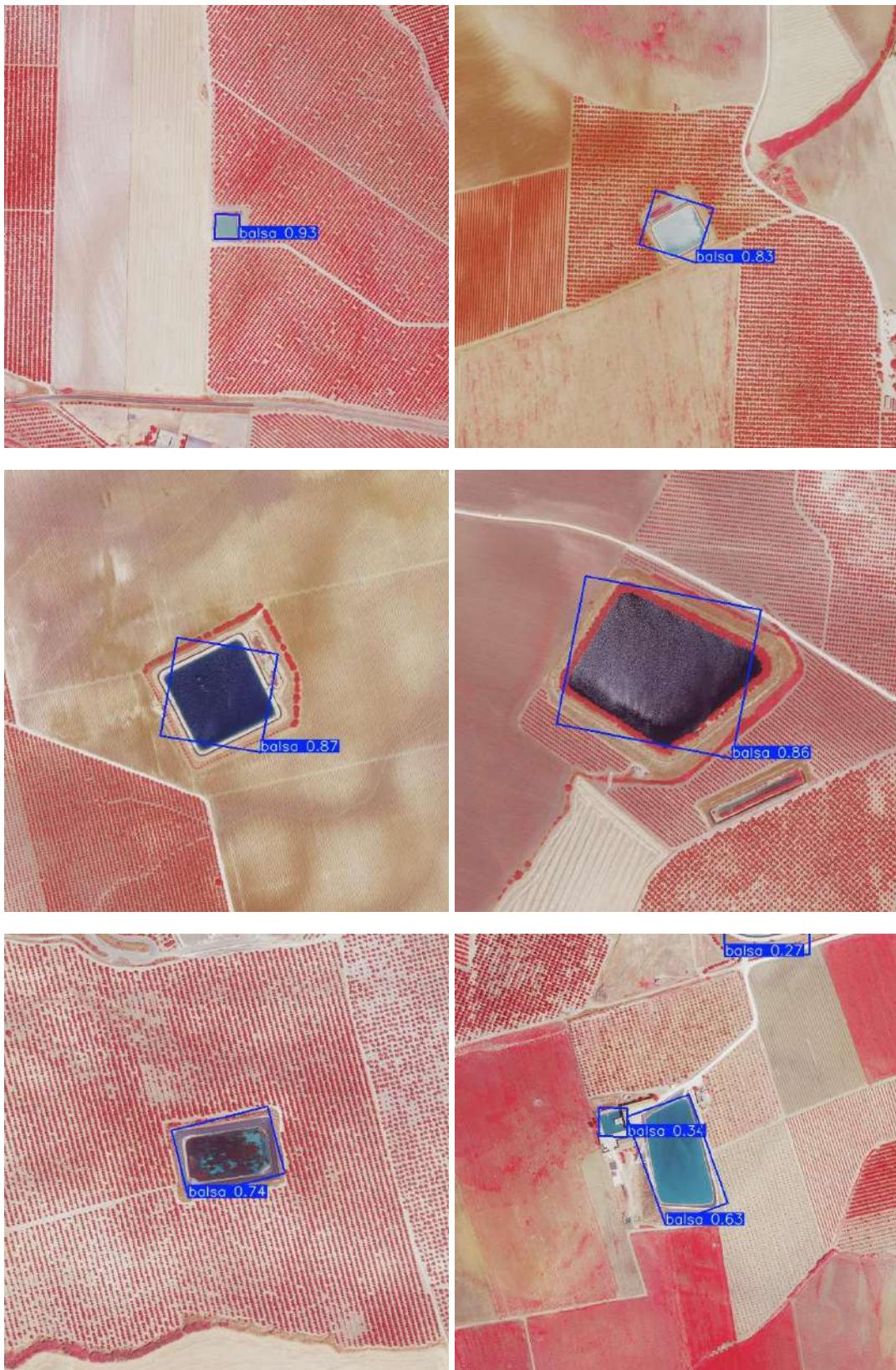


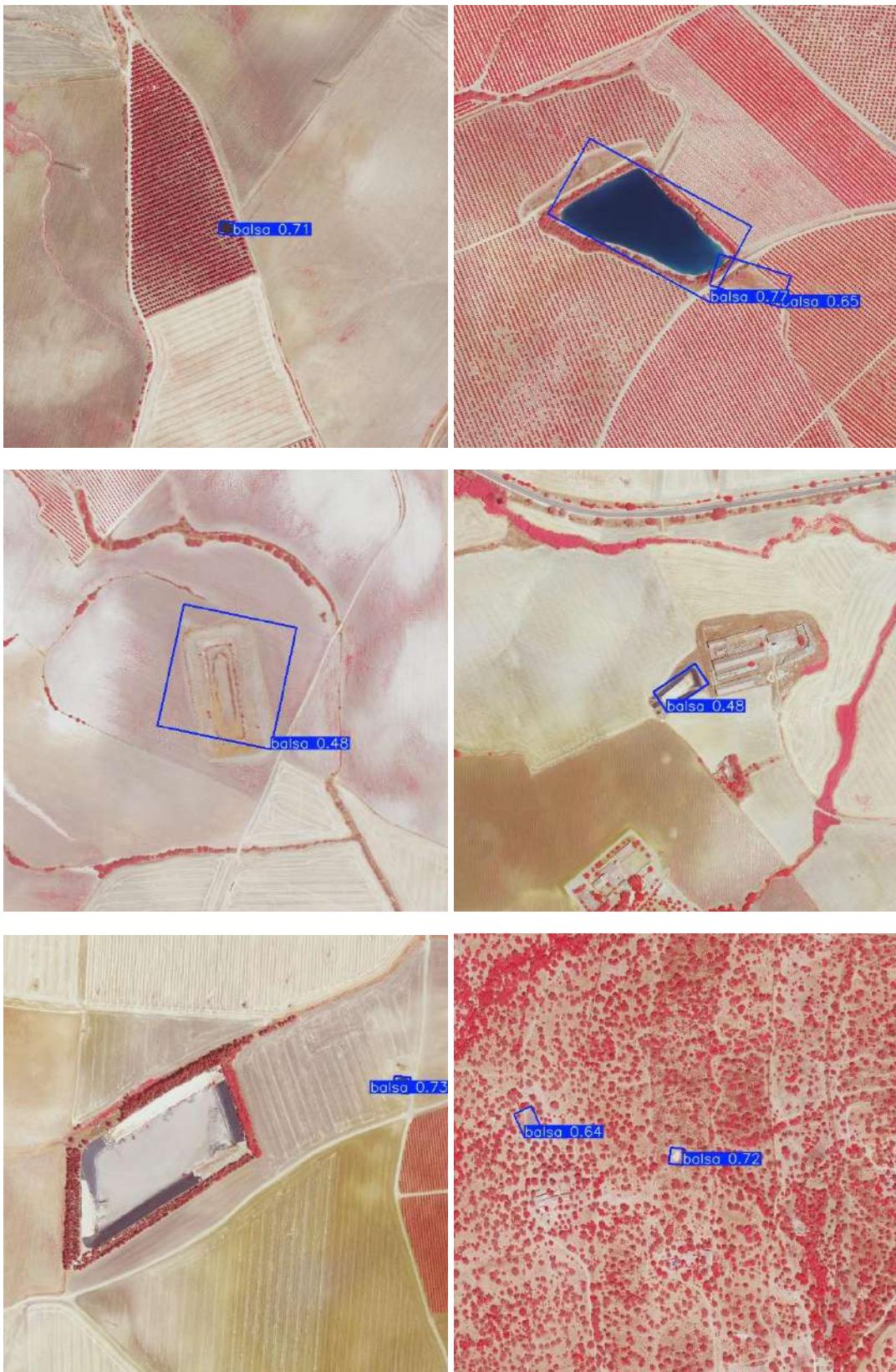


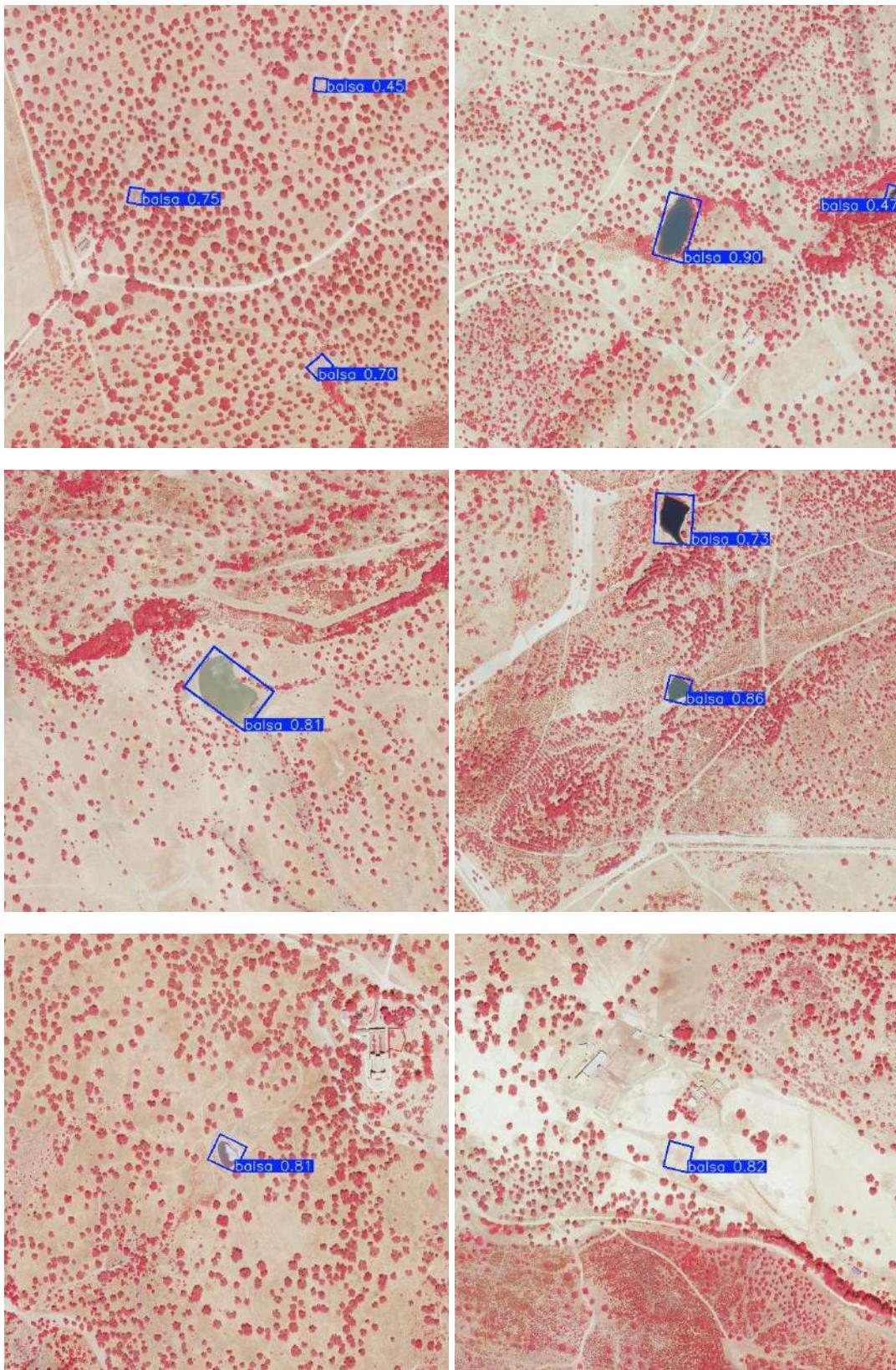


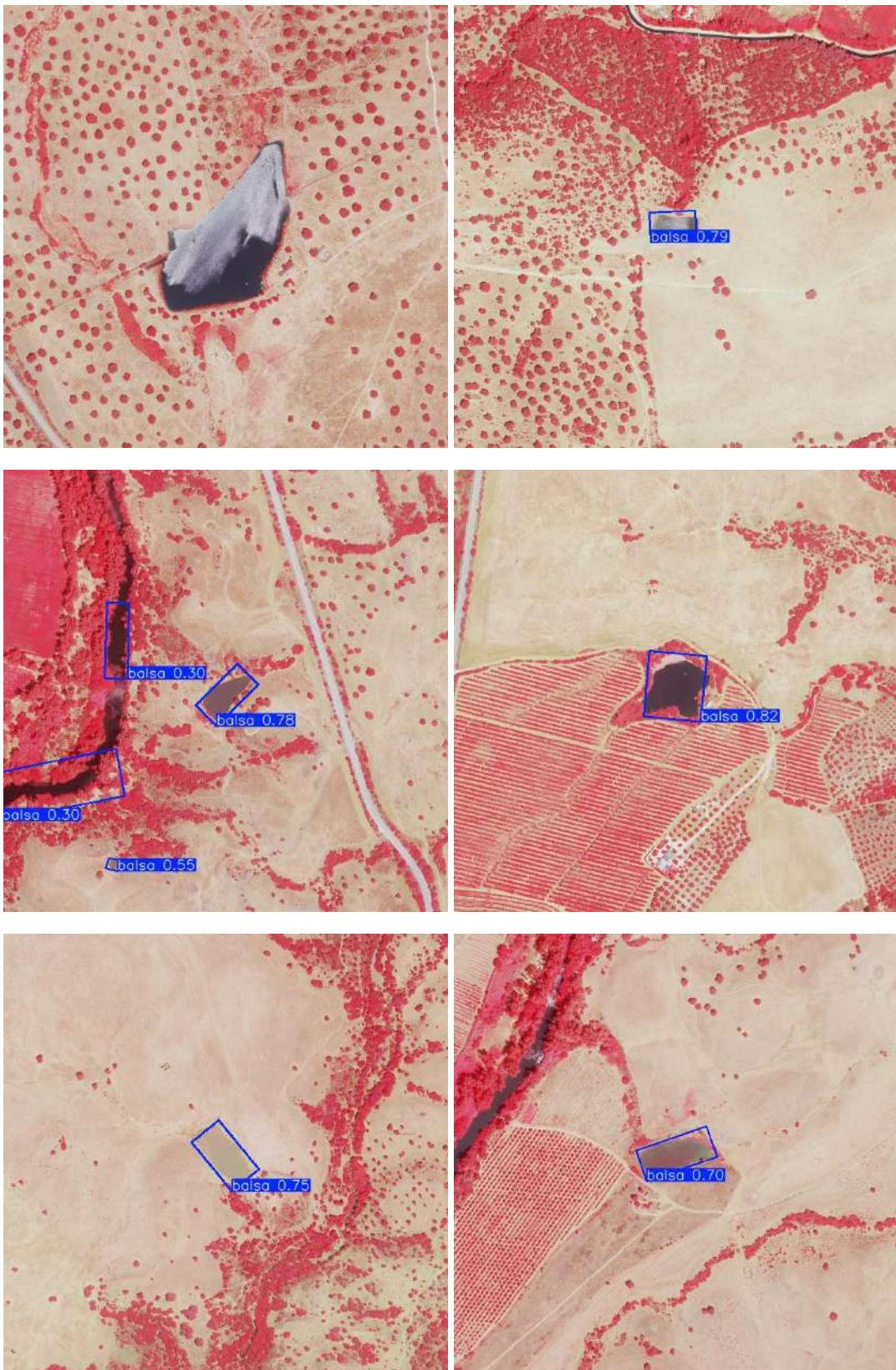


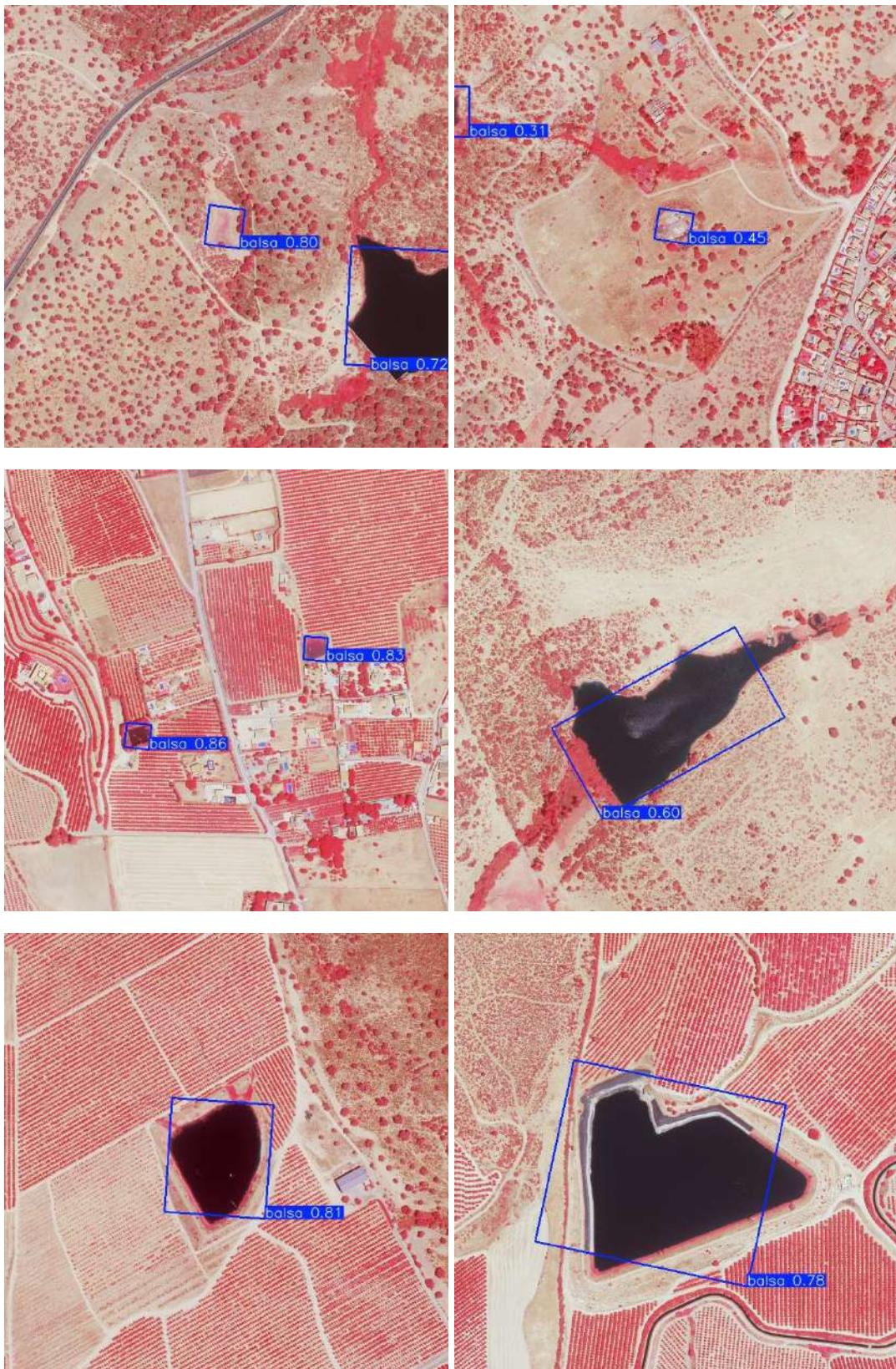
















Anexo 5: Notebooks y Modelo entrenado

Los notebooks con el código Python generado durante el desarrollo del proyecto y el archivo del modelo entrenado de detección de balsas se proporcionan en formato digital como archivos adjuntos a este Trabajo Fin de Máster.