Departamento de
Ingeniería Industrial
Universidad de Concepción

DESARROLLO DE UN MODELO BASADO EN ALGORITMOS DE DEEP LEARNING PARA LA DETECCIÓN Y CUANTIFICACIÓN DE ESTADO DE APERTURA EN IMÁGENES MICROSCÓPICAS DE ESTOMAS

POR

Martina Angélica Lara Arriagada

Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de Concepción
para optar al título profesional de Ingeniera Civil Industrial

Profesor(es) Guía
Dra. Mabel Vidal
Dr. Carlos Navarrete

Profesor Co-Guía
José Miguél Alvarez

Agosto 2025
Concepción (Chile)

A mi familia, por su paciencia infinita
mientras yo hablaba con un computador.
Y al computador, por finalmente hacerme caso.
A todos, gracias.

Agradecimientos

Quisiera expresar mi más profundo agradecimiento a todas las personas e instituciones que, de una forma u otra, hicieron posible la realización de esta tesis.

En primer lugar, agradezco infinitamente a mis profesores guía la Dra. Mabel Angelica Vidal Miranda y Dr. Carlos Camilo Navarrete Lizama, por su constante guía, su paciencia y su confianza en mi trabajo. Les agradezco también la libertad que me otorgarón para explorar mis propias ideas.

Este trabajo no habría sido posible sin el apoyo financiero de ANID – Programa Iniciativa Científica Milenio – NCN2024-047. Agradezco también a la Universidad de Talca, al laboratorio Castrolab y a la Universidad de Concepción por proveer los recursos y el entorno académico necesario para mi formación.

Al equipo de Phytolearning, gracias por las discusiones enriquecedoras y el apoyo técnico.

Finalmente, y de la manera más especial, agradezco a mi familia. A mis padres, Gloria Arriagada, Leonel Lara y Vicente Lara por su amor, su apoyo incondicional y por los valores que me inculcaron. Su sacrificio y su fe en mí han sido mi mayor motivación. A Kevin, gracias por tu amor, tu paciencia infinita y por ser mi refugio en los momentos de estrés.

A todos ustedes, gracias.

Resumen

Esta memoria de título aborda la necesidad de automatizar la identificación y medición de los estomas en imágenes microscópicas. Este proceso es fundamental para el análisis hídrico de cultivos, especialmente relevante ante el creciente impacto del cambio climático y la escasez de agua en la agricultura.

Los estomas son estructuras clave en el tejido epidérmico vegetal, esenciales para el intercambio de gases y la fotosíntesis, esto los convierte en indicadores vitales del estado hídrico de las plantas. El estudio propone el uso de metodologías de deep learning y computer vision para superar las limitaciones de los métodos manuales y semiautomatizados actuales, que son ineficientes, costosos y propensos a errores. La investigación esta centrada en dos objetivos principales:

Identificación del estoma, que consiste en la implementacion de un modelo de deep learning especializado en computer vision para la tarea de detección de objetos, basado en una arquitectura YOLO, se desarrolla para localizar estomas. Este modelo alcanza un alto rendimiento en métricas que evaluan distintas capacidades de los modelos. En precisión (0,957), recall (0,956), mAP@.50 (0,983) y mAP@.50:.95 (0,676), demostrando una capacidad robusta y precisa para identificar estomas. La determinación del estado de apertura, se logra mediante un modelo de segmentación de instancias que clasifica directamente el estado (abierto o cerrado). Los resultados del modelo tambien evidencian un buen rendimiento. Con mAP@.50 (0,889), mAP@.50:.95 (0,806), recall (0,906) y precisión (0,741), lo que indica su fiabilidad para la delimitación precisa de los estomas y su estado.

La metodología incluye la recolección y composición de diversos conjuntos de datos, provenientes de distintas fuentes, incorporando imágenes de especies como *Cicer arietinum* y *Arabidopsis thaliana*. Se realiza un preprocesamiento y re-etiquetado manual para asegurar la calidad y precisión de las anotaciones, utilizando máscaras de segmentación para delimitar las clases “stomata”, “pore-open” y “pore-closed”. Los resultados demuestran que los modelos desarrollados representan un avance significativo en la automatización del análisis estomático, proporcionando una herramienta fiable para el fenotipado y la gestión agrícola. Además, se explora el impacto de los mecanismos de atención en las arquitecturas del modelo, confirmando su potencial para mejorar el rendimiento de la segmentación. En resumen, esta memoria de título contribuye al campo del computer vision en la agricultura al ofrecer soluciones basadas en deep learning para la identificación y cuantificación del estado de apertura estomática, facilitando así la toma de decisiones informadas para el manejo hídrico de los cultivos.

Abstract

This thesis addresses the need to automate the identification and measurement of stomata in microscopic images. This process is fundamental for the water analysis of crops, especially relevant in the face of the growing impact of climate change and water scarcity in agriculture.

Stomata are key structures in the plant epidermal tissue, essential for gas exchange and photosynthesis. This makes them vital indicators of the water status of plants. The study proposes the use of deep learning and computer vision methodologies to overcome the limitations of current manual and semi-automated methods, which are inefficient, costly, and error-prone. The research is focused on two main objectives:

Stomata identification, which consists of the implementation of a deep learning model specialized in computer vision for the task of object detection, based on a YOLO architecture, developed to locate stomata. This model achieves high performance in metrics that evaluate different model capabilities: precision (0.957), recall (0.956), mAP@.50 (0.983), and mAP@.50:.95 (0.676), demonstrating robust and accurate capacity to identify stomata. The determination of the opening state is achieved through an instance segmentation model that directly classifies the state (open or closed). The model's results also show good performance, with mAP@.50 (0.889), mAP@.50:.95 (0.806), recall (0.906), and precision (0.741), indicating its reliability for the precise delimitation of stomata and their state.

The methodology includes the collection and composition of diverse datasets from different sources, incorporating images of species such as *Cicer arietinum* and *Arabidopsis thaliana*. Manual preprocessing and re-labeling are performed to ensure the quality and accuracy of the annotations, using segmentation masks to delimit the classes "stomata," "pore-open," and "pore-closed." The results demonstrate that the developed models represent a significant advance in the automation of stomatal analysis, providing a reliable tool for phenotyping and agricultural management. In addition, the impact of attention mechanisms in the model architectures is explored, confirming their potential to improve segmentation performance.

In summary, this thesis contributes to the field of computer vision in agriculture by offering deep learning–based solutions for the identification and quantification of stomatal opening states, thus facilitating informed decision-making for crop water management.

Índice General

Índice de Figuras

Índice de Tablas

Capítulo 1

Introducción

El presente capítulo aborda los antecedentes relacionados al problema de la identificación manual de estomas en imágenes microscópicas, así como tambien la medición del poro estomático. Además, se presentan los objetivos.

1.1. Antecedentes del problema

La Inteligencia Artificial (IA), y en particular, el campo *computer vision*, experimenta un crecimiento significativo en diversas industrias, destacándose su potencial en el ámbito agrícola debido a su capacidad para interpretar datos visuales y extraer información relevante. Esta tecnología permite automatizar tareas manuales y repetitivas que tradicionalmente requieren una alta inversión de tiempo y recursos, contribuyendo a la eficiencia de los procesos agrícolas. Se proyecta que el mercado global de IA en agricultura crecerá un 22,55 % hacia el año 2029 (Mordor Intelligence, 2024).

En el contexto biológico, los estomas son estructuras clave del tejido epidérmico vegetal, formadas por un poro estomático rodeado de dos células oclusivas, y en algunos casos, células subsidiarias. Estas estructuras son fundamentales en el intercambio de gases entre la planta y su entorno. De esta forma, su función es esencial para la fotosíntesis, los ciclos globales de carbono y oxígeno (Rovira et al., 2024).

El desarrollo de técnicas de *computer vision* y procesamiento de imágenes en la agricultura se ha visto impulsado por el equilibrio entre el costo computacional y la accesibilidad a GPUs (Unidades de procesamiento gráfico especializadas), que permiten realizar cálculos paralelos de manera eficiente. El rendimiento de la GPU se ha multiplicado por aproximadamente 7000 desde 2003 y el precio por rendimiento es 5600 veces superior (Stanford University Human-Centered Artificial Intelligence, 2024). Esto indica que, aunque el costo absoluto de las GPUs de alta gama puede ser elevado, la eficiencia en términos de rendimiento por dólar ha mejorado constantemente, haciendo que la computación de IA sea más accesible por unidad de capacidad. Aunque al utilizar metodos convencionales de automatización se superan muchas limitaciones de los métodos tradicionales manuales, aún presentan desafíos relevantes por resolver (Barbedo, 2017) . En este escenario, el presente estudio propone el uso de

nuevas metodologías basadas en *deep learning* para la identificación y cuantificación automática del estado abierto/cerrado de estomas en imágenes microscópicas, con el objetivo de mejorar la precisión del análisis hídrico de cultivos y facilitar la toma de decisiones en su manejo.

1.2. Relevancia del problema

El cambio climático afecta cada vez más a la agricultura y poder determinar a tiempo el estado de sequía de los cultivos sería información de gran valor. Es un hecho que hay un 70 % de cultivos de cereales dañados por la sequía en el Mediterráneo entre 2016–2018 (Organización de las Naciones Unidas, 2023). Así también, otros datos preocupantes indican que la escasez de agua puede acabar con los puestos de trabajo, como ocurrió durante la sequía de Ciudad del Cabo en 2018, que provocó la pérdida de los medios de subsistencia de 20.000 trabajadores agrícolas (Banco Mundial, 2025).

El presente estudio busca contribuir en la línea investigativa dedicada a la detección de estomas y extracción de parametros morfológicos asociados a ellos con *computer vision*. La biología moderna se enfoca en comprender los mecanismos funcionales y procesos moleculares de la célula viva en su entorno tisular nativo. De esto se despliega el campo que nos interesa, que sería entender las estrategias de adaptación celular en respuesta a estímulos internos y ambientales (Harter et al., 2012) del estoma, particularmente cómo se enfrenta a la sequía.

1.3. Formulación del problema

A pesar de los avances en IA y el creciente uso de técnicas de *computer vision* en el ámbito agrícola, la identificación y cuantificación del poro estomático en estomas continúa realizándose, en muchos casos, mediante métodos manuales o semiautomatizados, los cuales son ineficientes, costosos y propensos al error humano. Esta situación limita la posibilidad de implementar soluciones tecnológicas escalables para el monitoreo del estado hídrico de los cultivos, en especial en contextos de sequía cada vez más frecuentes y severos.

Considerando la importancia fisiológica de los estomas —estructuras clave para el intercambio gaseoso en las plantas— y el rol que juega el poro estomático como indicador del estado hídrico, se vuelve urgente desarrollar herramientas automatizadas que permitan detectar y medir esta estructura con precisión. Surge así la necesidad de investigar formas para diseñar un modelo basado en *deep learning* y procesamiento de imágenes microscópicas que identifique el estado de apertura del estoma, superando las limitaciones de los enfoques tradicionales.

Para ello, se plantean las siguientes preguntas de investigación:

- ¿Qué arquitectura de modelo basada en *deep learning* logra un mejor equilibrio entre métricas que miden distintas capacidades de los modelos?

- ¿Cuál es la combinación de hiperparámetros que logra un mejor equilibrio entre métricas que evaluan distintas capacidades del modelo?

1.3.1. Objetivo General

El objetivo general se puede definir como: "Desarrollar un modelo basado en *computer vision* y *deep learning* para la identificación del estoma y su estado de apertura, con el fin de proporcionar una herramienta que contribuya al análisis del estado hídrico de las plantas y extracción de parámetros morfológicos de ellos para así facilitar la toma de decisiones en la gestión agrícola.".

1.3.2. Objetivos Específicos

Los objetivos específicos que se esperan lograr con la Memoria de Título son:

1. Desarrollar técnicas de preprocesamiento de datos para optimizar la calidad del conjunto de datos, garantizando un entrenamiento eficiente del modelo y mejorando su robustez y capacidad de generalización.

2. Implementar modelos de *computer vision* basados en *deep learning* para la identificación automática del estoma y su determinación de estado abierto/cerrado en imágenes microscópicas.

3. Evaluar distintos modelos, determinando la mejor alternativa en términos de precisión y eficiencia computacional.

1.4. Estructura del Documento

El presente documento se estructura en cinco capítulos, diseñados para guiar al lector de manera progresiva desde la conceptualización del problema hasta la presentación y discusión de los hallazgos. La organización es la siguiente:

- **Capítulo 1. Introducción:** Se presenta el problema de investigación, contextualizando la importancia de la extracción automatizada de parametros morfológicos e identificación estomática en la fisiología vegetal. Se definen los objetivos generales, asi como tambien específicos del trabajo y se justifica la relevancia de aplicar técnicas de *Deep Learning* para esta tarea.

- **Capítulo 2. Marco Teórico y Estado del Arte:** Se revisan los fundamentos teóricos que sustentan este estudio, abarcando desde los conceptos biológicos de la morfología estomática hasta los principios de las redes neuronales convolucionales y los mecanismos de atención. Además, se analiza el estado del arte, examinando los trabajos previos y las soluciones existentes para la detección y segmentación de estomas mediante *computer vision*.

- **Capítulo 3. Metodología:** Se describe en detalle el diseño experimental y los procedimientos llevados a cabo. Se explica la construcción de los conjuntos de datos, el preprocesamiento de las imágenes, las arquitecturas de los modelos implementados y la estrategia de entrenamiento y validación.

- **Capítulo 4. Resultados y Discusión:** Se presentan los resultados cuantitativos y cualitativos obtenidos de los experimentos. Se exponen las métricas de rendimiento de los modelos de detección y segmentación, se analizan comparativamente las distintas configuraciones y se visualizan ejemplos del comportamiento de los modelos en los conjuntos de validación. Además se interpretan y analizan en profundidad los resultados. Se discuten las implicaciones de los hallazgos, se comparan con los trabajos revisados en el estado del arte y se abordan las fortalezas y debilidades de los modelos desarrollados.

- **Capítulo 5. Conclusiones y Trabajo Futuro:** Se sintetizan las conclusiones principales del estudio, respondiendo a los objetivos e hipótesis planteados en la introducción. Se discuten las limitaciones de la investigación y se proponen líneas de trabajo futuro para continuar y expandir los resultados obtenidos.

Capítulo 2

Estado del Arte y Marco Teórico

El objetivo de este capítulo es presentar una revisión del desarrollo y aplicación de modelos de deep learning, con énfasis en sus capacidades para procesar, analizar y extraer información relevante a partir de imágenes microscópicas. En particular, se exploran los avances más recientes en técnicas de detección y cuantificación aplicadas al análisis de estructuras biológicas, como los estomas.

2.1. ¿Qué son los estomas y por qué son importantes?

Los *estomas* son estructuras especializadas presentes en la epidermis de las plantas terrestres, cuya función principal es permitir el intercambio gaseoso entre el interior de la planta y la atmósfera circundante (Willmer & Fricker, 1996). Según Willmer y Fricker (1996), un estoma está compuesto por un poro estomático delimitado por dos células oclusivas (vease la Figura 2.1), cuyas paredes celulares presentan un engrosamiento diferencial que permite la apertura y cierre del poro en respuesta a cambios en la turgencia celular. Estas células son cruciales para regular la fotosíntesis y la transpiración. En muchas especies, estas células están además acompañadas por células subsidiarias que contribuyen al control del movimiento estomático y la eficiencia del uso del agua (Hetherington & Woodward, 2003).

La morfología de los estomas, que incluye las células oclusivas y el poro correspondiente, así como su densidad (número por unidad de área), tamaño y patrón de distribución, influye significativamente en la capacidad general de intercambio de gases de la planta y en sus respuestas fisiológicas a los cambios ambientales (Farooq et al., 2012). Por lo tanto, comprender el comportamiento y la morfología estomática es vital para desentrañar los mecanismos de adaptación de las plantas a diversos factores de estrés ambiental, como la sequía, las fluctuaciones de temperatura y los niveles variables de CO2 (Haworth et al., 2011; X. Li et al., 2022; Vico et al., 2011). Variaciones en el tamaño y la forma de las células oclusivas también impactan la regulación estomática (Martin & Glover, 2007).

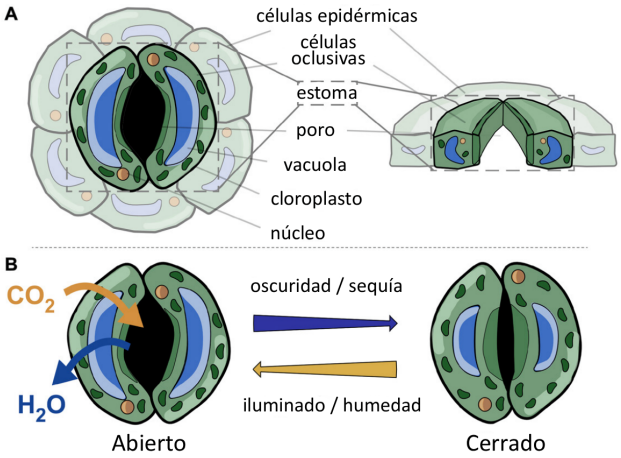


Figura 2.1: Diagrama de estructura morfológica de un estoma. Imagen de (Gibbs & Burgess, 2024).

El recuento y la medición de los estomas en imágenes microscópicas de la epidermis foliar es una de las actividades biológicas más típicas en el estudio de plantas (Willmer & Fricker, 1996). Aunque la segmentación manual permite una alta precisión, delimitar los estomas uno por uno y seguir los contornos irregulares es una tarea demandante en tiempo y esfuerzo (Jayakody et al., 2022). Poder mejorar la productividad de cultivos está estrechamente relacionado con tomar decisiones tempranas para el manejo de la sequía de la planta, lo cual contribuye a abordar un problema global (Commission, 2022; Deltares, 2018; Ogunrinde et al., 2025).

2.2. Inteligencia Artificial

La IA es el estudio y diseño de agentes racionales. Un agente racional es una entidad que percibe su entorno a través de sensores y actúa sobre él mediante actuadores de manera que maximiza su medida de rendimiento esperada. En términos más sencillos, es un sistema que actúa de la mejor manera posible para alcanzar sus objetivos, dadas las circunstancias y la información que posee (Russell & Norvig, 2020). La IA busca desarrollar sistemas capaces de realizar tareas que normalmente requieren inteligencia humana, como el razonamiento, la percepción y el aprendizaje. Históricamente, dos trabajos marcan su inicio formal. Por una parte, Turing (1950) propuso el Test de Turing, una prueba para evaluar la capacidad de una máquina para exhibir comportamiento inteligente indistinguible del de un humano. Por otra parte, McCarthy et al. (1956) acuñó el término "Inteligencia Artificial", definiéndola como "la ciencia e ingeniería de crear máquinas inteligentes", durante la histórica conferencia de Dartmouth.

La IA ha evolucionado rapidamente, adoptando distintos enfoques y paradigmas, desde los sistemas basados en reglas y lógica simbólica (Nilsson, 1980), hasta el Machine Learning y el Deep Learning, que dominan el panorama actual (Goodfellow et al., 2016; LeCun et al., 2015). El Machine Learning es el paradigma de IA donde los sistemas aprenden patrones y toman decisiones directamente de los

datos, sin programación explícita para cada tarea. Estos últimos enfoques han permitido avances en procesamiento de imágenes y distintas industrias, como lo es la agricultura.

2.2.1. Deep Learning

El **Deep Learning (DL)** es una subrama de la IA que se inspira en la estructura y función del cerebro humano, utilizando **redes neuronales artificiales** con múltiples capas (de ahí el término "profundo") para aprender representaciones de datos con múltiples niveles de abstracción (LeCun et al., 2015). A diferencia de los métodos tradicionales de Machine Learning que a menudo requieren una extracción de características manual, como se ilustra en la Figura 2.2, el DL puede aprender automáticamente las características relevantes directamente de los datos brutos, lo que lo hace particularmente potente para tareas como el procesamiento de imágenes, el reconocimiento de voz y el procesamiento del lenguaje natural.

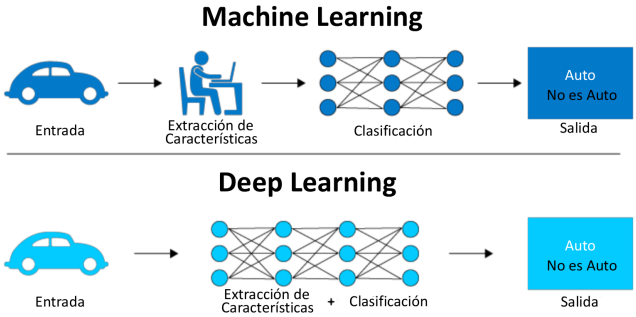


Figura 2.2: Diagrama que representa la diferencia entre el Machine Learning y DL. Imagen de (Almuiña, 2024).

Para entender el DL, es fundamental comprender el concepto de **red neuronal artificial**, que es su componente central. Las redes neuronales están compuestas por unidades interconectadas llamadas **neuronas** o **nodos**, organizadas en capas.

2.2.1.1. Perceptrón Simple

El **Perceptrón Simple** es el tipo más básico de red neuronal artificial, propuesto por Rosenblatt (1958). Consiste en una sola capa de neuronas de salida. Cada neurona recibe una o más entradas, a cada una de las cuales se le asigna un peso. Estas entradas ponderadas se suman y se pasan a través de una **función de activación** que determina la salida de la neurona. El perceptrón simple es capaz de aprender a clasificar datos que son **linealmente separables**. Matemáticamente, la salida de un perceptrón simple se calcula mediante la Ecuación (2.1).

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{2.1}$$

Donde:

- x_i: son las entradas.
- w_i: son los pesos correspondientes a cada entrada.
- b: es el sesgo (bias).
- $\sum_{i=1}^{n} w_i x_i + b$: es la suma ponderada de las entradas más el sesgo.
- $f(\cdot)$: es la función de activación.
- y: es la salida de la neurona.

La Figura 2.3 ilustra cómo esta compuesto un perceptron simple.

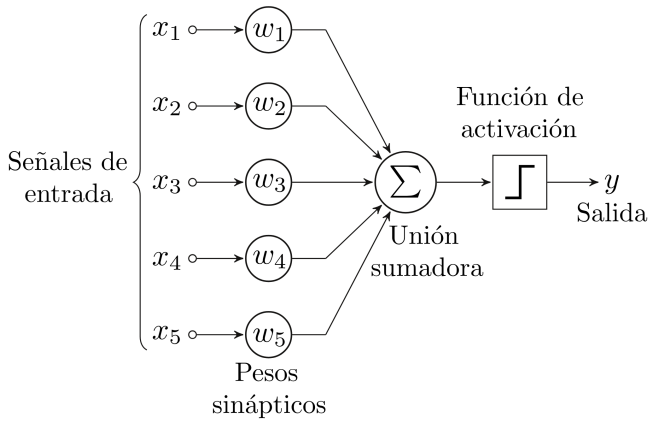


Figura 2.3: Representación de una neurona artificial realizando una suma ponderada de sus entradas. Cada entrada (x_i) se multiplica por un peso (w_i), se suman y se añade un sesgo antes de pasar por una función de activación para producir una salida (y). Imagen de (Colaboradores de Wikipedia, 2024).

2.2.1.2. Perceptrón Multicapa (Red Neuronal Feedforward)

Un **Perceptrón Multicapa (MLP)** o **Red Neuronal Feedforward** es una extensión del perceptrón simple que supera sus limitaciones. Los MLPs constan de al menos tres capas de nodos: una **capa de entrada**, una o más **capas ocultas** y una **capa de salida**. Cada neurona en una capa está conectada a todas las neuronas de la capa siguiente, y la información fluye en una sola dirección, desde la entrada hasta la salida ("feedforward").

A diferencia del perceptrón simple, las capas ocultas y el uso de funciones de activación no lineales como las que se muestran en la Tabla 2.1 permiten a los MLPs aprender y modelar relaciones complejas y no lineales en los datos, lo que los hace adecuados para una amplia gama de problemas que no son linealmente separables. El entrenamiento de MLPs generalmente se realiza utilizando algoritmos como la **retropropagación (backpropagation)** combinada con el descenso de gradiente para ajustar los pesos y sesgos de la red.

Tabla 2.1: El resumen de las funciones de activación se ha elaborado a partir de MIRANDA (2022).

Function	Equation
Linear	$f(x) = x$
Hard sigmoid	$f(x) = \text{máx}\left(0, \text{mín}\left(1, \frac{x+1}{2}\right)\right)$
Hiperbolic tangent	$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
SoftSign	$f(x) = \frac{x}{1+\lvert x\rvert}$
Rectified linear unit	$f(x) = \text{máx}(0, x)$
Leaky ReLU	$f(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha x, & \text{if } x \leq 0 \end{cases}$
SELU	$f(x) = \lambda \begin{cases} x, & \text{if } x > 0 \\ \alpha(e^x - \alpha) & \text{if } x \leq 0 \end{cases}$
Softplus	$f(x) = \log(1 + e^x)$
Softmax	$f(x) = \frac{e^{a_i}}{\sum_j e^{a_j}}$

2.2.1.3. Retropropagación

La **Retropropagación (Backpropagation)** es un algoritmo fundamental para entrenar redes neuronales artificiales, especialmente MLP. Fue introducido por Werbos (1974) y popularizado por Rumelhart et al. (1986). El algoritmo opera en dos fases: una pasada hacia adelante (forward pass) y una pasada hacia atrás (backward pass). Durante la pasada hacia adelante, las entradas se propagan a través de la red para generar una salida. En la pasada hacia atrás, se calcula el error entre la salida predicha y la salida deseada (el objetivo). Este error se propaga hacia atrás a través de la red, desde la capa de salida hasta la capa de entrada, utilizando la **regla de la cadena** del cálculo para determinar cómo cada peso y sesgo en la red contribuye al error total. El objetivo es calcular el gradiente de la función de pérdida con respecto a cada peso (w_{ij}) y sesgo (b_j) de la red.

La actualización de un peso w_{ij} entre la neurona i de la capa anterior y la neurona j de la capa actual se basa en el gradiente de la función de costo C con respecto a ese peso, como se define en la Ecuación 2.2:

$$\frac{\partial C}{\partial w_{ij}} = \frac{\partial C}{\partial a_j} \frac{\partial a_j}{\partial z_j} \frac{\partial z_j}{\partial w_{ij}} \tag{2.2}$$

donde a_j es la activación de la neurona j (la salida después de la función de activación), y z_j es la suma ponderada de las entradas a la neurona j antes de la función de activación. Similarmente, para el sesgo b_j, se puede ver en la Ecuación (2.3):

$$\frac{\partial C}{\partial b_j} = \frac{\partial C}{\partial a_j}\frac{\partial a_j}{\partial z_j}\frac{\partial z_j}{\partial b_j} \tag{2.3}$$

Estos gradientes son luego utilizados por un algoritmo de optimización, como el descenso de gradiente, para ajustar los pesos y sesgos de la red y reducir el error. Este proceso se repite iterativamente hasta que el error de la red es minimizado o se alcanzan otros criterios de parada.

2.2.1.4. Gradient Descent

Gradient Descent (GD) es un algoritmo de optimización iterativo utilizado para minimizar una función objetivo $J(\theta)$ parametrizada por los pesos de una red neuronal θ. Para funciones con múltiples entradas, GD utiliza el gradiente de J (un vector que contiene todas las derivadas parciales de J con respecto a θ, denotado como $\nabla_\theta J(\theta)$) para actualizar los parámetros iterativamente en la dirección del descenso más pronunciado. Esto se logra siguiendo la dirección opuesta al gradiente.

El descenso de gradiente presenta variantes que difieren en la cantidad de datos utilizados para calcular el gradiente de la función objetivo. Debido a la cantidad limitada de datos, en la práctica GD puede realizarse de diferentes maneras, a continuación se presentan algunas:

- **Batch Gradient Descent (BGD)**: Calcula el gradiente de la función de costo con respecto a los parámetros θ para todo el conjunto de datos de entrenamiento. El tamaño de los pasos se conoce como la tasa de aprendizaje η, y define cuánto deben moverse los parámetros en la dirección opuesta al gradiente en la iteración t para avanzar hacia el mínimo. Los parámetros se actualizan mediante la Ecuación (2.4):

  $$\theta_{t+1} = \theta_t - \eta \cdot \nabla_\theta J(\theta_t) \tag{2.4}$$

  BGD es computacionalmente eficiente ya que produce un gradiente de error y una convergencia estables. Convergerá al mínimo global si la función de pérdida es convexa y puede converger a un mínimo local si la función de pérdida no es convexa. Sin embargo, puede ser impráctico para grandes conjuntos de datos dado que necesita calcular los gradientes de la función de pérdida para todos los datos. Además, se ha demostrado que BGD tiene una convergencia lenta en comparación con otros métodos (Ruder, 2016).

- **Stochastic Gradient Descent (SGD)**: Esta variante realiza una actualización de parámetros para cada ejemplo de entrenamiento individual (x_n, y_n). El aprendizaje se realiza para cada ejemplo como en la Ecuación 2.5:

  $$\theta_{t+1} = \theta_t - \eta \cdot \nabla_\theta J(\theta_t; x_n; y_n) \tag{2.5}$$

Generalmente, se utiliza para aprender en línea y más rápido dado que SGD realiza una actualización a la vez y con una alta varianza que hace que la función objetivo fluctúe (Goodfellow et al., 2016). A pesar de mejorar el modelo debido a sus actualizaciones frecuentes, SGD aumenta el tiempo de ejecución, lo que lo hace computacionalmente costoso. Una desventaja es que este algoritmo puede converger a un mínimo local y presentar una alta varianza, sin alcanzar el resultado óptimo global.

Establecer los parámetros y elegir una tasa de aprendizaje adecuada puede ser una tarea desafiante. Por lo tanto, se han propuesto diferentes variantes de descenso de gradiente para mejorar el rendimiento del aprendizaje. Estos optimizadores funcionan modificando el componente de la tasa de aprendizaje, el componente del gradiente, o ambos (Bottou, 2012). Algunos ejemplos incluyen (Ruder, 2016):

- **Momentum**: Este método propuesto por Polyak (1964), fue diseñado para acelerar el aprendizaje y principalmente apunta a resolver el mal condicionamiento de la matriz Hessiana y la varianza en el gradiente estocástico. Momentum añade una fracción γ del vector actualizado del paso de tiempo anterior v_{t-1} al vector de actualización actual v_t, como se muestra en la Ecuación (2.6) y (2.7):

$$v_t = \gamma v_{t-1} + \eta \nabla_\theta J(\theta) \tag{2.6}$$

$$\theta = \theta - v_t \tag{2.7}$$

- **RMSprop (Root Mean Square Propagation)** (Hinton et al., 2012): Este método utiliza una tasa de aprendizaje adaptativa que modifica AdaGrad para un mejor rendimiento en un entorno no convexo. Funciona dividiendo la tasa de aprendizaje para un peso por un promedio móvil de las magnitudes de los gradientes recientes para ese peso, como se detalla en la Ecuación (2.8).

$$v_t = \gamma v_{t-1} + (1-\gamma) \cdot (\nabla_\theta J(\theta))^2 \tag{2.8}$$

$$\theta = \theta - \frac{\eta}{\sqrt{v_t + \epsilon}} \nabla_\theta J(\theta) \tag{2.9}$$

  donde γ es un hiperparámetro que pondera la contribución de v_{t-1} y el cuadrado del gradiente a v_t, y ϵ es una pequeña constante que evita la división por cero, vease en la Ecuación (2.9).

- **Adam (Adaptive Moment Estimation)** (Kingma & Ba, 2015): Calcula tasas de aprendizaje adaptativas para cada parámetro. Se considera una combinación de momentum y RMSprop que, además de usar el promedio móvil decreciente de los gradientes cuadrados pasados para las tasas

de aprendizaje específicas de los parámetros, también emplea un promedio móvil decreciente de los gradientes pasados en lugar del gradiente actual. Formalmente se muestra en la Ecuación (2.10):

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t \tag{2.10}$$

donde $\hat{v}_t$ y $\hat{m}_t$ son promedios corregidos por sesgo para asegurar que los valores no estén sesgados hacia 0.

Otros ejemplos de algoritmos de optimización de descenso de gradiente son: AdaDelta (Zeiler, 2012), AdaMax (Kingma & Ba, 2015), Nadam (Dozat, 2016), AMSGrad (Reddi et al., 2018) que son variaciones de los algoritmos explicados anteriormente.

2.2.1.5. Redes Neuronales Convolucionales

Las CNNs son redes neuronales capaces de procesar gran cantidad de imágenes y datos con alta dimensionalidad. Sin embargo, esto no asegura un buen rendimiento del modelo. Por esta razón, existen distintos métodos y tratamientos que se pueden aplicar a los datos o durante el entrenamiento para intentar mejorar su desempeño, tales como:

- **Dropout**: Técnica de regularización para redes neuronales que previene el sobreajuste (*overfitting*) al introducir aleatoriedad durante el entrenamiento (Srivastava et al., 2014).
- **Data Augmentation**: Técnica de aumentación de datos que, al introducir más diversidad, puede mejorar la capacidad de generalización del modelo.
- **Cross Validation**: Método que consiste en dividir repetidamente el conjunto de datos para entrenamiento, validación y prueba, y luego promediar sus métricas (Kohavi, 1995).
- **Batch Normalization**: Consiste en normalizar las entradas de cada capa de la red para que tengan una media cercana a cero y una varianza cercana a uno. Esto se realiza calculando la media y la varianza de cada mini-lote durante el entrenamiento y utilizando estos valores para normalizar las funciones de activación. El objetivo es reducir los cambios en la distribución de las funciones de activación internas de la red durante el entrenamiento, ya que estos cambios pueden dificultar la convergencia del modelo (Ioffe & Szegedy, 2015).

Además, tienen determinadas operaciones que las hacen características y diferenciadoras de otros tipos de redes. De estas, se destacan:

- **Convolución**: Como se puede observar en la Figura 2.4, al hacer las operaciones matriciales entre el kernel y los valores de los pixeles en la imagen segun la Ecuación (2.11), notamos que se puede reducir la dimensionalidad de la imagen fusionando las características, donde cada píxel de salida (i, j) es el resultado de una suma de productos de los píxeles vecinos con los pesos del kernel .

$$(S * K)(i, j) = \sum_{m=-M}^{M} \sum_{n=-N}^{N} S(i - m, j - n) \cdot K(m, n) \tag{2.11}$$

  - $(S * K)(i, j)$: Resultado de la convolución entre la imagen S y el kernel K en la posición (i, j).
  - $S(i - m, j - n)$: Valor del píxel de la imagen de entrada S desplazado por (m, n) respecto a la posición (i, j).
  - $K(m, n)$: Valor del kernel (o filtro) en la posición (m, n).
  - $\sum_{m=-M}^{M} \sum_{n=-N}^{N}$: Sumatoria doble que recorre todas las posiciones del kernel de tamaño $(2M + 1) \times (2N + 1)$.

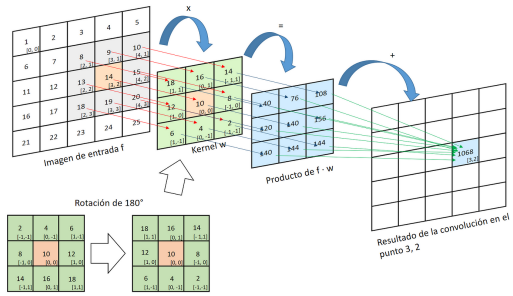


Figura 2.4: Ejemplo gráfico del proceso de convolución (Cuartas, 2020).

- **Feature maps**: los **mapas de características** (*feature maps*) son las salidas fundamentales generadas por las capas convolucionales. Representan patrones o características aprendidas que han sido detectadas en los datos de entrada, como una imagen. Según Ren et al. (2017), los mapas de características son esenciales para cómo los modelos de aprendizaje profundo interpretan la información visual. Se crean a través de la operación matemática de convolución, donde filtros (también conocidos como *kernels*) se deslizan sobre la imagen de entrada o el mapa de características de una capa anterior.

- **Pooling**: LeCun et al. (1998) El pooling reduce la resolución de las feature maps, disminuyendo la cantidad de parámetros y cómputo, además ayuda a que el modelo sea invariante a pequeñas traslaciones o distorsiones en los datos de entrada. Adicionalmente, hay distintos tipos que varian segun su aplicación.

- **Max Pooling**: Selecciona el valor máximo en una región local del mapa de activación, reduciendo la dimensionalidad y haciendo la representación más robusta a pequeñas traslaciones como se muestra en la Ecuación (2.12).

$$Y(i,j) = \underset{(m,n)\in\mathcal{R}_{i,j}}{\text{máx}} X(m,n) \tag{2.12}$$

  - $Y(i,j)$: Valor de la salida de la operación de pooling en la posición (i,j).
  - $X(m,n)$: Valor de entrada en la posición (m,n).
  - máx: Operador que selecciona el valor máximo dentro de un conjunto de valores.
  - $\mathcal{R}_{i,j}$: Región de entrada correspondiente a la posición (i,j) en la salida.

- **Average Pooling**: Calcula el promedio de los valores en la región local, vease en la Ecuación (2.13).

$$Y(i,j) = \frac{1}{|\mathcal{R}_{i,j}|} \sum_{(m,n)\in\mathcal{R}_{i,j}} X(m,n) \tag{2.13}$$

  - $|\mathcal{R}_{i,j}|$: Número total de elementos en la región de pooling (por ejemplo, 4 si es una ventana 2×2).

- **Region Proposal Network (RPN)**: Es un componente fundamental en arquitecturas de detección de objetos como Faster R-CNN (Ren et al., 2017). Su función principal es generar propuestas de regiones candidatas donde podrían encontrarse objetos, basándose en mapas de características convolucionales. Opera prediciendo las delimitaciones de objetos y su .ºbjetividad"(si contiene un objeto o no) en un número fijo de escalas y relaciones de aspecto predefinidas (áncoras) por cada ubicación espacial. Esto permite que el sistema de detección de objetos se enfoque eficientemente en áreas prometedoras de la imagen para una clasificación y regresión más detalladas.

- **RoIAlign**: Fue introducido en Mask R-CNN (He et al., 2017) como una mejora sobre RoIPooling. Su propósito es extraer pequeñas características espaciales de los mapas de características a partir de propuestas de regiones de interés (RoIs) de tamaño variable, de manera que sean consistentes para capas posteriores. A diferencia de RoIPooling, que realiza cuantificación de coordenadas, RoIAlign utiliza interpolación bilineal para calcular los valores exactos de los píxeles flotantes, preservando la información espacial precisa. Esto es crucial para tareas que requieren una alineación fina, como la segmentación de instancias.

Para automatizar la extracción de parámetros morfológicos, este trabajo se centra en dos tareas fundamentales del DL. Dado que cada tarea es abordada por arquitecturas de red específicas, los expe-

rimentos se enfocarán concretamente en la Detección de Objetos (Object Detection) y la Segmentación de Instancias (Instance Segmentation), por ser las más adecuadas para los objetivos planteados.

2.2.2. Procesamiento de Imágenes

El *Computer Vision* (CV) es un campo que estudia cómo hacer que las computadoras interpreten datos visuales para obtener conocimiento útil del entorno (Ballard & Brown, 1982). Además, este campo está estrechamente relacionado con el DL, ya que influye drásticamente en el procesamiento de imágenes mediante arquitecturas que aprenden automáticamente representaciones jerárquicas a partir de los datos crudos, eliminando en gran medida la necesidad de *feature engineering*. El desarrollo inicial se remonta al perceptrón propuesto por Rosenblatt (1958), pero en el contexto de procesamiento de imágenes, el verdadero punto de inflexión se dio con las CNNs, introducidas por LeCun et al. (1998) con el reconocimiento de dígitos manuscritos y posteriormente popularizadas por el desempeño sobresaliente en ImageNet (Deng et al., 2009). Estas redes explotan las estructuras espaciales de las imágenes mediante distintas operaciones para lograr la tarea a realizar. Además de las CNN convencionales, existen otros tipos de arquitecturas derivadas (Ren et al., 2017) y mecanismos útiles en el procesamiento de imágenes, como los mecanismos de atención propuestos por Vaswani et al. (2017).

2.2.3. Object Detection

La detección de objetos es un campo fundamental en CV. Esta se encarga de la identificación y localización de instancias de objetos de categorías específicas dentro de imágenes o secuencias de video. Este proceso no solo clasifica los objetos presentes (por ejemplo, identificando una “persona” o un “vehículo”), sino que también genera un recuadro delimitador preciso alrededor de cada instancia detectada, proporcionando sus coordenadas espaciales.

2.2.3.1. Faster R-CNN

Un derivado de las CNN es la Faster R-CNN (Ren et al., 2017), que se caracteriza por agregar a una red convolucional común, componentes diseñados específicamente para la detección de objetos, no solo clasificación general. Esto se logra integrando una red adicional que corre sobre el mapa de características y propone regiones que probablemente contengan objetos. En cada posición del *feature map*, se evalúan múltiples “cajas base” (de distintas formas y tamaños) para detectar objetos de diferentes escalas y relaciones de aspecto. A diferencia de una CNN común que solo produce una clase por imagen, Faster R-CNN realiza múltiples predicciones por imagen.

2.2.3.2. You Only Look Once

You Only Look Once (YOLO) (Redmon et al., 2016) es un modelo de detección de objetos en tiempo real que reformula la tarea como un problema de regresión directa, prediciendo las coordenadas de las cajas delimitadoras y las probabilidades de clase a partir de los píxeles de la imagen. La primera versión de YOLO, lanzada en 2016, consta de una arquitectura más básica en comparación con las versiones actuales.

Con el paso de los años, la arquitectura base se ha perfeccionado mediante la incorporación de nuevos componentes y técnicas. Redmon y Farhadi (2017) introdujeron YOLOv2 (también conocido como YOLO9000), que mejora la precisión incorporando cajas de anclaje (anchor boxes) y técnicas de entrenamiento multi-escala. Posteriormente, Redmon y Farhadi (2018) se lanza YOLOv3, que refina la arquitectura con una red *backbone* más potente (Darknet-53) y predicciones multi-escala mejoradas para una mejor detección de objetos de diferentes tamaños.

Más recientemente, Wang et al. (2023) se propone YOLOv7, estableciendo nuevos récords en el equilibrio velocidad-precisión, avanzando así tambien al modelo YOLOv10 Yang et al. (2024). Mientras que al dia de hoy ya se encuentra implementada la versión 12 lanzada este año por Tian y Shen (2025).

2.2.4. Instance Segmentation

A diferencia de la detección de objetos, que se limita a recuadros delimitadores, la segmentación de instancias genera una máscara de píxeles precisa para el contorno de cada objeto. Además, se diferencia de la segmentación semántica en que no solo clasifica cada píxel en una categoría, sino que también distingue entre distintas ocurrencias de la misma clase de objeto. Por ejemplo, si una imagen contiene varias personas, la segmentación de instancias genera una máscara separada para cada individuo, proporcionando una representación detallada y diferenciada que es crucial para tareas que requieren una comprensión granular de la escena.

2.2.4.1. U-Net

La U-Net es una arquitectura de CNN diseñada principalmente para la segmentación semántica de imágenes, es decir, para clasificar cada píxel de una imagen (véase en la Figura 2.5). Es introducida por Ronneberger et al. (2015) y se caracteriza por su arquitectura simétrica en forma de U, que consta de una trayectoria de contracción (encoder) para capturar el contexto y una trayectoria de expansión (decoder) para permitir una localización precisa. La clave de su funcionamiento reside en las conexiones de salto (skip connections) que transmiten características de alta resolución desde el encoder al decoder, permitiendo que la red combine información de contexto de las capas profundas con información espacial precisa de las capas superficiales. Esto es crucial para generar segmentaciones detalladas, especialmente en escenarios con datos de entrenamiento limitados, como en aplicaciones biomédicas.

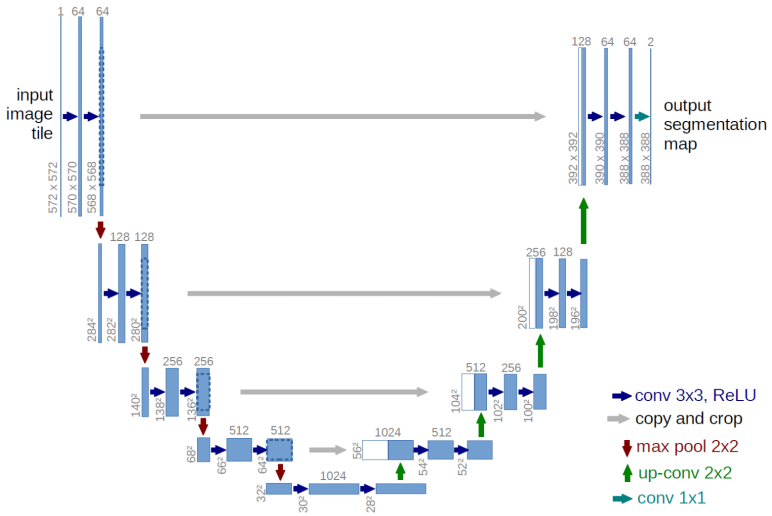


Figura 2.5: Arquitectura de U- Net. Imagen tomada de (DataScientest, 2024).

2.2.4.2. Mask R-CNN

Otra arquitectura importante es Mask R-CNN, la cual extiende Faster R-CNN para abordar el problema de segmentación de instancias, donde no solo se clasifica, sino que también se detectan múltiples objetos en una imagen, se localizan mediante cajas delimitadoras y se segmenta cada uno a nivel de píxel. Utiliza la arquitectura convencional de CNN, añade una RPN y emplea una operación de *RoIAlign* para extraer regiones con alineación espacial como se muestra en la Figura 2.6. Esto funciona a través de tres ramas paralelas: una predice la clase (clasificación), otra la caja delimitadora (coordenadas) y otra genera la máscara (segmentación binaria de la región del objeto).

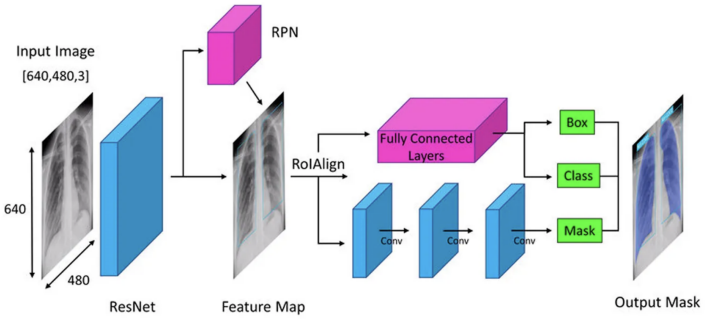


Figura 2.6: Mecanismo de funcionamiento de Mask R-CNN. Imagen adaptada de (Ultralytics, 2023)

2.2.5. Mecanismos de Atención para el Procesamiento de Imágenes

Los mecanismos de atención han emergido como componentes clave en arquitecturas de redes neuronales, permitiendo que los modelos enfoquen su procesamiento en las características más relevantes de los datos de entrada. Al asignar pesos a diferentes partes de una imagen o a diferentes canales de características, los mecanismos de atención mejoran la capacidad del modelo para capturar

dependencias y refinar las representaciones. Dos mecanismos notables son Coordinate Attention (CA) y Convolutional Block Attention Module (CBAM).

2.2.5.1. Coordinate Attention

CA es introducido por Hou et al. (2021) y es un mecanismo de atención ligero y eficiente que busca mejorar la capacidad de los modelos para capturar información de posición y relaciones inter-canal a través de dos pasos principales: el **embedding de información de coordenadas** y la **generación de atención de coordenadas**.

- **Embedding de información de coordenadas**: A diferencia de los mecanismos de atención de canal tradicionales que colapsan una característica espacialmente en un único vector, CA descompone el proceso de agrupación global en dos agrupaciones unidimensionales (1D). Esto significa que las características de entrada se promedian a lo largo de las dimensiones horizontal y vertical de forma separada, como se muestra en la Figura 2.7. Específicamente, se utilizan operaciones de Global Average Pooling 1D a lo largo de los ejes X e Y para codificar la información espacial,lo que permite que capture dependencias de largo alcance a lo largo de una dirección espacial (e.g., horizontal) y, al mismo tiempo, preserve información posicional precisa a lo largo de la otra dirección (e.g., vertical) (SERP.AI, 2022). El resultado son dos vectores de características que contienen información espacial a lo largo de las dos direcciones.

- **Generación de atención de coordenadas**: Los dos vectores de características resultantes se concatenan y luego se transforman mediante una operación convolucional 1×1. Esta transformación permite que el modelo aprenda las relaciones de dependencia entre los canales. La salida se divide luego en dos tensores separados, que se transforman mediante operaciones convolucionales 1×1 y funciones de activación para generar los mapas de atención de canal para cada dirección espacial. Finalmente, estos mapas de atención se multiplican con el mapa de características de entrada original para realzar las características relevantes y suprimir el ruido.

Han demostrado su eficacia en tareas como la **clasificación de imágenes**, **detección de objetos** y **segmentación semántica**, mejorando el rendimiento sin aumentar significativamente la complejidad computacional (SERP.AI, 2022).

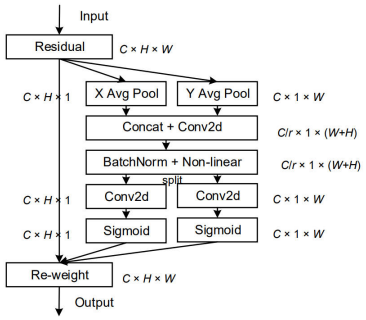


Figura 2.7: Arquitectura del mecanismo de atención CA. Imagen tomada de (Hou et al., 2021).

2.2.5.2. Convolutional Block Attention Module

El CBAM, propuesto por Woo et al. (2018), es un módulo de atención modular y ligero que puede insertarse en cualquier CNN para mejorar la representación de las características. Opera de manera secuencial, inferiendo mapas de atención tanto a lo largo de la dimensión del canal como de la dimensión espacial, y luego multiplicando estos mapas de atención con el mapa de características de entrada para la refinación adaptativa de las características.

- **Módulo de Atención de Canal (Channel Attention Module - CAM)**: Este módulo se enfoca en "qué" es significativo en la imagen. Para calcular la atención de canal, CBAM primero aplica operaciones de Global Max Pooling y Global Average Pooling en el mapa de características de entrada. Las dos salidas de pooling se pasan a una red MLP compartida para aprender las interdependencias entre canales. Las salidas de la MLP se combinan mediante una suma y se aplican a través de una función sigmoide para generar un mapa de atención de canal, que indica la importancia de cada canal. Presentando 2.14.

$$M_c(F) = \sigma(\text{MLP}(\text{AvgPool}(F)) + \text{MLP}(\text{MaxPool}(F))) \tag{2.14}$$

  donde F es el mapa de características de entrada y σ es la función sigmoide.

- **Módulo de Atención Espacial (Spatial Attention Module - SAM)**: Este módulo se enfoca en "dónde" es significativa la información. Toma como entrada el mapa de características refinado por el CAM. Primero, aplica operaciones de Global Max Pooling y Global Average Pooling a lo largo de la dimensión del canal para agregar información espacial. Las dos salidas de pooling (un mapa de características promediado y un mapa de características máximo a lo largo de los canales) se concatenan y luego se aplica una operación convolucional estándar para generar un mapa de atención espacial (vease la Figura 2.8). Este mapa de atención espacial resalta las ubicaciones espaciales más relevantes. La ecuación es 2.15:

$$M_s(F') = \sigma(\mathrm{f}^{7\times7}([\mathrm{AvgPool}(F'); \mathrm{MaxPool}(F')])) \tag{2.15}$$

donde F' es la característica de entrada (después de la atención de canal), $f^{7\times7}$ denota una operación de convolución con un filtro de tamaño 7×7, y $[\cdot;\cdot]$ es la concatenación.

CBAM refina las características de entrada de forma secuencial, aplicando primero atención de canal y luego atención espacial. Esta doble refinación mejora la capacidad de representación de las características y, consecuentemente, el rendimiento en diversas tareas de visión por computadora, incluyendo la clasificación de imágenes y la detección de objetos (demostrado en los conjuntos ImageNet, CIFAR–100, MS COCO Y VOC 2007) (Woo et al., 2018).

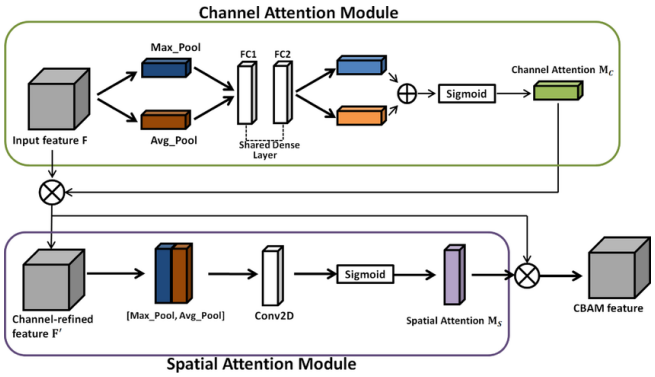


Figura 2.8: Arquitectura del mecanismo de atención CBAM. Imagen tomada de (Alirezazadeh et al., 2022).

2.3. Automatización de tareas manuales de identificación de estomas

El recuento manual de estomas es laborioso, lento y propenso al error humano, lo que ha motivado la automatización mediante técnicas avanzadas de procesamiento de imágenes y DL (Fetter et al., 2019).

El DL tiene una gran eficacia particularmente en el procesamiento de imágenes microscópicas celulares (Moen et al., 2019), donde se han aplicado distintos métodos para abordar los problemas de segmentación y clasificación. En el estudio de Xu et al. (2010), para resolver la segmentación de núcleos celulares en imágenes histopatológicas coloreadas, se aplica una CNN. Básicamente, la segmentación en las imágenes microscópicas se refiere al proceso de encontrar los límites de las células, los núcleos celulares y la estructura histológica con la precisión adecuada (Deshmukh & Mankar, 2014).

Sin embargo, la segmentación de imágenes microscópicas puede presentar distintas complicaciones por motivos como: la alta variabilidad de las células y su estructura, resolución limitada de las imágenes adquiridas, bajo contraste de los límites de los elementos celulares y la pérdida de datos al pasar de un objeto 3D a su imagen 2D (Deshmukh & Mankar, 2014).

El algoritmo PaCeQuant de Möller et al. (2017) permite extraer más de 27 descriptores morfométricos de células de pavimento en imágenes de microscopía 2D. Este software cuantifica automáticamente características como el número de lóbulos, el índice de complejidad y la elongación, mostrando alta concordancia con métodos manuales y facilitando la comparación de mutantes con alteraciones en la forma celular (ej., mutantes en ROP2, RIC4 y SPK1). Sin embargo, PaCeQuant no realiza la segmentación automática de las células, requiriendo que el usuario provea máscaras binarizadas de alta calidad donde las células estén claramente delimitadas.

El estudio de Fetter et al. (2019) presenta StomataCounter, una herramienta que emplea una CNN basada en AlexNet para la segmentación binaria de estomas. Entrenado con diversas especies (incluyendo *Arabidopsis thaliana, Populus balsamifera, Gossypium hirsutum y Picea glauca*), el modelo utiliza parches de imágenes microscópicas etiquetados y data augmentation para mejorar su capacidad de generalización. StomataCounter genera mapas de probabilidad que, mediante un posprocesamiento de detección de máximos locales, permiten identificar la ubicación precisa de cada estoma.

El trabajo de Wolny et al. (2020) describe una herramienta computacional modular para la segmentación 3D precisa de tejidos vegetales a resolución celular. Inicialmente, las imágenes 3D se preprocesan con normalización de intensidad y, opcionalmente, suavizado gaussiano 3D para reducir el ruido. Luego, se utiliza una red U-Net 3D, entrenada de forma supervisada con el conjunto de datos "Truth Annotated Datasets"(imágenes de raíces de *Arabidopsis thaliana* con membranas celulares marcadas manualmente). La segmentación de instancias celulares se logra construyendo un grafo 3D donde los voxels son nodos conectados por aristas ponderadas según la probabilidad de membrana. Finalmente, algoritmos de partición como GASP o Multicut se aplican para separar e identificar cada célula individualmente.

Zhu et al. (2021) desarrollaron un método basado en CNNs para calcular automáticamente el índice estomático en trigo. El enfoque combina Faster R-CNN (para la detección de estomas) con U-Net (para la segmentación de células epidérmicas), alcanzando una precisión del 95,35 % y un coeficiente R^2 superior a 0,98 respecto al conteo manual. Para mejorar la generalización, se utilizó un modelo ResNet101 preentrenado en ImageNet como base.

El trabajo de R. Li et al. (2022) presenta LeafNet, que utiliza CNNs jerárquicas para segmentar y cuantificar células epidérmicas y estomas en *Arabidopsis thaliana*. LeafNet, inicialmente preentrenado con datos de StomataCounter y refinado con anotaciones manuales, se compone de StomaNet (precisión del 98 % en detección de estomas, similar a StomataCounter) y LeafSeg (que supera a PlantSeg y PaCeQuant (Möller et al., 2017; Wolny et al., 2020) en segmentación de células pavimentosas). LeafNet demostró adaptabilidad a diferentes especies y tipos de microscopía.

Complementariamente, X. Li et al. (2022) desarrollaron un modelo YOLOv5 modificado para contar estomas y diferenciar su estado (abierto/cerrado). Este modelo incorpora un mecanismo CA (Hou et al., 2021), que mejora la precisión de detección hasta un 89,4 %. La inclusión de CA es crucial porque permite al modelo enfocarse en las características morfológicas clave de los estomas (como

los bordes del ostíolo y las células oclusivas), mejorando la discriminación entre estomas abiertos y cerrados al capturar tanto las dependencias inter-canal como la información posicional de estas estructuras sutiles.

En resumen, las herramientas basadas en DL —Faster R-CNN (Ren et al., 2017), U-Net (Ronneberger et al., 2015), Mask R-CNN (He et al., 2017), y YOLOv5 modificado (Redmon et al., 2016)— demuestran ventajas importantes en precisión, rapidez y robustez respecto a métodos tradicionales, indicando un gran potencial para aplicarse en esta área.

En este sentido, esta memoria de título tiene por finalidad aportar en la discusión al proponer un pipeline en dos etapas, capaz de identificar el estoma y de la misma forma extraer parametros morfologicos donde se permita cuantificar el estado de apertura abierto/cerrado.

Capítulo 3

Metodología

El desarrollo de modelos de DL exige el uso de conjuntos de datos que exhiban una alta diversidad de condiciones y características. Para cumplir con este requisito, se diseña una metodología de recolección y preparación de datos que integra múltiples fuentes, garantizando la variabilidad necesaria para el entrenamiento y la validación de los modelos. A continuación, se detalla el procedimiento empleado.

3.1. Recolección y Composición de los Datos

La construcción del conjunto de datos para este estudio se basa en tres fuentes distintas. La primera, proporcionada por el proyecto "Phytolearning: Núcleo Milenio en Resiliencia Vegetal", cuyas muestras son sometidas a un protocolo experimental específico que se describe en la siguiente subsección. Las dos fuentes restantes corresponden a conjuntos de datos públicos obtenidos de la plataforma Roboflow (https://roboflow.com/), los cuales son limpiados y modificados para adecuarlos a los objetivos de la presente investigación.

3.1.1. Datos del Proyecto Phytolearning

Este conjunto de datos proviene de un ensayo biológico diseñado para estudiar la respuesta de la planta *Arabidopsis thaliana* al estrés hídrico. A continuación, se describen las condiciones experimentales y el método de obtención de imágenes que realiza el proyecto Phytolearning para colocar a disposición tal conjunto de datos.

3.1.1.1. Tratamientos y Genotipos

Se establecen dos tratamientos principales para modular la condición hídrica de las plantas:

- **Well-watered (Control):** Plantas mantenidas con riego óptimo, asegurando entre un 90 % y un 100 % de la capacidad de campo durante todo el ensayo. Las imágenes de este grupo se identifican con la letra “W” en su nomenclatura.

- **Drought (Sequía):** Plantas sometidas a estrés hídrico mediante la suspensión completa del riego desde el día 0. Las mediciones se realizan en distintos puntos temporales para capturar diferentes niveles de estrés: día 0 (control, condición regada), día 6 (sequía moderada) y día 9 (sequía avanzada o extrema). Las imágenes de este tratamiento se identifican con la letra "D".

Adicionalmente, el ensayo incluye diferentes genotipos para introducir variabilidad genética:

- **Col-0:** Ecotipo silvestre de *Arabidopsis thaliana*, ampliamente utilizado como planta modelo de referencia.
- **hb6, nap, rav1:** Genotipos mutantes para factores de transcripción específicos. El efecto de estas mutaciones a nivel de tejido estomático es aún desconocido, por lo que el análisis de sus estomas es clave para comprender los mecanismos fisiológicos de respuesta a la sequía.

3.1.1.2. Condiciones de Crecimiento y Preparación de Muestras

Adicionalmente se describe el procedimiento que realiza la persona encargada del proyecto Phyto-learning. odas las plantas fueron cultivadas en una cámara de crecimiento (Pitec BIOREF-19L) bajo condiciones controladas: fotoperiodo de 16 horas de luz y 8 horas de oscuridad, temperatura constante de 22 °C, humedad relativa de ± 65 % y una irradiancia PAR de 80 μmol m^{-2} s^{-1}.

Para la obtención de las imágenes microscópicas, se utiliza la técnica de la impronta epidérmica (Klosinska et al., 2016). En los días de muestreo (0, 6 y 9), se toman hojas de las plantas y se aplica resina dental líquida sobre su superficie abaxial. Una vez solidificada la resina, se obtiene una impresión negativa de la epidermis. Sobre esta impronta, se aplica una fina capa de esmalte de uñas transparente. Cuando el esmalte seca, es desprendido cuidadosamente con cinta adhesiva transparente y montado en un portaobjetos para su observación en el microscopio. Es importante destacar que no se utiliza ningún tipo de tinción durante este proceso.

3.1.1.3. Proceso de Etiquetado

El etiquetado de este conjunto de datos presenta desafíos significativos debido a la naturaleza de las muestras obtenidas. La metodología de la impronta epidérmica, aunque efectiva, genera imágenes con artefactos como estomas borrosos o de baja resolución, presencia de burbujas de aire y, en ciertos casos, una difícil distinción del poro estomático. Estas características incrementan la complejidad del etiquetado manual y anticipan dificultades para el entrenamiento de los modelos. En la Figura 3.1 se presenta un ejemplo representativo de estas imágenes.

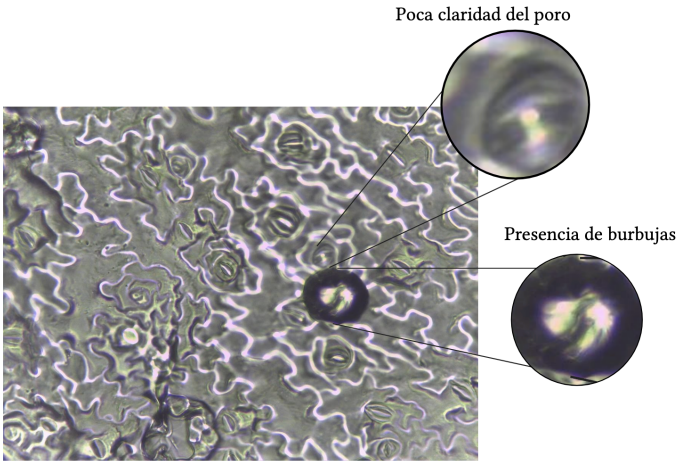


Figura 3.1: Ejemplo de una imagen del "Dataset de Phytolearning", donde se aprecian los desafíos para el etiquetado, como la baja resolución y la presencia de burbujas.

El conjunto de datos corresponde a 150 imágenes microscópicas de *Arabidopsis thaliana* en alta resolución (2048 $\times$ 1536 píxeles), las cuales incluyen un etiquetado original mediante cajas delimitadoras (bounding boxes).

Con el objetivo de entrenar un modelo de segmentación por instancia, se genera un conjunto de datos derivado a partir de este. Dicho proceso consiste en un etiquetado manual utilizando máscaras de segmentación, definidas por polígonos, para delinear con precisión las siguientes clases:

- **stomata:** Polígono que delimita el contorno completo de cada estoma.
- **pore-open:** Polígono que define el contorno del poro estomático cuando se encuentra en estado abierto.
- **pore-closed:** Polígono que define el contorno del poro estomático cuando se encuentra en estado cerrado.

Este etiquetado permite no solo localizar el estoma, sino también clasificar su estado de apertura.

3.1.2. Datos Obtenidos de Roboflow

Para ampliar la diversidad del conjunto de datos, se incorporan dos datasets públicos alojados en Roboflow, una plataforma en línea para el etiquetado y la gestión de datos de CV. Se opta por esta fuente sobre otras como Kaggle tras una búsqueda satisfactoria. Los proyectos en la versión gratuita de Roboflow son públicos, lo que permite el acceso y la descarga de los datos para su uso en otras investigaciones.

3.1.2.1. Datos de *Cicer arietinum*

El conjunto de datos extraído corresponde a 318 imágenes microscópicas de estomas de *Cicer arietinum*, con una resolución de 640×640 píxeles. Una limitación de este dataset es la ausencia de metadatos experimentales; no se encuentra información sobre los tratamientos aplicados, el equipo de microscopía utilizado u otras condiciones de la muestra.

El dataset original incluye las clases "guard" (estomas) y "pore" (poro). Sin embargo, mediante la revisión manual revela inconsistencias sistemáticas en el etiquetado. Como se observa en la Figura 3.2, se encuentran múltiples instancias donde la región etiquetada como estoma no contiene el poro central. Asimismo, se identifican estomas incompletos o cortados en los bordes de la imagen, introduciendo ruido en el conjunto de datos.

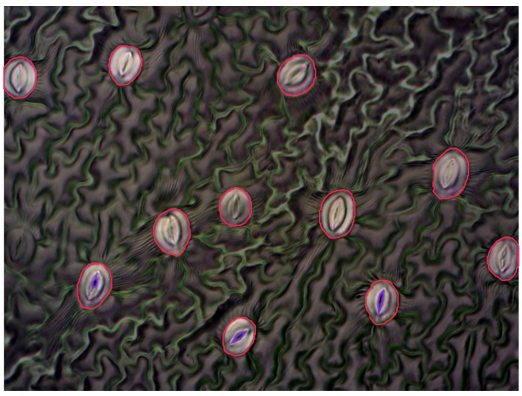


Figura 3.2: Muestra del etiquetado original en "Dataset de *Cicer arietinum*", donde se aprecian inconsistencias como la identificación de estomas sin poro central.

Debido a las inconsistencias detectadas, durante esta memoria de título se realiza un proceso de limpieza y re-etiquetado manual del conjunto completo de 318 imágenes. Este procedimiento tiene como objetivo no solo corregir los errores del etiquetado original, sino también enriquecer el dataset con anotaciones más detalladas. Para ello, se utilizan máscaras de segmentación para delimitar las siguientes clases: **stomata** (delimitación del contorno completo del estoma), **pore-open** (delimitación del poro estomático en estado abierto), **pore-closed**, (delimitación del poro estomático en estado cerrado). De esta forma, se asegura que cada estoma estuviera correctamente delineado y asociado a su respectivo poro.

3.1.2.2. Datos de *Arabidopsis thaliana* mediante Apilamiento de Enfoque

Adicionalmente, se incorpora un tercer conjunto de datos de *Arabidopsis thaliana*, también proveniente de la plataforma Roboflow. Al igual que el conjunto de *Cicer arietinum*, este carece de metadatos detallados sobre el microscopio utilizado o los tratamientos aplicados a las muestras.

No obstante, este dataset destaca por su innovador método de obtención de imágenes, diseñado para superar un desafío común en microscopía: la curvatura natural de la superficie foliar, que a menudo

impide obtener una imagen completamente nítida en una sola toma. Para solucionar esto, las imágenes no son fotogramas estáticos, sino el resultado de una técnica conocida como **apilamiento de enfoque (focus stacking)** (Clark & Brown, 2015). El proceso consiste en grabar un video corto de la impronta foliar mientras se varía continuamente el plano focal. Posteriormente, un algoritmo computacional analiza los fotogramas, extrae las regiones más nítidas de cada uno y las combina para generar una única imagen compuesta, completamente enfocada en todo el campo visual.

El conjunto de datos resultante consta de 991 imágenes con una resolución de 1248×928 píxeles. Además de su alta calidad de enfoque, este dataset introduce desafíos valiosos para la robustez del modelo, como una notable variabilidad en la saturación y una alta densidad estomática, como se ilustra en la Figura 3.3. Originalmente, el dataset es proporcionado con anotaciones de cajas delimitadoras (bounding boxes) para la clase `stomata`.

Con el fin de adaptarlo para el entrenamiento de modelos de segmentación por instancia, se selecciona un subconjunto de estas imágenes y se somete a un nuevo proceso de etiquetado manual. Se seleccionan 150 imágenes de las cuales se generan máscaras de segmentación poligonales para definir con precisión las clases `stomata`, `pore-open` y `pore-closed`.

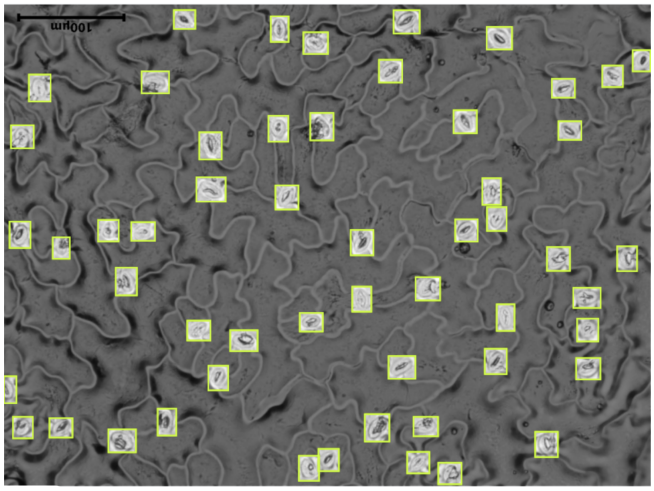


Figura 3.3: Muestra del "Dataset de *Arabidopsis thaliana*" obtenido por apilamiento de enfoque. Se evidencian los desafíos característicos del conjunto, como la alta densidad estomática y la variabilidad en la saturación de la imagen y poca claridad del poro.

3.2. Estrategia de Construcción de los Conjuntos de Datos

Para abordar los distintos objetivos de este estudio, se diseña una estrategia de construcción de datasets que parte de las tres fuentes de datos primarias descritas previamente. A partir de estas, se generan conjuntos de datos especializados y derivados, cada uno con un tipo de etiquetado específico para las tareas de detección de objetos y segmentación por instancia. El flujo de trabajo para la generación de estos conjuntos de datos se ilustra esquemáticamente en la Figura 3.4. El proceso comienza

con imágenes de estomas etiquetadas mediante cajas delimitadoras; luego, se genera un conjunto con máscaras de segmentación para los estomas mediante automatización de etiquetas. Finalmente, se crean tres conjuntos derivados que varían en sus clases: uno con máscaras de segmentación para 'stomata', 'pore-open' y 'pore-closed'; otro con etiquetado solo del estoma y el poro mediante máscaras de segmentación; y un tercero con las clases 'stomata-open' y 'stomata-closed', donde el estoma se delimita incorporando su estado de apertura, sin una segmentación independiente del poro.

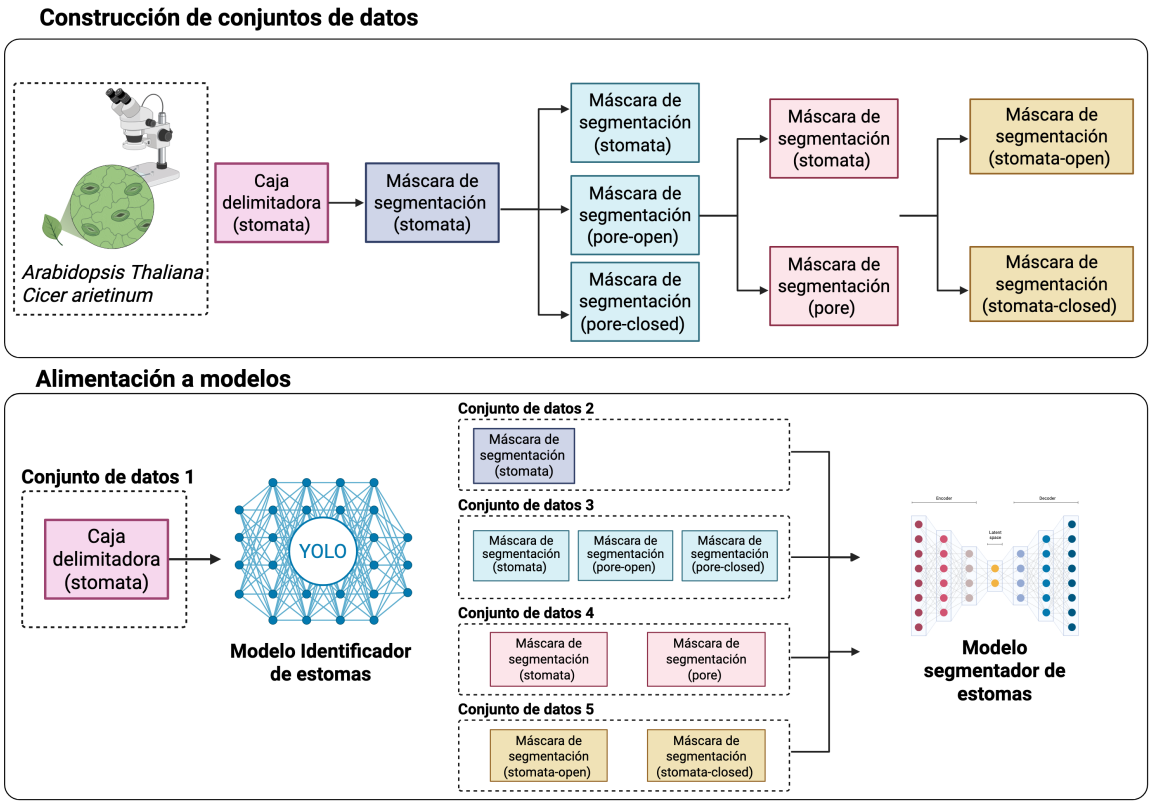


Figura 3.4: Diagrama de la estrategia de construcción de datasets. Se muestran las fuentes de datos primarias y su derivación en conjuntos específicos para los modelos de detección y segmentación, indicando el tipo de anotación en cada caso.

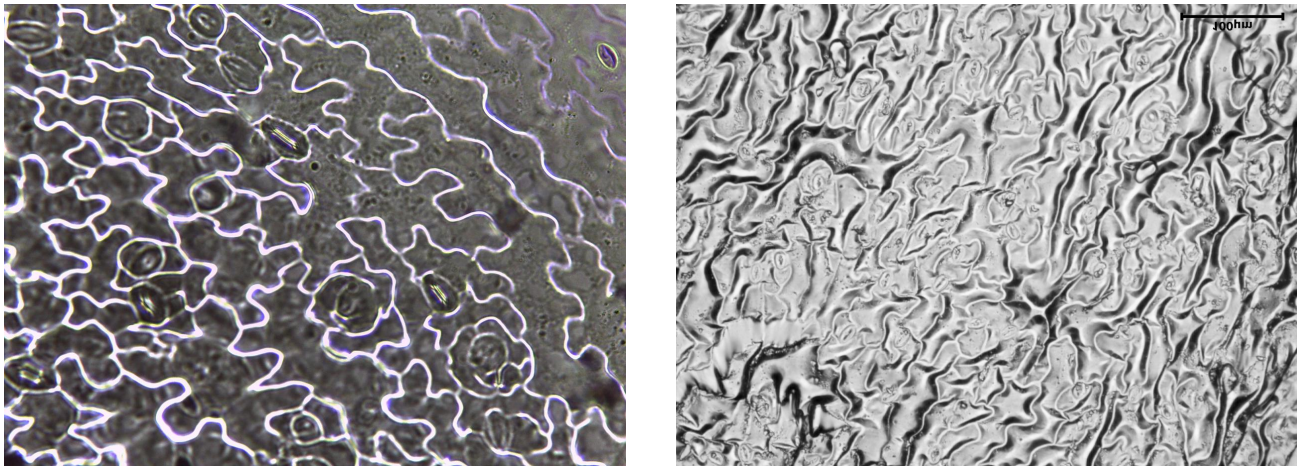
3.2.1. Conjunto de datos para el Modelo de Detección de Estomas

Este conjunto de datos es diseñado específicamente para la tarea de detección de objetos, cuyo objetivo es localizar estomas mediante cajas delimitadoras (bounding boxes). Para su construcción, se unifican dos de las fuentes de datos que ya contaban con este tipo de anotación: el dataset completo de Phytolearning y el de *Arabidopsis thaliana*. Este enfoque permite aprovechar las etiquetas preexistentes sin necesidad de un re-etiquetado manual.

El dataset consolidado suma un total de 1141 imágenes. Es particionado siguiendo una distribución estándar de **80 % para entrenamiento, 10 % para validación y 10 % para prueba**. Es fundamental destacar que el conjunto de prueba se reserva exclusivamente para la evaluación final del pipeline com-

pleto, garantizando que el modelo nunca fuera expuesto a estos datos durante las fases de entrenamiento o ajuste de hiperparámetros.

Figura 3.5 presenta ejemplos visuales que ilustran la diversidad de este conjunto de datos, el cual combina las características de ambas fuentes para entrenar un modelo de detección robusto. Vease a) donde la luminosidad es diferente a la de b), siendo esta ultima similar a escala de grises. Además de la presencia de burbujas y tamaños de los estomas difieren significativamente en ambas imágenes.



(a) Muestra de conjunto de datos proporcionado Phytolearning

(b) Muestra de conjunto de datos con Apilamiento de Enfoque

Figura 3.5: Ejemplos de imágenes del conjunto de datos para el modelo de detección. Se aprecia la variabilidad visual proveniente de las distintas fuentes de datos (Phytolearning y apilamiento de enfoque).

3.2.2. Conjuntos de Datos para el Modelo de Segmentación por Instancia

Para el entrenamiento de los modelos de segmentación por instancia, se diseña una estrategia que implica la construcción de un conjunto de datos base y la posterior derivación de variantes experimentales. El objetivo es entrenar un modelo capaz no solo de segmentar el contorno del estoma, sino también de identificar y clasificar el estado de su poro.

3.2.2.1. Composición del Conjunto de Datos Base

Se construye un conjunto de datos principal mediante la integración de tres fuentes de datos, todas ellas etiquetadas manualmente con máscaras de segmentación. Este conjunto base está compuesto por un total de 618 imágenes, distribuidas de la siguiente manera:

- 318 imágenes de *Cicer arietinum*.
- 150 imágenes de *Arabidopsis thaliana* (obtenidas por apilamiento de enfoque).
- 150 imágenes del proyecto Phytolearning.

Para cada imagen, se generan polígonos de segmentación para tres clases específicas: `stomata`, `pore-open` y `pore-closed`.

3.2.2.2. Proceso de Limpieza y Control de Calidad

Tras el etiquetado manual, se implementa un protocolo de control de calidad para garantizar la integridad lógica de las anotaciones. Se establece como criterio que cada instancia de la clase `stomata` deba contener obligatoriamente una instancia anidada de `pore-open` o `pore-closed`. Todas las imágenes o anotaciones que no cumplan con esta condición topológica fueron eliminadas del conjunto de datos, asegurando así la coherencia de las etiquetas para el entrenamiento.

3.2.2.3. Generación de Conjuntos de Datos Experimentales

Con el fin de realizar un análisis sobre cómo diferentes esquemas de etiquetado y la calidad de los datos influyen en el rendimiento del modelo, se generan varios conjuntos de datos derivados a partir del conjunto base. Las principales variantes experimentales son:

- **Dataset-Final:** El conjunto original de 618 imágenes con las clases `stomata`, `pore-open` y `pore-closed`.
- **Dataset-Final-Merged:** El conjunto original de 618 imágenes con las clases `stomata` y `pore`. Las clases `pore-open` y `pore-closed` se fusionan en una única clase `pore`, para evaluar un modelo que solo distingue entre estoma y poro, sin clasificar el estado de este último.
- **Dataset-Final-Filtered (Sin Phytolearning):** en base al Dataset Final Merged se crea este dataset sin considerar el conjunto de datos de Phytolearning. Se piensa que la calidad del etiquetado es más precario y puede estar afectando la convergencia del modelo.
- **Dataset-Filtered-Stomata-State:** Se crean dos nuevas clases, combinando la información del contorno del estoma con el estado de su poro (`stomata-open`) y (`stomata-closed`. El objetivo es entrenar un modelo que clasifique directamente el estoma completo y su estado abierto/cerrado. De igual forma no se considera al dataset de phytolearning.
- **Dataset-Single-stomata:** Se genera una versión del dataset "Dataset-Filtered-Stomata-State" mediante la aplicación de una herramienta de recorte que calcula la caja delimitadora más pequeña que contiene la máscara de segmentación. El objetivo es obtener imágenes recortadas que contengan un único estoma, el cual esta etiquetado como 'stomata-open' o 'stomata-closed'. La finalidad de crear este dataset radica en generar más imágenes, buscando así evitar el sobreajuste del modelo ante un número limitado de muestras. El procedimiento automatizado consiste en lo siguiente: para cada estoma etiquetado con un polígono de segmentación, primero se calcula su caja delimitadora mínima. Posteriormente, esta caja se utiliza como plantilla para recortar una sub-imagen de la imagen original. La máscara de segmentación se conserva intacta dentro de cada recorte, ajustando sus coordenadas al nuevo marco de referencia de la imagen generada. Esta estrategia transforma el conjunto de datos en uno nuevo de **6.416 imágenes**. Cada una de

estas imágenes contiene un único estoma, presenta una resolución variable (correspondiente al tamaño de su caja delimitadora) y hereda la etiqueta de su instancia original: `stomata-open` o `stomata-closed`.

3.3. Modelos y Estrategia Experimental

Para el desarrollo de la solución propuesta, se abordan dos tareas computacionales distintas: la **detección de objetos** (Object Detection) y la **segmentación por instancia** (Instance Segmentation). Para cada tarea, se diseñan y entrenan modelos especializados, utilizando arquitecturas, conjuntos de datos y configuraciones de hiperparámetros específicos. Esta sección detalla la metodología experimental seguida para cada uno de los modelos.

Para la ejecución de los experimentos se utiliza un nodo GPU dentro de un clúster de supercomputación, el cual corre bajo el sistema operativo Linux 5.14.0 (RHEL/CentOS 9). Este nodo está equipado con dos procesadores Intel Xeon Silver 4416+ (arquitectura Sapphire Rapids), configurados con un total de 40 núcleos físicos (20 núcleos por procesador y Hyper-Threading desactivado). Complementando esta capacidad de CPU, el nodo dispone de 257 GB de RAM y una GPU NVIDIA RTX A5000 con 24 GB de VRAM. Gran parte de los entrenamientos se realizan en la GPU, aprovechando su arquitectura para la computación paralela. Los modelos se entrenan en lenguaje Python 3.12. Mientras que las bibliotecas mayormente utilizadas para el preprocesamiento, entrenamiento y análisis son: OpenCV, ultralytics, pytorch, matplotlib y seaborn.

3.3.1. Modelo de Detección de Estomas

El objetivo de este primer modelo es la detección de estomas mediante cajas delimitadoras (*bounding boxes*). Para esta tarea, la experimentación se centra mayoritariamente en la arquitectura YOLOv11, explorando sus distintas escalas de complejidad (desde "nano" hasta "xlarge").

3.3.1.1. Estrategia Experimental y Optimización de Hiperparámetros

Se diseña una estrategia experimental iterativa para identificar la mejor configuración. Se utiliza el conjunto de datos unificado de 1.141 imágenes, particionado en 913 para entrenamiento y 115 para validación, se establece una configuración base para la mayoría de los experimentos: 100 épocas de entrenamiento, sin aumento de datos (*data augmentation*) y sin *early stopping*, con un momentum de 0,937.

La estrategia experimental se desarrolla en las siguientes fases:

1. **Establecimiento de la Línea Base:** El proceso se inicia con la arquitectura más ligera, YOLOv11n, para establecer una línea base de rendimiento y consumo de recursos. Se utiliza un

tamaño de lote (`batch_size`) pequeño de 8 y un tamaño de imagen de 512x512 píxeles. Se selecciona el optimizador Adam debido a su reportada estabilidad en la literatura, con una tasa de aprendizaje (`learning_rate`) inicial de 0,001.

2. **Escalado Incremental:** Tras confirmar la viabilidad computacional del entrenamiento, se procede a un escalado incremental. Se aumenta el `batch_size` a 32 y el `image_size` a 640x640 para aprovechar la capacidad de paralelización del hardware. Dado el buen rendimiento obtenido, se exploran progresivamente arquitecturas más complejas, hasta llegar al modelo YOLOv11x con 56,8 millones de parámetros.

3. **Análisis Comparativo:** Durante la exploración, se realizan pruebas comparativas para validar las elecciones de hiperparámetros. Un experimento con el optimizador SGD muestra un rendimiento inferior en comparación con Adam. Asimismo, una prueba con una tasa de aprendizaje más alta (0,005) genera una notable inestabilidad en la curva de pérdida, afectando negativamente la convergencia del modelo.

Este proceso permite concluir que la combinación más efectiva de hiperparámetros para esta tarea consiste en: un `batch_size` de 16, `image_size` de 640x640, `learning_rate` de 0,001 y el optimizador Adam. Cabe destacar que, si bien el entrenamiento involucra numerosos hiperparámetros adicionales, el análisis se centra en aquellos que se consideraran más influyentes para el rendimiento del modelo. Como se visualiza en la Tabla 3.1 la primera fila indica el nombre del experimento realizado mientras que en las columnas se ven los valores de la configuración de distintos hiperparámetros y recursos utilizados.

Tabla 3.1: Configuración de hiperparámetros y recursos para los experimentos del modelo de detección. Los modelos se abrevian en las cabeceras (ej. "n-exp1" corresponde a "YOLOv11n_exp1").

Modelos	**n-exp1**	**s-exp1**	**s-exp2**	**m-exp1**	**l-exp1**	**l-exp2**	**l-exp3**	**x-exp1**	**x-exp2**
Configuración de Hiperparámetros en el Entrenamiento									
Batch Size	8	32	32	32	32	16	16	8	16
Image Size	512	640	640	640	640	640	640	640	640
Learning Rate	0,001	0,001	0,001	0,001	0,001	0,005	0,001	0,001	0,001
Momentum	0,937	0,937	0,937	0,937	0,937	0,937	0,937	0,937	0,937
Optimizador	Adam	Adam	SGD	Adam	Adam	Adam	Adam	Adam	Adam
Recursos									
Parámetros	2,5M	9,4M	9,4M	20,0M	25,2M	25,2M	25,2M	56,8M	56,8M
Recurso Comp.	CPU (Intel Xeon Silver)	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000

Tras una evaluación de las métricas de rendimiento (que se detallarán en el capítulo de resultados), se selecciona la configuración del experimento **YOLOv11x_exp2** como la mejor para el modelo de

detección final. Dado el alto rendimiento general obtenido en estas pruebas, se determina que su balance entre precisión y complejidad computacional es el más adecuado, por lo que no se considera necesario explorar modificaciones de arquitecturas más profundas.

3.3.2. Modelo de Segmentación

Para la tarea de segmentar estomas y, simultáneamente, clasificar su estado (abierto o cerrado), se diseña una estrategia experimental. El objetivo es identificar la mejor combinación de datos, arquitectura e hiperparámetros. La investigación se divide en dos fases principales:

1. Una fase de **selección del conjunto de datos** más efectivo, manteniendo un modelo y una configuración base constantes.

2. Una fase de **optimización del modelo** sobre el dataset más robusto, explorando hiperparámetros y modificaciones de la arquitectura base.

3.3.2.1. Fase 1: Selección del Mejor Conjunto de Datos

Para aislar el impacto de la estructura y calidad de los datos, se diseña un experimento comparativo utilizando cinco variantes de datasets. En todos los casos, se emplea la misma arquitectura (YOLOv11x, con 62 millones de parámetros) y una configuración de hiperparámetros base (100 épocas, optimizador Adam y learning rate 0,001), como se detalla en la Tabla 4.4 notamos en las filas la variación de dataset en conjunto con los hiperparametros y características de etiquetado, así mismo en las columnas el nombre del experimento respectivo, presente en la sección 4.2.

Este análisis concluye que el **Dataset-Single-Stomata (DS5)** es la configuración de datos más prometedora. Por lo tanto, es seleccionado para las fases subsiguientes de optimización, para más detalles revisar la sección 4.2 .

3.3.2.2. Fase 2: Optimización del Modelo sobre el Dataset Seleccionado

Ya establecido el "Dataset-Single-Stomata" como el más efectivo, la segunda fase de la experimentación se centra en optimizar la arquitectura y los hiperparámetros para maximizar su rendimiento..

Optimización de Hiperparámetros Se lleva a cabo una exploración variando parámetros clave como la arquitectura (YOLOv11 y v12), el batch size, el learning rate, el optimizador y el uso de aumento de datos. La Tabla 3.2 detalla las configuraciones probadas, teniendo en la primera fila los nombres de los experimentos realizados y en el resto, la configuración de hiperparámetros durante el entrenamiento.

Tabla 3.2: Configuración y resultados de los experimentos del modelo de segmentación. Se abrevia el nombre (ej. "11-recorted" corresponde a "YOLOv11x_recorted" y "12-exp1" es "Yolo12x_seg_exp1") .

Modelos	**11_recorted**	**11_ULT4**	**11_ULT5**	**12_exp1**	**12_exp5**	**12_exp6**	**12_exp7**	**12_exp8**
Configuración del Entrenamiento								
Augment	No	No	No	No	Si	Si	Si	Si
Batch Size	64	64	16	16	64	8	8	8
Image Size	128	128	640	128	128	640	640	640
Learning Rate	0,001	0,01	0,01	0,01	0,0017	0,0017	0,01	0,0017
Optimizador	Adam	SGD (auto)	SGD (auto)	SGD	AdamW	AdamW	SGD	AdamW
Momentum	—	0,93	0,9	0,9	0,9	0,9	0,9	0,9
Early Stopping	No	25	25	10	5	15	25	30
Epochs	100	137/150	115/150	74/150	32/100	100/100	124/350	86/350
Dataset				Dataset-Single-Stomata				
Recursos								
Recurso Comp.	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000

Integración de Mecanismos de Atención. Paralelamente, se explora la integración de mecanismos de atención para potenciar la capacidad del modelo. Se evaluan dos módulos: **CBAM** y **(CA)**. Los experimentos con CBAM muestran una severa inestabilidad numérica, atribuida a la explosión de gradientes. En cambio, la variante C3CA (Coordinate Attention adaptada al bloque C3 de YOLO) demuestra ser estable y robusta, convirtiéndose en la arquitectura preferida para los experimentos finales de alto rendimiento, como se detalla en la Tabla 3.3, se visualiza que en la primera columna estan los distintos hiperparametros que varian su valor dependiendo el nombre del experimento.

Tabla 3.3: Configuración y resultados de los experimentos con mecanismos de atención.

Modelos	**Exp. 1**	**Exp. 2**	**Exp. 3**	**Exp. 4**	**Exp. 5**	**Exp. 6**
Configuración del Entrenamiento						
Módulo de Atención	CBAM	CBAM	C3CA	C3CA	C3CA	C3CA
Augment	Si	Si	Si	Si	No	Si
Batch Size	16	16	16	4	4	4
Image Size	640	640	640	1056	1056	1056
Learning Rate	—	0,01	0,0017	0,0001	0,0001	0,0001
Optimizador	AdamW	SGD	AdamW	AdamW	AdamW	AdamW
Early Stopping	25	No	5	25	25	25
Epochs	135/200	—	28/100	121/150	79/150	192/200
Recursos						
Recurso Comp.	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000
Notas	Loss NaN desde ep. 56	Loss NaN y métricas 0	—	—	—	—

Tras una evaluación y comparación cuantitativa de las métricas de rendimiento de todas las configuraciones experimentales (presentadas en el capítulo de Resultados), se selecciona el modelo y los hiperparámetros del experimento **yolov11x_coordAtt_exp6** como la configuración final para el modelo de segmentación.

3.3.3. Métricas de Evaluación para Modelos de Detección y Segmentación

La evaluación del rendimiento de los modelos de DL en tareas de clasificación, detección de objetos y segmentación de instancias es crucial. Para ello, se utilizan diversas métricas que cuantifican la precisión y efectividad de las predicciones del modelo. En este contexto, se definen los siguientes términos:

- **Verdaderos Positivos (TP - True Positives)**: Instancias positivas que son correctamente predichas como positivas por el modelo.
- **Verdaderos Negativos (TN - True Negatives)**: Instancias negativas que son correctamente predichas como negativas por el modelo.
- **Falsos Positivos (FP - False Positives)**: Instancias negativas que son incorrectamente predichas como positivas por el modelo (errores de Tipo I).
- **Falsos Negativos (FN - False Negatives)**: Instancias positivas que son incorrectamente predichas como negativas por el modelo (errores de Tipo II).

A continuación, se presentan las métricas más relevantes (Padilla et al., 2020):

3.3.3.1. Métricas Basadas en la Matriz de Confusión

Recall (Sensibilidad o Tasa de Verdaderos Positivos - TPR): Mide la proporción de verdaderos positivos que fueron identificados correctamente por el modelo, es decir, de todas las instancias positivas reales, cuántas fueron detectadas.Se calcula como se aprecia en la Ecuación 3.1.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.1}$$

Precision (Precisión): Mide la proporción de predicciones positivas que fueron correctas. Es la capacidad del modelo para evitar falsos positivos, vease en la Ecuación 3.2.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.2}$$

F1-score (Puntuación F1): Es la media armónica de la Precisión y el Recall. Proporciona un equilibrio entre ambas métricas, siendo útil cuando se busca un balance entre falsos positivos y falsos negativos. Como se ve en la Ecuación 3.3.

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.3}$$

3.3.3.2. Métricas para Detección de Objetos y Segmentación

Average Precision (AP): Representa el área bajo la curva de Precisión-Recall. AP es una métrica clave en la detección de objetos, calculando el promedio de precisión sobre diferentes umbrales de recall. Se calcula como la integral de la curva de Precisión-Recal, vease la Ecuación 3.4.

$$\text{AP} = \int_0^1 P(R)dR \tag{3.4}$$

Mean Average Precision (mAP): Es la media de la Average Precision (AP) calculada para todas las clases en el conjunto de datos. Es una métrica estándar para evaluar detectores de objetos multi-clase.

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} \text{AP}_i \tag{3.5}$$

donde N es el número total de clases y AP_i es el Average Precision para la clase i como se muestra en la Ecuación 3.5.

mAP@0.50 (mAP@50): Es el mAP calculado con un umbral de Intersección sobre Unión (IoU) de 0,50. Un IoU de 0,50 significa que para que una detección se considere correcta, su *bounding box* debe superponerse al menos un 50 % con la *ground truth*.

mAP@.50:.05:.95 (mAP@.50:.95 o mAP@95): Es el promedio del mAP calculado para varios umbrales de IoU, desde 0,50 hasta 0,95 en incrementos de 0,05. Esta métrica es más estricta y penaliza las localizaciones de objetos menos precisas como se ve en la Ecuación 3.6.

$$\text{mAP@.50:.95} = \frac{1}{10} \sum_{k \in \{0{,}50, 0{,}55, \ldots, 0{,}95\}} \text{mAP}_k \tag{3.6}$$

donde mAP_k es el mAP para un umbral IoU de k.

Curva Precision-Recall (P-R Curve): Es una gráfica que muestra la relación entre la Precisión y el Recall para diferentes umbrales de confianza del modelo. Permite visualizar el *trade-off* entre estas dos métricas. Un área mayor bajo la curva P-R indica un mejor rendimiento.

Capítulo 4

Resultados y Discusión

Este capítulo presenta y analiza los resultados obtenidos de los experimentos descritos en la metodología. El análisis se estructura en dos secciones principales, correspondiendo a los modelos de detección y segmentación, respectivamente.

4.1. Resultados para el Modelo de Detección

A continuación, se presentan los resultados para el modelo de detección de estomas.

4.1.1. Análisis Exploratorio del Conjunto de Datos de Detección

Para comprender las características y la distribución de los datos utilizados en el modelo de detección, se realiza un análisis exploratorio en dos etapas. Primero, se analizan las dos fuentes de datos ("Phytolearning" y "Apilamiento de enfoque") de forma individual para cuantificar sus diferencias. Posteriormente, se evalua el conjunto de datos unificado.

4.1.1.1. Caracterización de las Fuentes de Datos Individuales

El análisis de las fuentes por separado revela una marcada heterogeneidad. Como se detalla en la Tabla 4.1, existe una disparidad significativa tanto en el número total de objetos como en su densidad por imagen. El conjunto de "Apilamiento de enfoque" no solo aporta la mayoría de anotaciones (46.562 frente a 2.719), sino que también presenta un promedio de estomas por imagen considerablemente mayor (47,95) y con mayor variabilidad (desv. est. de 22,19).

Tabla 4.1: Estadísticas descriptivas del número de estomas por imagen para cada fuente.

Fuente	Nº Imágenes	Promedio	Desv. Est.	Mín.	P25	Mediana	P75	Máx.
Phytolearning	150	18,13	4,68	1	15,0	18,0	21,75	30
ADE	971	47,95	22,19	1	34,0	43,0	57,00	176

Esta diferencia en la densidad estomática se ilustra visualmente en la Figura 4.1. Esta heterogeneidad justifica la necesidad de un modelo robusto, capaz de operar eficazmente en imágenes con distribuciones de objetos muy distintas.

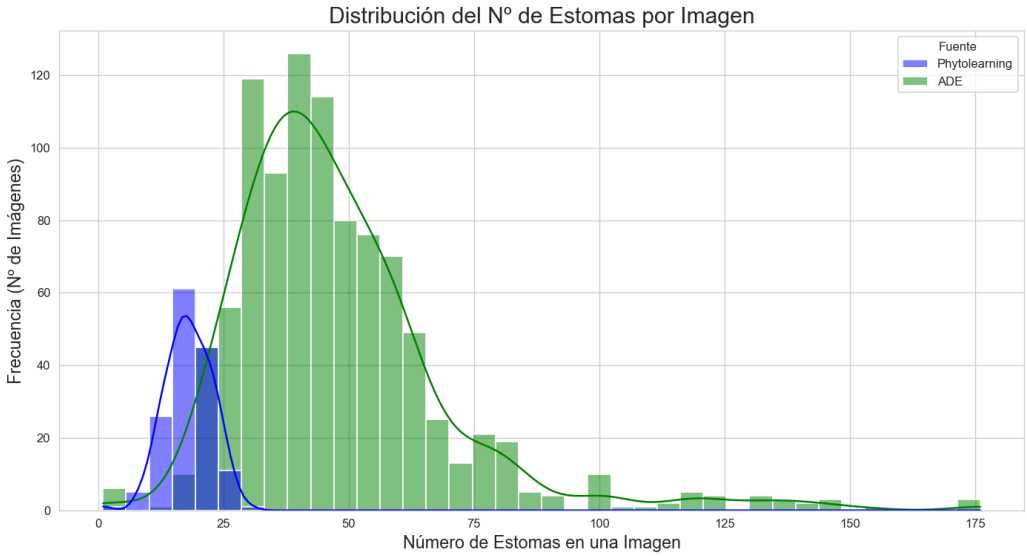


Figura 4.1: Comparación de la distribución de la densidad estomática (número de estomas por imagen) entre los conjuntos de datos "Phytolearning" y "ADE".

4.1.2. Análisis del Conjunto de Datos Unificado

Una vez consolidadas ambas fuentes, el conjunto de datos final para la detección de objetos contiene un total de 49.281 anotaciones. Como era de esperar, las estadísticas descriptivas del conjunto unificado, presentadas en la Tabla 4.2, reflejan la influencia predominante del dataset ADE debido a su mayor tamaño.

Tabla 4.2: Estadísticas descriptivas generales del conjunto de datos unificado para la detección.

Característica	Descripción	Valor
Conteo General	Nº Total de Anotaciones	49.281
Estomas por Imagen	Promedio	43,96
	Mediana	41,00
	Mínimo	1
	Máximo	176
Dimensiones de la caja (px)	Ancho (Promedio, Mín, Máx)	46,80, 2,44, 124,31
	Alto (Promedio, Mín, Máx)	44,02, 8,00, 111,17
	Ratio de Aspecto (Promedio)	1,11

La distribución del número de estomas por imagen en este conjunto combinado, mostrada en la

Figura 4.2, presenta un sesgo positivo, con una larga cola hacia valores altos, lo cual confirma la variabilidad que el modelo debe gestionar.

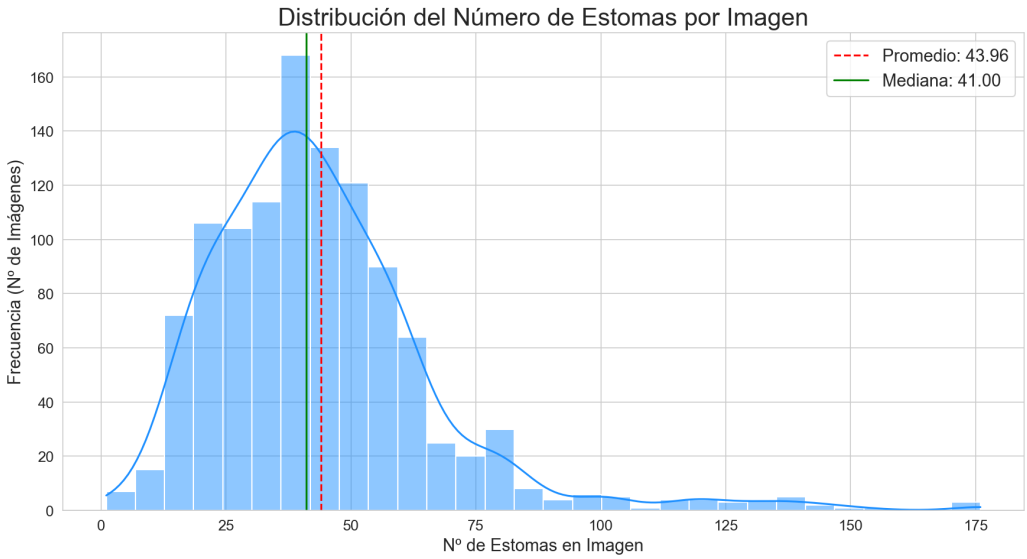


Figura 4.2: Distribución del número de estomas por imagen en el conjunto de datos unificado.

4.1.3. Resultados de experimentos para el Modelo de Detección

El objetivo de esta serie de experimentos es identificar la configuración de modelo e hiperparámetros que ofrezca el mejor rendimiento para la tarea de detección de estomas. La Tabla 4.3 resume las métricas de validación obtenidas para cada uno de los nueve experimentos realizados.

Tabla 4.3: Métricas de validación para los experimentos del modelo identificador.

Modelos	**n-exp1**	**s-exp1**	**s-exp2**	**m-exp1**	**l-exp1**	**l-exp2**	**l-exp3**	**x-exp1**	**x-exp2**
Métricas en la validación									
Precisión (P)	0,939	0,952	0,944	0,952	0,949	0,952	0,949	0,957	0,954
Recall (R)	0,900	0,938	0,94	0,952	0,959	0,952	0,953	0,954	0,956
mAP@.50	0,965	0,982	0,980	0,984	0,985	0,984	0,983	0,983	0,984
mAP@.50:.95	0,615	0,664	0,654	0,673	0,675	0,663	0,674	0,677	0,676
Recursos									
Parámetros	2,5M	9,4M	9,4M	20,0M	25,2M	25,2M	25,2M	56,8M	56,8M
Tiempo [Horas]	18,9	0,204	0,197	0,387	0,502	0,502	0,501	0,871	0,842
Recurso Comp.	CPU (Intel Xeon Silver)	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000

4.1.4. Análisis Comparativo de Experimentos

Un análisis general de los resultados revela varias tendencias clave. Primero, se observa una correlación positiva entre la escala del modelo (número de parámetros) y el rendimiento general, especialmente en la métrica más estricta, mAP@[.50:.95]. Segundo, todos los modelos demuestran un rendimiento alto en mAP@.50, indicando que la tarea de localizar estomas con un criterio de superposición moderado es abordada con gran eficacia por estas arquitecturas.

4.1.4.1. Análisis del Modelo con mejor renidmiento: YOLOv11x-exp2

Tras la evaluación, el modelo del experimento **YOLOv11x-exp2** emerge como la configuración con el rendimiento más robusto y equilibrado. Los resultados en el conjunto de validación indican un alto grado de fiabilidad. Teniendo una **precisión (P) de 0,954**, lo que significa que el 95,4 % de las detecciones realizadas por el modelo fueron correctas, un **recall (R) de 0,956**, demostrando que el modelo fue capaz de identificar el 95,6 % de todos los estomas realmente presentes, un valor de **mAP@.50 de 0,984**, que confirma un alto rendimiento mediante el criterio de una superposición (IoU) del 50 %. Sin embargo, la métrica más exigente, **mAP@[.50:.95]**, que promedia el rendimiento a través de umbrales de IoU crecientes, registra un valor de 0,676. Esta discrepancia subraya que, si bien la localización general de los estomas es excelente, la precisión en el ajuste fino de la caja delimitadora es el principal factor que limita el rendimiento bajo los criterios más estrictos.

4.1.4.2. Análisis de la Dinámica de Entrenamiento

Las curvas de entrenamiento y validación para el mejor modelo se presentan en la Figura 4.3. El análisis de las curvas de pérdida (*loss*) revela fluctuaciones, particularmente en las curvas de validación para las pérdidas de localización (`box_loss`). A pesar de estas oscilaciones, se observa una clara tendencia a la baja en todas las métricas de pérdida, indicando que el modelo converge adecuadamente.

Estas fluctuaciones son consistentes con el valor obtenido en la métrica mAP@[.50:.95]. Sugieren que el modelo encuentra dificultades en el ajuste de alta precisión de las cajas delimitadoras para ciertas muestras complejas, lo que es penalizado por los umbrales de IoU más altos y se refleja como ruido en la curva de pérdida de validación.

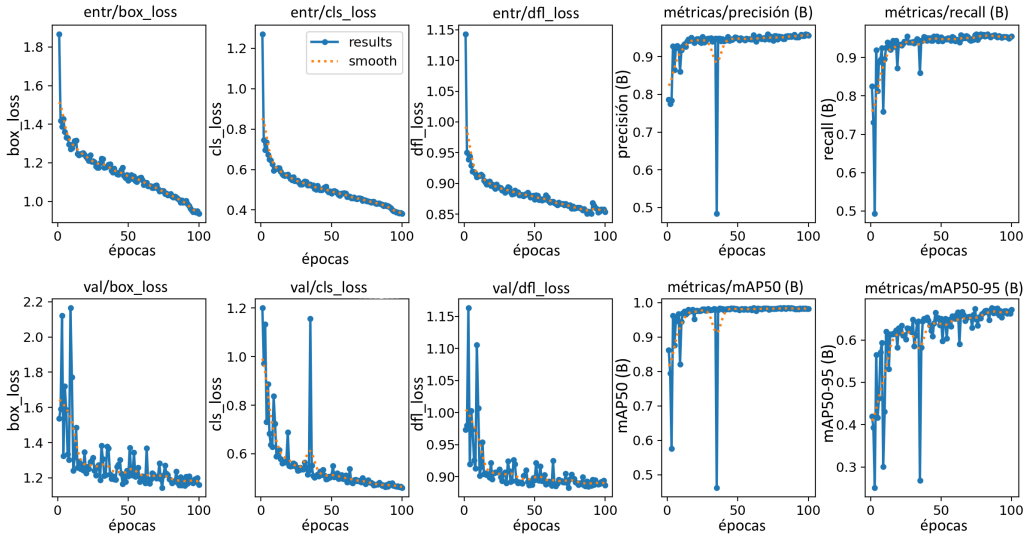


Figura 4.3: Curvas de pérdida en el conjunto validación y entrenamiento para el modelo identificador.

4.2. Resultados Modelo Segmentador

Sección presenta los resultados en base al modelo segmentador. Además, la selección del dataset para el mejor modelo, es "DS5".

El análisis de resultados para elección del dataset revela un progreso incremental:

- **Experimento 1 (DS1):** El dataset base, con tres clases, obtiene el rendimiento más bajo como se ve en la Tabla 4.4. Se atribuye este resultado al severo desequilibrio de clases, donde la subrepresentación de las clases `pore-closed` y `pore-open` dificulta el aprendizaje. Es decir, la cantidad de instancias presenta un desequilibrio donde cada clase stomata esta repartida en pore-open y pore-closed.

- **Experimento 2 (DS2):** Al fusionar los poros en una sola clase (`pore`), se equilibra el número de instancias por clase, lo que resulta en una mejora significativa de las métricas, validando la hipótesis de que el desequilibrio es un factor limitante.

- **Experimento 3 (DS3):** La eliminación de las imágenes de Phytolearning, aunque teóricamente mejora la calidad promedio, redujo el rendimiento. Esto sugiere que, en esta etapa, la cantidad de datos es más crítica para la generalización que la calidad individual de las muestras.

- **Experimento 4 (DS4):** Al simplificar la tarea (clasificar el estado del estoma sin segmentar el poro), se elimina la clase que más dificultades presenta, lo que conlleva a otra mejora sustancial en el rendimiento.

- **Experimento 5 (DS5):** La aplicación de una herramienta de recorte para generar más cantidad de imagenes incrementa drásticamente el número de muestras a 6.416, genera un salto cualitativo

en todas las métricas. Esta metodología es util y significativa debido a que se cumple la función de estar identificando el estado del poro y a la vez se soluciona el desbalanceo de clases.

Tabla 4.4: Comparación de rendimiento e hiperparámetros de los experimentos del modelo (ej. "exp1" corresponde a "YOLOv11x_exp1").

Modelos	**exp1**	**exp2**	**exp3**	**exp4**	**exp5**
Métricas de Rendimiento					
Precisión (P)	0,465	0,581	0,549	0,624	0,737
Recall (R)	0,502	0,585	0,564	0,676	0,862
mAP@.50	0,440	0,564	0,519	0,657	0,852
mAP@.50:.95	0,208	0,286	0,274	0,424	0,710
Hiperparámetros de Entrenamiento					
Imgsz	640	640	640	640	128
Batch Size	16	16	16	16	64
Epochs	100	100	100	100	100
Learning Rate	0,001	0,001	0,001	0,001	0,001
Optimizador	Adam	Adam	Adam	Adam	Adam
Datos y Modelo					
Parámetros (M)	62,05	62,05	62,05	62,05	62,05
Dataset	DS1	DS2	DS3	DS4	DS5
Clases	C1	C2	C2	C3	C3
N° de Imágenes	597	597	449	449	6416
Tiempo [Horas]	0,754	0,641	0,516	0,514	0,594

Nota: Se utilizan abreviaturas por motivos de espacio.
Datasets: DS1: Dataset-Final, **DS2:** Dataset-Final-Merged, **DS3:** Dataset-Final-Filtered, **DS4:** Dataset-Final-Stomata-State, **DS5:** Dataset-Single-Stomata.
Clases: C1: Pore-closed/pore-open/stomata, **C2:** stomata/pore, **C3:** Stomata-closed/stomata-open.

4.2.1. Análisis Exploratorio del Conjunto de Datos Base para Segmentación

Se realiza un análisis exploratorio del conjunto de datos base para segmentación con el fin de caracterizar su estructura y detectar posibles desafíos para el entrenamiento del modelo.

Las estadísticas descriptivas generales, resumidas en la Tabla 4.5, indican que el conjunto, tras el proceso de curación que elimina muestras incorrectamente etiquetadas de las 618 originales, consta de 597 imágenes. Este dataset presenta un promedio de 25,99 objetos (anotaciones) por imagen, y como se visualiza en la Figura 4.4, exhibe una variabilidad considerable.

Tabla 4.5: Estadísticas descriptivas del conjunto de datos base para segmentación. Se detallan las métricas por imagen y el conteo total por clase.

Categoría	Métrica	Valor
Estadísticas por Imagen	Total de Imágenes	597
	Promedio de objetos	25,99
	Desviación Estándar	15,68
	Mínimo de objetos	0
	Percentil 25 (P25)	16
	Mediana (P50)	22
	Percentil 75 (P75)	30
	Máximo de objetos	96
Conteo Total por Clase	`stomata`	7.757
	`pore-open`	4.548
	`pore-closed`	3.209

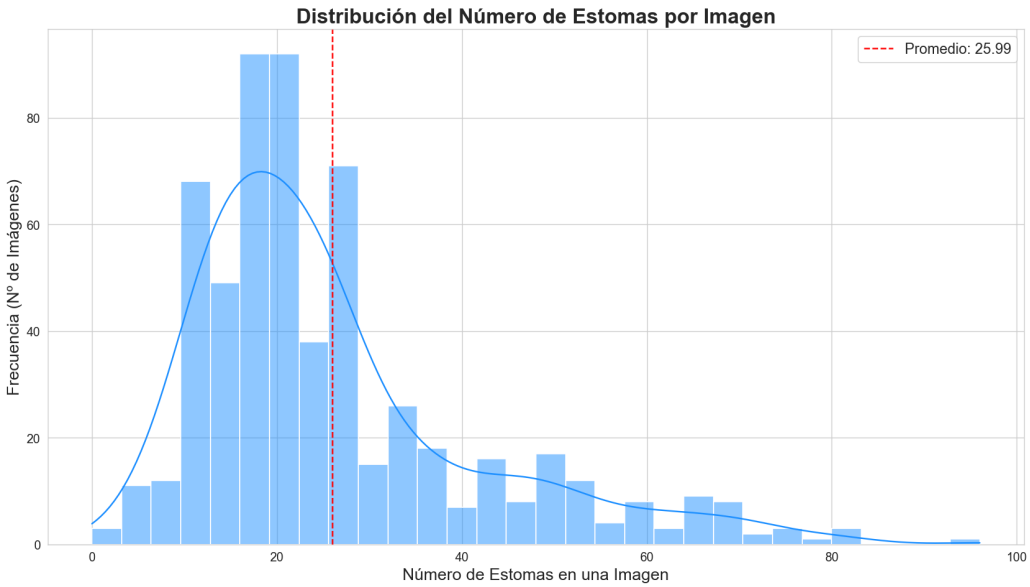


Figura 4.4: Distribución del número de anotaciones totales por imagen en el conjunto de datos base.

Un hallazgo significativo de este análisis es la presencia de **desequilibrio de clases**, como se cuantifica en la Tabla 4.5 y se ilustra en la Figura 4.5. Este desequilibrio plantea un desafío previsible para el entrenamiento: es de esperar que el modelo alcance un alto rendimiento en la segmentación de la clase `stomata`, pero que su capacidad para generalizar en las clases minoritarias sea limitada.

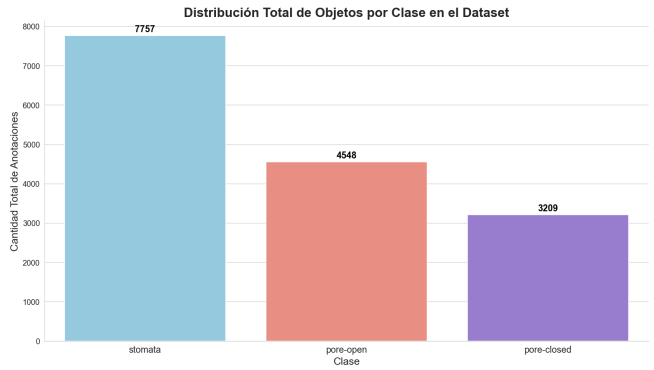


Figura 4.5: Distribución de instancias por clase en el dataset base, evidenciando el desequilibrio.

4.2.1.1. Análisis Exploratorio del Conjunto de Datos de Estomas Individuales

El conjunto de datos de estomas individuales demuestra ser el que produjo el mayor rendimiento. A continuación se presenta un análisis de dicho conjunto para comprender en profundidad las propiedades que pudieron contribuir a su eficacia.

Uno de los hallazgos principales se observa en la distribución de clases. Aunque persiste un desequilibrio entre `stomata-open` y `stomata-closed`, como se muestra en la Figura 4.6, la proporción es más equilibrada que la distribución de tres clases del conjunto base. Se hipotetiza que esta mejora en el balance, junto con la mayor calidad de los datos (al excluir las muestras de Phytolearning), compensa la reducción en el número de imágenes originales y facilita un aprendizaje más efectivo para el modelo.

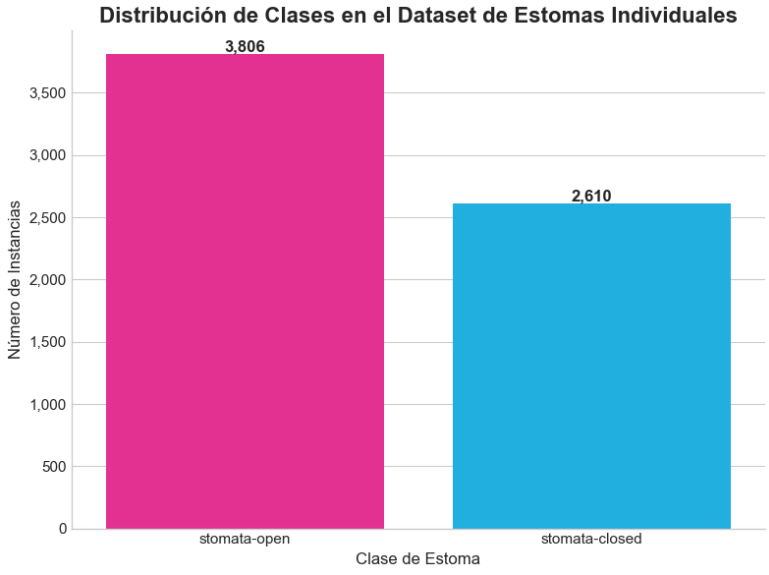


Figura 4.6: Distribución de clases en el conjunto de datos de estomas individuales.

Adicionalmente, este conjunto de datos resalta la ventaja de utilizar máscaras de segmentación sobre las cajas delimitadoras. Como se ilustra en la Figura 4.7, las máscaras se ajustan con alta fidelidad al

contorno real del estoma y su poro, proporcionando al modelo una información espacial mucho más precisa para el entrenamiento.

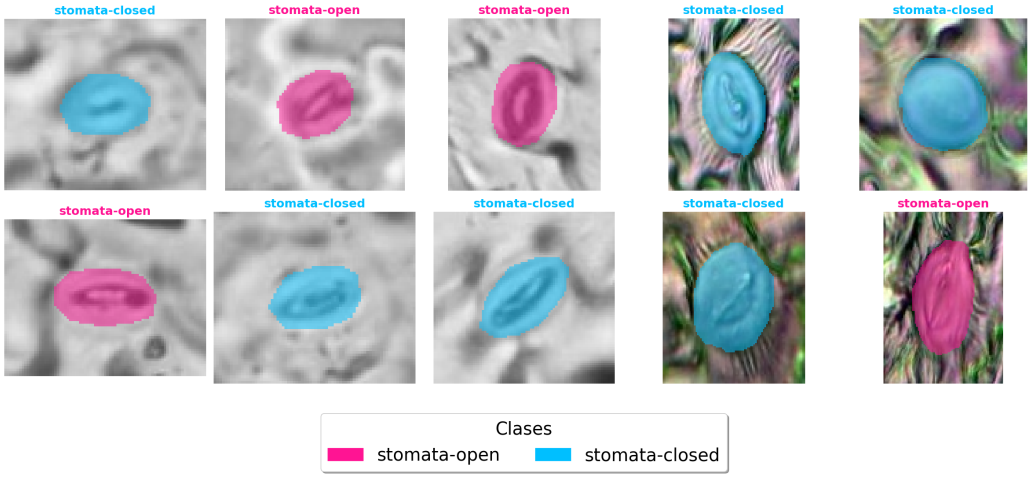


Figura 4.7: Ejemplos de anotaciones en el dataset de estomas individuales. Se aprecia la precisión de las máscaras de segmentación para las clases `stomata-open` y `stomata-closed`.

Finalmente, la alta variabilidad en la resolución de las imágenes, producto de la técnica de recorte, es una característica inherente y valiosa de este dataset (Figura 4.8). El gráfico muestra claramente dos *clusters* que corresponden a las fuentes de datos originales, lo que refleja las diferencias en el tamaño de los estomas y confirma un amplio rango de resoluciones, desde aproximadamente 30×30 hasta 130×130 píxeles. Esta diversidad de escalas, aunque compleja, funciona como una regularización efectiva, entrenando al modelo para que sea más robusto ante imágenes de diferentes tamaños.

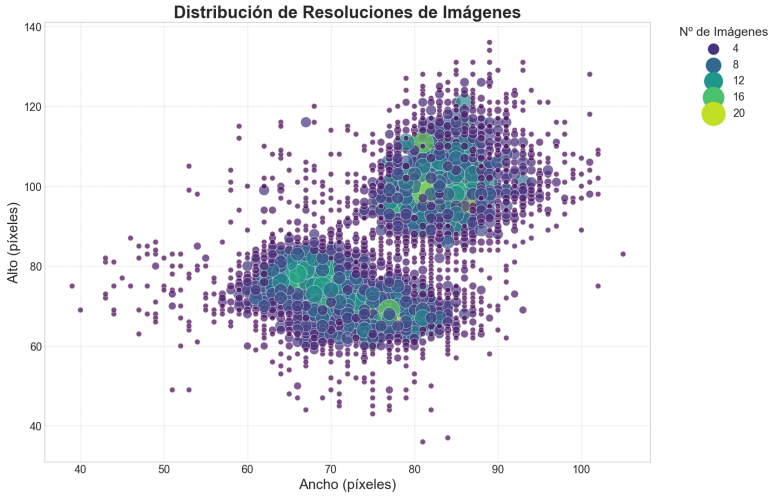


Figura 4.8: Distribución de las resoluciones de imagen en el dataset de estomas individuales. La dispersión de puntos indica una alta variabilidad en el tamaño de las imágenes.

El objetivo de esta serie de experimentos es identificar la configuración de modelo e hiperparámetros que ofrezca el mejor rendimiento para la tarea de segmentación de estomas. La Tabla 4.3 resume

las métricas de validación obtenidas para cada uno de los experimentos realizados ante la línea experimental donde se varían los hiperparámetros y la línea donde se varían las arquitecturas con mecanismos de atención.

4.2.2. Optimización de Hiperparámetros y Selección del Modelo Final

Habiendo establecido el "Dataset-Single-Stomata" como el más efectivo, la fase final de la experimentación se centra en una optimización sistemática de los hiperparámetros y la arquitectura del modelo para maximizar el rendimiento en la tarea de segmentación. Para ello, se evaluan diversas configuraciones, incluyendo variaciones en las arquitecturas YOLOv11.

La Tabla 4.6 presenta un resumen comparativo de las métricas de validación obtenidas para los experimentos más relevantes de esta fase.

Tabla 4.6: Métricas de validación para los experimentos del modelo de segmentación con enfoque en hiperparámetros.

Modelos	**11_recorted**	**11_ULT4**	**11_ULT5**	**12_exp1**	**12_exp5**	**12_exp6**	**12_exp7**	**12_exp8**
Métricas en la Validación								
Precisión (P)	0,737	0,754	0,747	0,733	0,664	0,723	0,753	0,726
Recall (R)	0,862	0,881	0,880	0,887	0,898	0,903	0,887	0,870
mAP@.50	0,852	0,869	0,877	0,875	0,826	0,872	0,886	0,853
mAP@.50:.95	0,710	0,733	0,764	0,757	0,666	0,781	0,805	0,733
Recursos								
Tiempo [Horas]	0,59	0,74	6,22	0,68	0,22	8,78	10,9	7,4
Recurso Comp.	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000

4.2.2.1. Análisis de Resultados de Optimización

El análisis comparativo de los resultados permite extraer varias conclusiones clave sobre el impacto de los hiperparámetros:

- **Impacto del Tamaño de Imagen:** Se observa una correlación positiva entre el tamaño de la imagen de entrada (`imgsz`) y el rendimiento, especialmente en la métrica `mAP@[.50:.95]`. Los experimentos con un `imgsz` de 640 (ej. `11_ULT5`) superan a aquellos con 128 (ej. `11_recorted`), lo que sugiere que la mayor resolución proporciona detalles cruciales para una segmentación de alta precisión.

- **Compromiso Rendimiento-Costo:** El análisis también revela un claro compromiso (*trade-off*) entre el rendimiento y el costo computacional. Por ejemplo, el experimento `12_exp7`, que alcanza el mAP más alto, también requiere el mayor tiempo de entrenamiento (10,9 horas).

- **Efecto del Aumento de Datos:** La inclusión de técnicas de aumento de datos (*data augmentation*) demuestra ser fundamental para mejorar la capacidad de generalización del modelo, como se evidencia al comparar las variantes que lo utilizan.

4.2.2.2. Selección del Modelo de Segmentación Final

Tras la evaluación, el modelo del experimento **12_exp7** es seleccionado como la configuración final, ya que alcanza el rendimiento más alto en la métrica más exigente y representativa de la calidad de la segmentación: **`mAP@[.50:.95]` con un valor de 0,805**. Este resultado indica una buena capacidad para delinear los contornos de los estomas con alta fidelidad.

De esta forma notamos que le cuesta aprender en determinadas muestras pero se logra ver una tendencia a la mejora en los graficos de las curvas de pérdida y métricas en la validación. Sin embargo se identifica irregularidades producto de la dificultad con los datos, en términos de resolución de la muestra con el etiquetado.

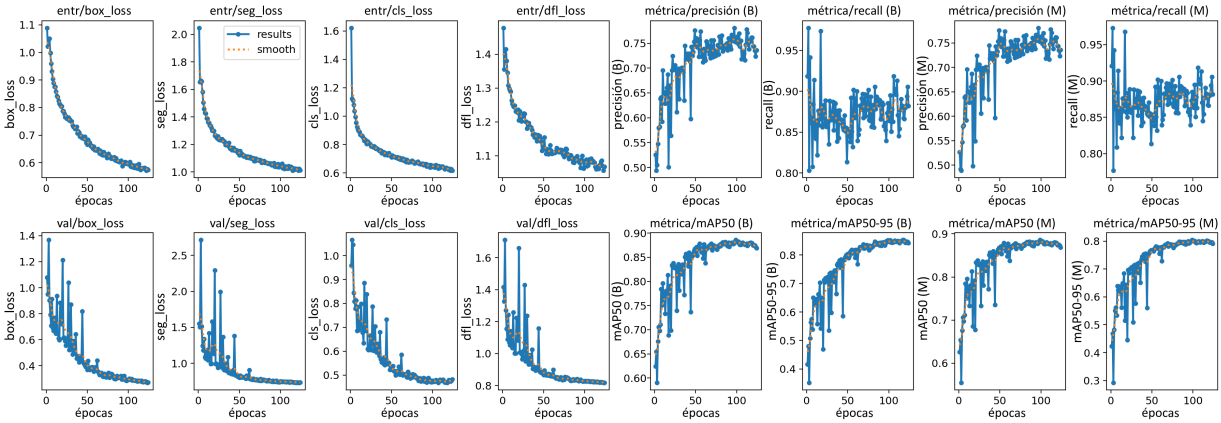
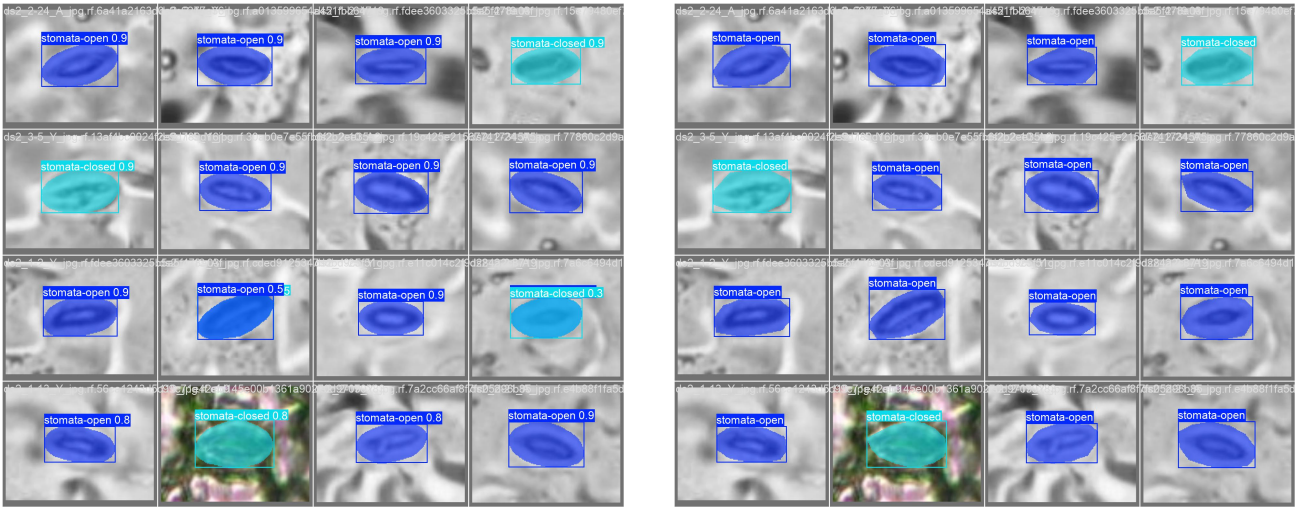


Figura 4.9: Curvas de pérdida en el conjunto validación y entrenamiento para el modelo de segmentación.

Figura 4.10 presenta ejemplos cualitativos de las predicciones de este modelo, donde se puede apreciar visualmente la precisión de las máscaras de segmentación generadas para ambas clases (`stomata-open` y `stomata-closed`).



(a) Muestra de predicciones del modelo con respecto al batch 1

(b) Muestra de etiquetas reales con respecto al batch 1

Figura 4.10: Ejemplos de predicciones del modelo de segmentación final (12_exp7) en el conjunto de validación. Se muestra la máscara de segmentación y la clase predicha para cada estoma con su grado de confianza.

4.2.3. Análisis de Arquitecturas con Mecanismos de Atención

Esta sección final evalúa el impacto de las modificaciones de arquitectura, específicamente, la integración de mecanismos de atención, en el rendimiento del modelo de segmentación. El objetivo es determinar si estas arquitecturas más complejas pueden superar el rendimiento del mejor modelo obtenido mediante la optimización de hiperparámetros.

4.2.3.1. Análisis Cuantitativo de Resultados

La Tabla 4.7 resume las métricas de validación obtenidas para los experimentos que utilizaron módulos de atención.

Tabla 4.7: Métricas de validación para los experimentos con mecanismos de atención.

Modelos	**Exp_1**	**Exp_2**	**Exp_3**	**Exp_4**	**Exp_5**	**Exp_6**
Métricas en la Validación						
Precisión (P)	0,743	—	0,731	0,754	0,717	0,741
Recall (R)	0,870	—	0,845	0,831	0,889	0,906
mAP@.50	0,881	—	0,848	0.859	0.865	0,889
mAP@.50:.95	0,802	—	0,707	0,780	0,734	0,806
Recursos						
Tiempo [Horas]	6,8	-	1,9	23,4	15,2	37,2
Recurso Comp.	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000	NVIDIA RTX A5000
Notas	Loss NaN desde ep. 56	Loss NaN y métricas 0	—	—	—	—

El análisis comparativo de los resultados es concluyente. El modelo del experimento **Exp_6**, que integra el módulo de atención C3CA, no solo es el de mejor rendimiento dentro de este grupo, sino que también supera al mejor de la fase anterior (el modelo `12_exp7`). Alcanza un **`mAP@[.50:.95]` de 0,806**, estableciendo un nuevo estado del arte para esta investigación. Este logro, aunque conlleva un costo computacional significativamente mayor (37,2 horas de entrenamiento), valida la hipótesis de que la integración de mecanismos de atención es una estrategia efectiva para mejorar la precisión de la segmentación. En el caso de usarse C3CA, ya que al utilizar CBAM las curvas de perdida crecen exponencialmente con valores NaN como se aprecia en la Tabla 4.7.

4.2.3.2. Análisis de la Dinámica de Entrenamiento

Las curvas de entrenamiento para el modelo `Exp_6` (Figura 4.11) refuerzan estos hallazgos. Se observa una convergencia estable y con menor fluctuación en comparación con los modelos sin atención, especialmente en las curvas de validación. Esto sugiere que el mecanismo de atención no solo mejora las métricas finales, sino que también contribuye a un proceso de aprendizaje más robusto y generalizable.

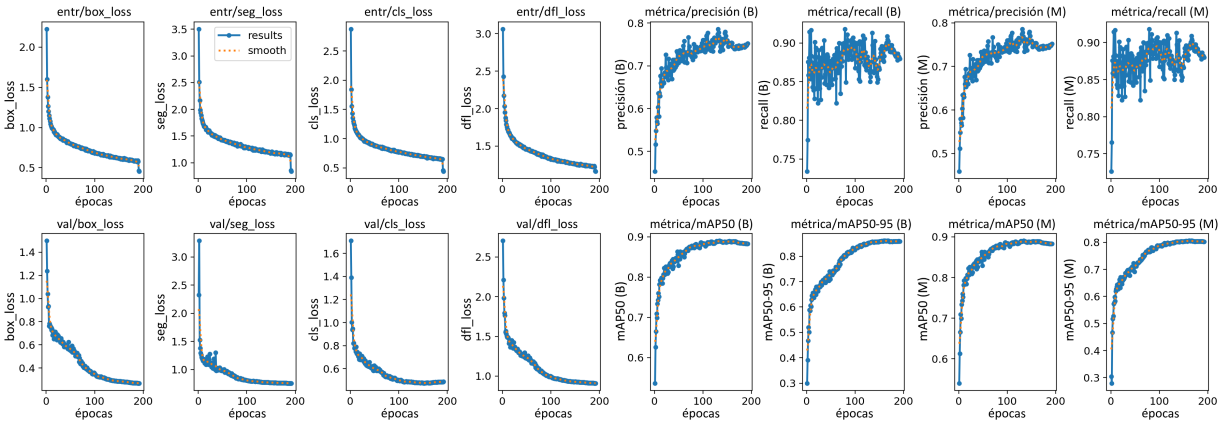
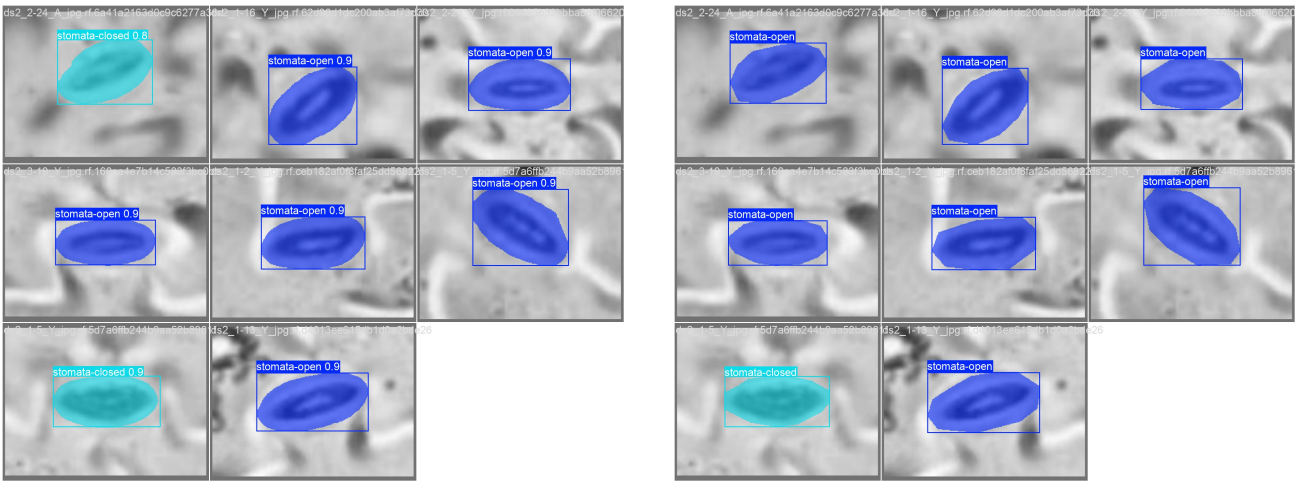


Figura 4.11: Curvas de pérdida y métricas durante el entrenamiento para el modelo con mecanismo de atención (Exp_6).

4.2.3.3. Análisis Cualitativo de Predicciones

Para complementar el análisis cuantitativo, la Figura 4.12 presenta ejemplos visuales de las predicciones del modelo `Exp_6`. Se observa una alta fidelidad entre las máscaras de segmentación predichas (izquierda) y las etiquetas reales (derecha), incluso en estomas con formas complejas. Los altos puntajes de confianza asociados a cada predicción son un testimonio adicional de la robustez del modelo final.



(a) Predicciones del modelo. **(b) Etiquetas reales (Ground Truth).**

Figura 4.12: Comparación visual entre las predicciones del modelo final (Exp_6) y las etiquetas reales para un batch de validación. Se destaca la precisión de las máscaras y la correcta clasificación del estado estomático (excepto en un caso).

Finalmente, al analizar la métrica de precisión, se observa que los modelos de identificación obtienen resultados superiores a los de segmentación (Figura 4.13). Esta diferencia se atribuye a que la tarea de detección mediante bounding boxes presenta una menor complejidad para el modelo en comparación con la segmentación, que requiere una delimitación precisa a nivel de píxel.

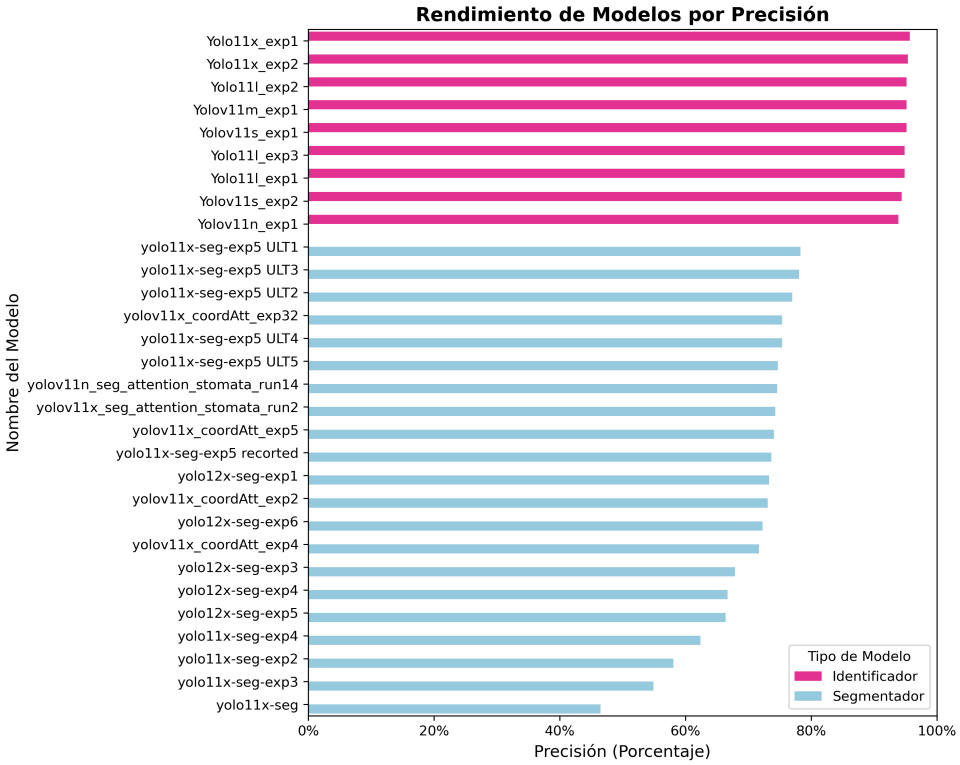


Figura 4.13: Variación de rendimiento para la métrica precisión en los experimentos realizados.

4.3. Comparación con la Literatura

Se realiza una comparación de los resultados obtenidos con los modelos desarrollados en este trabajo frente a los hallazgos publicados en la literatura científica. Esta comparativa busca contextualizar el rendimiento de los modelos.

4.3.1. Modelo de Detección de Estomas

El primer modelo desarrollado en este trabajo se enfoca en la tarea de detección de estomas utilizando cajas delimitadoras.

Características del Conjunto de Datos: El modelo es entrenado con un conjunto de **1.141 imágenes microscópicas de *Arabidopsis thaliana***. Estas imágenes, caracterizadas por su **alta resolución (1248 × 928 píxeles)** y formato RGB, fueron capturadas utilizando un microscopio confocal. El modelo está específicamente configurado para identificar los estomas bajo la clase "stomata".

Métricas de Rendimiento (Conjunto de Validación): Las métricas obtenidas en el conjunto de validación son: Precisión (0,954), Recall (0,956), mAP50 (0,984) y mAP5095 (0,676).

Tabla 4.8: Comparativa de Modelos para la Detección de Estomas

Modelo/Estudio	Arquitectura	Especie del Dataset	Características del Dataset (Tipo/Resolución/Conteo)	Precisión	Recall	mAP50 (AP@0.5)	mAP5095 (AP@0.5:0.95)	Otras Métricas (F1-score, Accuracy)
Nuestro Modelo	**YOLOv11x**	**Arabidopsis Thaliana**	**Imágenes microscópicas**	**0,954**	**0,956**	**0,984**	**0,676**	**—**
Chaplin et al. (2025)	YOLOv8-M	Trigo	Imágenes de campo	—	—	0,971	—	—
X. Li et al. (2022)	YOLOv5 mejorado + CA	Haba/Trigo	Imágenes microscópicas	0,934	—	0,968 (Haba)	—	Accuracy: 0,934 (Haba)
Sultana et al. (2021)	YOLO mejorado	Soja	Imágenes microscópicas	—	—	~0,99	—	—
Fetter et al. (2019)	AlexNet (DCNN)	Taxonómicamente diversa	Imágenes microscópicas	—	—	—	—	Accuracy: 0,942

Análisis de los Resultados de Detección: Los resultados obtenidos para el modelo de detección de estomas son altos y se posicionan de manera muy competitiva dentro del panorama actual de la literatura científica. En particular, el **mAP50 de 0,984** es un indicador sobresaliente, sugiriendo que el modelo logra una buena precisión en la detección de estomas a un umbral de intersección sobre unión (IoU) del 50 %. Este valor es comparable, e incluso superior, a muchos de los resultados reportados, como el 0,971 de Chaplin et al. (2025) con YOLOv8-M.

La **precisión (0,954)** y el **recall (0,956)** también son métricas muy sólidas. Un alto valor de precisión indica que la mayoría de las cajas delimitadoras predichas por el modelo corresponden realmente a estomas, minimizando los falsos positivos. Por otro lado, el alto recall asegura que el modelo es capaz de identificar la vasta mayoría de los estomas presentes en las imágenes, reduciendo los falsos negativos. Estos valores son directamente comparables con estudios que reportan métricas explícitas, como X. Li et al. (2022) y se encuentran en el rango superior de lo que se considera un rendimiento robusto en este tipo de tareas.

Es importante destacar la métrica **mAP50-95 (0,676)**. A diferencia de mAP50, mAP50-95 promedia el rendimiento del modelo a través de un rango de umbrales IoU más estrictos (del 50 % al 95 %). La falta de mAP50-95 en la mayoría de los trabajos comparados limita la comparación directa de esta métrica. Sin embargo, el valor obtenido de 0,676 es un buen indicador de que el modelo mantiene un rendimiento aceptable incluso cuando se exige una superposición muy precisa entre la predicción y la verdad fundamental. En síntesis, el modelo propuesto demuestra una capacidad robusta y precisa para identificar estomas en imágenes microscópicas, lo que lo convierte en una herramienta altamente fiable para aplicaciones de fenotipado.

4.3.2. Modelo de Segmentación de Instancia de Estomas (Estado de Apertura)

El segundo modelo desarrollado en este trabajo aborda una tarea significativamente más compleja: la segmentación de instancias de estomas con clasificación directa de su estado de apertura (abierto o cerrado).

Arquitectura: Este modelo emplea la arquitectura **YOLOv11**, que ha sido **mejorada con un mecanismo de atención *Coordinate Attention* (C3CA)**. Esta combinación potencia la capacidad del modelo para enfocar recursos computacionales en las características más relevantes de los estomas, mejorando tanto la localización como la discriminación del estado.

Características del Conjunto de Datos: El modelo es entrenado con un conjunto de **6.416 imágenes**, dividido en un 80 % para entrenamiento y un 20 % para validación. Una característica distintiva de este dataset es que **cada imagen contiene un solo estoma**, lo que facilita el aprendizaje granular de las características asociadas a los estados de apertura. Las imágenes presentan resoluciones variadas, aproximadamente 100x100 píxeles. El conjunto de datos incluye imágenes de dos especies de plantas: ***Arabidopsis thaliana* y *Cicer arietinum* (garbanzo)**, lo que enriquece la diversidad del entrenamiento y mejora el potencial de generalización del modelo entre especies dicotiledóneas y monocotiledóneas.

La tarea principal del modelo es la **segmentación de instancias**, lo que implica no solo identificar, sino también delimitar el contorno preciso de cada estoma mediante una máscara poligonal. Crucialmente, el modelo debe clasificar explícitamente los estomas identificados en sus estados de *stomata-open* y *stomata-closed*, proporcionando una cuantificación directa de su estado fisiológico.

Las métricas obtenidas en el conjunto de validación son las siguientes: Precisión (0,741), Recall (0,906), mAP50 (0,889) y mAP5095 (0,806).

Tabla 4.9: Comparación de Modelos de Segmentación de Instancias de Estomas (Estado de Apertura)

Modelo/Estudio	Arquitectura	Especie del Dataset	Características del Dataset (Tipo/Res/Estomas por img/Conteo)	Tarea/Salida	Precisión	Recall	mAP50	mAP5095	Otras Métricas (Accuracy, IoU)
Nuestro Modelo	**YOLOv11 + C3CA**	**Arabidopsis Thaliana & Cicer aeretenium**	**Microscópicas, ~100x100 px, Estoma único por imagen, 6416 imágenes**	**Segmentación (estado abierto/cerrado)**	**0,741**	**0,906**	**0,889**	**0,806**	—
Cong et al. (2023)	YOLO-X Modificado	Populus	Hojas vivas	Bounding box (estado abierto/cerrado)	—	—	0,969	—	Accuracy: 0,971
Petrie et al. (2021)	Mask R-CNN	Varias	Microscópicas, calidad/escala diversa, >60k estomas	Segmentación (general)	0,951	0,8334	—	—	F-score: 0,8861, IoU: 0,70
Takagi et al. (2023)	YOLOX + U-Net	Arabidopsis thaliana	Microscopía de campo claro	Detección + Segmentación de poros	—	—	0,875 (detección)	N/A	IoU: 0,745 (segmentación)

Análisis de los Resultados de Segmentación: La implementación de un modelo **YOLOv11 con CA** para la segmentación de instancia y la clasificación de estados abierto/cerrado en estomas. La investigación en la literatura actual revela que es difícil encontrar una comparación directa que combine exactamente estas características:

- **YOLOv11 con *CA***: La utilización de YOLOv11 junto con el mecanismo de atención CA, es particularmente innovadora en el contexto del análisis de estomas. Los estudios existentes, como se observa en la Tabla 4.9, emplean predominantemente versiones anteriores de YOLO (como YOLOv8 o YOLO-X) o arquitecturas como Mask R-CNN.
- **Clasificación de Estado Abierto/Cerrado con Segmentación de Instancia Pura**: Si bien algunos trabajos (por ejemplo,Cong et al. (2023)) abordan la clasificación de estados abierto/cerrado utilizando enfoques basados en YOLO, la mayoría no reportan métricas detalladas de precisión, recall y mAP por clase para la segmentación de instancia con polígonos.

Las métricas obtenidas por el modelo son prometedoras para una tarea de esta complejidad: El **mAP50 de 0,889** y el **mAP50-95 de 0,806** son valores sólidos, especialmente el mAP50-95, que demuestra la capacidad del modelo para generar segmentaciones de alta calidad y precisión incluso bajo umbrales de IoU estrictos. Esto es un indicador directo de la fiabilidad del modelo en la delimitación precisa de los estomas en sus diferentes estados. El recall de 0,906 es particularmente alto, lo que significa que el modelo es bueno identificando la mayoría de los estomas presentes en las imágenes, tanto abiertos como cerrados. La precisión de 0,741, aunque es la métrica con el valor más bajo en comparación con el recall, sigue siendo aceptable para la complejidad de la tarea de segmentación de instancia y clasificación de estados. Sugiere que, si bien el modelo es bueno para encontrar todos los estomas relevantes (alto recall), puede haber casos donde las predicciones de la clase (abierto/cerrado) no sean tan precisas, o donde las máscaras poligonales contengan pequeños errores que afectan esta métrica.

En conclusión, la implementación del modelo representa un avance en la automatización del análisis de estomas. Al integrar una segmentación de instancia precisa del estado de apertura utilizando una arquitectura con mecanismo de atención, el modelo aborda un área donde la literatura aún está en fase de exploración.

Capítulo 5

Conclusiones y Trabajo Futuro

Este capítulo final sintetiza las contribuciones fundamentales de la presente investigación, evalúa el cumplimiento de los objetivos propuestos y delinea las futuras líneas de trabajo que se derivan de los resultados obtenidos.

5.1. Conclusiones

La presente investigación culmina con el desarrollo y la validación de un pipeline computacional basado en DL, capaz de automatizar la detección y segmentación de estomas con una clasificación simultánea de su estado funcional (abierto o cerrado).

Se demuestra que el modelo de detección, basado en la arquitectura YOLOv11, alcanza una eficacia sobresaliente para la identificación y localización general de estomas, con un rendimiento robusto (mAP@.50 de 0,984). Adicionalmente, el modelo de segmentación que integra un mecanismo de atención C3CA, representa un avance significativo. Este modelo no solo delinea con alta precisión el contorno estomático, sino que también clasifica su estado, superando las métricas de configuraciones más simples y posicionándose como una solución novedosa y eficaz en comparación con la literatura existente.

La principal contribución de este trabajo radica en la automatización de un proceso tradicionalmente manual y laborioso. Al proporcionar una herramienta rápida y precisa para la fenotipificación de alto rendimiento, esta investigación ofrece un valioso recurso para la comunidad científica y agrícola. La capacidad de cuantificar rápidamente los parámetros estomáticos es fundamental para la toma de decisiones informadas en el estudio de la fisiología vegetal y, en particular, en la evaluación de la respuesta de las plantas a condiciones de estrés ambiental, como la sequía.

Es importante reconocer que los modelos entrenados y validados en un conjunto de datos específico. Aunque se diseña para ser diverso, la generalización a especies vegetales o condiciones de imagen radicalmente diferentes podría requerir un reentrenamiento o ajuste fino.

5.1.1. Presentación de Estudio en Conferencia

El paper **"Low_Cost Miscroscopic Imaging and YOLO-Based Stomatal Annalysis for Early Drought Detection in Plants"** está en proceso de revisión por la Sociedad Chilena de Ciencia de la Computación (SCCC) para ser presentado de acuerdo a la resolución el cinco de octubre en la "Conferencia Internacional de Jornadas Chilenas de la Computación".

En este paper se implementa y valida la efectividad de un pipeline completo automatizado (vease la Figura 5.1) que utiliza los dos modelos propuestos por la tesis, así tambien los resultados obtenidos y la metodología desarrollada abren múltiples y prometedoras líneas de investigación.

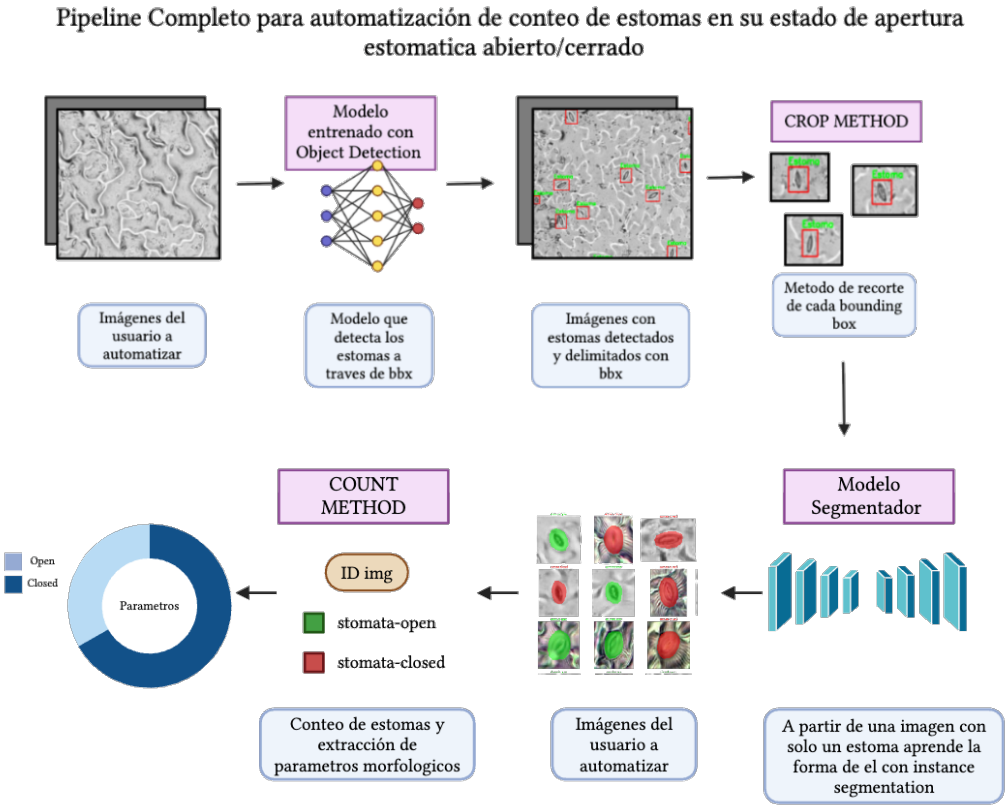


Figura 5.1: Diagrama del pipeline propuesto para la automatización de la extracción de parámetros morfológicos, desde la carga de imágenes hasta la generación de resultados.

El pipeline es validado en un conjunto de 50 imágenes de *Arabidopsis thaliana*, obteniendo las métricas que se visualizan en la Tabla 5.1, siendo así su precisión para identificar estomas abiertos de 95,37 %.

Tabla 5.1: Estadísticas de error e intervalos de confianza del 95 % para cada clase de modelo.

Estadística	Error Estomas Abiertos	Error Estomas Cerrados	Error de Segmentación
Estadísticas Descriptivas			
Media	4,63 %	16,72 %	17,34 %
Desviación Estándar	0,0822	0,1233	0,0936
Intervalo de Confianza (95 %)	±2,28 %	±3,42 %	±2,59 %
Límites del Intervalo de Confianza			
Límite Superior	6,91 %	20,14 %	19,93 %
Media	4,63 %	16,72 %	17,34 %
Límite Inferior	2,35 %	13,30 %	14,74 %

5.1.2. Trabajo Futuro

Los resultados obtenidos y la metodología desarrollada abren múltiples y prometedoras líneas de investigación y desarrollo futuro. Se proponen las siguientes:

- **Proponer una interfaz para el FrontEnd del pipeline:** El siguiente paso lógico es encapsular el pipeline completo en una herramienta de software con una interfaz gráfica de usuario (GUI). Esta aplicación permitiría a los investigadores sin experiencia en programación cargar sus propias imágenes y obtener resultados de manera intuitiva. La interfaz podría ofrecer opciones para seleccionar modelos pre-entrenados optimizados para distintas calidades de imagen (ej. “baja resolución” vs. “alta resolución”), culminando en la entrega de un reporte con estadísticas y parámetros morfológicos.

- **Mejora de la Generalización y Robustez del Modelo:** Para aumentar la aplicabilidad de los modelos, se puede expandir el conjunto de datos de entrenamiento para incluir una mayor diversidad de genotipos, e incluso diferentes especies de plantas. Esto mejoraría la capacidad de generalización del modelo. Una extensión de la GUI podría permitir al usuario seleccionar un modelo específicamente afinado para la especie que está analizando.

- **Expansión de las Capacidades Analíticas:** El pipeline actual se centra en la segmentación y clasificación del estado. Futuras versiones podrían enriquecerse para extraer automáticamente un rango más amplio de parámetros morfológicos.

Bibliografía

Alirezazadeh, P., Michael, S., & Stolzenburg, F. (2022). Improving Deep Learning-based Plant Disease Classification with Attention MechanismOptimierung der auf Deep Learning basierenden Klassifizierung von Pflanzenkrankheiten mit CBAM. *Gesunde Pflanzen*, *75*. https://doi.org/10.1007/s10343-022-00796-y

Almuiña, A. (2024). 12 Redes Neuronales | Curso Machine Learning con R [[Consultado el 8 de julio de 2025]]. https://albertotb.com/curso-ml-R/Rmd/12-nn/12-nn.html

Ballard, D. H., & Brown, C. M. (1982). *Computer Vision*. Prentice-Hall.

Banco Mundial. (2025). Agua:Panorama General [Consultado en julio de 2025]. https://www.bancomundial.org/es/topic/water/overview

Barbedo, J. G. A. (2017). A new automatic method for disease symptom segmentation in digital photographs of plant leaves. *European journal of plant pathology*, *147*(2), 349-364.

Bottou, L. (2012). Stochastic Gradient Descent Tricks. *Neural Networks: Tricks of the Trade*, 421-436.

Chaplin, E., Coleman, G., Merchant, A., & Salter, W. (2025). *FieldDino: Rapid In-Field Stomatal Anatomy and Physiology Phenotyping* (inf. téc.). Wiley Online Library.

Clark, D., & Brown, B. (2015). A rapid image acquisition method for focus stacking in microscopy. *Microscopy Today*, *23*(4), 18-25.

Colaboradores de Wikipedia. (2024). Perceptrón — Wikipedia, La enciclopedia libre [Consultado el 8 de julio de 2025]. https://commons.wikimedia.org/wiki/File:Perceptr%C3%B3n_5_unidades.svg

Commission, J. R. C. (.-. E. (2022). Drought in Europe – August 2022 [Accessed: 2024-05-15].

Cong, Z., Ma, Q., Yu, H., Li, J., & Zhang, H. (2023). Microscopy image recognition method of stomatal open and closed states in living leaves based on improved YOLO-X. *Plant Methods*, *19*(1), 111. https://doi.org/10.1007/s40626-023-00296-y

Cuartas, L. M. (2020). Convolución como una Operación Fundamental en Redes Neuronales Convolucionales [Personal communication and pedagogical material, based on common CNN principles.].

DataScientest. (2024). U-Net Architecture: A Detailed Guide [Accessed: 2024-05-15]. https://datascientest.com/es/u-net-lo-que-tienes-que-saber.

Deltares. (2018). Drought in cities: a global problem [Accessed: 2024-05-15]. https://www.deltares.nl/en/stories/drought-in-cities-a-global-problem

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 248-255.

Deshmukh, B. S., & Mankar, V. H. (2014). Segmentation of microscopic images: A survey. *2014 International Conference on Electronic Systems, Signal Processing and Computing Technologies*, 362-364.

Dozat, T. (2016). Incorporating nesterov momentum into adam. https://openreview.net/forum?id=OM0jvwB8jIp57ZJjtNEZ

Farooq, M., Hussain, M., Wahid, A., & Siddique, K. (2012). Drought stress in plants: an overview. *Plant responses to drought stress: From morphological to molecular features*, 1-33.

Fetter, K. C., Eberhardt, S., Barclay, R. S., Wing, S., & Keller, S. R. (2019). StomataCounter: a neural network for automatic stomata identification and counting. *New Phytologist*, *223*(3), 1671-1681.

Gibbs, J. A., & Burgess, A. J. (2024). Application of deep learning for the analysis of stomata: a review of current methods and future directions. *Journal of Experimental Botany*, *75*(21), 6704-6718.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Harter, K., Meixner, A. J., & Schleifenbaum, F. (2012). Spectro-microscopy of living plant cells. *Molecular plant*, *5*(1), 14-26.

Haworth, M., Elliott-Kingston, C., & McElwain, J. C. (2011). Stomatal control as a driver of plant evolution. *Journal of experimental botany*, *62*(8), 2419-2423.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, 2961-2969.

Hetherington, A. M., & Woodward, F. I. (2003). The role of stomata in sensing and driving environmental change. *Nature*, *424*(6951), 901-908.

Hinton, G., Srivastava, N., & Swersky, K. (2012). Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. http://www.cs.toronto.edu/~hinton/coursera/lecture6/lec6.pdf

Hou, Q., Zhou, D., & Feng, J. (2021). Coordinate Attention for Efficient Mobile Network Design. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13713-13722.

Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *Proceedings of the International Conference on Machine Learning*, 448-456.

Jayakody, S., Perera, M., & Jayaratne, D. (2022). DeepStomata: Deep Learning for Stomatal Segmentation, Detection, and Counting. *arXiv preprint arXiv:2209.07119*.

Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Klosinska, M., Picard, C. L., & Gehring, M. (2016). Conserved imprinting associated with unique epigenetic signatures in the Arabidopsis genus. *Nature Plants*, *2*(10), 1-8.

Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, *14*(2), 1137-1143.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature*, *521*(7553), 436-444. https://doi.org/10.1038/nature14539

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324. https://doi.org/10.1109/5.726791

Li, R., Wu, M., Song, Q., Zhao, R., Li, K., Guo, F., & Han, X. (2022). LeafNet: A hierarchical CNN for epidermal cell and stomatal segmentation and quantification in Arabidopsis thaliana. *Plant Methods*, *18*(1), 158. https://pmc.ncbi.nlm.nih.gov/articles/PMC8972303/

Li, X., Du, H., Dong, J., & Li, W. (2022). An automatic plant leaf stoma detection method based on YOLOv5. *Journal of Physics: Conference Series*, *2316*(1), 012015. https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/ipr2.12617

Martin, C., & Glover, B. J. (2007). Functional aspects of cell patterning in aerial epidermis. *Current Opinion in Plant Biology*, *10*(2), 155-162. https://www.sciencedirect.com/science/article/pii/S1369526606001853

McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (1956). A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence. *AI Magazine*, *27*(4), 12-14.

MIRANDA, M. A. V. (2022). *MACHINE LEARNING CLASSIFICATION OF SINGLE CELL RNA-SEQ ACROSS DIFFERENT TYPES OF CANCER* [Tesis doctoral, Departamento de Bioquímica Clínica e Inmunología Facultad de Farmacia . . .]. https://repositorio.udec.cl/server/api/core/bitstreams/d5ac3061-7cea-4876-9332-2ac622768674/content

Moen, E., Bannon, D., Kudo, T., Graf, W., Covert, M., & Van Valen, D. (2019). Deep learning for cellular image analysis. *Nature methods*, *16*(12), 1233-1246.

Möller, B. K., Dragwidge, J., Smith, R. M., Bird, D., Ma, X., Van De Peer, Y., Berleth, T., & Smith, R. S. (2017). PaCeQuant: A Tool for Quantitative Analysis of Pavement Cell Shape in 2D Images. *Frontiers in Plant Science*, *8*, 1050. https://doi.org/10.3389/fpls.2017.01050

Mordor Intelligence. (2024). *Mercado de IA en la agricultura - Crecimiento, Tendencias, Impacto de COVID-19 y Pronósticos (2024 - 2029)*. Mordor Intelligence. Consultado el 14 de julio de 2025, desde https://www.mordorintelligence.com/es/industry-reports/ai-in-agriculture-market

Nilsson, N. J. (1980). *Principles of Artificial Intelligence*. Tioga Publishing Company.

Ogunrinde, A. T., Oluwadare, F. S., Ifedayo, A. A., & Adebayo, A. T. (2025). Spatiotemporal Assessment of Drought Conditions and Their Impact on Crop Yields in Southwestern Nigeria using Remote Sensing and Machine Learning. *Journal of African Agricultural Research*, *forthcoming*.

Organización de las Naciones Unidas. (2023). Los datos sobre sequía muestran üna emergencia sin precedentes a escala planetaria" [Consultado en mayo de 2025]. https://news.un.org/es/story/2023/05/1511977

Padilla, R., Netto, S. L., & Da Silva, E. A. (2020). A survey on performance metrics for object-detection algorithms. *2020 international conference on systems, signals and image processing (IWSSIP)*, 237-242.

Petrie, A. J., Armstrong, J., Jones, G. B., Miller, M., Ledingham, K., Taylor, G., Hetherington, A. M., Murchie, E. H., & Pound, M. P. (2021). A generalised approach for high-throughput instance segmentation of stomata using Mask R-CNN. *Plant Methods*, *17*, 1-13.

Polyak, B. T. (1964). Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, *4*(5), 1-17.

Reddi, S. J., Kale, S., & Kumar, S. (2018). On the Convergence of Adam and Beyond. *Proceedings of the International Conference on Learning Representations (ICLR)*.

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

Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, Faster, Stronger. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7263-7271.

Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *arXiv preprint arXiv:1804.02767*.

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6), 1137-1149. https://doi.org/10.1109/TPAMI.2016.2577031

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, 234-241.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, *65*(6), 386.

Rovira, C., Díaz, F., & González, M. (2024). Importancia funcional de los estomas en el intercambio gaseoso y la regulación climática. *Revista Chilena de Fisiología Vegetal*, *21*(1), 34-49.

Ruder, S. (2016). An Overview of Gradient Descent Optimization Algorithms. *arXiv preprint arXiv:1609.04746*.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*, 533-536.

Russell, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach* (4th). Pearson.

SERP.AI. (2022). Coordinate Attention: A Novel Mechanism for Efficient Mobile Network Design [Accessed: 2024-07-07]. https://serp.ai/posts/coordinate-attention/.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, *15*, 1929-1958.

Stanford University Human-Centered Artificial Intelligence. (2024). *AI Index*. Consultado el 14 de julio de 2025, desde https://hai.stanford.edu/ai-index

Sultana, S., Lu, Y., Zeng, Y., Cao, Y., & Cai, H. (2021). Improving Stomatal Detection in Soybean Leaf Images Using an Enhanced YOLO Deep Learning Model. *Sustainability*, *13*(24), 13717.

Takagi, M., Hirata, R., Aihara, Y., Hayashi, Y., Mizutani-Aihara, M., Ando, E., Yoshimura-Kono, M., Tomiyama, M., Kinoshita, T., Mine, A., et al. (2023). Image-based quantification of Arabidopsis thaliana stomatal aperture from leaf images. *Plant and Cell Physiology*, *64*(11), 1301-1310.

Tian, Z., & Shen, C. (2025). YOLOv12: A Unified Framework for Real-Time Object Detection. *arXiv preprint arXiv:2501.XXXXX*.

Turing, A. M. (1950). Computing Machinery and Intelligence. *Mind*, *59*(236), 433-460.

Ultralytics. (2023, abril). What is Mask R-CNN and How Does it Work? [Consultado el 8 de julio de 2025].

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention Is All You Need. *Advances in Neural Information Processing Systems*, *30*.

Vico, G., Manzoni, S., Palmroth, S., & Katul, G. (2011). Effects of stomatal delays on the economics of leaf gas exchange under intermittent light regimes. *New Phytologist*, *192*(3), 640-652.

Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2023). YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7464-7475.

Werbos, P. J. (1974). Beyond Recognition, New Tools for Prediction and Analysis in the Behavioural Sciences.

Willmer, C., & Fricker, M. (1996). *Stomata*. Springer Netherlands. https://doi.org/10.1007/978-94-011-0579-8

Wolny, A., Cerrone, L., Vijayan, A., Tofanelli, R., Barro, A. V., Louveaux, M., Wenzl, C., Strauss, S., Wilson-Sánchez, D., Lymbouridou, R., et al. (2020). Accurate and versatile 3D segmentation of plant tissues at cellular resolution. *Elife*, *9*, e57613.

Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. (2018). CBAM: Convolutional Block Attention Module. *Proceedings of the European Conference on Computer Vision (ECCV)*, 3-19.

Xu, J., Xiang, D., Zhang, J., Bao, J., Hu, X., & Yao, G. (2010). Cell nucleus segmentation in histopathological images using a convolutional neural network. *Pattern Recognition Letters*, *31*(10), 1246-1250.

Yang, C., Wu, S., He, J., Cai, L., & Li, X. (2024). YOLOv10: Real-Time End-to-End Object Detection. *arXiv preprint arXiv:2404.07180*.

Zeiler, M. D. (2012). ADADELTA: An Adaptive Learning Rate Method. *arXiv preprint arXiv:1212.5701*.

Zhu, C., Hu, Y., Mao, H., Li, S., Li, F., Zhao, C., Luo, L., Liu, W., & Yuan, X. (2021). A deep learning-based method for automatic assessment of stomatal index in wheat microscopic images of leaf epidermis. *Frontiers in Plant Science*, *12*, 716784.

Apéndice

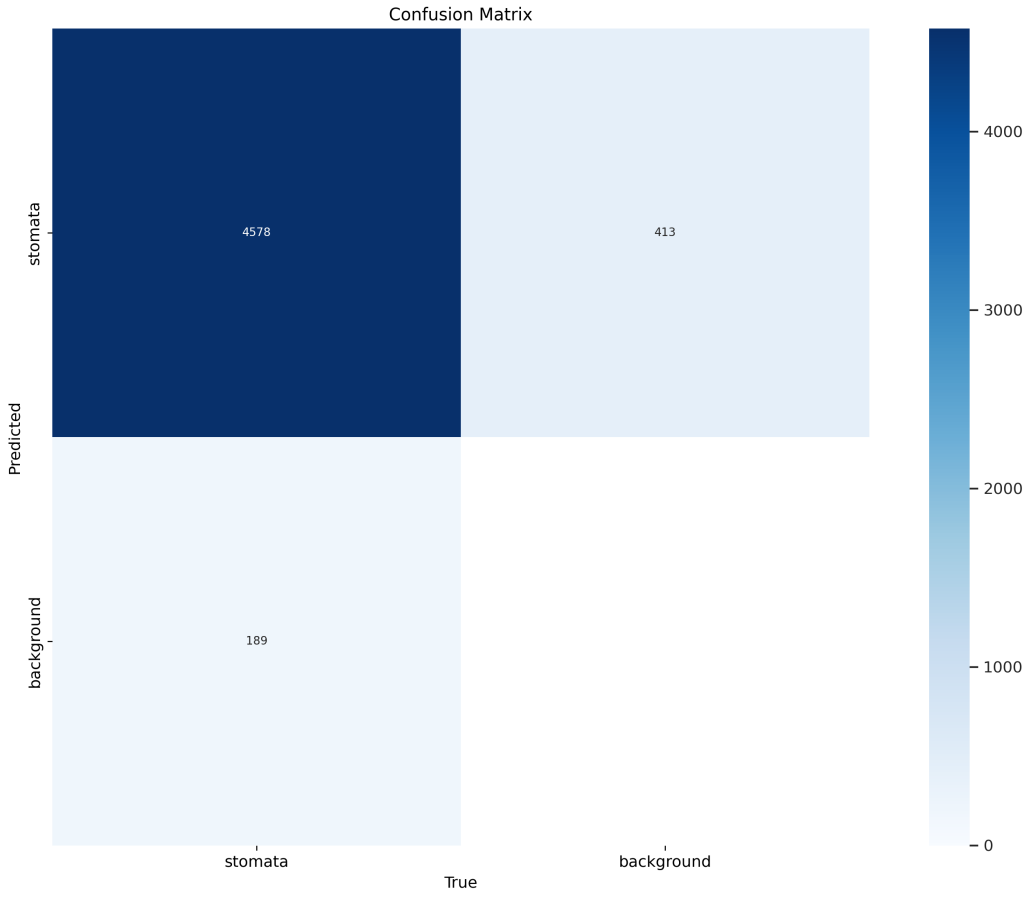


Figura 5.2: Matriz de confusión para el experimento s_exp2.

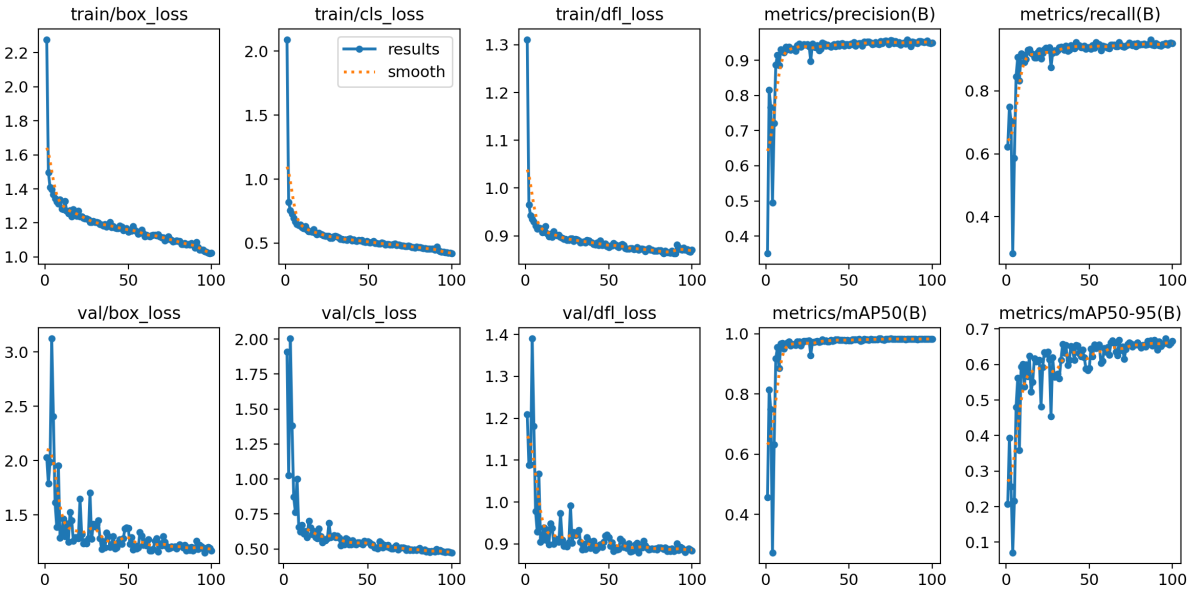


Figura 5.3: Curvas de perdida en el conjunto validación y entrenamiento para el experimento m_exp1.

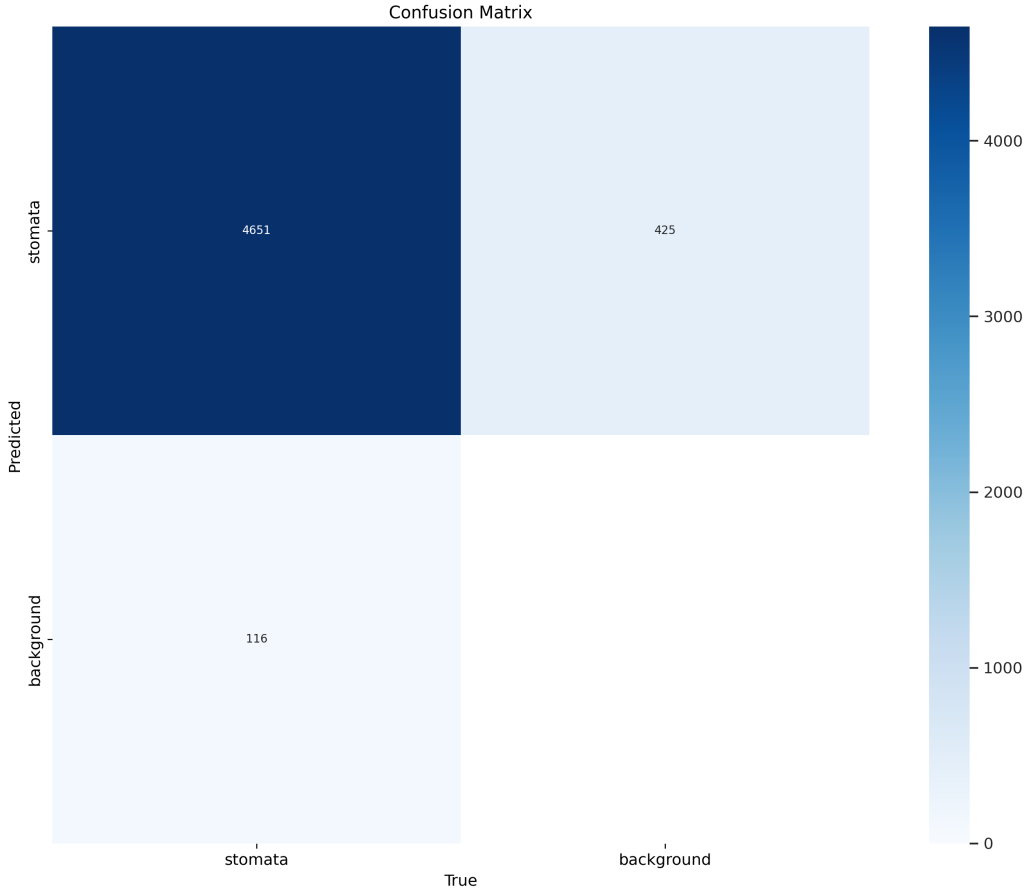


Figura 5.4: Matriz de confusión para el experimento m_exp1.

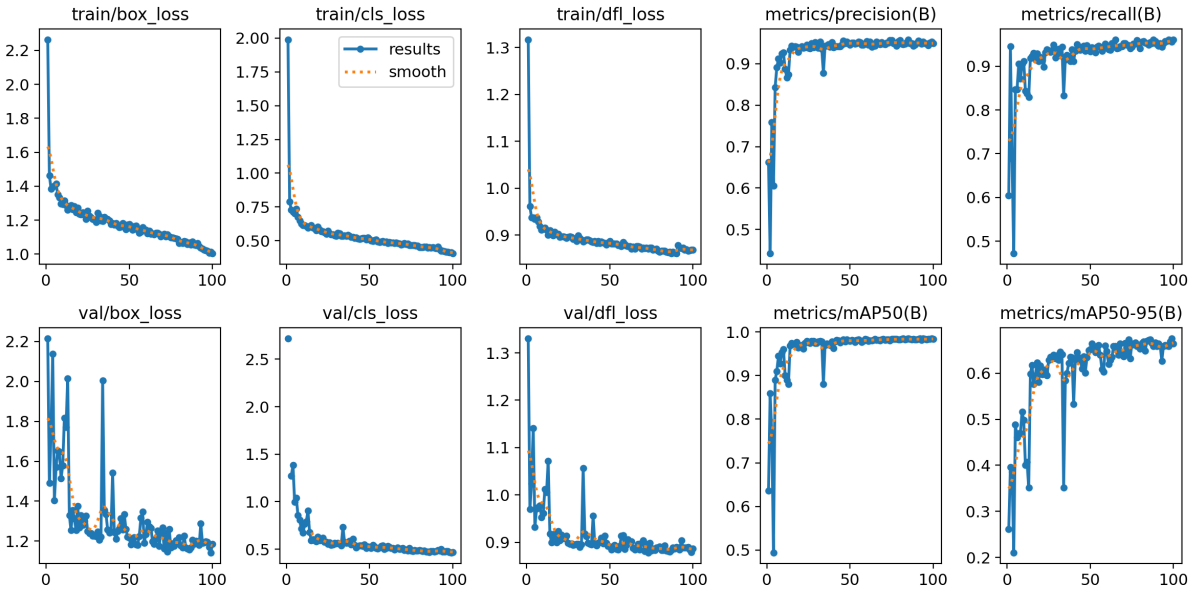


Figura 5.5: Curvas de perdida en el conjunto validación y entrenamiento para el experimento l_exp1.

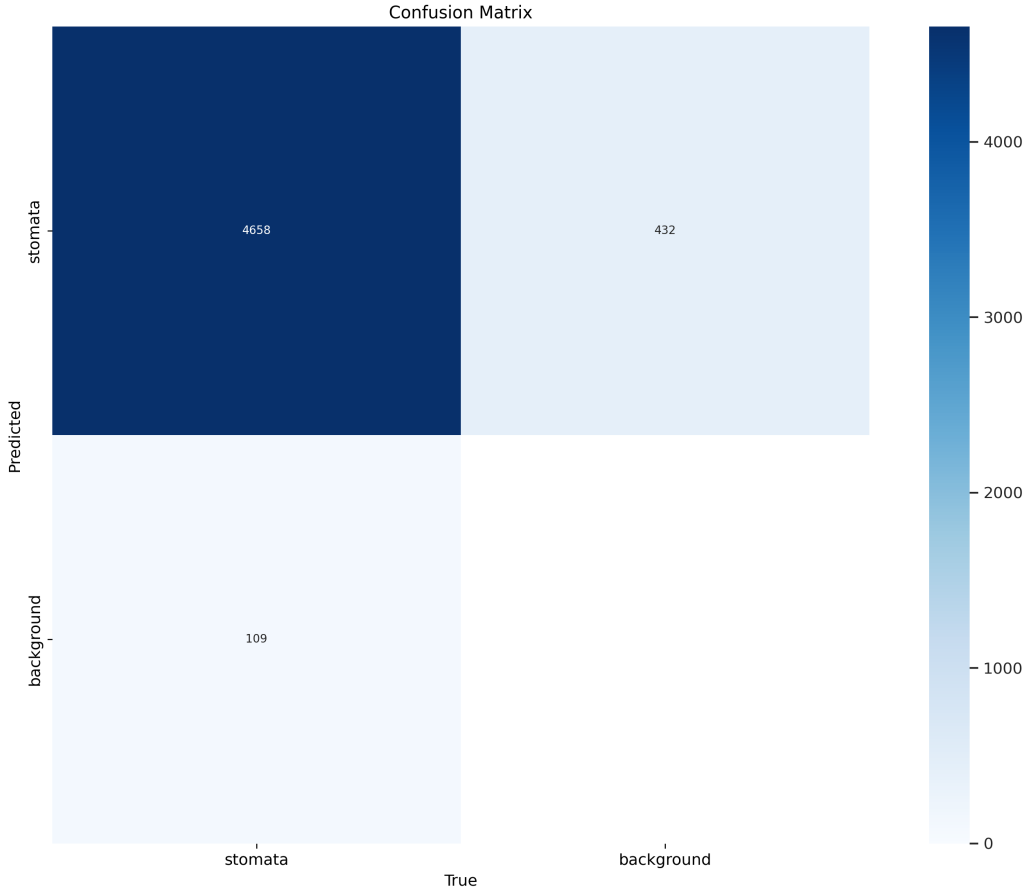


Figura 5.6: Matriz de confusión para el experimento l_exp1.

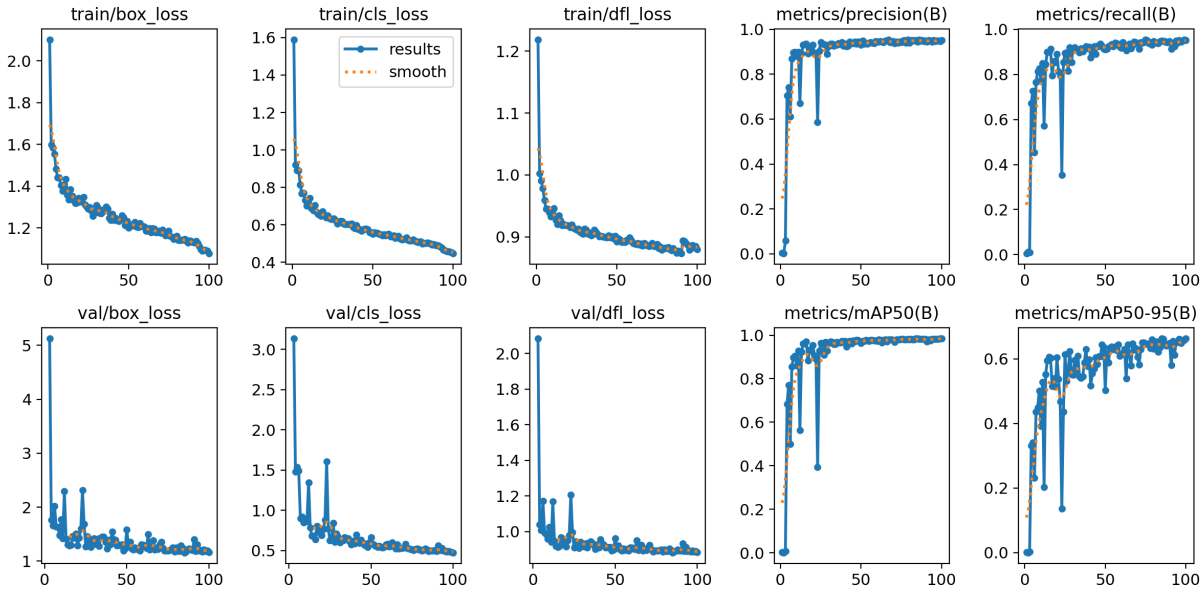


Figura 5.7: Curvas de perdida en el conjunto validación y entrenamiento para el experimento l_exp2.

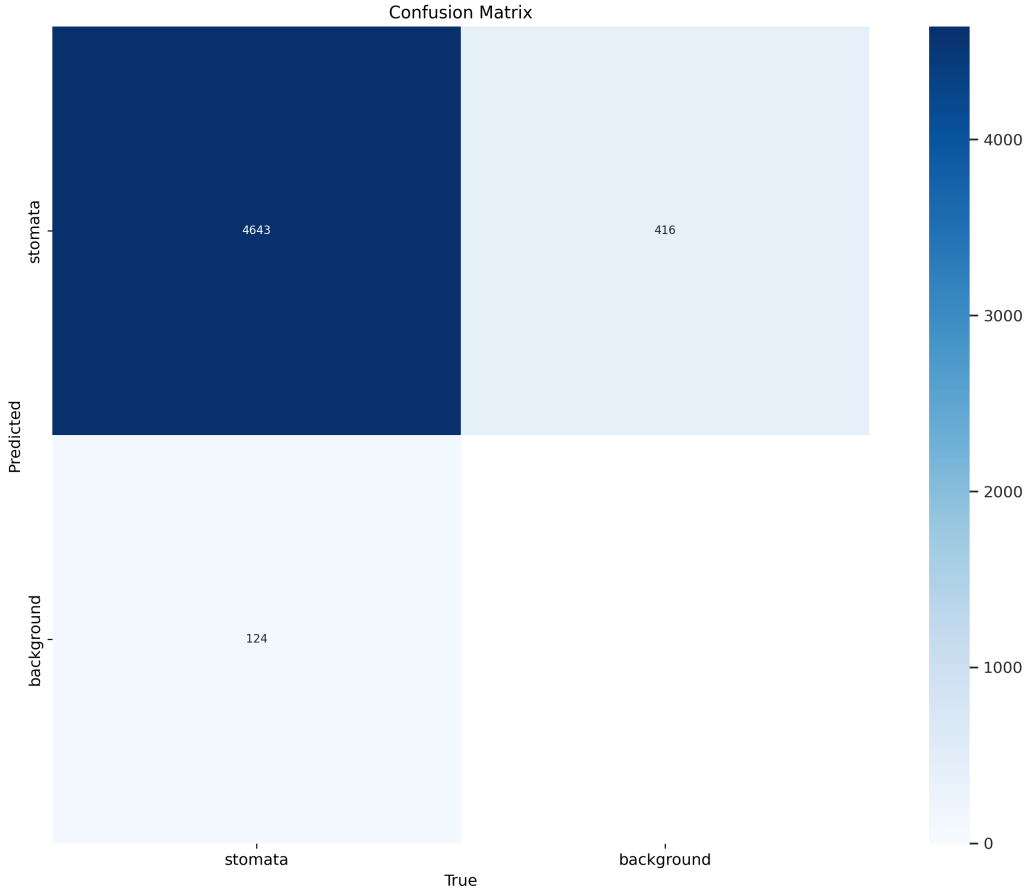


Figura 5.8: Matriz de confusión para el experimento l_exp2.

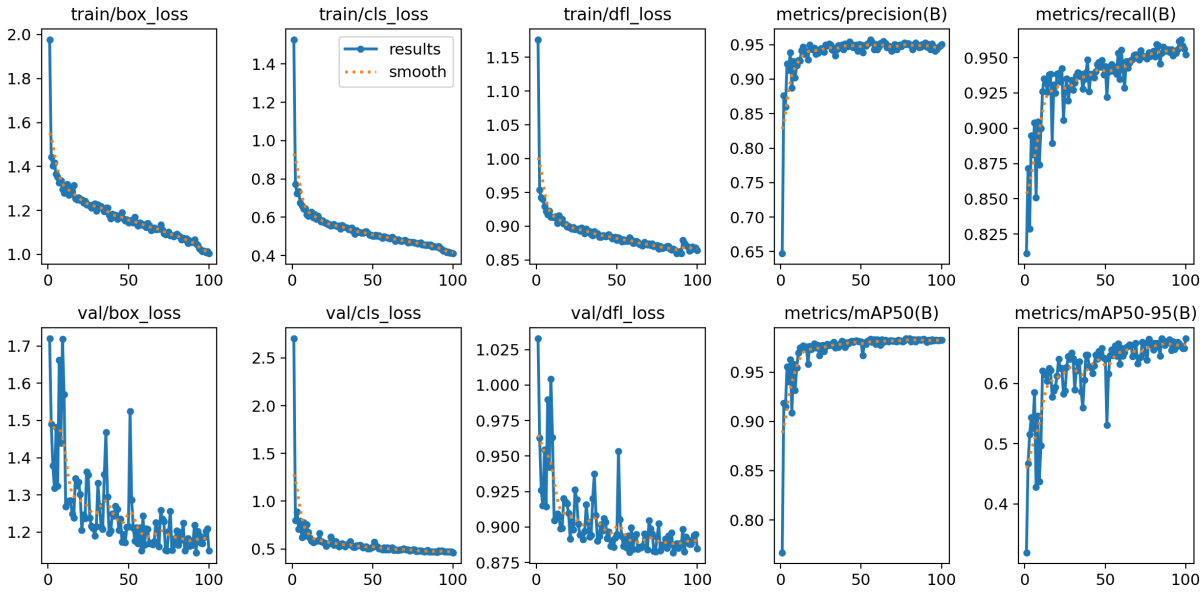


Figura 5.9: Curvas de perdida en el conjunto validación y entrenamiento para el experimento l_exp3.

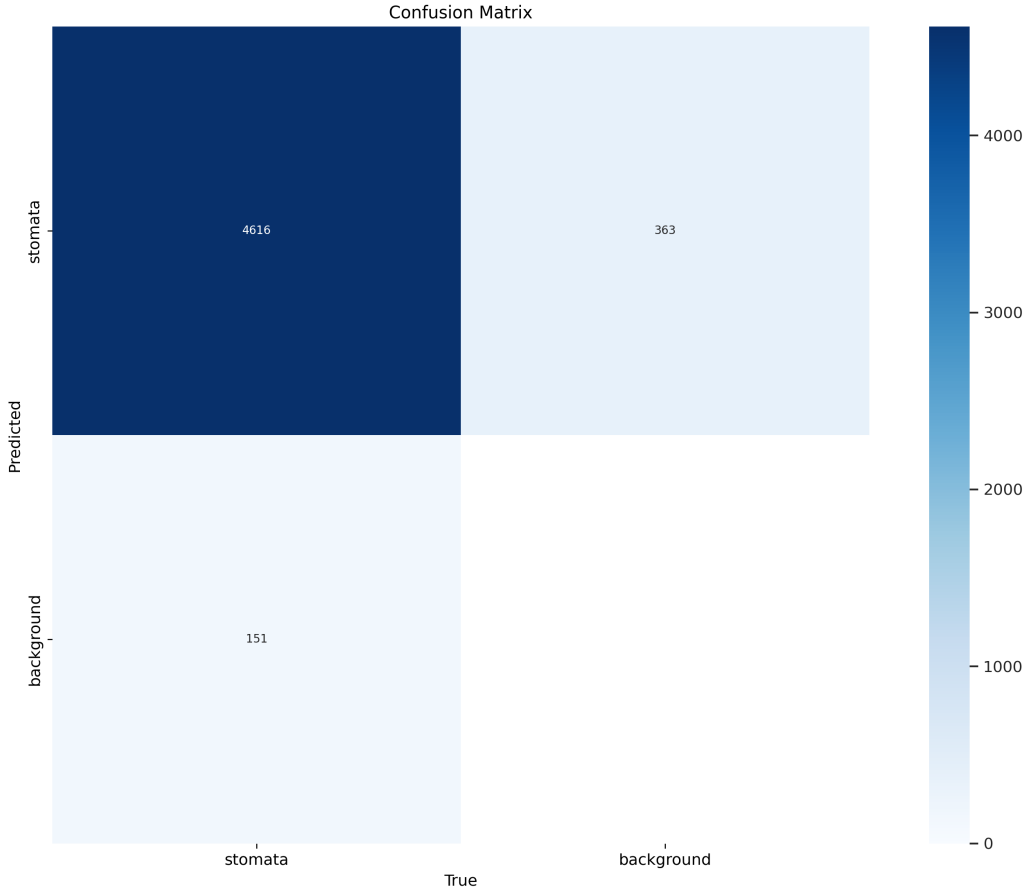


Figura 5.10: Matriz de confusión para el experimento l_exp3.

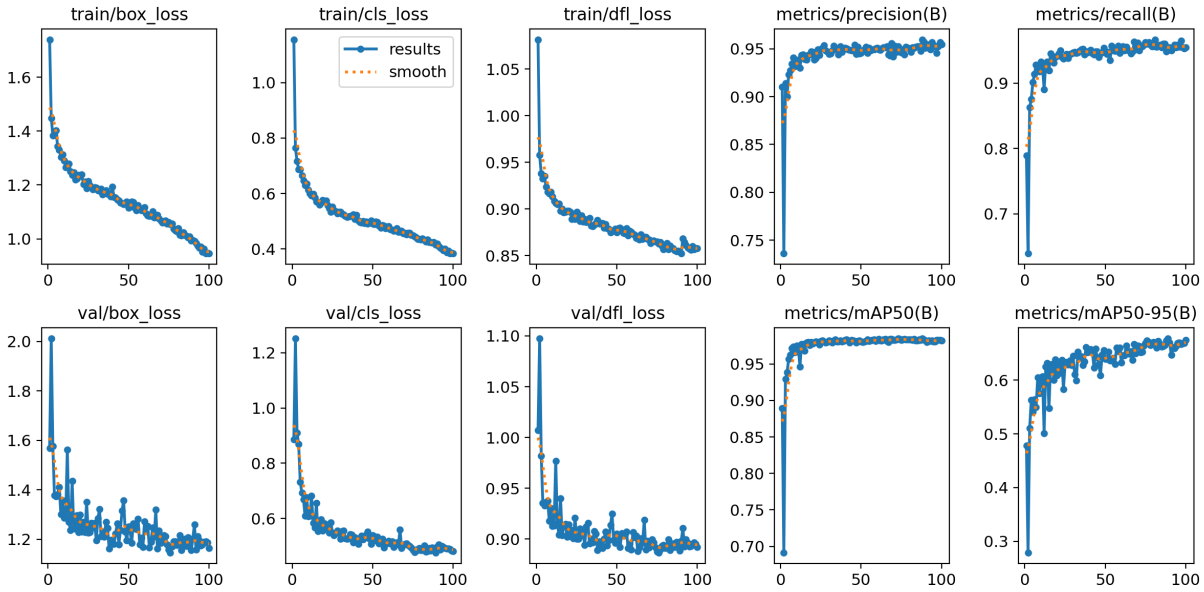


Figura 5.11: Curvas de perdida en el conjunto validación y entrenamiento para el experimento x_exp1.

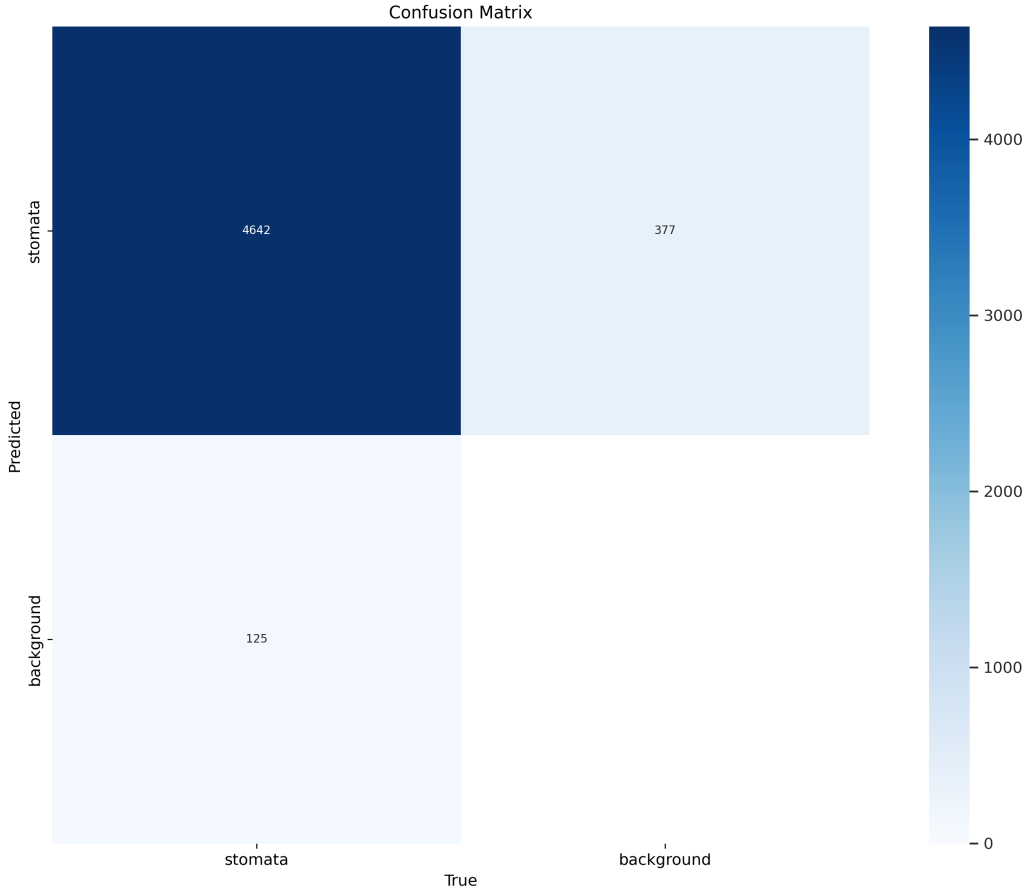


Figura 5.12: Matriz de confusión para el experimento x_exp1.

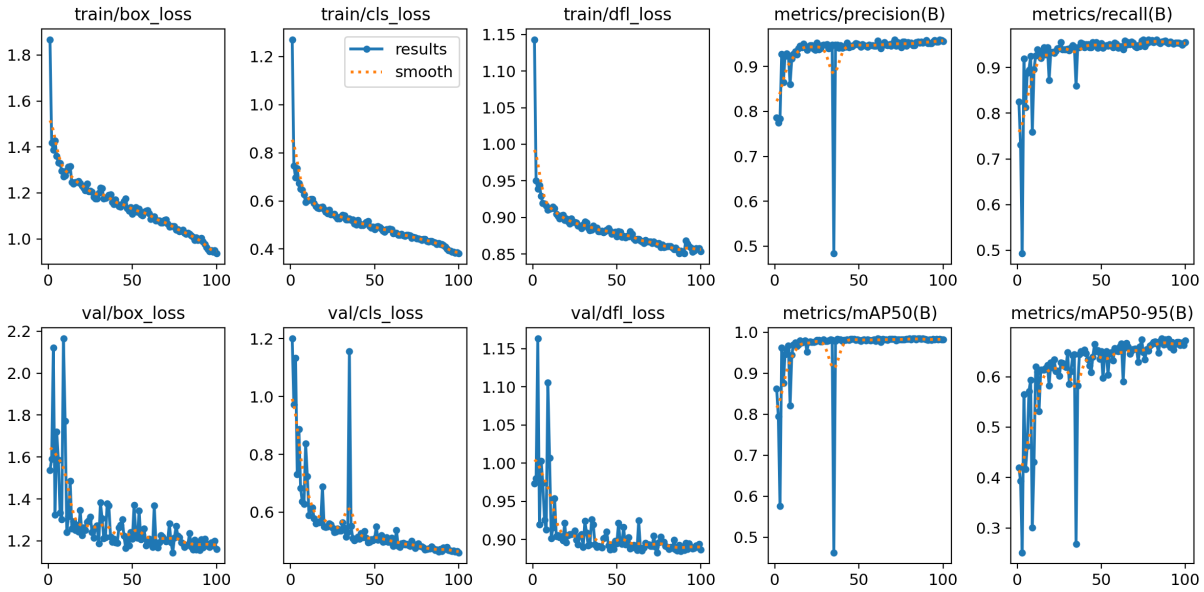


Figura 5.13: Curvas de perdida en el conjunto validación y entrenamiento para el experimento x_exp2.

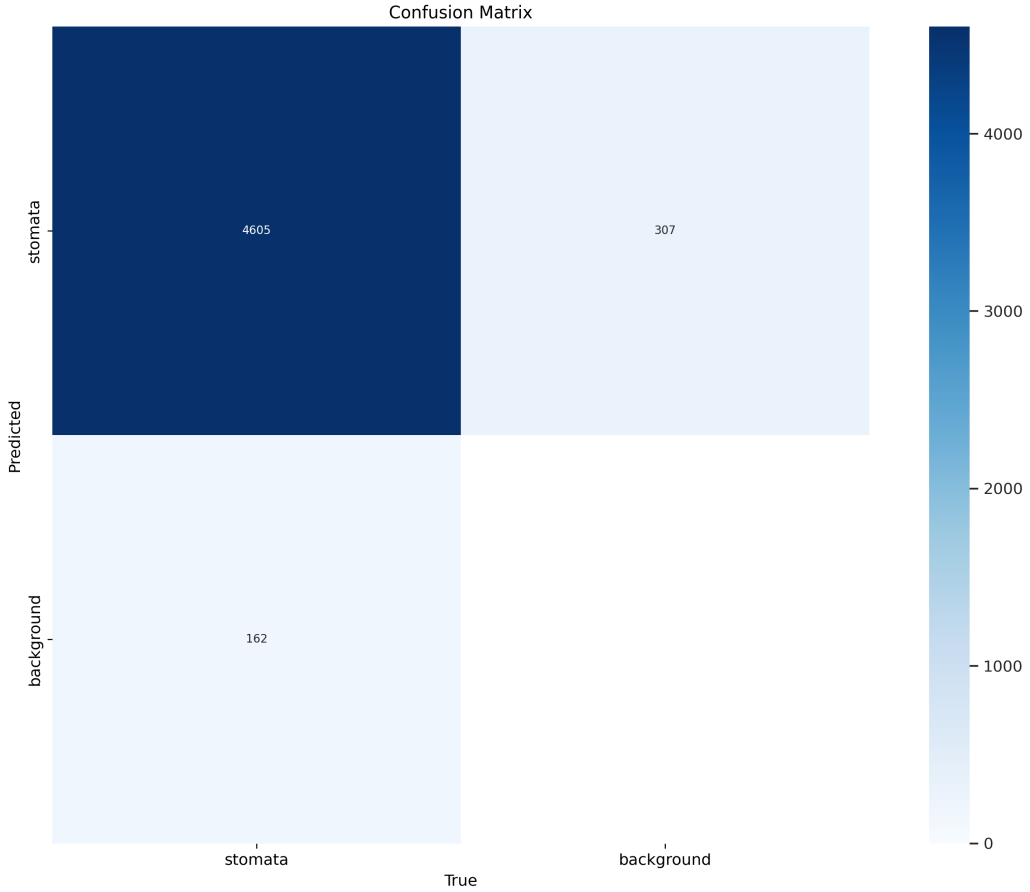


Figura 5.14: Matriz de confusión para el experimento x_exp2.

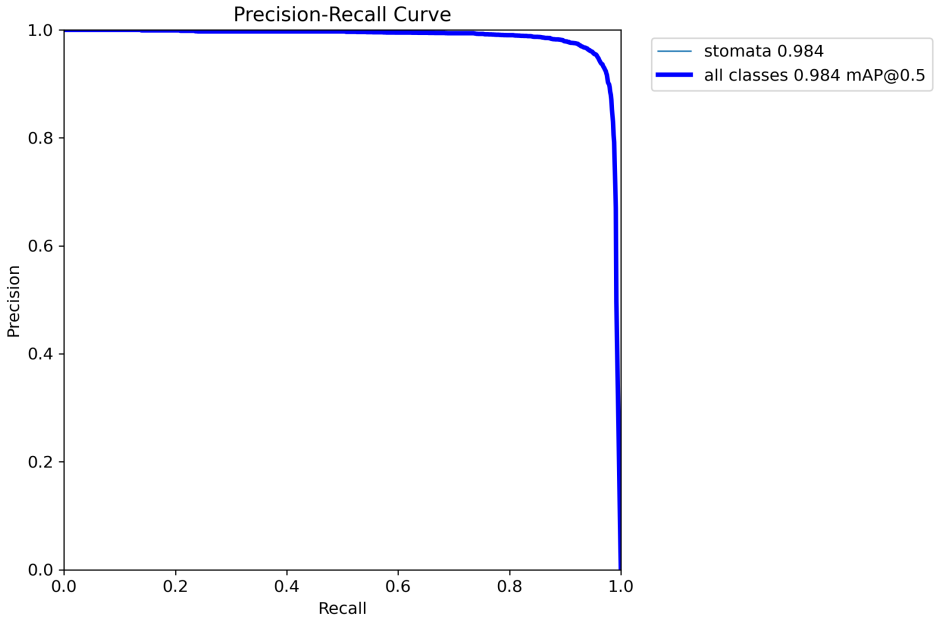


Figura 5.15: Curva Precision_Recall para experimento x_exp2.

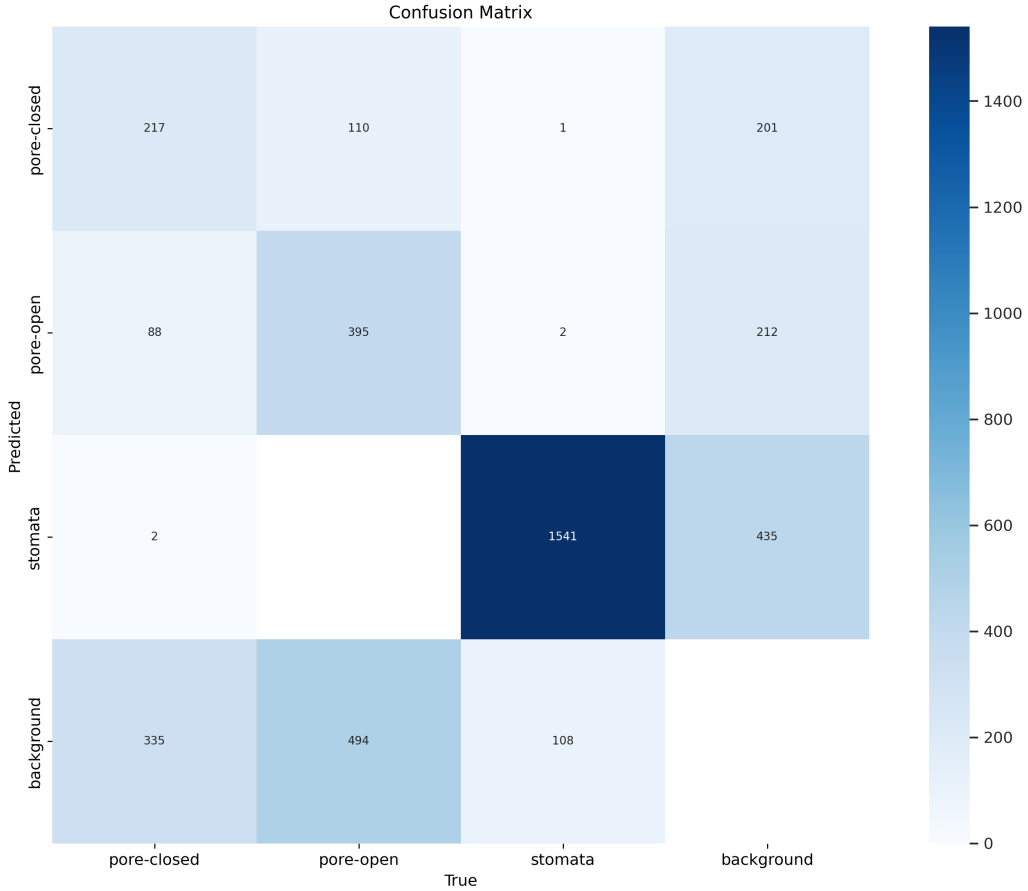


Figura 5.16: Matriz de confusión para el experimento 1.

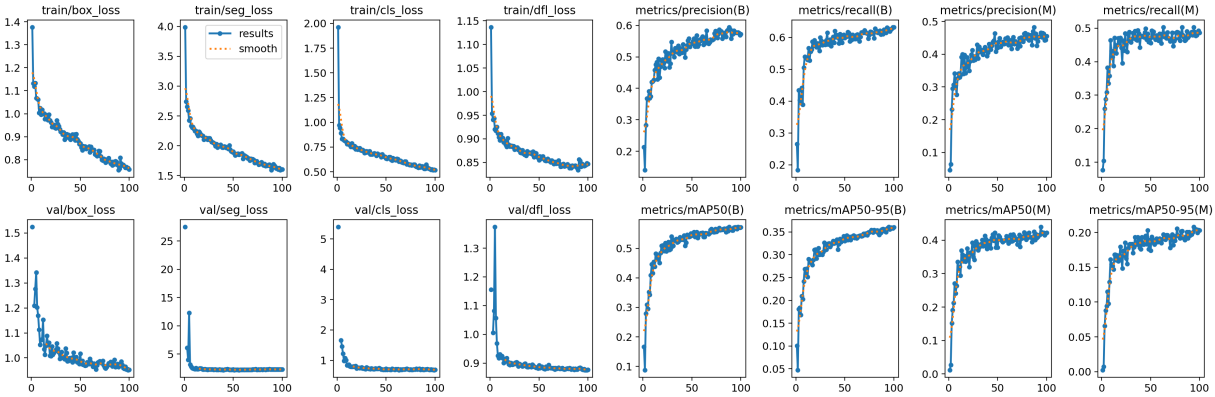


Figura 5.17: Curvas de perdida en el conjunto validación y entrenamiento para el experimento 1.

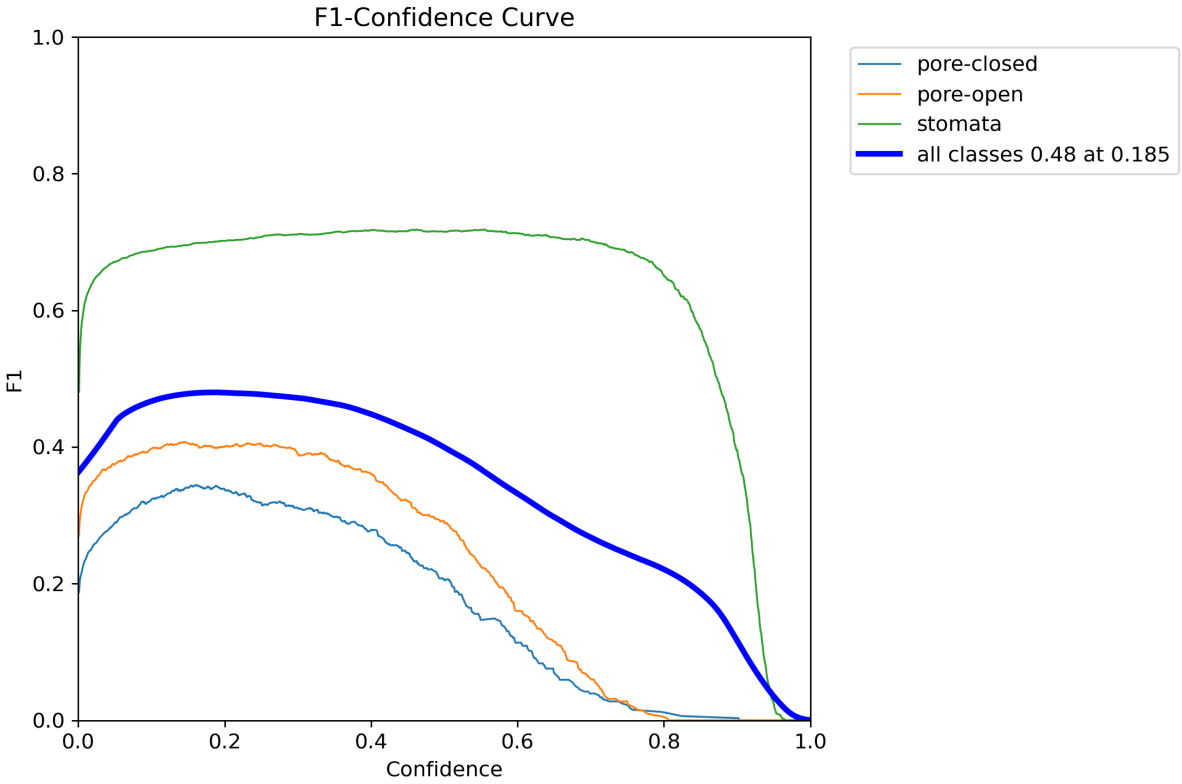


Figura 5.18: Curva de evolución para métrica F1_score en el experimento 1.

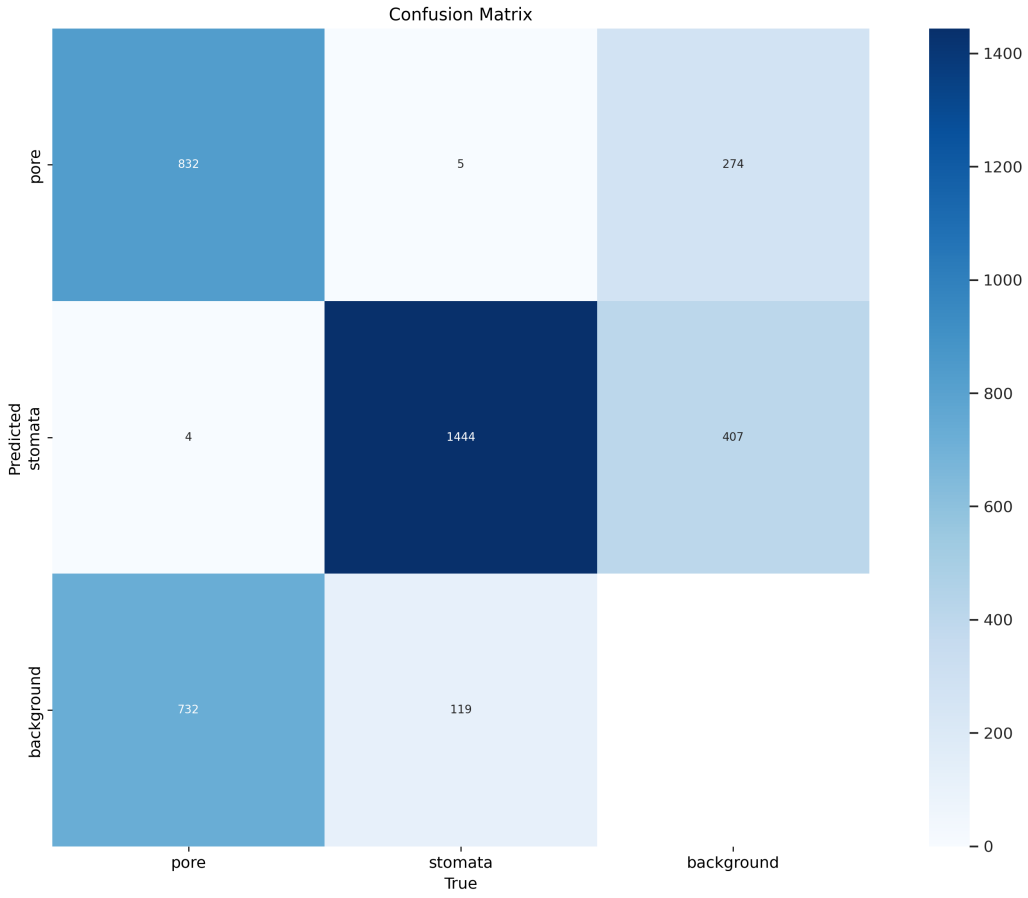


Figura 5.19: Matriz de confusión para el experimento 2.

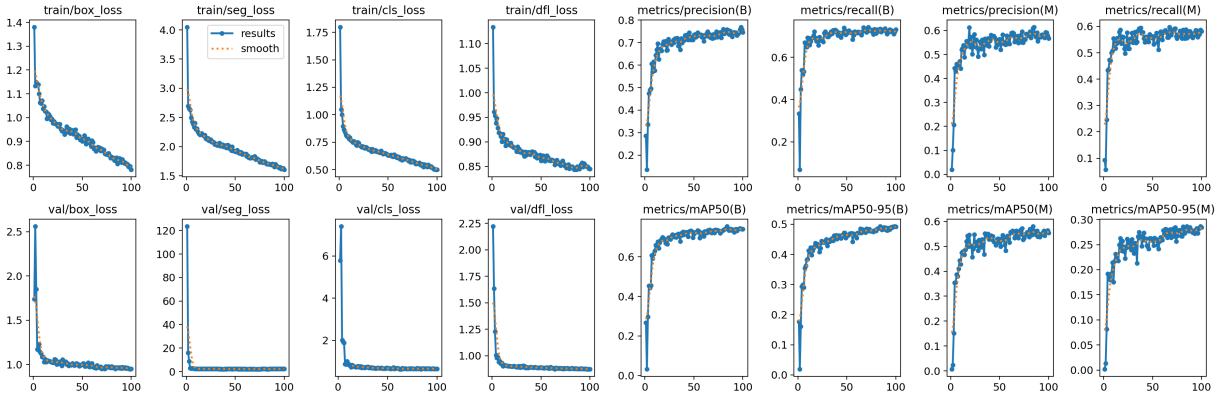


Figura 5.20: Curvas de perdida en el conjunto validación y entrenamiento para el experimento 2.

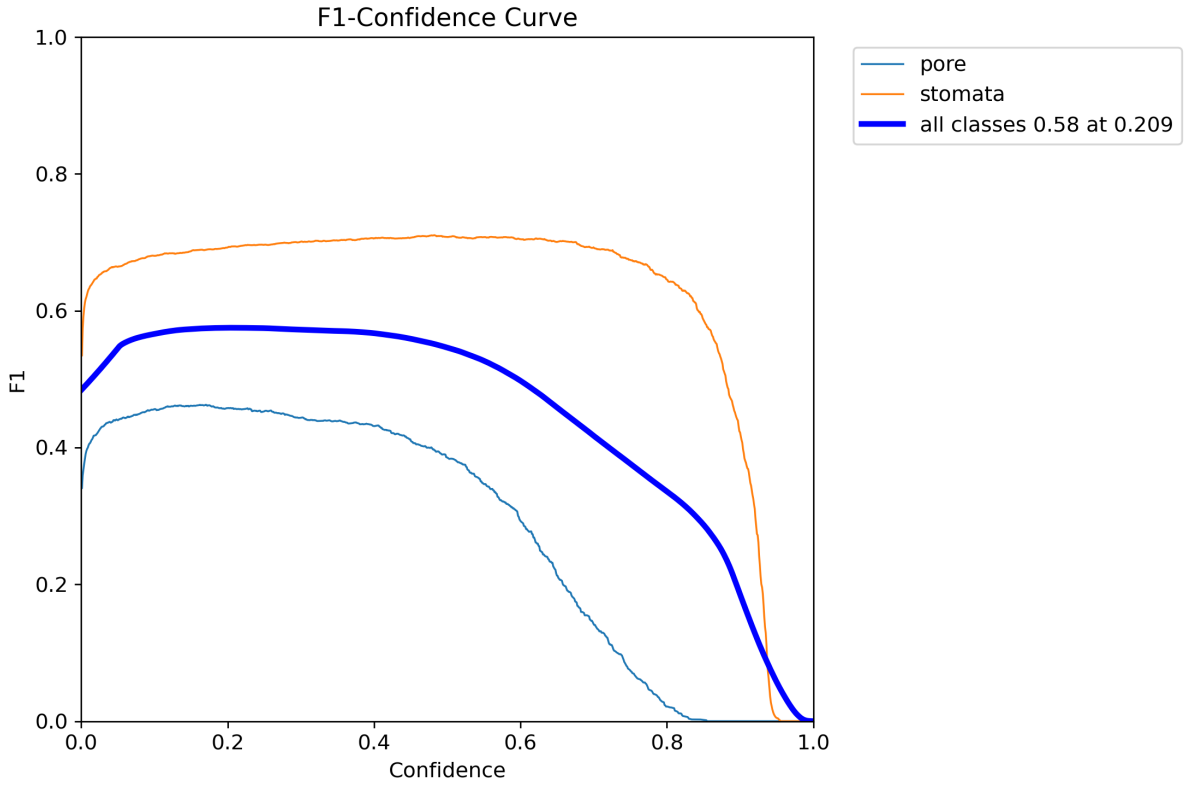


Figura 5.21: Curva de evolución para métrica F1_score en el experimento 2.

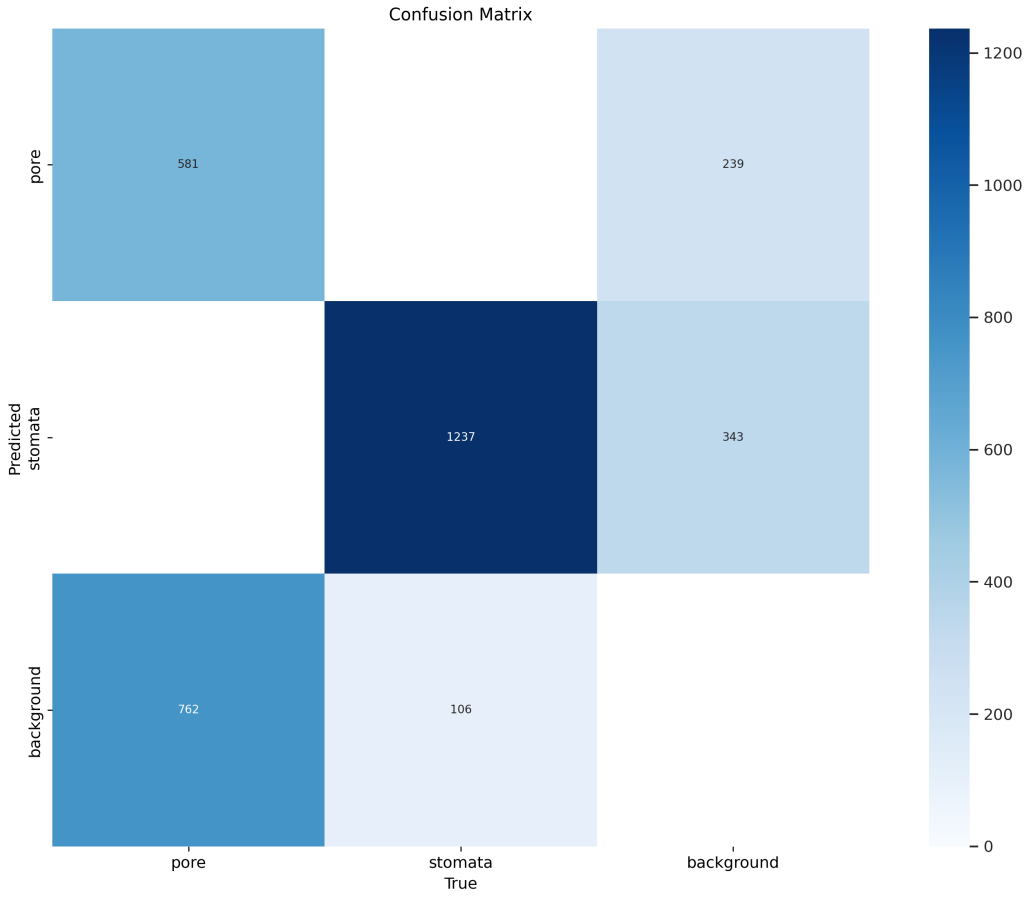


Figura 5.22: Matriz de confusión para el experimento 3.

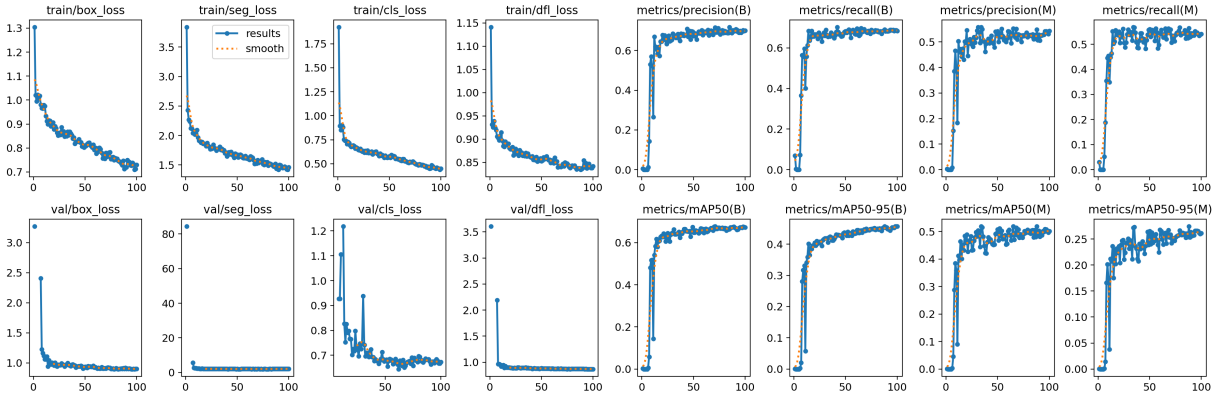


Figura 5.23: Curvas de perdida en el conjunto validación y entrenamiento para el experimento 3.

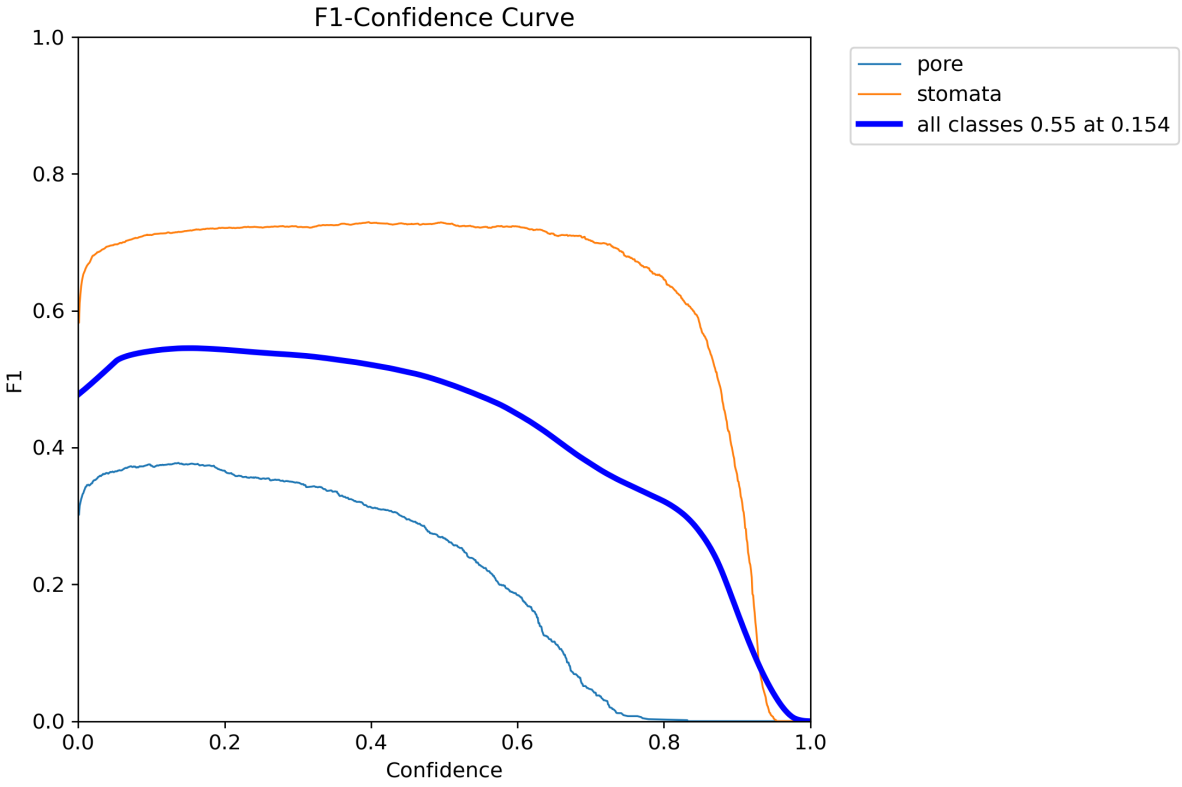


Figura 5.24: Curva de evolución para métrica F1_score en el experimento 3.

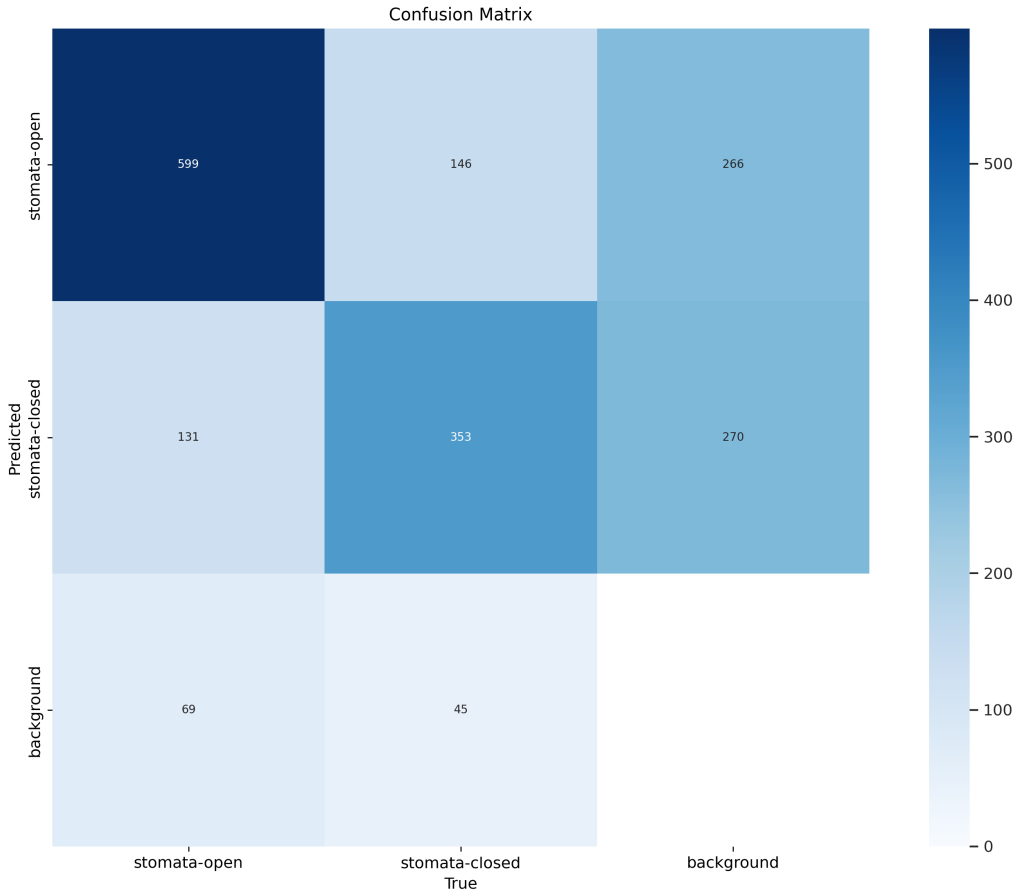


Figura 5.25: Matriz de confusión para el experimento 4.

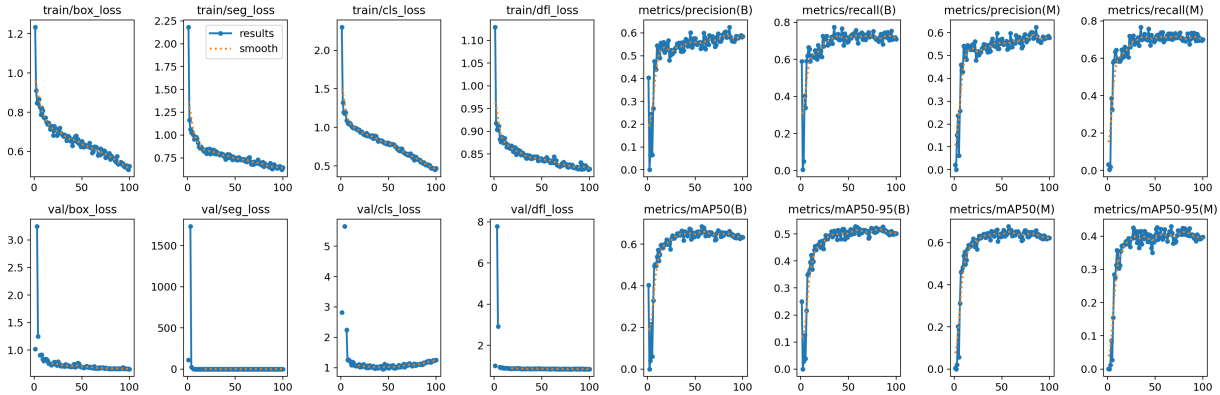


Figura 5.26: Curvas de perdida en el conjunto validación y entrenamiento para el experimento 4.

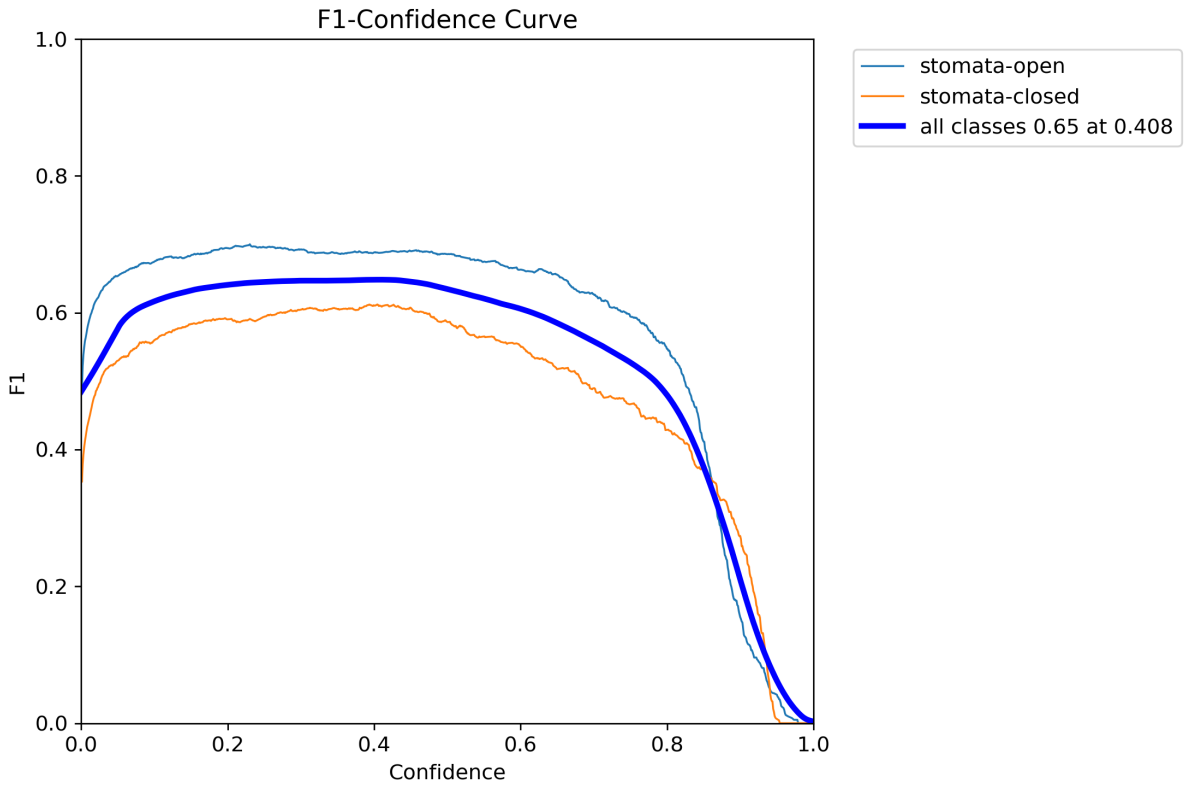


Figura 5.27: Curva de evolución para métrica F1_score en el experimento 4.

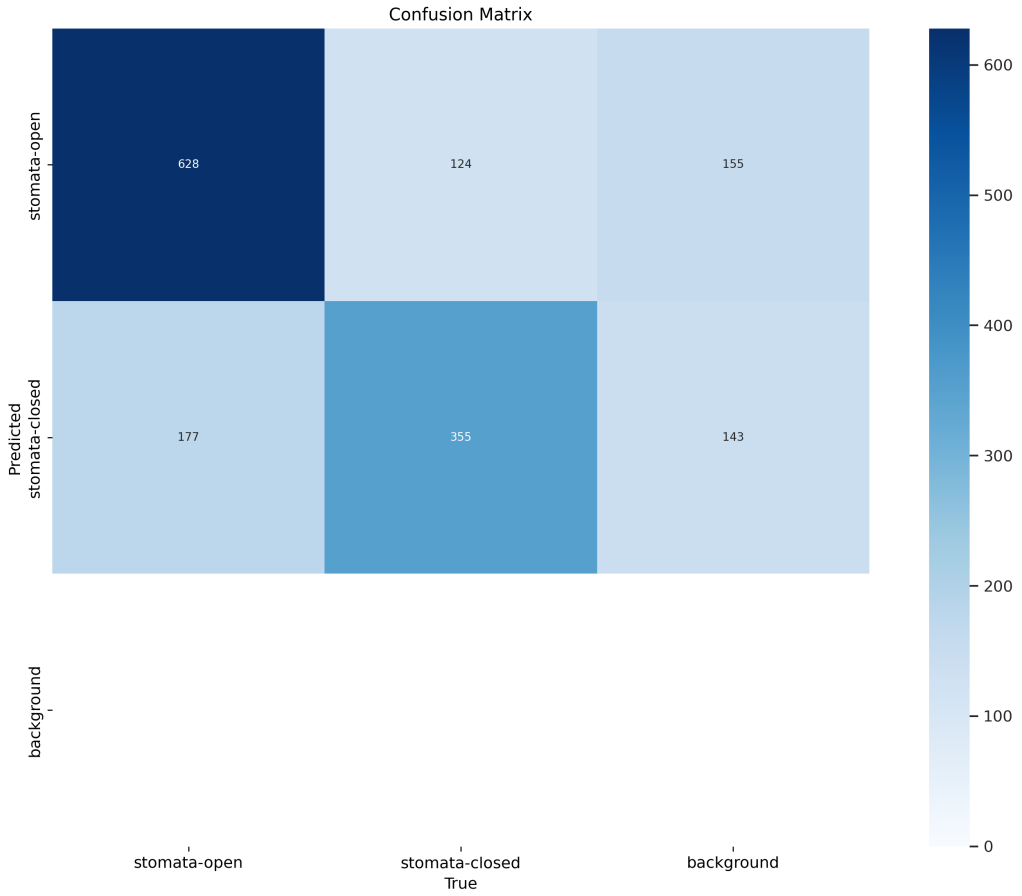


Figura 5.28: Matriz de confusión para el experimento 5.

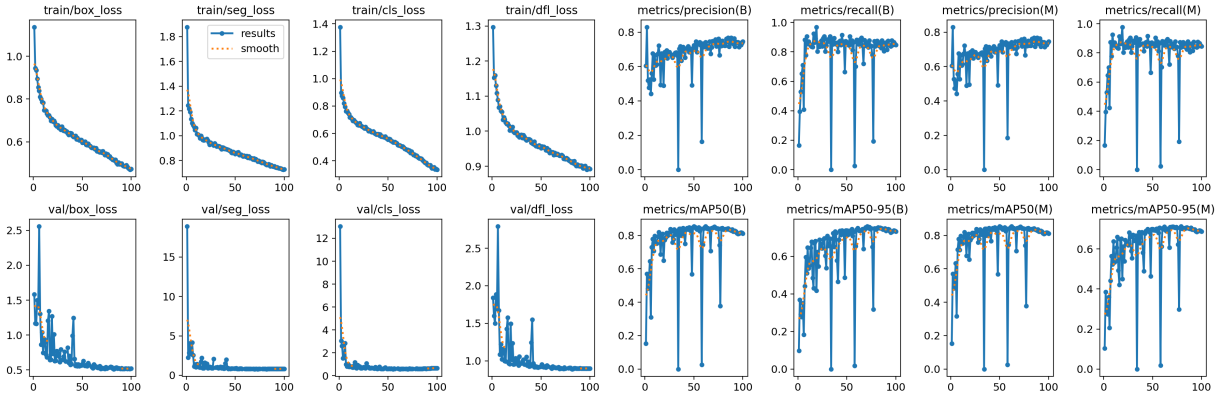


Figura 5.29: Curvas de perdida en el conjunto validación y entrenamiento para el experimento 5.

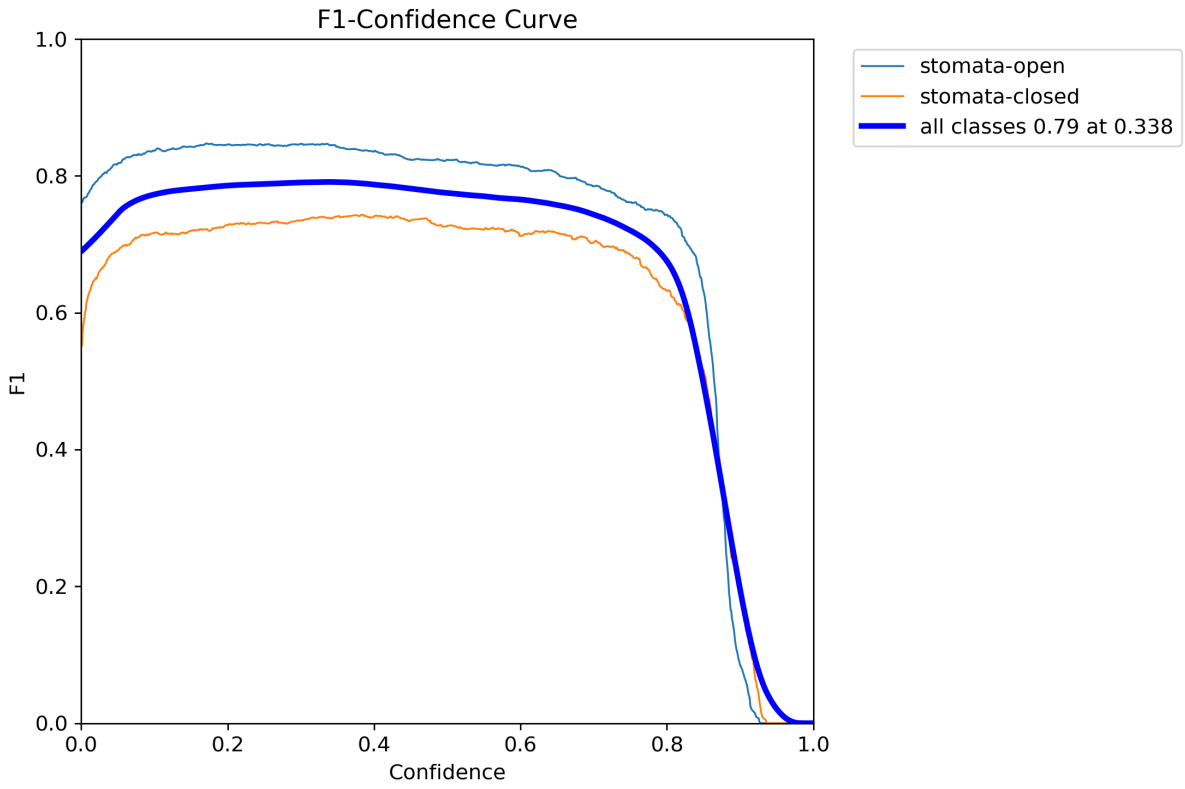


Figura 5.30: Curva de evolución para métrica F1_score en el experimento 5.

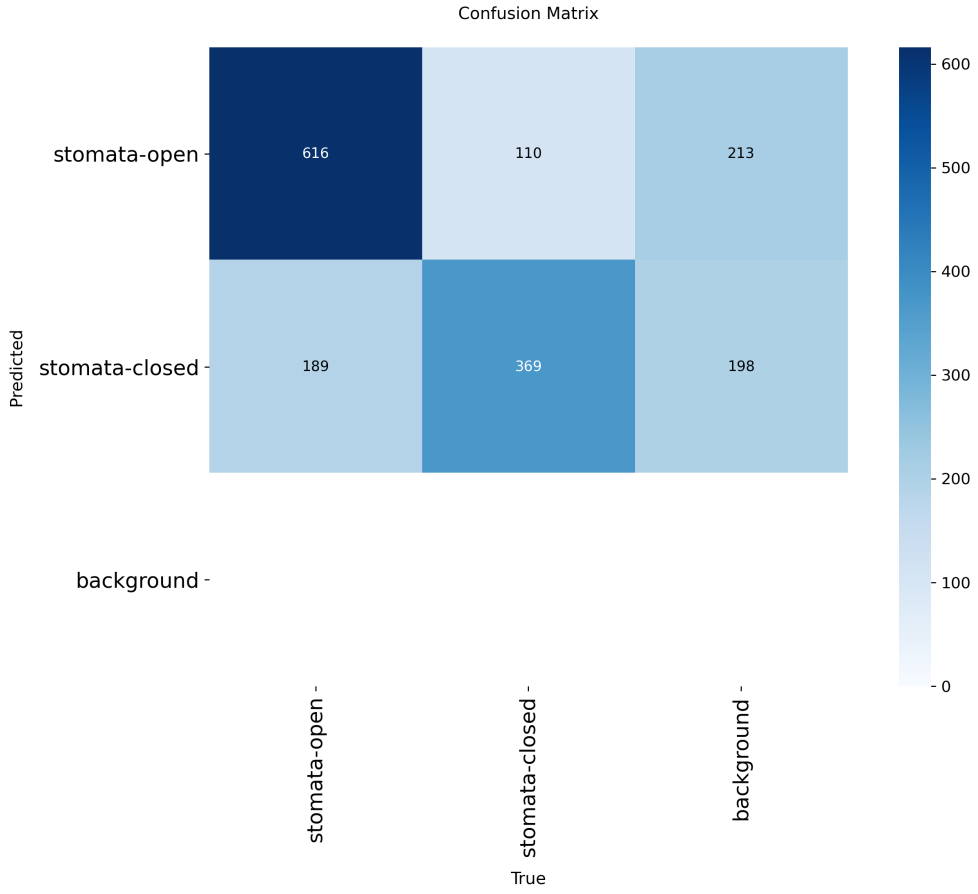


Figura 5.31: Matriz de confusión para el experimento 1.

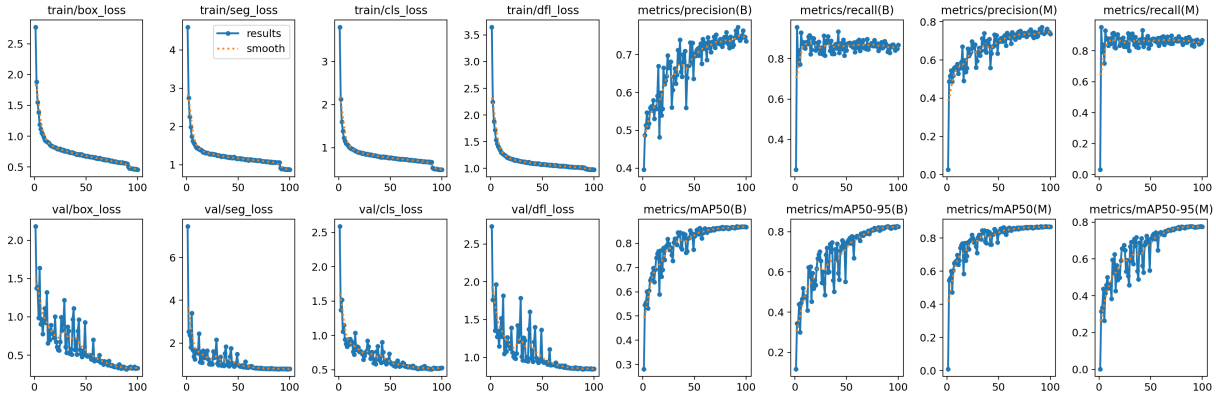


Figura 5.32: Curvas de perdida en el conjunto validación y entrenamiento para el experimento 1.

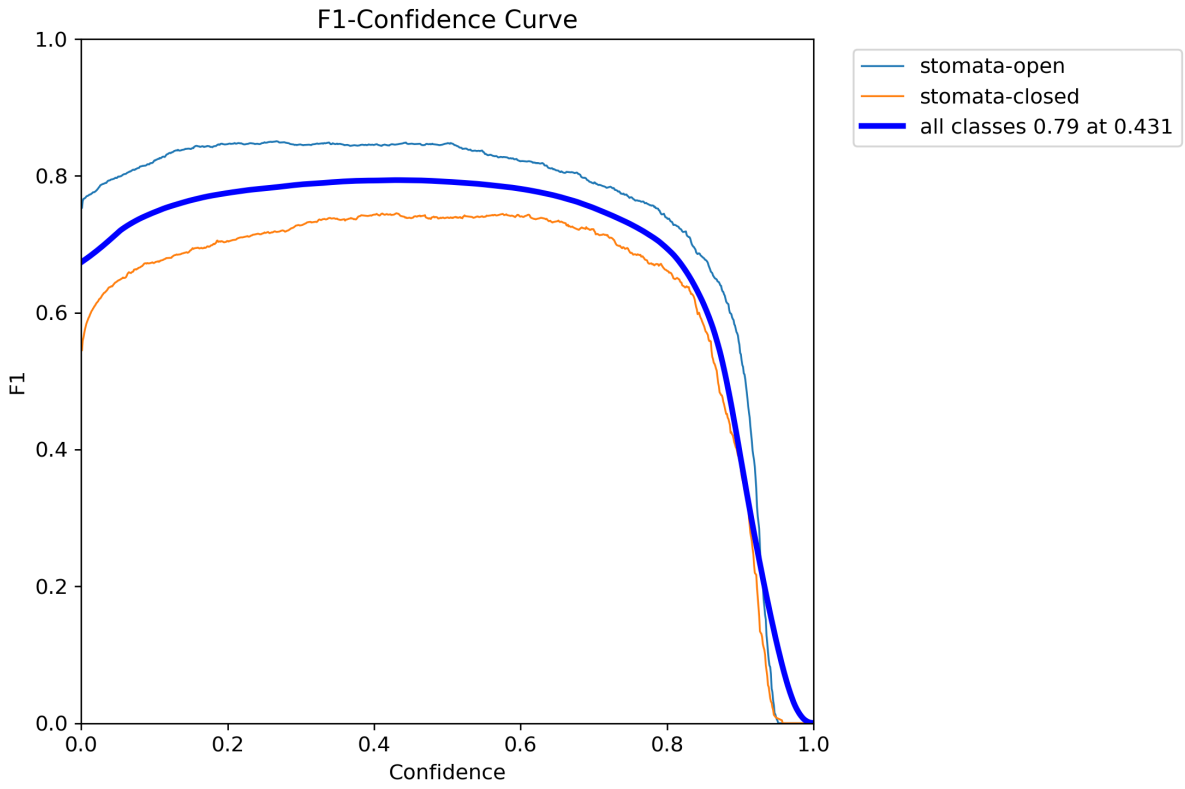


Figura 5.33: Curva de evolución para métrica F1_score en el experimento 1.

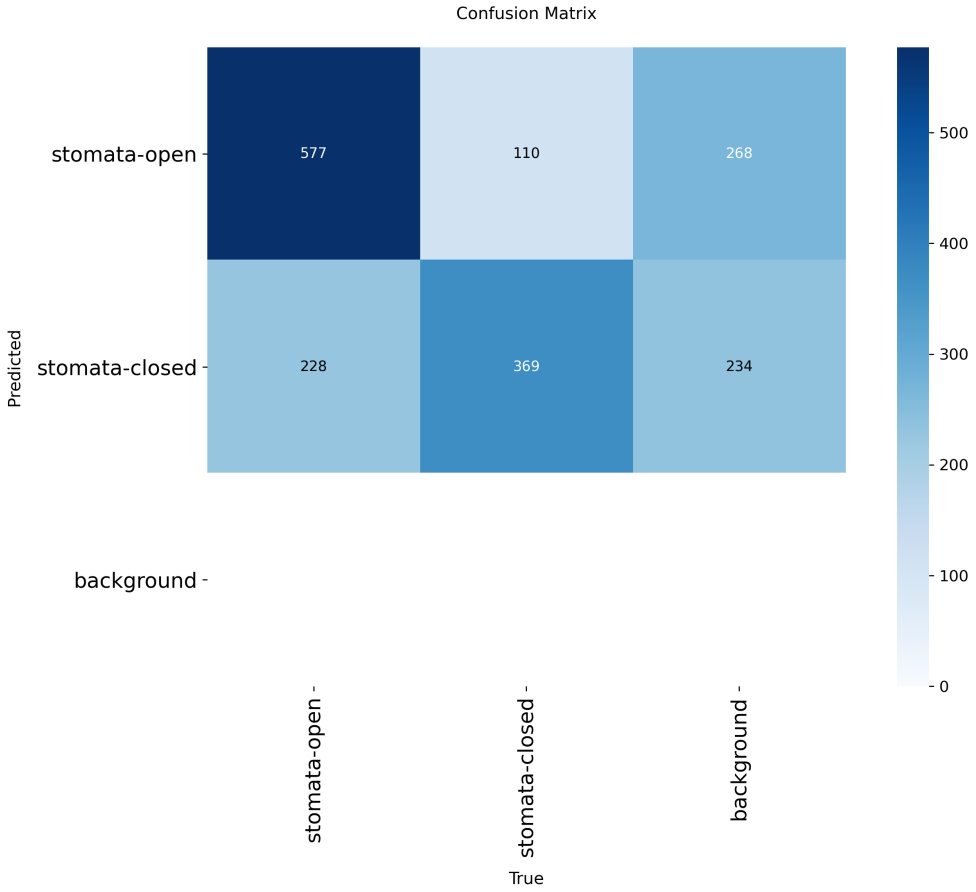


Figura 5.34: Matriz de confusión para el experimento 3.

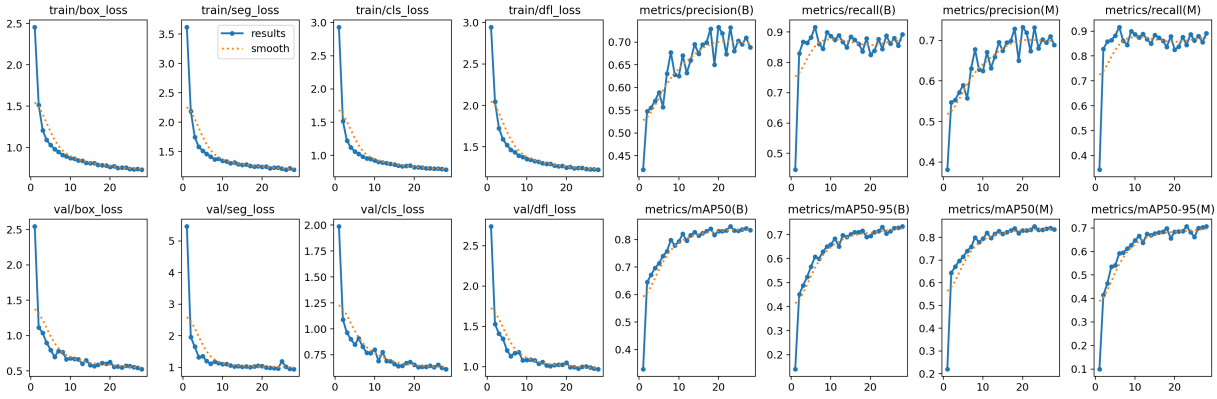


Figura 5.35: Curvas de perdida en el conjunto validación y entrenamiento para el experimento 3.

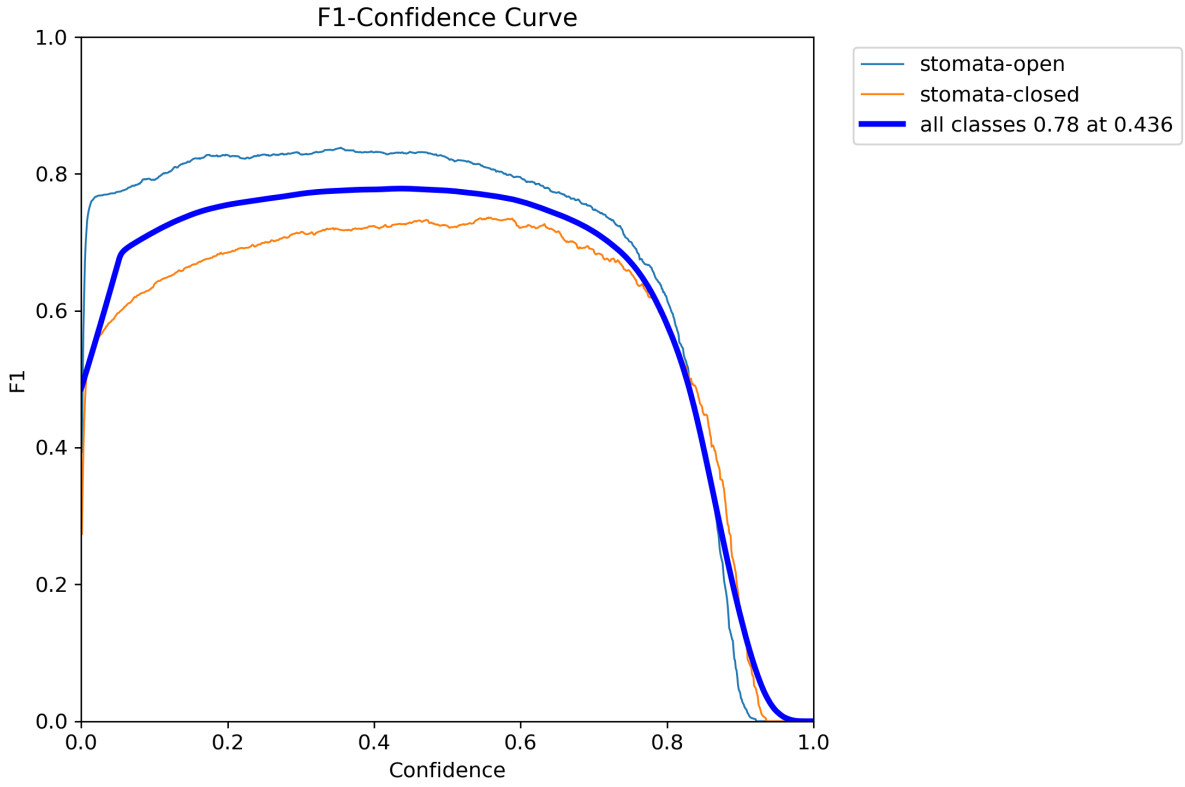


Figura 5.36: Curva de evolución para métrica F1_score en el experimento 3.

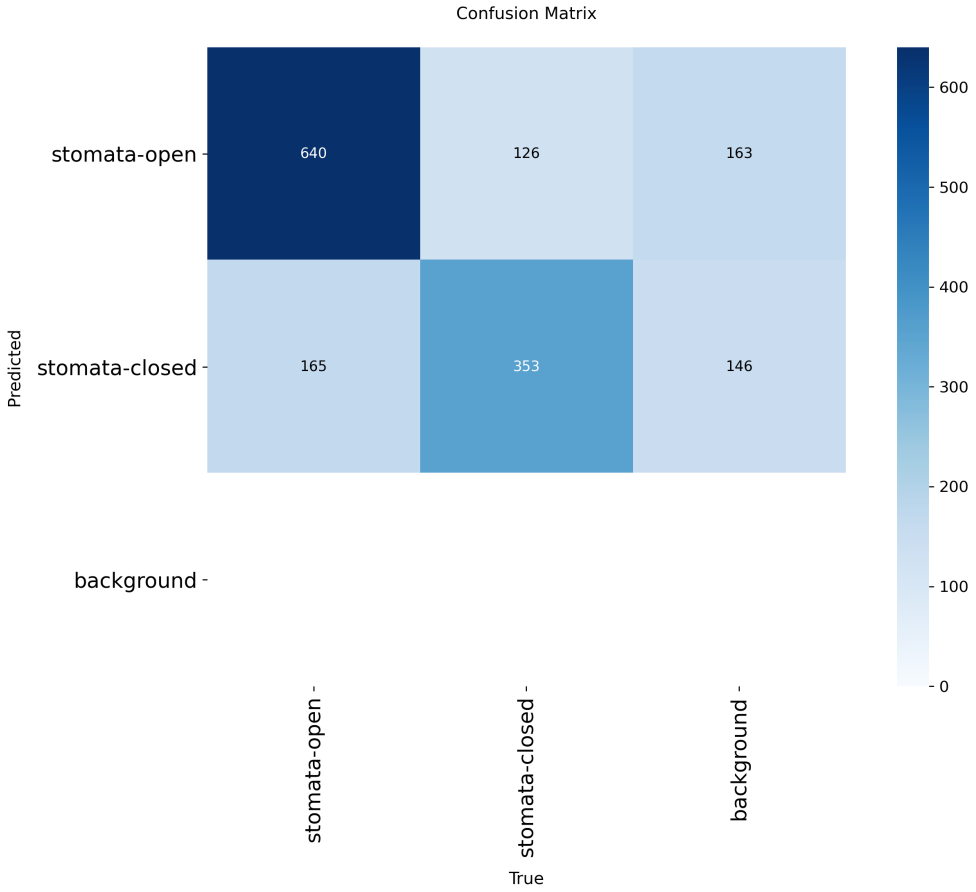


Figura 5.37: Matriz de confusión para el experimento 4.

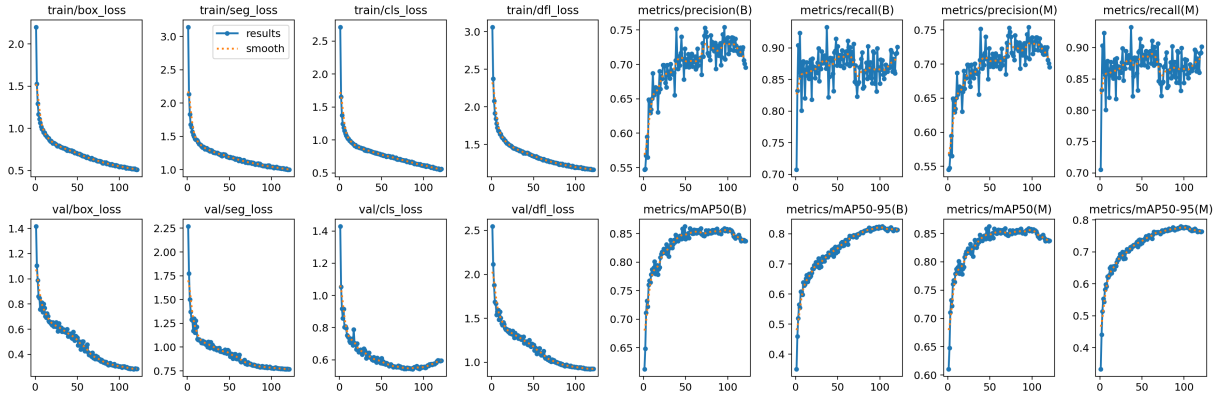


Figura 5.38: Curvas de perdida en el conjunto validación y entrenamiento para el experimento 4.

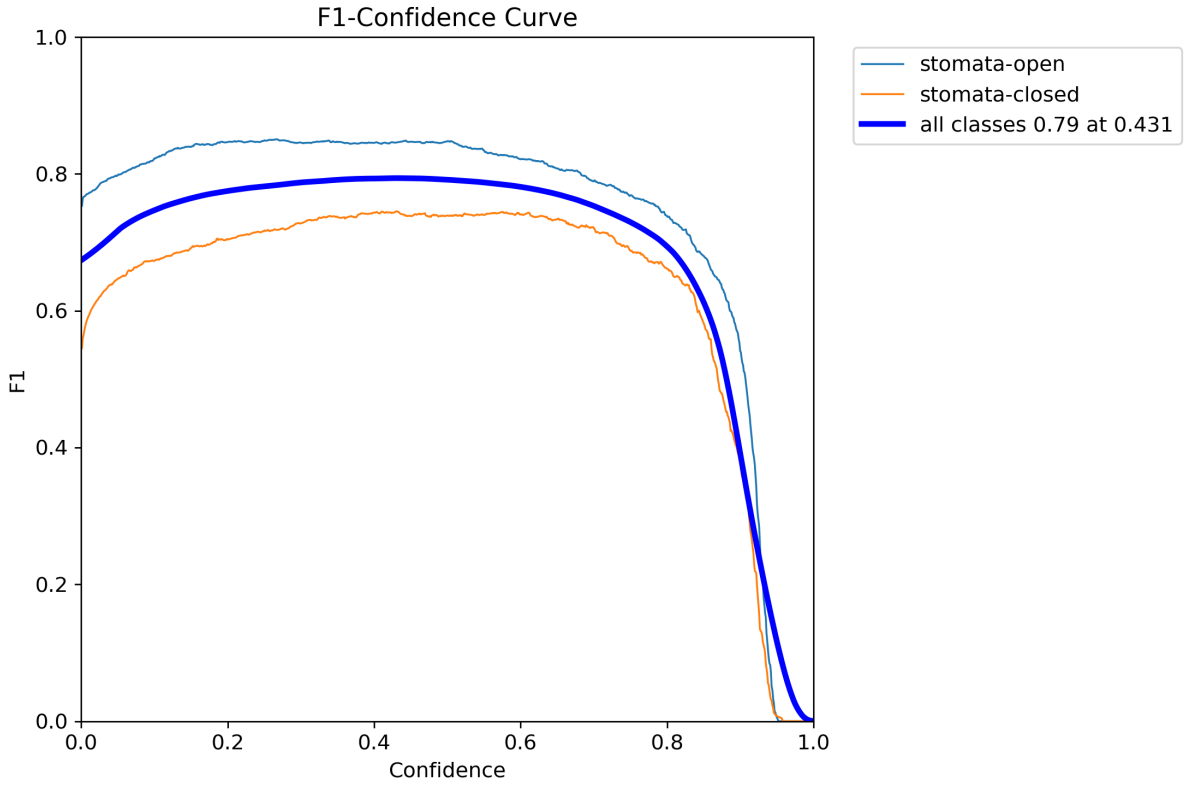


Figura 5.39: Curva de evolución para métrica F1_score en el experimento 4.

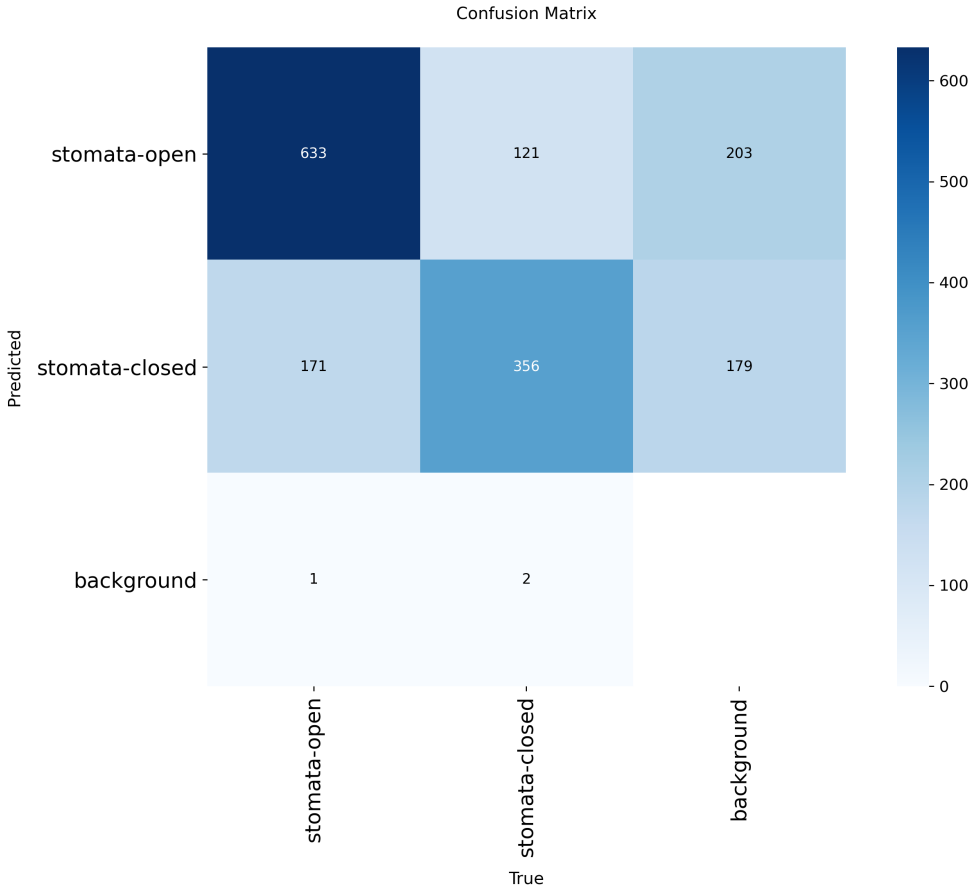


Figura 5.40: Matriz de confusión para el experimento 5.

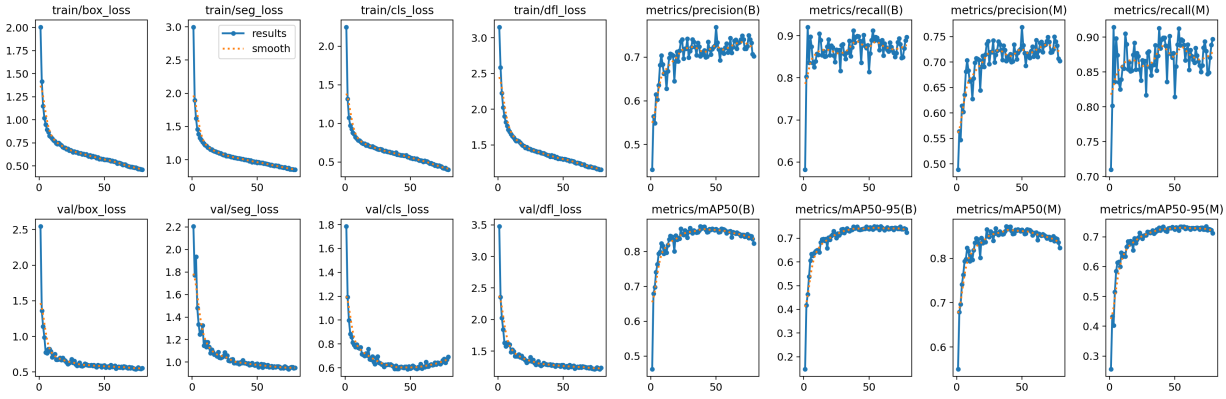


Figura 5.41: Curvas de perdida en el conjunto validación y entrenamiento para el experimento 5.

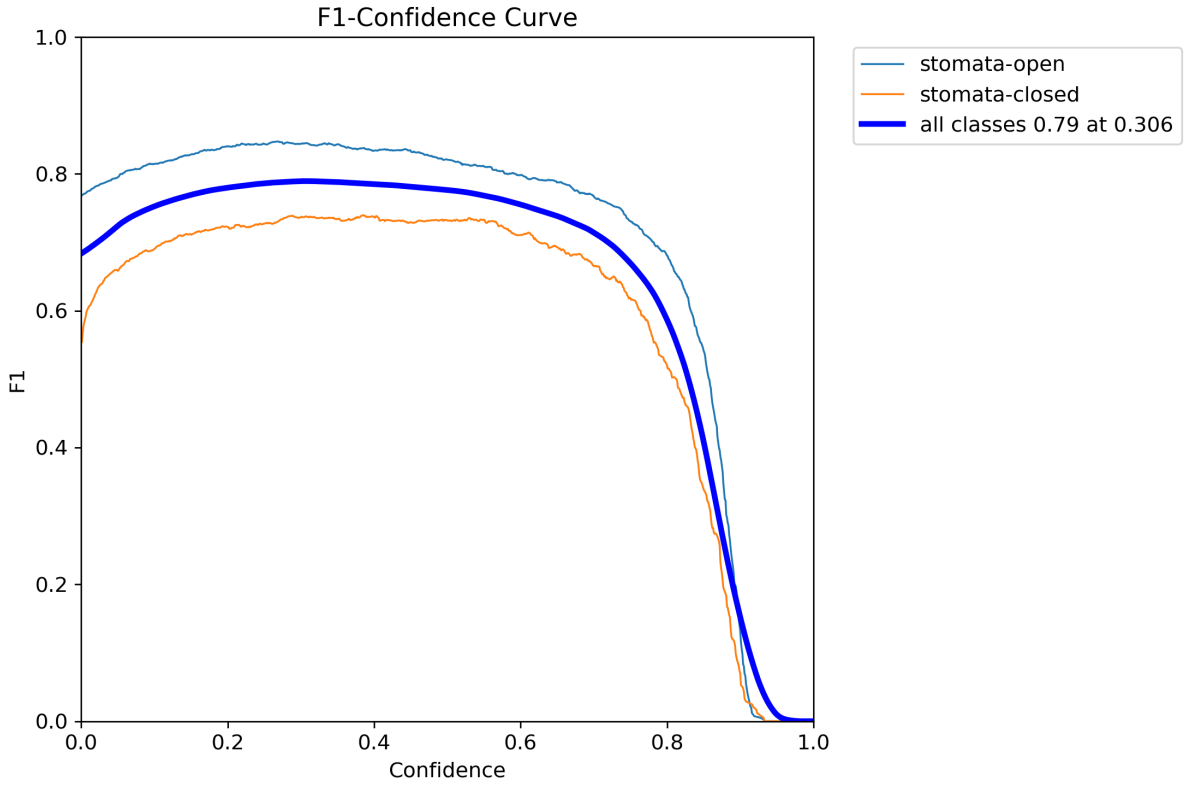


Figura 5.42: Curva de evolución para métrica F1_score en el experimento 5.

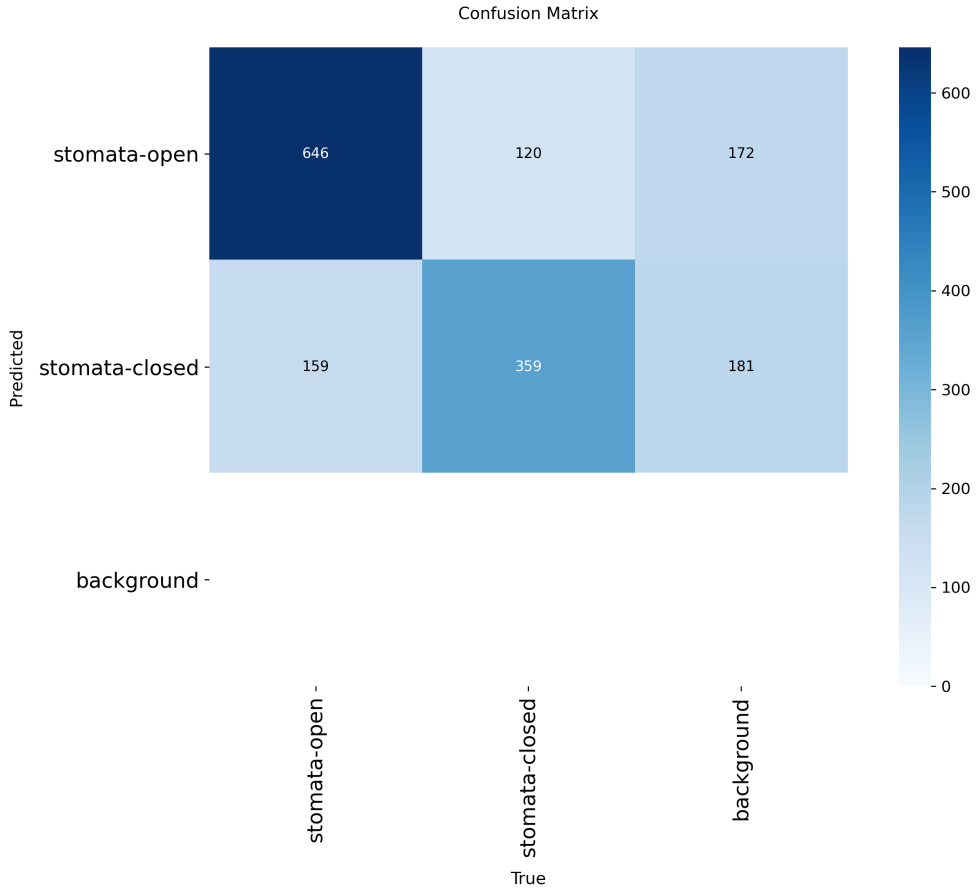


Figura 5.43: Matriz de confusión para el experimento 6.

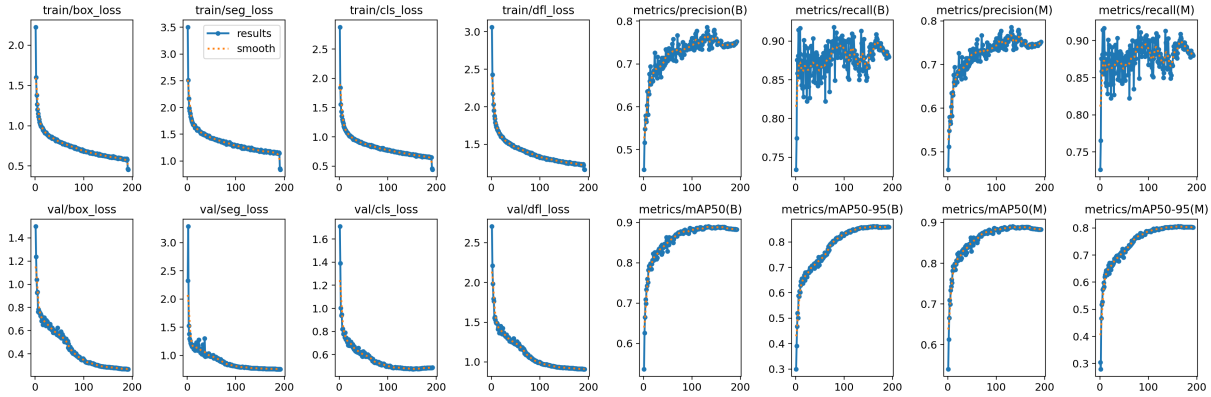


Figura 5.44: Curvas de perdida en el conjunto validación y entrenamiento para el experimento 6.

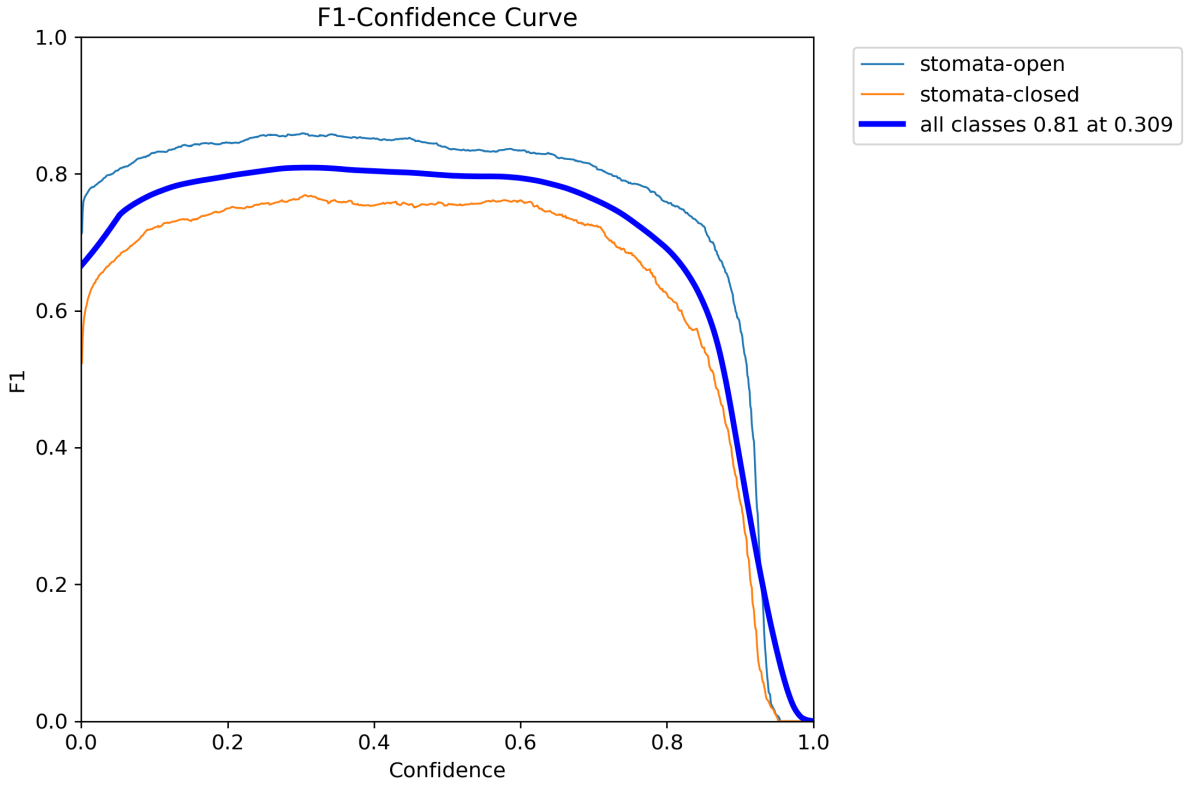


Figura 5.45: Curva de evolución para métrica F1_score en el experimento 6.