UNIVERSIDAD DE LA REPÚBLICA URUGUAY

FACULTAD DE INGENIERÍA

Identificación de malezas mediante el procesamiento de imágenes multiespectrales

Informe de Proyecto de Grado presentado por

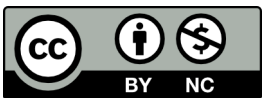
Domingo Mateo Bocchiardo Miguez
Gianfranco Caton Stefanoli Ortiz
Cristian González Núñez

en cumplimiento parcial de los requerimientos para la graduación de la carrera de Ingeniería en Computación de Facultad de Ingeniería de la Universidad de la República

Supervisores

Mercedes Marzoa
Facundo Benavides

Montevideo, 10 de junio de 2026



Agradecimientos

Queremos expresar nuestro más sincero agradecimiento a Mercedes Marzoa y Facundo Benavides, por su guía, disposición y acompañamiento a lo largo de este proyecto. Sus comentarios, sugerencias y apoyo fueron fundamentales para orientar el trabajo y sostener su desarrollo en cada etapa.

Agradecemos también al grupo MINA de la Facultad de Ingeniería por facilitar las capturas utilizadas en este trabajo y por el apoyo técnico brindado durante el proyecto. Asimismo, agradecemos a ClusterUY por la infraestructura de cómputo que hizo posible la experimentación realizada.

Extendemos además nuestro reconocimiento a la Facultad de Ingeniería y a todos los docentes que formaron parte de nuestra carrera, por la formación recibida y por brindarnos las herramientas que hicieron posible llegar hasta esta etapa.

Finalmente, agradecemos a nuestras familias, amigos y seres queridos, por el apoyo constante, la paciencia y el aliento durante todo este proceso. Su acompañamiento fue fundamental para poder culminar este trabajo.

Resumen

Las malezas constituyen una amenaza importante para la productividad agrícola por su competencia con los cultivos y por el aumento de poblaciones resistentes a herbicidas. En particular, la yerba carnicera (*Conyza* spp.) representa un problema relevante en sistemas productivos nacionales. En este contexto, las imágenes multiespectrales capturadas desde vehículos aéreos no tripulados (UAV) ofrecen una alternativa promisoria para apoyar estrategias de detección y manejo sitio-específico de malezas. Sin embargo, la disponibilidad de datasets multiespectrales adquiridos con UAV para detección de malezas es todavía limitada a nivel internacional, y más aún para malezas, cultivos y condiciones de suelo representativas de Uruguay.

Este trabajo aborda la construcción de un dataset multiespectral para detección de malezas en cultivos nacionales a partir de capturas realizadas con UAV, una tarea especialmente desafiante por la alta resolución de las imágenes, la necesidad de calibración radiométrica y la alineación precisa entre bandas espectrales. Se definió un flujo reproducible de preprocesamiento radiométrico y geométrico, se establecieron criterios consistentes de etiquetado y se realizó una caracterización estadística del conjunto final. El dataset obtenido contiene 500 imágenes multiespectrales etiquetadas manualmente y 6459 instancias distribuidas en tres clases morfológicas: plántula, roseta y dentada.

El dataset fue evaluado mediante segmentación de instancias con YOLOv8-s-seg y segmentación semántica con U-Net, comparando distintas configuraciones espectrales de entrada. Los resultados muestran que la incorporación de información multiespectral mejora el desempeño respecto de la línea base RGB, y que la configuración RGB + NIR fue la más robusta en ambas tareas. En particular, alcanzó un Mask mAP50-95 de 0,598 en segmentación de instancias y un Dice de 0,870 con IoU de 0,842 en segmentación semántica. En conjunto, estos resultados indican que el dataset construido es adecuado para entrenar modelos de detección de malezas en condiciones de campo reales, y que en las condiciones evaluadas la inclusión de la banda NIR sobre la línea base RGB resultó en el mejor desempeño para la identificación de *Conyza* spp.

Palabras clave: Detección de malezas, Agricultura de precisión, UAV, Yerba carnicera, Agronomía uruguaya, CNN, Multiespectral, SAHI

Índice general

Capítulo 1

Introducción

Las malezas representan una de las principales amenazas para la productividad agrícola, ya que compiten con los cultivos por recursos como agua, nutrientes y luz. A nivel global, se estima que las pérdidas de rendimiento por malezas oscilan entre el 10 % y el 12 % de la producción agrícola, y pueden superar el 40 % en regiones donde se aplican escasas medidas de control (Parven, Md Meftaul, Venkateswarlu, y Megharaj, 2024). Además, el uso intensivo de herbicidas genera resistencia en especies invasoras y efectos ambientales adversos; se estima que solo el 45 % del herbicida aplicado alcanza su objetivo (Parven y cols., 2024). En Uruguay, donde el sector agropecuario tiene un peso relevante en la economía (Uruguay XXI, 2024), las malezas resistentes a herbicidas afectan gran parte del área agrícola y condicionan las secuencias de cultivo y la carga de herbicidas (Kaspary y cols., 2024). En particular, la yerba carnicera (*Conyza* spp.) está presente durante todo el año en los sistemas de secano del país, posee una elevada capacidad reproductiva y de dispersión, y presenta casos confirmados de resistencia múltiple a glifosato e inhibidores de ALS en Uruguay (Kaspary y cols., 2024), lo que la convierte en una de las malezas más problemáticas del contexto productivo nacional.

En este contexto, el manejo sitio-específico de malezas (*Site-Specific Weed Management*, SSWM) propone intervenir únicamente en zonas con presencia real de maleza (Celikkan y cols., 2025). Para que este enfoque sea viable, se requieren herramientas de detección robustas y, por tanto, datasets adecuados. En particular, los vehículos aéreos no tripulados (*Unmanned Aerial Vehicles*, UAV) equipados con cámaras multiespectrales constituyen una plataforma especialmente útil para este problema. Gracias a su movilidad, los UAV relevan grandes superficies en poco tiempo y acceden a zonas de difícil monitoreo desde el suelo, lo que reduce el tiempo y el esfuerzo frente al relevamiento manual a pie. A partir de sus capturas es posible generar mapas preliminares de infestación que ubican las zonas afectadas dentro del lote y permiten dirigir el control terrestre hacia esas regiones, en lugar de recorrer y tratar el lote completo de forma uniforme (Sa, Chen, y cols., 2018). Además, los UAV adquieren imágenes de alta resolución espacial en condiciones reales de campo, con un nivel de deta-

lle difícil de obtener mediante otras alternativas áereas. A su vez, las imágenes multiespectrales aportan información que no está disponible en RGB y permiten calcular índices de vegetación útiles para mejorar la separabilidad entre cultivo y maleza (Sa, Chen, y cols., 2018). Sin embargo, la disponibilidad de datasets públicos multiespectrales adquiridos con UAV es limitada y no se cuenta con uno orientado a cultivos nacionales y malezas presentes en Uruguay, lo que dificulta la transferencia de métodos desarrollados en otros contextos (Celikkan y cols., 2025).

El presente trabajo se desarrolla en el marco de un proyecto del grupo MINA[1] de la Facultad de Ingeniería de la Universidad de la República, orientado a la investigación de técnicas para el control eficiente de malezas. Las capturas multiespectrales utilizadas fueron adquiridas por dicho grupo sobre un cultivo de vid en el departamento de Canelones, Uruguay.

El objetivo general de este trabajo es formalizar y ejecutar el proceso de construcción de un dataset multiespectral para detección de malezas en cultivos nacionales a partir de capturas realizadas desde un UAV, así como estudiar el uso de imágenes multiespectrales en algoritmos de aprendizaje profundo orientados a la detección y segmentación de malezas en condiciones reales de campo. El caso de estudio se focaliza en la yerba carnicera (*Conyza* spp.) en un cultivo de vid, por la relevancia agronómica de esta especie en sistemas productivos nacionales, aunque la metodología desarrollada podría extenderse a otras malezas y cultivos con las adaptaciones correspondientes en la adquisición de imágenes y los criterios de etiquetado. En particular, se relevan y seleccionan capturas representativas, se define un flujo reproducible de preprocesamiento radiométrico y geométrico, se establecen criterios consistentes de etiquetado con control de calidad y se realiza una caracterización estadística del dataset resultante.

Asimismo, el dataset se utiliza para evaluar experimentalmente el aporte de la información multiespectral al desempeño de modelos de aprendizaje profundo. Para ello se abordan dos tareas complementarias: segmentación de instancias con YOLOv8-s-seg y segmentación semántica con U-Net. En ambas líneas se comparan distintas configuraciones espectrales de entrada, incluyendo combinaciones de bandas e índices de vegetación, con el fin de analizar su efecto sobre la detección de malezas. El alcance del trabajo se centra en la construcción, documentación y evaluación experimental del dataset, si bien las condiciones específicas de adquisición limitan la generalización de los resultados, como se discute en el Capítulo 6.

1.1. Organización del documento

El informe se estructura de manera progresiva para acompañar el desarrollo del problema, desde sus fundamentos hasta la evaluación final de la propuesta. En primer lugar, el Capítulo 2 presenta el marco teórico que sustenta el trabajo, introduciendo los conceptos de manejo sitio-específico de malezas, la relevancia agronómica de *Conyza* spp. y los principios del uso de imágenes multiespectrales

[1] https://www.fing.edu.uy/inco/grupos/mina/

en agricultura de precisión. A continuación, el Capítulo 3 revisa los antecedentes más relevantes y sintetiza las principales brechas identificadas en la literatura, especialmente en lo referido a la disponibilidad de datasets multiespectrales aplicables al contexto nacional.

Luego, el Capítulo 4 desarrolla la parte central del trabajo, describiendo en detalle el proceso de construcción del dataset: adquisición de imágenes, preprocesamiento radiométrico y geométrico, definición de criterios de etiquetado y análisis estadístico del conjunto resultante. El Capítulo 5 presenta la experimentación realizada sobre el dataset y discute los resultados obtenidos en función de las hipótesis planteadas. Finalmente, el Capítulo 6 expone las conclusiones del estudio, sus limitaciones y las líneas de trabajo futuro. Como cierre, se incluyen las referencias bibliográficas y anexos con material complementario para facilitar la trazabilidad y reproducibilidad del trabajo.

Capítulo 2

Marco Teórico

En este capítulo se presentan los conceptos necesarios para comprender la construcción de un dataset multiespectral orientado a la detección de malezas en cultivos nacionales. El foco del trabajo está puesto en la identificación de la yerba carnicera mediante imágenes adquiridas con sensores multiespectrales y técnicas de visión por computadora. En agricultura de precisión, este problema se vincula con estrategias de manejo localizado y con sistemas de mapeo automático de malezas (Sa, Popović, y cols., 2018; Castellano, De Marinis, y Vessio, 2023; Celikkan y cols., 2025).

2.1. Malezas

Las malezas son especies vegetales que crecen de forma no deseada dentro del sistema productivo y compiten con el cultivo por recursos como luz, agua y nutrientes. Esta competencia reduce el rendimiento y la calidad, además de incrementar los costos de manejo (Irigoyen y Perrachón, 2012; Miranda, 2015). Aunque solo una parte de las especies vegetales se comporta como maleza, su impacto agronómico es alto por su capacidad de adaptación y persistencia en el sistema (Irigoyen y Perrachón, 2012).

Desde un punto de vista poblacional, las malezas problema combinan cuatro rasgos: (i) alta producción de diásporas, (ii) longevidad del banco de semillas, (iii) variabilidad en el ciclo de desarrollo y (iv) rápida respuesta adaptativa a la presión de control (química y no química).

En este contexto, el enfoque SSWM propone intervenir únicamente en zonas con presencia real de maleza, en lugar de aplicar tratamientos uniformes en todo el lote. Para que esta estrategia sea viable, se requieren métodos de detección confiables, georreferenciables y operativamente transferibles a campo (Sa, Popović, y cols., 2018; Goel, Kapur, y Vuppuluri, 2024; Qu y Su, 2024; Gerhards y cols., 2022; López-Granados, 2011).

2.1.1. Yerba carnicera (*Conyza* spp.)

La maleza de principal interés en este trabajo es la yerba carnicera (*Conyza bonariensis* y *Conyza sumatrensis*), ampliamente adaptada a los sistemas productivos uruguayos (Kaspary y cols., 2024; Miranda, 2015). Su importancia agronómica se asocia a su elevada capacidad competitiva, su alta producción de semillas (con valores reportados desde centenas de miles hasta más de 800 000 semillas por planta según especie, ambiente y condición de evaluación) y su eficiente dispersión principalmente por viento (Irigoyen y Perrachón, 2012; Kaspary y cols., 2024). En las Figuras 2.1 y 2.2 se presentan ejemplos visuales de *Conyza* spp. y un detalle de sus semillas, utilizados como referencia en este marco teórico.

Figura 2.1: Ejemplos visuales de yerba carnicera (*Conyza* spp.). Imagen tomada de Irigoyen y Perrachón (Irigoyen y Perrachón, 2012).

Figura 2.2: Detalle de semillas de *Conyza* con estructuras que favorecen su dispersión por viento. Imagen tomada de Kaspary et al. (Kaspary y cols., 2024).

En los últimos años también se ha reportado un aumento de resistencia a herbicidas en poblaciones de *Conyza*. En evaluaciones recientes más del 85 % de las poblaciones analizadas presentó resistencia múltiple a glifosato y al menos un inhibidor de ALS (acetolactato sintasa), mientras que una proporción muy baja resultó susceptible a todos los herbicidas evaluados (Kaspary y cols., 2024). Este escenario confirma que el problema excede el control químico generalizado

y requiere estrategias integradas de manejo (Kaspary y cols., 2024; Irigoyen y Perrachón, 2012).

Además de la resistencia a herbicidas, *Conyza* presenta rasgos biológicos que condicionan su monitoreo: germinación favorecida por luz, emergencia principalmente superficial, elevada producción y dispersión de semillas, y presencia simultánea de distintos estados fenológicos en el lote (Irigoyen y Perrachón, 2012; Miranda, 2015; Kaspary y cols., 2024). En términos operativos, esta dinámica incrementa la heterogeneidad espacial y temporal de la infestación, por lo que la detección debe contemplar variabilidad de tamaño, arquitectura y contexto de fondo (suelo, rastrojo y cultivo).

2.1.2. Implicancias para visión por computadora

El problema de detección de malezas por visión por computadora posee varios obstáculos que dificultan su solución. Entre ellos se encuentran: (i) el hecho de que las malezas, como las demás plantas, poseen diferente morfología según su estadio fenológico; (ii) la influencia de la iluminación y las condiciones ambientales que afectan la generalización de los métodos de detección; y (iii) la oclusión de las malezas por plantas, sombras, objetos, cultivos e incluso otras malezas (Z. Wu, Chen, Zhao, Kang, y Ding, 2021).

En particular, para la detección de la yerba carnicera existen desafíos por la variabilidad de tamaño, densidad y estado fenológico, junto con similitudes visuales con otras plantas presentes en el cultivo, especialmente en el estado de plántula (Miranda, 2015; Kaspary y cols., 2024; Sa, Popović, y cols., 2018; Castellano y cols., 2023; Celikkan y cols., 2025). A su vez, la heterogeneidad del fondo (suelo desnudo, rastrojo y cobertura parcial del cultivo) incrementa la variabilidad intra-clase y dificulta la generalización de los modelos.

En este contexto, el diseño del dataset debe contemplar diversidad en los distintos estadios de las malezas de la especie *Conyza* y, al mismo tiempo, controlar en la medida de lo posible factores de adquisición que pueden modificar la apariencia visual y radiométrica de las imágenes, como cambios de iluminación, nubosidad, sombras proyectadas, humedad del suelo, viento y hora del día.

2.2. Imágenes multiespectrales

Una imagen normal en color (RGB) está compuesta por tres bandas del espectro visible para el ojo humano: rojo (R), verde (G) y azul (B). En cambio, una imagen multiespectral se denomina así porque registra información en varias bandas espectrales diferenciadas, incluyendo bandas fuera del alcance del ojo humano. En este trabajo se emplean B, G, R, red-edge (RE), near-infrared (NIR) y pancromática (P), con rangos aproximados de 443–507 nm (B), 533–587 nm (G), 652–684 nm (R), 705–729 nm (RE), 785–899 nm (NIR) y 171–1098 nm (P). Mientras B, G y R pertenecen al visible (aprox. 400–700 nm), RE se ubica en la transición del rojo al infrarrojo cercano y NIR corresponde a radiación no visible para el ojo humano, aportando mayor sensibilidad a variables biofísicas

de la vegetación (Candiago, Remondino, De Giglio, Dubbini, y Gattelli, 2015; Sa, Popović, y cols., 2018). La Figura 2.3 presenta una representación conceptual de estas bandas y su ubicación relativa dentro del espectro electromagnético.

La utilidad de estas bandas en la agricultura se explica por la respuesta espectral de las hojas. La reflectancia, definida como la fracción de radiación incidente que es reflejada por una superficie, varía según la longitud de onda y las propiedades del material (Schaepman-Strub, Schaepman, Painter, Dangel, y Martonchik, 2006). En el espectro visible, la reflectancia es baja por absorción de clorofila; en NIR, la reflectancia aumenta por dispersión interna en la estructura foliar; y en longitudes de onda mayores, la señal se ve afectada por absorción de agua (Knipling, 1970; Tucker, 1979). Además, la región RE es especialmente útil para estimar clorofila y parámetros de vigor, por su sensibilidad a cambios en el contenido fotosintético (Horler, Dockray, Barber, y Barringer, 1983; Delegido, Verrelst, Alonso, y Moreno, 2011). Por ello, superficies que lucen similares en RGB pueden separarse con mayor robustez en las bandas RE y NIR.

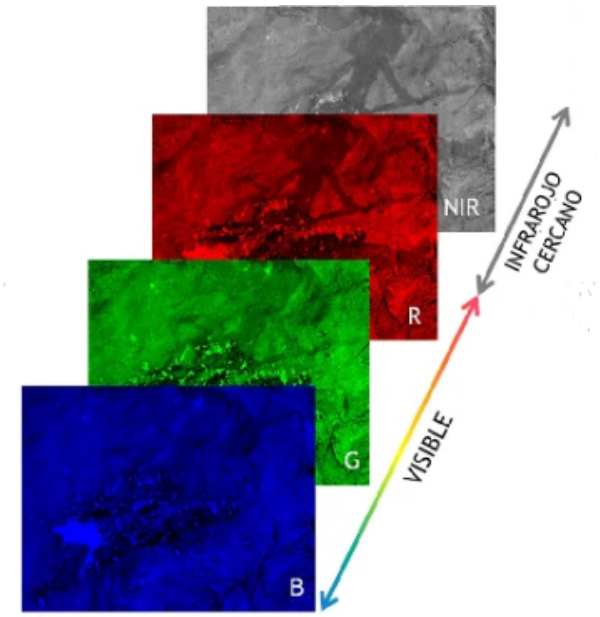


Figura 2.3: Representación conceptual de las bandas B, G, R y NIR sobre el espectro electromagnético, destacando la transición entre la región visible y el infrarrojo cercano. Adaptado de Sebastián et al. (Sebastián López y cols., 2013).

2.3. Índices de vegetación

Los índices de vegetación son combinaciones algebraicas entre bandas espectrales diseñadas para resaltar propiedades biofísicas de la vegetación. Su

principal ventaja es que reducen parte de la variabilidad causada por cambios de iluminación y facilitan la separación entre vegetación activa, suelo y otros elementos del fondo (Jackson y Huete, 1991).

Entre los índices más utilizados en aplicaciones agrícolas se encuentran los siguientes, donde R, G, RE y NIR representan reflectancias en las bandas roja, verde, red-edge e infrarrojo cercano, y L es un factor de ajuste del fondo del suelo:

$$NDVI = \frac{NIR - R}{NIR + R} \tag{2.1}$$

$$GNDVI = \frac{NIR - G}{NIR + G} \tag{2.2}$$

$$NDRE = \frac{NIR - RE}{NIR + RE} \tag{2.3}$$

$$SAVI = \frac{(1 + L)(NIR - R)}{NIR + R + L} \tag{2.4}$$

El índice NDVI (*Normalized Difference Vegetation Index*) es una de las formulaciones más utilizadas para estimar el vigor de la vegetación (Tucker, 1979; Jiang y cols., 2006). El índice GNDVI (*Green Normalized Difference Vegetation Index*) enfatiza la respuesta en la banda verde y se asocia al contenido de clorofila (Candiago y cols., 2015; Gitelson, Kaufman, y Merzlyak, 1996). El NDRE (*Normalized Difference Red-Edge Index*), basado en la banda red-edge, resulta útil para captar variaciones de la vegetación en etapas donde NDVI puede saturarse (Delegido y cols., 2011; Tilse, Siegmann, Börner, Menz, y Münz, 2025). Por su parte, SAVI (*Soil Adjusted Vegetation Index*) incorpora un factor de ajuste para atenuar la influencia del suelo expuesto (Huete, 1988).

Para detección de malezas, estos índices pueden incorporarse como canales adicionales del modelo o como variables derivadas para análisis exploratorio. En particular, aportan señales complementarias cuando la separación entre maleza y cultivo en RGB es baja.

2.4. Segmentación de imágenes

La segmentación de imágenes es el proceso de dividir una imagen en regiones con distintas características para extraer las regiones de interés. Estas regiones, de acuerdo con la percepción visual humana, deben ser significativas y no superpuestas (Yu y cols., 2023). En aplicaciones agrícolas, esta formulación permite delimitar con mayor precisión las regiones de cultivo, suelo y maleza, lo cual es útil para mapeo, estimación de cobertura y aplicación selectiva de control (Castellano y cols., 2023; Celikkan y cols., 2025).

A diferencia de la clasificación de imágenes (una etiqueta para toda la imagen) o de la detección por cajas delimitadoras, la segmentación produce máscaras espaciales detalladas (Minaee y cols., 2022; Long, Shelhamer, y Darrell,

2015). Esto mejora la interpretación agronómica del resultado y habilita métricas basadas en el área infestada.

2.4.1. Segmentación de instancias

La segmentación de instancias es la tarea que predice las regiones de píxeles contenidas en cada instancia individual de los objetos presentes en una imagen (Yu y cols., 2023).

En el caso de la detección de malezas, este enfoque segmenta cada objeto individual por separado, generando un polígono para cada planta detectada. En este enfoque, dos plantas de la misma clase se representan como dos instancias distintas (He, Gkioxari, Dollár, y Girshick, 2017).

Esta estrategia es útil cuando interesa contar individuos, analizar la distribución espacial o estudiar el tamaño por planta. Su principal desafío es el etiquetado, ya que requiere etiquetas más detalladas y suele ser más sensible a oclusiones y superposición entre plantas (Kirillov, He, Girshick, Rother, y Doll'ar, 2019; Celikkan y cols., 2025).

2.4.2. Segmentación semántica

La segmentación semántica asigna una clase a cada píxel sin distinguir objetos individuales (Yu y cols., 2023).

Este enfoque es especialmente adecuado para construir mapas de infestación y estimar cobertura de maleza, tareas directamente alineadas con estrategias SSWM. Además, suele requerir menor complejidad de etiquetado que la segmentación de instancias y permite entrenar modelos robustos cuando el objetivo principal es localizar zonas problemáticas más que contar plantas individuales (Sa, Popović, y cols., 2018; Castellano y cols., 2023; Goel y cols., 2024; Qu y Su, 2024).

La Figura 2.4 resume la diferencia conceptual entre segmentación semántica y segmentación de instancias. En la primera, todos los píxeles pertenecientes a una misma clase comparten una única etiqueta, mientras que en la segunda cada objeto individual se representa como una instancia separada.

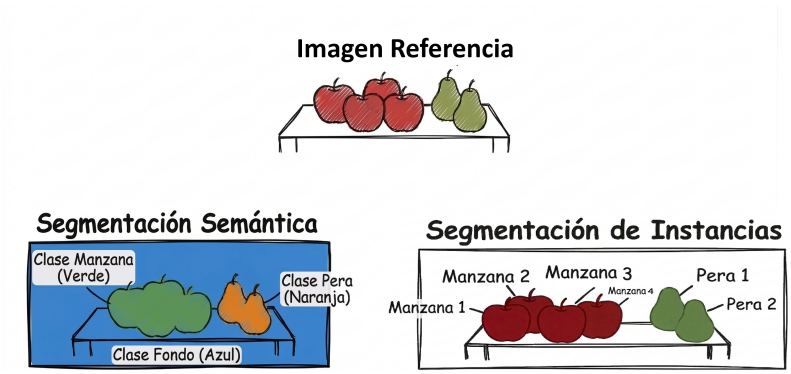


Figura 2.4: Comparación conceptual entre segmentación semántica y segmentación de instancias. En la segmentación semántica, todos los píxeles de una misma clase comparten la misma etiqueta; en la segmentación de instancias, además de la clase, se distingue cada objeto individual por separado.

2.5. Técnicas y modelos de aprendizaje profundo

En este trabajo, los modelos de aprendizaje profundo no constituyen el objetivo final en sí mismo, sino una herramienta para evaluar la utilidad y robustez del dataset construido. En particular, interesa analizar si el conjunto de datos permite entrenar detectores capaces de localizar malezas pequeñas en escenas de alta resolución y con información multiespectral. En las siguientes subsecciones se presentan técnicas y modelos utilizados en el proyecto.

2.5.1. Redes neuronales convolucionales

Las redes neuronales convolucionales (*Convolutional Neural Networks*, CNN) son arquitecturas especialmente adecuadas para el análisis de imágenes, ya que explotan la estructura espacial de los píxeles mediante filtros convolucionales (LeCun, Bottou, Bengio, y Haffner, 1998; Krizhevsky, Sutskever, y Hinton, 2012). Como se observa en la Figura 2.5, un filtro convolucional se desplaza sobre regiones locales de la imagen de entrada y genera un mapa de características que resume la respuesta del filtro ante esos patrones.

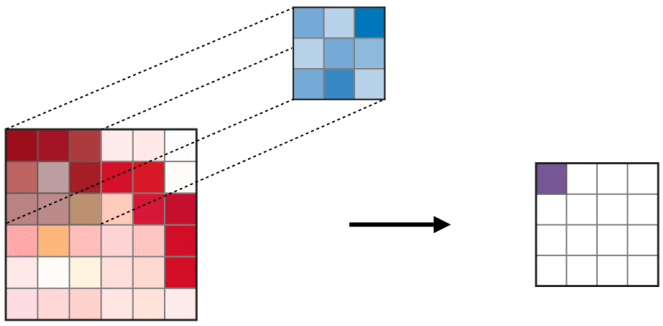


Figura 2.5: Ejemplo visual de aplicación de un filtro convolucional. El filtro, representado en azul, se desplaza sobre regiones locales de la imagen de entrada. En cada posición se combinan los valores de los píxeles cubiertos por el filtro para producir un valor en el mapa de características de salida, representado a la derecha. Tomada de (Amidi y Amidi, 2018).

En las redes convolucionales clásicas, este mecanismo permite aprender representaciones jerárquicas: las capas iniciales capturan bordes, texturas y patrones simples, mientras que las capas más profundas combinan esas respuestas para modelar estructuras de mayor complejidad. Este principio también está presente en arquitecturas modernas utilizadas en tareas de detección y segmentación, aunque estas incorporan otros componentes adicionales, como conexiones residuales, bloques encoder-decoder o cabezales específicos para predicción.

En visión por computadora, las CNN constituyen la base de numerosos modelos modernos de clasificación, detección y segmentación (Redmon, Divvala, Girshick, y Farhadi, 2016; Redmon y Farhadi, 2017).

2.5.2. Detectores de una etapa: YOLO

Dentro de los enfoques de detección de objetos, los modelos de una etapa se caracterizan por predecir directamente la ubicación y clase de los objetos en una sola pasada sobre la imagen. Entre ellos, la familia *You Only Look Once* (YOLO) se ha consolidado como una de las más utilizadas por su equilibrio entre velocidad y desempeño (Redmon y cols., 2016; Redmon y Farhadi, 2017). Versiones más recientes de esta arquitectura, como YOLOv8 (Jocher, Qiu, y Chaurasia, 2023), incorporan capacidades de segmentación de instancias, extendiendo las predicciones más allá de cajas delimitadoras para generar máscaras a nivel de píxel para cada objeto detectado. Esto permite caracterizar con mayor precisión la forma y distribución espacial de las instancias, lo que resulta especialmente relevante en aplicaciones agrícolas, donde la delimitación exacta de plantas o malezas aporta información más rica que una simple caja. Sin embargo, al trabajar con tamaños de entrada relativamente acotados, el reescalado de fotografías de gran resolución puede provocar pérdida de detalle espacial y

dificultar la detección de objetos pequeños (Akyon, Altinuc, y Temizel, 2022; Liu y cols., 2024).

2.5.3. U-Net, ResNet y EfficientNet

La arquitectura U-Net (Ronneberger, Fischer, y Brox, 2015) constituye uno de los enfoques más consolidados para tareas de segmentación semántica a nivel de píxel. Su diseño sigue una estructura encoder-decoder simétrica: el encoder contrae progresivamente la resolución espacial para capturar representaciones de alto nivel, mientras que el decoder la recupera mediante bloques de upsampling (operaciones que incrementan la resolución espacial). Un elemento central de esta arquitectura son las *skip connections*, que concatenan los mapas de características del camino contractivo con los del expansivo, preservando información espacial de alta frecuencia que de otro modo se perdería en las capas de pooling (operaciones que reducen la resolución espacial). Esto permite obtener predicciones precisas a nivel de píxel incluso cuando el conjunto de entrenamiento es de tamaño reducido.

Una práctica habitual consiste en reemplazar el encoder original de U-Net por una red preentrenada, aprovechando el aprendizaje previo sobre grandes conjuntos de datos. En este trabajo se utilizaron dos encoders preentrenados: ResNet18 (He, Zhang, Ren, y Sun, 2016) y EfficientNet-b0 (Tan y Le, 2020).

ResNet (He y cols., 2016) es una arquitectura basada en el aprendizaje residual que introduce conexiones de identidad entre capas para mitigar el problema de degradación del gradiente en redes profundas. EfficientNet (Tan y Le, 2020), por su parte, utiliza un método de escalado compuesto que balancea de forma conjunta la profundidad, el ancho y la resolución de la red, logrando una relación más eficiente entre cantidad de parámetros y desempeño. En ambos casos, el preentrenamiento sobre ImageNet permite transferir características jerárquicas de propósito general al dominio agrícola, reduciendo los requerimientos de datos etiquetados y favoreciendo la convergencia.

2.5.4. Detección de objetos pequeños en imágenes de alta resolución

La segmentación de objetos pequeños constituye un desafío importante cuando las imágenes originales tienen alta resolución y los objetos de interés ocupan una fracción muy reducida de la escena. En estos casos, el reescalado directo a tamaños de entrada convencionales puede provocar pérdida de detalle espacial y degradar la capacidad del detector para localizar instancias pequeñas (Akyon y cols., 2022; Liu y cols., 2024; Tessore, 2025). Este problema es especialmente relevante en aplicaciones agrícolas, donde las malezas pueden aparecer dispersas, parcialmente ocluidas y con tamaños reducidos respecto del fondo total de la imagen.

2.5.5. Slicing Aided Fine-Tuning (SF) y Slicing Aided Hyper Inference (SAHI)

Slicing Aided Fine-Tuning (SF) y *Slicing Aided Hyper Inference* (SAHI) son estrategias orientadas a mejorar la detección de objetos pequeños en imágenes de gran tamaño (Akyon y cols., 2022). Su idea central consiste en trabajar sobre recortes solapados de la imagen, de modo de preservar mayor detalle espacial que el que se mantiene al reescalar la imagen completa a un tamaño de entrada reducido.

En la etapa de entrenamiento, SF genera parches a partir de las imágenes originales y utiliza estos recortes como muestras de entrada para el detector. Esto aumenta la proporción de objetos pequeños visibles en cada ejemplo y favorece que el modelo aprenda patrones que podrían diluirse al entrenar directamente sobre imágenes reescaladas (Akyon y cols., 2022).

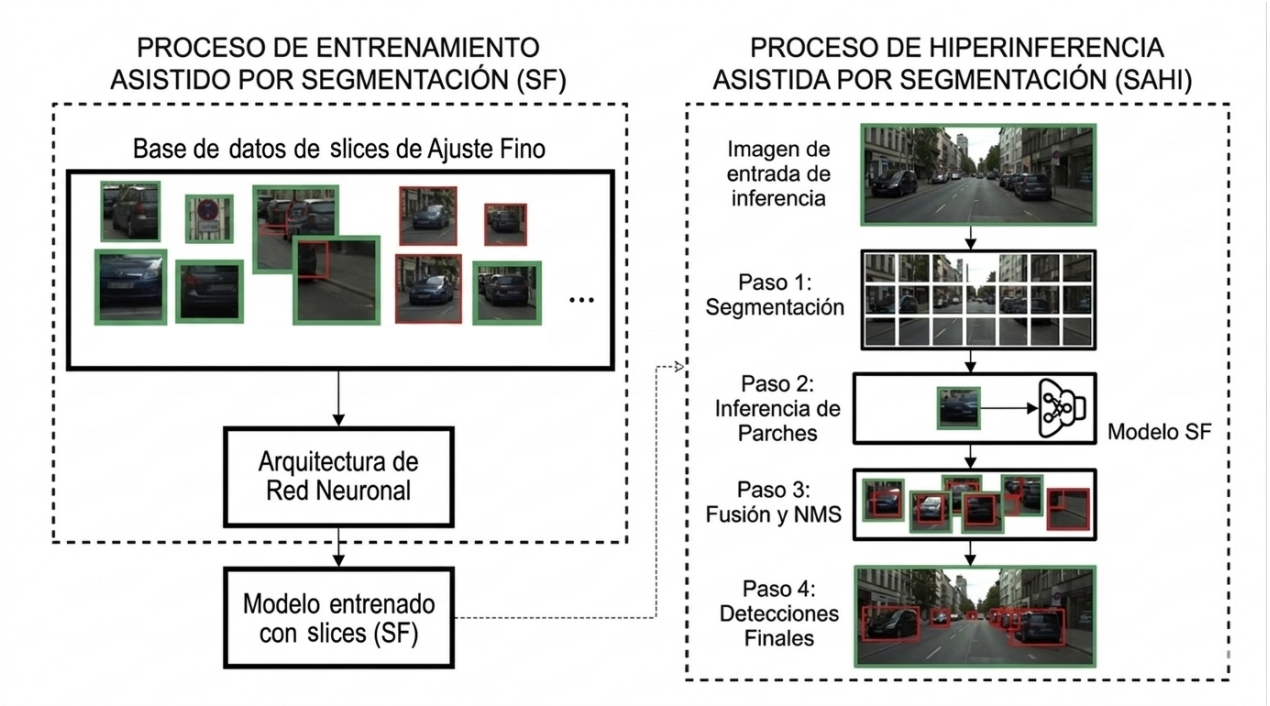


Figura 2.6: Esquema conceptual de *Slicing Aided Fine-Tuning* (SF) y *Slicing Aided Hyper Inference* (SAHI), mostrando el particionado de una imagen de alta resolución en recortes solapados y la posterior fusión de predicciones en el sistema de coordenadas original.

Como se observa en la Figura 2.6, en la etapa de inferencia SAHI divide la imagen en parches solapados, ejecuta la detección sobre cada uno de ellos y luego reproyecta las predicciones al sistema de coordenadas original. Posterior-

mente, las detecciones solapadas se fusionan mediante etapas de posprocesado, típicamente basadas en *Non-Maximum Suppression* (NMS), que ante múltiples detecciones superpuestas sobre un mismo objeto conserva únicamente la de mayor confianza, con el fin de eliminar duplicados y conservar un conjunto final coherente de cajas detectadas (Akyon y cols., 2022). De esta manera, se incrementa la probabilidad de detectar objetos pequeños o dispersos que podrían pasar desapercibidos si la inferencia se realizara sobre la imagen completa reducida (Akyon y cols., 2022; Tessore, 2025).

Estas estrategias resultan especialmente relevantes en aplicaciones agrícolas, donde las malezas suelen ocupar una fracción muy pequeña de la escena total. Como contrapartida, tanto SF como SAHI incrementan el costo computacional, debido a la necesidad de procesar múltiples recortes por imagen durante el entrenamiento, la inferencia, o ambos (Akyon y cols., 2022; Liu y cols., 2024).

2.5.6. Alineación de datos multiespectrales

En problemas de detección sobre imágenes multiespectrales, una decisión relevante consiste en cómo combinar la información proveniente de distintas bandas o variables derivadas. Una alternativa es la fusión a nivel de entrada, donde varias bandas espectrales e índices de vegetación se integran como canales de un mismo tensor de entrada para el modelo. Este enfoque permite explotar información complementaria respecto de RGB, aunque también exige consistencia en la calibración radiométrica, la alineación entre bandas y la normalización de los datos (Sa, Chen, y cols., 2018; Celikkan y cols., 2025; J. Wu, Wu, y Miao, 2025). En este contexto, la utilidad de la fusión no debe asumirse a priori, sino evaluarse experimentalmente sobre el dataset construido.

2.6. Métricas de evaluación

La evaluación de detectores de objetos requiere métricas que contemplen no solo la clasificación correcta de las instancias, sino también la calidad de su localización espacial. En este tipo de problemas, medidas como precisión, exhaustividad, *Intersection over Union* y *mean Average Precision* permiten caracterizar de forma complementaria el desempeño de los modelos y analizar su utilidad para el problema de interés (Everingham, Van Gool, Williams, Winn, y Zisserman, 2010; Lin y cols., 2014).

2.6.1. Precisión y exhaustividad

Para evaluar la detección de una clase de interés es necesario distinguir cuatro casos posibles. En este trabajo, la clase positiva corresponde a la presencia de maleza, mientras que la clase negativa corresponde al fondo u otros elementos que no pertenecen a la clase evaluada.

- Verdadero positivo (TP): una maleza presente en la imagen es detectada correctamente por el modelo.

- Falso positivo (FP): el modelo predice una maleza en una región donde no hay una maleza de la clase evaluada.
- Falso negativo (FN): una maleza presente en la imagen no es detectada por el modelo.
- Verdadero negativo (TN): una región que no contiene maleza es correctamente ignorada por el modelo.

La Figura 2.7 representa estos cuatro casos de forma conceptual. Los puntos rellenos corresponden a objetos reales de la clase positiva, mientras que los círculos vacíos representan elementos que no pertenecen a esa clase. La región encerrada por la elipse representa las predicciones positivas del modelo. Por lo tanto, los puntos rellenos dentro de la elipse son verdaderos positivos, los círculos vacíos dentro de la elipse son falsos positivos, los puntos rellenos fuera de la elipse son falsos negativos y los círculos vacíos fuera de la elipse son verdaderos negativos.

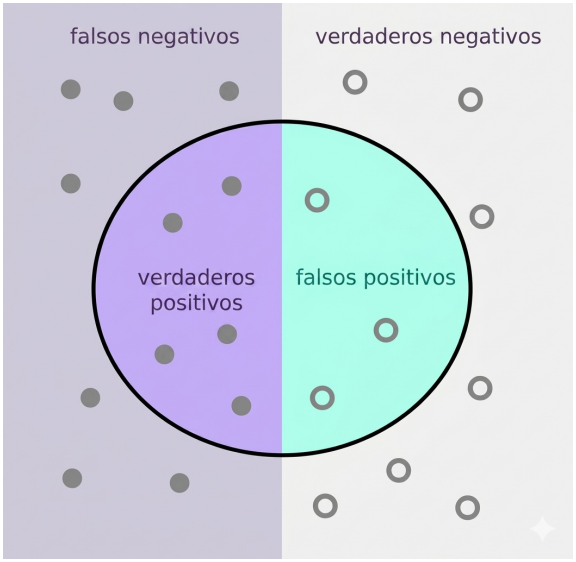


Figura 2.7: Representación visual de verdaderos positivos, falsos positivos, falsos negativos y verdaderos negativos para una clase positiva. Adaptado de (Zuccarelli, 2020)

A partir de estos casos se definen la precisión y la exhaustividad. La precisión mide qué proporción de las detecciones realizadas por el modelo fue correcta. Se calcula como:

$$Precision = \frac{TP}{TP + FP} \tag{2.5}$$

La exhaustividad, también denominada *recall*, mide qué proporción de los objetos reales de la clase evaluada fue detectada por el modelo. Se calcula como:

$$Recall = \frac{TP}{TP + FN} \tag{2.6}$$

Estas métricas permiten analizar errores complementarios. Una baja precisión indica que el modelo produce demasiados falsos positivos, lo que en detección de malezas puede llevar a sobreestimar la infestación. Una baja exhaustividad indica que el modelo deja sin detectar malezas reales, lo que puede afectar la utilidad agronómica del sistema al omitir zonas que requieren intervención (Everingham y cols., 2010).

2.6.2. Intersection over Union

La métrica *Intersection over Union* (IoU) cuantifica el grado de solapamiento entre una predicción y la etiqueta de referencia. Se define como el cociente entre el área de intersección y el área de unión de ambas regiones. En detección de objetos, esta métrica se utiliza como criterio para determinar si una predicción se considera correcta: cuanto mayor es el valor de IoU, mayor es la coincidencia espacial entre la caja predicha y la caja real (Everingham y cols., 2010; Lin y cols., 2014).

La Figura 2.8 ilustra el cálculo de esta métrica a partir del solapamiento entre una caja predicha y una caja real. En este caso, la región común a ambas cajas corresponde a la intersección, mientras que el área total cubierta por ambas regiones corresponde a la unión.

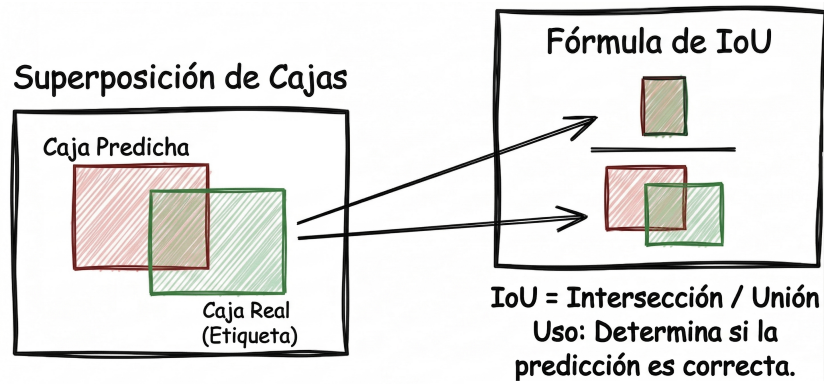


Figura 2.8: Ejemplo conceptual del cálculo de *Intersection over Union* (IoU) entre una caja predicha y una etiqueta. La métrica se obtiene como el cociente entre el área de intersección y el área de unión de ambas regiones.

2.6.3. Average Precision y mean Average Precision

La *Average Precision* (AP) resume el desempeño del detector integrando la relación entre precisión y exhaustividad a lo largo de distintos umbrales de confianza. Cuando se promedia este valor sobre una o varias clases, se obtiene la *mean Average Precision* (mAP), una de las métricas más utilizadas en detección de objetos (Everingham y cols., 2010; Lin y cols., 2014). En particular, mAP50 evalúa las detecciones correctas usando un umbral de $IoU = 0,5$, mientras que mAP50-95 promedia el desempeño sobre múltiples umbrales de IoU, desde 0,50 hasta 0,95, y constituye una medida más exigente de la calidad de localización (Lin y cols., 2014). La Figura 2.9 ilustra de forma conceptual la relación entre AP para una clase y mAP para múltiples clases, así como la diferencia entre mAP50 y mAP50-95.

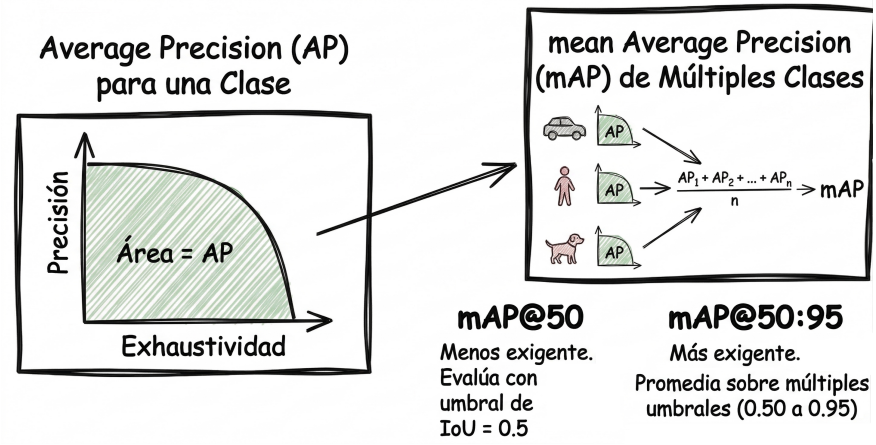


Figura 2.9: Representación conceptual de la *Average Precision* (AP) para una clase y de la *mean Average Precision* (mAP) como promedio sobre múltiples clases. También se ilustra la diferencia entre mAP50 y mAP50-95.

2.6.4. Coeficiente Dice

El coeficiente Dice, también conocido como *Dice Similarity Coefficient* (DSC), es una métrica utilizada para medir el grado de solapamiento entre una región predicha y su correspondiente etiqueta de referencia (Dice, 1945). En segmentación de imágenes, esta métrica resulta especialmente útil porque cuantifica de forma directa qué tan bien coinciden espacialmente las máscaras predichas con las máscaras reales.

Para una clase dada, el coeficiente Dice se define como:

$$Dice = \frac{2TP}{2TP + FP + FN} \tag{2.7}$$

donde TP representa la cantidad de verdaderos positivos, FP la cantidad de falsos positivos y FN la cantidad de falsos negativos.

De forma equivalente, si A representa la máscara de referencia y B la máscara predicha, el coeficiente puede expresarse como:

$$DSC = \frac{2|A \cap B|}{|A| + |B|} \tag{2.8}$$

donde $|A \cap B|$ representa el área de intersección entre ambas máscaras, $|A|$ el área de la máscara de referencia y $|B|$ el área de la máscara predicha. Esta formulación es equivalente a la anterior, ya que en segmentación binaria se cumple que $|A \cap B| = TP$, $|A| = TP + FN$ y $|B| = TP + FP$.

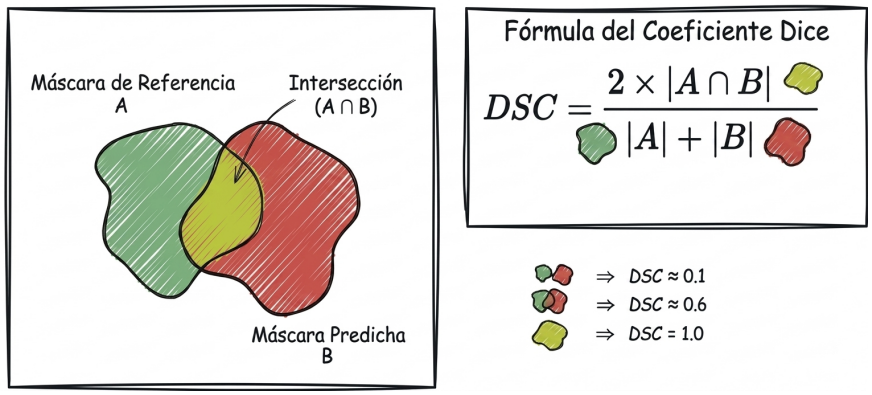


Figura 2.10: Interpretación visual del coeficiente Dice. La métrica cuantifica el solapamiento entre la máscara de referencia A y la máscara predicha B a partir del área de intersección entre ambas.

La Figura 2.10 ilustra esta interpretación geométrica del coeficiente Dice y muestra cómo su valor aumenta a medida que crece el solapamiento entre la predicción y la referencia.

El coeficiente Dice toma valores entre 0 y 1, donde 1 indica coincidencia perfecta entre la predicción y la etiqueta, y 0 indica ausencia total de solapamiento. En problemas de segmentación semántica, esta métrica es ampliamente utilizada porque resume de forma interpretable el equilibrio entre regiones correctamente detectadas y errores de sobresegmentación o subsegmentación (Dice, 1945).

2.6.5. F1-score

El *F1-score* es una métrica que resume en un único valor el compromiso entre precisión (*precision*) y exhaustividad (*recall*). Se define como la media armónica entre ambas magnitudes (van Rijsbergen, 1979; Sokolova y Lapalme, 2009):

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{2.9}$$

donde:

$$Precision = \frac{TP}{TP + FP} \tag{2.10}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.11}$$

En segmentación de imágenes, el *F1-score* permite evaluar de forma conjunta la capacidad del modelo para detectar correctamente los píxeles de una clase y evitar falsas detecciones. Un valor alto de *F1-score* indica que el modelo mantiene un buen equilibrio entre precisión y exhaustividad, lo que resulta deseable cuando interesa tanto detectar la mayor cantidad posible de regiones relevantes como minimizar los errores de clasificación (Sokolova y Lapalme, 2009).

2.6.6. Pixel Accuracy

La métrica *Pixel Accuracy* mide la proporción de píxeles correctamente clasificados respecto del total de píxeles evaluados. En segmentación semántica, esta métrica ofrece una medida global del desempeño del modelo a nivel de píxel (Long y cols., 2015).

Para el caso binario, puede expresarse como:

$$Pixel\ Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.12}$$

donde TN representa la cantidad de verdaderos negativos.

En el caso multiclase, la métrica se interpreta de manera análoga como la fracción de píxeles cuya clase fue predicha correctamente sobre el total de píxeles de la imagen o del conjunto evaluado. Su principal ventaja es que resulta simple e intuitiva de interpretar. Sin embargo, en problemas con fuerte desbalance entre clases, como ocurre frecuentemente en imágenes agrícolas donde el fondo ocupa gran parte de la escena, esta métrica puede presentar valores altos aun cuando el desempeño sobre las clases de interés no sea igualmente bueno. Por esta razón, suele complementarse con métricas de solapamiento como IoU o Dice (Long y cols., 2015).

2.6.7. Interpretación de métricas en detección de malezas

En el contexto de este trabajo, estas métricas no solo permiten comparar modelos, sino también analizar indirectamente la calidad del dataset construido. Un desempeño consistente en precisión, exhaustividad y mAP sugiere que el conjunto de datos contiene suficiente información para aprender patrones discriminativos y localizar adecuadamente las malezas. A la vez, diferencias marcadas entre configuraciones de entrada permiten estudiar el aporte de la información multiespectral frente a alternativas basadas únicamente en RGB.

Capítulo 3

Revisión de antecedentes

En este capítulo se revisan antecedentes relevantes para la detección de malezas mediante visión por computadora, con énfasis en el uso de imágenes multiespectrales adquiridas con UAV, la construcción de datasets y las estrategias de segmentación y mapeo aplicadas al contexto agrícola. La revisión incluye tanto herramientas comerciales como trabajos académicos vinculados a la adquisición de datos, el preprocesamiento radiométrico y geométrico, el diseño de etiquetas y la evaluación de modelos. El propósito es presentar los principales enfoques reportados en la literatura, así como sus aportes y limitaciones, a fin de establecer un marco de referencia para el desarrollo posterior del trabajo.

3.1. Evaluación de PIX4Dfields en detección de yerba carnicera

Como antecedente comercial, se consideró *PIX4Dfields*, una plataforma orientada al procesamiento agronómico de imágenes capturadas con drones. La herramienta permite generar ortomosaicos (imágenes compuestas, geométricamente corregidas y georreferenciadas, construidas a partir de múltiples fotografías aéreas superpuestas), índices de vegetación, mapas de vigor y mapas de prescripción para agricultura de precisión, con énfasis en un flujo de trabajo rápido, *offline* y orientado al análisis operativo de lotes completos.

Entre sus funcionalidades se encuentra *Magic Tool*, una herramienta de clasificación espectral supervisada que opera a partir de muestras seleccionadas por el usuario sobre el ortomosaico. A partir de estos ejemplos, la plataforma segmenta clases de interés, como cultivo, suelo o maleza, y produce mapas temáticos utilizables en tareas de monitoreo y prescripción.

Sobre esta base, se evaluó el desempeño de *Magic Tool* para la detección de yerba carnicera. Al descender al nivel de malezas puntuales y de tamaño pequeño, la herramienta mostró limitaciones asociadas a su esquema de clasificación espectral. En particular, se observó un *trade-off* entre exhaustividad (*recall*) y precisión, ya que no fue posible mantener simultáneamente valores altos de

ambas métricas en escenarios con malezas pequeñas o dispersas.

En un escenario de alta sensibilidad, al definir pocas restricciones para el fondo o permitir un rango espectral amplio para la clase maleza, se detectó la mayoría de las plantas objetivo. Sin embargo, también aumentaron los falsos positivos, por ejemplo sobre sombras, bordes de cultivo, suelo húmedo y huellas de maquinaria, lo que condujo a una sobreestimación de la infestación.

En un escenario de alta precisión, al entrenar rigurosamente la clase fondo para reducir ruido, los resultados fueron más limpios, pero el *recall* disminuyó de forma marcada. Como se observa en la Figura 3.1, esta configuración produce un mapa visualmente más depurado, pero deja sin detectar varias plantas de yerba carnicera presentes en la escena. En particular, la herramienta omitió malezas pequeñas que no llenaban la celda, plantas ubicadas en bordes de la grilla y ejemplares con firmas espectrales levemente distintas. En términos operativos, esto se traduce en falsos negativos relevantes para el control en campo, lo que sugiere que la herramienta resulta más adecuada para detección de malezas a gran escala que para el mapeo fino de individuos aislados.

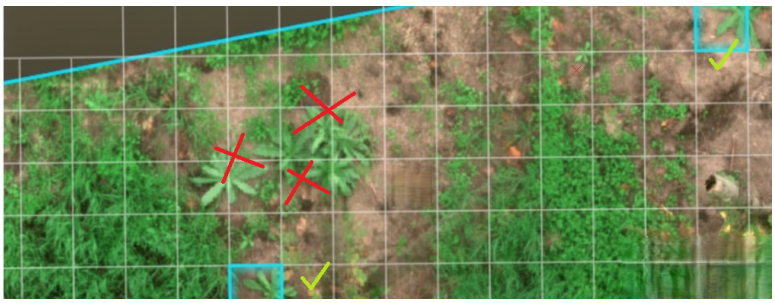


Figura 3.1: Escenario de alta precisión y bajo *recall* en *Magic Tool*. Las cruces rojas indican plantas de yerba carnicera presentes en la escena que no fueron detectadas por la herramienta, es decir, falsos negativos. Los recuadros celestes muestran regiones seleccionadas como detecciones de maleza por *Magic Tool*. Las marcas verdes indican detecciones consideradas correctas. La imagen evidencia que, aunque algunas detecciones son precisas, varias plantas presentes no son identificadas.

3.2. WeedMap: mapeo semántico de malezas sobre ortomosaicos multiespectrales

Un antecedente metodológico central es *WeedMap*, propuesto por Sa et al. (Sa, Chen, y cols., 2018), cuyo objetivo es desarrollar un sistema completo de segmentación semántica y mapeo de cultivo y maleza a partir de imágenes multiespectrales capturadas con UAV. El trabajo se concentra en el aprovechamiento de información multiespectral para generar mapas de infestación utilizables en estrategias de *Site-Specific Weed Management*. A diferencia de enfoques que

operan sobre imágenes aisladas, los autores plantean un flujo orientado a lotes completos, donde el resultado final no consiste únicamente en una predicción por imagen, sino en un mapa continuo del cultivo, georreferenciado y con detalle suficiente para apoyar decisiones de manejo localizado.

El estudio se desarrolla sobre cultivos de remolacha azucarera en Suiza y Alemania, utilizando ortomosaicos multiespectrales obtenidos con cámaras RedEdge-M y Sequoia. En total, el conjunto cubre aproximadamente 16 554 m^2 e incluye ocho ortomosaicos capturados a 10 m de altura, con resoluciones cercanas a 1 cm/píxel y fuerte presión de malezas. A partir de las bandas capturadas, los autores construyen distintas configuraciones de entrada con RGB, RE, NIR, CIR (*Color Infrared*) y NDVI. El trabajo explicita que la calidad del resultado depende de una cadena de preprocesamiento bien definida, que incluye alineación entre bandas, calibración radiométrica global y construcción consistente del ortomosaico. En consecuencia, el desempeño del modelo queda estrechamente condicionado por la calidad de las etapas previas.

Desde el punto de vista computacional, el principal problema abordado es el tamaño de los ortomosaicos. Procesarlos directamente con una red neuronal implicaría reescalar la imagen y perder detalle espacial, algo particularmente problemático cuando las malezas ocupan muy pocos píxeles. Para evitarlo, *WeedMap* propone una estrategia de particionado en teselas del mismo tamaño que la entrada de la red, combinada con un esquema de *sliding window*. De este modo, se preserva la resolución original sin exceder las limitaciones de memoria de GPU.

La Figura 3.2 ilustra resultados cualitativos de WeedMap sobre un ortomosaico de prueba. La comparación entre imagen RGB, etiqueta de referencia y predicción permite observar la dificultad de segmentar malezas pequeñas en escenas agrícolas de gran escala, así como la importancia de preservar el detalle espacial mediante estrategias de particionado.

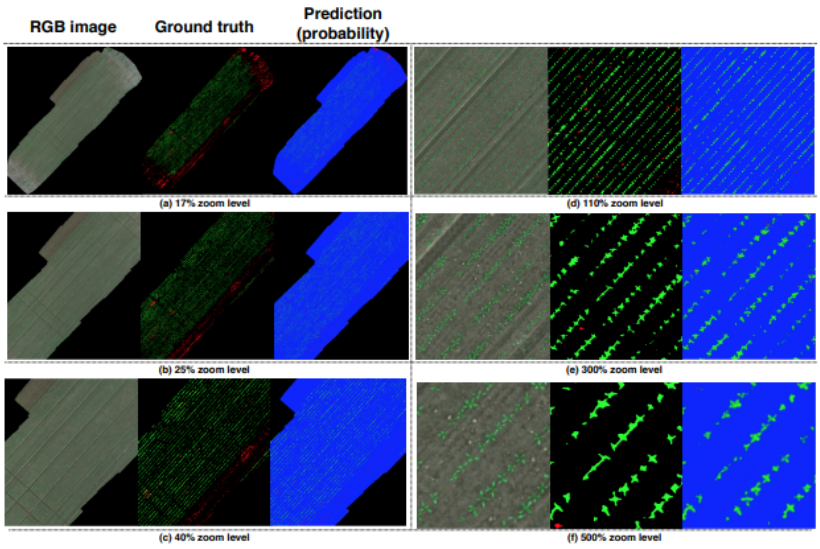


Figura 3.2: Resultados cualitativos de segmentación semántica reportados en WeedMap sobre un ortomosaico del conjunto de prueba. Cada fila muestra un nivel de ampliación distinto, y las columnas presentan la imagen RGB, la etiqueta de referencia y la predicción probabilística del modelo. Tomada de (Sa, Chen, y cols., 2018).

En los experimentos, los autores comparan múltiples configuraciones de canales y muestran que la información multiespectral mejora de forma clara el desempeño respecto de una línea base RGB. Su mejor modelo, entrenado con nueve canales, alcanzó un AUC (*Area Under the ROC Curve*, que resume la capacidad discriminativa del modelo) de 0,839 para fondo, 0,863 para cultivo y 0,782 para maleza, mientras que la configuración base con SegNet (una arquitectura encoder-decoder para segmentación semántica) y solo RGB obtuvo 0,607, 0,681 y 0,576, respectivamente. Además, el trabajo señala que el canal NDVI contribuye de forma importante a separar vegetación y fondo, aportando evidencia empírica sobre la utilidad de variables derivadas en este tipo de problema.

No obstante, el propio trabajo también pone de manifiesto limitaciones relevantes. Los autores señalan que la segmentación de malezas continúa siendo el aspecto más difícil del problema, principalmente por el tamaño reducido de las plantas y por la variabilidad natural de forma, tamaño y apariencia. Asimismo, destacan la dificultad de construir modelos espaciotemporales robustos cuando cambia el estado fenológico de las plantas o cuando se pasa de un lote a otro. En conjunto, *WeedMap* constituye un antecedente sólido por integrar adquisición, calibración, alineación, particionado y modelado dentro de un mismo pipeline, aunque su formulación permanece centrada en remolacha azucarera, una única clase de maleza y ortomosaicos ya construidos.

3.3. WeedsGalore: dataset multiespectral y multitemporal para segmentación de cultivo y malezas

Uno de los antecedentes más relevantes es *WeedsGalore*, presentado por Celikkan et al. (Celikkan y cols., 2025). Este trabajo aborda de forma explícita la construcción de un dataset UAV para segmentación de cultivo y malezas en maíz, con énfasis en el uso de información multiespectral, la representación de distintos estadios fenológicos y la disponibilidad de etiquetas densas con calidad suficiente para entrenar y evaluar modelos modernos. Frente a la escasez de datasets públicos para detección de malezas en condiciones de campo reales, el aporte de *WeedsGalore* radica no solo en la cantidad de datos, sino también en un diseño orientado al estudio de generalización, robustez y valor agregado de las bandas no visibles.

El conjunto de datos fue adquirido en un lote de maíz de aproximadamente 1 840 m^2 ubicado en Potsdam, Alemania, donde las malezas se desarrollaron de forma natural, sin interferencia experimental que alterara las condiciones agronómicas habituales. Las capturas se realizaron en cuatro fechas distintas entre fines de mayo y mediados de junio de 2023, cubriendo diferentes momentos de crecimiento del cultivo y de las malezas. Para ello se utilizó un DJI Phantom P4 Multispectral, equipado con bandas B, G, R, RE y NIR, volando a 5 m de altura y obteniendo un GSD (*Ground Sampling Distance*, la distancia en terreno entre centros de píxeles adyacentes) de 2,5 mm. Esta resolución permitió distinguir individuos pequeños y generar etiquetas finas.

La Figura 3.3 muestra ejemplos de WeedsGalore en las cuatro fechas de adquisición consideradas por los autores, junto con sus máscaras semánticas. La comparación permite observar cómo cambia la cobertura vegetal a lo largo del tiempo y cómo aumenta la complejidad del etiquetado a medida que crecen el cultivo y las malezas. Este aspecto resulta especialmente relevante porque el dataset no captura una única condición del lote, sino distintos estados fenológicos y niveles de cobertura.

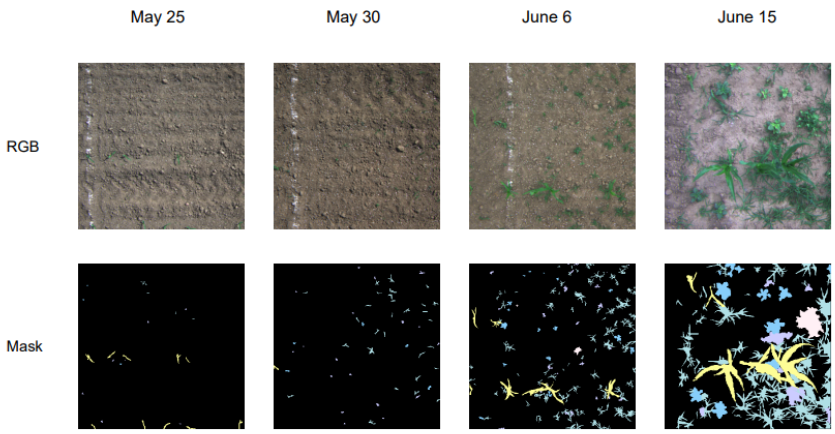


Figura 3.3: Ejemplos de imágenes y máscaras semánticas de WeedsGalore en distintas fechas de adquisición, mostrando la evolución de la cobertura vegetal y la complejidad del etiquetado multitemporal. Tomada de (Celikkan y cols., 2025).

Un aspecto metodológico especialmente valioso es que los autores deciden no anotar ortomosaicos finales, sino imágenes crudas alineadas por banda. Según el trabajo, la ortorrectificación y la alineación global pueden introducir artefactos que afectan la calidad de las máscaras finas, por lo que optan por alinear las bandas a nivel de captura y luego recortar la región central de 600×600 píxeles para anotar. A su vez, para asegurar diversidad espacial, el lote fue dividido en parches y se muestrearon 48 ubicaciones georreferenciadas, combinando escenas de distintas fechas.

En total, el dataset contiene 156 imágenes etiquetadas, divididas espacialmente en entrenamiento, validación y prueba con proporción 70:15:15. Las etiquetas fueron realizadas por dos anotadores y revisadas por tres expertos, lo que apunta a controlar la calidad y consistencia del etiquetado. El conjunto incluye segmentación semántica y de instancias, con clases para maíz y cuatro categorías de maleza: *amaranth*, *barnyard grass*, *quickweed* y *weed other*. En total se reportan 10 031 instancias de maleza y 2 169 de cultivo, lo que da una densidad promedio muy alta de plantas por imagen y convierte al dataset en un escenario considerablemente más exigente que otros conjuntos públicos comparables.

Sobre este conjunto, los autores evalúan modelos de segmentación semántica y de instancias con entradas RGB y multiespectrales. En la formulación de tres clases, DeepLabv3+ (una arquitectura de segmentación semántica basada en convoluciones dilatadas) pasa de 79,33 de mIoU (*mean Intersection over Union*, el promedio de IoU sobre todas las clases) con RGB a 82,90 con entrada multiespectral; en la formulación de seis clases, el mismo modelo mejora de 50,81 a 55,52 de mIoU. También se reportan mejoras con MaskFormer y en seg-

mentación de instancias, aunque más moderadas. Además, el trabajo incorpora variantes probabilísticas para cuantificar incertidumbre y analiza el comportamiento fuera de distribución, mostrando que el dataset también resulta útil para estudiar calibración y confiabilidad de los modelos.

En conjunto, *WeedsGalore* se presenta como una referencia sólida tanto por la calidad de su diseño experimental como por la evidencia cuantitativa que aporta sobre el uso de bandas multiespectrales. Sin embargo, el trabajo se desarrolla sobre maíz, especies de maleza específicas y condiciones agronómicas correspondientes a un contexto distinto.

3.4. MSU-Net para reconocimiento de malezas en imágenes UAV multiespectrales

Como antecedente orientado principalmente al modelado, resulta relevante el trabajo de Wu, Wu y Miao (J. Wu y cols., 2025), que estudia la identificación de malezas en cultivo de trigo sarraceno mediante imágenes multiespectrales obtenidas con UAV. A diferencia de trabajos centrados en la construcción del dataset o en el mapeo a escala de lote, este artículo se concentra en adaptar una arquitectura de segmentación para recibir entradas multibanda de forma flexible y en analizar qué bandas o combinaciones espectrales resultan más efectivas para separar maleza y cultivo.

El estudio se realizó sobre un campo experimental de trigo sarraceno en Shanxi, China, y utilizó un DJI Phantom 4 Multispectral con cinco bandas: R, G, B, NIR y RE. Las capturas se realizaron el 5 de agosto de 2023 a 30 m de altura, con un GSD de 0,82 cm/píxel, superposición longitudinal del 80 % y lateral del 70 %. Los autores emplean *TimeSync* para sincronización temporal y *PIX4Dmapper* para generar ortofotos multiespectrales. En cuanto al etiquetado, utilizan *Labelme* sobre las imágenes RGB del mismo lote, ya que la herramienta no admite directamente la lectura de datos multiespectrales. Debido a la altura de vuelo y a la dificultad para distinguir tipos de maleza individuales desde una vista cenital, todas las malezas se agrupan en una sola clase.

La Figura 3.4 muestra ejemplos de imágenes RGB utilizadas por los autores y sus correspondientes máscaras de etiquetado. La figura permite observar la formulación semántica del problema: las regiones correspondientes a maleza se representan como una única clase, sin distinguir especies ni instancias individuales. Este criterio resulta coherente con la escala de adquisición y con el objetivo del trabajo, centrado en delimitar áreas infestadas más que en identificar plantas particulares.

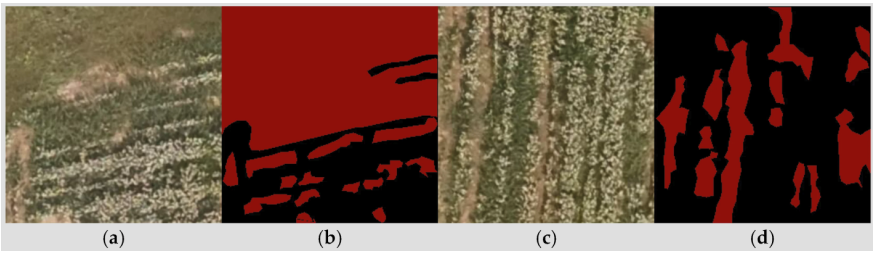


Figura 3.4: Ejemplos de imágenes RGB y sus correspondientes etiquetas para reconocimiento de malezas en trigo sarraceno mediante MSU-Net. Las imágenes muestran dos escenas representativas y sus máscaras semánticas asociadas. Tomada de (J. Wu y cols., 2025).

En la etapa de preprocesamiento, el trabajo normaliza cada banda de forma independiente y adapta operaciones de aumento de datos para trabajar directamente sobre archivos TIFF multicanal. Los autores combinan las cinco bandas en un tensor multiespectral y aplican transformaciones como volteos, rotaciones, desenfoque gaussiano, ruido, realce y transformaciones afines, evitando cambios de brillo y contraste por las particularidades radiométricas de los datos. Sobre esta base proponen *MSU-Net*, una variante de U-Net cuyo principal aporte es un módulo de entrada capaz de aceptar un número variable de canales y el reemplazo de la activación ReLU por Swish para mejorar la estabilidad del entrenamiento frente a entradas multiespectrales.

Experimentalmente, el trabajo compara cinco configuraciones de banda única y nueve combinaciones multibanda. En el caso de banda única, la banda azul obtiene los mejores resultados, con mPA (*mean Pixel Accuracy*, el promedio de exactitud por clase a nivel de píxel) de 0,75, mIoU de 0,61, Dice de 0,87 y F1 de 0,80. En el caso multibanda, la combinación R+G+B+NIR resulta la más efectiva, con mPA de 0,76, mIoU de 0,65, Dice de 0,85 y F1 de 0,78. Además, el modelo propuesto se compara con U-Net, DenseASPP, PSPNet y DeepLabv3, alcanzando un equilibrio favorable entre precisión y consumo de recursos. Los autores también muestran ejemplos de mapas de maleza construidos sobre escenas grandes, en situaciones de concentración de maleza, bordes de parcela y zonas regulares del campo, lo que sugiere un posible uso para planificación de aplicaciones selectivas.

Este antecedente muestra que el valor de cada banda no debe asumirse a priori, sino medirse empíricamente en función del cultivo, el sensor, la altura de vuelo y la formulación del problema. Sin embargo, también presenta limitaciones para una transferencia directa a otros contextos: la tarea se formula con una única clase de maleza, no diferencia especies, y los propios autores reconocen que la menor resolución espacial del sensor multiespectral restringe parte de su aporte, proponiendo como trabajo futuro la fusión de información RGB y multiespectral. Además, el estudio se desarrolla sobre trigo sarraceno y malezas presentes en un contexto agronómico específico de China.

3.5. Detección y conteo de flores de manzano

Otro antecedente metodológico relevante es el proyecto *Detección y conteo de flores de manzano* (Tessore, 2025). En dicho trabajo se aborda un problema frecuente en visión por computadora aplicada al agro: el procesamiento de imágenes de alta resolución cuando los objetos de interés ocupan una fracción pequeña de la escena.

Tessore parte de imágenes de alta resolución, por ejemplo de 1242×2208 y 1920×1080, y utiliza YOLOv8n como modelo base con entrada de 640×640. El problema observado es que, al reescalar imágenes grandes a ese tamaño, muchos objetos pequeños pierden detalle y se vuelven difíciles de detectar. Para mitigar este efecto, el trabajo propone una estrategia en dos etapas: *Slicing Aided Fine-Tuning* (SF) en entrenamiento y *Slicing Aided Hyper Inference* (SAHI) en inferencia.

La Figura 3.5 resume el flujo metodológico seguido en este trabajo. El esquema permite visualizar la comparación entre un modelo base entrenado sobre el dataset original y una segunda configuración que incorpora recortes generados mediante SF, evaluada posteriormente con y sin SAHI. Esta comparación resulta relevante para este proyecto porque evidencia cómo las estrategias de recorte pueden integrarse al entrenamiento y a la inferencia sin modificar la arquitectura del detector.

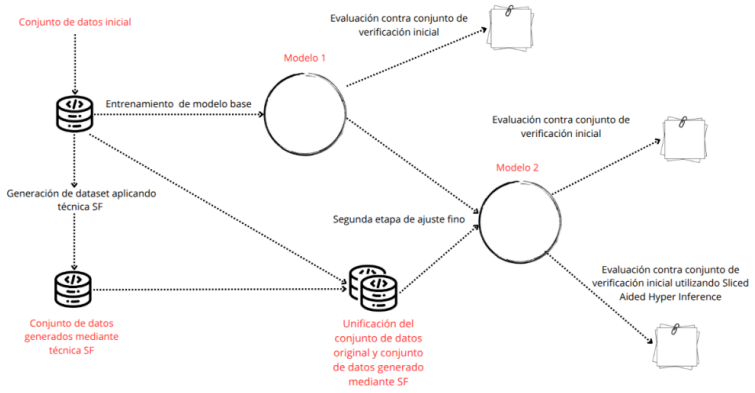


Figura 3.5: Flujo de trabajo propuesto para evaluar el impacto de *Slicing Aided Fine-Tuning* y *Slicing Aided Hyper Inference* en la detección de objetos pequeños en imágenes de alta resolución. Tomada de (Tessore, 2025).

En entrenamiento, SF genera parches de 640×640 con solapamiento del 20 % ($O = 0{,}2$), aumentando la proporción de objetos pequeños visibles por muestra y permitiendo un ajuste fino más efectivo del detector. En inferencia, SAHI divide cada imagen en parches, ejecuta detección por parche, reproyecta las

predicciones al sistema de coordenadas original y fusiona los resultados mediante posprocesamiento basado en IoU y NMS. Los resultados reportados muestran mejoras consistentes en el desempeño: por ejemplo, un aumento de mAP@50 de 0,547 a 0,621 en un ensayo y de 0,501 a 0,602 en otro, junto con mejoras adicionales en la métrica *Loc*.

Como contrapartida, el método incrementa el costo computacional, particularmente en inferencia, debido al procesamiento de múltiples parches por imagen. Aun así, el trabajo constituye una referencia metodológica valiosa para tareas en las que la preservación del detalle espacial resulta crítica para detectar objetos pequeños.

3.6. Síntesis de antecedentes

Los antecedentes revisados muestran que la detección de malezas en condiciones de campo continúa siendo un problema desafiante, especialmente cuando las plantas objetivo son pequeñas, presentan alta variabilidad morfológica o aparecen sobre fondos heterogéneos.

En este sentido, la evaluación de herramientas comerciales como *PIX4Dfields* evidencia que las soluciones disponibles pueden resultar útiles para mapeos generales a escala de lote, aunque presentan limitaciones importantes cuando se pretende detectar individuos puntuales con alta precisión y exhaustividad de forma simultánea. A su vez, los trabajos académicos coinciden en que la información multiespectral aporta ventajas frente a configuraciones basadas únicamente en RGB. Tanto *WeedMap* como *WeedsGalore* y *MSU-Net* reportan mejoras al incorporar bandas no visibles e índices de vegetación, lo que respalda el interés por estudiar combinaciones espectrales más allá del espectro visible. Asimismo, estos antecedentes destacan que el desempeño final no depende solo del modelo, sino también de la calidad del pipeline de adquisición y preprocesamiento, incluyendo calibración radiométrica, alineación entre bandas, construcción consistente de las entradas y preservación del detalle espacial.

Respecto de la preservación del detalle espacial, el antecedente sobre detección y conteo de flores de manzano ilustra el problema que plantean las imágenes de alta resolución cuando los objetos de interés ocupan pocos píxeles. En el presente trabajo, las imágenes alineadas y calibradas poseen una resolución de 2445×1890 píxeles, equivalente a 4,62 megapíxeles, lo que representa aproximadamente 11,3 veces la cantidad de píxeles de una entrada estándar de 640×640 como la utilizada por defecto en YOLOv8 y aproximadamente 92 veces la cantidad de píxeles de una entrada de ResNet18 y EfficientNet-b0, 224×224. En este contexto, un reescalado directo puede implicar una pérdida sustantiva de detalle espacial, por lo que las estrategias basadas en recortes o *slicing* adquieren relevancia metodológica, en tanto permiten preservar información local durante el entrenamiento y la inferencia, aunque con un mayor costo computacional.

A partir de la revisión, es posible identificar que, si bien trabajos como *WeedMap* y *WeedsGalore* combinan adquisición multiespectral, documentación del pipeline y comparación de configuraciones espectrales, ninguno de ellos abor-

da la detección de *Conyza* spp. en sistemas productivos nacionales, donde las condiciones agronómicas, las especies presentes y la disponibilidad de datos difieren sustancialmente de los contextos en los que fueron desarrollados. Esta brecha justifica la construcción y caracterización del dataset presentado en este trabajo, así como su evaluación mediante distintas configuraciones de entrada.

Capítulo 4

Parte Central

En este capítulo se presenta la parte central del trabajo, enfocada en el proceso de construcción del dataset multiespectral orientado a la detección de yerba carnicera (*Conyza* spp.) y en las principales decisiones metodológicas adoptadas durante el proyecto. Dado que el problema abordado involucra imágenes adquiridas en condiciones reales de campo, fue necesario resolver desafíos asociados tanto a la selección y preparación de las capturas como a la definición de un esquema de etiquetado consistente y a la posterior evaluación experimental del conjunto construido.

Con este fin, en primer lugar se describe la obtención y selección de las capturas, junto con las características principales de las imágenes y del dispositivo utilizado para su adquisición. Luego se desarrolla el preprocesamiento aplicado, incluyendo las etapas necesarias para asegurar un uso adecuado de las capturas en las tareas posteriores. A continuación, se presenta el proceso de etiquetado y la caracterización estadística de la versión final del dataset. Finalmente, se expone la planificación de la experimentación sobre el conjunto construido, incluyendo la configuración de los ensayos y la prueba con varios modelos para analizar su utilidad en tareas de detección y segmentación de malezas.

4.1. Obtención y selección de capturas

Las imágenes utilizadas para crear el dataset fueron seleccionadas de un conjunto de capturas realizadas por el grupo MINA [1] de la Facultad de Ingeniería. El conjunto de imágenes seleccionado corresponde a capturas de un cultivo de vid cercano a la localidad de Empalme Olmos, en el departamento de Canelones, tomadas el 29 de septiembre de 2025.

El dispositivo utilizado para realizar las capturas fue la cámara multiespectral MicaSense RedEdge-P, mostrada en la Figura 4.1. La cámara posee 6 sensores, cada uno destinado a capturar una banda del espectro electromagnético (MicaSense, 2021).

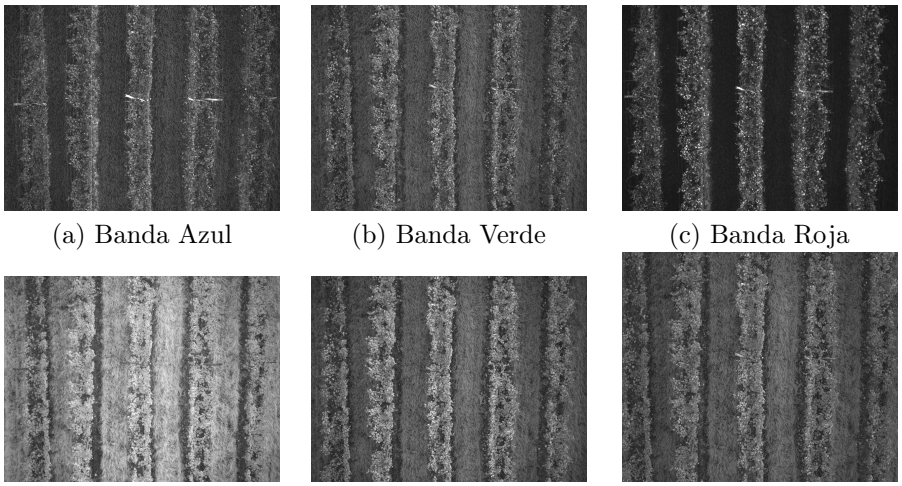
[1] https://www.fing.edu.uy/inco/grupos/mina/

Figura 4.1: Cámara multiespectral MicaSense RedEdge-P utilizada para la adquisición de imágenes. Tomada de (Blue Skies Drone Shop, s.f.).

Las bandas B, G, R, RE y NIR son capturadas por los sensores con una resolución espacial de 1,58 megapíxeles (1456 × 1088), mientras que la banda P es capturada con una resolución espacial de 5,1 megapíxeles (2464 × 2056). Al realizarse una captura, se obtienen 6 archivos, cada uno representando una de las bandas (MicaSense, 2021). En la Tabla 4.1 se muestran los rangos de longitud de onda del espectro electromagnético representados por cada banda y en la Figura 4.2 se observa un ejemplo de las imágenes que conforman una captura.

Banda	**Longitud de onda central**	**Ancho de banda**
Banda Azul (B)	475 nm	32 nm
Banda Verde (G)	560 nm	27 nm
Banda Roja (R)	668 nm	14 nm
Banda Red-edge (RE)	717 nm	12 nm
Banda Infrarrojo cercano (NIR)	842 nm	57 nm
Banda Pancromática (P)	634,5 nm	463 nm

Tabla 4.1: Bandas espectrales capturadas por la cámara MicaSense RedEdge-P, con longitud de onda central y ancho de banda.



(a) Banda Azul (b) Banda Verde (c) Banda Roja

(d) Banda RE (e) Banda NIR (f) Banda Pancromática

Figura 4.2: Conjunto de seis imágenes de la captura multiespectral 0174, una por cada banda adquirida por la cámara.

En la obtención de las capturas seleccionadas, la cámara multiespectral MicaSense RedEdge-P se montó sobre un dron DJI Matrice 350 RTK. En la Figura 4.3 se muestra el dron utilizado durante la campaña de adquisición. Durante la captura, el dron sobrevoló el cultivo a $12\,m$ de altura. La altura fue programada antes del inicio del vuelo y se mantuvo aproximadamente constante, salvo pequeñas variaciones debidas al relieve del terreno.

Figura 4.3: Dron DJI Matrice 350 RTK utilizado para la adquisición de imágenes. Tomada de (Rippercorp, s.f.).

Dado que, al momento de la captura, la cámara se encontraba perpendicular al suelo, se puede calcular la resolución espacial al nivel del suelo (Ground

Sampling Distance, GSD), que indica la distancia en el terreno que representa cada píxel de la imagen, de manera simplificada (ver Sección A.1). Para realizar el cálculo se consideran dos aspectos técnicos de la cámara, la longitud focal (f) y el tamaño de píxel en sensor (p).

$$GSD = \frac{h \times p}{f} \tag{4.1}$$

Para los sensores de las bandas B, G, R, RE y NIR la longitud focal es $5{,}5\,mm$ y el tamaño de píxel en sensor es $3{,}45\,\mu m$. Para el sensor de la banda P los valores son $10{,}3\,mm$ y $3{,}45\,\mu m$ respectivamente (MicaSense, 2021). Entonces los valores de GSD son los siguientes:

$$GSD_i = \frac{12\,m \times 3{,}45\,\mu m}{5{,}5\,mm} = 0{,}75\,\text{cm/px}, \quad \forall i \in \{B, G, R, RE, NIR\} \tag{4.2}$$

$$GSD_P = \frac{12\,m \times 3{,}45\,\mu m}{10{,}3\,mm} = 0{,}40\,\text{cm/px} \tag{4.3}$$

La importancia del GSD reside en que, bajo las mismas condiciones, cuanto menor es su valor, aumenta la posibilidad de detectar malezas pues generalmente son de tamaño pequeño respecto a la captura y la discriminación entre maleza y otras plantas se basa parcialmente en sus características morfológicas, las cuales se aprecian con mayor detalle a menor GSD.

Los GSD calculados en las Ecuaciones 4.2 y 4.3 se encuentran en el mismo orden de magnitud que los antecedentes referenciados en la Sección 3.3 (Celikkan y cols., 2025) y en la Sección 3.4 (J. Wu y cols., 2025). Estos valores indican que la resolución espacial obtenida es suficiente para distinguir malezas individuales en las imágenes, lo que resulta coherente con los requerimientos del problema abordado.

Las capturas de la cámara fueron programadas para asegurar una superposición del 75 % entre capturas contiguas. Esto permite construir un mapa del cultivo a través de la ortorrectificación y mosaicado, procesos por los cuales se crea una unión de las imágenes. El mapa resultante se denomina ortomosaico, el cual permite obtener una visión general de la zona recorrida en una imagen. En este proyecto se decidió no crear un ortomosaico puesto que no se consideró como parte del alcance la localización de las malezas en el cultivo donde se obtuvieron las capturas.

Además, al inicio del vuelo se tomó una captura del panel de calibración. Dicho panel otorga información que es utilizada en el preprocesamiento de las imágenes. El proceso se explica en la Sección 4.2.

En cuanto a la selección de capturas, el conjunto seleccionado originalmente contenía 947 capturas. No obstante, en primer lugar, se descartaron capturas con el fin de conseguir un conjunto donde se tuviera certeza de que el dron sobrevolara el cultivo a $12\,m$ de altura.

Dentro del conjunto de capturas, como se mencionó, se hallaba una captura del panel de calibración de la cámara RedEdge-P. A pesar de ser utilizada durante el preprocesamiento de imágenes, no forma parte del dataset pues no es

una captura ni de los cultivos ni de malezas, y a su vez la captura fue tomada a $1\,m$ de altura.

También se descartaron capturas correspondientes al inicio del vuelo debido a que durante estas capturas el dron aún está en proceso de alcanzar los $12\,m$ de altura. En dichos momentos la altura aún no es coherente con el resto de las imágenes y el dron, mientras asciende, no recorre zonas del cultivo. Por esta razón se descartaron 41 capturas.

En total, 905 capturas fueron seleccionadas como primer paso para la formación del dataset. No obstante, más adelante en la Sección 4.3 se describe otro paso de refinamiento de la selección de imágenes.

4.2. Preprocesamiento de imágenes

Una vez obtenido el conjunto de imágenes a utilizar, se avanzó con el preprocesamiento de las capturas. Como se mencionó en la Sección 4.1, por cada captura de la cámara se obtienen 6 imágenes, cada una correspondiente a una banda del espectro electromagnético. En dichas imágenes, los valores de los píxeles tienen una profundidad de 12 bits, lo que implica un rango de valores crudos entre 0 y $2^{12} - 1 = 4095$. Este rango es considerablemente mayor que el de una imagen convencional de 8 bits y permite representar con mayor resolución radiométrica las diferencias de intensidad capturadas por cada sensor, aspecto que resulta relevante durante la calibración radiométrica descrita a continuación.

Las cámaras multiespectrales no registran directamente la reflectancia de las superficies capturadas. En cambio, registran una señal numérica afectada por una combinación de factores que incluyen la sensibilidad del sensor, el tiempo de exposición, la ganancia electrónica, el viñeteado óptico, la iluminación ambiente y la geometría de la escena (Guo y cols., 2019).

Por este motivo, el primer paso del preprocesamiento es la calibración radiométrica, cuyo objetivo es transformar los valores crudos registrados por los sensores en valores comparables entre capturas y con significado físico.

El siguiente paso es la alineación de bandas. Este proceso es necesario debido a que el resultado de cada captura consiste de 6 imágenes, cada una tomada por un sensor distinto, lo cual genera un desalineamiento entre las imágenes capturadas (MicaSense, 2024c).

El último paso es la construcción del stack de captura. Una vez alineadas las imágenes, se construye una nueva imagen multicanal con las bandas necesarias en cada caso.

En la Figura 4.4 se observa el diagrama de flujo del preprocesamiento de imágenes.

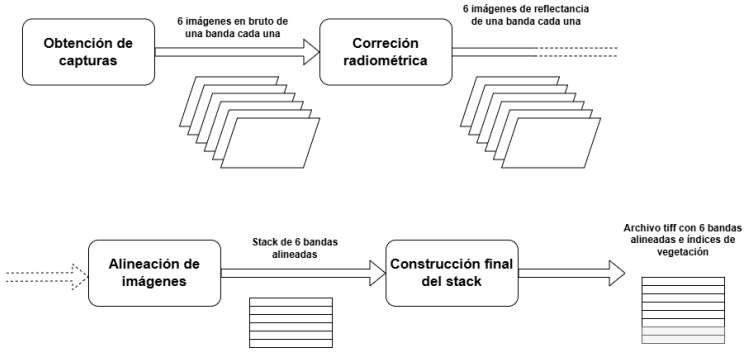


Figura 4.4: Diagrama de flujo del preprocesamiento de imágenes.

4.2.1. Corrección radiométrica

La corrección radiométrica aplicada a las imágenes se definió con base en las especificaciones y manuales de la cámara utilizada, MicaSense RedEdge-P (MicaSense, 2018, 2016, 2024a), así como en el código presente en el repositorio de procesamiento de imágenes de MicaSense (MicaSense, 2024b).

El proceso de corrección radiométrica se compone de tres etapas aplicadas a cada imagen de cada captura. En primer lugar, se convierte la imagen cruda en una imagen de radiancia, corrigiendo los efectos del nivel de negro, la ganancia, el tiempo de exposición y el viñeteado del sensor. En segundo lugar, se transforma la imagen de radiancia en una imagen de reflectancia, para lo cual es necesario conocer la irradiancia incidente en la escena; dicha irradiancia se estima a partir de una captura del panel de calibración realizada durante el vuelo. Por último, se aplica una corrección de distorsión de lente para eliminar las aberraciones ópticas introducidas por la óptica de la cámara (MicaSense, 2018, 2016). A continuación se describen en detalle las magnitudes involucradas y cada una de las etapas mencionadas.

Para comprender la calibración radiométrica de los sensores es fundamental comprender tres magnitudes físicas: radiancia, reflectancia e irradiancia. Dada una superficie en la cual incide radiación electromagnética, la irradiancia (E) es la densidad del flujo de energía incidente sobre una superficie por unidad de área, por lo que sus unidades son Wm^{-2}. En condiciones reales, la irradiancia incluye tanto la componente directa proveniente del sol como componentes difusas debidas a la dispersión atmosférica (Schaepman-Strub y cols., 2006; Liang, 2004).

Por otro lado, la radiancia (L) es la cantidad de energía electromagnética que viaja en una dirección específica, definida como la potencia por unidad de área proyectada y por unidad de ángulo sólido, y se expresa en $Wm^{-2}sr^{-1}nm^{-1}$. Esta es la magnitud físicamente medida por los sensores remotos, e incluye contribuciones tanto de la radiación reflejada por la superficie como de efectos atmosféricos (Schaepman-Strub y cols., 2006; Liang, 2004).

Por último, la reflectancia (ρ) representa la relación entre la radiación reflejada por una superficie y la radiación incidente sobre ella. Si bien puede interpretarse como una propiedad del material, en rigor depende de la geometría de iluminación y observación, siendo descrita formalmente mediante la función de distribución bidireccional de reflectancia (BRDF). En aplicaciones prácticas de teledetección, lo que se estima suele ser un factor de reflectancia bidireccional (BRF) bajo condiciones específicas de adquisición (Schaepman-Strub y cols., 2006; Liang, 2004). En lo que sigue, se utiliza el término "reflectancia" como simplificación práctica de esta estimación.

El resultado final de la corrección radiométrica es obtener imágenes que representen la reflectancia de los objetos visibles en cada una de ellas para la banda correspondiente. La reflectancia puede acotarse a una parte de la energía del espectro electromagnético en un cierto rango de longitudes de onda, como lo son las seis bandas capturadas por la cámara.

El primer paso para lograr lo descrito es transformar los valores de la imagen a radiancia (MicaSense, 2018). Al igual que la reflectancia, la radiancia puede acotarse a rangos de longitud de onda del espectro electromagnético.

La transformación a radiancia se realiza con operaciones píxel a píxel en cada una de las imágenes. Para un píxel en la posición (x, y) de una imagen se obtiene la radiancia de la siguiente manera (MicaSense, 2018):

$$L(x,y) = V(x,y) * \frac{a_1}{g} * \frac{p(x,y) - p_{BL}}{t_e + a_2 y - a_3 t_e y} \tag{4.4}$$

Siendo:

- $p(x, y)$: valor del píxel (x, y) normalizado sobre el rango de valores posibles para el píxel. Las imágenes son de 12 bits de profundidad, por lo que el valor original del píxel se divide entre 2^{12}.
- p_{BL}: valor normalizado del nivel de negro. Es un valor de offset propio de los sensores de la cámara que se asigna a los píxeles incluso en ausencia total de luz.
- a_1, a_2, a_3: coeficientes de calibración definidos de fábrica.
- g: ganancia del sensor. Es el factor de amplificación aplicado a la imagen.
- t_e: tiempo de exposición de la cámara. Es el tiempo durante el cual los sensores capturan luz incidente para formar la imagen.
- $V(x, y)$: modelo de viñeteado.

Los valores de g, t_e y p_{BL} son específicos de cada imagen y se obtienen de los metadatos EXIF embebidos en cada archivo, mientras que los coeficientes de calibración a_1, a_2, a_3 y los coeficientes de viñeteado se almacenan en campos XMP definidos por MicaSense (MicaSense, 2018).

El modelo de viñeteado refiere a la corrección necesaria para contrarrestar la disminución de la sensibilidad lumínica de los píxeles a medida que se alejan del

centro del sensor. El modelo puede calcularse de la siguiente manera (MicaSense, 2018):

$$r(x,y) = \sqrt{(x-c_x)^2 + (y-c_y)^2} \tag{4.5}$$

$$k(r) = 1 + \sum_{i=0}^{5} k_i * r^{i+1} \tag{4.6}$$

$$V(x,y) = \frac{1}{k(r(x,y))} \tag{4.7}$$

Siendo:

- (c_x, c_y): posición del centro de la imagen.
- $k_i \quad \forall i \in \{0,1,2,3,4,5\}$: coeficientes polinómicos de viñeteado definidos de fábrica.

Una vez obtenidas las imágenes que representan la radiancia, el siguiente paso es realizar la transformación a reflectancia. Para que esto sea posible es necesario que durante el vuelo se hayan realizado capturas del panel de calibración. Se debe realizar al menos una captura del panel, aunque es posible hacer más de una, por ejemplo al comienzo y al final del vuelo (MicaSense, 2016, 2024a). Dichas capturas deben realizarse a aproximadamente $1\,m$ de altura (MicaSense, 2024a). En el caso del conjunto de imágenes seleccionado se tomó una captura al principio del vuelo.

El panel de calibración es fabricado de tal manera que posee una superficie cuyos valores de reflectancia están definidos de antemano para cada una de las bandas que captura la cámara MicaSense RedEdge-P. Dicha superficie, a su vez, puede aproximarse como una superficie Lambertiana. Para poder utilizar el panel en el cálculo de la irradiancia es necesario identificar la región correspondiente a su superficie en cada una de las imágenes de la captura. Esto se realiza de forma automática: la cámara puede detectar el panel durante la adquisición y almacenar la región en la metadata de la imagen; de lo contrario, se detecta el código QR impreso en el panel y, a partir de las dimensiones físicas conocidas del mismo, se estima la ubicación de la superficie de calibración mediante una transformación de perspectiva (MicaSense, 2024b).

Una superficie Lambertiana es una superficie ideal que refleja energía uniformemente en todas direcciones, es decir, su radiancia es independiente del ángulo de observación. Bajo esta hipótesis, se cumple una relación particular entre la reflectancia (ρ), la irradiancia (E) y la radiancia (L) (Schaepman-Strub y cols., 2006):

$$L = \frac{\rho E}{\pi} \tag{4.8}$$

Entonces, para cada imagen de la captura del panel, sobre la superficie del mismo, se cumple la siguiente relación:

$$E = \frac{L\pi}{\rho} \tag{4.9}$$

Por ende, al conocer la reflectancia en cada banda capturada de antemano, y calculando la radiancia en la superficie del panel de calibración, se logra hallar la irradiancia. Para calcular la radiancia de la superficie se lleva a cabo el mismo proceso utilizado para el resto de las imágenes, descrito por las Ecuaciones 4.4, 4.5, 4.6 y 4.7, y se calcula el promedio dentro de la superficie (MicaSense, 2016).

$$E_i = \frac{\pi}{\rho_i |S_i|} \sum_{(x,y)\in S_i} L_i(x,y) \quad \forall i \in \{B, G, R, RE, NIR, P\} \tag{4.10}$$

Siendo:

- S_i: superficie Lambertiana del panel de calibración en la imagen capturada para la banda $i \in \{B, G, R, RE, NIR, P\}$.

Una vez obtenida la irradiancia sobre el panel para cada banda, es necesario establecer dos supuestos para definir la manera en la que las imágenes de radiancia se transforman en imágenes de reflectancia. En primer lugar, se asume que para cada banda las imágenes capturadas tienen una misma irradiancia, es decir, que la energía incidente sobre las superficies es la misma para un determinado rango de longitudes de onda. Se considera que esta suposición es válida debido a que durante el vuelo las condiciones ambientales no variaron considerablemente. En caso de haber tenido más capturas del panel en distintos momentos del vuelo, se podrían haber realizado aproximaciones con más puntos de referencia. En la Figura 4.5 se muestra la imagen correspondiente a la banda pancromática de la captura del panel de calibración utilizada.

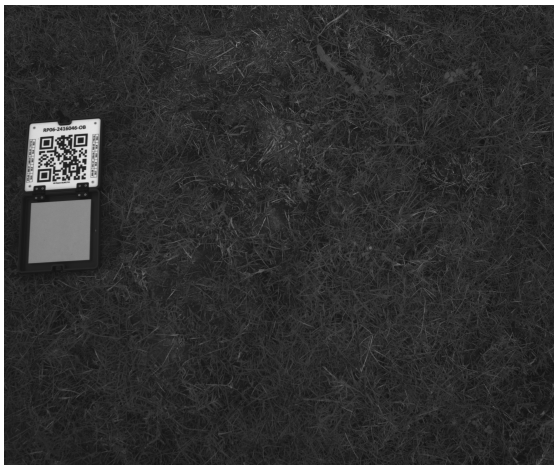


Figura 4.5: Imagen de captura del panel de calibración correspondiente a la banda pancromática.

La segunda suposición es que, para calcular la reflectancia de las superficies capturadas, se aproxima su comportamiento al de superficies Lambertianas. Esta aproximación implica ignorar los efectos de anisotropía de la reflectancia, asociados a la dependencia angular descrita por la BRDF, lo cual constituye una fuente potencial de error en superficies naturales como la vegetación (Schaepman-Strub y cols., 2006; Liang, 2004).

$$\rho_i = \frac{L_i \pi}{E_i} \quad \forall i \in \{B, G, R, RE, NIR, P\} \tag{4.11}$$

Siendo la radiancia (L_i) la representada por los valores de la imagen de radiancia, y la irradiancia (E_i) la hallada para la banda correspondiente a partir de la captura del panel de calibración.

De esta manera se obtuvieron las imágenes de reflectancia para todas las capturas. Una vez hecho esto, se realiza la corrección debida a la distorsión del lente. Para ello se utiliza un modelo de cámara *pinhole* con coeficientes de distorsión radial y tangencial almacenados en la metadata de cada imagen (MicaSense, 2018). A partir de dichos parámetros se calcula un mapa de remapeo que transforma cada píxel de la imagen distorsionada a su posición corregida (MicaSense, 2024b).

4.2.2. Alineación/registro de bandas

Después de la corrección radiométrica el siguiente paso aplicado fue la alineación/registro de bandas. Esto conlleva la realineación de las imágenes ya que, como se indicó, las imágenes que corresponden a una misma captura no están alineadas debido a la diferencia de posición de los sensores en la cámara.

Este proceso puede realizarse con software propietario como Pix4D (Pix4D, 2025). No obstante, se decidió llevarlo a cabo en base al código presente en el repositorio de procesamiento de imágenes de MicaSense (MicaSense, 2024b), con el fin de no depender de software propietario y mantener un flujo de trabajo completamente reproducible con herramientas de código abierto.

Para realizar la alineación, en primer lugar se eligió la banda a tomar como referencia, es decir, aquella banda a la cual el resto de las bandas se alinearían. Se eligió la banda pancromática (P) debido a la posición central del sensor pancromático en la cámara.

La alineación de imágenes se llevó a cabo estimando las transformaciones que se le han de aplicar a cada una para aproximarla a la imagen de referencia. Con el fin de lograr esto se tomó una captura (ID: 0042) para la cual se calcularon las transformaciones necesarias y luego se aplicaron al resto de las capturas del conjunto. Esta decisión se basa en que el dron vuela a una altura programada fijada de antemano, 12 m, aún así, más adelante se llevó a cabo una verificación de la alineación para validar esta decisión.

Las transformaciones en sí mapean los puntos de un plano en las coordenadas de otro plano, por lo que pueden representarse como funciones (H_i) que pueden mapear un punto de una banda en particular (p_i) en un punto en la imagen de referencia (p').

$$p' = H_i(p_i) \tag{4.12}$$

Además, las transformaciones son homografías, es decir, transformaciones proyectivas invertibles entre dos planos que conservan la colinealidad de los puntos. Estas se representan como matrices de 3×3 que actúan sobre coordenadas homogéneas, donde un punto 2D (x, y) se expresa como el vector $(x, y, 1)$ para permitir la representación matricial de la transformación:

$$H_i = \begin{bmatrix} a_i & b_i & c_i \\ d_i & e_i & f_i \\ g_i & j_i & k_i \end{bmatrix} \tag{4.13}$$

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = H_i \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{4.14}$$

Siendo:

- (x', y'): Coordenadas homogéneas del punto p'.
- (x_i, y_i): Coordenadas homogéneas del punto p_i.
- w': Constante de escala de las coordenadas homogéneas.

El primer paso para calcular las transformaciones fue utilizar el algoritmo SIFT (*Scale Invariant Feature Transform*, ver Sección A.2) para encontrar los puntos característicos de la imagen de referencia, la correspondiente a la banda P. Dichos puntos tienen asociados descriptores, los cuales son invariantes a las rotaciones, traslaciones y cambios de escala que se les puedan aplicar a la

imagen (Rey Otero y Delbracio, 2014). Por ende, son utilizados como puntos de referencia para la alineación. Se utilizó la implementación de SIFT disponible en la biblioteca scikit-image (van der Walt y cols., 2014).

Una vez calculados los puntos característicos de la imagen de referencia, se procedió de la misma manera con las imágenes de las bandas restantes. Obtenidos los puntos característicos y descriptores de toda la captura, se llevó a cabo el emparejamiento entre los puntos de la imagen de referencia y cada una de las imágenes restantes utilizando nuevamente la biblioteca scikit-image, empleando la distancia euclidiana para medir la distancia entre descriptores y aplicando Lowe's ratio test (Lowe, 1999), que rechaza un emparejamiento si la razón entre la distancia al mejor y al segundo mejor descriptor supera un umbral de 0,8, para no incluir emparejamientos ambiguos.

Luego, para cada emparejamiento de puntos característicos se utilizó el algoritmo Random Sample Consensus (RANSAC, ver Sección A.3) para eliminar emparejamientos atípicos con muestreo de 8 ejemplos por iteración y un máximo de 5 000 iteraciones. Esto tiene como objetivo eliminar posibles emparejamientos accidentales que no corresponden a las mismas regiones de las zonas capturadas.

Habiendo obtenido emparejamientos de puntos característicos sin ejemplos atípicos se estima la transformación para la alineación de las imágenes. La estimación se realiza aplicando el método Total Least Squares (TLS) para obtener homografías definidas por matrices de dimensiones 3×3 que transformen las coordenadas de un punto característico en las coordenadas de su pareja en la imagen de referencia.

Las homografías obtenidas incorporan el factor de escala entre las resoluciones de la banda pancromática y cada una de las demás. Por ende, la relación entre las coordenadas a las que se aplica la homografía y las resultantes puede expresarse de forma distinta a la presentada en la Ecuación 4.14, donde se trata de imágenes en la misma resolución espacial.

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = H_i \begin{bmatrix} x_i \frac{w}{w_{low}} \\ y_i \frac{h}{h_{low}} \\ 1 \end{bmatrix} \tag{4.15}$$

Siendo:

- h, w: Altura y ancho de la imagen de referencia (banda pancromática).
- h_{low}, w_{low}: Altura y ancho de imagen de la banda para la cual se calcula la homografía.

Una vez obtenidas las 5 homografías, el último paso es aplicar pansharpening y llevar a cabo la alineación de las bandas. A través del pansharpening, usando como base la imagen pancromática, se aumenta la resolución espacial de las demás bandas. Particularmente se utiliza el algoritmo Smoothing Filter-based Intensity Modulation (SFIM, ver Sección A.4), el cual es radiométricamente preciso, pues preserva las características espectrales de las bandas (GIS Geography, 2025). Este paso está incluido en el proceso planteado por MicaSense (MicaSense, 2024b).

El algoritmo SFIM fue aplicado utilizando para cada banda (B_i) su homografía correspondiente (H_i) y la banda pancromática (P). El primer paso es el cálculo de la razón píxel a píxel entre la banda particular y la banda pancromática. Para calcularla se aplica la homografía inversa a la imagen pancromática, y se halla la matriz de razones entre ambas bandas.

$$P_{low}^{i} = P \circ H_i^{-1} \tag{4.16}$$

$$r = \frac{B_i}{P_{low}^{i}} \circ H_i \tag{4.17}$$

Una vez obtenidas las razones, la alineación y el pansharpening se aplican multiplicando píxel a píxel las razones por la banda pancromática.

$$B_i^{aligned}(x, y) = r(x, y)P(x, y) \tag{4.18}$$

Como último paso se recortan los bordes de la imagen, dado que no todas las bandas poseen información válida en los extremos capturados por la banda pancromática. Las imágenes resultantes tienen un tamaño de 2445×1890 px y un GSD de 0,40 cm/px, como se indica en la Ecuación 4.3, debido al uso de la banda pancromática como referencia en la alineación.

Verificación de la alineación

Con el objetivo de validar cuantitativamente que el proceso de registro mejoró la correspondencia espacial entre bandas, se compararon los stacks alineados con sus contrapartes sin alinear mediante un conjunto de métricas complementarias. Se seleccionaron 5 capturas distribuidas a lo largo del vuelo para evaluar la consistencia de la alineación. Las métricas utilizadas se describen a continuación; una definición formal de cada una se incluye en Fundamentos auxiliares (Apéndice A.5).

Correlación cruzada de fase (PCC). Estima el desplazamiento subpíxel residual entre cada banda y la banda pancromática de referencia. Para ello se normaliza cada banda a media cero y varianza unitaria, y se calcula la correlación cruzada normalizada en el dominio de la frecuencia, refinada mediante sobremuestreo matricial con factor 10 (Guizar-Sicairos, Thurman, y Fienup, 2008). Un desplazamiento residual cercano a cero indica buena alineación geométrica.

Coeficiente de correlación cruzada normalizada (NCC). Como métrica de validación mide la similitud lineal entre la intensidad de cada banda y la banda pancromática. Valores cercanos a 1 indican alta correspondencia espacial (Zitová y Flusser, 2003). Dado que las bandas multiespectrales y la pancromática capturan rangos espectrales distintos, no se espera correlación perfecta; sin embargo, un incremento significativo tras el alineamiento evidencia una mejora en la correspondencia geométrica.

Información mutua normalizada (NMI). Cuantifica la dependencia estadística entre pares de bandas sin asumir una relación lineal, lo cual la hace adecuada para comparar bandas con respuestas espectrales diferentes (Studholme, Hill, y Hawkes, 1999). Se calculó para todos los pares posibles entre las 5 bandas multiespectrales (excluyendo la pancromática). Un valor de NMI mayor indica mayor coherencia en la información espacial compartida.

Raíz del Error Cuadrático Medio (RMSE) basado en SIFT. Cuantifica el desalineamiento geométrico residual entre cada banda multiespectral y la banda RedEdge de referencia mediante emparejamiento de puntos característicos. Se detectan puntos SIFT en ambas bandas, se emparejan los descriptores aplicando Lowe's ratio test, y se filtran correspondencias atípicas con RANSAC. El RMSE se calcula como la raíz de la media de las distancias euclídeas al cuadrado entre los pares de puntos inliers. Un valor cercano a 0 px indica alta correspondencia geométrica.

Resultados. La Tabla 4.2 presenta los valores promedio de cada métrica, obtenidos sobre las 5 capturas evaluadas, comparando stacks alineados y sin alinear.

Métrica	**Alineado**	**Sin alinear**	**Mejora**
Desplazamiento medio (px)	1,00	572,15	+571,15
NCC medio	0,67	0,07	+0,60
NMI medio	1,052	1,006	+0,046
SIFT RMSE medio (px)	1,83	96,12	+94,30

Tabla 4.2: Comparación de métricas de alineación entre stacks alineados y sin alinear, promediados sobre 5 capturas. Para el desplazamiento (PCC) y el SIFT RMSE, valores menores indican mejor alineación; para NCC y NMI, valores mayores son preferibles.

Los resultados evidencian que el proceso de alineación redujo el desplazamiento residual medio de 572,15 px a 1,00 px, lo cual confirma la corrección geométrica efectiva de las bandas. La correlación cruzada normalizada aumentó de 0,07 a 0,67, indicando un incremento sustancial en la correspondencia espacial entre bandas y la referencia pancromática. De manera complementaria, la información mutua normalizada presentó un aumento consistente en todos los pares de bandas, y el RMSE basado en SIFT disminuyó tras la alineación, confirmando la reducción del desalineamiento geométrico residual.

Estos resultados validan que las homografías estimadas a partir de una única captura de referencia son aplicables de manera consistente al resto del conjunto, y que el proceso de registro preserva la correspondencia espacial necesaria para reutilizar un mismo conjunto de etiquetas en distintas configuraciones de bandas.

4.2.3. Stack de capturas e índices de vegetación

Una vez que se alinean las imágenes de todas las capturas se puede proceder a crear los stacks de capturas. Los stacks son imágenes en las que sus canales representan una banda del espectro o un índice de vegetación calculado.

Como se describe en la Sección 4.5 se utilizaron distintas combinaciones para experimentar con el dataset. En cada una de ellas se seleccionaron las bandas necesarias. Con el fin de poner a disposición toda la información utilizada se decidió formar stacks de capturas con todas las bandas (B, G, R, RE, NIR, P) y los índices de vegetación utilizados (NDVI).

El paso común a todas las configuraciones de stack es la conversión de la información a una representación de 8 bits por píxel. Al terminar el paso descrito en la Subsección 4.2.2 se obtienen imágenes representadas por matrices donde todos sus valores se encuentran entre 0 y 1; estos valores no pueden representarse de manera fidedigna con enteros sin signo de 8 bits. Por ende, en el caso de las bandas multiespectrales las matrices son multiplicadas por 255 y se redondea al entero más cercano, mientras que en el caso del NDVI se aplica normalización min-max ($x' = \frac{x - x_{min}}{x_{max} - x_{min}}$, donde x_{min} y x_{max} son los valores mínimo y máximo del NDVI en la imagen) y se multiplica el resultado por 255.

4.3. Etiquetado de imágenes

El etiquetado de las imágenes se llevó a cabo por todo el equipo del proyecto utilizando la plataforma Roboflow [2].

Debido a que el etiquetado se realizó manualmente, y con el objetivo de definir las etiquetas a partir de la inspección visual humana, se utilizaron imágenes RGB. Estas imágenes se obtuvieron a partir de las capturas del dataset, seleccionando los primeros tres canales de cada captura. Gracias a la alineación de las bandas en cada captura, proceso descrito en la Subsección 4.2.2, las etiquetas definidas sobre las imágenes RGB resultan útiles para cualquier combinación de bandas.

El etiquetado se realizó mediante segmentación de instancias, definiendo un polígono por cada instancia de maleza en lugar de *bounding boxes* o máscaras de segmentación semántica. Esta decisión se tomó con el fin de obtener un dataset que permita una detección precisa de la maleza. Además, a partir de las etiquetas de segmentación de instancias es posible generar máscaras de segmentación semántica y *bounding boxes* para detección de objetos, por lo que el dataset resultante puede utilizarse en cualquiera de estas tres tareas.

En particular, con el fin de permitir el entrenamiento de modelos de segmentación semántica, se generaron máscaras a partir de las etiquetas de instancia. Para ello, se rasterizaron los polígonos de cada captura sobre una imagen del mismo tamaño que la original, asignando a cada píxel un valor entero correspondiente a la clase de la instancia que lo contiene y reservando el valor cero para el fondo. En los casos puntuales en que dos polígonos de instancias distintas

[2] https://app.roboflow.com/

presentaron solapamiento, se priorizó la instancia de menor área para preservar la representación de los ejemplares más pequeños, que de otro modo podrían quedar parcial o totalmente ocultos por instancias mayores.

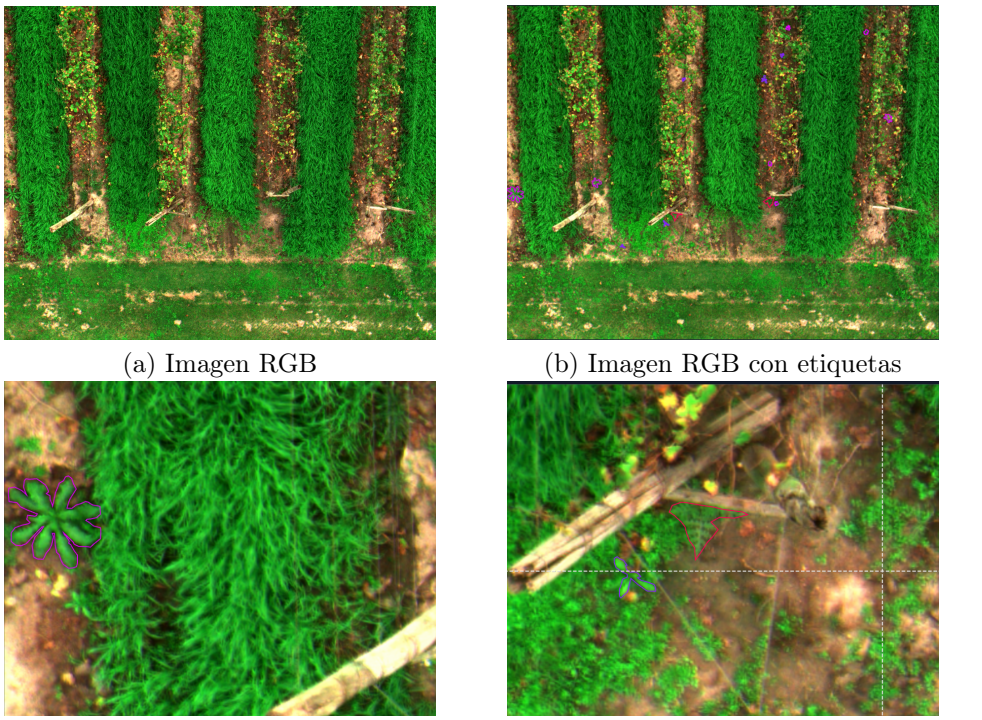
De las 905 capturas seleccionadas, tal como se mencionó en la Sección 4.1, se etiquetaron 500 capturas. El etiquetado manual de instancias es una tarea costosa en tiempo, por lo que al alcanzar las 500 capturas etiquetadas se decidió detener el proceso para ajustarse a las restricciones temporales del proyecto.

El proyecto con las imágenes RGB y sus etiquetas se encuentra disponible en Roboflow [3]. En cuanto al dataset con los *stacks* multiespectrales y sus etiquetas, este se encuentra disponible en el repositorio del proyecto [4].

El procedimiento de etiquetado consistió en delimitar manualmente cada instancia de maleza mediante polígonos ajustados, en la medida de lo posible, a su contorno visible, con el fin de representar fielmente la forma de cada ejemplar y minimizar la inclusión de píxeles de fondo. La Figura 4.6 muestra un ejemplo del resultado, sus ampliaciones permiten apreciar el nivel de detalle alcanzado e incluyen ejemplos de rosetas, plántulas y hojas dentadas.

[3] https://universe.roboflow.com/proyectogradoweeddetection/weed-detection-multispectral-izara

[4] https://github.com/cristiangdev/weed-detection-dataset-pipeline



(a) Imagen RGB

(b) Imagen RGB con etiquetas

(c) Zoom en etiqueta de roseta

(d) Zoom en etiqueta de plántula y dentada

Figura 4.6: Ejemplo del proceso de etiquetado manual de la captura 0367 del dataset.

Cabe señalar que el esquema de tres clases descrito a continuación no fue definido al inicio del proyecto, sino que se fue consolidando a lo largo del propio proceso de etiquetado. En una primera instancia se trabajó con una única clase denominada *weed*, que agrupaba todas las instancias de maleza identificadas; en la práctica, esa versión preliminar cubría únicamente lo que más adelante se consolidaría como la clase *roseta*. Posteriormente se incorporó la clase *plántula* para representar las instancias en estadio temprano de desarrollo, decisión motivada por las aplicaciones previstas en el marco del proyecto del grupo MINA, en el cual se planifica el control de la maleza mediante robots terrestres que también pueden actuar de forma efectiva sobre los ejemplares de menor tamaño. Finalmente, la clase *dentada* se separó del resto al observar un subconjunto de rosetas con hojas de borde aserrado, característica morfológica asociada principalmente a *Conyza sumatrensis*. Cada cambio en el esquema implicó una revisión de las instancias ya etiquetadas para mantener la coherencia interna del dataset, y la versión final se documentó mediante los criterios morfológicos detallados en la Subsección 4.3.1.

4.3.1. Criterios de etiquetado

El objetivo del etiquetado fue señalar en cada imagen la presencia de malezas para permitir el entrenamiento posterior de modelos de aprendizaje automático orientados a su detección, paso necesario para un control oportuno de las malezas. En coherencia con el objetivo aplicado del proyecto del grupo MINA en el cual se enmarca este trabajo, se decidió etiquetar únicamente malezas en estado de plántula y roseta, excluyendo deliberadamente los estados más avanzados de desarrollo. Esta decisión responde a que la eficacia del control químico de *Conyza* spp. disminuye significativamente una vez que las plantas elongan su tallo, mientras que en estadios tempranos (plántula y roseta) los herbicidas habituales mantienen niveles de control satisfactorios (Kaspary y cols., 2024). Por lo tanto, el dataset está orientado a detectar las malezas en la ventana temporal en que la intervención agronómica resulta efectiva.

A su vez, las malezas presentan distinta morfología según su etapa de crecimiento, así como diferentes perfiles en sus hojas. Con el fin de contemplar estas diferencias se definieron tres clases: plántula; roseta; dentada.

Desde un punto de vista botánico, la clase plántula representa las malezas en sus etapas tempranas de crecimiento, cuando sus hojas aún no están plenamente desarrolladas y aún pueden estar en forma de copa. La clase roseta representa la maleza en su etapa intermedia de crecimiento, sus hojas ya se han desarrollado y la planta toma una forma circular desde una perspectiva cenital. Por último, la clase dentada representa a las rosetas con hojas dentadas. Estas corresponden a las malezas cuyas hojas presentan bordes dentados a diferencia de la mayoría de las rosetas; esta característica se observa principalmente en las *Conyza sumatrensis*. En las Figuras 2.1 y 4.7 pueden observarse ejemplos visuales de referencia y ejemplos del dataset para cada una de las tres clases.

Para la correcta identificación de las clases se definieron criterios con base en la manera en que las malezas aparecen en las imágenes. Los siguientes criterios no son todos necesarios, especialmente debido a la oclusión parcial de las malezas, ni tampoco son todos suficientes por su cuenta. No obstante, estos fueron la base sobre la cual se llevó a cabo el etiquetado, buscando siempre que la clase elegida para cada instancia sea aquella para la cual se cumple la mayor cantidad de características.

Plántulas

- Distancia máxima entre extremos de hojas dentro del rango de 15 px a 35 px (aproximadamente 6 cm a 14 cm).
- A lo sumo 6 hojas.
- Posible presencia de cotiledones (las primeras hojas embrionarias de la plántula, generalmente de forma más simple y tamaño menor que las hojas verdaderas) los cuales se presentan como extensiones pequeñas que parten del centro de la planta.
- Hojas de forma predominantemente elíptica.

- Posible disparidad entre el largo de las hojas.

Rosetas

- Distancia máxima entre extremos de hojas dentro del rango de 40 px a 125 px (aproximadamente 16 cm a 50 cm).
- En caso de no estar parcialmente oculta, poseen al menos 4 hojas.
- Generalmente, desde una vista cenital, los extremos de sus hojas forman una geometría circular.
- Hojas lineares.
- Sin presencia de cotiledones.

Dentadas

- Hojas con borde dentado o aserrado.
- Color verde más oscuro respecto de las rosetas.
- Todas las demás características propias de las rosetas.

Cabe destacar que las distancias expresadas en píxeles solo son directamente aplicables a las imágenes completas del dataset, las cuales poseen un GSD nominal de 0,40 cm/px determinado por la altura de vuelo y la configuración de cámara empleadas durante la captura. Cualquier operación posterior que modifique la resolución espacial efectiva alteraría la equivalencia entre píxeles y centímetros, invalidando los rangos en píxeles aquí reportados. Las equivalencias en centímetros, en cambio, reflejan el tamaño físico real de las plantas y son las que sustentan biológicamente la definición de cada clase morfológica. Los rangos en píxeles se incluyen porque constituyen la unidad operativa utilizada durante el proceso de anotación sobre las imágenes.

(a) Plántula (b) Roseta (c) Dentada

Figura 4.7: Ejemplos de malezas en el dataset según las clases definidas: plántula, roseta y dentada.

4.3.2. Control de calidad del etiquetado

Con el fin de asegurar la calidad del dataset se llevaron a cabo dos procedimientos además de la definición de criterios. Uno de ellos fue la validación cruzada del etiquetado, posible gracias a que el proceso se llevó a cabo en dos etapas. En una primera etapa se identificaron todas las posibles instancias de malezas y luego se asignaron las clases correspondientes. Las imágenes analizadas por cada etiquetador en ambas etapas fueron distintas y particularmente en la segunda etapa se procuró corregir los polígonos dibujados durante la primera etapa.

El otro procedimiento fue la resolución de dudas en conjunto, articulada mediante una clase auxiliar específica denominada *dudosa*. El conjunto de imágenes a clasificar se repartía entre los integrantes del equipo y cada uno trabajaba de forma autónoma sobre el subconjunto asignado, rotulando como *dudosa* todas aquellas instancias sobre las que no tenía absoluta certeza, especialmente en la frontera entre *plántula* y *roseta*. Posteriormente, los casos así marcados se revisaban en conjunto y se reasignaban a una de las tres clases finales por voto mayoritario, de modo que la clase *dudosa* no figura como tal en el dataset publicado. Este mecanismo evitó que las clases finales reflejaran la decisión individual del primer anotador en los casos ambiguos.

4.4. Estadísticas del dataset

Al finalizar el proceso de etiquetado se obtuvo la versión final del dataset, compuesta por 500 imágenes. A partir de la información registrada en la plataforma Roboflow se realizó una caracterización descriptiva del conjunto final.

En total se anotaron 6 459 instancias de maleza, distribuidas en tres clases: 3 567 rosetas, 2 599 plántulas y 293 dentadas. La Tabla 4.3 resume esta distribución. Se observa un predominio de la clase roseta, que concentra el 55,23 % de las instancias, seguida por plántula con 40,24 %, mientras que la clase dentada representa el 4,54 % del total. Esta distribución debe tenerse en cuenta al momento de interpretar los resultados experimentales, en particular para la clase menos representada.

Clase	Cantidad	Porcentaje
Roseta	3 567	55,23 %
Plántula	2 599	40,24 %
Dentada	293	4,54 %
Total	6 459	100,00 %

Tabla 4.3: Distribución de instancias etiquetadas por clase en el dataset final.

Además de la distribución por clase, resulta pertinente analizar cómo se distribuyen las etiquetas entre imágenes, dado que esta variable incide directamente en la complejidad de las escenas y, en consecuencia, en la dificultad de

las tareas de detección y segmentación. En promedio, el dataset contiene 12,92 etiquetas por imagen.

La Figura 4.8 muestra la distribución de imágenes por rango de etiquetas. Se observa que 10 imágenes (2,0 %) no presentan etiquetas y 23 imágenes (4,6 %) contienen una única instancia. La mayor concentración se encuentra en el intervalo de 2 a 17 etiquetas, con 374 imágenes (74,8 %). En los intervalos superiores se registran 44 imágenes (8,8 %) con entre 18 y 33 etiquetas, 28 imágenes (5,6 %) con entre 34 y 49, 15 imágenes (3,0 %) con entre 50 y 65, y 6 imágenes (1,2 %) con entre 66 y 81 etiquetas.

En conjunto, estos resultados indican que el dataset está compuesto mayoritariamente por escenas de baja y media densidad de malezas, aunque conserva un subconjunto menor de imágenes con alta concentración de instancias. Esta variabilidad resulta valiosa para la evaluación experimental, ya que permite analizar el desempeño de los modelos tanto en situaciones relativamente simples como en escenarios más exigentes.

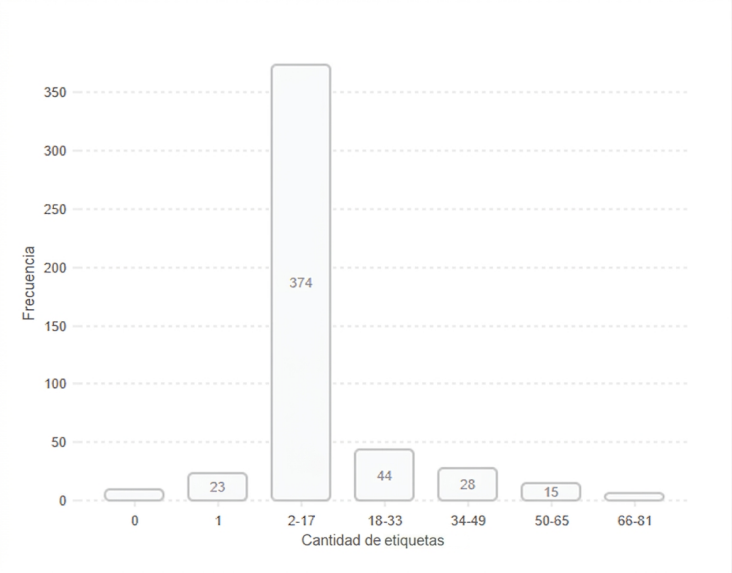


Figura 4.8: Distribución de la cantidad de instancias de maleza por imagen en el dataset.

El área promedio de las etiquetas de rosetas es 3 175,4 px^2, la de las etiquetas de plántula es 673,2 px^2 y la de las dentadas 4 524,9 px^2. El área promedio de las etiquetas en general es 2 229,8 px^2. A pesar de que existen etiquetas donde las instancias están parcialmente ocultas, la diferencia en los promedios entre áreas reafirma la diferencia de tamaño entre las plántulas y las rosetas/dentadas. En la Figura 4.9 se puede observar la distribución del área de cada una de las clases.

Las etiquetas en total ocupan un $0,828\,\%$ del área total de todas las imágenes del dataset, siendo un $0,696\,\%$ ocupado por las etiquetas de rosetas, $0,075\,\%$

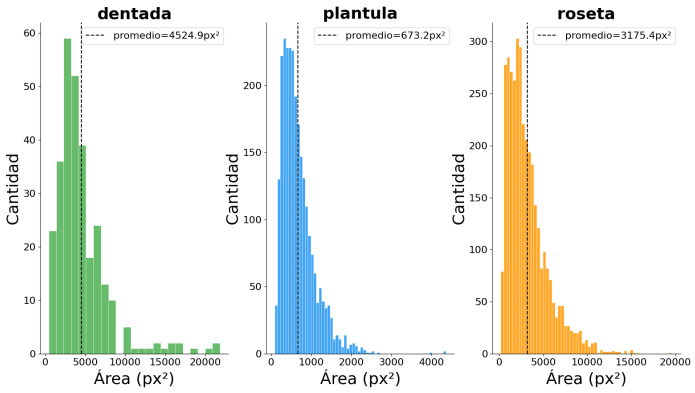


Figura 4.9: Distribución del área de las etiquetas del dataset final por clase.

por las etiquetas de plántulas y $0,057\,\%$ por las etiquetas de dentadas. Estos datos, resumidos en la Tabla 4.4, refuerzan la noción de que las malezas ocupan un espacio reducido en las imágenes.

Clase	Porcentaje
Roseta	0,696 %
Plántula	0,075 %
Dentada	0,057 %
Total	0,828 %

Tabla 4.4: Área de las imágenes del dataset cubierta por cada clase.

4.5. Experimentación sobre el dataset

Con el fin de verificar la utilidad del dataset y analizar el desempeño de distintas bandas multiespectrales e índices de vegetación se planteó entrenar instancias de modelos para la detección de malezas. Se decidió probar el dataset tanto en la tarea de segmentación de instancias como en la de segmentación semántica.

Para llevar a cabo la segmentación semántica fue necesario generar máscaras densas a partir de los polígonos producidos durante el etiquetado. Cada polígono fue rasterizado sobre una máscara de un único canal asignando a sus píxeles el identificador entero correspondiente a su clase (1, 2 y 3 para dentada, plántula y roseta, respectivamente), mientras que los píxeles no cubiertos por ningún polígono fueron asignados a la clase background con valor 0.

Para la experimentación se decidió utilizar dos modelos, YOLOv8 para la segmentación de instancias y U-Net con encoders ResNet18 y EfficientNet-b0

para la segmentación semántica. La elección de YOLOv8 se basa en primer lugar en el hecho de que permite abordar simultáneamente localización y segmentación de instancias individuales utilizando imágenes multiespectrales con un equilibrio entre desempeño y costo computacional, y además ha sido utilizado en trabajos anteriores (Badgujar, Poulose, y Gan, 2024; Geng, Yu, Yuan, Ma, y Li, 2024; Joy y cols., 2025; Gallagher y Oughton, 2025). En cuanto al modelo U-Net, al igual que YOLOv8 ha sido utilizado en trabajos anteriores tanto con el encoder ResNet (Wang y cols., 2024) (Zhang y cols., 2026) como el encoder EfficientNet (Jain, Mishra, Arya, Chaudhary, y Yadav, 2025).

Tanto en las instancias de YOLOv8 como en las de U-Net fue necesario adaptar la primera capa convolucional para admitir más de tres canales de entrada. La modificación se realizó mediante la biblioteca PyTorch, sustituyendo dicha capa por una nueva capa convolucional con idéntico tamaño de kernel y stride, pero con un número de canales de entrada coincidente con la cantidad de bandas de cada configuración. Para favorecer una adaptación adecuada del modelo a los canales adicionales, los pesos correspondientes a los nuevos canales se inicializaron como el promedio de los pesos asociados a los canales RGB del modelo preentrenado, técnica empleada previamente en otros trabajos (Jain y cols., 2025; Rise, Uney, y Huang, 2026).

Para la experimentación con YOLOv8 se utilizó la biblioteca Ultralytics (Jocher y cols., 2023), de la cual se emplearon tanto la implementación del modelo como las funciones para su entrenamiento y evaluación. Para U-Net se utilizaron dos bibliotecas: Segmentation Models PyTorch (Iakubovskii, 2019), de donde se obtuvo la implementación de la arquitectura, y PyTorch (Ansel y cols., 2024), sobre la cual se programaron los scripts de entrenamiento y evaluación. El seguimiento del entrenamiento en tiempo real y el registro de las ejecuciones con sus respectivas configuraciones, hiperparámetros y métricas resultantes se gestionó mediante la herramienta Weights & Biases (Biewald, 2020).

Dado el reducido tamaño de las malezas en relación con el área total (ver Sección 4.4) de las imágenes, un reescalado directo a la resolución de entrada de los modelos (640×640 para YOLOv8 y 224×224 para U-Net con ambos encoders) implicaría la pérdida de gran parte de las instancias más pequeñas. Para mitigar este problema se adoptaron las técnicas de Slicing Aided Fine-tuning (SF) y Slicing Aided Hyper Inference (SAHI) (Akyon y cols., 2022), que consisten en entrenar y realizar inferencia sobre recortes de las imágenes originales, fusionando posteriormente las predicciones a nivel de imagen completa. Concretamente, se generaron 10 000 recortes de 640×640 píxeles con un solapamiento del 20 % entre recortes contiguos, los cuales fueron utilizados como entrada para el entrenamiento.

Todas las instancias de entrenamiento se ejecutaron en el clúster de cómputo ClusterUY (Nesmachnow y Iturriaga, 2019), sobre nodos con idéntica configuración de hardware: CPU AMD EPYC 7763 y una única GPU NVIDIA A40 por entrenamiento. En el Apéndice D se puede ver con mayor detalle el uso del ClusterUY.

En el Capítulo 5 se desarrolla con mayor detalle la experimentación con el dataset y se presentan sus resultados.

Todo el código necesario para llevar a cabo los procedimientos mencionados en este capítulo se encuentra disponible en un repositorio público[5]. En el Apéndice E se describe en detalle su uso.

[5] https://github.com/cristiangdev/weed-detection-dataset-pipeline

Capítulo 5

Experimentación

En este capítulo se presenta el diseño experimental definido para evaluar la utilidad del dataset construido y analizar, de forma controlada, el aporte de la información multiespectral en la detección de malezas. En particular, se comparan distintas configuraciones de entrada, incluyendo una línea base RGB y alternativas que incorporan bandas espectrales adicionales e índices de vegetación.

Con este fin, se describen la infraestructura de cómputo utilizada, los modelos considerados, las hipótesis de trabajo, las variables controladas, las configuraciones espectrales evaluadas y las métricas empleadas. Finalmente, se presentan los resultados consolidados y su análisis en relación con los objetivos del trabajo.

5.1. Infraestructura y modelos considerados

5.1.1. Infraestructura de cómputo

La totalidad de los entrenamientos reportados en este trabajo se ejecutaron sobre *ClusterUY* (Nesmachnow y Iturriaga, 2019), la plataforma nacional de supercómputo de Uruguay. En particular, los ensayos se realizaron en nodos equipados con procesadores AMD EPYC 7763 y GPUs NVIDIA A40 con 48 GB de memoria GDDR6, de la familia Ampere. Cada ensayo utilizó una única GPU, sin paralelismo multi-GPU. En el Apéndice D se describe con mayor detalle la infraestructura de ClusterUY y el flujo de trabajo que se utilizó para la ejecución de trabajos en la plataforma.

5.1.2. Modelos considerados

En este capítulo se reportan dos líneas experimentales efectivamente ejecutadas. La primera corresponde a YOLOv8 en su variante de segmentación de instancias, específicamente en la configuración YOLOv8-s-seg con pesos preentrenados, entrenada mediante la implementación provista por la biblioteca Ultralytics (Jocher y cols., 2023) en su versión 8.3. La elección de este modelo

respondió a dos razones principales. En primer lugar, permite abordar simultáneamente la localización y segmentación de instancias individuales de maleza, lo que resulta consistente con el tipo de etiquetado poligonal disponible en el dataset. En segundo lugar, constituye una arquitectura ampliamente utilizada en tareas de detección, con un equilibrio adecuado entre desempeño y costo computacional.

La segunda línea experimental corresponde a U-Net (Ronneberger y cols., 2015) para segmentación semántica, utilizando la implementación disponible en la biblioteca Segmentation Models PyTorch (Iakubovskii, 2019) junto con código de entrenamiento desarrollado con PyTorch (Ansel y cols., 2024). La elección de esta arquitectura respondió a su uso extendido en tareas de segmentación de imágenes. Como encoder principal se utilizó ResNet18 (He y cols., 2016), sobre el cual se ejecutaron las cinco configuraciones espectrales. Adicionalmente, con el objetivo de evaluar el efecto del encoder sobre el desempeño, se entrenaron dos ensayos complementarios con encoder EfficientNet-b0 (Tan y Le, 2020) sobre las configuraciones RGB y RGB + NIR. La elección de estas dos configuraciones respondió a que RGB constituye la línea base del estudio y RGB + NIR fue la combinación con mejor desempeño en los resultados obtenidos con YOLOv8 y U-Net con ResNet18.

5.2. Diseño experimental

5.2.1. Hipótesis

Se planteó como hipótesis principal que el dataset construido, al haber sido diseñado mediante un proceso controlado de selección, preprocesamiento y etiquetado, constituye una base adecuada para el entrenamiento de modelos de aprendizaje profundo orientados a la detección de malezas. En particular, se espera que la consistencia de las capturas, la calidad de las etiquetas y la alineación entre bandas faciliten el aprendizaje de patrones discriminativos.

De forma complementaria, se planteó que el uso de información multiespectral podría mejorar el desempeño respecto de configuraciones basadas únicamente en RGB. Asimismo, se consideró que el aporte de bandas adicionales e índices de vegetación podría variar según la representación de entrada utilizada.

5.2.2. Variables controladas y supuestos

Los experimentos fueron diseñados siguiendo un esquema de comparación controlada, en el cual se varía exclusivamente la representación espectral utilizada como entrada del modelo, mientras que el resto de los componentes del pipeline experimental se mantiene constante dentro de cada línea de experimentación. De esta manera, cualquier diferencia observada en el desempeño puede atribuirse principalmente a la información espectral disponible para el modelo.

Todos los ensayos se realizaron sobre el dataset descrito en el Capítulo 4, compuesto por capturas previamente sometidas al preprocesamiento presentado

en la Sección 4.2, incluyendo la corrección radiométrica y la alineación entre bandas. Esto permitió reutilizar de forma consistente las etiquetas generadas sobre representaciones RGB en las distintas configuraciones multiespectrales evaluadas.

Las imágenes del dataset fueron etiquetadas manualmente mediante polígonos que describen la forma observable de cada planta en la escena. Este tipo de etiqueta resulta especialmente adecuado para el problema abordado, ya que las malezas presentan contornos irregulares, tamaños reducidos y frecuentes situaciones de superposición parcial con cultivo, suelo o rastrojo. El esquema de etiquetado contempla tres categorías de maleza: *plántula*, *roseta* y *dentada*, definidas a partir de morfologías observadas durante el proceso de etiquetado (ver Sección 4.3).

Un aspecto que conviene precisar antes de presentar los ensayos es el tratamiento diferencial que ambos enfoques realizan sobre el fondo de la escena, dado que sus formulaciones imponen decisiones distintas a tres niveles: la generación de etiquetas, el aprendizaje y la evaluación. A nivel de etiquetas, YOLOv8-s-seg opera directamente sobre las anotaciones poligonales y no requiere una etiqueta explícita para el fondo: este queda definido implícitamente como la ausencia de polígono. U-Net, en cambio, demanda máscaras densas, por lo que las etiquetas se obtienen rasterizando los polígonos sobre una imagen del mismo tamaño que la original y asignando a cada píxel un valor entero correspondiente a la clase de la instancia que lo contiene, reservando el valor cero para el fondo. A nivel de aprendizaje, esta diferencia se traduce en que YOLOv8 aprende únicamente a proponer y clasificar instancias de maleza, mientras que U-Net aprende a clasificar cada píxel entre cuatro categorías: *plántula*, *roseta*, *dentada* y fondo. A nivel de evaluación, las métricas se computan de manera consistente con cada formulación. En YOLOv8-s-seg, las métricas por clase y los promedios macro contemplan únicamente las tres clases de maleza, ya que el fondo no constituye una clase entrenable del modelo. En U-Net, el fondo se excluye del promedio macro de Dice, IoU, precisión y recall, dado que su preponderancia entre los píxeles del dataset (ver Sección 4.4) inflaría artificialmente los valores agregados; la métrica de *pixel accuracy*, en cambio, se computa sobre la totalidad de los píxeles e incluye al fondo, reflejando la proporción global de píxeles correctamente clasificados. Estas decisiones se retomarán en el análisis de las matrices de confusión (Sección 5.4 y Sección 5.5) y en la discusión de los resultados.

Las imágenes originales del dataset poseen una resolución de 2445×1890 píxeles, incompatible con la entrada de tamaño fijo requerida por las arquitecturas evaluadas. Alimentar la red con la imagen completa habría exigido un redimensionamiento agresivo, con la consecuente pérdida de detalle espacial crítica para la detección de instancias pequeñas de malezas. Como alternativa, se adoptó una estrategia de *tiling* con recortes de 640×640 píxeles y un solapamiento del 20 % entre tiles adyacentes. El solapamiento mitiga el truncamiento de instancias en los bordes de los recortes, asegurando que cada planta aparezca íntegra en al menos un tile. Este enfoque se alinea con estrategias ampliamente utilizadas para detección y segmentación sobre imágenes de alta resolución (Akyon y cols., 2022; Tessore, 2025).

La partición del dataset en entrenamiento, validación y prueba se realizó *antes* del *tiling*, sobre las 500 imágenes originales, con una proporción del 80 % para entrenamiento, 10 % para validación y 10 % para prueba. La estratificación se efectuó por composición multi-clase de las instancias presentes en cada imagen, de modo de preservar en cada subconjunto una distribución comparable de las clases *plántula*, *roseta* y *dentada*. Recién a partir de esa partición se generaron los recortes, procesando cada subconjunto de manera independiente. Esta decisión es deliberada: realizar el *tiling* antes del split habría dado lugar a que recortes solapados o derivados de una misma imagen pudieran quedar repartidos entre distintos subconjuntos, generando una contaminación informativa entre entrenamiento, validación y prueba.

La misma partición se utilizó en ambas líneas experimentales. En YOLOv8, los recortes de 640×640 se emplearon directamente como entrada del modelo. En U-Net, esos mismos recortes constituyeron la base del experimento y fueron posteriormente reescalados a la resolución de entrada utilizada por la arquitectura. Los parámetros completos del proceso de *slicing* se reportan en el Apéndice B.

Como supuestos de trabajo, se consideró que el preprocesamiento produjo capturas radiométrica y geométricamente comparables, que la alineación entre bandas preservó la correspondencia espacial entre imágenes y etiquetas, y que la partición del dataset resultó representativa de la variabilidad presente en el conjunto de datos. Cabe señalar que, al tratarse de capturas provenientes de una única localidad, fecha y cultivo, el alcance de esta representatividad se limita a las condiciones específicas del escenario estudiado.

5.2.3. Configuración experimental

Se ejecutaron cinco ensayos completos por cada modelo, variando exclusivamente la representación espectral de entrada y manteniendo constante el resto de la configuración general dentro de cada línea experimental.

En el caso de YOLOv8-s-seg se utilizó la implementación provista por la biblioteca Ultralytics en su serie 8.3, con una resolución de entrada de 640×640 píxeles, un máximo de 400 épocas, un tamaño de lote de 8 imágenes, el optimizador por defecto de Ultralytics (*auto*, que selecciona automáticamente entre SGD y AdamW en función del número de iteraciones de entrenamiento) y la misma partición del dataset descrita anteriormente. Asimismo, se utilizó un criterio de *early stopping* con una paciencia de 50 épocas, en el que el entrenamiento se detenía en caso de no mejorar el mAP50-95 luego de 50 épocas. Además, se mantuvo la configuración de *data augmentation* provista por defecto por YOLOv8 en todos los ensayos. En cuanto a la función de pérdida, se utilizó la definida por defecto en la configuración de Ultralytics, basada en una combinación lineal de tres componentes: Complete Intersection over Union (CIoU), que penaliza la diferencia entre cajas predichas y reales considerando solapamiento, distancia y relación de aspecto; Binary Cross Entropy (BCE), utilizada para la clasificación de las instancias; y Distribution Focal Loss (DFL), que mejora la regresión de los bordes de las cajas delimitadoras (Jocher y cols., 2023). La Tabla 5.1 resume

los parámetros generales comunes a todas las ejecuciones reportadas con este modelo. La configuración completa de hiperparámetros de entrenamiento, los pesos de la función de pérdida y las transformaciones de *data augmentation* se detallan en el Apéndice B.

Parámetro	Valor
Modelo	YOLOv8-s-seg
Implementación	Ultralytics YOLOv8 (serie 8.3)
Pesos iniciales	Preentrenados
Resolución de entrada	640 × 640
Épocas máximas	400
Batch size	8
Optimizador	Auto (SGD / AdamW)
Early stopping	Paciencia de 50 épocas
Data augmentation	Configuración por defecto de YOLOv8
Partición del dataset	80 train / 10 val / 10 test

Tabla 5.1: Configuración general de los ensayos reportados con YOLOv8-s-seg.

En el caso de U-Net se utilizó la implementación disponible en Segmentation Models PyTorch. Todos los ensayos se ejecutaron sobre la misma partición del conjunto de recortes utilizada en YOLOv8. Los recortes originales de 640 × 640 se reescalaron a una resolución de entrada de 224 × 224 píxeles, manteniéndose constante esta configuración en todos los ensayos reportados. La salida del modelo corresponde a máscaras de segmentación semántica con cuatro clases: *background*, *plántula*, *roseta* y *dentada*. Se utilizaron dos encoders preentrenados: ResNet18 como encoder principal para las cinco configuraciones espectrales, y EfficientNet-b0 como encoder complementario para las configuraciones RGB y RGB + NIR. La Tabla 5.2 resume la configuración general utilizada. Los hiperparámetros completos de entrenamiento y las transformaciones de *data augmentation* aplicadas en estos ensayos se detallan en el Apéndice B.

Parámetro	Valor
Modelo	U-Net
Encoders	ResNet18 / EfficientNet-b0
Implementación	Segmentation Models PyTorch + PyTorch
Pesos iniciales del encoder	Preentrenados (ImageNet)
Entrada efectiva	Recortes de 640 × 640 reescalados
Resolución de entrada	224 × 224
Épocas máximas	400
Clases de salida	background, plántula, roseta, dentada
Partición del dataset	80 train / 10 val / 10 test

Tabla 5.2: Configuración general de los ensayos reportados con U-Net.

Dado que tanto YOLOv8 como U-Net están diseñados originalmente para

trabajar con imágenes RGB de tres canales, fue necesario adaptar la primera capa convolucional para aceptar el número de canales correspondiente a cada configuración experimental. En la configuración RGB se utilizaron los pesos preentrenados estándar para los tres canales visibles. En las configuraciones con canales adicionales, los pesos correspondientes a los canales RGB se conservaron a partir del preentrenamiento, mientras que los canales extra se inicializaron utilizando la media de los pesos RGB. En el caso de la entrada monocanal basada en NDVI, la inicialización también se realizó a partir de dicha media. El detalle del procedimiento de adaptación y la normalización aplicada a la entrada se incluyen en el Apéndice B.

5.2.4. Configuraciones espectrales evaluadas

La configuración base correspondió a la representación RGB, compuesta por tres canales del espectro visible. A partir de esta línea base se evaluaron cuatro variantes adicionales: NDVI como entrada monocanal, RGB + NDVI, RGB + NIR y RGB + NIR + RE. La Tabla 5.3 resume las configuraciones consideradas, que fueron utilizadas tanto en YOLOv8 como en U-Net.

Configuración	# canales	Inicialización
RGB	3	RGB
NDVI	1	Media de RGB
RGB + NDVI	4	RGB + media de RGB
RGB + NIR	4	RGB + media de RGB
RGB + NIR + RE	5	RGB + media de RGB

Tabla 5.3: Configuraciones espectrales evaluadas en los ensayos reportados.

Estas configuraciones fueron seleccionadas con el objetivo de comparar distintas formas de incorporar información espectral adicional al modelo. En particular, se buscó evaluar, por un lado, el aporte de bandas no visibles como NIR y RE y, por otro, una representación derivada como NDVI. De esta forma, el diseño experimental permite comparar tanto bandas originales como variables espectrales compuestas bajo un mismo esquema metodológico. La literatura consultada sobre trabajos en distintos cultivos no indicó que existiera una banda o índice de vegetación en particular que mejore la detección de malezas de manera general.

5.2.5. Tiempos de entrenamiento e inferencia

Como referencia del costo computacional de la experimentación, se reportan a continuación los tiempos observados sobre la infraestructura de ClusterUY descrita en el Apéndice D. Todos los valores reportados corresponden a ensayos ejecutados sobre nodos con GPU NVIDIA A40.

Para los ensayos con YOLOv8-s-seg, el entrenamiento sobre 400 épocas (o su detención anticipada por *early stopping*) insumió entre aproximadamente 7 y 18

horas según la configuración espectral, midiendo el límite inferior en la configuración NDVI monocanal y el superior en RGB + NIR + RE. Las configuraciones intermedias —RGB + NIR y RGB + NDVI— se ubicaron en torno a las 12 horas. El tiempo de inferencia por imagen, medido sobre el conjunto de prueba, se mantuvo en el orden de los 3 milisegundos para todas las configuraciones, contemplando las etapas de preprocesamiento, inferencia y posprocesamiento.

Para U-Net con encoder ResNet18, el entrenamiento del ensayo RGB sobre 400 épocas insumió aproximadamente 8 horas y media, valor que puede considerarse representativo del costo de un ensayo individual con esta arquitectura, dado que el resto de las configuraciones se ejecutó sobre la misma infraestructura y con idéntica cantidad de épocas máximas.

5.3. Métricas

Para los ensayos de segmentación de instancias con YOLOv8 se reportan como métricas principales el *mean Average Precision* en el rango $IoU = 0{,}50{:}0{,}95$ (ver Sección 2.6), tanto para cajas delimitadoras (**Box mAP50-95**) como para máscaras de segmentación (**Mask mAP50-95**). Dado que el etiquetado se realizó mediante polígonos, la métrica sobre máscaras es la más informativa para evaluar la delimitación espacial de cada planta. De forma complementaria, durante el seguimiento del entrenamiento se consideraron también *precision*, *recall* y mAP50.

Para los ensayos de segmentación semántica con U-Net se reportan *Dice*, *Intersection over Union* (IoU), *precision*, *recall* y *pixel accuracy* (ver Sección 2.6), tanto a nivel global como desagregadas por clase (*dentada*, *plántula* y *roseta*), con el fin de identificar posibles diferencias de dificultad entre categorías.

5.4. Resultados de segmentación de instancias

Las curvas de entrenamiento de todos los ensayos reportados en esta sección, incluyendo la *loss* y el mAP50-95 sobre el conjunto de validación a lo largo de las épocas, se incluyen en el Apéndice C.

La Tabla 5.4 resume los resultados obtenidos con YOLOv8-s-seg para las distintas configuraciones espectrales evaluadas. Se muestran las métricas tanto para la detección mediante cajas delimitadoras (bounding boxes) como para las máscaras de segmentación de instancias. En ambos casos, la referencia utilizada corresponde a las cajas y máscaras derivadas de los polígonos del etiquetado original. Las métricas son los promedios macro (promedios entre clases) de la precisión (P), recall (R), mAP50, mAP50-95 y F1. Cabe destacar que la precisión, el recall y el F1 se calculan basados en los niveles de confianza que maximizan el F1 para cada clase.

La mejor configuración en esta línea experimental fue RGB + NIR, con un Box mAP50-95 de 0,770 y un Mask mAP50-95 de 0,598. El esquema RGB + NIR + RE obtuvo resultados muy próximos, con valores de 0,766 y 0,590

Input	Box					Mask				
	P	R	mAP50	mAP50-95	F1	P	R	mAP50	mAP50-95	F1
RGB	**0,924**	0,853	0,912	0,761	0,887	0,923	0,852	0,909	0,570	0,886
RGB + NIR	0,904	0,883	**0,922**	**0,770**	**0,893**	**0,928**	0,858	**0,919**	**0,598**	**0,892**
RGB + NIR + RE	0,891	**0,893**	0,919	0,766	0,892	0,890	**0,887**	0,916	0,590	**0,892**
RGB + NDVI	**0,924**	0,863	0,915	0,755	0,892	0,923	0,862	0,905	0,574	**0,892**
NDVI	0,893	0,809	0,859	0,657	0,849	0,884	0,802	0,850	0,460	0,841

Tabla 5.4: Resultados de YOLOv8-s-seg para diferentes combinaciones de canales.

respectivamente, lo que sugiere que la incorporación de la banda NIR aporta una mejora consistente respecto de RGB, mientras que el aporte adicional de red-edge resulta más acotado en esta formulación.

La combinación RGB + NDVI también mostró un comportamiento competitivo, especialmente en segmentación por máscaras, aunque sin superar a las variantes que incorporan NIR. En cambio, NDVI como entrada monocanal obtuvo los resultados más bajos de la línea experimental, lo que indica que, si bien concentra información útil, no resulta suficiente por sí solo para igualar el desempeño de las alternativas multicanal.

La Tabla 5.5 presenta el detalle por clase para las métricas de segmentación por máscara. Este análisis permite observar que el efecto de la información espectral no es uniforme entre las distintas categorías de maleza.

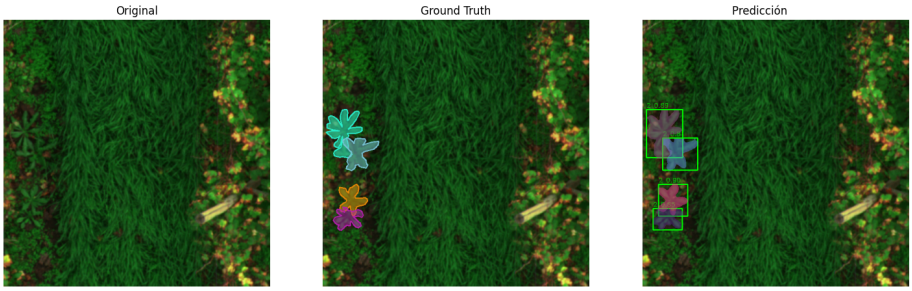
Clase	Input	mAP50	mAP50-95	Precision	Recall	F1
Dentada	RGB	0,8819	0,5744	0,9190	0,8257	0,8699
Dentada	RGB + NIR	**0,9061**	**0,6423**	**0,9367**	0,8545	**0,8938**
Dentada	RGB + NIR + RE	0,9046	0,6165	0,9051	**0,8666**	0,8854
Dentada	RGB + NDVI	0,8487	0,5755	0,9149	0,8364	0,8739
Dentada	NDVI	0,8634	0,5142	0,8795	0,8182	0,8477
Plántula	RGB	0,8978	0,5369	0,8936	0,8377	0,8648
Plántula	RGB + NIR	0,8965	0,5349	0,8988	0,8263	0,8610
Plántula	RGB + NIR + RE	0,8898	0,5388	0,8272	**0,8798**	0,8527
Plántula	RGB + NDVI	**0,9163**	**0,5461**	**0,9006**	0,8433	**0,8710**
Plántula	NDVI	0,8026	0,3901	0,8732	0,7494	0,8066
Roseta	RGB	0,9485	0,6021	**0,9573**	0,8912	0,9231
Roseta	RGB + NIR	0,9534	**0,6156**	0,9499	0,8935	0,9208
Roseta	RGB + NIR + RE	**0,9548**	0,6142	0,9371	**0,9157**	0,9263
Roseta	RGB + NDVI	0,9487	0,6018	0,9542	0,9078	**0,9304**
Roseta	NDVI	0,8853	0,4754	0,9001	0,8394	0,8687

Tabla 5.5: Resultados por clase obtenidos con YOLOv8-s-seg sobre el conjunto de prueba, desglosados por configuración espectral de entrada.

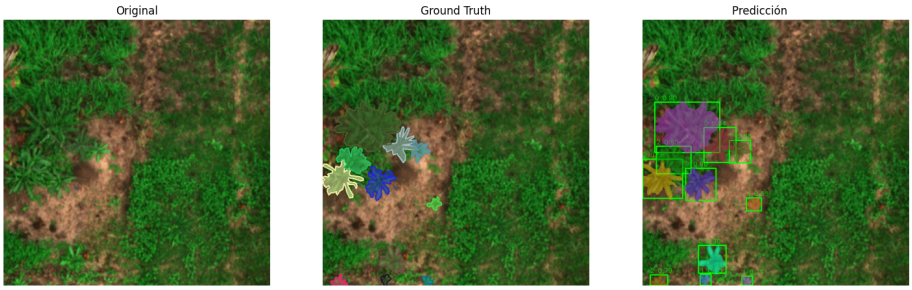
A nivel de clase, la categoría *dentada* mostró su mejor desempeño con RGB + NIR. En *plántula*, la mejor configuración fue RGB + NDVI, aunque con diferencias acotadas respecto de las restantes variantes multicanal. En *roseta*, los mejores resultados en mAP50-95 se obtuvieron con RGB + NIR y RGB + NIR + RE, nuevamente con diferencias pequeñas entre ambas configuraciones.

En conjunto, estos resultados refuerzan la idea de que el aporte de la información espectral depende también de la clase considerada.

Además de las métricas agregadas, resulta útil complementar el análisis con una inspección cualitativa de las predicciones generadas por el modelo de mejor desempeño, correspondiente a la entrada RGB + NIR. La Figura 5.1 presenta dos escenas del conjunto de prueba en las que se comparan la imagen original, las etiquetas de referencia y la predicción obtenida por YOLOv8-s-seg. En términos generales, puede observarse que el modelo recupera varias instancias de maleza presentes en la imagen y reproduce de forma razonable tanto su localización como su delimitación espacial. A su vez, la comparación visual permite identificar limitaciones que no siempre quedan reflejadas con el mismo nivel de detalle en las métricas globales, particularmente en instancias pequeñas, de contorno difuso o con menor nivel de confianza.



(a) Escena 1.



(b) Escena 2.

Figura 5.1: Ejemplos cualitativos de predicciones obtenidas con YOLOv8-s-seg usando entrada RGB + NIR sobre dos imágenes del conjunto de prueba. En cada caso se muestra la imagen original (izquierda), las etiquetas de referencia (centro) y la predicción del modelo (derecha).

Con el propósito de complementar las métricas globales, la Figura 5.2 presenta la matriz de confusión normalizada obtenida sobre el conjunto de prueba para el modelo YOLOv8-s-seg con entrada RGB + NIR. Los mayores valores se concentran sobre la diagonal principal y las confusiones entre las distintas clases

de maleza son reducidas, lo que sugiere que las categorías morfológicas definidas durante el etiquetado resultaron suficientemente consistentes y discriminables entre sí. La columna y la fila correspondientes a *background* requieren una interpretación particular. La fila de *background* contiene los falsos negativos del modelo: la fracción de instancias reales de cada clase que no fueron detectadas y, por lo tanto, fueron asignadas al fondo. Estos valores son moderadamente bajos (entre 0,06 y 0,12), indicando una capacidad razonable de detección. La columna de *background*, en cambio, contiene los falsos positivos del modelo: regiones del fondo que fueron incorrectamente identificadas como maleza. Esta columna está normalizada sobre el total de falsos positivos, por lo que sus valores expresan cómo se distribuyen dichos errores entre las clases predichas, no la proporción de fondo confundido con cada clase.

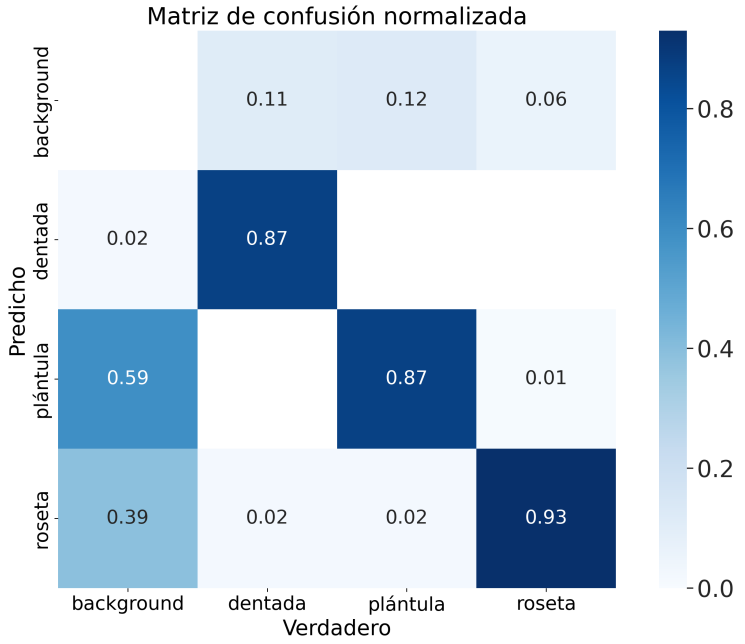


Figura 5.2: Matriz de confusión normalizada obtenida por YOLOv8-s-seg con entrada RGB + NIR sobre el conjunto de prueba para las clases *dentada*, *plántula* y *roseta*.

En conjunto, la matriz respalda la validez del esquema de etiquetado adoptado y, simultáneamente, identifica la separación entre objeto y *background* como uno de los aspectos más desafiantes del problema, particularmente para las clases morfológicamente más sutiles.

5.5. Resultados de segmentación semántica

Las curvas de entrenamiento de todos los ensayos reportados en esta sección, incluyendo la *loss* y el mIoU sobre el conjunto de validación a lo largo de las épocas, se incluyen en el Apéndice C.

La Tabla 5.6 resume los resultados globales obtenidos con U-Net para las distintas configuraciones espectrales consideradas. Las métricas Dice, IoU, precisión y recall se reportan como promedios macro (promedios entre clases), excluyendo la clase background de dicho promedio debido a su gran preponderancia en las imágenes (como se señaló en la Sección 4.4), lo que implicaría una inflación de los resultados obtenidos en cuanto a la detección de malezas. Pixel Accuracy, en cambio, se calcula sobre la totalidad de los píxeles de la imagen, incluyendo el fondo, por lo que refleja la proporción global de píxeles correctamente clasificados.

Input	Dice	mIoU (macro)	Precision	Recall	Pixel Acc
RGB	0,8635	0,8349	0,9160	**0,9033**	**0,9981**
RGB + NIR	**0,8697**	**0,8417**	0,9265	0,9023	**0,9981**
RGB + NIR + RE	0,8681	0,8406	**0,9274**	0,9007	**0,9981**
RGB + NDVI	0,8671	0,8391	0,9264	0,8978	0,9980
NDVI	0,8099	0,7845	0,9253	0,8385	0,9973
Encoder EfficientNet-b0					
RGB (Eff.)	**0,8779**	**0,8488**	0,9184	**0,9147**	**0,9982**
RGB + NIR (Eff.)	0,8736	0,8449	**0,9192**	0,9114	**0,9982**

Tabla 5.6: Resultados globales de U-Net.

A nivel global, considerando los ensayos con encoder ResNet18, la mejor configuración volvió a ser RGB + NIR, que alcanzó los mayores valores de Dice y mIoU. La configuración RGB + NIR + RE obtuvo resultados muy próximos, mientras que RGB + NDVI mostró una mejora moderada respecto de la línea base RGB. En cambio, NDVI como entrada monocanal volvió a presentar el peor desempeño global, particularmente en Dice e IoU. En cuanto a los valores de Pixel Accuracy, su cercanía a 1 se debe a la preponderancia del background en las imágenes como se señala en la Sección 4.4.

La Tabla 5.7 presenta el detalle de resultados por clase. Este análisis permite observar que el comportamiento de las configuraciones no es uniforme para todas las categorías de maleza. Se muestran las métricas Dice, IoU, precisión y recall.

Clase	**Input**	Dice	IoU	Precision	Recall
Dentada	RGB	0,9613	0,9558	0,9750	0,9789
Dentada	RGB + NIR	0,9712	**0,9653**	0,9810	**0,9815**
Dentada	RGB + NIR + RE	**0,9713**	0,9651	**0,9836**	0,9789
Dentada	RGB + NDVI	0,9648	0,9591	0,9786	0,9784
Dentada	NDVI	0,9465	0,9413	0,9756	0,9631
Dentada	RGB (Eff.)	**0,9777**	**0,9715**	**0,9870**	**0,9823**
Dentada	RGB + NIR (Eff.)	0,9700	0,9645	0,9811	0,9813
Plántula	RGB	0,8219	0,7927	0,9019	**0,8715**
Plántula	RGB + NIR	**0,8419**	**0,8143**	0,9317	0,8683
Plántula	RGB + NIR + RE	0,8236	0,7973	0,9186	0,8661
Plántula	RGB + NDVI	0,8333	0,8064	0,9299	0,8598
Plántula	NDVI	0,7737	0,7526	**0,9443**	0,7853
Plántula	RGB (Eff.)	0,8350	0,8063	0,9047	**0,8860**
Plántula	RGB + NIR (Eff.)	**0,8448**	**0,8152**	**0,9121**	0,8845
Roseta	RGB	0,8072	0,7560	0,8711	**0,8596**
Roseta	RGB + NIR	0,7960	0,7456	0,8669	0,8570
Roseta	RGB + NIR + RE	**0,8093**	**0,7592**	**0,8799**	0,8571
Roseta	RGB + NDVI	0,8034	0,7519	0,8707	0,8554
Roseta	NDVI	0,7095	0,6595	0,8558	0,7671
Roseta	RGB (Eff.)	**0,8210**	**0,7686**	0,8634	**0,8759**
Roseta	RGB + NIR (Eff.)	0,8059	0,7549	**0,8643**	0,8685

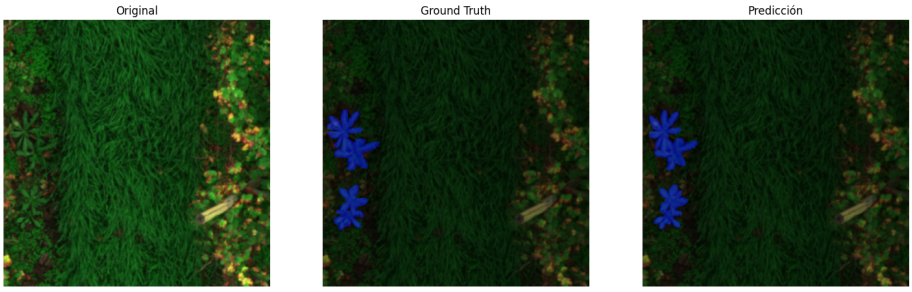
Tabla 5.7: Resultados por clase obtenidos con U-Net sobre el conjunto de prueba, desglosados por configuración espectral de entrada. Se señala con *Eff.* en la columna input los casos con el encoder EfficientNet-b0.

La clase *dentada* fue la más sencilla de segmentar en todas las configuraciones, con valores de Dice superiores a 0,94 incluso en el caso monocanal. En *plántula*, la mejor configuración con ResNet18 fue RGB + NIR, lo que sugiere que esta clase se beneficia particularmente de la incorporación de información espectral adicional. En *roseta*, las diferencias entre configuraciones fueron menores; el mejor valor con ResNet18 se obtuvo con RGB + NIR + RE, aunque la ganancia respecto de RGB resultó reducida.

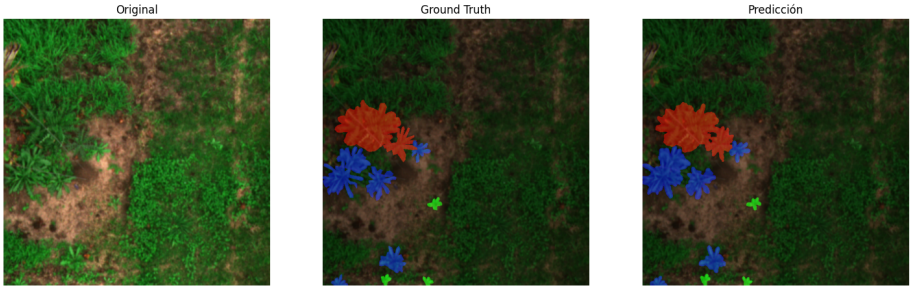
Los ensayos con EfficientNet-b0 mostraron mejoras en las tres clases para la configuración RGB, destacándose un aumento de recall en *roseta* (de 0,8596 a 0,8759) y en *plántula* (de 0,8715 a 0,8860). En la configuración RGB + NIR, EfficientNet-b0 obtuvo el mejor Dice global para *plántula* (0,8448) entre todos los ensayos, superando incluso a la misma configuración con ResNet18 (0,8419). En conjunto, estos resultados indican que la utilidad de cada configuración espectral depende tanto de la clase considerada como del encoder utilizado.

Para complementar las métricas globales y por clase, la Figura 5.3 presenta dos ejemplos cualitativos de segmentación obtenidos con U-Net utilizando entrada RGB + NIR. Al igual que en los resultados de YOLOv8-s-seg, se observa una correspondencia visual razonable entre las regiones etiquetadas y las máscaras predichas. Estos ejemplos también permiten apreciar mejor los casos en los que el modelo segmenta adecuadamente plantas bien definidas y aquellos en los

que la delimitación se vuelve más difícil debido al tamaño reducido de algunas instancias o al bajo contraste con el fondo.



(a) Escena 1.



(b) Escena 2.

Figura 5.3: Ejemplos cualitativos de predicciones obtenidas con U-Net usando entrada RGB + NIR sobre dos imágenes del conjunto de prueba. En cada caso se muestra la imagen original (izquierda), las etiquetas de referencia (centro) y la predicción del modelo (derecha). Verde indica *plántula*, azul indica *roseta* y rojo *dentada*.

Con el mismo propósito que en la sección anterior, la Figura 5.4 presenta la matriz de confusión normalizada obtenida por U-Net con entrada RGB + NIR sobre el conjunto de prueba. Al igual que en YOLOv8-s-seg, los valores se concentran sobre la diagonal principal, lo que indica que las confusiones entre clases de maleza son reducidas también en la tarea de segmentación semántica.

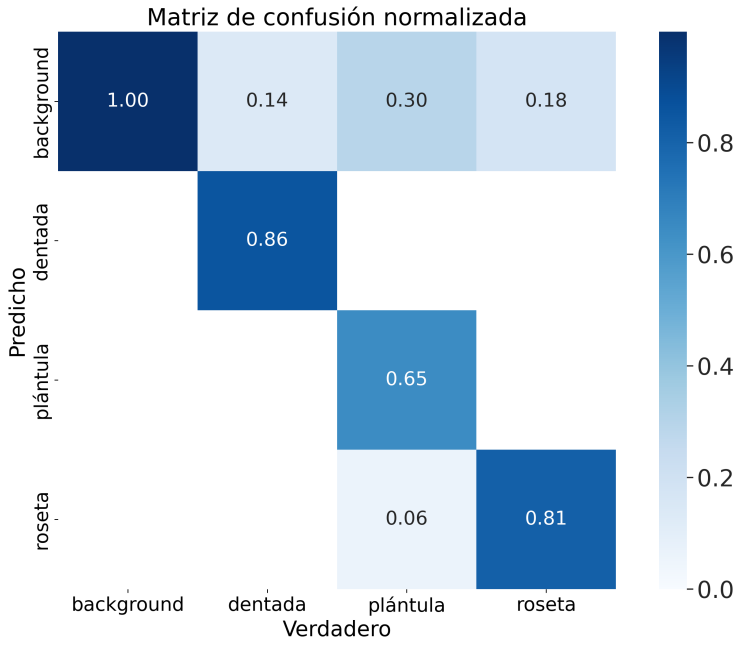


Figura 5.4: Matriz de confusión normalizada obtenida por U-Net con encoder ResNet18 y entrada RGB + NIR sobre el conjunto de prueba para las clases *dentada*, *plántula* y *roseta*.

5.6. Análisis comparativo de ensayos

Es relevante señalar que los resultados de segmentación de instancias y segmentación semántica no son directamente comparables en valor absoluto, ya que responden a tareas distintas y se evalúan con métricas diferentes. Sin embargo, sí resulta significativo que ambas líneas muestren tendencias consistentes en el efecto de las configuraciones espectrales, ya que esto fortalece la validez de las conclusiones extraídas.

Consideradas en conjunto, ambas líneas experimentales muestran una tendencia consistente: la incorporación de información multiespectral mejora el desempeño respecto de una representación puramente RGB, y la banda NIR es el aporte adicional más frecuentemente beneficioso entre las configuraciones evaluadas, aunque con excepciones según el encoder y la clase. En efecto, RGB + NIR obtuvo el mejor desempeño global tanto en YOLOv8-s-seg como en U-Net con encoder ResNet18. Sin embargo, en el caso de U-Net con encoder EfficientNet-b0 no se observó una mejora respecto de RGB.

La configuración RGB + NIR + RE también mostró resultados competitivos en ambos casos, pero sin superar de forma clara a RGB + NIR. Esto sugiere que,

en las condiciones de este dataset y para las arquitecturas consideradas, la banda red-edge aporta información complementaria, aunque su efecto incremental es menor que el de NIR.

La configuración RGB + NDVI presentó un comportamiento intermedio. En segmentación semántica mejoró levemente respecto de RGB y, en segmentación de instancias, mantuvo resultados competitivos aunque por debajo de las configuraciones con NIR. NDVI como entrada monocanal, en cambio, fue la alternativa más débil en ambas líneas experimentales. Este resultado puede explicarse, al menos en parte, por la pérdida de información de textura espacial y diversidad espectral al reducir la entrada a un único canal derivado. Adicionalmente, la inicialización de los pesos a partir de la media de los canales RGB puede resultar menos adecuada para un índice derivado como NDVI que para una banda espectral cruda, y la tendencia del NDVI a saturarse en zonas de alta densidad vegetal podría limitar su capacidad discriminativa en determinadas escenas. En consecuencia, su valor parece ser mayor como canal complementario que como única representación de entrada.

Adicionalmente, los ensayos complementarios con encoder EfficientNet-b0 sugieren que el desempeño de U-Net puede mejorar al utilizar un encoder con mayor exactitud en tareas de clasificación previa. En la configuración RGB, EfficientNet-b0 superó a ResNet18 en Dice (0,8779 vs. 0,8635) y mIoU (0,8488 vs. 0,8349), alcanzando el mejor resultado global entre todos los ensayos de segmentación semántica. Esto indica que, para una misma configuración espectral, la elección del encoder puede tener un efecto comparable al de la incorporación de bandas adicionales.

5.7. Discusión de resultados

A partir de los resultados y tendencias identificadas en las secciones anteriores, esta sección discute los hallazgos principales en relación con la literatura existente y las limitaciones del diseño experimental.

En primer lugar, los resultados presentados en la sección anterior aportan evidencia a favor de la hipótesis de que el dataset construido constituye una base útil para el entrenamiento de modelos de detección y segmentación de malezas.

La convergencia entre los resultados de segmentación de instancias y segmentación semántica respalda esta lectura: ambas familias de modelos alcanzan niveles de desempeño no triviales sobre las tres clases morfológicas definidas, lo que indica que el dataset contiene señal suficiente para aprender el problema y no se limita a una arquitectura particular.

Como se señaló en el análisis de las matrices de confusión (Figuras 5.2 y 5.4), la baja proporción de confusión entre clases en ambas líneas experimentales respalda que los criterios de definición de clases permitieron definir conjuntos de objetos internamente coherentes e independientes entre sí.

Un análisis más detallado de las matrices revela, sin embargo, una asimetría sistemática entre ambos modelos en el manejo del fondo. En YOLOv8-s-seg (Figura 5.2), la columna correspondiente a *background* concentra valores sig-

nificativos sobre las filas de *plántula* (0,59) y *roseta* (0,39), que expresan la distribución de los falsos positivos del modelo entre las clases predichas, es decir, predicciones realizadas sobre regiones de fondo real que fueron asignadas a alguna clase de maleza. Cabe señalar que esta columna no presenta diagonal porque YOLOv8 no aprende el fondo como una clase entrenable: las celdas asociadas a *background* en la matriz son un recurso de Ultralytics para contabilizar las predicciones sin correspondencia con etiquetas y las etiquetas sin correspondencia con predicciones (Jocher y cols., 2023). En cambio, en U-Net (Figura 5.4) la columna de *background* se concentra prácticamente por completo en la diagonal (1,00), mientras que su fila acumula valores no despreciables sobre las columnas de *plántula* (0,30), *roseta* (0,18) y *dentada* (0,14), correspondientes a falsos negativos: píxeles de maleza clasificados como fondo. Esta asimetría se explica porque U-Net trata al fondo como una de sus cuatro clases de salida y, dada su preponderancia en los píxeles del dataset (ver Sección 4.4), el modelo aprende un fuerte sesgo a su favor que lo vuelve conservador frente a la duda. En síntesis, ambos modelos enfrentan la separación maleza/fondo de manera distinta: YOLOv8-s-seg tiende a sobre-predecir maleza sobre regiones de fondo (falsos positivos), mientras que U-Net tiende a omitir píxeles de maleza por asignarlos a la clase fondo (falsos negativos).

La coincidencia entre YOLOv8 y U-Net en señalar a RGB + NIR como la configuración más robusta es consistente con la interpretación de que la información espectral en el infrarrojo cercano aporta información para el problema abordado. Estos resultados coinciden con los obtenidos en (J. Wu y cols., 2025), donde se buscó detectar malezas en un cultivo de trigo sarraceno y se obtuvieron mejores resultados utilizando una modificación de la arquitectura U-Net. Además, tanto en este trabajo como en (J. Wu y cols., 2025) la combinación RGB + NIR superó en desempeño a la combinación RGB + NIR + RE a pesar de que esta última incorpora más información. No obstante, no puede descartarse que este resultado refleje en parte la estrategia de inicialización utilizada para los canales adicionales o el estado fenológico particular de la fecha de captura, en el cual el contraste en la banda red-edge podría ser reducido.

El hecho de que agregar más información no repercuta en un mejor desempeño es un resultado que se ha observado en otros trabajos. Por ejemplo, en el caso de (Sa, Chen, y cols., 2018), donde la combinación NIR + R logra mejores resultados que la combinación NIR + R + NDVI. En (Sa, Popović, y cols., 2018) la combinación de 9 canales RGB + CIR + NDVI + NIR + RE proporcionó mejores resultados que combinaciones de 11 y 12 canales. Como se señaló en la sección anterior, los ensayos con EfficientNet-b0 también evidencian este fenómeno, sugiriendo que un encoder con mayor exactitud en tareas de clasificación previa podría extraer suficiente información discriminativa a partir de las bandas RGB, aunque no es posible descartar que el reescalado a 224×224 limite el aprovechamiento de la información NIR independientemente del encoder.

Además, cabe destacar que la combinación RGB + NIR obtuvo los mejores resultados dentro del grupo de configuraciones analizadas y para el cultivo y especie de maleza particulares de este estudio. En otros escenarios, la configuración óptima podría ser distinta. Por ejemplo, en (Sa, Popović, y cols., 2018),

donde se trabaja sobre cultivos de remolacha azucarera en Alemania, la combinación RGB + NIR fue superada por varias alternativas, incluyendo RGB + CIR + NDVI + RE.

Cabe destacar que la mejora de resultados debida al uso de canales multiespectrales y del índice de vegetación NDVI no es uniforme entre las clases definidas. En la segmentación de instancias, para la clase plántula, las combinaciones RGB + NDVI y RGB + NIR + RE obtuvieron un mAP50-95 mayor que el baseline RGB. En la clase roseta, tanto RGB + NIR como RGB + NIR + RE mostraron un mejor desempeño que el baseline. Sin embargo, la mayor mejora se observó en la clase dentada. En la línea de segmentación semántica, la clase plántula fue la que presentó mayores mejoras al agregar la banda NIR. En el caso de las rosetas, la configuración RGB + NIR también logró superar el baseline RGB. Esto sugiere que los resultados dependen no solo de las características espectrales de cada clase, sino también de factores propios de cada arquitectura, como la resolución de entrada efectiva y la forma en que cada modelo procesa la información multicanal.

A continuación se contrastan los resultados obtenidos con los reportados en trabajos recientes sobre detección de malezas con imágenes UAV. Estas comparaciones son indicativas, ya que las diferencias en formulación de la tarea, número de clases, resolución espacial y condiciones de adquisición impiden una comparación directa.

En (J. Wu y cols., 2025) se llevó a cabo una detección binaria de la maleza en cultivos de trigo sarraceno, agrupando todas sus instancias en una sola clase e identificándolas con segmentación semántica. Con esta metodología, el mejor resultado obtenido con una arquitectura de U-Net modificada reportó un mIoU de $0,65$, el cual es menor que el mIoU obtenido en este trabajo con U-Net (encoder ResNet18, entrada RGB + NIR): $0,8417$ en promedio entre clases, y $0,9653$, $0,8143$ y $0,7456$ para dentada, plántula y roseta respectivamente.

En la Figura 5.5 se observan tres ejemplos de imágenes RGB del dataset utilizado en (J. Wu y cols., 2025) con las predicciones obtenidas por el mejor modelo con la combinación de entrada RGB + NIR.

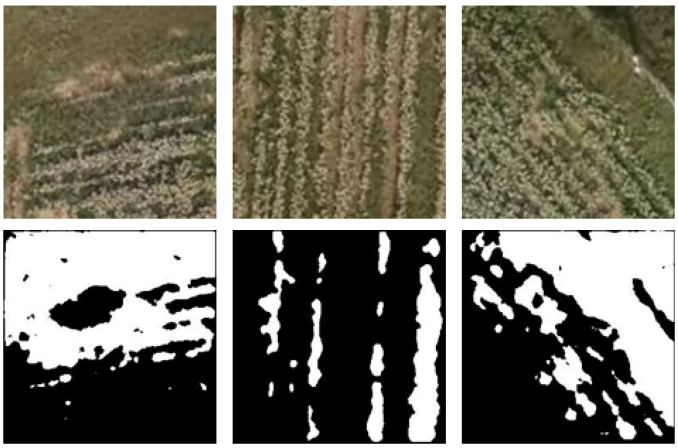


Figura 5.5: Ejemplos de imágenes del trabajo de (J. Wu y cols., 2025). (solo canales RGB, con fines visuales) junto con las inferencias obtenidas con la arquitectura MSU-Net y las bandas RGB + NIR. Tomada de (J. Wu y cols., 2025).

En (Celikkan y cols., 2025) se construye un dataset público de detección de malezas en cultivos de maíz con imágenes tomadas con UAV. Se definieron 6 clases, una para el cultivo, 4 para diferentes especies de malezas y una de background. Con la arquitectura DeepLabV3 con backbone ResNet50 obtienen un mIoU de $0,8294$, incluyendo tanto el IoU del background como el del cultivo. Considerando tan solo las clases de malezas el IoU reportado es $0,7731$. Estos valores de IoU son similares a los obtenidos en este trabajo con U-Net (encoder ResNet18, entrada RGB + NIR) tanto en promedio entre clases como por cada una en particular.

En la Figura 5.6 se observan dos imágenes RGB pertenecientes al dataset utilizado en (Celikkan y cols., 2025) con sus respectivas etiquetas.

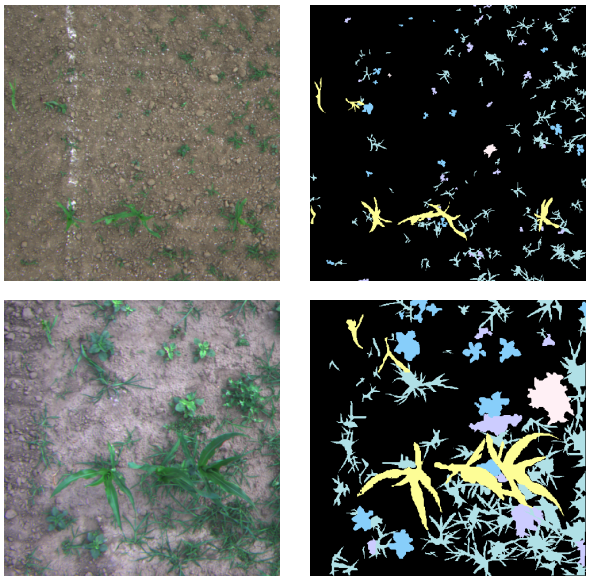


Figura 5.6: Ejemplos de imágenes RGB (izquierda) y sus correspondientes etiquetas semánticas (derecha) del dataset de (Celikkan y cols., 2025). Tomada de (Celikkan y cols., 2025).

En otro caso reciente, en (Wang, Wang, Ibrahim, Severtson, y Mian, 2026) se construye un dataset para la identificación de malezas en dos cultivos de cebada en Australia. En este trabajo se definieron tres clases: cultivo, maleza y background. Reportaron un mIoU máximo de $0,756$, promediando entre todas las clases, y un IoU de $0,635$ para la clase maleza, utilizando una arquitectura propia que supera los resultados que obtuvieron utilizando U-Net. Tanto en mIoU como en IoU para las clases de maleza, las métricas obtenidas en este trabajo con U-Net (encoder ResNet18, entrada RGB + NIR) son numéricamente mayores. Sin embargo, se ha de señalar que en (Wang y cols., 2026) se utilizaron imágenes con un GSD de 6,00 cm/px, considerablemente más alto que el GSD de 0,40 cm/px de las imágenes utilizadas en este proyecto, lo que limita la comparabilidad directa entre ambos trabajos.

En la Figura 5.7 se observa una comparación entre el ground truth y la predicción de una imagen del dataset de (Wang y cols., 2026).

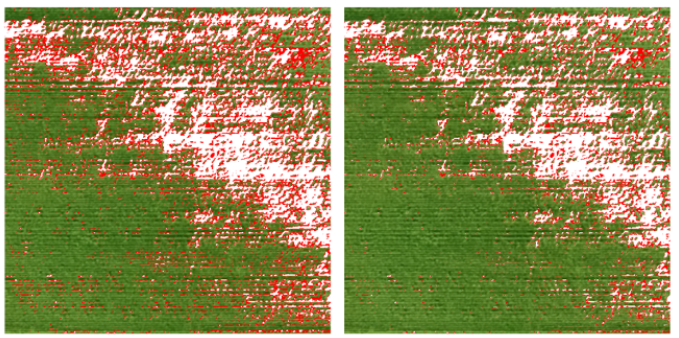


Figura 5.7: Ejemplo de ground truth (izquierda) y predicción (derecha) de una imagen del dataset de (Wang y cols., 2026). Verde indica cultivo, rojo indica maleza y blanco indica background. Tomada de (Wang y cols., 2026).

En un caso de segmentación de instancias, en (Joy y cols., 2025) se utilizó YOLOv8 sobre un dataset de malezas en un cultivo de remolachas azucareras, definiendo tan solo dos clases: cultivo y maleza. Utilizando el modelo YOLOv8-s-seg lograron un Mask mAP50 de $0,728$ frente al $0,919$ obtenido en este trabajo con YOLOv8-s-seg y entrada RGB + NIR, valor que es además inferior a cada uno de los Mask mAP50 por clase: $0,9061$ para dentadas, $0,8965$ para plántulas y $0,9534$ para rosetas (Joy y cols., 2025). No obstante, las diferencias en cultivo (remolacha azucarera vs. vid), número de clases (2 vs. 3) y resolución de las imágenes limitan la comparabilidad directa de estos valores.

En la Tabla 5.8 se resumen las comparaciones realizadas, donde cada entrada corresponde a un trabajo o a una línea experimental de este proyecto y se reporta el cultivo, número de clases (sin contar la clase fondo/background), arquitectura utilizada y métricas de comparación.

Trabajo	**Cultivo**	**Clases**	**Arquitectura**	**Métrica**
(J. Wu y cols., 2025)	Trigo sarraceno	1	MSU-Net	mIoU = 0,650
(Celikkan y cols., 2025)	Maíz	6	DeepLabV3 (ResNet50)	mIoU = 0,829
(Wang y cols., 2026)	Cebada	3	Propia (VISA)	mIoU = 0,756
(Joy y cols., 2025)	Remolacha	2	YOLOv8-s-seg	mAP50 = 0,728
Este trabajo	Vid	3	U-Net (ResNet18)	mIoU = 0,842
Este trabajo	Vid	3	YOLOv8-s-seg	mAP50 = 0,919

Tabla 5.8: Comparación con trabajos recientes de detección/segmentación de malezas con UAV.

Capítulo 6

Conclusiones y Trabajo Futuro

6.1. Conclusiones

El presente trabajo se desarrolló en el marco de un proyecto de investigación del grupo MINA de la Facultad de Ingeniería de la Universidad de la República, y abordó dos objetivos centrales: la construcción de un dataset multiespectral etiquetado para la detección de malezas del género *Conyza* en cultivos nacionales, generando además una guía reproducible para la futura construcción de datasets similares, y la evaluación experimental del aporte de distintas configuraciones espectrales al desempeño de modelos de aprendizaje profundo en tareas de detección y segmentación de malezas.

Respecto del primer objetivo, se construyó un dataset compuesto por 500 imágenes multiespectrales capturadas con una cámara MicaSense RedEdge-P montada sobre un dron DJI Matrice 350 RTK, en un cultivo de vid cercano a Empalme Olmos, Canelones. El dataset incluye 6 bandas espectrales (B, G, R, RE, NIR, P) y el índice NDVI procesadas mediante un pipeline de corrección radiométrica, alineación interbanda basada en homografías estimadas por SIFT y RANSAC, y pansharpening con el algoritmo SFIM (Bocchiardo, Stefanoli, y González, 2026b). En total se anotaron 6 459 instancias de maleza distribuidas en tres clases morfológicas: *roseta* (55, 23 %), *plántula* (40, 24 %) y *dentada* (4, 54 %), mediante un etiquetado de segmentación de instancias realizado en dos etapas con validación cruzada entre anotadores y resolución conjunta de casos ambiguos. La alineación entre bandas permitió que un único conjunto de etiquetas, definido sobre la representación RGB, fuera reutilizado de manera consistente en todas las configuraciones espectrales evaluadas. Además, a partir de estas etiquetas se crearon máscaras utilizables para tareas de segmentación semántica.

El etiquetado se restringió a estadios tempranos (plántula y roseta), excluyendo plantas en estados más avanzados de desarrollo. Esta decisión responde a

que la eficacia del control químico de *Conyza* disminuye significativamente una vez que las plantas elongan su tallo.

Respecto del segundo objetivo, se llevaron a cabo dos líneas experimentales: segmentación de instancias con YOLOv8-s-seg y segmentación semántica con U-Net (encoder ResNet18). En ambas líneas se evaluaron cinco combinaciones de entrada: RGB, NDVI, RGB + NDVI, RGB + NIR y RGB + NIR + RE, todas bajo un esquema de comparación controlada en el que se varió exclusivamente la representación espectral mientras se mantuvo constante el resto del pipeline.

La combinación RGB + NIR obtuvo el mejor desempeño global en ambas tareas. En segmentación de instancias alcanzó un Mask mAP50-95 de 0,598 frente a 0,570 de la línea base RGB. En segmentación semántica, obtuvo un Dice de 0,8697 y un IoU de 0,8417 frente a 0,8635 y 0,8349 respectivamente para RGB. Estos resultados indican que la incorporación de la banda NIR aporta información discriminativa efectiva para la separación entre maleza y fondo en las condiciones del dataset. Esto, a su vez, sugiere que utilizar toda la información multiespectral no garantiza mejores resultados, puesto que los resultados fueron mejores que los obtenidos con la combinación RGB + NIR + RE.

El análisis por clase reveló que el beneficio de las imágenes multiespectrales no es uniforme. Por ejemplo, en segmentación de instancias la clase dentada fue la que más se benefició de la incorporación de NIR, mientras que en segmentación semántica fue la clase plántula la que mostró la mayor mejora con esa misma banda. Esta asimetría sugiere que el aporte de la información espectral depende no solo de las características morfológicas y espectrales de cada clase, sino también de la manera en que cada arquitectura hace uso de dicha información.

En cuanto a la conveniencia de emplear cámaras multiespectrales, esta debe ponderarse frente a su costo. Si bien la incorporación de la banda NIR mejora el desempeño respecto de una representación RGB, la mejora es marginal en términos absolutos y debe contrastarse con el costo de adquisición de una cámara multiespectral, considerablemente superior al de una RGB convencional. Por ello, el uso de sensores multiespectrales no resulta universalmente justificado, sino que depende de la aplicación y de qué malezas o estadios fenológicos se busca detectar, dado que el aporte de la información espectral no es uniforme entre las clases definidas en este proyecto.

En síntesis, este trabajo realiza tres contribuciones principales. En primer lugar, se construye y publica un dataset multiespectral etiquetado para la detección de *Conyza* en condiciones reales de cultivo en Uruguay, un recurso del que no se dispone en la literatura previa para esta especie y región. En segundo lugar, se documenta un pipeline de preprocesamiento reproducible (Bocchiardo, Stefanoli, y González, 2026a), integrando alineación, calibración radiométrica y pansharpening en un flujo unificado. En tercer lugar, se aporta evidencia empírica de que el dataset permite entrenar modelos de segmentación semántica y de instancias con desempeño competitivo. Además, la incorporación de la banda NIR sobre una representación RGB mejora los resultados en dos de las tres arquitecturas evaluadas, mientras que la incorporación de canales multiespectrales adicionales no se traduce en mejoras consistentes en el desempeño de los

modelos.

El dataset, el código del pipeline y los procedimientos de entrenamiento e inferencia se publican de forma abierta. El Apéndice E describe el uso del repositorio y el flujo de trabajo para reproducir los experimentos.

6.2. Limitaciones del estudio

El alcance de los resultados obtenidos está sujeto a varias restricciones que deben explicitarse.

En primer lugar, el dataset fue construido a partir de capturas realizadas en una única fecha (29 de septiembre de 2025), en un solo lote de vid, en una localidad específica del departamento de Canelones. Esto limita la variabilidad temporal, espacial y agronómica representada en el conjunto de datos. No se dispone de evidencia sobre el comportamiento de los modelos entrenados frente a capturas en diferentes estadios fenológicos del cultivo, condiciones de iluminación distintas, otros cultivos o diferentes regiones geográficas.

En segundo lugar, las tres clases definidas, *plántula*, *roseta* y *dentada*, responden a criterios visuales establecidos durante el etiquetado que fueron sustentados con información disponible sobre el género *Conyza* spp. No obstante, sus distinciones no fueron validadas rigurosamente por botánicos especialistas en malezas.

Por último, dado que las imágenes presentan un solapamiento del 75 % entre sí, es posible que exista contaminación entre los subconjuntos de entrenamiento, validación y prueba.

6.3. Trabajo futuro

A partir de las limitaciones identificadas y los resultados obtenidos, se proponen las siguientes líneas de trabajo futuro.

En cuanto a la experimentación sobre el dataset, sería pertinente evaluar configuraciones espectrales adicionales, incluyendo otros índices de vegetación como NDRE, GNDVI o SAVI, así como combinaciones que no incluyan las bandas RGB, tal como se lleva a cabo en trabajos similares (Sa, Popović, y cols., 2018). También resultaría relevante explorar otras arquitecturas, tales como versiones más recientes de YOLO, así como modificaciones específicas aplicadas a la arquitectura U-Net (J. Wu y cols., 2025). Además, podrían considerarse enfoques multimodales que integren los canales no visibles al ojo humano junto con índices de vegetación, con el fin de explotar en mayor medida la información multiespectral (Castellano y cols., 2023).

Respecto de la construcción de futuros datasets, se recomienda incorporar capturas de múltiples fechas, lotes y cultivos para evaluar la capacidad de generalización de los modelos entrenados. Asimismo, sería valioso ampliar la cobertura a otras especies de maleza y estadios fenológicos más avanzados o más tempranos que los representados actualmente. En este sentido, sería conveniente

buscar cooperación con expertos en el ámbito agronómico con el fin tanto de mejorar la identificación de malezas en el proceso de etiquetado como de entender cómo maximizar la utilidad de los modelos entrenados en aplicaciones reales. Además, al momento de obtener las capturas se recomienda tomar una o más capturas del panel de calibración u otra herramienta disponible con el objetivo de llevar a cabo una corrección radiométrica completa, minimizando la influencia de las condiciones ambientales.

Otra línea de trabajo consiste en generar ortomosaicos a partir de las capturas individuales. Al constituir una representación georreferenciada y continua del lote, habilitaría la generación de mapas de distribución de malezas, de utilidad directa para el control localizado. No obstante, el mosaicado introduce artefactos geométricos y radiométricos y puede reducir la resolución espacial efectiva, por lo que convendría evaluar su impacto sobre la detección de instancias de menor tamaño, como las del estadio de plántula.

Referencias

Akyon, F. C., Altinuc, S. O., y Temizel, A. (2022). Slicing aided hyper inference and fine-tuning for small object detection. En *2022 ieee international conference on image processing (icip)* (pp. 966–970). doi: 10.1109/ICIP46576.2022.9897990

Amidi, A., y Amidi, S. (2018). *Convolutional neural networks cheatsheet.* https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks. (CS 230 - Deep Learning, Stanford University)

Ansel, J., Yang, E., He, H., Gimelshein, N., Jain, A., Voznesensky, M., ... Chintala, S. (2024, abril). PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. En *29th acm international conference on architectural support for programming languages and operating systems, volume 2 (asplos '24)*. ACM. Descargado de https://docs.pytorch.org/assets/pytorch2-2.pdf doi: 10.1145/3620665.3640366

Badgujar, C. M., Poulose, A., y Gan, H. (2024). Agricultural object detection with you only look once (yolo) algorithm: A bibliometric and systematic literature review. *Computers and Electronics in Agriculture*, *223*, 109090. Descargado de https://www.sciencedirect.com/science/article/pii/S0168169924004812 doi: https://doi.org/10.1016/j.compag.2024.109090

Biewald, L. (2020). *Experiment tracking with weights and biases.* Descargado de https://www.wandb.com/ (Software available from wandb.com)

Blue Skies Drone Shop. (s.f.). *Micasense rededge-p dls 2.* https://www.blueskiesdroneshop.com/cdn/shop/products/RedEdge-P_DLS2.png?v=1635268384. (Consultado el 25 de abril de 2026)

Bocchiardo, D. M., Stefanoli, G. C., y González, C. (2026a). *Guía reproducible para la construcción de datasets multiespectrales de detección de malezas adquiridos con UAV* (Reporte técnico). Montevideo, Uruguay: Facultad de Ingeniería, Universidad de la República. Descargado de https://github.com/cristiangdev/weed-detection-dataset-pipeline/blob/main/PIPELINE.md (Grupo MINA)

Bocchiardo, D. M., Stefanoli, G. C., y González, C. (2026b). *Identificacción de malezas mediante el procesamiento de imágenes multiespectrales.* Software, repositorio en GitHub. Descargado de https://github.com/

cristiangdev/weed-detection-dataset-pipeline (Licencia CC-BY-4.0)

Candiago, S., Remondino, F., De Giglio, M., Dubbini, M., y Gattelli, M. (2015). Evaluating multispectral images and vegetation indices for precision farming applications from uav images. *Remote Sensing*, *7*(4), 4026–4047. doi: 10.3390/rs70404026

Castellano, G., De Marinis, P., y Vessio, G. (2023). Weed mapping in multispectral drone imagery using lightweight vision transformers. *Neurocomputing*, *562*, 126914. Descargado de https://www.sciencedirect.com/science/article/pii/S0925231223010378 doi: https://doi.org/10.1016/j.neucom.2023.126914

Celikkan, E., Kunzmann, T., Yeskaliyev, Y., Itzerott, S., Klein, N., y Herold, M. (2025). Weedsgalore: A multispectral and multitemporal UAV-based dataset for crop and weed segmentation in agricultural maize fields. En *Proceedings of the IEEE/CVF winter conference on applications of computer vision (WACV)* (pp. 4767–4777). doi: 10.1109/WACV61041.2025.00467

Delegido, J., Verrelst, J., Alonso, L., y Moreno, J. (2011). Evaluation of sentinel-2 red-edge bands for empirical estimation of green lai and chlorophyll content. *Sensors*, *11*(7), 7063–7081. doi: 10.3390/s110707063

Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, *26*(3), 297–302. doi: 10.2307/1932409

EOS Data Analytics. (2021, 24 de 9). *Panchromatic and pansharpened satellite imagery.* Descargado 2025-12-07, de https://eos.com/make-an-analysis/panchromatic/ (Last updated: 06.11.2023)

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., y Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, *88*(2), 303–338. doi: 10.1007/s11263-009-0275-4

Gallagher, J., y Oughton, E. (2025, 01). Surveying you only look once (yolo) multispectral object detection advancements, applications, and challenges. *IEEE Access*, *PP*, 1-1. doi: 10.1109/ACCESS.2025.3526458

Geng, T., Yu, H., Yuan, X., Ma, R., y Li, P. (2024). Research on segmentation method of maize seedling plant instances based on uav multispectral remote sensing images. *Plants*, *13*(13). Descargado de https://www.mdpi.com/2223-7747/13/13/1842 doi: 10.3390/plants13131842

Gerhards, R., Vailati-Riboni, M., Christensen, S., Santin-Montanyá, M. I., Kudsk, P., Otto, S., ... others (2022). Advances and challenges in site-specific weed management. *Weed Research*, *62*(6), 417–433. doi: 10.1111/wre.12578

GIS Geography. (2025, junio). *What is pansharpening in remote sensing.* Descargado 2026-01-14, de https://gisgeography.com/pansharpening/ (Accessed: 2026-01-14)

Gitelson, A. A., Kaufman, Y. J., y Merzlyak, M. N. (1996). Use of a green channel in remote sensing of global vegetation from eos-modis. *Remote Sensing of Environment*, *58*(3), 289–298. doi: 10.1016/S0034-4257(96)00072-7

Goel, D., Kapur, B., y Vuppuluri, P. (2024, 08). Machine learning interventions for weed detection using multispectral imagery and unmanned aerial vehicles – a systematic review. *arXiv*. doi: 10.48550/arXiv.2408.06727

Guizar-Sicairos, M., Thurman, S. T., y Fienup, J. R. (2008, enero). Efficient subpixel image registration algorithms. *Optics Letters*, *33*(2), 156. doi: 10.1364/OL.33.000156

Guo, Y., Senthilnath, J., Wu, W., Zhang, X., Zeng, Z., y Huang, H. (2019). Radiometric calibration for multispectral camera of different imaging conditions mounted on a uav platform. *Sustainability*, *11*(4). Descargado de https://www.mdpi.com/2071-1050/11/4/978 doi: 10.3390/su11040978

Hassanpour, M., Dadras Javan, F., y Azizi, A. (2019). Band to band registration of multi-spectral aerial imagery – relief displacement and miss-registration error. *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLII-4/W18*, 467–472. doi: 10.5194/isprs-archives-XLII-4-W18-467-2019

He, K., Gkioxari, G., Dollár, P., y Girshick, R. (2017). Mask R-CNN. En *Proceedings of the ieee international conference on computer vision (ICCV)* (pp. 2961–2969). doi: 10.1109/ICCV.2017.322

He, K., Zhang, X., Ren, S., y Sun, J. (2016). Deep residual learning for image recognition. En *Proceedings of the ieee conference on computer vision and pattern recognition (cvpr)* (pp. 770–778). IEEE. doi: 10.1109/CVPR.2016.90

Horler, D. N. H., Dockray, M., Barber, J., y Barringer, A. R. (1983). Red edge measurements for remotely sensing plant chlorophyll content. *Advances in Space Research*, *3*(2), 273–277. doi: 10.1016/0273-1177(83)90130-8

Hossein-nejad, Z., y Nasri, M. (2016). Image registration based on sift features and adaptive ransac transform. En *2016 international conference on communication and signal processing (iccsp)* (p. 1087-1091). doi: 10.1109/ICCSP.2016.7754318

Huete, A. R. (1988). A soil-adjusted vegetation index (SAVI). *Remote Sensing of Environment*, *25*(3), 295–309. doi: 10.1016/0034-4257(88)90106-X

Iakubovskii, P. (2019). *Segmentation models pytorch.* https://github.com/qubvel/segmentation_models.pytorch. GitHub.

Irigoyen, A., y Perrachón, J. (2012). Yerba carnicera (*Conyza bonaerensis*): aumenta su presencia en praderas y cultivos en el uruguay. *Revista Plan Agropecuario*(142).

Jackson, R. D., y Huete, A. R. (1991). Interpreting vegetation indices. *Preventive Veterinary Medicine*, *11*(3), 185-200. Descargado de https://www.sciencedirect.com/science/article/pii/S0167587705800042 doi: https://doi.org/10.1016/S0167-5877(05)80004-2

Jain, R., Mishra, A., Arya, Chaudhary, A., y Yadav, M. (2025, diciembre). Deep Transfer Learning for Forest Classification Using Multispectral Satellite Imagery. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *1052*, 263-270. doi: 10.5194/isprs-annals-X-5-W2-2025-263-2025

Jiang, Z., Huete, A. R., Chen, J., Chen, Y., Li, J., Yan, G., y Zhang, X. (2006). Analysis of ndvi and scaled difference vegetation index retrievals of vegetation fraction. *Remote Sensing of Environment*, *101*(3), 366–378. doi: 10.1016/j.rse.2006.01.003

Jocher, G., Qiu, J., y Chaurasia, A. (2023, enero). *Ultralytics YOLO.* Descargado de https://github.com/ultralytics/ultralytics

Joy, J., Kelvin, B., Howatt, K., Aderholdt, W., Khan, M., Peters, T., y Sun, X. (2025). Edge-deployable segmentation and prescription mapping of post-emergence weeds in sugar beet crops for uav-based precision spraying. *Journal of Agriculture and Food Research*, *24*, 102422. Descargado de https://www.sciencedirect.com/science/article/pii/S2666154325007938 doi: https://doi.org/10.1016/j.jafr.2025.102422

Kaspary, T., Waller, M., García, M., Fernández, E., García, A., y Cabrera, M. (2024, septiembre). Conyzas resistentes a herbicidas: estado actual y opciones de manejo. *Revista INIA*(78). (Edición Setiembre 2024)

Kirillov, A., He, K., Girshick, R., Rother, C., y Doll'ar, P. (2019). Panoptic segmentation. En *Proceedings of the ieee/cvf conference on computer vision and pattern recognition (CVPR)* (pp. 9404–9413). doi: 10.1109/CVPR.2019.00963

Knipling, E. B. (1970). Physical and physiological basis for the reflectance of visible and near-infrared radiation from vegetation. *Remote Sensing of Environment*, *1*(3), 155–159. doi: 10.1016/S0034-4257(70)80021-9

Krizhevsky, A., Sutskever, I., y Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. En *Advances in neural information processing systems 25 (NeurIPS)* (pp. 1097–1105).

LeCun, Y., Bottou, L., Bengio, Y., y Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324. doi: 10.1109/5.726791

Liang, S. (2004). *Quantitative remote sensing of land surfaces.* Hoboken, NJ: John Wiley & Sons.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... Zitnick, C. L. (2014). Microsoft coco: Common objects in context. En *Computer vision – eccv 2014* (pp. 740–755). Springer.

Liu, K., Fu, Z., Jin, S., Chen, Z., Zhou, F., Jiang, R., ... Ye, J. (2024). Esod: Efficient small object detection on high-resolution images. *IEEE Transactions on Image Processing*. doi: 10.1109/TIP.2024.3501853

Long, J., Shelhamer, E., y Darrell, T. (2015). Fully convolutional networks for semantic segmentation. En *Proceedings of the ieee conference on computer vision and pattern recognition (CVPR)* (pp. 3431–3440). doi: 10.1109/CVPR.2015.7298965

Lowe, D. (1999). Object recognition from local scale-invariant features. En *Proceedings of the seventh ieee international conference on computer vision* (Vol. 2, p. 1150-1157 vol.2). doi: 10.1109/ICCV.1999.790410

López-Granados, F. (2011). Weed detection for site-specific weed management: mapping and real-time approaches. *Weed Research*, *51*(1), 1–11. doi: 10.1111/j.1365-3180.2010.00829.x

MicaSense. (2016). *Use of calibrated reflectance panels for MicaSense data.* https://support.micasense.com/hc/en-us/articles/115000765514-Use-of-Calibrated-Reflectance-Panels-For-MicaSense-Data. (Accessed: 2026-03-01)

MicaSense. (2018). *Radiometric calibration model for micasense sensors.* https://support.micasense.com/hc/en-us/articles/115000351194-Radiometric-Calibration-Model-for-MicaSense-Sensors. (Accessed: 2025-12-02)

MicaSense. (2021). *Rededge-p integration guide.* https://support.micasense.com/hc/en-us/articles/4410824602903-RedEdge-P-Integration-Guide. (Accessed: 2026-02-28)

MicaSense. (2024a). *How to use the calibrated reflectance panel (CRP).* https://support.micasense.com/hc/en-us/articles/1500001975662-How-to-use-the-Calibrated-Reflectance-Panel-CRP. (Accessed: 2026-03-01)

MicaSense. (2024b). *Micasense image processing.* https://github.com/micasense/imageprocessing. GitHub. (Accessed: 2025-09-04)

MicaSense. (2024c). *Why are the images from the camera not aligned to each other?* https://support.micasense.com/hc/en-us/articles/215173147-Why-are-the-images-from-the-camera-not-aligned-to-each-other. (Accessed: 2025-12-10)

Minaee, S., Boykov, Y., Porikli, F., Plaza, A. J., Kehtarnavaz, N., y Terzopoulos, D. (2022). Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(7), 3523–3542. doi: 10.1109/TPAMI.2021.3059968

Miranda, R. D. (2015). *Control de carniceras (Conyza sumatrensis y Conyza bonariensis) en pre y post cosecha de soja* (Tesis de grado). Facultad de Agronomía, Universidad de la República, Montevideo, Uruguay.

Nesmachnow, S., y Iturriaga, S. (2019). Cluster-uy: Collaborative scientific high performance computing in uruguay. En M. Torres y J. Klapp (Eds.), *Supercomputing* (pp. 188–202). Cham: Springer International Publishing.

Parven, A., Md Meftaul, I., Venkateswarlu, K., y Megharaj, M. (2024). Herbicides in modern sustainable agriculture: environmental fate, ecological implications, and human health concerns. *International Journal of Environmental Science and Technology*, *22*, 1181–1202. doi: 10.1007/s13762-024-05818-y

Pix4D. (2025). *Radiometric correction - pix4dfields.* https://support.pix4d.com/hc/en-us/articles/360022919691. (Accessed: 2026-03-04)

Qu, H.-R., y Su, W.-H. (2024, 02). Deep learning-based weed–crop recognition for smart agricultural equipment: A review. *Agronomy*, *14*, 363. doi: 10.3390/agronomy14020363

Redmon, J., Divvala, S., Girshick, R., y Farhadi, A. (2016). You only look once: Unified, real-time object detection. En *Proceedings of the ieee conference on computer vision and pattern recognition (cvpr)* (pp. 779–788). doi: 10.1109/CVPR.2016.91

Redmon, J., y Farhadi, A. (2017). Yolo9000: Better, faster, stronger. En *Proceedings of the ieee conference on computer vision and pattern recognition (cvpr)* (pp. 7263–7271). doi: 10.1109/CVPR.2017.690

Rey Otero, I., y Delbracio, M. (2014). Anatomy of the SIFT Method. *Image Processing On Line*, *4*, 370–396. (https://doi.org/10.5201/ipol.2014.82)

Rippercorp. (s.f.). *Dji matrice 350 rtk*. https://rippercorp.com/cdn/shop/files/431__1.png?v=1738210100&width=3840. (Consultado el 25 de abril de 2026)

Rise, B., Uney, M., y Huang, X. (2026). Two-stage transfer learning for airborne multi-spectral image classifiers. *Signal Processing*, *240*, 110358. Descargado de https://www.sciencedirect.com/science/article/pii/S0165168425004748 doi: https://doi.org/10.1016/j.sigpro.2025.110358

Ronneberger, O., Fischer, P., y Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. En *Medical image computing and computer-assisted intervention – miccai 2015* (Vol. 9351, pp. 234–241). Cham: Springer International Publishing. doi: 10.1007/978-3-319-24574-4_28

Sa, I., Chen, Z., Popović, M., Khanna, R., Liebisch, F., Nieto, J., y Siegwart, R. (2018). weednet: Dense semantic weed classification using multispectral images and mav for smart farming. *IEEE Robotics and Automation Letters*, *3*(1), 588-595. doi: 10.1109/LRA.2017.2774979

Sa, I., Popović, M., Khanna, R., Chen, Z., Lottes, P., Liebisch, F., ... Siegwart, R. (2018). Weedmap: A large-scale semantic weed mapping framework using aerial multispectral imaging and deep neural network for precision farming. *Remote Sensing*, *10*(9). Descargado de https://www.mdpi.com/2072-4292/10/9/1423 doi: 10.3390/rs10091423

Schaepman-Strub, G., Schaepman, M., Painter, T., Dangel, S., y Martonchik, J. (2006). Reflectance quantities in optical remote sensing—definitions and case studies. *Remote Sensing of Environment*, *103*(1), 27-42. Descargado de https://www.sciencedirect.com/science/article/pii/S0034425706001167 doi: https://doi.org/10.1016/j.rse.2006.03.002

Sebastián López, M., Palomo Arroyo, M., Rincón Ramírez, J. A., Ormeño Villajos, S., y Vicent García, J. M. (2013). Métodos de documentación, análisis y conservación no invasivos para el arte rupestre postpaleolítico: radiometría de campo e imágenes multiespectrales. ensayos en la cueva del tío garroso (alacón, teruel). En *La ciencia y el arte iv: Ciencias experimentales y conservación del patrimonio* (pp. 279–287). Madrid: Ministerio de Educación, Cultura y Deporte.

Sokolova, M., y Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427–437. doi: 10.1016/j.ipm.2009.03.002

Studholme, C., Hill, D., y Hawkes, D. (1999). An overlap invariant entropy measure of 3d medical image alignment. *Pattern Recognition*, *32*(1), 71-86. Descargado de https://www.sciencedirect.com/science/article/pii/S0031320398000910 doi: https://doi.org/10

.1016/S0031-3203(98)00091-0

Tan, M., y Le, Q. V. (2020). *Efficientnet: Rethinking model scaling for convolutional neural networks.* Descargado de https://arxiv.org/abs/1905.11946

Tessore, F. (2025). *Detección y conteo de flores de manzano.* Proyecto de grado, Facultad de Ingeniería, Universidad de la República. (Documento interno)

Tilse, A., Siegmann, B., Börner, A., Menz, G., y Münz, K. (2025). Multispectral camera systems in agriculture: Correcting for soil effects in ndre for robust chlorophyll estimation. *Precision Agriculture*, *26*. (Article number 78) doi: 10.1007/s11119-025-10267-9

Tucker, C. J. (1979). Red and photographic infrared linear combinations for monitoring vegetation. *Remote Sensing of Environment*, *8*(2), 127–150. doi: 10.1016/0034-4257(79)90013-0

Uruguay XXI. (2024). *Informe agrícola — noviembre 2024* (Inf. Téc.). Uruguay XXI. Descargado de https://www.uruguayxxi.gub.uy/uploads/informacion/3c65f7d9c47fd2235bbc5752bd3ad5c2f005ec88.pdf (Informe consultado 07 diciembre 2025)

van der Walt, S. J., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., ... the scikit-image contributors (2014, junio). scikit-image: image processing in Python. *PeerJ*, *2*, e453. Descargado de https://doi.org/10.7717/peerj.453 doi: 10.7717/peerj.453

van Rijsbergen, C. J. (1979). *Information retrieval* (2.ª ed.). London: Butterworths.

Wang, H., Ibrahim, M., Miao, Y., Severtson, D., Mansoor, A., y Mian, A. S. (2024). Multispectral remote sensing for weed detection in west australian agricultural lands. En *2024 international conference on digital image computing: Techniques and applications (dicta)* (p. 624-631). doi: 10.1109/DICTA63115.2024.00095

Wang, H., Wang, X., Ibrahim, M., Severtson, D., y Mian, A. (2026). Bawseg: A uav multispectral benchmark for barley weed segmentation. *Remote Sensing*, *18*(6). Descargado de https://www.mdpi.com/2072-4292/18/6/915 doi: 10.3390/rs18060915

Wu, J., Wu, X., y Miao, R. (2025). Research on buckwheat weed recognition in multispectral uav images based on msu-net. *Agriculture*, *15*(14), 1471. Descargado de https://doi.org/10.3390/agriculture15141471 doi: 10.3390/agriculture15141471

Wu, Z., Chen, Y., Zhao, B., Kang, X., y Ding, Y. (2021, 05). Review of weed detection methods based on computer vision. *Sensors*, *21*, 3647. doi: 10.3390/s21113647

Yu, Y., Wang, C., Fu, Q., Kou, R., Huang, F., Yang, B., ... Gao, M. (2023). Techniques and challenges of image segmentation: A review. *Electronics*, *12*(5), 1199. Descargado de https://www.mdpi.com/2079-9292/12/5/1199 doi: 10.3390/electronics12051199

Zhang, Z., Hu, Q., Fang, H., Liu, W., Feng, R., Chen, S., ... Lu, W. (2026). Trigefnet: A tri-stream multimodal enhanced fusion network for landslide

segmentation from remote sensing imagery. *Remote Sensing*, *18*(2). Descargado de https://www.mdpi.com/2072-4292/18/2/186 doi: 10.3390/rs18020186

Zitová, B., y Flusser, J. (2003). Image registration methods: a survey. *Image and Vision Computing*, *21*(11), 977-1000. Descargado de https://www.sciencedirect.com/science/article/pii/S0262885603001379 doi: https://doi.org/10.1016/S0262-8856(03)00137-9

Zuccarelli, E. (2020, 29 de diciembre). *Performance metrics in machine learning – part 1: Classification.* https://medium.com/data-science/performance-metrics-in-machine-learning-part-1-classification-6c6b8d8a8c92. (TDS Archive, Medium. Accedido: 27 de abril de 2026)

Anexo A

Fundamentos auxiliares

En este anexo se define y profundiza en conceptos que fueron presentados o mencionados a lo largo de este informe, pero que no fueron explicados en detalle.

A.1. Ground Sampling Distance (GSD)

Ground Sampling Distance (GSD) es la distancia real que existe entre los centroides de dos píxeles contiguos en una imagen. Esto implica que para que un objeto pueda ser representado en la imagen debe tener un tamaño igual o mayor que el GSD.

De manera general, el GSD puede expresarse de la siguiente manera.

$$GSD = \frac{\sqrt{d^2 + h^2} \times p}{f \times \cos(\theta)} \tag{A.1}$$

Siendo:

- h: Distancia vertical del sensor al suelo.
- d: Distancia horizontal del nadir al punto observado.
- p: Tamaño físico de un píxel del sensor (largo o ancho).
- f: Distancia focal del lente.
- θ: Ángulo entre el nadir y el punto observado visto desde el sensor.

En la Figura A.1 se observa una representación de los parámetros del cálculo del GSD, donde se representa un dron que captura imágenes de manera oblicua.

En caso de que se capturen imágenes desde un punto de vista cenital, entonces $d = 0$ y $\theta = 0$. Por ende, se puede simplificar de la siguiente manera.

$$GSD = \frac{h \times p}{f} \tag{A.2}$$

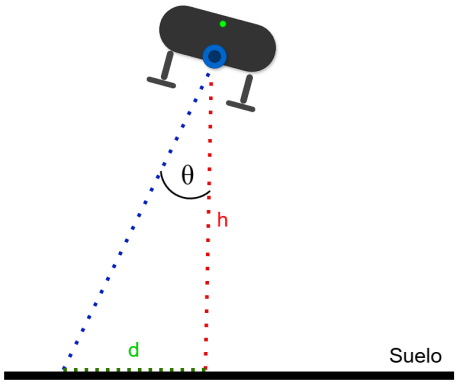


Figura A.1: Diagrama geométrico de los parámetros involucrados en el cálculo del GSD.

A.2. Scale Invariant Feature Transform (SIFT)

El algoritmo SIFT detecta una serie de puntos clave a partir de una representación multiescala de la imagen. Esta representación multiescala consiste en una familia de imágenes cada vez más difuminadas. Cada punto clave es una estructura similar a una mancha cuya posición central (x, y) y escala característica σ se localizan con precisión. SIFT calcula la orientación dominante θ sobre una región que rodea cada uno de estos puntos clave. Para cada punto clave, la cuádrupla (x, y, σ, θ) define el centro, tamaño y orientación de un parche normalizado donde se calcula el descriptor SIFT. Como resultado de esta normalización, los descriptores de puntos clave SIFT son, en teoría, invariantes a cualquier traslación, rotación y cambio de escala. El descriptor codifica la distribución del gradiente espacial alrededor de un punto clave mediante un vector. Este vector de características se utiliza generalmente para emparejar puntos clave extraídos de diferentes imágenes (Rey Otero y Delbracio, 2014).

A.3. Algoritmo RANSAC

El método de RANSAC es utilizado para eliminar emparejamientos atípicos. El algoritmo se basa en muestrear un subconjunto de emparejamientos, calcular un modelo para la transformación geométrica entre las imágenes y evaluar la cantidad de emparejamientos que son consistentes con el modelo utilizando un umbral predefinido. Este proceso se realiza en varias iteraciones, refinando el modelo para maximizar la cantidad de emparejamientos consistentes hasta que se arribe a un máximo o a una condición de parada (Hossein-nejad y Nasri, 2016).

A.4. Pansharpening y algoritmo SFIM

El pansharpening es una técnica creada con el fin de aumentar la resolución espacial de imágenes a partir de una imagen pancromática (GIS Geography, 2025). Las imágenes pancromáticas representan rangos de longitud de onda más amplios que las bandas típicas presentes en las capturas multiespectrales, lo que facilita tener una resolución espacial mayor gracias a que se debe muestrear áreas más pequeñas para capturar la suficiente energía para poder formar la imagen (EOS Data Analytics, 2021).

Existen diversos algoritmos de pansharpening. Uno de ellos es el algoritmo Smoothing Filter-based Intensity Modulation (SFIM), en el cual para cada banda (B_i) se aplica una operación a partir de la banda pancromática (P) y la banda pancromática muestreada a la resolución espacial de la banda (P_{low}).

$$B_i^{pan} = \frac{B_i}{P_{low}} P \tag{A.3}$$

A.5. Correlación cruzada de fase (PCC)

La correlación cruzada de fase estima el desplazamiento espacial entre dos imágenes a partir de la fase de su espectro cruzado (Guizar-Sicairos y cols., 2008). Dadas dos imágenes f y g, su espectro cruzado normalizado se define como:

$$Q(u,v) = \frac{F(u,v) \cdot G^*(u,v)}{|F(u,v) \cdot G^*(u,v)|} \tag{A.4}$$

donde F y G son las transformadas de Fourier de f y g, y G^* denota el conjugado complejo. La transformada inversa de Q produce un pico en la posición correspondiente al desplazamiento $(\Delta x, \Delta y)$ entre ambas imágenes. La precisión subpíxel se obtiene mediante sobremuestreo local de la DFT en un entorno reducido del pico inicial, utilizando multiplicación matricial en lugar de una FFT completa.

En la implementación utilizada, previo al cálculo de la correlación se normaliza cada banda a media cero y varianza unitaria. La normalización previa reduce la influencia de diferencias radiométricas entre bandas, favoreciendo la detección de desplazamientos puramente geométricos.

A.6. Coeficiente de correlación cruzada normalizada (NCC)

El NCC mide la similitud lineal entre dos señales normalizadas (Zitová y Flusser, 2003). Para dos imágenes representadas como vectores **a** y **b**, se define como:

$$\mathrm{NCC}(\mathbf{a}, \mathbf{b}) = \frac{(\mathbf{a} - \bar{a})^\top (\mathbf{b} - \bar{b})}{\|\mathbf{a} - \bar{a}\| \, \|\mathbf{b} - \bar{b}\|} \tag{A.5}$$

donde $\bar{a}$ y $\bar{b}$ son las medias de cada imagen. El NCC toma valores en $[-1, 1]$, donde 1 indica correlación lineal perfecta, 0 indica ausencia de correlación lineal y -1 indica anticorrelación.

A.7. Información mutua normalizada (NMI)

La información mutua cuantifica la cantidad de información compartida entre dos variables aleatorias sin asumir una relación funcional específica entre ellas (Studholme y cols., 1999). Para dos imágenes con distribuciones marginales de entropía $H(A)$ y $H(B)$ y entropía conjunta $H(A, B)$, la información mutua normalizada se define como:

$$\mathrm{NMI}(A, B) = \frac{H(A) + H(B)}{H(A, B)} \tag{A.6}$$

donde $H(\cdot)$ denota la entropía de Shannon. El NMI alcanza su valor mínimo de 1 cuando las imágenes son estadísticamente independientes, y crece a medida que aumenta la dependencia entre ellas. En el contexto de registro de imágenes, un NMI mayor tras la alineación indica que las bandas comparten mayor estructura espacial.

A.8. RMSE basado en SIFT

El RMSE basado en SIFT cuantifica el desalineamiento geométrico residual entre dos bandas mediante el emparejamiento de puntos característicos (Hassanpour, Dadras Javan, y Azizi, 2019). A diferencia de métricas píxel a píxel, esta medida evalúa directamente la correspondencia espacial de estructuras detectadas en ambas imágenes.

El procedimiento es el siguiente: se detectan puntos característicos SIFT (Rey Otero y Delbracio, 2014) en la banda objetivo y en la banda de referencia (RedEdge). Los descriptores se emparejan aplicando Lowe's ratio test (Lowe, 1999) con umbral de 0,8, y las correspondencias atípicas se filtran mediante RANSAC con transformación afín. Para los M pares de puntos inliers resultantes, con coordenadas $\mathbf{p}_k$ en la banda objetivo y $\mathbf{q}_k$ en la referencia, el RMSE se define como:

$$\mathrm{RMSE} = \sqrt{\frac{1}{M} \sum_{k=1}^{M} \|\mathbf{p}_k - \mathbf{q}_k\|^2} \tag{A.7}$$

donde $\|\mathbf{p}_k - \mathbf{q}_k\|$ es la distancia euclídea entre el par de puntos emparejados k. Un valor cercano a 0 px indica alta correspondencia geométrica entre las bandas. Esta métrica es complementaria al desplazamiento estimado por PCC,

ya que captura desalineamientos locales y no uniformes que un desplazamiento global rígido no detecta.

Anexo B

Hiperparámetros y configuración

En este anexo se documentan los hiperparámetros utilizados y las configuraciones de entrenamiento empleadas en los experimentos reportados en el Capítulo 5. En todos los casos, los valores que se presentan corresponden a los efectivamente utilizados durante las corridas cuyos resultados se reportan.

B.1. Generación de recortes (*slicing*)

A partir de las 500 imágenes originales de 2445×1890 píxeles se generaron recortes para el entrenamiento y la evaluación de los modelos. La Tabla B.1 resume los parámetros utilizados en el proceso de recorte.

Parámetro	Valor
Tamaño del recorte	640×640 px
Solapamiento	20 % (stride de 512 px)
Visibilidad mínima de etiqueta	10 %
Recortes totales generados	10 000
Recortes con maleza	4 309
Recortes vacíos incluidos	2 000
Conjunto final	6 309 recortes
Partición (train / val / test)	5 047 / 631 / 631

Tabla B.1: Parámetros de generación de recortes.

El parámetro de visibilidad mínima indica la fracción mínima del área de la caja delimitadora de una etiqueta que debe quedar contenida dentro de un recorte para que dicha etiqueta sea incluida en el recorte. Etiquetas con visibilidad inferior al umbral se descartan del recorte correspondiente.

B.2. YOLOv8-s-seg

La Tabla B.2 presenta los hiperparámetros de entrenamiento utilizados en todos los ensayos de segmentación de instancias con YOLOv8-s-seg. Se utilizó la configuración por defecto provista por la biblioteca Ultralytics, salvo los parámetros explícitamente indicados.

Parámetro	Valor
Modelo base	YOLOv8-s-seg (preentrenado)
Resolución de entrada	640×640 px
Épocas máximas	400
Batch size	8
Optimizador	Auto (SGD / AdamW)
Learning rate inicial (lr_0)	0,01
Learning rate final (lr_f)	0,01
Momentum	0,937
Weight decay	5×10^{-4}
Warmup (épocas)	3
Warmup momentum	0,8
Warmup bias lr	0,1
Early stopping (paciencia)	50 épocas
Mixed precision (AMP)	Sí
Seed	0

Tabla B.2: Hiperparámetros de entrenamiento de YOLOv8-s-seg.

La función de pérdida de YOLOv8-s-seg combina tres componentes con los pesos indicados en la Tabla B.3.

Componente	Peso
Box loss (CIoU)	7,5
Classification loss (BCE)	0,5
Distribution Focal Loss (DFL)	1,5

Tabla B.3: Componentes de la función de pérdida de YOLOv8-s-seg y sus pesos.

En cuanto al aumento de datos, se utilizó la configuración por defecto de Ultralytics para todas las corridas. La Tabla B.4 detalla los parámetros de aumento de datos aplicados.

Parámetro	Valor
Flip horizontal	50 %
Flip vertical	0 %
HSV — tono (H)	0,015
HSV — saturación (S)	0,7
HSV — valor (V)	0,4
Escala	0,5
Traslación	0,1
Mosaic	1,0 (desactivado últimas 10 épocas)
Mixup	0,0
Rotación	0°
Shear	0°

Tabla B.4: Parámetros de aumento de datos de YOLOv8-s-seg (configuración por defecto de Ultralytics).

B.3. U-Net

La Tabla B.5 presenta los hiperparámetros de entrenamiento utilizados en todos los ensayos de segmentación semántica con U-Net. Los ensayos con encoder EfficientNet-b0 utilizaron la misma configuración que los ensayos con ResNet18, con la única diferencia del encoder.

Parámetro	Valor
Arquitectura	U-Net (Segmentation Models PyTorch)
Encoder	ResNet18 / EfficientNet-b0 (preentrenados en ImageNet)
Entrada efectiva	Recortes de 640 × 640 reescalados
Resolución de entrada	224 × 224 px
Clases de salida	4 (background, plántula, roseta, dentada)
Épocas máximas	400
Batch size	8
Optimizador	AdamW
Learning rate	1×10^{-3}
Weight decay	1×10^{-4}
Scheduler	CosineAnnealingLR ($T_{max} = 400$)
Función de pérdida	DiceLoss ($1 - Dice$, multiclass) + CrossEntropyLoss
Early stopping (paciencia)	50 épocas (sobre pérdida de validación)
Mixed precision (AMP)	Sí
Seed	42

Tabla B.5: Hiperparámetros de entrenamiento de U-Net.

En cuanto al aumento de datos, se aplicaron únicamente transformaciones geométricas simples durante el entrenamiento, detalladas en la Tabla B.6. No se aplicaron transformaciones de color ni deformaciones elásticas, con el objetivo

de preservar la fidelidad radiométrica de las bandas multiespectrales.

Parámetro	Valor
Flip horizontal	50 %
Flip vertical	50 %

Tabla B.6: Parámetros de aumento de datos de U-Net.

B.4. Adaptación de la primera capa convolucional

En ambos modelos, la primera capa convolucional fue modificada para aceptar el número de canales correspondiente a cada configuración espectral. En las configuraciones con más de 3 canales, los pesos de los canales RGB se conservaron del preentrenamiento y los canales adicionales se inicializaron con la media de los pesos RGB. En la configuración monocanal (NDVI), los pesos se inicializaron igualmente a partir de dicha media. Esta estrategia busca mantener las representaciones aprendidas durante el preentrenamiento mientras se extiende el modelo a entradas multicanal.

B.5. Normalización de la entrada

Para U-Net, las imágenes de entrada se normalizaron al rango $[0, 1]$. En el caso de imágenes de 8 bits, los valores se dividieron por 255. No se aplicó normalización por media y desviación estándar por canal.

Para YOLOv8, se utilizó la normalización por defecto provista por Ultralytics, que escala los valores de los píxeles al rango $[0, 1]$.

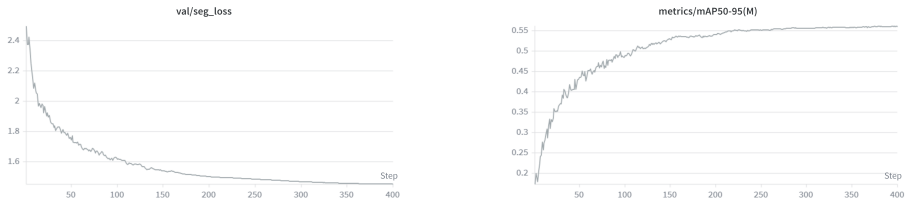
Anexo C

Resultados complementarios

En este anexo se presentan las curvas de entrenamiento correspondientes a los ensayos presentados en este informe, tanto de YOLOv8 como de U-Net.

C.1. Curvas de entrenamiento con YOLOv8

En esta sección se presentan las curvas de entrenamiento correspondientes a los ensayos de segmentación de instancias con YOLOv8-s-seg reportados en el Capítulo 5. Para cada combinación espectral de entrada se muestran la pérdida de validación y el mAP50-95 de las máscaras de predicción sobre el conjunto de validación en función de las épocas de entrenamiento. Estas curvas permiten visualizar la convergencia de los ensayos y complementan las métricas finales presentadas en la Sección 5.4.

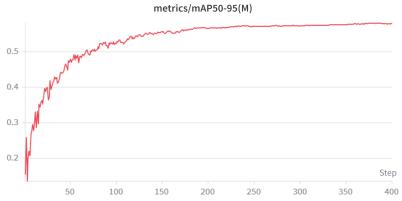


(a) Loss de validación.

(b) Mask mAP50-95 de validación.

Figura C.1: Curvas de entrenamiento de YOLOv8-s-seg con entrada RGB.

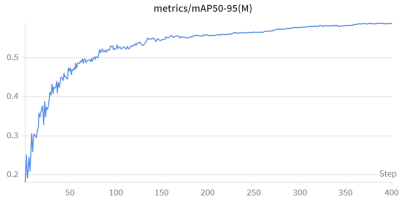
(a) Loss de validación.



(b) Mask mAP50-95 de validación.

Figura C.2: Curvas de entrenamiento de YOLOv8-s-seg con entrada RGB + NIR.

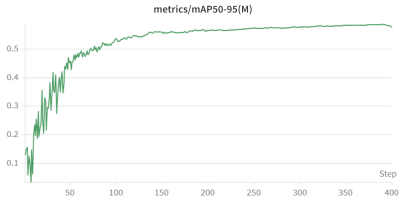
(a) Loss de validación.



(b) Mask mAP50-95 de validación.

Figura C.3: Curvas de entrenamiento de YOLOv8-s-seg con entrada RGB + NIR + RE.

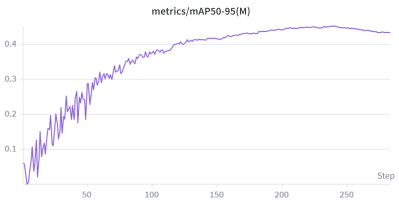
(a) Loss de validación.



(b) Mask mAP50-95 de validación.

Figura C.4: Curvas de entrenamiento de YOLOv8-s-seg con entrada RGB + NDVI.

(a) Loss de validación.

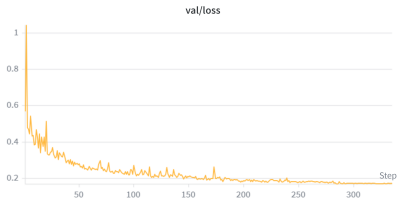


(b) Mask mAP50-95 de validación.

Figura C.5: Curvas de entrenamiento de YOLOv8-s-seg con entrada NDVI.

C.2. Curvas de entrenamiento con U-Net

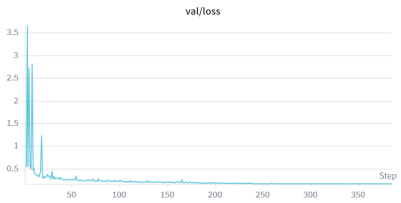
A continuación se presentan las curvas de entrenamiento correspondientes a los ensayos de segmentación semántica con U-Net reportados en el Capítulo 5. Para cada configuración espectral y encoder se muestran la pérdida de validación y el mIoU de validación en función de las épocas de entrenamiento. Estas curvas permiten verificar la convergencia de cada ensayo y complementan las métricas finales presentadas en la Sección 5.5.



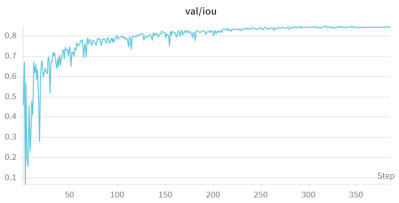
(a) Loss de validación.

(b) mIoU de validación.

Figura C.6: Curvas de entrenamiento de U-Net con entrada RGB y encoder ResNet18.

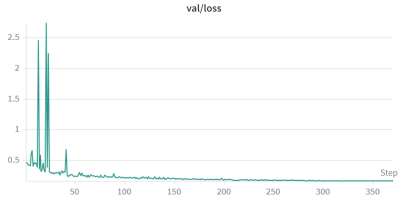


(a) Loss de validación.

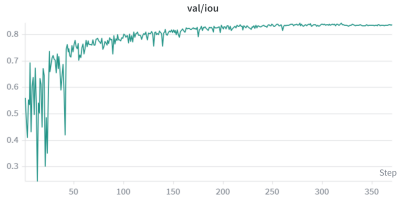


(b) mIoU de validación.

Figura C.7: Curvas de entrenamiento de U-Net con entrada RGB + NIR y encoder ResNet18.

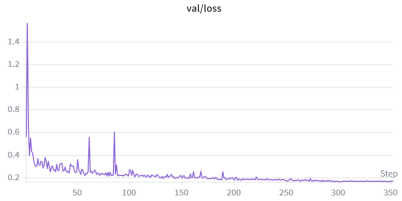


(a) Loss de validación.

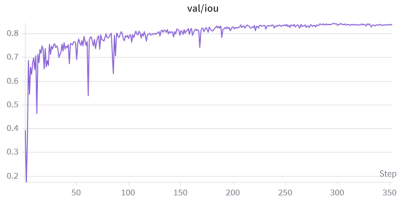


(b) mIoU de validación.

Figura C.8: Curvas de entrenamiento de U-Net con entrada RGB + NIR + RE y encoder ResNet18.

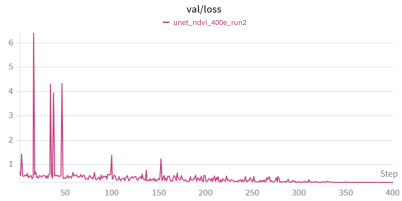


(a) Loss de validación.

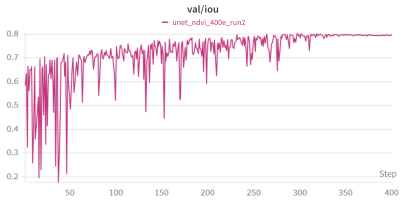


(b) mIoU de validación.

Figura C.9: Curvas de entrenamiento de U-Net con entrada RGB + NDVI y encoder ResNet18.

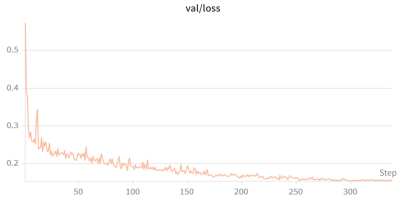


(a) Loss de validación.



(b) mIoU de validación.

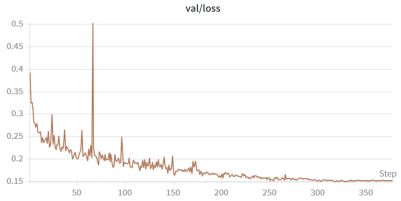
Figura C.10: Curvas de entrenamiento de U-Net con entrada NDVI y encoder ResNet18.



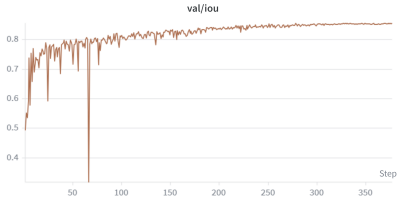
(a) Loss de validación.

(b) mIoU de validación.

Figura C.11: Curvas de entrenamiento de U-Net con entrada RGB y encoder EfficientNet-b0.



(a) Loss de validación.



(b) mIoU de validación.

Figura C.12: Curvas de entrenamiento de U-Net con entrada RGB + NIR y encoder EfficientNet-b0.

Anexo D

Guía de uso de ClusterUY

Este anexo es una guía breve para ejecutar trabajos en ClusterUY[1], la plataforma de computación de alto desempeño del Centro Nacional de Supercomputación del Uruguay. No sustituye a la documentación oficial; el contenido se apoya principalmente en la guía de ejecución de trabajos[2], que fue la base del uso que se hizo de la plataforma durante este proyecto.

D.1. Infraestructura y consideraciones generales

ClusterUY agrupa servidores con procesadores Intel Xeon y AMD EPYC, distintas configuraciones de RAM y GPUs NVIDIA P100, A100 y A40, interconectados por una red Ethernet de 10 Gbps. Cada nodo cuenta además con un disco SSD local de 300 GB para almacenamiento temporal de alta velocidad. Antes de diseñar un trabajo conviene consultar la página de recursos disponibles[3] para dimensionar los pedidos de memoria y cómputo y elegir el tipo de GPU más adecuado.

Un punto determinante es que todos los nodos corren CentOS 7. Las versiones de `glibc`, compiladores y bibliotecas del sistema base son considerablemente más antiguas que las que asume el software científico moderno, por lo que muchos *wheels* de Python, CUDA y cuDNN actuales no funcionan directamente sobre el entorno nativo. La forma recomendada de ejecutar software en ClusterUY es, por tanto, a través de contenedores Singularity, que desacoplan el entorno de ejecución del sistema base. Este punto se retoma más adelante.

El acceso se realiza por SSH contra el nodo de `login`. Esta máquina es únicamente un punto de entrada para editar archivos, lanzar trabajos y tareas livianas de gestión; no debe usarse para cómputo, compilación, descompresión de archivos grandes ni nada que consuma recursos de forma intensiva. Toda carga de trabajo debe ejecutarse en los nodos de cómputo, mediante `sbatch` o

[1] https://www.cluster.uy/
[2] https://www.cluster.uy/ayuda/como_ejecutar/
[3] https://www.cluster.uy/ayuda/recursos_disponibles/

sesiones interactivas. Para copiar volúmenes grandes de datos, la documentación recomienda usar el puerto alternativo `10022` con `scp` o `rsync`.

D.2. Envío de trabajos con SLURM

La gestión de recursos se realiza con SLURM. El comando `sbatch` recibe un *script* que describe el trabajo mediante directivas `#SBATCH` al inicio y lo encola hasta que haya recursos disponibles. Un esqueleto mínimo es el siguiente:

```
#!/bin/bash
#SBATCH --job-name=mi_trabajo
#SBATCH --ntasks=1
#SBATCH --mem=4G
#SBATCH --time=06:00:00
#SBATCH --tmp=10G
#SBATCH --partition=normal
#SBATCH --qos=normal
#SBATCH --mail-type=ALL
#SBATCH --mail-user=usuario@correo
#SBATCH --output=slurm-%j.out

# comandos del trabajo
```

Las directivas más relevantes especifican núcleos (`-ntasks`, `-cpus-per-task`), memoria (`-mem`), tiempo máximo (`-time`), espacio temporal en disco local (`-tmp`) y, cuando corresponde, GPU (`-gres`). Un trabajo no se cancela por exceder los núcleos o la memoria pedidos, pero sí si supera el `-time`; la documentación recomienda sobreestimar el tiempo en un 15 %.

Cada trabajo corre sobre una combinación de partición y QoS (*Quality of Service*). Las particiones públicas son `normal` y `besteffort`. La primera es la opción habitual para trabajos que deben ejecutarse sin interrupciones. La segunda ofrece más recursos pero sus trabajos pueden ser pospuestos o reencolados, por lo que conviene para tareas cortas o con *checkpoints* periódicos. El acceso a GPU requiere además la QoS correspondiente: `-qos=gpu` en `normal` o `-qos=besteffort_gpu` en `besteffort`; de lo contrario el trabajo queda en espera indefinidamente. El tipo de GPU se elige con `-gres`: `-gres=gpu:1` asigna cualquiera disponible, y los modificadores `p100`, `a100` o `a40` piden un modelo específico.

D.3. Almacenamiento temporal de alta velocidad

El `home` del usuario se monta por NFS, lo que hace que leer o escribir muchos archivos pequeños sea notoriamente más lento que hacerlo sobre el disco local del nodo. Para mitigarlo, cada servidor expone un directorio `/scratch/$USER/`

respaldado por un SSD local. Su capacidad debe reservarse explícitamente desde el *script* con la directiva `-tmp=xxxG`[4].

El patrón recomendado es: al inicio, copiar los datos de entrada desde el `home` hacia `/scratch`; durante el trabajo, leer y escribir sobre ese directorio local; al finalizar, copiar de vuelta al `home` solo los artefactos que interesa persistir y limpiar el resto. Conviene usar un identificador único como `/scratch/$USER/$SLURM_JOB_ID` para evitar colisiones con otros trabajos del mismo usuario en el mismo nodo. El impacto es especialmente notorio en cargas de trabajo con lecturas intensivas, como el entrenamiento de modelos sobre datasets de imágenes.

D.4. Ejecución con contenedores Singularity

Como se anticipó, la combinación de CentOS 7 y la evolución rápida del software científico hace que, en la práctica, el camino más confiable para ejecutar *stacks* modernos en ClusterUY sean los contenedores. La plataforma soporta Singularity[5], una tecnología de contenedores pensada para entornos HPC que permite importar y ejecutar imágenes Docker directamente. Dentro del contenedor el usuario conserva sus credenciales y accede por defecto a su `home`, al directorio actual y a `/tmp` del clúster, por lo que puede leer datos y escribir resultados de forma transparente.

La invocación más directa apunta a una imagen de un registro Docker:

```
singularity exec docker://python:3.10 python mi_script.py
```

Para flujos más elaborados conviene crear un entorno virtual dentro del contenedor, persistirlo en el `home` y reutilizarlo entre invocaciones. Las dependencias se instalan una sola vez contra el intérprete del contenedor y luego se consumen desde los `sbatch` posteriores:

```
singularity exec docker://python:3.10 \
    python -m venv ~/venvs/mi_env

singularity exec docker://python:3.10 \
    ~/venvs/mi_env/bin/pip install <dependencias>

singularity exec docker://python:3.10 \
    ~/venvs/mi_env/bin/python mi_script.py
```

Cuando se requiere GPU, Singularity admite la opción `-nv`, que expone los *drivers* NVIDIA del nodo al contenedor:

```
singularity exec --nv docker://<imagen_con_cuda> \
    ~/venvs/mi_env/bin/python entrenamiento.py
```

[4] https://www.cluster.uy/ayuda/tips/#uso-del-espacio-temporal-de-alta-velocidad

[5] https://www.cluster.uy/ayuda/singularity/

Es importante elegir una imagen cuya versión de CUDA sea compatible con los *drivers* del nodo asignado. El primer uso de una imagen implica su descarga y conversión al formato interno de Singularity, lo que puede demorar varios minutos; en invocaciones posteriores queda en la caché del usuario. Por último, los contenedores se usan tal como se descargan: no es posible modificarlos desde el clúster, por lo que la personalización del entorno debe hacerse mediante un *venv* externo como el del ejemplo, o construyendo una imagen propia en otra máquina.

D.5. Sesiones interactivas

Además de `sbatch`, SLURM permite abrir sesiones interactivas sobre un nodo de cómputo. Son útiles para depurar un *script*, probar un comando, hacer instalaciones puntuales dentro de un contenedor o cualquier tarea liviana que no justifique un trabajo por lotes.

La forma más rápida es el comando `interactivo`, que abre una shell con un tiempo máximo de 30 minutos. Las variantes `-g` y `-gn` asignan la sesión en `besteffort` o `normal` respectivamente; `-gpu` y `-gpun` hacen lo mismo adjuntando una GPU. Para sesiones más largas o con recursos específicos se usa `srun` con `-pty`:

```
srun --job-name=debug --time=01:00:00 \
     --partition=normal --qos=normal \
     --ntasks=1 --mem=8G --pty bash -l
```

D.6. Monitoreo y diagnóstico

El estado de los trabajos se consulta con `squeue`; `squeue -u <usuario>` filtra la cola a los trabajos propios. Para información detallada de un trabajo, como el nodo asignado o el motivo por el que aún no comenzó a ejecutarse, se usa `scontrol show job <ID>`. La cancelación se hace con `scancel <ID>`.

La salida estándar y la de error se vuelcan automáticamente en `slurm-<ID>.out` en el directorio desde el que se lanzó el `sbatch`, lo que permite revisar la traza sin conectarse al nodo. Configurar `-mail-type=ALL` junto a `-mail-user` envía una notificación por correo al inicio y al fin del trabajo, algo útil en ejecuciones largas.

Para el diagnóstico *post mortem* del consumo de recursos de un trabajo ya finalizado, la documentación oficial sugiere usar `sacct`, que reporta el tiempo de CPU, el *wall clock* y la memoria residente máxima:

```
sacct -j <jobid> --format=User,Job,JobName,CPUTime,Elapsed,MaxRSS
```

Comparar el `MaxRSS` con la memoria pedida mediante `-mem` permite ajustar futuros trabajos: evitar sub-dimensionamientos, que fuerzan el uso de *swap*, y sobre-dimensionamientos, que retrasan innecesariamente la planificación.

Anexo E

Instructivo de uso del repositorio

Este anexo describe el repositorio del proyecto[1] y el flujo de trabajo necesario para reproducir los experimentos, desde las capturas crudas de la cámara MicaSense hasta la inferencia con los modelos entrenados. En la documentación presente en el repositorio se puede ver en mayor detalle cómo llevar a cabo la ejecución de cada módulo.

En primer lugar, se presentan las funcionalidades ofrecidas por el repositorio y posteriormente se describe el flujo de ejecución.

E.1. Funcionalidades del repositorio

El repositorio implementa un pipeline completo de detección de malezas a partir de imágenes multiespectrales adquiridas con la cámara MicaSense RedEdge-P. Las funcionalidades que ofrece se agrupan en los siguientes bloques.

Alineamiento de bandas multiespectrales. Calcula y aplica las matrices de transformación necesarias para corregir el desplazamiento entre los sensores de la cámara, produciendo *stacks* multiespectrales coregistrados en formato TIFF.

Calibración radiométrica. Convierte los valores crudos del sensor a reflectancia calibrada utilizando las capturas del panel de referencia, generando los productos definitivos sobre los que se entrenan los modelos.

Construcción de configuraciones espectrales. Permite generar variantes del dataset seleccionando subconjuntos de bandas y añadiendo índices de vegetación (NDVI, GNDVI, NDRE, entre otros), lo que habilita la comparación

[1] https://github.com/cristiangdev/weed-detection-dataset-pipeline

sistemática del impacto de distintas combinaciones espectrales sobre el desempeño de los modelos.

Preparación de datasets para entrenamiento. Incluye particionado estratificado multietiqueta en `train`/`val`/`test`, recorte de las imágenes en *tiles* con transformación consistente de las anotaciones, conversión de polígonos a máscaras de segmentación semántica, y generación de reportes de distribución de clases.

Entrenamiento de modelos de segmentación multiespectral. Soporta dos arquitecturas, YOLOv8-seg y U-Net (con *backbone* configurable), ambas adaptadas para aceptar un número arbitrario de canales de entrada. La adaptación preserva la inicialización preentrenada de los canales RGB y extiende los canales adicionales a partir del promedio de dichos pesos.

Evaluación sobre el conjunto de test. Calcula y exporta las métricas estándar de cada arquitectura: mAP50, mAP50-95, precisión, *recall* y F1 (para *bounding box* y máscara) para YOLO; Dice, IoU, F1, precisión, *recall* y matriz de confusión por clase para U-Net.

Inferencia sobre nuevas imágenes. Aplica un modelo entrenado a un directorio de TIFFs multiespectrales y genera visualizaciones con las predicciones superpuestas sobre la composición RGB.

Herramientas auxiliares. Provee utilidades para la verificación visual de anotaciones, la inspección de la calidad del alineamiento y el seguimiento de experimentos mediante `wandb`.

E.2. Flujo de ejecución

A nivel de directorios, el repositorio se organiza en dos bloques funcionales que se corresponden con las etapas del pipeline: `preprocessing/`, que realiza la calibración radiométrica y el alineamiento de las bandas espectrales capturadas por los distintos sensores de la cámara; y `models/`, que agrupa la preparación de datos, el entrenamiento, la evaluación y la inferencia de los modelos, e incluye los subdirectorios `yolo/` y `unet/`, uno por cada arquitectura empleada. Conceptualmente el flujo está compuesto por seis etapas secuenciales que se describen a continuación: calibración radiométrica, alineamiento de bandas, preparación del dataset, entrenamiento, evaluación e inferencia. Cabe aclarar que la calibración radiométrica y el alineamiento de bandas, si bien son etapas conceptualmente distintas, están implementadas dentro del mismo script del módulo `preprocessing/`.

Calibración radiométrica. Convierte los valores crudos del sensor, provistos por la cámara MicaSense RedEdge-P como TIFFs de 16 bits, a reflectancia calibrada a partir de las capturas del panel de referencia. En la implementación del repositorio este paso se ejecuta de forma conjunta con el alineamiento de bandas dentro del módulo `preprocessing/`; los *stacks* multiespectrales finales se persisten en formato uint8 para reducir el volumen de datos sobre los que operan las etapas posteriores.

Alineamiento de bandas. El módulo `preprocessing/` calcula las *warp matrices* que corrigen el desplazamiento entre los sensores de la cámara MicaSense y las aplica para producir un *stack* multiespectral alineado en formato TIFF. El punto de entrada es `align_image_set.py`, y el script `verify_alignment.py` permite inspeccionar visualmente la calidad del resultado.

Preparación del dataset. Esta etapa agrupa varios scripts independientes en `models/`: `split_dataset.py` realiza el particionado estratificado multietiqueta en train/val/test; `slice_dataset.py` recorta las imágenes en *tiles* de 640×640 px y transforma las anotaciones YOLO asociadas; `channel_builder.py` genera variantes del dataset seleccionando subconjuntos de bandas y añadiendo índices de vegetación definidos en `vegetation_indexes.py`; `models/unet/masks.py` convierte anotaciones YOLO o COCO a máscaras de segmentación semántica para U-Net; y `dataset_stats.py` reporta la distribución de clases por split.

Entrenamiento. Los scripts `models/yolo/train.py` y `models/unet/train_unet.py` entrenan respectivamente un YOLOv8-seg y un U-Net, ambos adaptados para aceptar un número arbitrario de canales de entrada. La configuración de YOLO se gestiona mediante un archivo `config.ini`, mientras que U-Net recibe sus hiperparámetros por línea de comandos.

Evaluación. `models/yolo/eval.py` evalúa un modelo YOLO sobre el split de test y exporta a CSV las métricas globales y por clase para bounding box y máscara. `models/unet/eval_unet.py` hace lo propio para U-Net y serializa los resultados en JSON, junto con la matriz de confusión por clase.

Inferencia. El script `models/inference.py` ejecuta un modelo YOLO entrenado sobre un directorio de TIFFs multiespectrales y genera PNGs en los que las predicciones se superponen sobre la visualización RGB, permitiendo especificar de forma independiente las bandas entregadas al modelo y las usadas para la visualización.