



TRABAJO FIN DE GRADO  
GRADO EN INGENIERÍA INFORMÁTICA  
MENCIÓN EN COMPUTACIÓN



# Variación de patrones de venación foliar en hojas de *Raphanus Raphanistrum L.* sometidas a estrés salino

**Estudiante:** Javier Rodríguez Rodríguez

**Dirección:** María Noelia Barreira Rodríguez

Jaime Fagúndez Díaz

A Coruña, diciembre de 2025.

*A mi familia, por su apoyo incondicional en cada etapa de mi vida académica.*

### **Agradecimientos**

Quiero expresar mi sincero agradecimiento a mis tutores por su atención constante y la rapidez con la que siempre respondieron a mis dudas, facilitando en gran medida el desarrollo de este trabajo. Asimismo, agradezco a mi familia y amigos por su apoyo incondicional a lo largo de este proceso.

## **Resumen**

La venación foliar constituye un rasgo morfoanatómico de gran relevancia en el estudio de la fisiología y la adaptación de las plantas. Las características estructurales de la red de nervios, como su densidad, distribución y grado de ramificación, suelen mantenerse relativamente estables dentro de una especie, aunque pueden presentar cierta plasticidad ante condiciones de estrés abiótico como la salinidad.

En este trabajo se desarrolla una herramienta basada en técnicas de visión por computador y aprendizaje profundo para el análisis automático de la venación en hojas de *Raphanus raphanistrum* L. A partir de imágenes escaneadas a tamaño real, la aplicación identifica los nervios foliares mediante segmentación semántica, extrae el esqueleto venoso y calcula métricas que caracterizan su estructura. Posteriormente, estas métricas se emplean para evaluar la variación de los patrones de venación en plantas sometidas a diferentes niveles de estrés salino.

Los resultados obtenidos muestran el potencial de la metodología propuesta para automatizar el estudio de la venación foliar y aportar información cuantitativa que contribuya a comprender los mecanismos de respuesta morfológica de las plantas frente a condiciones ambientales adversas.

## **Abstract**

Leaf venation is a morpho-anatomical trait of great relevance in plant physiology and adaptation studies. The structural characteristics of the vein network, such as its density, distribution, and branching degree, tend to remain relatively stable within a species, although they may exhibit certain plasticity under abiotic stress conditions such as salinity.

This work presents a tool based on computer vision and deep learning techniques for the automatic analysis of venation in *Raphanus raphanistrum* L. leaves. From high-resolution scanned images, the application identifies the leaf veins through semantic segmentation, extracts the venous skeleton, and computes metrics that characterize its structure. These metrics are then used to assess the variation of venation patterns in plants subjected to different levels of salt stress.

The results demonstrate the potential of the proposed methodology to automate the study of leaf venation and provide quantitative information that contributes to understanding the morphological response mechanisms of plants under adverse environmental conditions

**Palabras clave:**

- Morfología vegetal
- Análisis Cuantitativo
- Fenotipado Automatizado
- Adaptación al Estrés Abiótico
- Patrones Vasculares
- Esqueletización de Venas
- Procesamiento de Imágenes Biológicas
- Visión Artificial en Botánica
- *Raphanus Raphanistrum*
- Redes Neuronales Convolucionales

**Keywords:**

- Plant Morphology
- Quantitative analysis
- Automated Phenotyping
- Abiotic Stress Adaptation
- Vascular Patterns
- Vein Skeletonization
- Biological Image Processing
- Computer Vision in Botany
- Leaf Vein Network
- Convolutional Neural Networks (CNNs)

# Índice general

---

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Contextualización . . . . .	1
1.2	Relevancia de la especie de estudio . . . . .	1
1.3	Justificación del trabajo . . . . .	3
1.4	Objetivos . . . . .	3
1.5	Estructura de la memoria . . . . .	4
<b>2</b>	<b>Estado del arte</b>	<b>5</b>
2.1	LeafJ . . . . .	5
2.2	La segmentación de imágenes en biología . . . . .	7
2.2.1	Métodos clásicos de segmentación . . . . .	7
2.2.2	Segmentación mediante aprendizaje profundo . . . . .	7
2.3	Arquitecturas . . . . .	10
2.3.1	U-Net . . . . .	10
2.3.2	Feature Pyramid Network (FPN) . . . . .	11
2.3.3	Pyramid Scene Parsing Network (PSPNet) . . . . .	12
2.3.4	DeepLabV3 . . . . .	14
2.4	Síntesis . . . . .	15
<b>3</b>	<b>Materiales</b>	<b>16</b>
3.1	Lenguajes de programación . . . . .	16
3.2	Frameworks de aprendizaje profundo . . . . .	16
3.3	Procesamiento de imágenes . . . . .	17
3.4	Interfaz gráfica . . . . .	17
3.5	Gestión del proyecto . . . . .	17
3.6	Recursos computacionales . . . . .	17
3.7	Conjunto de datos y etiquetado . . . . .	18
3.8	Entorno de Desarrollo . . . . .	18

---

3.9	Redacción de la Memoria . . . . .	18
<b>4</b>	<b>Metodología de Desarrollo y Planificación</b>	<b>19</b>
4.1	Metodología de trabajo: SCRUM . . . . .	19
4.2	Planificación inicial . . . . .	20
4.3	Seguimiento . . . . .	21
4.4	Gestión de riesgos . . . . .	21
4.5	Estimación de costes . . . . .	23
<b>5</b>	<b>Extracción de venas</b>	<b>25</b>
5.1	Introducción . . . . .	25
5.2	Anotación y preprocesado de imágenes . . . . .	26
5.3	Materiales y métodos . . . . .	26
5.3.1	Configuración del entrenamiento . . . . .	29
5.3.2	Funciones de pérdida . . . . .	30
5.3.3	Métricas de evaluación . . . . .	31
5.4	Experimentos . . . . .	32
5.5	Resultados . . . . .	33
5.5.1	Comparación entre funciones de pérdida . . . . .	34
5.5.2	Comparación entre encoders . . . . .	34
5.5.3	Comparación entre arquitecturas . . . . .	35
5.5.4	Resultados visuales . . . . .	36
5.6	Discusión . . . . .	36
<b>6</b>	<b>Caracterización del patrón de venación</b>	<b>39</b>
6.1	Introducción . . . . .	39
6.2	Obtención del esqueleto . . . . .	39
6.3	Métricas . . . . .	40
6.4	Patrón . . . . .	42
<b>7</b>	<b>Análisis estadístico ANOVA</b>	<b>48</b>
7.1	Metodología . . . . .	48
7.1.1	Resultados del ANOVA . . . . .	49
7.2	Conclusiones . . . . .	50
<b>8</b>	<b>Herramienta para segmentación y análisis</b>	<b>52</b>
8.1	Introducción . . . . .	52
8.2	Ánalisis . . . . .	52
8.3	Diseño y arquitectura . . . . .	53

---

8.4	Interfaz de usuario . . . . .	54
8.5	Flujo de la aplicación . . . . .	56
8.6	Implementación . . . . .	57
8.6.1	Estructura de módulos . . . . .	60
8.6.2	Clase principal: SegTkApp . . . . .	60
8.6.3	Manejo de tareas en segundo plano . . . . .	60
8.6.4	Segmentación y cálculo de métricas . . . . .	61
8.6.5	Actualización de la interfaz . . . . .	62
8.6.6	Exportación de resultados . . . . .	62
8.6.7	Resumen . . . . .	62
8.7	Exportación de resultados . . . . .	62
8.8	Pruebas . . . . .	63
8.9	Problemas . . . . .	63
8.9.1	Concurrencia durante la inferencia . . . . .	64
8.9.2	Rendimiento en máquinas con recursos limitados . . . . .	64
8.9.3	Problemas menores . . . . .	64
8.10	Conclusiones . . . . .	65
<b>9</b>	<b>Conclusiones y Trabajo Futuro</b>	<b>66</b>
9.1	Conclusiones generales . . . . .	66
9.2	Conclusión final . . . . .	68
9.3	Trabajo futuro . . . . .	69
<b>A</b>	<b>Resultados de entrenamiento</b>	<b>72</b>
<b>B</b>	<b>Métricas de pétalos</b>	<b>74</b>
<b>C</b>	<b>Gráficas de Entrenamiento</b>	<b>83</b>
C.1	Experimento 1 . . . . .	83
C.2	Experimento 2 . . . . .	86
C.3	Experimento 3 . . . . .	89
	<b>Bibliografía</b>	<b>94</b>

# Índice de figuras

---

1.1	Ejemplo ilustrativo de venación foliar en una hoja . . . . .	2
1.2	Ejemplar de <i>Raphanus raphanistrum</i> . . . . .	3
2.1	Análisis de venación con LeafJ en ImageJ. . . . .	6
2.2	Arquitectura U-Net. . . . .	11
2.3	Arquitectura Feature Pyramid Network (FPN). . . . .	12
2.4	Arquitectura Pyramid Scene Parsing Network (PSPNet). . . . .	13
2.5	Arquitectura DeepLabV3. . . . .	14
4.1	Planificación inicial del TFG mediante un diagrama de Gantt (24 semanas). . . . .	21
4.2	Seguimiento del TFG mediante un diagrama de Gantt con dependencias entre tareas (34 semanas). . . . .	22
5.1	Ejemplo del proceso de anotación con CVAT.ai. . . . .	27
5.2	Ejemplos de imágenes descartadas durante la fase de anotación debido a que las venas no se podían diferenciar claramente. . . . .	28
5.3	Flujo de los experimentos de segmentación: a partir del dataset se evaluaron comparativas de funciones de pérdida, encoders y arquitecturas; los resultados se integraron en métricas y ficheros CSV. . . . .	33
5.4	Resultados de segmentación: (a) Imagen original, (b) Máscara de referencia, (c) Predicción de la U-Net. . . . .	37
5.5	Detalle de venas finas. . . . .	38
6.1	Obtención del esqueleto a partir de una imagen binaria. . . . .	40
6.2	Máscaras obtenida de aplicar umbralización. . . . .	43
6.3	Vistas representativas de la segmentación y etiquetado de aréolas. . . . .	45
6.4	Imágenes con los puntos entre bifurcaciones marcados. . . . .	46
6.5	Comparación de las distribuciones de métricas. . . . .	47

---

8.1	Diagrama de casos de uso de la herramienta de segmentación y análisis de venación. . . . .	53
8.2	Diagrama UML de la herramienta de segmentación y análisis de venación. . . . .	55
8.3	Mockup de la interfaz gráfica de la herramienta de segmentación y análisis. . . . .	56
8.4	Diagrama de secuencia del flujo de uso de la herramienta. . . . .	58
8.5	Interfaz de la herramienta al iniciarse, antes de cargar una imagen. . . . .	59
8.6	Ejemplo de la herramienta con el overlay de la imagen segmentada. . . . .	59
8.7	Diagrama de flujo de tareas en segundo plano y comunicación con la interfaz mediante colas. . . . .	61
8.8	Ejemplo de archivo generado tras procesar un lote de imágenes. . . . .	63
8.9	Ejemplo de archivo generado al exportar las métricas de una imagen individual.	63
9.1	Ejemplo de errores en el esqueleto generado a partir de una máscara. . . . .	68
9.2	Ejemplo esquemático de anastomosis. . . . .	70
C.1	Evolución de las funciones de pérdida para el Fold 1 en el Experimento 1 . . . . .	84
C.2	Evolución de las funciones de pérdida para el Fold 2 en el Experimento 1 . . . . .	85
C.3	Evolución de las funciones de pérdida para el Fold 3 en el Experimento 1 . . . . .	86
C.4	Evolución de las funciones de pérdida para el Fold 1 del Experimento 2 . . . . .	87
C.5	Evolución de las funciones de pérdida para el Fold 2 del Experimento 2 . . . . .	88
C.6	Evolución de las funciones de pérdida para el Fold 3 del Experimento 2 . . . . .	89
C.7	Evolución de las funciones de pérdida para el Fold 1 del Experimento 3 . . . . .	90
C.8	Evolución de las funciones de pérdida para el Fold 2 del Experimento 3 . . . . .	91
C.9	Evolución de las funciones de pérdida para el Fold 3 del Experimento 3 . . . . .	92

# Índice de tablas

---

2.1	Ventajas y limitaciones de los principales métodos clásicos de segmentación . . . . .	8
4.1	Estimación de costes actualizada por iteraciones y perfiles. . . . .	23
5.1	Resultados medios (media $\pm$ desviación típica) para distintas funciones de pérdida. . . . .	34
5.2	Resultados medios (media $\pm$ desviación típica) para U-Net con distintos encoders. . . . .	35
5.3	Resultados medios (media $\pm$ desviación típica) por arquitectura (encoder: resnet34). . . . .	35
6.1	Promedios de las métricas de venación obtenidas para cada grupo experimental.	43
6.2	Comparación entre las métricas obtenidas mediante segmentación automática (predicción) y a partir del <i>ground truth</i> (manual). Los valores se expresan como media $\pm$ desviación típica. . . . .	44
6.3	Diferencia media absoluta (MAD) entre las métricas obtenidas mediante segmentación automática y el <i>ground truth</i> , incluyendo desviación típica. . . . .	44
7.1	Resultados del ANOVA aplicado a las métricas obtenidas del <i>ground truth</i> . . . . .	49
A.1	Métricas de entrenamiento para arquitectura por fold. . . . .	72
A.2	Métricas de entrenamiento para encoders por fold. . . . .	72
A.3	Métricas de entrenamiento para loss por fold. . . . .	73
B.1	Métricas de venación por imagen. . . . .	74
B.2	Métricas de venación por imagen. . . . .	78

# Capítulo 1

# Introducción

---

## 1.1 Contextualización

La arquitectura vascular de los pétalos, conocida como **venación foliar**, desempeña un papel fundamental en la fisiología vegetal. A través de esta red se transportan agua, nutrientes y fotoasimilados, además de conferir soporte mecánico a los órganos vegetales. Las características del patrón de venación —como densidad, distribución, conectividad o ramificación— han sido utilizadas tradicionalmente como criterios *taxonómicos* y *filogenéticos*, permitiendo diferenciar especies y estudiar relaciones evolutivas. En los últimos años, su análisis también ha cobrado relevancia como **índicador fisiológico**, ya que modificaciones en la venación pueden reflejar respuestas adaptativas frente a distintos tipos de estrés ambiental, incluyendo sequía, temperaturas extremas o presencia de contaminantes. La figura 1.1 muestra un ejemplo de venación foliar en una planta.

Entre los distintos estreses abióticos, la **salinidad** constituye uno de los factores más limitantes para la agricultura moderna. La acumulación de sales en el suelo provoca tanto una disminución en la absorción de agua como toxicidad iónica, afectando directamente al crecimiento, la morfología foliar y la productividad de los cultivos. Estudios recientes han mostrado que la salinidad puede inducir cambios en la densidad de venas, la ramificación y la conectividad de la red vascular, afectando así la eficiencia en el transporte de agua y nutrientes. Comprender cómo varía la red vascular bajo estrés salino es, por tanto, de gran relevancia para el desarrollo de estrategias de mejora genética y de manejo agronómico que incrementen la resiliencia de las especies vegetales frente a condiciones adversas.

## 1.2 Relevancia de la especie de estudio

La especie *Raphanus raphanistrum* L. (rábano silvestre), perteneciente a la familia *Brassicaceae*, constituye un modelo de gran interés para este tipo de estudios. Se trata de una planta



Figura 1.1: Ejemplo ilustrativo de venación foliar en una hoja.

anual que crece de forma espontánea en campos cultivados y márgenes de carreteras, siendo considerada en muchos contextos como una mala hierba. Sin embargo, esta condición le ha permitido desarrollar una notable capacidad de adaptación a entornos adversos, incluyendo suelos pobres, períodos de sequía y condiciones salinas. En la Figura 1.2 se muestra una imagen del ejemplar mencionado.

Desde el punto de vista genético, *R. raphanistrum* está estrechamente emparentada con el rábano cultivado (*Raphanus sativus*), lo que la convierte en una especie **modelo para el estudio de caracteres agronómicamente relevantes** en cultivos de interés económico. Analizar cómo varían sus patrones de venación en condiciones de estrés salino puede aportar información extrapolable a especies cultivadas, especialmente dentro de la misma familia (*Brassicaceae*), que incluye también cultivos como la colza, la col o el brócoli.

Además, el rábano silvestre es una especie ampliamente utilizada en **estudios ecológicos y evolutivos**, ya que combina un ciclo de vida corto con una alta plasticidad fenotípica. Estas características permiten observar cambios morfológicos y fisiológicos en respuesta a distintos tratamientos de manera relativamente rápida, lo cual es especialmente útil en experimentos de estrés controlados en laboratorio o invernadero.

En este contexto, el estudio de la venación foliar en *Raphanus raphanistrum* bajo distintos niveles de salinidad no solo contribuye a comprender sus mecanismos de adaptación, sino que también genera conocimiento potencialmente transferible a programas de **mejora genética y manejo agronómico**. Así, la elección de esta especie proporciona un equilibrio adecuado entre **relevancia científica, aplicabilidad práctica y viabilidad experimental**.



Figura 1.2: Ejemplar de *Raphanus raphanistrum*.

### 1.3 Justificación del trabajo

El análisis tradicional de la venación se ha basado en métodos manuales (microscopía, dibujo botánico, cuantificación visual), los cuales son laboriosos, subjetivos y de difícil escalabilidad. Existen herramientas como *ImageJ* y su plugin *LeafJ*, que permiten un análisis semi-automatizado, pero presentan limitaciones importantes: sensibilidad al ruido de la imagen, necesidad de intervención manual y dificultades para adaptarse a venación muy fina o de bajo contraste.

Estas limitaciones justifican la necesidad de explorar soluciones basadas en **visión por computador y aprendizaje profundo**, capaces de extraer automáticamente las venas, generar esqueletos vasculares y calcular métricas cuantitativas que caractericen la estructura foliar.

### 1.4 Objetivos

El presente Trabajo de Fin de Grado tiene como finalidad el desarrollo de una herramienta informática que permita analizar de forma automática la venación foliar en pétalos de *Raphanus raphanistrum* sometidas a distintos niveles de estrés salino. La propuesta combina metodologías de procesado de imagen, redes neuronales profundas y extracción de métricas estructurales, con el objetivo de caracterizar la red de nervios foliares y estudiar su variación en función del tratamiento aplicado.

En primer lugar, se plantea un *objetivo general*, que constituye la meta principal del proyecto: diseñar e implementar una herramienta funcional que integre modelos de segmentación semántica y algoritmos de procesado de imagen para el estudio automatizado de la venación foliar.

A partir de este objetivo general, se definen una serie de *objetivos específicos* que orientan

el desarrollo y evaluación de la propuesta:

- Entrenar y evaluar modelos de segmentación semántica basados en aprendizaje profundo, optimizados para la identificación de la red de nervios en imágenes digitales.
- Integrar los modelos seleccionados en una herramienta capaz de extraer automáticamente el esqueleto venoso de cada pétalo.
- Definir e implementar métricas que permitan cuantificar y caracterizar la arquitectura de la venación de forma objetiva y reproducible.
- Aplicar la herramienta desarrollada a un conjunto de imágenes de pétalos sometidos a diferentes niveles de estrés salino (control, 100 y 200 mM de NaCl).
- Analizar estadísticamente la variación de los patrones de venación entre tratamientos, con el fin de explorar la plasticidad morfológica de la especie frente a la salinidad.

## 1.5 Estructura de la memoria

La memoria se organiza en varios capítulos que reflejan las fases principales del trabajo. En el Capítulo 2 se presentan los fundamentos teóricos relacionados con el procesado de imagen, la segmentación y las arquitecturas de redes neuronales profundas empleadas. El Capítulo 3 describe las tecnologías y materiales utilizados, incluyendo conjuntos de datos, librerías y recursos de cómputo. El Capítulo 4 detalla la planificación temporal y el seguimiento del desarrollo del proyecto. El Capítulo 5 se centra en la extracción de venas mediante redes neuronales, detallando la metodología de entrenamiento y los resultados obtenidos. El Capítulo 6 aborda la caracterización del patrón de venación a partir del esqueleto extraído e incluye el análisis comparativo entre tratamientos. El Capítulo 7 presenta el estudio estadístico realizado mediante ANOVA para evaluar diferencias significativas entre los tratamientos y correlacionarlas con las métricas de venación. En el Capítulo 8 se presenta la herramienta final desarrollada, desde su diseño hasta la implementación y validación. Finalmente, el Capítulo 9 recoge las principales conclusiones y propone posibles líneas de trabajo futuro.

## Capítulo 2

# Estado del arte

---

El análisis automatizado de la venación foliar requiere técnicas de **procesamiento y segmentación de imágenes**. En este capítulo se revisan los métodos más relevantes desarrollados en la literatura, desde enfoques clásicos hasta técnicas modernas de aprendizaje profundo. El objetivo es contextualizar la elección metodológica adoptada en este Trabajo Fin de Grado.

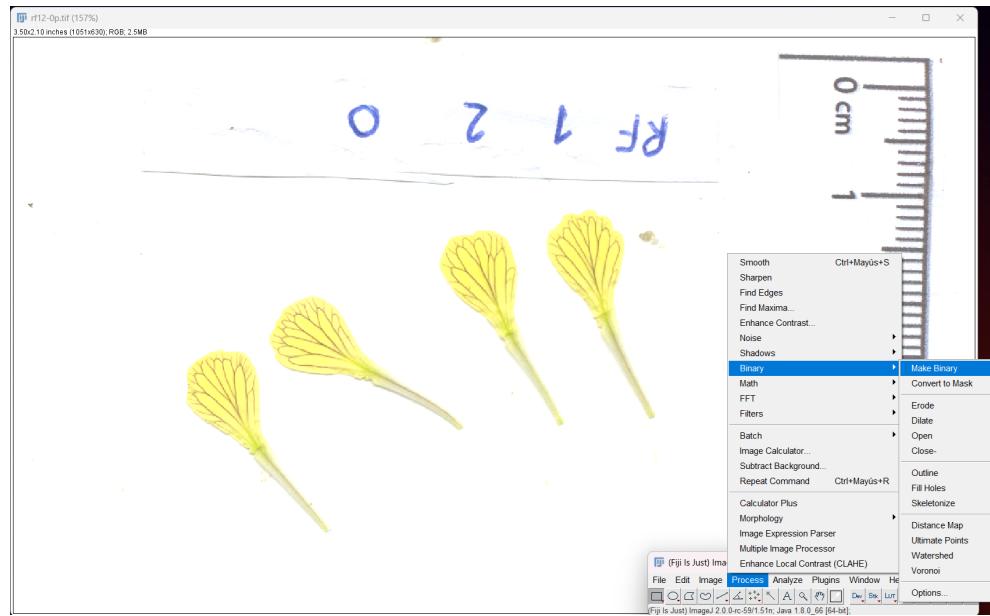
### 2.1 LeafJ

En el ámbito botánico, una de las herramientas más utilizadas ha sido el **plugin LeafJ** de *ImageJ* [1]. Este permite extraer métricas básicas de venación a partir de imágenes digitales. Se puede ver un ejemplo de uso en la Figura 2.1.

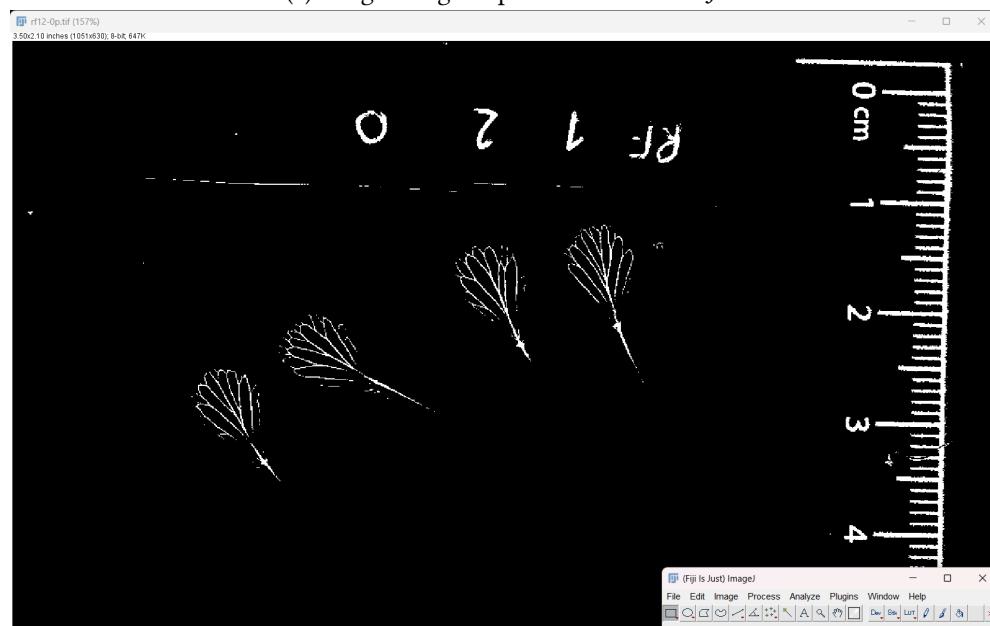
Entre sus funcionalidades se encuentra la posibilidad de calcular parámetros como la densidad de venas, la longitud total del sistema vascular, el número de cruces y ramificaciones, así como métricas relacionadas con la forma del pétalo y su nervadura principal. También permite generar esqueletizaciones a partir de la imagen de entrada, lo que facilita el conteo de segmentos y la caracterización de patrones de ramificación. Estas prestaciones han hecho de LeafJ una herramienta de referencia en estudios de morfometría foliar, particularmente útil para analizar pétalos con una arquitectura vascular bien definida.

No obstante, LeafJ presenta varias limitaciones significativas. En primer lugar, requiere imágenes con alto contraste y poco ruido para funcionar adecuadamente, lo que restringe su aplicabilidad en condiciones reales con material vegetal diverso. Además, precisa de intervención manual en distintas etapas del proceso de análisis, introduciendo subjetividad y aumentando considerablemente el tiempo de procesamiento. Finalmente, no se adapta bien al estudio de venación muy fina, particularmente en estructuras delicadas como los pétalos de *Raphanus raphanistrum*, donde los detalles vasculares son especialmente tenues.

Estas restricciones hacen que no sea adecuado para el problema planteado en este TFG, justificando el desarrollo de un nuevo método basado en técnicas más robustas.



(a) Imagen original procesada con LeafJ.



(b) Máscara binaria generada por LeafJ.

Figura 2.1: Análisis de venación con LeafJ en ImageJ.

## 2.2 La segmentación de imágenes en biología

La **segmentación de imágenes** es el proceso mediante el cual una imagen digital se divide en regiones homogéneas que representan estructuras de interés. En términos generales, consiste en asignar a cada píxel una etiqueta que lo clasifique como perteneciente a un objeto o al fondo. En el caso de la venación foliar, el problema se plantea como una **segmentación binaria**: distinguir qué píxeles corresponden a venas y cuáles al resto del pétalo o al fondo de la imagen. Este proceso constituye la base de cualquier análisis cuantitativo posterior, ya que de la calidad de la segmentación depende la fiabilidad de métricas como densidad, longitud de venas o número de ramificaciones.

### 2.2.1 Métodos clásicos de segmentación

Antes del auge del aprendizaje profundo, la segmentación de venas se abordaba mediante algoritmos clásicos de procesado de imagen. Entre los más utilizados en biología se encuentran:

- **Umbralización:** técnicas globales como el método de Otsu [2] o umbrales adaptativos que separan venas del fondo en función de la intensidad.
- **Detección de bordes:** operadores como Sobel o Canny [3], útiles para resaltar estructuras lineales finas.
- **Watershed:** segmentación por regiones a partir de gradientes [4], especialmente aplicada a estructuras ramificadas.
- **Clustering no supervisado:** algoritmos como *k-means*, que agrupan píxeles según su intensidad o color.

Estos métodos han sido empleados en estudios botánicos y biomédicos, pero presentan limitaciones importantes: son sensibles al ruido, dependen de un buen preprocesamiento y requieren ajuste manual de parámetros para cada caso. En la Tabla 2.1 se resumen las ventajas y limitaciones de estos métodos.

### 2.2.2 Segmentación mediante aprendizaje profundo

En la última década, el **aprendizaje profundo** ha transformado la segmentación de imágenes en disciplinas como la biomedicina, la ecología y la botánica. A diferencia de los métodos clásicos basados en filtrado, umbralización o segmentación por contornos, los modelos de *deep learning* aprenden representaciones jerárquicas directamente de los datos anotados. Esto les permite adaptarse tanto a la variabilidad morfológica de los objetos de estudio (por ejemplo,

Tabla 2.1: Ventajas y limitaciones de los principales métodos clásicos de segmentación.

Método	Ventajas	Limitaciones
<i>Umbralización</i>	Simple, rápido, no requiere entrenamiento	Muy sensible a iluminación y ruido; poco adaptable a variaciones
<i>Detección de bordes</i>	Resalta estructuras finas	Puede producir bordes fragmentados; depende de parámetros
<i>Watershed</i>	Buena segmentación de regiones conectadas	Tiende a la sobresegmentación; requiere filtrado previo
<i>Clustering (k-means)</i>	No supervisado; separa regiones por intensidad	No captura estructuras complejas; resultados inconsistentes

venas de diferente grosor o complejidad) como a las condiciones variables de adquisición de las imágenes (iluminación, ruido, resolución, etc.) [5, 6].

En un esquema típico de segmentación semántica, la entrada de la red neuronal es una imagen en escala de grises o en color (RGB) de tamaño fijo, mientras que la salida es un mapa de segmentación que clasifica cada píxel de la imagen en una de las clases posibles. En este trabajo, las entradas son imágenes de pétalos y las salidas son máscaras binarias donde los píxeles se etiquetan como *vena* (valor 1) o *fondo* (valor 0). Este enfoque se conoce como **segmentación densa por píxel**, ya que cada píxel de la imagen original recibe una predicción.

El entrenamiento de una red de segmentación supervisada se basa en disponer de un conjunto de imágenes anotadas manualmente. Durante el proceso de entrenamiento:

1. La imagen de entrada se pasa a través de la red, que genera una máscara de salida.
2. Esta predicción se compara con la máscara real (*ground truth*) mediante una **función de pérdida**, que cuantifica el error. Entre las más utilizadas en segmentación binaria destacan la *Binary Cross Entropy*, el *Dice Loss* y la *Focal Loss* [7, 8, 9]. Además, es posible combinar diferentes funciones de pérdida con el fin de aprovechar las ventajas de cada una y mejorar el rendimiento del modelo.
3. A partir de la pérdida, se calculan los gradientes mediante retropropagación y se actualizan los pesos de la red con un **optimizador**, normalmente variantes del método de gradiente descendiente, como Adam o SGD.

En la práctica, el aprendizaje profundo requiere un gran número de ejemplos de entrenamiento, lo que hace imposible procesar todo el conjunto de datos a la vez debido a las limitaciones de memoria de la GPU. Por este motivo, el conjunto de entrenamiento se divide en **lotes** o *batches*, que son subconjuntos de imágenes que se procesan en cada paso de entrenamiento. De esta forma, el modelo actualiza sus parámetros de manera progresiva, reduciendo el consumo de memoria y mejorando la estabilidad del aprendizaje.

El número de **iteraciones** se define como la cantidad de actualizaciones de los parámetros de la red que se realizan después de procesar un batch. Cuando la red ha visto y procesado todos los batches que componen el conjunto de entrenamiento completo, se dice que ha completado una **época** (*epoch*). Normalmente, el entrenamiento se lleva a cabo durante múltiples épocas, hasta que el modelo converge y es capaz de generalizar correctamente a imágenes no vistas.

Además, para evaluar el rendimiento del modelo de forma objetiva es necesario dividir el conjunto de datos en varios subconjuntos: el conjunto de **entrenamiento** (train), utilizado para ajustar los parámetros de la red; el conjunto de **validación** (val), que permite monitorizar el desempeño durante el entrenamiento y ajustar hiperparámetros; y finalmente el conjunto de **prueba** (test), reservado para medir la capacidad de generalización en imágenes completamente nuevas. Una división típica en problemas de segmentación consiste en emplear aproximadamente un 70–80% de los datos para entrenamiento, un 10–15% para validación y el 10–15% restante para prueba, aunque estas proporciones pueden variar en función del tamaño del conjunto de datos y de la estrategia experimental adoptada.

Con el fin de evitar el sobreajuste, se emplea un conjunto de validación independiente que permite monitorizar el rendimiento del modelo en datos distintos al entrenamiento. Las métricas más utilizadas en segmentación binaria son el **coeficiente de Dice**, el **Intersección sobre Unión (IoU)** y la **precisión pixel a pixel** [10]. Estas métricas permiten evaluar tanto la exactitud global como la capacidad del modelo para detectar correctamente estructuras delgadas y ramificadas, como las venas.

Una vez entrenado, el modelo puede aplicarse a imágenes nuevas, esto es, se puede realizar *inferencia*, sin necesidad de intervención manual. La red procesa la imagen de entrada y produce una máscara binaria en la que los objetos a segmentar aparecen resaltados. Este proceso se ejecuta en segundos, lo que lo convierte en una herramienta eficiente para el análisis de grandes volúmenes de datos.

El uso de aprendizaje profundo en segmentación ofrece varias ventajas frente a los métodos tradicionales:

- Permite **automatizar** un proceso que, de forma manual, sería costoso y lento.
- Capta **patrones morfológicos complejos** imposibles de modelar con reglas fijas.

- Se adapta a diferentes especies y condiciones experimentales siempre que exista un conjunto suficiente de datos anotados.

De este modo, el aprendizaje profundo constituye una herramienta clave para la extracción precisa de venas en imágenes de pétalos y abre la puerta a un análisis cuantitativo robusto.

## 2.3 Arquitecturas

Diversas arquitecturas de redes convolucionales han sido aplicadas con éxito en la segmentación de imágenes biomédicas y naturales. En las siguientes subsecciones se describen en detalle las más relevantes para este TFG, junto con un esquema representativo de cada una.

### 2.3.1 U-Net

Presentada en 2015, U-Net fue diseñada específicamente para la segmentación biomédica [11]. Su estructura simétrica en forma de “U” combina un camino de contracción (*encoder*<sup>1</sup>), encargado de capturar el contexto global de la imagen mediante operaciones de convolución y pooling, con un camino de expansión (*decoder*<sup>2</sup>), que utiliza convoluciones transpuestas para recuperar la resolución espacial original. Un aspecto clave de esta arquitectura son las conexiones *skip*<sup>3</sup>, que transfieren información de las capas del encoder al decoder, preservando detalles de bajo nivel esenciales para estructuras delgadas y ramificadas como las venas foliares. Cada bloque de convolución en el encoder suele estar formado por dos capas de convolución 3x3 seguidas de una activación ReLU y una operación de *max pooling*<sup>4</sup>, mientras que el decoder realiza un proceso inverso, combinando *upsampling*<sup>5</sup> con convoluciones para refinar la segmentación. La Figur 2.2 muestra el esquema de la arquitectura U-Net.

Otra ventaja importante de U-Net es su capacidad para entrenarse con conjuntos de datos reducidos. Esto se logra mediante el uso intensivo de *data augmentation*, que genera variaciones sintéticas de las imágenes originales, y la preservación de información a través de las conexiones *skip*, que mejora la propagación de gradiente durante el entrenamiento. Sin embargo, U-Net también tiene limitaciones: el número de parámetros crece rápidamente con la profundidad de la red, lo que puede aumentar el tiempo de entrenamiento y requerir más memoria, especialmente al trabajar con imágenes de alta resolución.

En el contexto del análisis de venación, U-Net puede ser especialmente eficaz para segmentar estructuras finas y ramificadas, debido a su habilidad para combinar información global y

<sup>1</sup> *Encoder*: módulo de la red que extrae características y reduce la resolución de la imagen.

<sup>2</sup> *Decoder*: módulo encargado de reconstruir la imagen segmentada recuperando la resolución espacial.

<sup>3</sup> *Skip connections*: conexiones directas que transfieren características entre capas simétricas del encoder y del decoder.

<sup>4</sup> *Max pooling*: operación que reduce la resolución de la imagen reteniendo el valor máximo en cada región.

<sup>5</sup> *Upsampling*: operación que aumenta la resolución de la imagen interpolando o replicando valores.

local de manera eficiente.

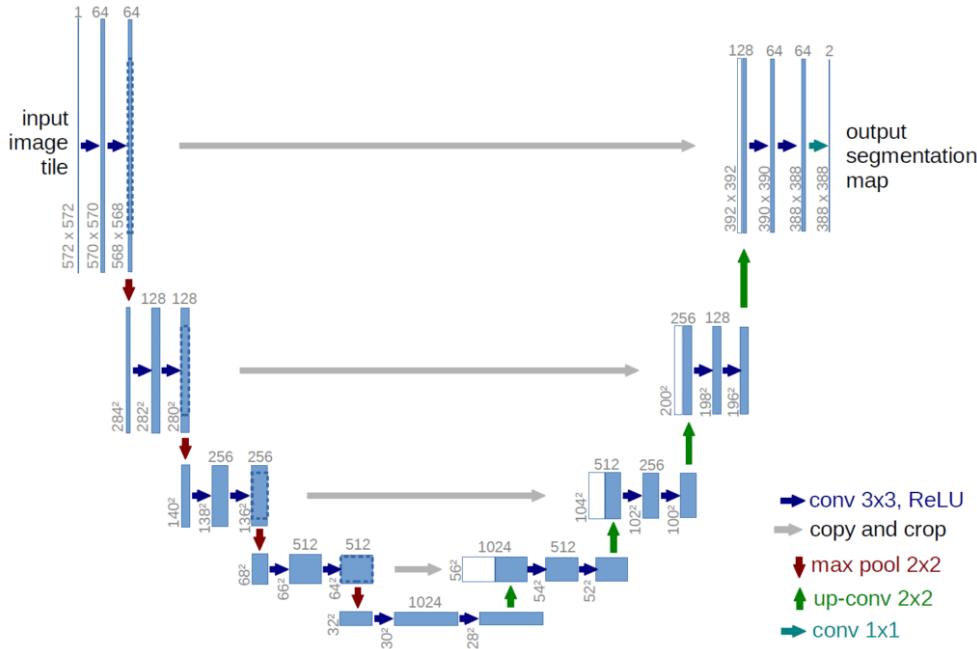


Figura 2.2: Arquitectura U-Net.

### 2.3.2 Feature Pyramid Network (FPN)

Las Feature Pyramid Networks (FPN) [12] introducen un mecanismo de pirámide de características multi-escala que mejora significativamente la capacidad de segmentación en imágenes con estructuras de diferentes tamaños. El núcleo de la FPN consiste en un *backbone* de red convolucional clásica (por ejemplo, ResNet), que extrae mapas de activación a distintos niveles jerárquicos. Estos mapas se combinan mediante una topología en la que los mapas de características de alta semántica pero baja resolución se transmiten hacia arriba y se fusionan con mapas de menor semántica pero mayor resolución. Esta fusión se realiza mediante convoluciones y operaciones de adición, generando representaciones enriquecidas en varios niveles de la pirámide. La Figura 2.3 muestra el esquema representativo de la arquitectura FPN.

Esta combinación permite que la red mantenga información semántica profunda (proporcionada por capas profundas) y detalles espaciales precisos (proporcionados por capas superficiales), lo que resulta crucial para detectar tanto venas principales como ramificaciones secundarias en los pétalos. FPN es especialmente eficiente en términos computacionales, ya que reutiliza características intermedias sin necesidad de duplicar cálculos, y se integra fácilmente con cabezales de detección o segmentación. No obstante, FPN puede ser más compleja de entrenar que U-Net y depende en gran medida de un *backbone* preentrenado para alcan-

zar su máximo rendimiento, lo que puede limitar su flexibilidad en conjuntos de datos muy específicos o pequeños.

Para el análisis de venación, la capacidad de FPN de combinar información multi-escala podría permitir segmentar con precisión venas de distintos grosores, preservando la conectividad y los detalles finos.

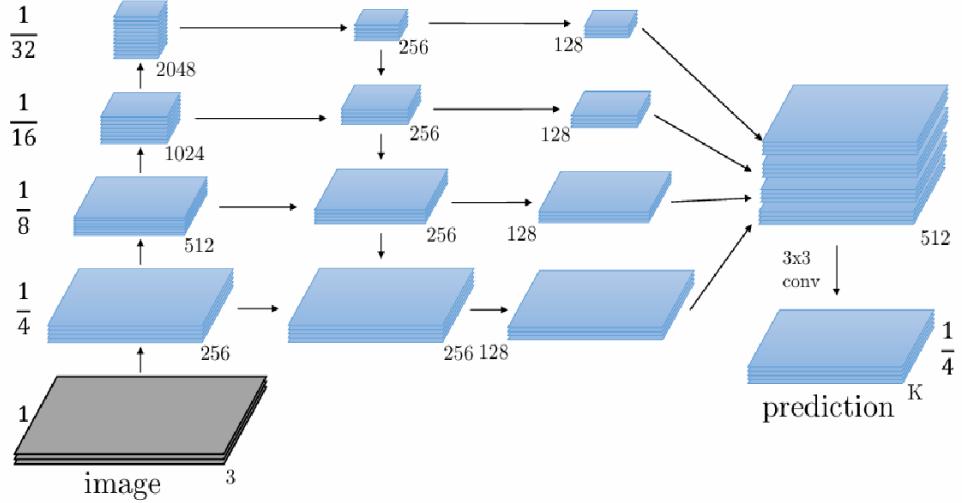


Figura 2.3: Arquitectura Feature Pyramid Network (FPN).

### 2.3.3 Pyramid Scene Parsing Network (PSPNet)

PSPNet [13] incorpora un módulo de *pyramid pooling* que extrae y combina características a distintas escalas de la imagen, capturando dependencias de contexto global y local. La red primero procesa la imagen mediante un *backbone* convolucional profundo (por ejemplo, ResNet) para generar mapas de características. A continuación, el módulo de *pyramid pooling* realiza agrupamientos (*pooling*) de distintas resoluciones, convirtiendo información espacial de diferentes escalas en vectores de características que se concatenan y refinan mediante convoluciones 1x1. La Figura 2.4 muestra el esquema de la arquitectura PSPNet.

Esta estrategia permite a PSPNet comprender cómo se relacionan las estructuras locales con el contexto global, lo que es especialmente útil cuando la disposición de la venación sigue patrones simétricos o jerárquicos. La ventaja principal de PSPNet es que mejora la segmentación en regiones ambiguas o poco definidas, donde la información local por sí sola podría inducir errores. Entre sus limitaciones se encuentra la mayor complejidad computacional y la necesidad de GPUs potentes para entrenar con imágenes de alta resolución.

En estudios de venación foliar, PSPNet podría identificar patrones de venas principales y secundarias, considerando simultáneamente la morfología global del pétalo.

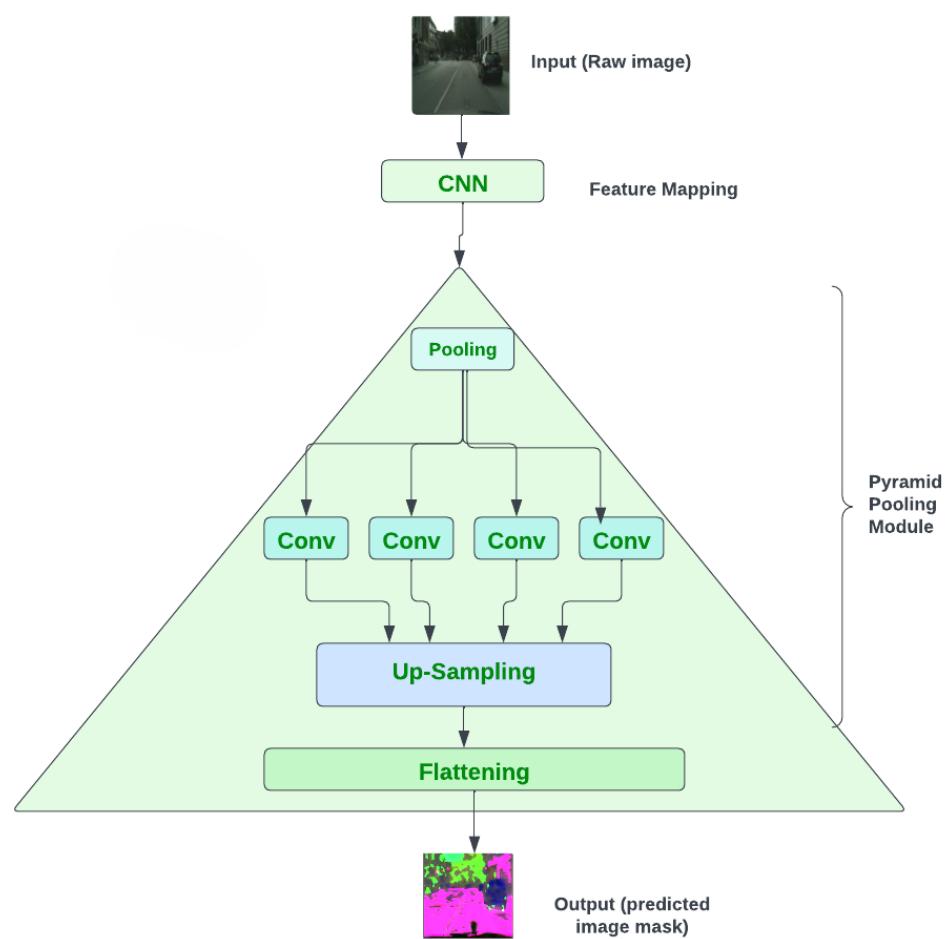


Figura 2.4: Arquitectura Pyramid Scene Parsing Network (PSPNet).

### 2.3.4 DeepLabV3

DeepLabV3 [14] introduce dos innovaciones clave: las convoluciones dilatadas (*atrous convolutions*) y el módulo ASPP (*Atrous Spatial Pyramid Pooling*). Las convoluciones dilatadas permiten ampliar el campo receptivo sin perder resolución ni aumentar el número de parámetros, lo que facilita la captura de relaciones espaciales a largo alcance. El módulo ASPP aplica múltiples convoluciones dilatadas con distintas tasas de expansión en paralelo, combinando la información de varias escalas en un solo mapa de características. La Figura 2.5 muestra un esquema de la arquitectura DeepLabV3.

Esta arquitectura podría resultar muy eficaz para segmentar estructuras finas y dispersas, como las venas secundarias en los pétalos, ya que integra contexto local y global sin sacrificar resolución. Otra ventaja de DeepLabV3 es su capacidad de adaptarse a distintos *backbones*, lo que permite balancear precisión y eficiencia computacional según los recursos disponibles. Sin embargo, su complejidad y el uso intensivo de memoria pueden dificultar su entrenamiento con imágenes muy grandes o con conjuntos de datos limitados, requiriendo estrategias de *patching* o reducción de resolución.

En el análisis de venación, DeepLabV3 podría destacar por su habilidad para segmentar estructuras de distintos grosores y posiciones, preservando la continuidad de las venas incluso en regiones con baja visibilidad.

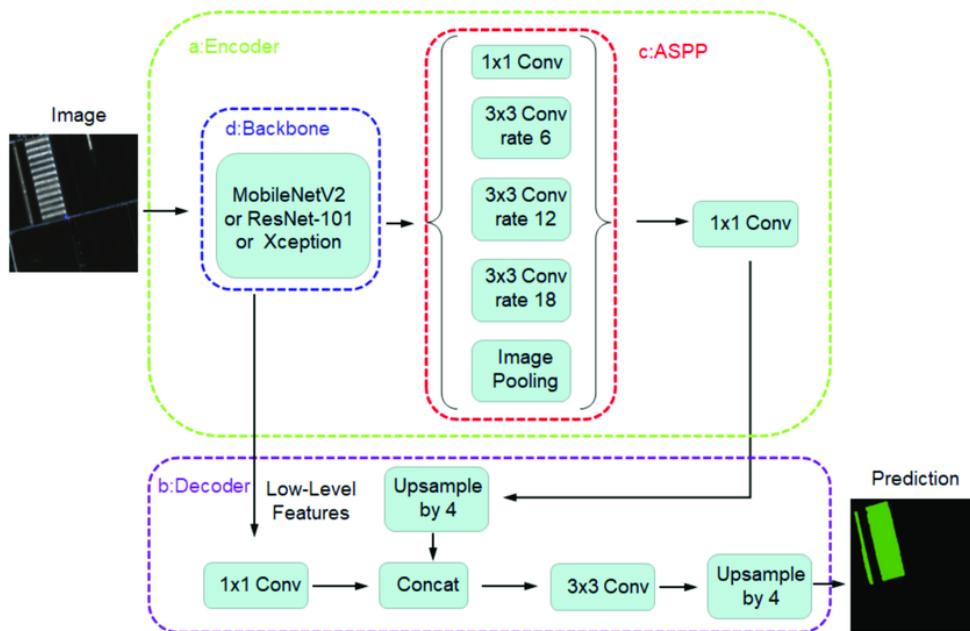


Figura 2.5: Arquitectura DeepLabV3.

## 2.4 Síntesis

Los métodos clásicos han sido útiles en entornos controlados, pero presentan limitaciones críticas para el análisis de venación en pétalos con venas finas y bajo contraste. Herramientas como LeafJ son insuficientes para este caso. Las arquitecturas modernas de aprendizaje profundo, en particular U-Net y sus variantes, ofrecen una solución más robusta y escalable, lo que motiva su aplicación en este TFG.

# Capítulo 3

# Materiales

---

En este capítulo se describirán los materiales necesarios para el desarrollo del Trabajo de Fin de Grado. Se detallarán los lenguajes de programación y librerías empleados, los frameworks de aprendizaje profundo, las herramientas utilizadas para el procesamiento de imágenes y la interfaz de usuario, así como la gestión del proyecto y los recursos de cómputo disponibles. Finalmente, se incluirá la descripción del conjunto de datos y del proceso de etiquetado.

## 3.1 Lenguajes de programación

El desarrollo se realizará en Python, utilizando la versión 3.11 [15], dado que constituye el estándar de facto en investigación en visión por computador y aprendizaje profundo. Su amplia comunidad y ecosistema de librerías lo convierten en la opción más adecuada para implementar modelos de segmentación y métricas asociadas.

En comparación con otros lenguajes como C++ o Java, Python permitirá una implementación más ágil y reproducible, además de una integración directa con librerías de procesamiento de imágenes y frameworks de aprendizaje profundo (*deep learning*).

## 3.2 Frameworks de aprendizaje profundo

Para el desarrollo de los modelos se empleará la versión **2.7.0+cu126** de **PyTorch** [16]., que ofrece flexibilidad y soporte para entrenamiento en GPU. Adicionalmente, se hará uso de la librería **Segmentation Models PyTorch (SMP)** [17], que proporciona implementaciones optimizadas de arquitecturas de referencia como U-Net, FPN, PSPNet y DeepLabV3.

El empleo de estas herramientas permitirá centrarse en la experimentación y análisis de resultados, sin necesidad de implementar las arquitecturas desde cero.

### 3.3 Procesamiento de imágenes

El preprocessamiento y análisis de las imágenes se llevará a cabo mediante diversas librerías de Python, en sus versiones más recientes estables:

- **scikit-image (versión 0.25.2)**: para operaciones de esqueletización, etiquetado de regiones y medidas morfológicas [18].
- **OpenCV (versión 4.11.0)**: para transformaciones, filtrados y operaciones de bajo nivel [19].
- **NumPy (versión 2.2.6)**: para la representación matricial y el cálculo numérico eficiente [20].

### 3.4 Interfaz gráfica

Se desarrollará una interfaz gráfica de usuario mediante la librería estándar **Tkinter 8.6** [21]. Esta permitirá a los usuarios interactuar con los modelos entrenados sin necesidad de conocimientos técnicos, incluyendo la carga de imágenes, la ejecución de segmentaciones, la visualización de resultados y la exportación de métricas.

Tkinter será escogida por su carácter multiplataforma, facilidad de integración en Python y ausencia de dependencias externas adicionales.

### 3.5 Gestión del proyecto

La gestión del código se llevará a cabo mediante **Git**, almacenando los cambios en un repositorio privado en **GitHub**. De esta forma, se podrá garantizar un control de versiones riguroso y mantener un historial detallado de la evolución del proyecto.

### 3.6 Recursos computacionales

El entrenamiento de los modelos se llevará a cabo en un equipo de altas prestaciones con las siguientes características:

- **GPU**: NVIDIA Tesla P100-PCIE-16GB.
- **CPU**: Intel(R) Xeon(R) CPU E5-2650 v4, con 48 núcleos.
- **Memoria RAM**: 500 GB.
- **Sistema operativo**: Linux 12.

Estas especificaciones garantizarán que los experimentos de segmentación puedan entrenarse en tiempos razonables, facilitando la comparación de múltiples arquitecturas y configuraciones.

### 3.7 Conjunto de datos y etiquetado

El conjunto de datos estará formado por imágenes digitales de pétalos de *Raphanus raphanistrum* escaneadas a alta resolución.

El proceso de anotación se llevará a cabo con la herramienta **CVAT.ai** [22], una plataforma especializada en el etiquetado de imágenes para visión por computador. Mediante esta herramienta se generarán máscaras binarias en las que las venas estarán representadas como la clase positiva y el resto del pétalo como fondo.

La utilización de CVAT.ai permitirá una mayor eficiencia y consistencia en la creación de las máscaras, además de facilitar la gestión de múltiples anotadores si fuese necesario.

### 3.8 Entorno de Desarrollo

Para el desarrollo de la aplicación y los experimentos se utilizó el **entorno de desarrollo integrado (IDE) Visual Studio Code (VS Code)**. Este IDE, desarrollado por Microsoft, ofrece un editor de código altamente configurable, soporte para depuración, integración con sistemas de control de versiones como Git y una amplia variedad de extensiones que facilitan la programación en Python y otros lenguajes [23].

### 3.9 Redacción de la Memoria

La redacción de la memoria se realizó utilizando **Overleaf**, un editor colaborativo basado en la web que permite trabajar con **L<sup>A</sup>T<sub>E</sub>X** de forma simultánea y sin necesidad de instalar software adicional. **L<sup>A</sup>T<sub>E</sub>X** es un sistema de preparación de documentos que destaca por su capacidad para generar documentos de alta calidad tipográfica, especialmente útil para la inclusión de fórmulas matemáticas, figuras y referencias bibliográficas [24, 25].

## Capítulo 4

# Metodología de Desarrollo y Planificación

---

En este capítulo se presentará la planificación inicial, el seguimiento del proyecto, la gestión de riesgos, la estimación de costes y la metodología de trabajo adoptada. La planificación del trabajo se organizará en varias iteraciones, cada una con objetivos concretos y entregables intermedios. Esto facilitará tanto el seguimiento del progreso como la detección temprana de problemas.

### 4.1 Metodología de trabajo: SCRUM

El proyecto adoptará la metodología ágil **SCRUM** [26], un marco de trabajo ampliamente utilizado en el desarrollo de software y proyectos de investigación. SCRUM se basa en ciclos iterativos e incrementales llamados *sprints*, que permiten dividir el trabajo en bloques manejables, priorizar tareas y garantizar la entrega continua de valor.

En cuanto a la asignación de roles, el **Product Owner** fue asumido en parte por el tutor, quien definió los objetivos globales del proyecto, supervisó el alcance y validó los entregables parciales. El rol de **Scrum Master** se fusionó con el propio estudiante, encargado de organizar las iteraciones, resolver bloqueos técnicos y gestionar la planificación. Finalmente, el **equipo de desarrollo** estuvo formado únicamente por el estudiante, responsable de la implementación del código, el entrenamiento de redes neuronales, la preparación de *datasets* y la redacción de la memoria.

El trabajo se dividió en sprints, equivalentes a las iteraciones planificadas, con objetivos claros y entregables parciales. Dentro de cada sprint, se realizarán reuniones de planificación para establecer los objetivos a corto plazo, y revisiones al finalizar cada iteración con el fin de evaluar el progreso y ajustar el plan de acción [27]. La organización en iteraciones se alinearán con estos principios, de modo que cada fase del proyecto representará un sprint con

entregables parciales y retroalimentación inmediata.

El uso de SCRUM permitirá mantener la flexibilidad frente a imprevistos, detectar problemas tempranamente y asegurar que el producto final cumpla tanto los objetivos técnicos como los plazos establecidos.

## 4.2 Planificación inicial

La planificación inicial se diseñará para abarcar el conjunto completo del desarrollo, con una distribución temporal de tareas que asegurará la finalización de la memoria a tiempo. La Figura 4.1 mostrará el diagrama de Gantt propuesto al inicio.

El desarrollo del proyecto se organizará siguiendo un enfoque iterativo e incremental. En lugar de plantear una única fase larga de análisis, implementación y pruebas, se dividirá el trabajo en varias iteraciones consecutivas que facilitarán el seguimiento del progreso, la detección temprana de problemas y la adaptación a las necesidades que vayan surgiendo.

Cada iteración se centrará en un objetivo concreto, manteniendo cierta solapación entre ellas para garantizar un flujo de trabajo realista y flexible.

Las principales iteraciones previstas serán las siguientes:

1. **Revisión bibliográfica y diseño:** se realizará una búsqueda y estudio de literatura relacionada con la segmentación de imágenes biomédicas y botánicas, y se definirá la estrategia de trabajo.
2. **Etiquetado del dataset:** se abordará el proceso de anotación manual del conjunto de imágenes de pétalos, generando las máscaras binarias necesarias para los entrenamientos posteriores.
3. **Entrenamientos y métricas de segmentación:** se llevarán a cabo experimentos con las distintas redes, probando diferentes configuraciones y funciones de pérdida, y recopilando métricas de rendimiento.
4. **Aplicación software (análisis, desarrollo + pruebas):** se desarrollará la aplicación de escritorio con Tkinter, que permitirá cargar un modelo entrenado, realizar inferencias y obtener métricas de venación. Las pruebas se ejecutarán en paralelo al propio desarrollo.
5. **Redacción de la memoria:** se documentarán los avances de manera progresiva, reservando la última fase para la integración y revisión final.

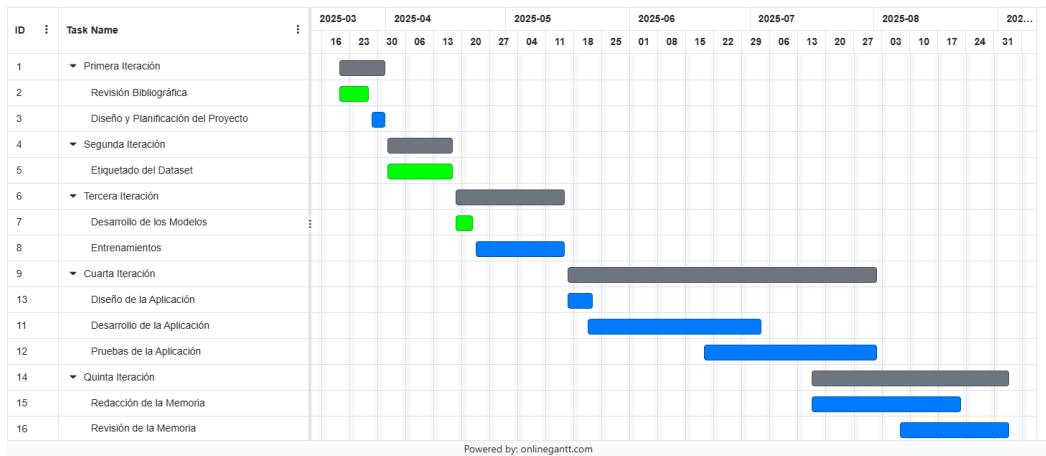


Figura 4.1: Planificación inicial del TFG mediante un diagrama de Gantt (24 semanas).

### 4.3 Seguimiento

El desarrollo del proyecto se ajustó en gran medida a la planificación inicial, aunque se produjeron modificaciones en la distribución temporal debido a factores académicos y logísticos. Inicialmente, la entrega de la memoria estaba prevista para el mes de septiembre, pero el calendario se amplió hasta noviembre con el fin de disponer de más tiempo para completar los experimentos, optimizar los modelos y realizar un análisis más exhaustivo de los resultados.

Este reajuste permitió una mejor integración entre las fases de entrenamiento, validación y desarrollo de la aplicación, aunque también supuso un mayor solapamiento entre iteraciones. En particular, la fase de entrenamiento se prolongó más de lo previsto debido a la necesidad de ajustar los hiperparámetros. Asimismo, la fase de redacción se extendió en paralelo a los últimos experimentos y a la preparación del análisis estadístico final.

La Figura 4.2 refleja esta planificación revisada, en la que se incluyen las dependencias entre tareas y la extensión temporal del proyecto hasta noviembre.

### 4.4 Gestión de riesgos

Como en todo proyecto de desarrollo e investigación, será necesario identificar desde el inicio los posibles riesgos que puedan comprometer el cumplimiento de los objetivos o los plazos previstos. La gestión de riesgos se planteará de manera proactiva, es decir, no sólo describiendo las amenazas potenciales, sino también estableciendo estrategias de mitigación adaptadas a cada caso.

Los principales riesgos detectados durante el desarrollo del TFG serán los siguientes:

- **Retrasos derivados del calendario académico.** La coincidencia del proyecto con el

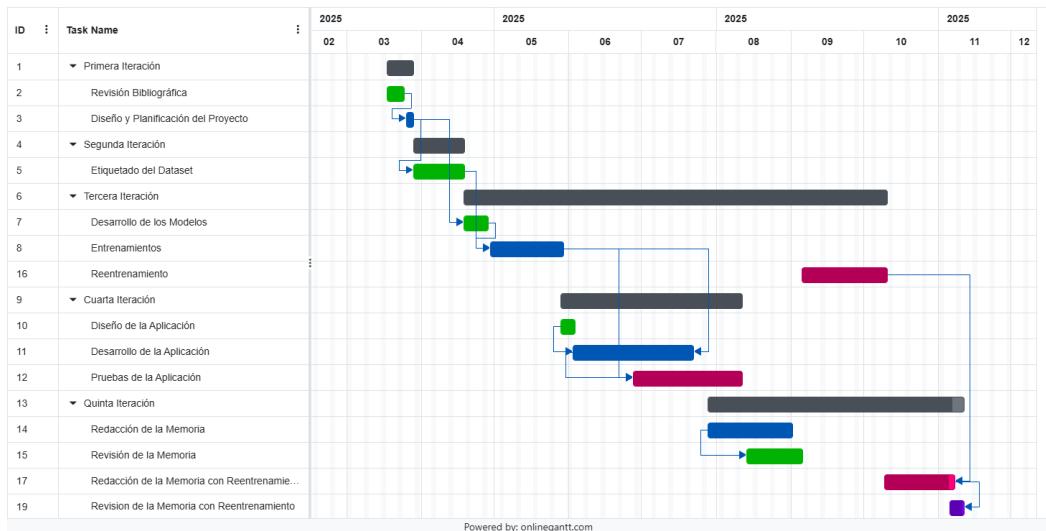


Figura 4.2: Seguimiento del TFG mediante un diagrama de Gantt con dependencias entre tareas (34 semanas).

periodo de exámenes finales suponía un riesgo claro de ralentización. Para reducir su impacto, se decidió flexibilizar la planificación temporal, destinando menos horas a las iteraciones activas durante los meses de mayor carga académica y compensando posteriormente con una mayor dedicación en agosto y la primera semana de septiembre. Gracias a esta medida, se mantuvo la coherencia global con los plazos previstos sin comprometer la calidad del trabajo.

- **Complejidad técnica de la segmentación de venas.** El problema abordado presentaba una dificultad técnica significativa: las venas en los pétalos de *Raphanus raphanistrum* son estructuras finas, de bajo contraste y con fuerte desbalance de clases respecto al fondo. Para mitigar este riesgo, se diseñó una estrategia experimental basada en la comparación sistemática de diferentes arquitecturas y funciones de pérdida, lo que permitió identificar configuraciones robustas y reducir la dependencia de una única solución.
- **Limitación de recursos computacionales.** El entrenamiento de redes de segmentación profunda puede resultar muy costoso en términos de tiempo y hardware. Inicialmente, existía el riesgo de que los recursos locales disponibles fuesen insuficientes. Este riesgo fue mitigado gracias a la disponibilidad de una máquina proporcionada por la tutora que permitió entrenar modelos de forma más eficiente y garantizar la ejecución de los experimentos en condiciones adecuadas.

En conjunto, la gestión de estos riesgos fue esencial para asegurar que el proyecto avanzase de manera continua y ordenada. La identificación temprana de problemas potenciales,

junto con la definición de medidas de contingencia realistas, permitió mantener la estabilidad del cronograma y alcanzar los resultados esperados.

## 4.5 Estimación de costes

La estimación de costes se calculará en base a dos perfiles: desarrollador novel (20€/h) y supervisor (45€/h). La Tabla 4.1 resumirá la distribución.

Tabla 4.1: Estimación de costes actualizada por iteraciones y perfiles.

<b>Iteración</b>	<b>Horas Novel</b>	<b>Coste Novel</b>	<b>Horas Supervisor</b>	<b>Coste Supervisor</b>	<b>Total</b>
<i>Revisión bibliográfica y diseño</i>	40	800€	6	270€	1.070€
<i>Etiquetado del dataset</i>	45	900€	4	180€	1.080€
<i>Entrenamientos y métricas</i>	130	2.600€	18	810€	3.410€
<i>Aplicación</i>	115	2.300€	18	810€	3.110€
<i>Redacción de la memoria</i>	70	1.400€	8	360€	1.760€
<b>Totales</b>	<b>400</b>	<b>8.000€</b>	<b>54</b>	<b>2.430€</b>	<b>10.430€</b>

En términos globales, los costes planificados y los derivados del seguimiento se mantendrán muy próximos, ya que los ajustes realizados durante la ejecución no supondrán desviaciones significativas. El coste total estimado del proyecto será de **10.430 €**, considerando tanto el esfuerzo del desarrollador novel como el de supervisión académica. De esta cantidad, **8.000 €** corresponden al trabajo del desarrollador novel (400 horas a 20 €/h), mientras que los **2.430 €** restantes provienen de la supervisión académica (54 horas a 45 €/h). La distribución por tareas refleja que las fases más costosas son los entrenamientos y métricas (3.410 €) y el desarrollo de la aplicación con sus pruebas (3.110 €), seguidas del etiquetado del dataset (1.080 €), la redacción de la memoria (1.760 €) y la revisión bibliográfica y diseño inicial (1.070 €). En conjunto, la mayor parte de la inversión se concentra en las etapas técnicas de implementación y validación, mientras que las fases de diseño y documentación representan una fracción menor del presupuesto total.

Además del esfuerzo humano, resulta imprescindible considerar el coste asociado al uso de la infraestructura de cómputo empleada en el proyecto. El hardware utilizado, corresponde a un entorno de alto rendimiento habitualmente disponible en instalaciones universitarias o centros de investigación.

El acceso a este tipo de recursos puede realizarse de dos formas principales: mediante la compra directa de equipos o mediante el uso de servicios en la nube. Una **GPU NVIDIA Tesla P100 de 16 GB** tiene un coste de mercado aproximado de entre **2.000 y 3.000 €**, mientras que un servidor con procesadores Xeon de altas prestaciones y grandes cantidades de memoria RAM puede superar fácilmente los **10.000 €**. En cambio, las plataformas de cómputo en la nube ofrecen tarifas por horas de uso, que en el caso de instancias con características equivalentes suelen situarse entre **1,5 y 3 €/hora**.

En este proyecto, el acceso al hardware se realizó a través de recursos ya disponibles en la universidad, por lo que no supuso un coste económico adicional directo. No obstante, en un contexto de transferencia tecnológica o aplicación industrial, sería necesario contemplar esta inversión en la estimación global del presupuesto, dado que el entrenamiento de modelos de segmentación profunda requiere una infraestructura computacional considerable.

## Capítulo 5

# Extracción de venas

---

### 5.1 Introducción

La extracción automática de venas en pétalos constituye un paso fundamental para el análisis cuantitativo de la venación foliar. Esta tarea permite obtener información estructural clave para evaluar posibles alteraciones derivadas de condiciones de estrés, como el estrés salino. Desde el punto de vista computacional, el problema puede abordarse como una tarea de **segmentación semántica binaria**, en la que cada píxel de la imagen debe clasificarse como parte de una vena o del fondo.

Los enfoques clásicos de procesado de imagen (umbralización adaptativa, filtros de Gabor, transformadas morfológicas, etc.) han mostrado utilidad en ciertos escenarios, pero presentan limitaciones importantes cuando la red de venación es compleja, con venas muy finas y bajo contraste con el fondo. En este contexto, los métodos de **aprendizaje profundo** y, en particular, las arquitecturas de redes convolucionales para segmentación, ofrecen un rendimiento superior al aprender de manera automática las características relevantes directamente de los datos. Esto significa que, en lugar de diseñar manualmente filtros o reglas específicas, la red es capaz de detectar primero patrones simples (como bordes o texturas) y después combinarlos progresivamente en patrones más complejos (como formas ramificadas o estructuras complejas de venación), lo que le permite adaptarse mejor a la variabilidad presente en las imágenes [28].

En este capítulo se describe el proceso seguido para preparar los datos, entrenar y evaluar distintos modelos de segmentación profunda orientados a la extracción de venas.

## 5.2 Anotación y preprocessado de imágenes

El proceso de anotación y preparación de imágenes se llevó a cabo empleando la herramienta **CVAT.ai**<sup>1</sup>, ampliamente utilizada en proyectos de visión por computador para la creación de máscaras de segmentación.

A partir de un total de **67 imágenes originales**, cada una con **cuatro pétalos**, se generaron máscaras binarias en las que se representaban únicamente las venas visibles. Durante la anotación se descartaron **6 pétalos individuales** y **3 imágenes completas**, debido a problemas de calidad —como desenfoque, iluminación deficiente o estructuras de venación poco discernibles— que impedían una segmentación fiable.

En la Figura 5.1 se muestra un ejemplo del etiquetado realizado, donde se observa una imagen original junto con su versión anotada en CVAT.

Asimismo, en la Figura 5.2 se ilustran ejemplos de imágenes descartadas debido a que no se podían diferenciar claramente las venas. En el primer caso se eliminó toda la imagen por falta de visibilidad adecuada, mientras que en el segundo únicamente se descartó un pétalo concreto con el mismo problema de claridad en la venación.

Una vez completada la anotación, las imágenes fueron sometidas a un proceso adicional de **segmentación individual de pétalos**, con el objetivo de disponer de muestras independientes y homogéneas para el entrenamiento de la red U-Net. Este procedimiento se implementó en Python, utilizando las librerías OpenCV y NumPy. Para cada imagen, se empleó su máscara asociada para detectar regiones conectadas y recortar automáticamente cada pétalo, aplicando un pequeño margen adicional (5 píxeles) para evitar pérdidas de borde. Luego, para cada máscara de pétalo, se generó una nueva imagen binaria que contenía únicamente la venación correspondiente a ese pétalo.

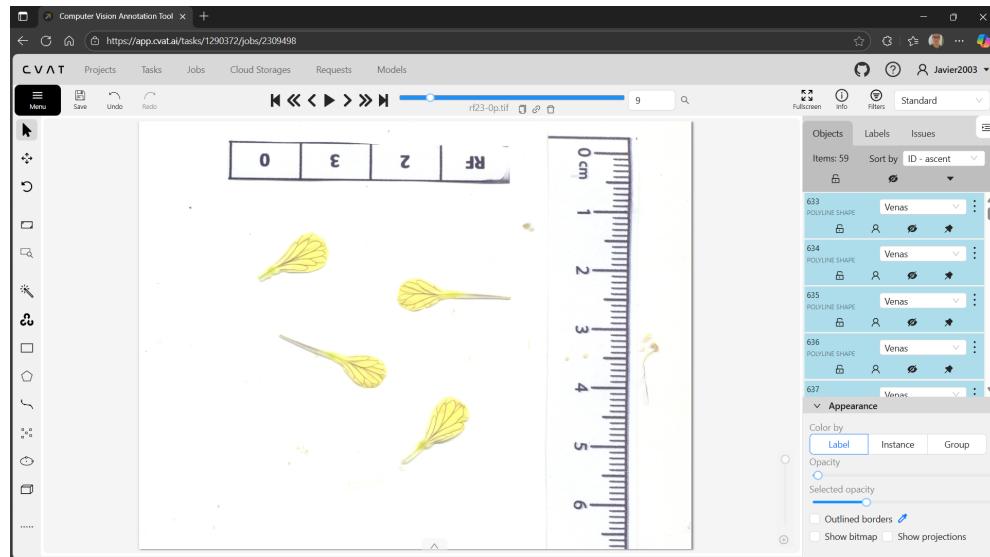
Este proceso permitió generar un **conjunto de datos de alta calidad**, optimizado para el entrenamiento de modelos de segmentación profunda en el dominio específico de la venación foliar. Tras los descartes, el conjunto final quedó compuesto por **64 imágenes completas** y **262 pétalos válidos**, manteniendo la consistencia y calidad necesarias para los experimentos.

## 5.3 Materiales y métodos

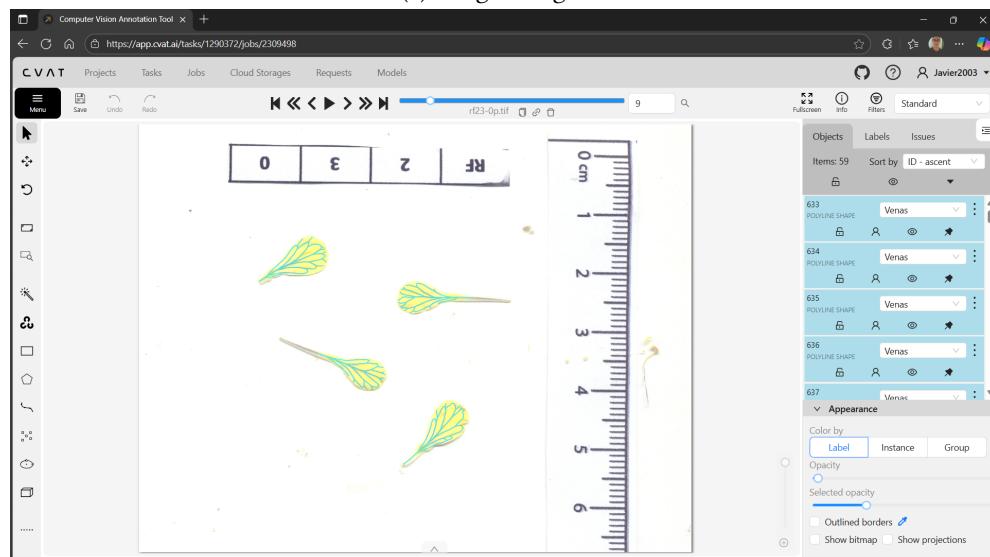
El éxito de un sistema de segmentación basado en aprendizaje profundo no depende únicamente de la arquitectura de red empleada, sino también de un conjunto de decisiones metodológicas que abarcan desde la preparación de los datos hasta la configuración de los hiperparámetros de entrenamiento. En este proyecto se adoptó un enfoque sistemático que buscó equilibrar la robustez del modelo con la viabilidad computacional, teniendo siempre en cuenta

---

<sup>1</sup><https://www.cvat.ai/>

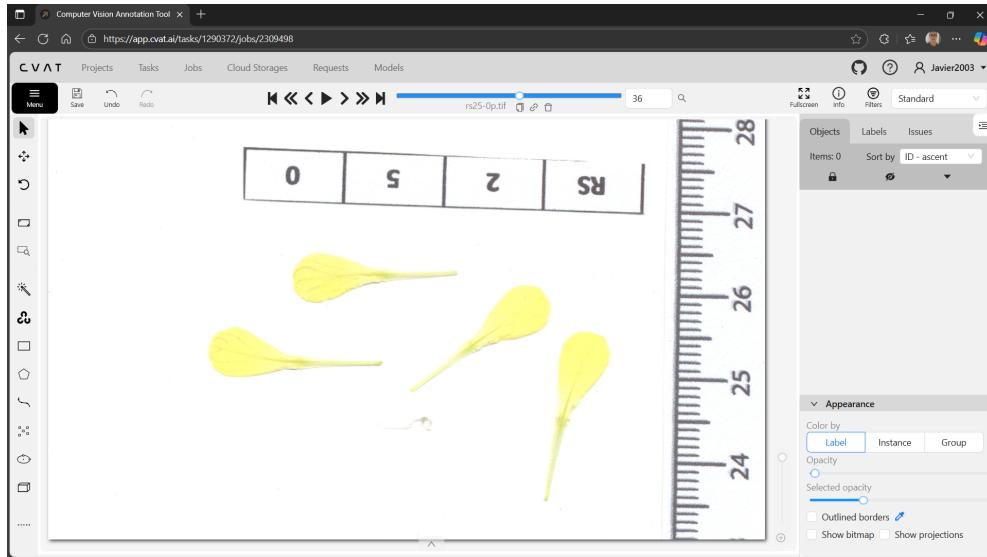


(a) Imagen original

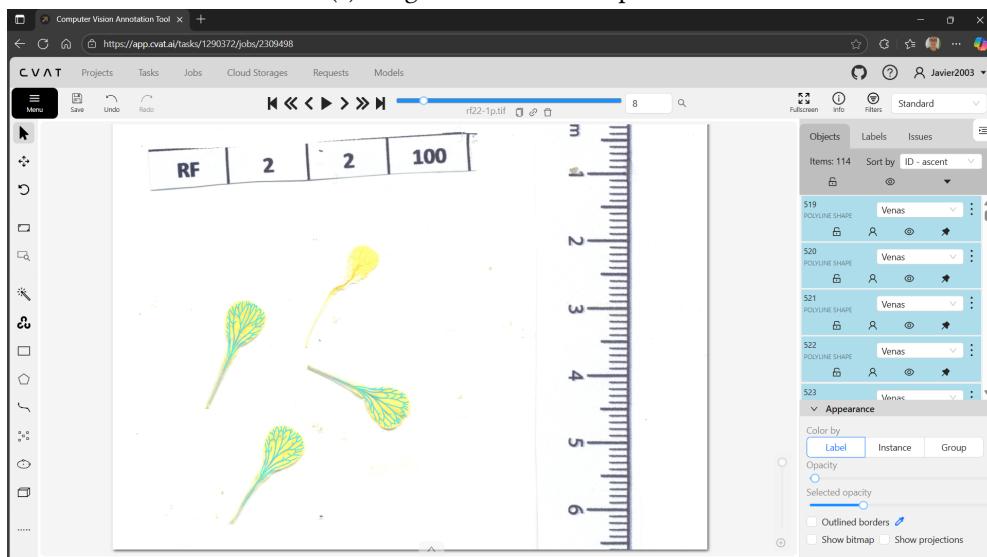


(b) Máscara anotada

Figura 5.1: Ejemplo del proceso de anotación con CVAT.ai.



(a) Imagen descartada completa



(b) Imagen con un pétalo descartado

Figura 5.2: Ejemplos de imágenes descartadas durante la fase de anotación debido a que las venas no se podían diferenciar claramente.

las particularidades del problema: la extracción de venas finas y con bajo contraste en pétalos de *Raphanus raphanistrum*.

En esta sección se describen las configuraciones de entrenamiento seleccionadas, las funciones de pérdida evaluadas y las métricas empleadas para la evaluación de los modelos. Además, se detallan las estrategias aplicadas para abordar los principales retos de la tarea, como el fuerte desbalanceo de clases o la variabilidad morfológica entre imágenes. El objetivo de esta descripción es proporcionar la trazabilidad completa del proceso experimental, de forma que tanto las elecciones como los resultados puedan reproducirse y validarse en el futuro.

### 5.3.1 Configuración del entrenamiento

Se combinó cada arquitectura con diferentes **codificadores** (*encoders*) (ResNet34, ResNet50, EfficientNet-B0 y MobileNet\_v2), seleccionados por su equilibrio entre rendimiento y coste computacional [29, 30, 31].

Los principales hiperparámetros fueron los siguientes:

- Tamaño de la imagen de entrada: **224 × 224 píxeles**.
- Tamaño de lote (*batch size*): **8**.
- Optimizador: **Adam** con tasa de aprendizaje inicial de  $2 \times 10^{-4}$  y planificador coseno con tasa mínima de  $1 \times 10^{-5}$ .
- Número máximo de 250 épocas, con **early stopping**, una técnica que detiene el entrenamiento cuando la pérdida de validación deja de mejorar tras un número determinado de épocas (pacienza = 10,  $\delta = 0.001$ ). De este modo se evita el sobreajuste y se reduce el tiempo de cómputo.
- Normalización empleando las estadísticas propias del *dataset*. A los valores de cada canal se les restó la media correspondiente y se dividieron por la desviación típica, de forma que las intensidades quedaran centradas y escaladas. Se calculó la media y desviación estándar por canal procesando todas las imágenes del conjunto de entrenamiento. Cada imagen se redimensionó a 640x640 píxeles y se sumaron los valores de cada canal (R, G, B) y sus cuadrados, obteniendo así la media y desviación estándar global por canal. Esta normalización asegura que las intensidades estén centradas y escaladas de manera consistente durante el entrenamiento de la red neuronal.
- Estrategias de **aumento de datos**. Durante el entrenamiento, se aplicaron transformaciones aleatorias a las imágenes y sus máscaras correspondientes para incrementar la diversidad del conjunto de datos. Estas transformaciones incluyeron volteos horizontales y verticales con probabilidad 0.5, rotaciones de 90 grados aleatorias, y ajustes de

brillo y contraste. La aplicación de estas variaciones mejora la generalización del modelo al simular diferentes orientaciones y condiciones de iluminación.

El conjunto de datos se dividió siguiendo una estrategia de **3-fold cross-validation**<sup>2</sup>, con particiones de 70% para entrenamiento, 15% para validación y 15% para prueba para cada fold. De este modo, cada modelo fue entrenado y evaluado en tres configuraciones distintas, y los resultados finales se calcularon como la media y la desviación típica. Para asegurar la reproducibilidad de los experimentos, se fijó la misma **semilla aleatoria**<sup>3</sup> para cada proceso de división de datos y entrenamiento.

### 5.3.2 Funciones de pérdida

La segmentación de venas plantea el reto de un fuerte **desbalanceo de clases**, dado que la proporción de píxeles pertenecientes a venas es mucho menor que la de fondo. Para abordar este problema, se evaluaron distintas funciones de pérdida, cada una con sus propiedades matemáticas.

**Binary Cross-Entropy (BCE) Loss:** Es la función estándar en tareas de clasificación binaria [32]. Se define como:

$$\mathcal{L}_{BCE} = -\frac{1}{N} \sum_{i=1}^N [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)] \quad (5.1)$$

donde  $N$  es el número total de píxeles,  $y_i \in \{0, 1\}$  es la etiqueta real del píxel  $i$ , y  $p_i \in [0, 1]$  es la probabilidad predicha de que dicho píxel pertenezca a la clase positiva (vena).

**Dice Loss:** Optimiza directamente la superposición entre predicción y referencia [7]. Su definición es:

$$\mathcal{L}_{Dice} = 1 - \frac{2 \sum_{i=1}^N y_i p_i + \epsilon}{\sum_{i=1}^N y_i + \sum_{i=1}^N p_i + \epsilon} \quad (5.2)$$

donde  $\epsilon$  es un término de suavizado para evitar divisiones por cero. Esta función es especialmente útil en contextos con fuerte desbalanceo de clases.

**Focal Loss:** Ajusta el peso de los ejemplos en función de la dificultad, resultando adecuada

<sup>2</sup> Técnica de validación donde los datos se dividen en 3 partes y se realizan 3 entrenamientos diferentes, rotando qué parte se usa para validación.

<sup>3</sup> Valor numérico que inicializa los generadores de números aleatorios, permitiendo reproducir exactamente la misma secuencia de números pseudo-aleatorios.

en contextos con clases minoritarias [9]. Se expresa como:

$$\mathcal{L}_{Focal} = -\frac{1}{N} \sum_{i=1}^N [\alpha(1-p_i)^\gamma y_i \log(p_i) + (1-\alpha)p_i^\gamma (1-y_i) \log(1-p_i)] \quad (5.3)$$

donde  $\alpha \in [0, 1]$  es un parámetro de equilibrio entre clases y  $\gamma \geq 0$  controla el grado de penalización de ejemplos fáciles.

**BCE + Dice Loss:** Aunque en general se pueden combinar distintas funciones de pérdida según la tarea, en segmentación es habitual combinar la Binary Cross-Entropy con la Dice Loss, buscando un equilibrio entre la optimización a nivel de píxel y la superposición global entre predicción y referencia [33]. Se define como:

$$\mathcal{L}_{BCE+Dice} = \mathcal{L}_{BCE} + \mathcal{L}_{Dice} \quad (5.4)$$

De esta forma, la BCE penaliza errores en cada píxel individual, mientras que la Dice Loss corrige el sesgo hacia la clase mayoritaria y asegura una mejor segmentación de regiones pequeñas y delgadas como las venas.

### 5.3.3 Métricas de evaluación

En este problema, cada píxel se clasifica como positivo (perteneciente a una vena) o negativo (fondo). Los verdaderos positivos ( $TP$ ) son aquellos píxeles correctamente identificados como venas, mientras que los falsos positivos ( $FP$ ) son píxeles predichos como venas pero que en realidad corresponden al fondo. Los falsos negativos ( $FN$ ) corresponden a píxeles de venas que el modelo predijo incorrectamente como fondo, y los verdaderos negativos ( $TN$ ) son los píxeles correctamente identificados como fondo. Esta definición permite evaluar de manera precisa la segmentación a nivel de píxel, considerando tanto la exactitud como la cobertura en la detección de venas finas y ramificadas.

La evaluación de los modelos se realizó mediante las siguientes métricas, habituales en segmentación semántica:

**IoU (Intersection over Union):** Mide la superposición entre la máscara predicha y la máscara real, calculando el cociente entre la intersección y la unión de ambas:

$$IoU = \frac{TP}{TP + FP + FN} \quad (5.5)$$

**Precisión:** Indica qué proporción de los píxeles predichos como positivos (venas) son realmente positivos:

$$Precision = \frac{TP}{TP + FP} \quad (5.6)$$

**Recall (Exhaustividad):** Refleja la capacidad del modelo para detectar correctamente todos los píxeles positivos de la máscara real:

$$Recall = \frac{TP}{TP + FN} \quad (5.7)$$

**F1-score:** Es la media armónica entre precisión y recall, equilibrando ambos aspectos de la predicción:

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \quad (5.8)$$

## 5.4 Experimentos

Se diseñaron tres bloques experimentales para evaluar de manera sistemática el impacto de las decisiones arquitectónicas y de entrenamiento, como se muestra en la Figura 5.3. Cada bloque se ejecutó bajo el esquema de 3-fold descrito anteriormente.

La evaluación de los modelos de segmentación de venas se abordó mediante tres comparativas principales: funciones de pérdida, encoders y arquitecturas. En cuanto a las funciones de pérdida, se consideró que la clase positiva, correspondiente a las venas, representa una proporción muy pequeña del total de píxeles, lo que genera un fuerte desbalanceo de clases. Por ello, se evaluaron varias funciones de pérdida con diferentes propiedades. La Binary Cross-Entropy (BCE) penaliza errores a nivel de píxel y constituye un referente estándar. La Dice Loss, por su parte, optimiza la superposición global entre predicción y referencia, resultando especialmente útil para segmentar regiones pequeñas y delgadas. La combinación BCE + Dice Loss busca un equilibrio entre la precisión por píxel y la cobertura global de la estructura de venas, estrategia habitual en segmentación biomédica. Finalmente, la Focal Loss pondera los píxeles difíciles, reduciendo la influencia de los fáciles y mejorando la detección de clases minoritarias. Esta comparativa permitió determinar qué función de pérdida ofrecía el mejor compromiso entre exactitud por píxel y capacidad de capturar venas finas y ramificadas.

Respecto a los encoders, se evaluaron modelos preentrenados en ImageNet por su capacidad de extraer características jerárquicas de manera eficiente. Los encoders ResNet34 y ResNet50 presentan bloques residuales profundos que facilitan la captura de patrones complejos de venación. EfficientNet-B0 destaca por su escalado compuesto de profundidad, ancho y resolución, optimizando la relación entre precisión y coste computacional, mientras que mobilenet\_v2 ofrece un diseño ligero y eficiente, adecuado para entrenamientos rápidos o entornos con recursos limitados. La comparación de estos encoders permitió seleccionar aquellos que balancean mejor la capacidad de representación de características con el coste de entrenamiento y la memoria requerida.

En cuanto a las arquitecturas de segmentación, se eligieron modelos ampliamente utilizados:

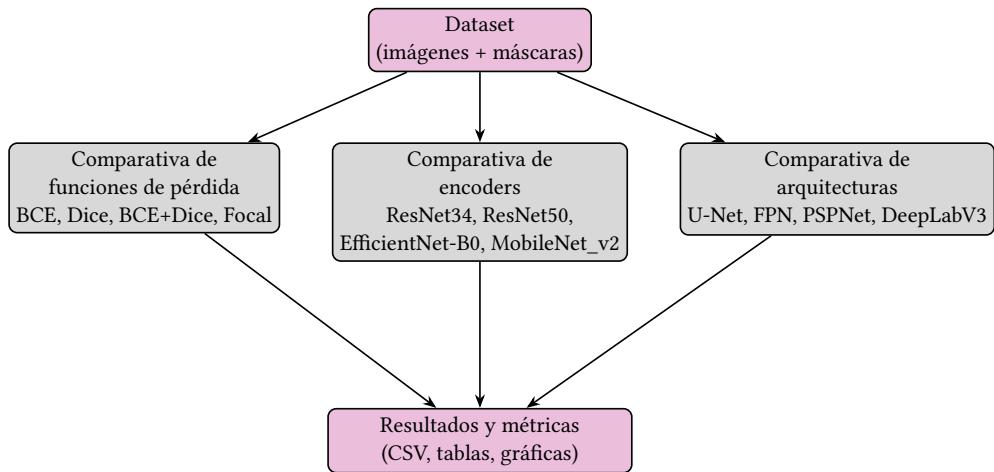


Figura 5.3: Flujo de los experimentos de segmentación: a partir del dataset se evaluaron comparativas de funciones de pérdida, encoders y arquitecturas; los resultados se integraron en métricas y ficheros CSV.

zados en imágenes biomédicas y naturales. U-Net, con sus conexiones en *skip connections*, es especialmente eficaz en la segmentación de estructuras finas y detalladas. DeepLabV3 utiliza convoluciones dilatadas y un módulo de agrupamiento espacial (*ASPP*) para capturar contexto a diferentes escalas, mientras que FPN combina características de múltiples resoluciones, mejorando la detección de detalles finos. PSPNet incorpora un módulo de *pyramid pooling* que integra contexto global, lo que resulta útil para estructuras complejas de venación. La comparativa de arquitecturas permitió evaluar cómo cada modelo maneja la segmentación de venas finas y ramificadas, considerando tanto la resolución espacial como la eficiencia computacional.

En conjunto, estas comparativas proporcionaron un marco integral para seleccionar la combinación óptima de función de pérdida, encoder y arquitectura, con el objetivo de lograr segmentaciones precisas y coherentes de las venas en los pétalos.

Para garantizar reproducibilidad y trazabilidad, se desarrolló un sistema automatizado de ejecución mediante *scripts* que almacenaban tanto los *logs* de entrenamiento como las métricas de validación y prueba en ficheros CSV.

## 5.5 Resultados

Para la evaluación de los modelos de segmentación de venas, la métrica más relevante es el **F1-score**, ya que refleja el equilibrio entre precisión y recall. Esta métrica es especialmente útil en nuestro caso porque permite evaluar tanto la capacidad del modelo para detectar correctamente las venas como para evitar falsas detecciones, lo cual es crítico dada la finura y

la complejidad de la red venosa en los pétalos.

A continuación se presentan las tablas resumen, donde se muestra la media obtenida en los tres folds así como su desviación típica, de las métricas principales para cada bloque experimental. Los resultados por fold se incluyen en el anexo A.

### 5.5.1 Comparación entre funciones de pérdida

Los resultados medios por función de pérdida se resumen en la Tabla 5.1.

Tabla 5.1: Resultados medios (media ± desviación típica) para distintas funciones de pérdida.

<b>Función de pérdida</b>	<b>IoU</b>	<b>Precisión</b>	<b>Recall</b>	<b>F1</b>
BCE	0.701± 0.004	0.823± 0.015	0.826± 0.009	0.823± 0.003
BCE + Dice	0.706± 0.011	0.799± 0.003	0.858± 0.020	0.827± 0.008
Dice	0.721± 0.025	0.812± 0.024	0.865± 0.014	0.837± 0.017
Focal	0.680± 0.012	0.808± 0.019	0.812± 0.008	0.809± 0.008

Las funciones **Dice** y **BCE + Dice** mostraron los mejores valores de IoU y F1, con una ventaja clara para **Dice**, que alcanzó el rendimiento global más alto. Esto confirma su capacidad para equilibrar adecuadamente precisión y recall, permitiendo una segmentación más completa de las venas sin introducir exceso de ruido. La **BCE** aislada destacó por su precisión, con pocas falsas detecciones fuera de las venas, pero obtuvo un menor recall, lo que implica pérdida de continuidad en la red vascular. Por su parte, la **Focal** resultó ser la menos competitiva, con valores inferiores tanto en cobertura como en fidelidad de las predicciones.

### 5.5.2 Comparación entre encoders

Se compararon distintos encoders empleando U-Net como arquitectura. Los resultados medios se muestran en la Tabla 5.2.

Las diferencias entre encoders fueron muy reducidas, pero mobilenet\_v2 y efficientnet-b0 obtuvieron los mejores resultados globales, ambos con un F1 medio de 0.832. Esto indica que encoders ligeros y eficientes pueden igualar o incluso superar en rendimiento a modelos más pesados como resnet34 o resnet50, ofreciendo un equilibrio óptimo entre precisión, recall y coste computacional.

Tabla 5.2: Resultados medios (media  $\pm$  desviación típica) para U-Net con distintos encoders.

<b>Encoder</b>	<b>IoU</b>	<b>Precisión</b>	<b>Recall</b>	<b>F1</b>
<i>efficientnet-b0</i>	0.713 $\pm$ 0.027	0.802 $\pm$ 0.029	0.865 $\pm$ 0.029	0.832 $\pm$ 0.019
<i>mobilenet_v2</i>	0.713 $\pm$ 0.007	0.804 $\pm$ 0.008	0.862 $\pm$ 0.003	0.832 $\pm$ 0.005
<i>resnet34</i>	0.705 $\pm$ 0.018	0.801 $\pm$ 0.005	0.854 $\pm$ 0.027	0.826 $\pm$ 0.013
<i>resnet50</i>	0.704 $\pm$ 0.019	0.792 $\pm$ 0.007	0.863 $\pm$ 0.036	0.825 $\pm$ 0.013

A pesar de que los resultados muestran diferencias mínimas entre encoders, se optó por mantener resnet34 como base en las comparaciones de arquitecturas. Esta elección se justifica por varios motivos:

- Proporciona una convergencia más estable y predecible entre folds.
- Ofrece un equilibrio adecuado entre profundidad y coste computacional, situándose entre opciones ligeras y pesadas.
- Es un encoder ampliamente utilizado y validado en tareas de segmentación con U-Net, lo que facilita la reproducibilidad y la comparación con trabajos previos.

### 5.5.3 Comparación entre arquitecturas

Tabla 5.3 muestra los resultados medios para las arquitecturas evaluadas.

Tabla 5.3: Resultados medios (media  $\pm$  desviación típica) por arquitectura (encoder: resnet34).

<b>Arquitectura</b>	<b>IoU</b>	<b>Precisión</b>	<b>Recall</b>	<b>F1</b>
<i>DeepLabV3</i>	0.513 $\pm$ 0.002	0.552 $\pm$ 0.004	0.879 $\pm$ 0.006	0.676 $\pm$ 0.002
<i>FPN</i>	0.631 $\pm$ 0.001	0.692 $\pm$ 0.012	0.879 $\pm$ 0.020	0.773 $\pm$ 0.001
<i>PSPNet</i>	0.499 $\pm$ 0.016	0.545 $\pm$ 0.016	0.861 $\pm$ 0.077	0.665 $\pm$ 0.014
<i>U-Net</i>	0.694 $\pm$ 0.021	0.797 $\pm$ 0.002	0.842 $\pm$ 0.033	0.818 $\pm$ 0.015

La **U-Net** destacó como la arquitectura más efectiva, con un IoU y un F1 notablemente superiores al resto (alrededor de 0.69 y 0.82, respectivamente). **FPN** y **DeepLabV3** presentaron recalls elevados pero precisiones bajas, lo que sugiere sobresegmentación y pérdida de definición en los bordes. **PSPNet** fue la que mostró el peor rendimiento general, con las métricas más bajas en casi todos los apartados. En conjunto, los resultados consolidan a U-Net como la arquitectura más equilibrada y fiable para la segmentación de venas en pétalos.

En todos los casos, la evolución del entrenamiento mostró un descenso progresivo y consistente de la función de pérdida tanto en entrenamiento como en validación, con curvas que evolucionaron de manera paralela y sin una separación significativa entre ambas. Esto confirma la correcta capacidad de generalización de los modelos y la ausencia de sobreentrenamiento gracias al uso de parada temprana. La cantidad de épocas hasta la detención varió según la configuración evaluada, pero en todos los escenarios se alcanzó una convergencia adecuada. Las gráficas detalladas de la evolución de la pérdida para cada función se incluyen en el Anexo C.

#### 5.5.4 Resultados visuales

Además de las métricas cuantitativas, resulta fundamental evaluar visualmente la calidad de las máscaras generadas. En la Figura 5.4 se presentan ejemplos comparativos entre la imagen original, la máscara de referencia (*ground truth*) y la predicción de la red seleccionada (U-Net con ResNet34 y Dice). Se observa que la red consigue reproducir correctamente las venas principales, aunque presenta dificultades con la venación más fina, como se puede ver en la Figura 5.5.

### 5.6 Discusión

Las conclusiones prácticas que se extraen de estos experimentos son las siguientes:

- **Mejor arquitectura:** U-Net ofrece el mejor equilibrio entre IoU, precisión, recall y F1 en nuestras pruebas, consolidándose como la opción más adecuada para este problema. Su capacidad para preservar la estructura fina de las venas y evitar la sobresegmentación observada en arquitecturas como FPN o DeepLabV3 la convierte en la alternativa más robusta.
- **Encoder recomendado:** tanto mobilenet\_v2 como efficientnet-b0 alcanzaron los F1 medios más altos (0.832), demostrando que encoders ligeros pueden rendir igual o mejor que opciones más pesadas. Sin embargo, se mantuvo resnet34 como referencia en las comparaciones de arquitecturas por su estabilidad de entrena-



(a) Imagen original.



(b) Máscara de referencia.



(c) Predicción de la U-Net.

Figura 5.4: Resultados de segmentación: (a) Imagen original, (b) Máscara de referencia, (c) Predicción de la U-Net.

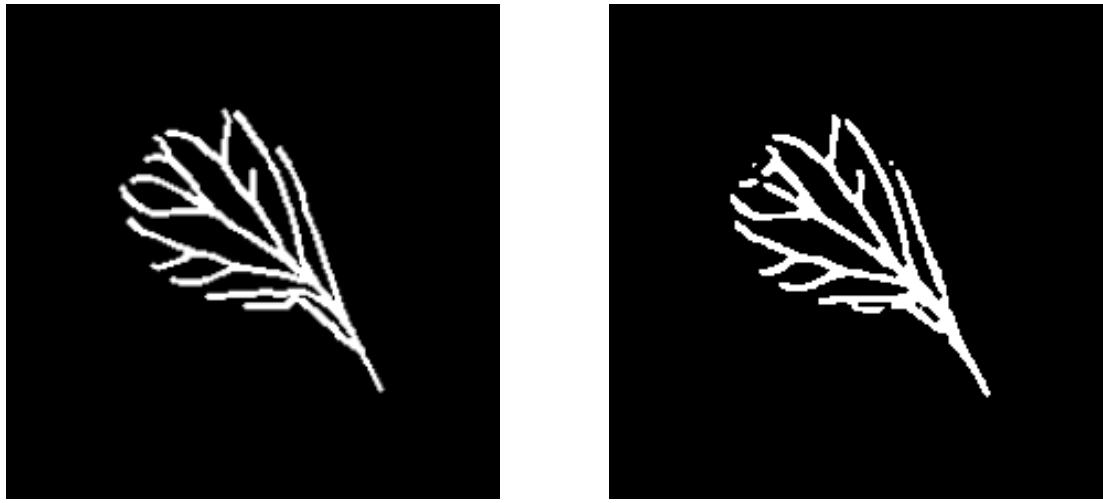


Figura 5.5: Detalle de venas finas.

miento, equilibrio entre profundidad y coste computacional, y su amplia validación en la literatura, además de la poca diferencia que se encontró entre los encoders evaluados.

- **Función de pérdida:** la función **Dice loss** y la combinación **BCE + Dice** fueron las más efectivas, con ventaja clara para **Dice**, que ofreció la mejor compensación entre precisión y recall y el mayor F1 global.
- **Precisión vs recall:** las arquitecturas como DeepLabV3 y FPN tienden a favorecer un recall alto a costa de precisión, generando sobresegmentación y ruido adicional. En contraste, U-Net mantiene un equilibrio más estable entre ambas métricas, reflejándose en un F1 superior.
- **Variabilidad por fold:** las desviaciones típicas observadas fueron bajas en la mayoría de los experimentos, indicando una alta consistencia entre folds y confirmando la estabilidad del entrenamiento y la fiabilidad de los resultados.

Con base en estos resultados, la configuración elegida para la segmentación final es: **U-Net con encoder resnet34**, entrenada con **Dice loss**. Esta combinación ofrece el mejor equilibrio entre rendimiento, estabilidad y eficiencia computacional, siendo además fácilmente adaptable a distintos tamaños de conjunto de datos y limitaciones de hardware.

## Capítulo 6

# Caracterización del patrón de venación

---

## 6.1 Introducción

La segmentación de venas constituye el primer paso para la extracción del patrón de venación, pero no es suficiente por sí sola para caracterizarlo de manera completa. A partir de las máscaras segmentadas es necesario llevar a cabo un proceso de caracterización que permita describir de forma objetiva la organización de la red vascular. Esta caracterización resulta especialmente relevante para detectar posibles alteraciones fisiológicas derivadas de condiciones de estrés, como el estrés salino.

La caracterización de patrones vasculares en órganos vegetales ha sido abordada en estudios previos centrados en hojas, en los que se ha demostrado la utilidad de métricas estructurales como la densidad de venación, el tamaño de aréolas o el número de ramificaciones para diferenciar entre tratamientos y especies [34, 35]. En este capítulo se presentan las técnicas empleadas para obtener el esqueleto de las máscaras segmentadas, las métricas definidas a partir de dicho esqueleto y el análisis comparativo entre grupos experimentales.

## 6.2 Obtención del esqueleto

Una vez obtenidas las máscaras binarias correspondientes a las venas, el siguiente paso consiste en transformar dichas máscaras en un *esqueleto*. El esqueleto es una representación unidimensional que conserva la conectividad de la red vascular original, reduciéndola a su forma más simple sin pérdida de información topológica.

Para este propósito se utilizó el algoritmo de Zhang-Suen para imágenes 2D, el cual aplica un proceso iterativo de adelgazamiento que reduce las venas a líneas de un píxel de grosor, manteniendo la conectividad de la estructura [36]. Este método, ampliamente empleado en

procesamiento de imágenes biomédicas y botánicas, facilita el cálculo posterior de métricas de longitud, ramificación y geometría.

En la Figura 6.1 se muestra un ejemplo ilustrativo del proceso: a partir de una máscara binaria segmentada se obtiene el esqueleto correspondiente, que conserva la estructura global de la venación.

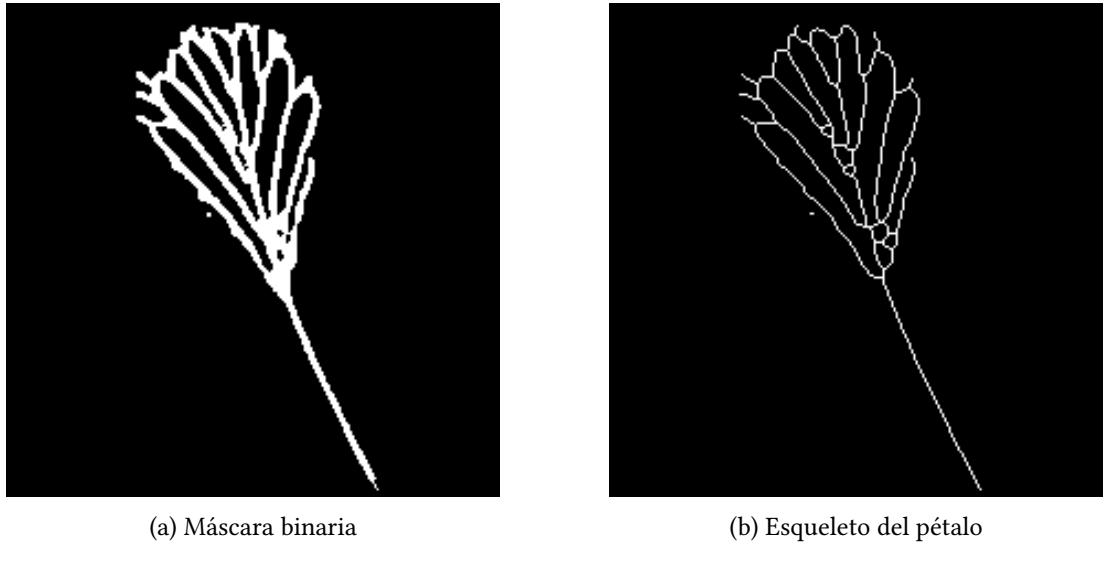


Figura 6.1: Obtención del esqueleto a partir de una imagen binaria.

### 6.3 Definición y cálculo de métricas

Una vez generado el esqueleto, se definieron un conjunto de métricas cuantitativas para caracterizar los patrones de venación. Estas métricas permiten describir tanto propiedades globales como locales de la red vascular [37, 38, 39, 40]

- **Densidad de venas (VD):** relación entre el área total ocupada por las venas  $A_{venas}$  y el área del pétalo  $A_{pétalo}$ .

$$VD = \frac{A_{venas}}{A_{pétalo}} \quad (6.1)$$

Para calcular el área del pétalo se aplicó una umbralización automática basada en el color del pétalo en el espacio HSV a la imagen original. La imagen RGB se convirtió al espacio HSV, y a continuación se definió un rango de valores de matiz (Hue), saturación (Saturation) y brillo (Value) que correspondían al color característico del pétalo. Los píxeles cuya combinación de HSV se encontraba dentro de este rango se clasificaron como pertenecientes al pétalo, mientras que el resto se consideró fondo. De este modo se generó una máscara binaria con todos los pétalos segmentados (Figura 6.2).

Realizando un análisis de componentes conectados se puede extraer cada pétalo de forma independiente y, de este modo, calcular su área, esto es, el número de píxeles que lo formó.

- **Grosor de venas (VT):** ancho medio de las venas calculado mediante la transformada de distancia sobre los puntos centrales del esqueleto (nodos con grado 2), evitando bifurcaciones y extremos. En cada punto, el grosor local se estima como el doble de la distancia al píxel de fondo más cercano, ya que esta distancia equivale al radio de una vena.

$$VT = \frac{2}{N_c} \sum_{i=1}^{N_c} r_i \quad (6.2)$$

Donde  $r_i$  representa el radio local en puntos del esqueleto con grado 2 y  $N_c$  el número total de puntos de la vena.

- **Tamaño de aréolas (AS):** diámetro equivalente promedio de las aréolas delimitadas por las venas, medido en píxeles.

$$AS = \frac{1}{N_a} \sum_{j=1}^{N_a} 2\sqrt{\frac{A_j}{\pi}} \quad (6.3)$$

Donde  $A_j$  corresponde al área de la aréola  $j$  y  $N_a$  al número total de aréolas. El término  $2\sqrt{\frac{A_j}{\pi}}$  representa el diámetro equivalente de la aréola. Este cálculo se realiza invirtiendo la máscara para identificar regiones cerradas por venas mediante etiquetado de componentes conectados. El número de regiones corresponde al número de aréolas, mientras que el área media de estas regiones da el tamaño característico. Se puede ver la imagen resultante en la Figura 6.3.

- **Número de aréolas (NA):** conteo de regiones internas delimitadas por la red venosa.

$$NA = M \quad (6.4)$$

- **Ángulo de ramificación (BA):** valor promedio de los ángulos  $\theta_k$  formados en los puntos de bifurcación, medido en grados.

$$BA = \frac{1}{N_b} \sum_{k=1}^{N_b} \frac{1}{d_k} \sum_{i=1}^{d_k} \theta_{ik} \quad (6.5)$$

Donde  $N_b$  es el número de bifurcaciones. Tras esqueletizar la red vascular, se construye un grafo donde los nodos representan píxeles de la vena. En cada nodo de bifurcación (grado  $\geq 3$ ) se calculan todos los ángulos  $\theta_{ik}$  formados entre pares de ramas adyacentes,

y se calcula primero el promedio de estos ángulos dentro de cada bifurcación. A continuación, se promedian los valores obtenidos para todas las bifurcaciones para obtener el ángulo medio global de ramificación (BA). En la Figura 6.4 se pueden observar los puntos de bifurcación detectados en una imagen.

- **Distancia entre venas (VVD):** separación media entre venas adyacentes, medido en píxeles.

$$VVD = \frac{1}{P} \sum_{l=1}^P s_l \quad (6.6)$$

Donde  $s_l$  representa la distancia entre venas y  $P$  el número de mediciones realizadas. Para un conjunto aleatorio de 200 puntos del esqueleto situados entre bifurcaciones, se calcula la dirección perpendicular local a la vena y se mide la distancia hasta la vena adyacente más cercana en ambas direcciones.

- **Número de venas principales (MV):** cantidad de ramas que se extienden desde los extremos del esqueleto hasta una bifurcación y cuya longitud (en número de píxeles) supera un umbral mínimo  $l_{min}$ .

$$MV = \sum_{m=1}^M [L_m \geq L_{\min}] \quad (6.7)$$

Donde  $L_m$  es la longitud de la  $m$ -ésima vena detectada,  $L_{\min}$  la longitud mínima establecida para considerar una vena como principal, y  $M$  el número total de ramas analizadas. Para su cálculo se esqueletiza la red venosa, se identifican los extremos del esqueleto (nodos de grado 1) y se recorren las ramas hasta bifurcaciones o extremos opuestos. Solo aquellas con longitud mayor o igual a  $L_{\min}$  se contabilizan como venas principales.

## 6.4 Análisis del patrón de venación en *Raphanus raphanistrum* sometido a estrés salino

Para estudiar el efecto del **estrés salino** sobre la organización vascular de los pétalos, se siguió un flujo experimental sistemático. En primer lugar, cada imagen de pétalo se procesó mediante el modelo de segmentación previamente entrenado (U-Net con el encoder seleccionado) para generar una máscara binaria en la que los píxeles correspondientes a venas se distinguen claramente del fondo. A continuación, a partir de esta máscara se obtuvo el esqueleto de la red venosa, reduciendo cada vena a su trazo central y preservando la topología y conectividad de la red. Este paso facilitó la cuantificación de métricas estructurales como longitud, ramificación y ángulos de bifurcación. Finalmente, sobre el esqueleto se calcularon

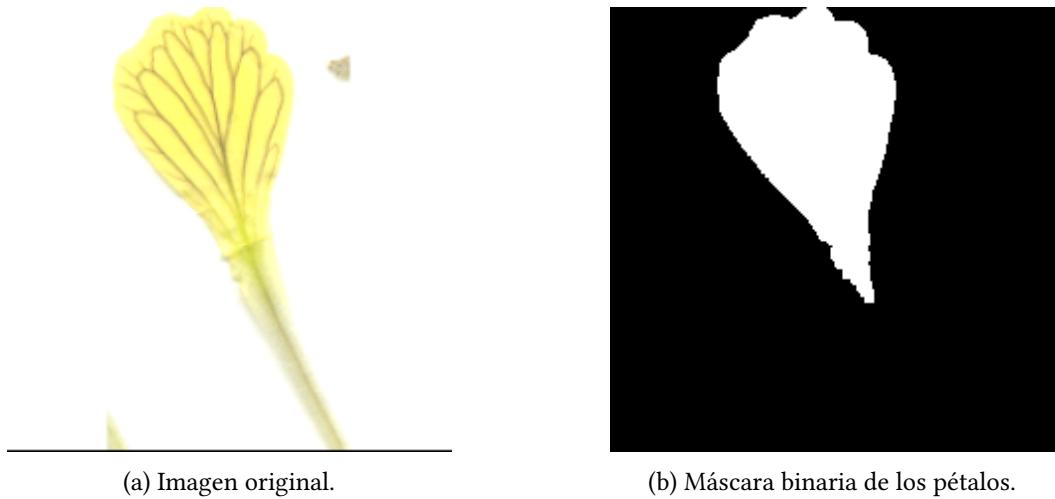


Figura 6.2: Máscaras obtenida de aplicar umbralización.

las métricas de venación seleccionadas, incluyendo densidad de venación, tamaño y número de aréolas, grosor de venas, ángulo de ramificación y distancia entre venas. Los resultados se agruparon según las condiciones experimentales para permitir comparaciones entre el grupo control y los grupos sometidos a estrés salino.

Los resultados individuales de cada imagen se encuentran en el apéndice B. La Tabla 6.1 recoge los valores promedio obtenidos en cada grupo.

Tabla 6.1: Promedios de las métricas de venación obtenidas para cada grupo experimental.

Grupo	VD	VT	AS	NA	BA	VVD	MV
Control	0.479 ± 0.130	3.523 ± 0.633	18.218 ± 6.635	5.430 ± 2.495	113.613 ± 2.163	9.193 ± 1.188	7.696 ± 4.313
Estrés leve	0.482 ± 0.047	3.428 ± 0.223	15.652 ± 3.177	5.792 ± 2.273	114.013 ± 1.263	8.306 ± 1.050	8.597 ± 3.551
Estrés fuerte	0.547 ± 0.125	3.586 ± 0.618	14.275 ± 4.476	5.806 ± 2.330	113.806 ± 2.334	8.466 ± 1.152	6.764 ± 3.151

**Extracción de Métricas** Para validar la calidad de las métricas calculadas a partir de las segmentaciones automáticas, se compararon con aquellas obtenidas directamente de las máscaras de referencia (*ground truth*). En la Tabla 6.2 se presentan los promedios de cada métrica en ambos casos, mientras que en la Figura 6.5 se representan las distribuciones de algunas métricas clave mediante diagramas de caja.

En la Tabla 6.3 se muestra la diferencia media absoluta (MAD) entre las métricas obtenidas mediante segmentación automática y las derivadas del *ground truth*. Esta métrica cuantifica la desviación promedio de las predicciones respecto a las referencias manuales, proporcionando una medida clara de la precisión alcanzada.

Tabla 6.2: Comparación entre las métricas obtenidas mediante segmentación automática (predicción) y a partir del *ground truth* (manual). Los valores se expresan como media  $\pm$  desviación típica.

	<b>VD</b>	<b>VT</b>	<b>AS</b>	<b>NA</b>	<b>BA</b>	<b>VVD</b>	<b>MV</b>
<i>Pred.</i>	0.502 $\pm$ 0.111	3.511 $\pm$ 0.527	16.106 $\pm$ 5.240	5.671 $\pm$ 2.366	113.809 $\pm$ 1.968	8.672 $\pm$ 1.208	7.706 $\pm$ 3.778
<i>GT</i>	0.464 $\pm$ 0.100	3.705 $\pm$ 0.171	15.291 $\pm$ 5.611	5.158 $\pm$ 2.660	109.033 $\pm$ 3.769	8.179 $\pm$ 1.209	7.667 $\pm$ 3.542

Tabla 6.3: Diferencia media absoluta (MAD) entre las métricas obtenidas mediante segmentación automática y el *ground truth*, incluyendo desviación típica.

<i>Métrica</i>	<i>MAD <math>\pm</math> DT</i>
<i>Densidad de Venas (VD)</i>	0.033 $\pm$ 0.022
<i>Grosor de Venas (VT)</i>	0.699 $\pm$ 0.283
<i>Tamaño de Areolas (AS)</i>	2.352 $\pm$ 3.260
<i>Número de Areolas (NA)</i>	1.300 $\pm$ 1.127
<i>Ángulo de Ramificación (BA)</i>	4.951° $\pm$ 3.195°
<i>Distancia Vena a Vena (VVD)</i>	0.643 $\pm$ 0.495
<i>Venas Principales (MV)</i>	1.685 $\pm$ 1.526

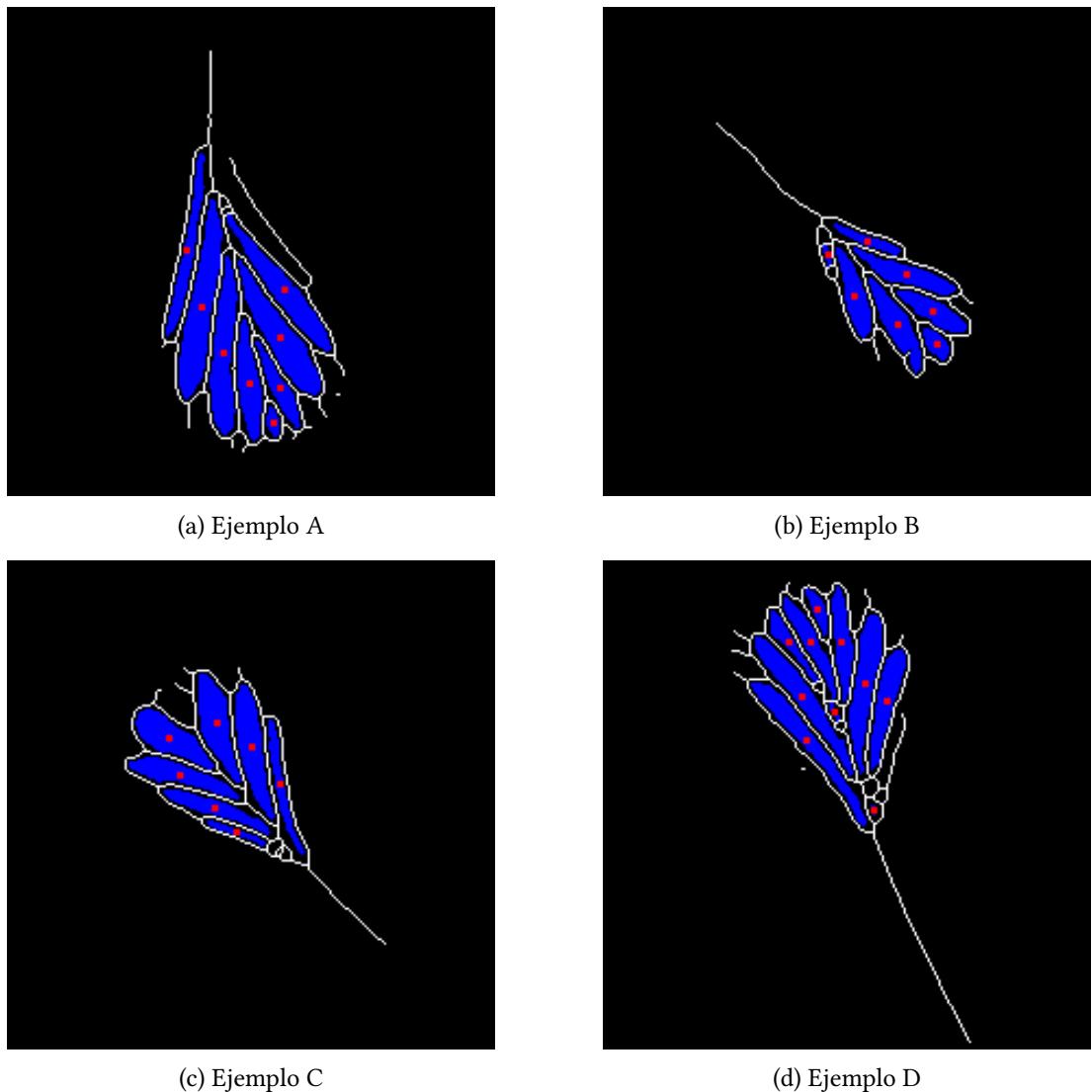
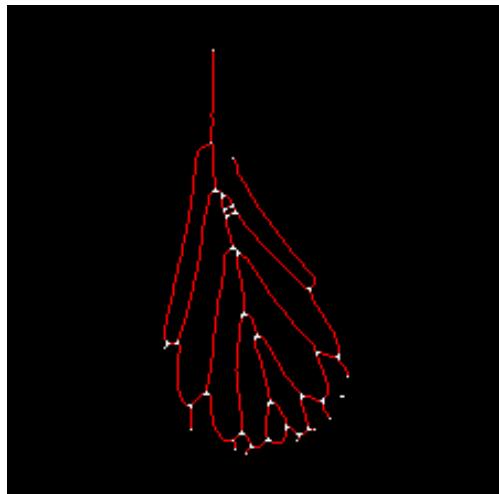
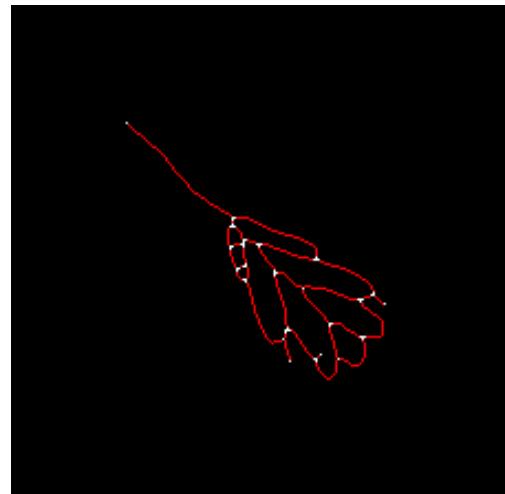


Figura 6.3: Vistas representativas de la segmentación y etiquetado de aréolas.

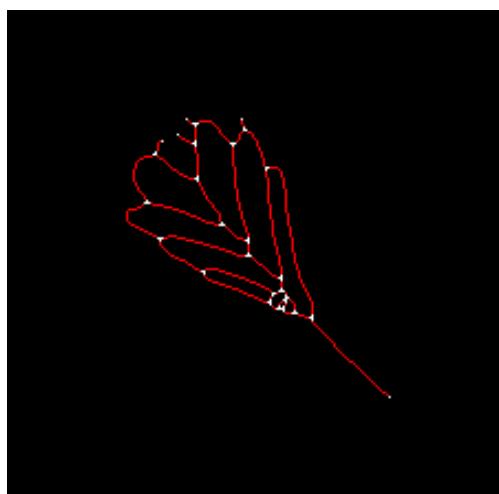
La comparación revela que las diferencias entre las métricas derivadas de la segmentación automática y las de referencia son, en general, reducidas. En particular, la **densidad de venas (VD)** y la **distancia vena a vena (VVD)** presentan los valores de MAD más bajos (0.033 y 0.013, respectivamente), lo que indica una alta fidelidad en la reconstrucción de la estructura venosa principal. Las mayores discrepancias se observan en el **tamaño de las areolas (AS)** y en el **ángulo de ramificación (BA)**, con desviaciones medias de 2.352 y 4.951°, respectivamente, lo que sugiere que las predicciones tienden a simplificar las regiones más finas o complejas de la red. Aun así, el bajo error medio en el resto de métricas confirma que el modelo conserva de forma coherente la geometría global de las venas y reproduce con buena precisión los patrones estructurales presentes en el *ground truth*.



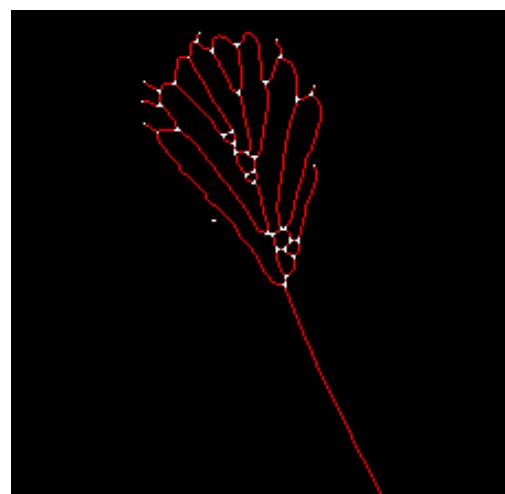
(a) Ejemplo A



(b) Ejemplo B



(c) Ejemplo C



(d) Ejemplo D

Figura 6.4: Imágenes con los puntos entre bifurcaciones marcados.

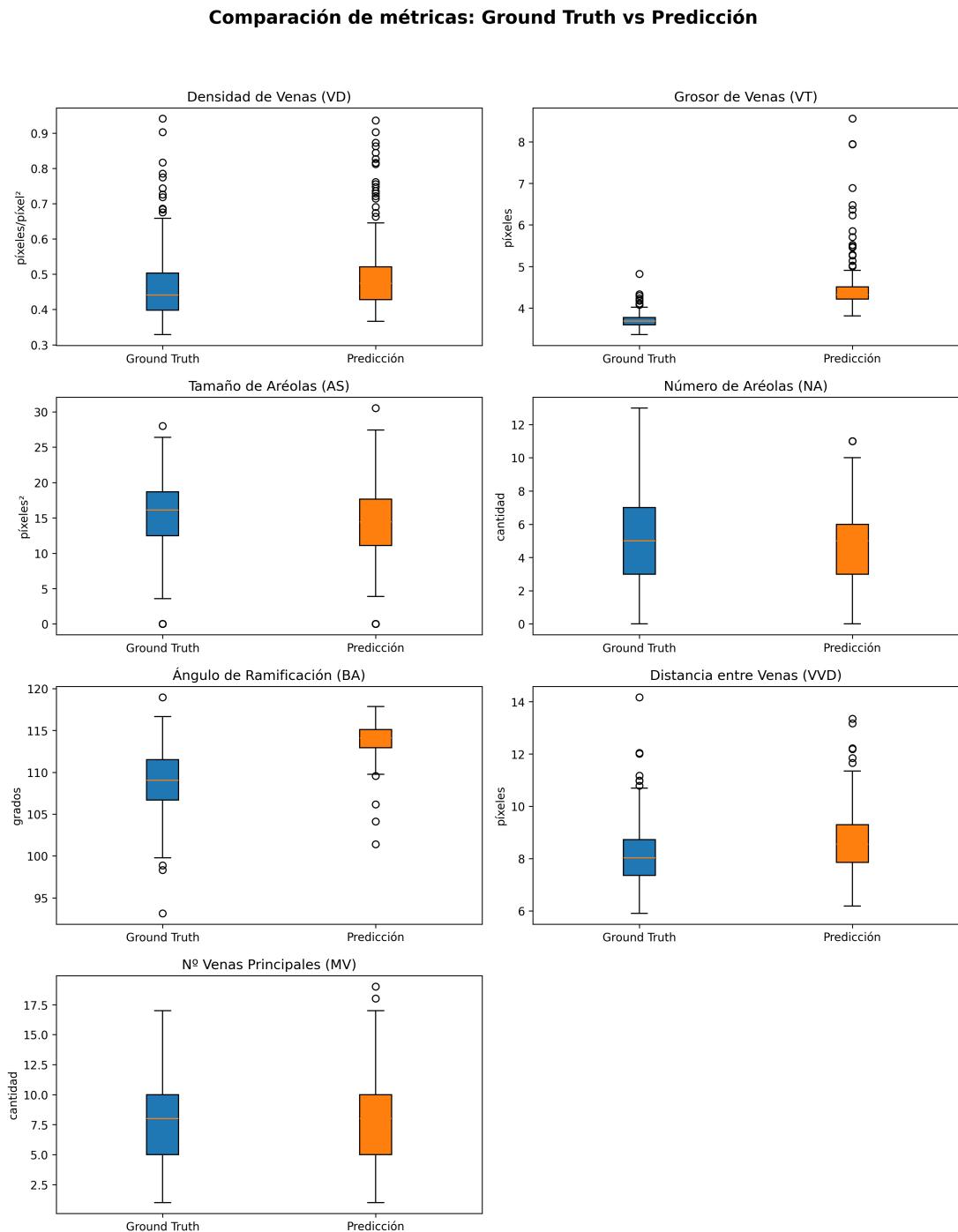


Figura 6.5: Comparación de las distribuciones de métricas.

## Capítulo 7

# Análisis estadístico ANOVA

---

Una vez caracterizadas las métricas morfológicas asociadas al patrón de venación, se procedió a evaluar si existían diferencias significativas entre los tratamientos experimentales aplicados. Para ello se empleó un análisis de la varianza de un factor (ANOVA) [41], seguido de un test post-hoc de Tukey en aquellos casos en los que se detectaron diferencias.

El análisis estadístico se realizó exclusivamente sobre las métricas obtenidas a partir de las máscaras de referencia (*ground truth*). Esta decisión se fundamenta en que se dispone de segmentaciones manuales para todas las imágenes. Aunque se ha comprobado que las métricas derivadas de las segmentaciones automáticas son muy similares a las obtenidas a partir del *ground truth*, se optó por evitar cualquier posible fuente de imprecisión en el análisis estadístico. De este modo, al basar el ANOVA únicamente en los datos del *ground truth*, se garantiza que los resultados reflejen con la mayor fidelidad posible la organización real de la red vascular.

No obstante, disponer de un método automático de segmentación resulta fundamental, ya que permite realizar análisis estadísticos sobre grandes conjuntos de imágenes en los que no se cuente con máscaras manuales.

## 7.1 Metodología

El ANOVA de un factor permite determinar si la media de una métrica difiere significativamente entre varios grupos. En este estudio, los grupos corresponden a los diferentes tratamientos (*Control*, *Estrés leve*, *Estrés fuerte*). El procedimiento seguido fue el siguiente:

1. A partir de las métricas extraídas (densidad de venas, grosor, tamaño de aréolas, número de aréolas, ángulo de ramificación, distancia vena a vena y número de venas principales), se asociaron las observaciones al tratamiento correspondiente en función del nombre de la imagen.

2. Se verificaron los supuestos de normalidad y homogeneidad de varianzas. Dado que los tamaños muestrales eran comparables y las distribuciones no mostraron grandes asimetrías, se aplicó el ANOVA de un factor.
3. Para cada métrica se calculó el estadístico F y su p-valor asociado.
4. En aquellos casos con diferencias significativas ( $p < 0.05$ ), se aplicó un test de comparaciones múltiples de Tukey HSD [42], que permite identificar qué pares de tratamientos difieren significativamente.

### 7.1.1 Resultados del ANOVA

La Tabla 7.1 muestra los valores de F y p-valor obtenidos para cada métrica. Se observaron diferencias estadísticamente significativas en la **densidad de venas (VD)**, el **tamaño de aréolas (AS)** y el **ángulo de ramificación (BA)**, con valores de  $p < 0.05$ . En particular, VD y AS presentaron una significancia muy alta ( $p < 0.001$ ), indicando que estas métricas varían notablemente entre tratamientos. El resto de las métricas no mostraron diferencias significativas.

Tabla 7.1: Resultados del ANOVA aplicado a las métricas obtenidas del *ground truth*.

Métrica	F	p
Densidad de Venas (VD)	12.10	<b>1.02e-05</b>
Grosor de Venas (VT)	0.01	0.990
Tamaño de Aréolas (AS)	8.82	<b>0.000206</b>
Número de Aréolas (NA)	1.10	0.334
Ángulo de Ramificación (BA)	3.39	<b>0.0354</b>
Distancia Vena a Vena (VVD)	0.49	0.615
Número de Venas Principales (MV)	1.97	0.142

El test de Tukey HSD permitió identificar entre qué tratamientos se producen las diferencias detectadas por el ANOVA. Los resultados se resumen a continuación:

- **Densidad de venas (VD):** se observaron diferencias significativas entre los grupos *Control* y *Estrés fuerte* ( $p < 0.001$ ), así como entre *Estrés leve* y *Estrés fuerte* ( $p = 0.0015$ ). En ambos casos, el tratamiento de *estrés fuerte* presentó una **menor densidad de venas**. No se encontraron diferencias entre *Control* y *Estrés leve* ( $p = 0.3681$ ).

- **Tamaño de aréolas (AS):** el grupo *Control* difirió significativamente de los grupos *Estrés leve* ( $p = 0.010$ ) y *Estrés fuerte* ( $p = 0.0002$ ), mostrando un **tamaño de aréolas menor** en las plantas control. No se detectaron diferencias entre *Estrés leve* y *Estrés fuerte* ( $p = 0.433$ ).
- **Ángulo de ramificación (BA):** se encontró una diferencia significativa entre *Estrés fuerte* y *Estrés leve* ( $p = 0.0268$ ), indicando un **mayor ángulo de ramificación** bajo *estrés fuerte*. No se observaron diferencias entre el grupo *Control* y los tratamientos de estrés.
- Para el resto de métricas (**número de aréolas (NA)**, **distancia vena a vena (VVD)** y **número de venas principales (MV)**), no se hallaron diferencias estadísticamente significativas entre tratamientos ( $p_{ajustado} > 0.05$ ).

En conjunto, estos resultados indican que las condiciones de estrés afectan principalmente la **estructura de la red de venas secundarias**. En concreto, el *estrés fuerte* reduce la densidad de venas y aumenta el tamaño medio de las aréolas, reflejando una red vascular más laxa. Asimismo, el *ángulo de ramificación* tiende a aumentar bajo estrés intenso, lo que sugiere una modificación en el patrón de ramificación de las venas. Por el contrario, no se detectaron efectos significativos sobre el número de aréolas, la distancia entre venas principales ni el número de venas principales.

## 7.2 Conclusiones

El análisis ANOVA reveló efectos significativos del tratamiento en las métricas **tamaño de aréolas (AS)**, **densidad de venas (VD)** y **ángulo de ramificación (BA)**. Sin embargo, el test post-hoc de Tukey confirmó diferencias únicamente para **AS** y **BA**, indicando que los cambios estructurales más evidentes se manifiestan en la organización y geometría de la red vascular, más que en su densidad global.

En particular, las plantas sometidas a *estrés leve* y *estrés fuerte* presentaron un **incremento significativo en el tamaño de las aréolas** respecto al grupo *Control*, lo que sugiere una reorganización de la red vascular que reduce la cantidad de venas secundarias y amplía las áreas interveinales. Además, se observó que el **ángulo de ramificación (BA)** fue mayor bajo *estrés fuerte* en comparación con *estrés leve*, lo que podría reflejar una alteración en el patrón de crecimiento de las venas principales en respuesta a condiciones de estrés más severo.

Aunque la **densidad de venas (VD)** mostró diferencias significativas en el ANOVA, el test de Tukey no confirmó contrastes estadísticamente sólidos entre tratamientos, por lo que no puede afirmarse de forma concluyente que el estrés salino modifique este parámetro de manera sistemática.

La alta variabilidad intra-grupo observada (véase el Apéndice B) indica que la respuesta vascular no es homogénea entre individuos del mismo tratamiento, posiblemente debido a diferencias fenotípicas naturales o microvariaciones en la intensidad del estrés aplicado.

En conjunto, los resultados evidencian que el estrés salino induce modificaciones morfológicas en la red de venación de los pétalos, especialmente en el tamaño y disposición de las aréolas, y en menor medida en la geometría de ramificación. Futuros estudios con un número mayor de muestras y un control más preciso de las condiciones experimentales permitirán validar y profundizar en estos patrones de respuesta vascular.

## Capítulo 8

# Herramienta para segmentación y análisis

---

## 8.1 Introducción

Con el objetivo de automatizar el flujo de trabajo desarrollado en los capítulos anteriores —desde la segmentación de venación hasta la extracción de métricas cuantitativas— se implementó una herramienta software con interfaz gráfica de usuario (GUI). La aplicación integra en un único entorno las operaciones de carga del modelo, segmentación de imágenes, cálculo de métricas y procesamiento por lotes, facilitando la trazabilidad y la reproducibilidad del análisis.

## 8.2 Análisis

El desarrollo de la herramienta se guió por un conjunto de requisitos funcionales y no funcionales.

Los **requisitos funcionales** se centran en las capacidades que la aplicación debe ofrecer al usuario final. Entre ellos, se incluyó la posibilidad de cargar pesos de un modelo de segmentación entrenado, abrir imágenes en diversos formatos estándar y ejecutar inferencias para obtener máscaras binarias de venación. Además, se consideró esencial poder visualizar tanto la imagen de entrada como la máscara y la superposición (*overlay*), ajustando el umbral de binarización y la transparencia. La herramienta también debía calcular métricas directamente desde una máscara binaria, ya fuera generada por el modelo o cargada desde archivo, así como soportar el procesamiento masivo de imágenes de una carpeta. Finalmente, se estableció como requisito la exportación de resultados, tanto en formato tabular (CSV) como en imágenes derivadas.

Por otro lado, los **requisitos no funcionales** definieron criterios de calidad. Se priorizó

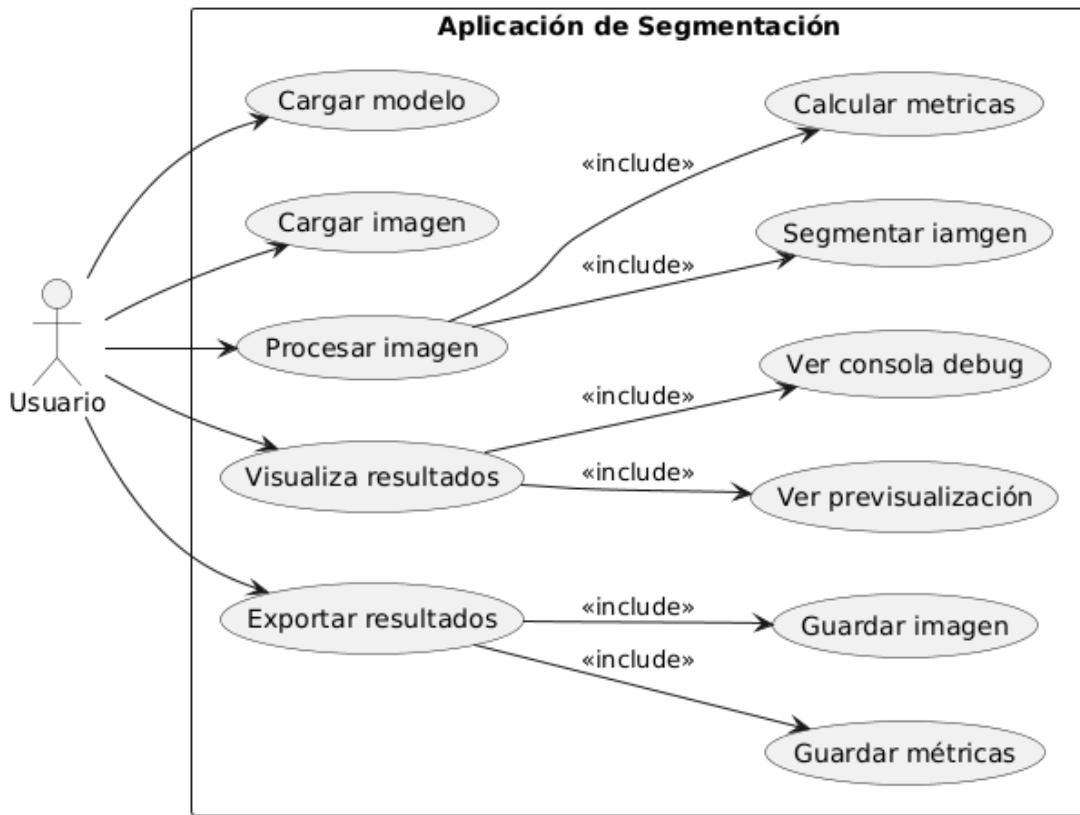


Figura 8.1: Diagrama de casos de uso de la herramienta de segmentación y análisis de venación.

la portabilidad en CPU, con la posibilidad de aprovechar GPU si está disponible. También se buscó la usabilidad mediante una interfaz clara basada en pestañas y controles directos, junto con la reproducibilidad mediante un registro de acciones en una consola integrada de depuración.

La relación entre actores y funcionalidades principales se resume en el diagrama de casos de uso mostrado en la Figura 8.1.

### 8.3 Diseño y arquitectura

Los principales módulos que componen la herramienta son los siguientes:

- **models**: El módulo `models` implementa la carga dinámica de pesos en formato `.pth`, junto con la configuración del tipo de arquitectura y `encoder`. De forma predeterminada, la aplicación utiliza una arquitectura `Unet` con `resnet34` como `encoder`, aunque el usuario puede seleccionar otras opciones compatibles. El preprocesamiento incluye redimensionado de entrada, normalización y preparación del tensor. Tras la inferencia,

los mapas de probabilidad se convierten en máscaras binarias mediante la aplicación de un umbral configurable en la interfaz. Esta máscara constituye la base para el análisis de venación posterior.

- **metrics**: El módulo `metrics` implementa funciones para extraer medidas cuantitativas a partir de máscaras binarias. Estas métricas se calculan tanto a partir de imágenes procesadas por el modelo como de máscaras proporcionadas por el usuario. El resultado puede visualizarse en la aplicación y exportarse en formato tabular.
- **ui\_components**: define elementos reutilizables de la interfaz gráfica.
- **theme**: gestiona el estilo visual y la apariencia de la aplicación.

**Cálculo de métricas.** El diagrama UML de clases, mostrado en la Figura 8.2, permite visualizar de manera global cómo interactúan los distintos componentes del software:

- **Clase principal *SegTkApp***: orquesta la interacción entre la GUI y la lógica de procesamiento.
- **Modelo de segmentación *CamVidModel***: encapsula la arquitectura y pesos de la red neuronal para generar máscaras binarias.
- **Cálculo de métricas *Metrics***: contiene métodos para computar métricas de venación y estadísticas derivadas.
- **Componentes de interfaz y estilo (*Theme*, *ScrollableFrame*, *ttk.Style*)**: gestionan la apariencia visual y la navegabilidad dentro de la GUI.
- **Clases auxiliares *PetalVeinDataset* e *Image***: soportan el procesamiento por lotes y la representación de imágenes.

## 8.4 Interfaz de usuario

La interfaz gráfica se estructura en pestañas organizadas en tarjetas (*cards*), lo que facilita la navegación.

El panel de entrada permite seleccionar los pesos del modelo y configurar parámetros básicos. A continuación, la sección de previsualización muestra la imagen original, la máscara obtenida y la superposición, con controles de umbral y transparencia. La sección de métricas presenta los resultados del análisis en una tabla, mientras que el módulo de procesamiento por lotes automatiza la ejecución sobre carpetas completas de imágenes. Por último, se incluye una consola de depuración para seguimiento de mensajes y errores.

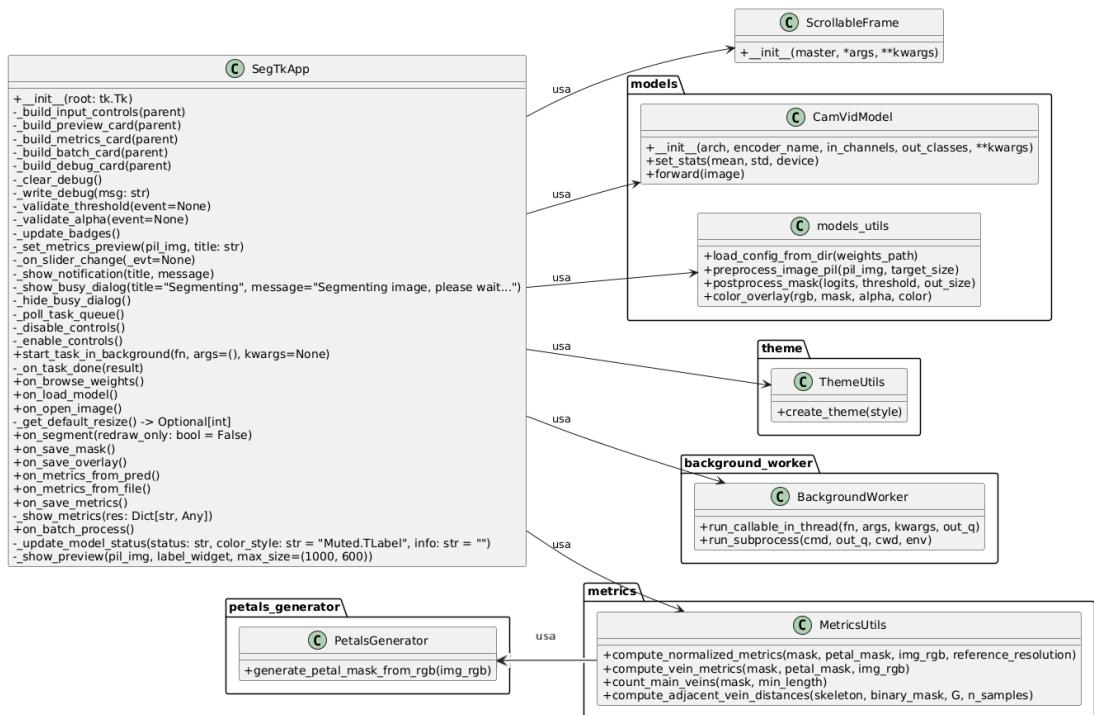


Figura 8.2: Diagrama UML de la herramienta de segmentación y análisis de venación.



Figura 8.3: Mockup de la interfaz gráfica de la herramienta de segmentación y análisis.

La Figura 8.3 muestra un prototipo visual de la aplicación con los principales paneles integrados.

## 8.5 Flujo de la aplicación

La aplicación desarrollada está diseñada para facilitar la segmentación automática de venas en imágenes de pétalos y el análisis cuantitativo de los resultados. Su interfaz gráfica guía al usuario a través de un flujo de trabajo intuitivo, desde la carga del modelo hasta la exportación de métricas, permitiendo tanto el procesamiento de imágenes individuales como de lotes completos. La siguiente descripción detalla cada paso del flujo, destacando las funcionalidades disponibles y los parámetros ajustables que permiten personalizar la segmentación y la visualización de los resultados:

1. **Carga del modelo:** el usuario selecciona un archivo de pesos preentrenados (.pth) y, de forma automática, el sistema busca en la misma carpeta un archivo config.json. Este archivo contiene la información necesaria para reconstruir la arquitectura exacta con la que se entrenó el modelo, incluyendo el tipo de red ("arch"), el encoder utilizado ("encoder\_name") y la función de pérdida empleada durante el entrenamiento ("loss\_fn"). De esta manera se garantiza que los pesos se cargan sobre una configuración coherente y reproducible. Una vez inicializado, se muestra información detallada

sobre la red, el *encoder* y el dispositivo de cálculo (CPU o GPU), confirmando que el modelo está listo para la inferencia.

2. **Apertura de imagen:** mediante la pestaña *Input*, el usuario puede visualizar la imagen original que se desea segmentar. Esta sección habilita los controles de inferencia y proporciona una previsualización de la imagen, facilitando la selección de las regiones de interés y asegurando que se procesa el archivo correcto.
3. **Segmentación:** al pulsar el botón *Segment*, el modelo realiza la inferencia sobre la imagen cargada y genera tanto la máscara binaria resultante como un *overlay* superpuesto sobre la imagen original. El usuario puede ajustar parámetros como **Threshold**, que controla el umbral a partir del cual un pixel se clasificará como vena a partir de la imagen generada por la red, y **Overlay alpha**, que define la transparencia del *overlay*, permitiendo una visualización más clara y personalizada de la segmentación.
4. **Cálculo de métricas:** desde la pestaña *Metrics*, la aplicación calcula automáticamente métricas de evaluación como Dice, IoU, precisión y sensibilidad, mostrando los resultados de manera inmediata. Además, se ofrece la opción de exportar estos datos a un archivo CSV, lo que facilita su posterior análisis estadístico y la comparación entre distintas imágenes o configuraciones del modelo.
5. **Procesamiento por lotes:** esta funcionalidad permite procesar automáticamente todas las imágenes contenidas en una carpeta, generando para cada una su máscara correspondiente y guardando un resumen de métricas en `results/metrics_summary.csv`. Esto agiliza el análisis de grandes conjuntos de datos y asegura que todos los resultados se almacenan de manera organizada y reproducible.

Este comportamiento se representa de manera detallada en el diagrama UML de secuencia mostrado en la Figura 8.4, donde se observa la interacción entre los distintos módulos de la aplicación a lo largo del flujo de trabajo.

## 8.6 Implementación

Una vez definidos los requisitos y la arquitectura de la herramienta, se procedió a su implementación en **Python 3.11.2**, utilizando las librerías descritas previamente. El desarrollo siguió un enfoque modular y orientado a mantener la responsividad de la interfaz, facilitando así el mantenimiento y la extensibilidad del software.

A continuación se muestran ejemplos reales del funcionamiento de la herramienta. En la Figura 8.5 se observa la interfaz inicial sin imagen cargada, mientras que en la Figura 8.6 se

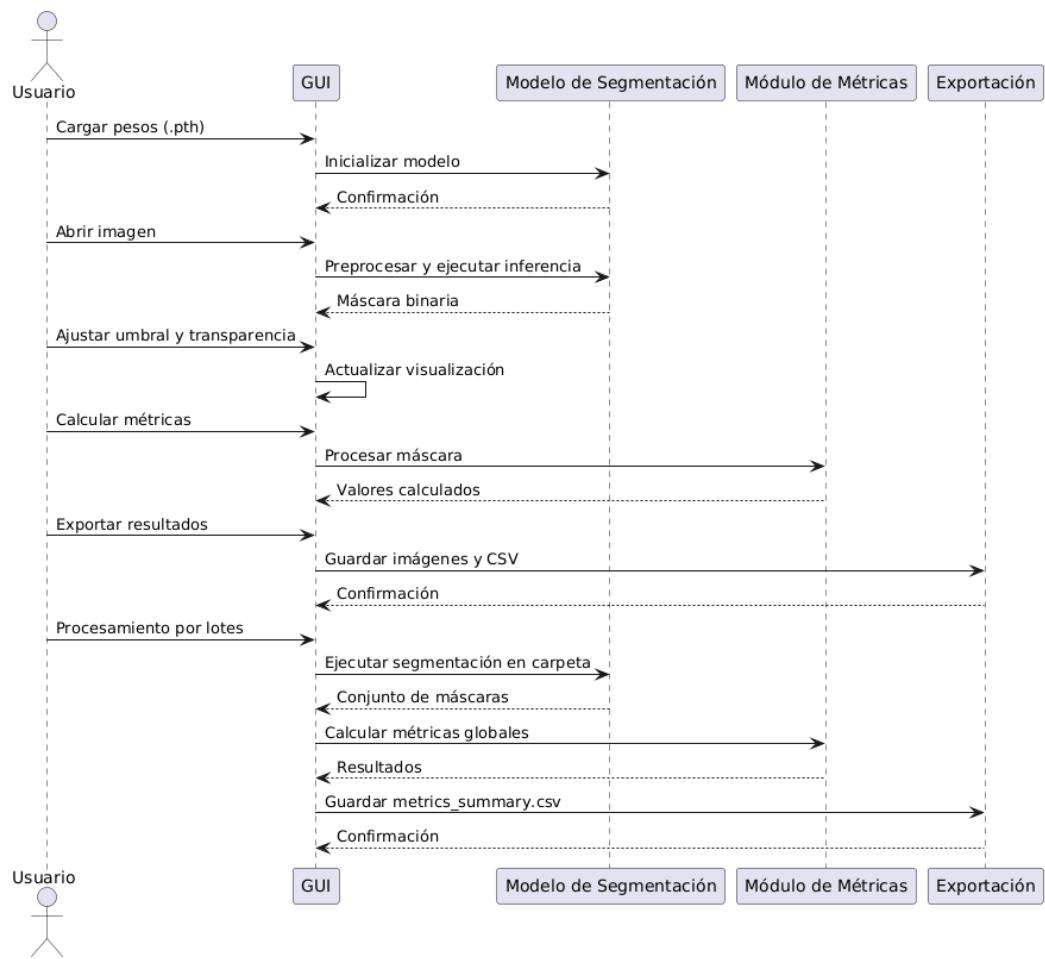


Figura 8.4: Diagrama de secuencia del flujo de uso de la herramienta.

presenta un caso de uso completo con una imagen segmentada y la correspondiente máscara superpuesta sobre el pétalo.

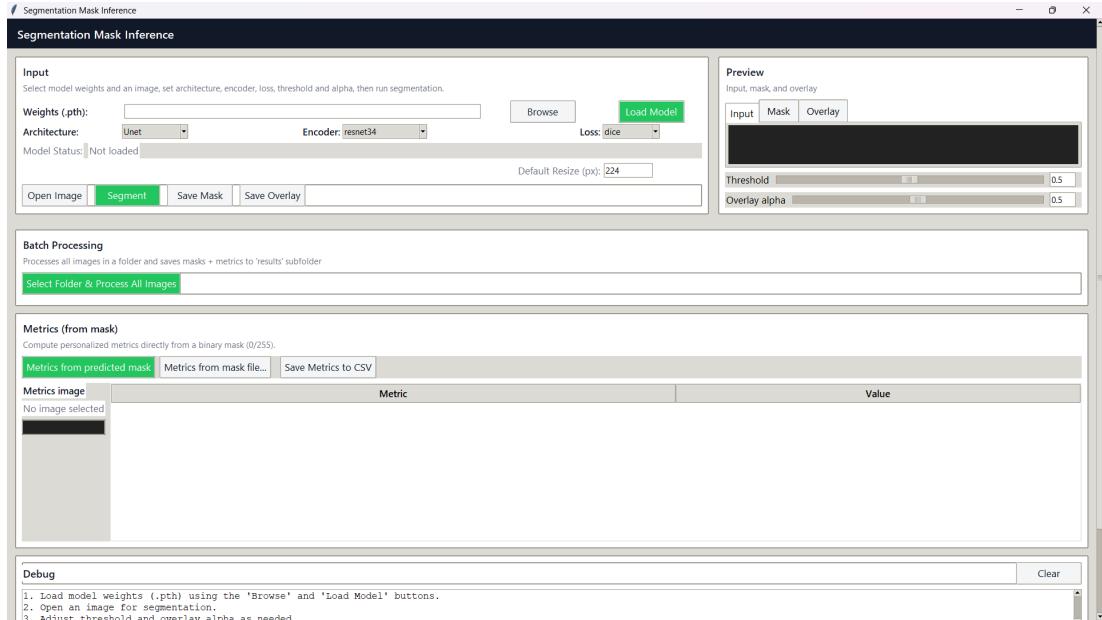


Figura 8.5: Interfaz de la herramienta al iniciarse, antes de cargar una imagen.

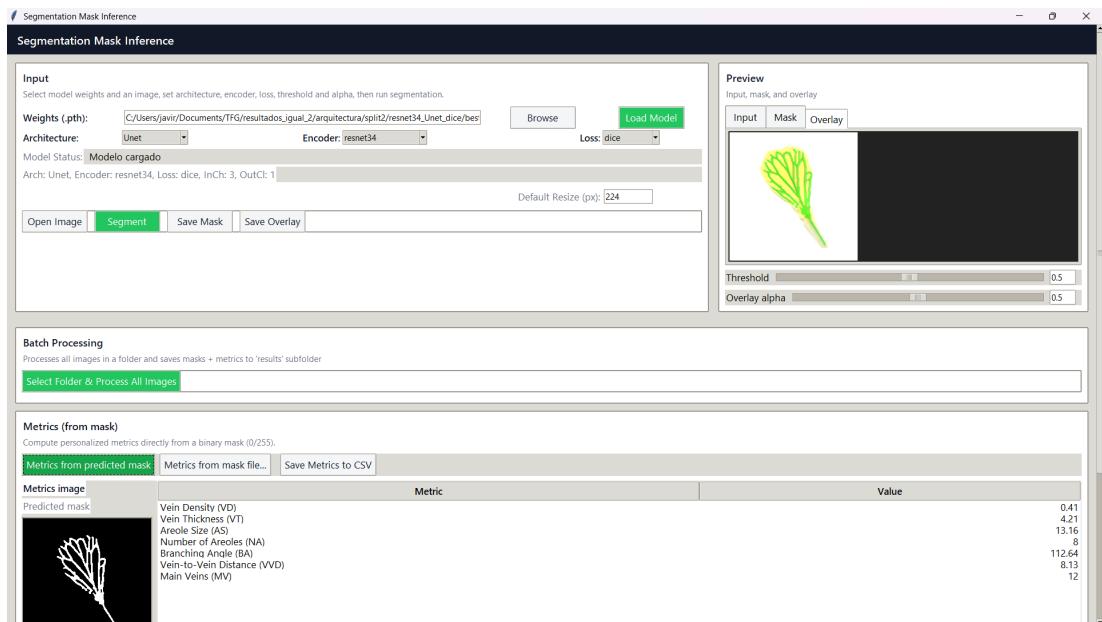


Figura 8.6: Ejemplo de la herramienta con el overlay de la imagen segmentada.

### 8.6.1 Estructura de módulos

El código se organiza en los siguientes módulos principales:

- `app.py`: punto de entrada de la aplicación.
- `main.py`: contiene la clase principal `SegTkApp` y la lógica que construye y coordina la interfaz gráfica y los distintos paneles.
- `models.py`: implementa el modelo de segmentación `CamVidModel` y funciones auxiliares para el preprocesamiento y postprocesamiento de imágenes.
- `metrics.py`: funciones para el cálculo y agregación de métricas de evaluación a partir de máscaras binarias y comparaciones con ground truth.
- `background_worker.py`: utilidades para ejecutar tareas en segundo plano, gestión de colas de mensajes y comunicación segura con la GUI para mantener la interfaz responsive.
- `ui_components.py`: componentes reutilizables de la interfaz para mantener el código de la GUI modular y legible.
- `theme.py`: gestión de estilos, paletas de color y configuración de `ttk.Style` para mantener una apariencia coherente en la aplicación.
- `petals_generator.py`: utilidades para generar recortes, máscaras o datos auxiliares relacionados con los pétalos.

### 8.6.2 Clase principal: `SegTkApp`

La clase `SegTkApp` (en `main.py`) constituye el núcleo de la herramienta. En su método `__init__` se construye la interfaz gráfica y se inicializan las variables de estado. La interfaz se organiza en paneles o *cards* que agrupan funcionalidades relacionadas: carga de modelo y parámetros, previsualización de resultados, cálculo de métricas, procesamiento por lotes y consola de depuración.

Cada panel se construye mediante métodos auxiliares (como `_build_input_controls` o `_build_preview_card`), lo que mejora la legibilidad y facilita la evolución del código.

### 8.6.3 Manejo de tareas en segundo plano

En aplicaciones interactivas resulta fundamental mantener la interfaz siempre receptiva, incluso cuando se ejecutan operaciones que requieren un tiempo de cómputo elevado. Si

estas tareas se realizaran directamente en el hilo principal, la interfaz quedaría bloqueada, impidiendo la interacción del usuario y transmitiendo una sensación de fallo o lentitud.

Para evitar este problema, se recurre a la ejecución en segundo plano, delegando los procesos costosos a hilos o subprocessos separados mientras la interfaz continúa funcionando con normalidad. Para evitar la creación ilimitada de hilos, se implementa una agrupación de hilos (*thread pool*) que limita el número máximo de hilos concurrentes, optimizando así el uso de recursos del sistema, concretamente reservando siempre un núcleo libre para evitar que el sistema quede bloqueado.

Con el fin de prevenir bloqueos durante operaciones intensivas —como la segmentación o el procesamiento por lotes— se desarrolló el módulo `background_worker`. Este módulo ofrece funciones como `run_callable_in_thread` y `run_subprocess`, que ejecutan tareas en hilos o subprocessos independientes y comunican los resultados mediante una cola de mensajes (`queue.Queue`), tal como se ilustra en la Figura 8.7.

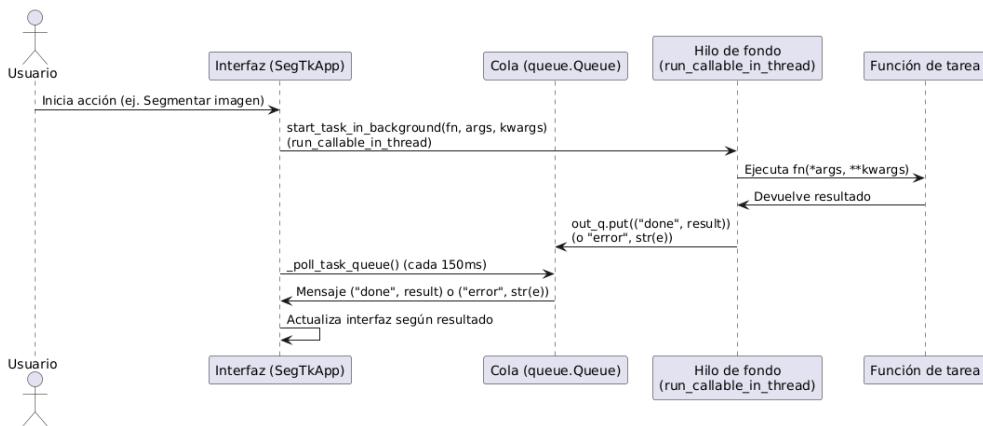


Figura 8.7: Diagrama de flujo de tareas en segundo plano y comunicación con la interfaz mediante colas.

#### 8.6.4 Segmentación y cálculo de métricas

La segmentación se lleva a cabo en el método `on_segment`, que lanza un hilo en segundo plano para ejecutar la función `_segment_task`. Esta función aplica el preprocesamiento a la imagen, ejecuta el modelo de segmentación, obtiene la máscara resultante y genera la superposición con la imagen original. Los resultados son enviados a la cola para su visualización en la interfaz.

El cálculo de métricas, en cambio, se ejecuta directamente en el hilo principal (métodos `on_metrics_from_pred` y `on_metrics_from_file`), ya que, en este caso, se trata de operaciones rápidas. No obstante, la arquitectura permite trasladar también estas operaciones a un hilo independiente si fuera necesario.

### 8.6.5 Actualización de la interfaz

Todas las modificaciones de la GUI se realizan desde el hilo principal, siguiendo las buenas prácticas de Tkinter. Cuando una tarea en segundo plano finaliza, envía un mensaje con los resultados a la cola, y el método `_poll_task_queue` se encarga de actualizar los elementos gráficos correspondientes (imágenes, tablas, barras de progreso, etc.).

Este enfoque garantiza la responsividad de la aplicación y mejora la experiencia de usuario.

### 8.6.6 Exportación de resultados

La exportación de resultados se implementa mediante los métodos `on_save_mask`, `on_save_overlay` y `on_save_metrics`. Estos permiten guardar tanto las imágenes procesadas (en formato PNG) como las métricas (en CSV), utilizando cuadros de diálogo nativos para la selección de ubicación.

En el caso del procesamiento por lotes, los resultados se almacenan automáticamente en una subcarpeta `results`, que incluye las máscaras y un fichero `metrics_summary.csv` con las métricas agregadas.

### 8.6.7 Resumen

En conjunto, la implementación de la herramienta sigue principios de diseño modular y responsividad. Las tareas intensivas se delegan a hilos en segundo plano, mientras que la interfaz se actualiza de forma asíncrona. Esta estrategia, unida a la organización clara en módulos, asegura la mantenibilidad, la extensibilidad y la usabilidad de la aplicación.

## 8.7 Exportación de resultados

La herramienta permite exportar tanto imágenes como ficheros tabulares. En el caso de las imágenes, se generan las máscaras binarias y las superposiciones con el pétalo original, en formatos estándar. En cuanto a los datos, las métricas pueden guardarse de forma individual en un archivo CSV, o bien de forma agregada cuando se procesa un lote de imágenes, en cuyo caso se genera un fichero `metrics_summary.csv` dentro de la carpeta `results/`. Este archivo incluye información relevante como el nombre de la imagen, los parámetros aplicados y las métricas calculadas.

En la Figura 8.8 se muestra un ejemplo de los archivos generados tras procesar un lote de imágenes, mientras que en la Figura 8.9 se observa el archivo CSV generado al exportar las métricas de una imagen individual.

```

image_name,mask_name,threshold,image_size,Vein Density (VD),Vein Thickness (VT),Areole Size (AS),Number of Areoles (NA),Branching Angle (BA),Vein-to-Vein Distance (VVD),Main Veins
rf11-2p.tif,rf11-2p_mask.png,0.5,446x318,0.5255368543238538,4,536749412132091,42406,46701651901,11,106.46913580246914,2,16507067526,17
rf12-0p.tif,rf12-0p_mask.png,0.5,1051x360,0,1344712430426716,2,3129348033996206,39868,95295259392,11,114.21428571428571,0.999385289941482,33
rf13-2p.tif,rf13-2p_mask.png,0.5,892x626,0,4324683965402528,2,735960974902533,20181,654293017866,23,113.58992805755396,1,162046055704658,28
rf14-1p.tif,rf14-1p_mask.png,0.5,961x923,0,224125487177017,2,3869946134630957,13360,130244863697,37,113.20975609756097,0.7692537858039448,38
rf15-0p.tif,rf15-0p_mask.png,0.5,1111x1213,0,5960025659824047,2,1745227244543264,16465,892249219685,30,114.18848167539267,0.6261785786508065,25
rf16-1p.tif,rf16-1p_mask.png,0.5,982x939,0,5630046948356807,2,4235065186780496,17054,913575255556,29,113.93877551020408,0.7581927773501859,26
rf18-0p.tif,rf18-0p_mask.png,0.5,1023x705,0,29259917238264886,2,2472759223086145,19153,1739954241,24,112.97142857142858,0.6733107867795494,59
rf21-0p.tif,rf21-0p_mask.png,0.5,903x914,0,5625804689074631,2,4317516341537043,13089,655619277615,38,111.48780487804878,0.617245624142591,26
rf22-1p.tif,rf22-1p_mask.png,0.5,1946x754,0,4568478599950581,2,57495751008204644,15154,7053948974,31,111.07826086956521,0.8534792579774306,31
rf23-0p.tif,rf23-0p_mask.png,0.5,1114x864,0,4679953106682298,2,461338545504512,21784,009925345843,22,115.4296875,0.7412481045494304,30
rf25-0p.tif,rf25-0p_mask.png,0.5,966x783,0,491124260355028,2,4141426583287374,20461,997548263384,23,112.40625,0.8579388684008534,34
rf26-2p.tif,rf26-2p_mask.png,0.5,886x818,0,5245401003417436,2,4013021442628952,12335,433141485898,40,114.27312775330397,0.9880584896792408,24
rf28-0p.tif,rf28-0p_mask.png,0.5,1046x737,0,4340772396327952,2,43485880827878397,19643,5761756708097,24,112.91379310344827,0.7957666515527503,31
rf44-0p.tif,rf44-0p_mask.png,0.5,1051x154,0,4581364181420611,2,485189969659913,93948,60012778544,5,112.39106145251397,0.8194882091332489,33
rf51-2p.tif,rf51-2p_mask.png,0.5,506x383,0,48280836739600863,3,4560760458736297,19835,702609307424,24,108.47540983606558,1.9117336284173494,11
rf510-1p.tif,rf510-1p_mask.png,0.5,1818x732,0,52991706113744,2,309904710571746,16724,138839793838,28,112.9042316258352,0.8414062736497754,48
rf52-1p.tif,rf52-1p_mask.png,0.5,1189x966,2,4641094911962972,19598,888044919827,25,112.60310421286032,0.7105331033311353,36
rf53-0p.tif,rf53-0p_mask.png,0.5,1229x552,0,3918109830638062,2,1314174423978907,14568,424657064203,28,113.925,0.7403474558531795,43
rf54-2p.tif,rf54-2p_mask.png,0.5,913x855,0,8107982096232749,2,4165036416499880,15914,176972717274,31,110.93793103448276,0.831443053175914,34
rf56-0p.tif,rf56-0p_mask.png,0.5,921x693,0,43562830605189723,2,3066717255347473,31582,904591528733,15,113.71246068389777,1.002756797446927,55
rf57-0p.tif,rf57-0p_mask.png,0.5,1274x750,0,4045647149460709,2,11201976498998,17974,914469534197,24,112.76210526315789,0.6764209398241525,60
rf58-1p.tif,rf58-1p_mask.png,0.5,1880x585,0,54241173734949,2,378911465712104,2,3791436161381146,19294,153258471204,23,111.7126625689933,1.176496820630995,21
rf59-0p.tif,rf59-0p_mask.png,0.5,525x342,0,4629672099355624,3,4830494295711594,28257,127683149563,16,108.49569707401034,1.902294486946174,23
rf71-2p.tif,rf71-2p_mask.png,0.5,913x874,0,547704646176991,2,4641094911962972,14545,377885605198,34,112.17994017094017,0.8188934410322953,18
rf72-0p.tif,rf72-0p_mask.png,0.5,451x288,0,4698950469526791,4,815332796207024,24696,518231441873,16,108.84782608695652,2,310134301288684,21
rf73-2p.tif,rf73-2p_mask.png,0.5,953x726,0,5029815966958371,2,539705888655302,22750,05279356331,21,113.20295209252029,0.901812639504928,22
rf74-2p.tif,rf74-2p_mask.png,0.5,882x657,0,42372147671758087,2,5367450528524023,17532,908096268011,27,112.64383561643835,0.9965027501370165,24
rf75-0p.tif,rf75-0p_mask.png,0.5,1002x626,0,3779114165712104,2,3791436161381146,19294,153258471204,23,111.7126625689933,1.176496820630995,21
rf76-0p.tif,rf76-0p_mask.png,0.5,1138x487,0,36353591160220994,2,281095328111916,14243,972606941727,25,111.93577981651376,0.8886548465167178,33
rf77-1p.tif,rf77-1p_mask.png,0.5,1138x705,0,5123758814217874,2,31790.31969507373,16,112.35714285714285,0.8258954205149989,35
rf78-2p.tif,rf78-2p_mask.png,0.5,862x626,0,538298623367455,2,69561542867887,16786,8660346103688,28,109.6376811594203,1.085644577157274,24
rf79-0p.tif,rf79-0p_mask.png,0.5,1051x914,0,49375382731169626,2,474447104258849,22317,303575529944,22,113.27659574468085,0.7251743792054,26

```

Figura 8.8: Ejemplo de archivo generado tras procesar un lote de imágenes.

```

timestamp,image_name,Vein Density (VD),Vein Thickness (VT),Areole Size (AS),Number of Areoles (NA),Branching Angle (BA),Vein-to-Vein Distance (VVD),Main Veins (MV)
2025-09-09 18:15:46,Predicted mask,0.7509065550906555,6.426644225659938,25754.158704765287,18,110.32369942196532,2,163359882960355,6

```

Figura 8.9: Ejemplo de archivo generado al exportar las métricas de una imagen individual.

## 8.8 Pruebas

Para garantizar la fiabilidad de la herramienta, se llevaron a cabo pruebas en diferentes niveles. En primer lugar, se realizaron **pruebas unitarias** de los módulos principales —carga de modelos, inferencia, cálculo de métricas y exportación— con el fin de validar que cada componente funcionaba de forma independiente. Posteriormente, se efectuaron **pruebas de integración**, verificando la correcta comunicación entre los distintos módulos. Finalmente, se llevaron a cabo **pruebas de aceptación** junto con los tutores del proyecto, quienes confirmaron que la herramienta cumplía con los objetivos planteados y respondía a los requisitos definidos en la fase de análisis.

## 8.9 Resolución de limitaciones

Durante la implementación de la aplicación se identificaron varios problemas relacionados tanto con la ejecución concurrente de tareas como con el rendimiento en máquinas con recursos limitados. A continuación se describen los más relevantes y las estrategias empleadas para solucionarlos.

### 8.9.1 Concurrencia durante la inferencia

Al ejecutar la inferencia sobre imágenes de forma secuencial, se detectaron bloqueos y retrasos en la interfaz, especialmente con imágenes de gran resolución o múltiples procesos de cálculo de métricas. Esto se debía a que las operaciones de inferencia y cálculo son computacionalmente intensivas y, si se ejecutan en el hilo principal, bloquean la GUI.

La solución consistió en ejecutar estas tareas en hilos separados y comunicar su estado mediante colas (`queue`) hacia la interfaz. De esta manera, la GUI permanece reactiva y se pueden mostrar mensajes de progreso y resultados en tiempo real. Para lograrlo, se implementaron dos funciones clave:

- Función que permite ejecutar cualquier tarea en un hilo independiente y recibir mensajes de estado durante su ejecución.
- Función destinada a lanzar subprocessos de alta carga computacional (entrenamiento o inferencia), garantizando que la interfaz no se bloquee.

### 8.9.2 Rendimiento en máquinas con recursos limitados

En máquinas antiguas o con poca memoria y CPU, la inferencia de modelos grandes podía provocar que la aplicación o incluso todo el sistema se “congelara”. Esto ocurre cuando la RAM se llena o la CPU se sobrecarga. Para mitigarlo se implementaron:

- Todas las imágenes de entrada se redimensionan previamente a un tamaño fijo, de manera que el modelo procese siempre entradas homogéneas. Esto reduce el consumo de memoria y garantiza que el tiempo de inferencia sea estable independientemente de la resolución original de las imágenes.
- Se utiliza una agrupación de hilos (*thread pool*) para controlar el grado de concurrencia. El número máximo de hilos se establece como el número de núcleos del procesador menos uno (con un mínimo de un hilo), reservando siempre un núcleo libre para evitar que el sistema quede bloqueado.

### 8.9.3 Problemas menores

Durante la implementación también se identificaron problemas menos críticos pero importantes de resolver:

- **Exportación de métricas:** se detectaron errores al generar CSV si las rutas contenían caracteres especiales. Se solucionó normalizando los nombres de archivo y rutas.

- **Visualización del overlay:** al cambiar el *alpha* mientras se procesaba la imagen, podía producirse un parpadeo en la interfaz. Se resolvió actualizando la imagen únicamente al completar la inferencia.
- **Carga de modelos incorrectos:** se considera incorrecta cuando, al intentar cargar los pesos, se produce una excepción por incompatibilidad entre la arquitectura, el *encoder* o el número de clases definidos y los almacenados en el archivo. En estos casos, la aplicación detecta el error y muestra un mensaje claro al usuario indicando que los pesos no son compatibles.

Estas soluciones permiten que la aplicación funcione de manera estable y eficiente, incluso en condiciones adversas o en equipos con recursos limitados.

## 8.10 Conclusiones

La herramienta desarrollada automatiza el pipeline de segmentación y caracterización de venación, integrando en una interfaz intuitiva todas las funcionalidades necesarias: carga de modelos, inferencia, visualización, cálculo de métricas y procesamiento por lotes, junto con la exportación de resultados en formatos adecuados para análisis y documentación. De este modo, se facilita la reproducibilidad y la trazabilidad del trabajo realizado, a la vez que se sientan las bases para futuras mejoras y extensiones.

## Capítulo 9

# Conclusiones y Trabajo Futuro

---

### 9.1 Conclusiones generales

El presente Trabajo de Fin de Grado ha abordado el problema de la extracción y caracterización de venas en pétalos de *Raphanus raphanistrum*, con el objetivo de analizar posibles diferencias morfológicas derivadas de condiciones de estrés salino.

Los principales logros alcanzados se pueden resumir en los siguientes puntos:

- Se desarrolló un **pipeline de segmentación** basado en arquitecturas de aprendizaje profundo, siendo la combinación de arquitectura **U-Net**, encoder **resnet34** y función de perdida **Dice** la que ofreció el mejor equilibrio entre rendimiento y coste computacional.
- Se obtuvieron **esqueletos de las máscaras venosas**, que permitieron definir métricas cuantitativas para caracterizar la red vascular.
- El análisis estadístico (ANOVA y test de Tukey HSD) mostró que el **estrés salino afecta de manera significativa la morfología de la red vascular**. En particular:
  - La **densidad de venas (VD)** disminuye significativamente bajo *estrés fuerte*, tanto respecto al grupo *Control* como al de *estrés leve*.
  - El **tamaño de aréolas (AS)** aumenta bajo condiciones de estrés, siendo mayor en los tratamientos salinos que en el grupo *Control*.
  - El **ángulo de ramificación (BA)** también se incrementa en el tratamiento de *estrés fuerte*, indicando una red vascular con ramificaciones más abiertas.

En conjunto, estos resultados reflejan una **red venosa más laxa y menos densa** bajo estrés salino intenso, lo que podría representar una adaptación morfológica para optimizar el flujo y la distribución de recursos en condiciones adversas.

- El resto de las métricas (**número de aréolas, distancia vena a vena y número de venas principales**) no mostraron diferencias significativas, lo que sugiere que los cambios se concentran principalmente en la red de venas secundarias.
- Se implementó una **herramienta software** que automatiza el flujo completo: carga de imágenes, predicción de máscaras, ajuste de umbrales y exportación de resultados en formato tabular, tanto para imágenes individuales como en lotes.

No obstante, es importante señalar que, aunque los resultados estadísticos apuntan a diferencias entre tratamientos, **no se puede afirmar con total certeza que estas diferencias se reproduzcan sistemáticamente**.

El conjunto de datos presenta ciertas limitaciones que pueden haber influido en los valores obtenidos:

- Aunque el conjunto incluye algo más de **200 imágenes**, sigue siendo un **dataset relativamente pequeño** tanto para el entrenamiento de modelos de segmentación profunda como para extraer conclusiones estadísticas sólidas sobre las diferencias entre niveles de estrés salino. Un número reducido de muestras puede limitar la capacidad del modelo para generalizar y aumentar la variabilidad de los resultados.
- Algunas imágenes muestran **pétalos doblados, dañados o con iluminación irregular**, lo que puede afectar tanto la segmentación como la extracción de esqueletos.
- Las máscaras de referencia fueron **etiquetadas manualmente por un único operador**, lo que introduce un sesgo subjetivo en la delimitación de las venas.
- Los algoritmos de procesamiento, como `skeletonize` de Python, pueden **introducir errores topológicos** en zonas finas o mal segmentadas. Se puede ver en la Figura 9.1 como crea nuevas aureolas que no existen en la realidad.
- Dado que las venas representan estructuras con **pocos píxeles respecto al total de la imagen**, una pequeña variación en la segmentación puede provocar **grandes diferencias relativas** en las métricas derivadas (por ejemplo, en VD o AS).

Estas fuentes de error y variabilidad deben tenerse en cuenta al interpretar los resultados, ya que podrían enmascarar o amplificar las diferencias reales entre tratamientos.

A pesar de los resultados obtenidos, el trabajo presenta una serie de limitaciones:

- **Desbalanceo de clases** en la segmentación, lo que dificulta la detección de venas muy finas.
- **Errores en la interpretación de ramificaciones**, ya que el método actual contabiliza tanto divisiones reales como fusiones (anastomosis) como si fueran ramificaciones.



Figura 9.1: Ejemplo de errores en el esqueleto generado a partir de una máscara

- **Falta de validación biológica directa**, ya que las métricas extraídas aún no se han correlacionado con parámetros fisiológicos (ej. eficiencia hídrica, fotosíntesis).
- **Variabilidad natural** entre muestras que puede haber enmascarado posibles efectos sutiles del estrés salino.

## 9.2 Conclusión final

En definitiva, este TFG demuestra que la combinación de técnicas de segmentación profunda con el análisis cuantitativo de esqueletos constituye una estrategia eficaz para caracterizar patrones de venación en pétalos.

Los resultados obtenidos confirman que el **estrés salino, especialmente en su nivel más intenso, induce alteraciones estructurales en la red venosa**, reduciendo su densidad y ampliando tanto el tamaño de las aréolas como el ángulo de ramificación.

Sin embargo, debido a las limitaciones del conjunto de datos y del proceso de segmentación, estos resultados deben interpretarse con **precaución**, ya que podrían verse afectados por factores técnicos y no biológicos.

Aun así, el trabajo establece un marco de análisis reproducible y extensible, demostrando el potencial de los métodos de visión por computador y aprendizaje profundo en el estudio de la morfología vegetal bajo estrés ambiental. Este enfoque sienta las bases para estudios posteriores que busquen confirmar estas tendencias con un mayor número de muestras, mejor calidad de datos y validación biológica complementaria.

### 9.3 Trabajo futuro

De cara a proyectos futuros, se abren distintas líneas de investigación y mejora:

- **Ampliación del conjunto de datos:** incrementar el número de muestras es una prioridad para mejorar la robustez tanto del entrenamiento del modelo de segmentación como de los análisis estadísticos. Un **dataset más amplio y equilibrado** permitiría reducir el sesgo causado por la variabilidad entre pétalos (estado físico, iluminación, orientación) y facilitaría una evaluación más fiable de las diferencias entre niveles de estrés salino. Además, una mayor cantidad de imágenes contribuiría a un mejor ajuste de los modelos de aprendizaje profundo y a una menor dependencia de la *data augmentation*.
- **Mejoras en la segmentación:** evaluar arquitecturas más recientes como *Transformers* para visión o nnU-Net, explorar el uso de modelos como *Segment Anything Model (SAM)* para la inicialización de máscaras y anotación semiautomática, y ampliar la base de datos con muestras más diversas junto con técnicas de *data augmentation* avanzadas.
- **Ampliación de métricas de venación:** incorporar métricas topológicas basadas en grafos, como la conectividad global, la longitud media de ciclos o la centralidad de nodos; además de desarrollar métodos para diferenciar entre **ramificaciones** y **anastomosis**. Actualmente, ambas se contabilizan como ramificaciones, lo que puede enmascarar diferencias estructurales significativas. La **anastomosis** representa un reto abierto. Se trata de fusiones de venas que aparecen típicamente hacia los extremos del pétalo, formando estructuras que rodean el margen. A diferencia de las ramificaciones, que parten del eje basal y se orientan distalmente, las anastomosis suelen ocurrir con ángulos cercanos a los 90° y tienden a cerrar espacios. Su detección automática plantea dificultades tanto morfológicas como topológicas, ya que los esqueletos actuales no distinguen entre bifurcaciones y fusiones. Se pueden observar las diferencias en la Figura 9.2.
- **Validación biológica:** colaborar con especialistas en fisiología vegetal para vincular métricas venosas con parámetros fisiológicos (eficiencia en el transporte, adaptaciones al estrés), así como realizar experimentos adicionales con otros tipos de estrés abiótico (sequía, temperatura) donde la respuesta morfológica pueda ser más evidente. También podrían ayudar estos expertos con el etiquetado de una dataset más amplio y fiable, de forma que nos aseguraríamos de tener un buen ground truth.
- **Mejoras en la herramienta:** añadir una interfaz gráfica que permita a usuarios no expertos ajustar parámetros de forma visual, incluir un sistema de reportes automáticos

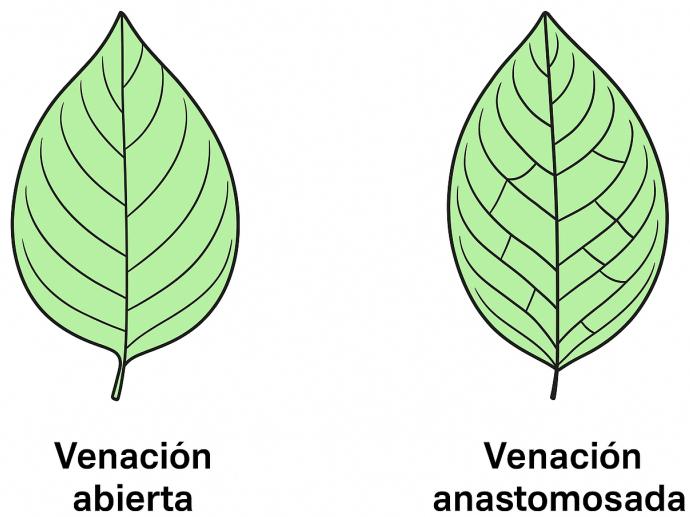


Figura 9.2: Ejemplo esquemático de anastomosis.

con gráficos, tablas y métricas resumidas, y mejorar la compatibilidad para el análisis en lotes y la exportación de datos a formatos estándar de análisis estadístico (ej. R, SPSS).

# **Apéndices**

## Apéndice A

# Resultados de entrenamiento

---

Resultados de los entrenamientos de los modelos de segmentación, incluyendo métricas por combinación de arquitectura, encoder y fold.

En la Tabla A.1 se muestran las métricas de evaluación para las diferentes arquitecturas.  
En la Tabla A.2 se presentan las métricas para los diferentes encoders.  
En la Tabla A.3 se detallan las métricas de pérdida.

Tabla A.1: Métricas de entrenamiento para arquitectura por fold.

Fold	Arquitectura	Encoder	Loss	Test Loss	IoU	Precisión	Recall	F1
fold1	DeepLabV3	resnet34	dice	0.8903	0.5116	0.5528	0.8721	0.6751
fold1	FPN	resnet34	dice	0.8888	0.6327	0.6847	0.8930	0.7744
fold1	PSPNet	resnet34	dice	0.8932	0.4817	0.5631	0.7725	0.6485
fold1	Unet	resnet34	dice	0.8914	0.6695	0.7986	0.8043	0.8007
fold2	DeepLabV3	resnet34	dice	0.8826	0.5113	0.5473	0.8843	0.6753
fold2	FPN	resnet34	dice	0.8828	0.6308	0.7060	0.8560	0.7731
fold2	PSPNet	resnet34	dice	0.8819	0.5041	0.5314	0.9059	0.6688
fold2	Unet	resnet34	dice	0.8821	0.7062	0.7963	0.8617	0.8271
fold3	DeepLabV3	resnet34	dice	0.8831	0.5156	0.5546	0.8794	0.6790
fold3	FPN	resnet34	dice	0.8819	0.6300	0.6842	0.8886	0.7723
fold3	PSPNet	resnet34	dice	0.8822	0.5126	0.5417	0.9058	0.6764
fold3	Unet	resnet34	dice	0.8824	0.7053	0.7948	0.8609	0.8260

Tabla A.2: Métricas de entrenamiento para encoders por fold.

Fold	Arquitectura	Encoder	Loss	Test Loss	IoU	Precisión	Recall	F1
fold1	Unet	efficientnet-b0	dice	0.8904	0.6864	0.7934	0.8348	0.8127
fold1	Unet	mobilenet_v2	dice	0.8892	0.7203	0.8136	0.8636	0.8370
fold1	Unet	resnet34	dice	0.8903	0.6887	0.7964	0.8348	0.8146
fold1	Unet	resnet50	dice	0.8905	0.6833	0.7947	0.8297	0.8109
fold2	Unet	efficientnet-b0	dice	0.8818	0.7408	0.8347	0.8680	0.8503
fold2	Unet	mobilenet_v2	dice	0.8822	0.7071	0.7995	0.8588	0.8275
fold2	Unet	resnet34	dice	0.8813	0.7245	0.7998	0.8844	0.8395
fold2	Unet	resnet50	dice	0.8823	0.7054	0.7976	0.8587	0.8265
fold3	Unet	efficientnet-b0	dice	0.8812	0.7130	0.7794	0.8926	0.8315
fold3	Unet	mobilenet_v2	dice	0.8822	0.7112	0.7998	0.8643	0.8304
fold3	Unet	resnet34	dice	0.8831	0.7009	0.8059	0.8421	0.8231
fold3	Unet	resnet50	dice	0.8809	0.7219	0.7841	0.9008	0.8376

## APÉNDICE A. RESULTADOS DE ENTRENAMIENTO

---

Tabla A.3: Métricas de entrenamiento para loss por fold.

Fold	Arquitectura	Encoder	Loss	Test Loss	IoU	Precisión	Recall	F1
fold1	Unet	resnet34	bce	0.0366	0.7061	0.8399	0.8169	0.8272
fold1	Unet	resnet34	bce dice	0.7870	0.7182	0.7953	0.8813	0.8352
fold1	Unet	resnet34	dice	0.8890	0.7122	0.7951	0.8716	0.8305
fold1	Unet	resnet34	focal	0.1932	0.6827	0.8035	0.8203	0.8108
fold2	Unet	resnet34	bce	0.0421	0.6984	0.8104	0.8342	0.8216
fold2	Unet	resnet34	bce dice	0.7837	0.6986	0.7997	0.8465	0.8218
fold2	Unet	resnet34	dice	0.8815	0.7495	0.8393	0.8747	0.8561
fold2	Unet	resnet34	focal	0.1981	0.6907	0.8294	0.8051	0.8154
fold3	Unet	resnet34	bce	0.0418	0.6988	0.8183	0.8259	0.8215
fold3	Unet	resnet34	bce dice	0.7839	0.7001	0.8009	0.8467	0.8227
fold3	Unet	resnet34	dice	0.8827	0.7016	0.8012	0.8491	0.8237
fold3	Unet	resnet34	focal	0.1964	0.6679	0.7914	0.8103	0.7999

## Apéndice B

# Métricas de pétalos

---

Los resultados de las métricas aplicadas sobre las predicciones del modelo con arquitectura UNet, encoder resnet34 y función de pérdida Dice se muestran desglosadas para cada imagen en la Tabla B.1

En la Tabla B.2 se muestran los resultados de las métricas aplicadas sobre los groundtruth.

Tabla B.1: Métricas de venación por imagen.

Imagen	Umbral	Tamaño	VD	VT	AS	NA	BA	VVD	MV
rf12-0p_region1.png	0.5	224x224	0.414	3.915	15.999	9	113.684	7.489	6
rf12-0p_region2.png	0.5	224x224	0.406	4.173	17.371	4	113.544	8.609	13
rf12-0p_region3.png	0.5	224x224	0.429	4.185	13.836	11	113.802	8.883	11
rf12-0p_region4.png	0.5	224x224	0.388	4.018	16.138	7	114.162	10.500	16
rf13-2p_region1.png	0.5	224x224	0.450	4.266	18.922	4	113.390	7.556	9
rf13-2p_region2.png	0.5	224x224	0.449	4.379	17.980	5	117.736	9.052	7
rf13-2p_region3.png	0.5	224x224	0.466	4.228	14.351	4	115.091	8.492	8
rf13-2p_region4.png	0.5	224x224	0.479	4.312	14.388	5	114.828	9.101	6
rf14-1p_region1.png	0.5	224x224	0.508	4.384	14.870	7	111.417	7.909	5
rf14-1p_region2.png	0.5	224x224	0.490	4.114	12.734	6	113.647	7.511	8
rf14-1p_region6.png	0.5	224x224	0.504	4.377	9.449	2	111.846	8.223	16
rf14-1p_region7.png	0.5	224x224	0.468	4.059	15.686	3	114.407	6.830	12
rf15-0p_region1.png	0.5	224x224	0.422	4.194	17.773	7	113.842	8.746	9
rf15-0p_region2.png	0.5	224x224	0.412	4.157	18.129	7	114.814	9.417	5
rf15-0p_region3.png	0.5	224x224	0.441	4.207	17.385	7	115.055	9.254	6
rf15-0p_region4.png	0.5	224x224	0.447	4.059	16.454	8	112.297	8.853	9
rf16-1p_region1.png	0.5	224x224	0.478	4.236	14.451	5	115.116	9.043	4
rf16-1p_region2.png	0.5	224x224	0.530	4.273	13.631	9	113.229	7.796	4
rf16-1p_region3.png	0.5	224x224	0.511	4.470	15.612	6	114.353	10.185	5
rf16-1p_region4.png	0.5	224x224	0.527	4.710	18.284	4	114.706	8.446	10
rf18-0p_region1.png	0.5	224x224	0.368	4.161	27.441	4	115.769	10.044	15
rf18-0p_region2.png	0.5	224x224	0.367	4.150	21.987	7	112.800	8.921	6
rf18-0p_region3.png	0.5	224x224	0.387	4.198	21.924	7	114.490	9.207	12
rf18-0p_region4.png	0.5	224x224	0.394	4.151	22.898	6	112.121	7.534	11
rf21-1p_region1.png	0.5	224x224	0.532	4.535	12.424	8	113.077	7.395	9
rf21-1p_region2.png	0.5	224x224	0.551	4.423	12.028	4	116.341	7.080	5
rf21-1p_region4.png	0.5	224x224	0.558	4.231	12.486	8	114.247	7.182	5
rf21-1p_region5.png	0.5	224x224	0.541	4.421	12.878	5	115.385	7.532	9
rf22-1p_region1.png	0.5	224x224	0.493	4.346	10.667	8	114.655	7.122	11
rf22-1p_region2.png	0.5	224x224	0.519	4.278	10.781	9	113.712	7.556	9
rf22-1p_region3.png	0.5	224x224	0.498	4.254	13.390	6	111.876	7.648	11

.....(continúa en la página siguiente).....

APÉNDICE A. RESULTADOS DE ENTRENAMIENTO

---

Tabla B.1 – (*viene de la página anterior*)

Imagen	Umbral	Tamaño	VD	VT	AS	NA	BA	VVD	MV
rf23-0p_region1.png	0.5	224x224	0.421	4.375	21.110	6	113.731	9.423	5
rf23-0p_region2.png	0.5	224x224	0.437	4.694	19.296	3	116.667	9.657	8
rf23-0p_region3.png	0.5	224x224	0.460	4.554	17.290	4	116.719	9.242	7
rf23-0p_region4.png	0.5	224x224	0.425	4.419	17.993	7	115.636	9.611	4
rf25-1p_region1.png	0.5	224x224	0.569	4.759	8.336	6	113.284	7.205	8
rf25-1p_region2.png	0.5	224x224	0.507	4.602	13.845	4	115.227	9.054	6
rf25-1p_region3.png	0.5	224x224	0.455	4.304	10.388	2	111.207	8.080	13
rf25-1p_region4.png	0.5	224x224	0.531	4.308	10.806	5	113.478	7.377	8
rf26-2p_region1.png	0.5	224x224	0.599	4.747	9.432	11	112.424	7.538	6
rf26-2p_region2.png	0.5	224x224	0.577	4.344	9.574	8	113.125	6.765	7
rf26-2p_region3.png	0.5	224x224	0.570	4.695	11.091	7	115.588	7.933	8
rf26-2p_region4.png	0.5	224x224	0.577	4.487	10.213	7	114.512	6.259	7
rf28-0p_region1.png	0.5	224x224	0.414	4.352	20.903	5	113.913	9.123	8
rf28-0p_region2.png	0.5	224x224	0.431	4.290	19.386	3	116.038	9.912	9
rf28-0p_region3.png	0.5	224x224	0.390	4.227	16.015	7	114.474	8.556	9
rf28-0p_region4.png	0.5	224x224	0.412	4.128	23.398	5	115.976	9.028	11
rf44-0p_region1.png	0.5	224x224	0.435	4.508	0.0	0	113.529	9.965	12
rf44-0p_region2.png	0.5	224x224	0.447	4.586	24.773	1	113.385	9.923	10
rf44-0p_region3.png	0.5	224x224	0.432	4.302	3.909	1	113.559	11.348	11
rf51-2p_region1.png	0.5	224x224	0.812	6.481	4.866	3	117.857	8.021	1
rf51-2p_region2.png	0.5	224x224	0.845	7.940	5.528	1	117.000	8.788	1
rf51-2p_region3.png	0.5	224x224	0.816	6.887	4.517	2	114.000	8.403	1
rf51-2p_region4.png	0.5	224x224	0.729	6.367	6.346	2	116.471	8.552	1
rf510-1p_region1.png	0.5	224x224	0.521	4.278	13.871	4	113.333	7.465	15
rf510-1p_region2.png	0.5	224x224	0.524	4.239	13.524	5	113.762	6.778	14
rf510-1p_region3.png	0.5	224x224	0.547	4.414	19.210	4	112.875	7.514	13
rf510-1p_region4.png	0.5	224x224	0.566	4.371	9.838	3	113.913	8.624	14
rf52-1p_region1.png	0.5	224x224	0.437	4.267	18.854	5	116.471	8.083	10
rf52-1p_region2.png	0.5	224x224	0.475	4.519	16.105	4	112.247	8.659	8
rf52-1p_region3.png	0.5	224x224	0.423	4.319	9.400	2	112.274	9.408	17
rf52-1p_region4.png	0.5	224x224	0.437	4.539	15.989	4	114.500	10.204	8
rf53-0p_region1.png	0.5	224x224	0.386	4.191	20.083	7	111.818	8.979	9
rf53-0p_region2.png	0.5	224x224	0.384	3.989	17.769	9	113.250	9.854	3
rf53-0p_region3.png	0.5	224x224	0.395	4.089	23.423	5	114.400	10.298	13
rf53-0p_region4.png	0.5	224x224	0.372	4.151	16.188	8	115.823	10.514	8
rf54-2p_region1.png	0.5	224x224	0.903	4.246	14.632	7	113.168	8.054	5
rf54-2p_region2.png	0.5	224x224	0.618	4.336	12.460	4	111.944	8.366	9
rf54-2p_region3.png	0.5	224x224	0.935	4.126	12.501	10	114.109	7.108	9
rf54-2p_region4.png	0.5	224x224	0.827	4.267	15.911	5	114.783	7.802	7
rf56-0p_region1.png	0.5	224x224	0.445	4.003	16.384	7	113.832	7.641	15
rf56-0p_region2.png	0.5	224x224	0.423	4.345	7.047	1	112.105	9.401	13
rf56-0p_region3.png	0.5	224x224	0.451	4.218	22.889	2	114.211	8.333	8
rf56-0p_region4.png	0.5	224x224	0.469	4.236	18.054	8	115.234	9.086	12
rf57-0p_region1.png	0.5	224x224	0.412	4.079	20.105	4	115.443	9.536	19
rf57-0p_region2.png	0.5	224x224	0.411	4.061	23.129	5	112.692	9.021	11
rf57-0p_region3.png	0.5	224x224	0.407	4.183	20.778	6	115.862	8.280	16
rf57-0p_region4.png	0.5	224x224	0.399	3.875	19.649	9	113.436	9.000	14
rf58-1p_region1.png	0.5	224x224	0.513	4.382	11.167	7	114.000	8.073	14
rf58-1p_region4.png	0.5	224x224	0.480	4.126	11.933	8	115.062	7.651	13
rf58-1p_region5.png	0.5	224x224	0.517	4.233	10.822	11	114.286	6.994	9
rf58-1p_region6.png	0.5	224x224	0.527	4.091	13.003	10	114.469	7.205	14
rf59-0p_region3.png	0.5	224x224	0.747	5.710	4.992	4	111.579	10.276	1
rf59-0p_region4.png	0.5	224x224	0.873	7.944	0.0	0	104.118	13.172	1
rf71-2p_region1.png	0.5	224x224	0.520	4.601	13.917	5	114.462	8.295	7
rf71-2p_region5.png	0.5	224x224	0.502	4.501	14.581	5	115.333	8.189	4
rf71-2p_region6.png	0.5	224x224	0.504	4.590	13.105	6	113.342	8.659	6

..... (continúa en la página siguiente) .....

APÉNDICE A. RESULTADOS DE ENTRENAMIENTO

---

Tabla B.1 – (*viene de la página anterior*)

Imagen	Umbral	Tamaño	VD	VT	AS	NA	BA	VVD	MV
rf71-2p_region7.png	0.5	224x224	0.570	4.429	11.465	6	113.600	6.934	4
rf72-0p_region1.png	0.5	224x224	0.755	5.518	6.561	4	112.174	7.077	1
rf72-0p_region2.png	0.5	224x224	0.721	5.463	6.820	5	109.833	7.944	2
rf72-0p_region3.png	0.5	224x224	0.691	5.851	9.093	2	106.154	9.007	3
rf72-0p_region4.png	0.5	224x224	0.714	5.286	7.851	3	113.684	8.020	2
rf73-2p_region1.png	0.5	224x224	0.543	4.461	16.276	5	112.615	8.592	3
rf73-2p_region2.png	0.5	224x224	0.518	4.224	14.710	6	114.677	9.000	3
rf73-2p_region3.png	0.5	224x224	0.457	4.172	16.741	2	114.286	12.219	14
rf74-2p_region1.png	0.5	224x224	0.490	4.290	12.940	3	113.478	8.639	8
rf74-2p_region3.png	0.5	224x224	0.484	4.377	16.038	6	114.677	8.971	5
rf74-2p_region4.png	0.5	224x224	0.468	4.278	16.083	5	115.385	8.554	5
rf74-2p_region5.png	0.5	224x224	0.466	4.180	16.144	6	115.556	8.939	5
rf75-0p_region1.png	0.5	224x224	0.449	4.457	15.435	8	113.333	9.796	1
rf75-0p_region2.png	0.5	224x224	0.412	4.308	16.341	7	114.000	10.216	5
rf75-0p_region3.png	0.5	224x224	0.398	4.387	21.801	3	114.615	11.661	6
rf75-0p_region4.png	0.5	224x224	0.385	4.260	24.553	4	115.227	11.042	6
rf76-0p_region1.png	0.5	224x224	0.386	4.171	20.250	4	117.600	8.854	9
rf76-0p_region2.png	0.5	224x224	0.378	4.328	18.017	6	114.107	10.146	3
rf76-0p_region3.png	0.5	224x224	0.405	4.308	16.293	6	114.400	9.062	7
rf76-0p_region4.png	0.5	224x224	0.389	4.283	14.057	6	113.333	9.535	9
rf77-1p_region1.png	0.5	224x224	0.470	4.554	12.481	3	114.600	8.226	5
rf77-1p_region2.png	0.5	224x224	0.455	4.671	13.535	3	114.667	8.865	7
rf77-1p_region3.png	0.5	224x224	0.521	4.874	0.0	0	112.500	8.488	11
rf77-1p_region4.png	0.5	224x224	0.555	4.760	3.909	1	111.134	8.166	9
rf78-2p_region1.png	0.5	224x224	0.639	5.267	11.268	3	116.270	8.634	5
rf78-2p_region2.png	0.5	224x224	0.542	4.434	11.046	5	112.895	8.150	11
rf78-2p_region3.png	0.5	224x224	0.594	4.566	10.991	6	115.932	6.614	7
rf78-2p_region4.png	0.5	224x224	0.592	4.793	10.974	5	114.295	7.944	5
rf79-1p_region1.png	0.5	224x224	0.466	4.647	12.619	5	113.750	9.485	6
rf79-1p_region2.png	0.5	224x224	0.462	4.728	5.863	1	113.667	9.311	10
rf79-1p_region5.png	0.5	224x224	0.461	4.340	12.305	6	114.000	8.198	5
rf79-1p_region6.png	0.5	224x224	0.458	4.434	16.574	3	115.610	9.022	9
rs21-0p_region1.png	0.5	224x224	0.412	4.390	17.011	3	111.964	9.023	9
rs21-0p_region2.png	0.5	224x224	0.426	4.357	13.982	4	114.000	9.867	7
rs21-0p_region3.png	0.5	224x224	0.409	4.252	23.634	2	114.130	8.853	9
rs21-0p_region4.png	0.5	224x224	0.423	4.175	21.154	2	112.826	8.503	10
rs22-1p_region1.png	0.5	224x224	0.524	4.447	13.596	4	112.615	13.346	9
rs22-1p_region2.png	0.5	224x224	0.507	4.493	12.420	7	115.556	7.570	1
rs22-1p_region3.png	0.5	224x224	0.491	4.145	14.039	5	113.810	6.789	6
rs22-1p_region4.png	0.5	224x224	0.503	4.171	15.352	3	114.146	6.898	11
rs23-2p_region1.png	0.5	224x224	0.493	4.138	11.195	6	114.255	7.286	5
rs23-2p_region4.png	0.5	224x224	0.474	4.296	15.686	4	113.265	7.278	4
rs23-2p_region6.png	0.5	224x224	0.557	4.762	8.564	2	112.326	9.294	6
rs23-2p_region7.png	0.5	224x224	0.562	4.487	10.416	2	116.591	7.313	7
rs24-1p_region1.png	0.5	224x224	0.380	4.263	9.027	1	113.731	9.433	15
rs24-1p_region2.png	0.5	224x224	0.405	4.747	0.0	0	116.923	11.339	12
rs24-1p_region3.png	0.5	224x224	0.412	4.664	0.0	0	114.167	10.517	15
rs24-1p_region4.png	0.5	224x224	0.429	4.316	12.099	2	116.842	7.871	10
rs27-2p_region1.png	0.5	224x224	0.524	4.524	11.747	5	111.161	7.801	7
rs27-2p_region2.png	0.5	224x224	0.455	3.816	14.486	8	111.257	10.111	14
rs27-2p_region3.png	0.5	224x224	0.502	4.310	14.353	4	111.951	8.384	4
rs28-1p_region1.png	0.5	224x224	0.482	4.211	12.746	8	113.707	7.326	5
rs28-1p_region2.png	0.5	224x224	0.482	4.370	12.771	3	114.783	8.400	6
rs28-1p_region3.png	0.5	224x224	0.476	4.228	10.750	5	112.909	8.600	4
rs28-1p_region5.png	0.5	224x224	0.572	4.791	9.652	4	112.941	8.509	3
rs29-0p_region1.png	0.5	224x224	0.736	5.480	6.323	6	110.483	6.912	1

..... (continúa en la página siguiente) .....

APÉNDICE A. RESULTADOS DE ENTRENAMIENTO

---

Tabla B.1 – (*viene de la página anterior*)

Imagen	Umbral	Tamaño	VD	VT	AS	NA	BA	VVD	MV
rs29-0p_region2.png	0.5	224x224	0.762	5.004	5.532	6	111.200	7.723	1
rs29-0p_region3.png	0.5	224x224	0.674	5.023	7.536	6	111.889	6.181	1
rs29-0p_region4.png	0.5	224x224	0.663	5.134	6.989	5	114.000	6.776	3
rs41-2p_region1.png	0.5	224x224	0.470	4.244	17.549	6	111.711	8.826	7
rs41-2p_region2.png	0.5	224x224	0.444	4.157	16.537	7	113.532	8.500	10
rs41-2p_region3.png	0.5	224x224	0.444	4.143	16.823	7	115.761	9.017	11
rs41-2p_region4.png	0.5	224x224	0.429	4.211	18.865	7	115.541	8.151	8
rs42-1p_region1.png	0.5	224x224	0.521	4.368	7.210	2	114.417	7.591	18
rs42-1p_region2.png	0.5	224x224	0.465	4.260	15.711	4	115.385	8.027	11
rs42-1p_region3.png	0.5	224x224	0.503	4.224	14.254	7	110.514	8.354	5
rs42-1p_region4.png	0.5	224x224	0.458	4.098	12.884	7	115.500	8.903	9
rs51-1p_region1.png	0.5	224x224	0.537	4.369	16.309	5	115.472	8.663	4
rs51-1p_region2.png	0.5	224x224	0.493	4.277	16.150	5	115.263	8.847	5
rs51-1p_region3.png	0.5	224x224	0.516	4.663	15.306	1	113.265	8.411	9
rs51-1p_region5.png	0.5	224x224	0.521	4.293	17.502	4	113.276	8.231	6
rs52-2p_region1.png	0.5	224x224	0.433	4.068	13.259	2	113.262	9.541	12
rs52-2p_region2.png	0.5	224x224	0.414	4.176	21.807	6	111.356	10.011	5
rs52-2p_region3.png	0.5	224x224	0.421	4.566	25.266	3	115.932	12.188	11
rs52-2p_region4.png	0.5	224x224	0.405	4.172	22.763	6	112.593	10.745	6
rs53-0p_region1.png	0.5	224x224	0.429	4.240	4.068	1	112.353	10.371	9
rs53-0p_region2.png	0.5	224x224	0.408	4.216	0.0	0	110.172	9.000	15
rs53-0p_region3.png	0.5	224x224	0.445	4.394	25.632	1	115.038	9.932	9
rs53-0p_region4.png	0.5	224x224	0.440	4.091	10.138	4	114.677	11.844	10
rs61-1p_region1.png	0.5	224x224	0.417	4.125	17.670	8	113.214	8.845	7
rs61-1p_region2.png	0.5	224x224	0.436	4.086	18.228	6	115.769	7.989	6
rs61-1p_region3.png	0.5	224x224	0.426	4.240	20.565	6	115.068	8.330	7
rs61-1p_region4.png	0.5	224x224	0.512	4.503	12.062	3	115.200	8.167	6
rs62-0p_region1.png	0.5	224x224	0.471	4.367	23.245	5	115.472	10.479	6
rs62-0p_region3.png	0.5	224x224	0.477	4.316	23.810	4	116.604	10.508	10
rs63-2p_region1.png	0.5	224x224	0.512	4.279	10.878	9	110.722	7.189	8
rs63-2p_region2.png	0.5	224x224	0.535	4.305	12.792	9	111.089	7.919	6
rs63-2p_region3.png	0.5	224x224	0.522	4.319	11.834	8	115.135	7.558	6
rs63-2p_region4.png	0.5	224x224	0.505	4.434	17.504	5	115.574	8.286	9
rs71-0p_region1.png	0.5	224x224	0.465	4.163	16.342	4	114.387	8.076	7
rs71-0p_region2.png	0.5	224x224	0.414	4.250	22.776	2	109.750	8.759	11
rs71-0p_region3.png	0.5	224x224	0.477	4.499	16.329	5	113.673	7.958	6
rs71-0p_region4.png	0.5	224x224	0.441	4.365	22.079	2	114.340	8.280	11
rs710-1p_region1.png	0.5	224x224	0.426	4.236	16.699	6	114.787	9.306	6
rs710-1p_region2.png	0.5	224x224	0.400	4.295	20.850	4	113.577	11.016	10
rs710-1p_region3.png	0.5	224x224	0.398	4.332	18.405	6	115.532	9.597	4
rs710-1p_region4.png	0.5	224x224	0.407	4.294	18.432	5	112.563	10.619	9
rs72-1p_region1.png	0.5	224x224	0.419	4.193	17.668	4	112.056	7.628	11
rs72-1p_region2.png	0.5	224x224	0.418	4.403	9.293	3	115.043	8.366	11
rs72-1p_region3.png	0.5	224x224	0.380	4.015	17.848	3	115.200	8.452	14
rs72-1p_region5.png	0.5	224x224	0.402	4.248	20.037	4	115.000	8.601	10
rs74-0p_region1.png	0.5	224x224	0.495	4.598	19.691	4	115.636	9.522	5
rs74-0p_region2.png	0.5	224x224	0.511	4.670	24.573	3	114.706	9.446	7
rs74-0p_region3.png	0.5	224x224	0.493	4.389	16.016	4	113.710	8.929	7
rs74-0p_region4.png	0.5	224x224	0.514	4.193	15.966	4	113.898	7.187	8
rs76-2p_region1.png	0.5	224x224	0.463	4.506	16.401	6	114.429	8.626	5
rs76-2p_region2.png	0.5	224x224	0.498	4.343	13.000	6	110.842	8.362	7
rs76-2p_region3.png	0.5	224x224	0.475	4.275	14.878	6	114.911	8.476	9
rs76-2p_region4.png	0.5	224x224	0.444	4.163	13.045	6	115.862	8.148	6
rs77-2p_region1.png	0.5	224x224	0.588	4.541	10.801	6	115.238	7.897	4
rs77-2p_region2.png	0.5	224x224	0.583	4.581	9.302	6	114.545	7.610	3
rs77-2p_region3.png	0.5	224x224	0.553	4.398	12.682	3	111.536	7.672	6

..... (continúa en la página siguiente) .....

APÉNDICE A. RESULTADOS DE ENTRENAMIENTO

---

Tabla B.1 – (*viene de la página anterior*)

Imagen	Umbbral	Tamaño	VD	VT	AS	NA	BA	VVD	MV
rs77-2p_region4.png	0.5	224x224	0.560	4.908	0.0	0	112.773	8.000	10
rs78-1p_region1.png	0.5	224x224	0.495	4.648	13.937	7	114.439	8.709	8
rs78-1p_region2.png	0.5	224x224	0.476	4.522	14.311	6	114.578	8.883	8
rs78-1p_region3.png	0.5	224x224	0.475	4.500	17.651	5	115.385	7.352	6
rs78-1p_region4.png	0.5	224x224	0.491	4.550	17.885	4	114.677	7.735	8
rs79-2p_region1.png	0.5	224x224	0.486	4.207	0.0	0	109.556	7.784	15
rs79-2p_region2.png	0.5	224x224	0.509	4.184	17.075	1	111.892	6.763	9
rs79-2p_region3.png	0.5	224x224	0.499	4.332	12.593	4	112.364	8.017	7
rs79-2p_region4.png	0.5	224x224	0.483	4.440	8.402	2	115.000	7.698	10
rs81-1p_region1.png	0.5	224x224	0.436	4.162	19.296	6	112.643	10.122	4
rs81-1p_region2.png	0.5	224x224	0.454	4.514	14.576	4	114.000	9.610	8
rs81-1p_region3.png	0.5	224x224	0.469	4.294	16.364	3	114.000	10.062	11
rs81-1p_region4.png	0.5	224x224	0.429	4.285	14.819	6	114.286	9.210	5
rs83-0p_region4.png	0.5	224x224	0.645	6.229	4.370	1	111.300	8.092	5
rs84-2p_region1.png	0.5	224x224	0.412	4.300	15.777	2	114.923	9.337	15
rs84-2p_region2.png	0.5	224x224	0.498	4.370	13.317	2	116.000	8.333	9
rs84-2p_region3.png	0.5	224x224	0.523	4.571	13.057	3	111.846	8.147	7
rs84-2p_region4.png	0.5	224x224	0.522	4.427	12.311	4	117.209	7.509	5
rs85-0p_region1.png	0.5	224x224	0.391	4.040	30.550	1	115.000	8.816	12
rs85-0p_region2.png	0.5	224x224	0.433	4.498	23.523	3	115.227	9.511	6
rs85-0p_region3.png	0.5	224x224	0.401	4.232	25.681	1	113.182	9.564	11
rs85-0p_region4.png	0.5	224x224	0.391	4.263	11.961	3	115.610	9.532	8
rs86-2p_region1.png	0.5	224x224	0.503	4.118	12.014	8	113.586	7.164	8
rs86-2p_region2.png	0.5	224x224	0.470	4.329	12.483	3	112.528	7.567	8
rs86-2p_region3.png	0.5	224x224	0.474	4.568	16.015	3	116.667	8.432	4
rs86-2p_region4.png	0.5	224x224	0.462	4.410	15.696	5	114.590	8.613	6
rs87-2p_region3.png	0.5	224x224	0.863	8.556	4.068	1	101.400	8.024	1
rs88-1p_region1.png	0.5	224x224	0.433	4.513	0.0	0	114.207	9.160	12
rs88-1p_region2.png	0.5	224x224	0.451	4.306	11.020	3	115.455	8.753	10

Tabla B.2: Métricas de venación por imagen.

Imagen	Umbbral	Tamaño	VD	VT	AS	NA	BA	VVD	MV
rf12-0p_region1.png	0.5	224x224	0.421	3.702	15.593	8	109.669	7.107	6
rf12-0p_region2.png	0.5	224x224	0.411	3.858	16.627	8	107.552	8.713	10
rf12-0p_region3.png	0.5	224x224	0.391	3.595	16.135	9	108.174	7.480	12
rf12-0p_region4.png	0.5	224x224	0.414	3.744	17.328	8	110.305	10.788	17
rf13-2p_region1.png	0.5	224x224	0.418	3.491	18.565	5	109.109	8.133	8
rf13-2p_region2.png	0.5	224x224	0.398	3.552	18.125	4	109.432	8.382	8
rf13-2p_region3.png	0.5	224x224	0.433	3.470	19.254	4	114.200	8.398	7
rf13-2p_region4.png	0.5	224x224	0.467	3.638	15.902	6	106.726	8.288	7
rf14-1p_region1.png	0.5	224x224	0.474	3.790	16.531	6	107.641	7.973	11
rf14-1p_region2.png	0.5	224x224	0.482	3.661	14.391	4	113.788	7.344	16
rf14-1p_region6.png	0.5	224x224	0.528	3.722	15.371	6	109.800	7.272	14
rf14-1p_region7.png	0.5	224x224	0.469	3.722	15.001	7	112.600	7.565	7
rf15-0p_region1.png	0.5	224x224	0.406	3.780	18.607	7	111.093	8.468	7
rf15-0p_region2.png	0.5	224x224	0.384	3.659	21.230	7	110.432	8.694	6
rf15-0p_region3.png	0.5	224x224	0.413	3.717	18.327	8	112.941	8.636	5
rf15-0p_region4.png	0.5	224x224	0.440	3.599	15.267	9	113.936	8.237	7
rf16-1p_region1.png	0.5	224x224	0.466	3.696	17.309	5	113.793	9.865	5
rf16-1p_region2.png	0.5	224x224	0.544	3.885	17.500	5	108.118	8.612	5
rf16-1p_region3.png	0.5	224x224	0.472	3.624	13.965	8	112.247	9.873	4
rf16-1p_region4.png	0.5	224x224	0.516	3.955	16.701	5	113.400	7.668	10
rf18-0p_region1.png	0.5	224x224	0.350	3.716	28.001	4	112.681	8.600	15
rf18-0p_region2.png	0.5	224x224	0.348	3.654	24.300	7	110.464	8.767	9
rf18-0p_region3.png	0.5	224x224	0.368	3.635	19.748	8	108.423	8.173	14

.....(continúa en la página siguiente).....

APÉNDICE A. RESULTADOS DE ENTRENAMIENTO

---

Tabla B.2 – (*viene de la página anterior*)

Imagen	Umbral	Tamaño	VD	VT	AS	NA	BA	VVD	MV
rf18-0p_region4.png	0.5	224x224	0.392	3.825	22.852	6	105.474	7.361	10
rf21-1p_region1.png	0.5	224x224	0.550	3.985	14.252	7	99.785	8.093	6
rf21-1p_region2.png	0.5	224x224	0.561	3.876	9.045	7	108.482	7.602	8
rf21-1p_region4.png	0.5	224x224	0.570	3.861	12.580	8	105.056	7.347	7
rf21-1p_region5.png	0.5	224x224	0.531	3.760	12.248	9	111.486	7.402	5
rf22-1p_region1.png	0.5	224x224	0.554	3.977	10.964	12	107.426	7.054	11
rf22-1p_region2.png	0.5	224x224	0.531	3.727	9.713	13	108.766	7.492	11
rf22-1p_region3.png	0.5	224x224	0.565	3.950	11.219	10	106.221	8.186	14
rf23-0p_region1.png	0.5	224x224	0.406	3.981	20.907	6	111.933	9.195	8
rf23-0p_region2.png	0.5	224x224	0.396	3.609	22.943	2	116.066	8.307	10
rf23-0p_region3.png	0.5	224x224	0.403	3.698	22.480	3	110.333	8.521	8
rf23-0p_region4.png	0.5	224x224	0.379	3.632	20.570	6	104.716	9.860	4
rf25-1p_region1.png	0.5	224x224	0.559	3.884	9.703	8	114.560	6.564	5
rf25-1p_region2.png	0.5	224x224	0.450	3.656	13.721	5	115.000	7.337	6
rf25-1p_region3.png	0.5	224x224	0.433	3.713	12.753	3	107.062	8.413	11
rf25-1p_region4.png	0.5	224x224	0.497	3.562	10.993	6	107.105	5.977	6
rf26-2p_region1.png	0.5	224x224	0.579	3.885	8.452	13	105.683	8.734	5
rf26-2p_region2.png	0.5	224x224	0.578	3.817	11.867	7	106.621	7.296	7
rf26-2p_region3.png	0.5	224x224	0.578	3.892	10.050	8	104.019	7.932	5
rf26-2p_region4.png	0.5	224x224	0.624	3.816	8.264	12	103.200	8.044	8
rf28-0p_region1.png	0.5	224x224	0.416	3.950	21.438	4	104.013	8.050	9
rf28-0p_region2.png	0.5	224x224	0.414	3.748	21.613	6	109.674	10.511	5
rf28-0p_region3.png	0.5	224x224	0.380	3.765	23.343	5	110.653	10.000	8
rf28-0p_region4.png	0.5	224x224	0.404	3.774	23.600	5	111.078	8.283	11
rf44-0p_region1.png	0.5	224x224	0.386	3.680	0.0	0	104.586	9.983	12
rf44-0p_region2.png	0.5	224x224	0.395	3.598	25.657	1	105.130	10.695	8
rf44-0p_region3.png	0.5	224x224	0.375	3.595	3.568	1	106.737	10.331	14
rf51-2p_region1.png	0.5	224x224	0.718	4.178	6.241	4	107.155	7.520	2
rf51-2p_region2.png	0.5	224x224	0.684	4.333	5.554	3	98.323	6.855	2
rf51-2p_region3.png	0.5	224x224	0.744	4.296	5.410	2	102.952	6.361	3
rf51-2p_region4.png	0.5	224x224	0.676	4.114	6.700	3	101.671	6.785	1
rf510-1p_region1.png	0.5	224x224	0.540	3.693	13.851	10	107.703	6.405	8
rf510-1p_region2.png	0.5	224x224	0.531	3.762	13.553	9	110.530	6.524	13
rf510-1p_region3.png	0.5	224x224	0.519	3.682	18.005	6	110.460	6.832	13
rf510-1p_region4.png	0.5	224x224	0.510	3.753	11.573	4	112.000	6.127	11
rf52-1p_region1.png	0.5	224x224	0.428	3.680	15.377	9	109.095	7.540	8
rf52-1p_region2.png	0.5	224x224	0.416	3.772	16.089	3	107.273	8.609	11
rf52-1p_region3.png	0.5	224x224	0.386	3.545	20.933	4	112.042	7.967	13
rf52-1p_region4.png	0.5	224x224	0.424	3.682	18.425	6	106.559	8.822	7
rf53-0p_region1.png	0.5	224x224	0.378	3.605	17.518	12	107.384	9.357	5
rf53-0p_region2.png	0.5	224x224	0.369	3.559	19.554	8	114.935	7.828	3
rf53-0p_region3.png	0.5	224x224	0.371	3.595	22.246	7	113.730	9.216	11
rf53-0p_region4.png	0.5	224x224	0.351	3.481	19.883	8	111.355	9.670	5
rf54-2p_region1.png	0.5	224x224	0.903	3.743	13.462	9	107.171	7.385	10
rf54-2p_region2.png	0.5	224x224	0.616	3.782	13.796	4	104.276	7.052	11
rf54-2p_region3.png	0.5	224x224	0.941	3.407	13.842	8	107.478	7.286	12
rf54-2p_region4.png	0.5	224x224	0.816	3.645	13.417	7	111.491	8.534	9
rf56-0p_region1.png	0.5	224x224	0.446	3.696	17.623	9	111.523	7.855	10
rf56-0p_region2.png	0.5	224x224	0.404	3.747	5.711	2	98.894	8.831	13
rf56-0p_region3.png	0.5	224x224	0.405	3.411	22.295	2	111.931	8.599	10
rf56-0p_region4.png	0.5	224x224	0.453	3.743	17.791	9	110.531	7.800	10
rf57-0p_region1.png	0.5	224x224	0.400	3.631	20.201	7	111.169	8.333	12
rf57-0p_region2.png	0.5	224x224	0.406	3.615	20.860	8	108.493	8.782	8
rf57-0p_region3.png	0.5	224x224	0.397	3.736	20.956	6	109.902	8.085	15
rf57-0p_region4.png	0.5	224x224	0.398	3.573	18.630	10	114.750	7.912	12
rf58-1p_region1.png	0.5	224x224	0.506	3.765	11.071	10	106.088	7.924	12

..... (continúa en la página siguiente) .....

APÉNDICE A. RESULTADOS DE ENTRENAMIENTO

---

Tabla B.2 – (*viene de la página anterior*)

Imagen	Umbral	Tamaño	VD	VT	AS	NA	BA	VVD	MV
rf58-1p_region4.png	0.5	224x224	0.477	3.603	13.509	10	108.274	7.236	13
rf58-1p_region5.png	0.5	224x224	0.482	3.660	12.018	6	104.046	6.750	14
rf58-1p_region6.png	0.5	224x224	0.545	3.838	13.048	10	106.626	7.461	13
rf59-0p_region3.png	0.5	224x224	0.642	4.199	6.265	6	103.966	7.414	1
rf59-0p_region4.png	0.5	224x224	0.774	4.822	4.222	1	93.158	6.833	2
rf71-2p_region1.png	0.5	224x224	0.444	3.754	13.322	4	105.480	8.034	8
rf71-2p_region5.png	0.5	224x224	0.490	3.795	12.713	5	103.541	7.829	7
rf71-2p_region6.png	0.5	224x224	0.479	3.810	12.701	8	107.304	8.149	4
rf71-2p_region7.png	0.5	224x224	0.505	3.458	12.634	5	107.512	7.266	5
rf72-0p_region1.png	0.5	224x224	0.726	3.936	7.767	4	105.968	7.034	1
rf72-0p_region2.png	0.5	224x224	0.686	3.930	7.361	5	107.939	7.177	2
rf72-0p_region3.png	0.5	224x224	0.683	3.926	8.717	4	108.356	6.148	1
rf72-0p_region4.png	0.5	224x224	0.634	4.012	8.539	4	107.647	7.363	2
rf73-2p_region1.png	0.5	224x224	0.487	3.642	16.393	3	108.207	7.864	6
rf73-2p_region2.png	0.5	224x224	0.451	3.498	16.262	6	111.138	8.606	4
rf73-2p_region3.png	0.5	224x224	0.429	3.547	14.339	3	109.660	14.171	16
rf74-2p_region1.png	0.5	224x224	0.410	3.570	11.815	2	114.750	8.534	8
rf74-2p_region3.png	0.5	224x224	0.422	3.518	17.594	6	108.136	9.011	4
rf74-2p_region4.png	0.5	224x224	0.427	3.560	16.130	7	105.659	8.441	1
rf74-2p_region5.png	0.5	224x224	0.437	3.629	15.840	7	110.759	8.198	6
rf75-0p_region1.png	0.5	224x224	0.395	3.604	18.522	7	114.000	9.176	2
rf75-0p_region2.png	0.5	224x224	0.367	3.560	19.427	6	110.400	9.414	5
rf75-0p_region3.png	0.5	224x224	0.360	3.724	22.933	3	109.017	10.111	5
rf75-0p_region4.png	0.5	224x224	0.347	3.715	23.227	5	111.968	9.661	3
rf76-0p_region1.png	0.5	224x224	0.348	3.503	20.813	4	113.333	8.085	9
rf76-0p_region2.png	0.5	224x224	0.329	3.578	17.874	5	108.906	9.763	5
rf76-0p_region3.png	0.5	224x224	0.357	3.571	16.862	5	110.727	8.715	8
rf76-0p_region4.png	0.5	224x224	0.350	3.633	14.713	6	111.831	10.073	8
rf77-1p_region1.png	0.5	224x224	0.416	3.635	8.373	2	104.780	8.119	9
rf77-1p_region2.png	0.5	224x224	0.415	3.784	17.707	2	105.158	9.318	7
rf77-1p_region3.png	0.5	224x224	0.518	4.082	6.431	2	109.317	7.914	9
rf77-1p_region4.png	0.5	224x224	0.502	3.777	0.0	0	103.732	7.944	11
rf78-2p_region1.png	0.5	224x224	0.588	3.867	11.192	5	106.134	8.185	3
rf78-2p_region2.png	0.5	224x224	0.546	3.747	13.139	6	111.415	6.893	9
rf78-2p_region3.png	0.5	224x224	0.581	3.773	10.175	8	107.709	7.119	4
rf78-2p_region4.png	0.5	224x224	0.566	3.874	10.490	7	105.023	7.903	5
rf79-1p_region1.png	0.5	224x224	0.414	3.619	10.270	6	113.852	8.399	5
rf79-1p_region2.png	0.5	224x224	0.388	3.710	4.897	3	107.333	9.379	7
rf79-1p_region5.png	0.5	224x224	0.420	3.628	16.361	5	110.783	7.962	6
rf79-1p_region6.png	0.5	224x224	0.411	3.592	14.041	4	112.077	9.028	10
rs21-0p_region1.png	0.5	224x224	0.361	3.610	19.449	3	113.898	8.497	8
rs21-0p_region2.png	0.5	224x224	0.375	3.752	24.773	3	112.177	9.569	9
rs21-0p_region3.png	0.5	224x224	0.357	3.685	24.437	1	113.483	7.939	8
rs21-0p_region4.png	0.5	224x224	0.390	3.595	22.272	3	111.250	7.447	8
rs22-1p_region1.png	0.5	224x224	0.474	3.583	13.014	5	112.977	12.042	10
rs22-1p_region2.png	0.5	224x224	0.478	3.773	14.083	6	104.442	7.929	3
rs22-1p_region3.png	0.5	224x224	0.446	3.557	16.293	3	110.900	6.267	8
rs22-1p_region4.png	0.5	224x224	0.500	3.741	15.834	4	110.620	6.456	10
rs23-2p_region1.png	0.5	224x224	0.452	3.742	13.819	5	114.000	7.244	5
rs23-2p_region4.png	0.5	224x224	0.441	3.623	18.816	3	113.510	6.682	8
rs23-2p_region6.png	0.5	224x224	0.484	3.765	14.184	1	108.646	9.585	6
rs23-2p_region7.png	0.5	224x224	0.512	3.650	0.0	0	112.703	5.909	10
rs24-1p_region1.png	0.5	224x224	0.340	3.582	22.212	5	114.000	8.815	3
rs24-1p_region2.png	0.5	224x224	0.334	3.509	21.524	4	105.600	9.118	3
rs24-1p_region3.png	0.5	224x224	0.351	3.791	19.615	4	110.930	9.558	2
rs24-1p_region4.png	0.5	224x224	0.407	3.748	17.313	6	116.571	8.011	1

..... (continúa en la página siguiente) .....

APÉNDICE A. RESULTADOS DE ENTRENAMIENTO

---

Tabla B.2 – (*viene de la página anterior*)

Imagen	Umbral	Tamaño	VD	VT	AS	NA	BA	VVD	MV
rs27-2p_region1.png	0.5	224x224	0.475	3.658	13.716	6	102.000	7.703	5
rs27-2p_region2.png	0.5	224x224	0.480	3.489	13.699	10	109.984	7.590	13
rs27-2p_region3.png	0.5	224x224	0.434	3.576	18.142	4	118.966	7.267	5
rs28-1p_region1.png	0.5	224x224	0.453	3.807	15.120	5	115.345	6.598	10
rs28-1p_region2.png	0.5	224x224	0.411	3.465	13.800	3	114.643	7.633	7
rs28-1p_region3.png	0.5	224x224	0.450	3.679	12.085	7	113.342	7.155	6
rs28-1p_region5.png	0.5	224x224	0.491	3.879	12.038	3	106.138	8.595	3
rs29-0p_region1.png	0.5	224x224	0.658	3.843	7.265	6	102.186	6.255	2
rs29-0p_region2.png	0.5	224x224	0.657	3.740	7.689	5	107.388	7.140	1
rs29-0p_region3.png	0.5	224x224	0.608	3.823	8.776	6	106.364	6.484	1
rs29-0p_region4.png	0.5	224x224	0.599	3.536	8.226	6	106.114	5.982	2
rs41-2p_region1.png	0.5	224x224	0.441	3.654	18.004	6	109.955	8.303	10
rs41-2p_region2.png	0.5	224x224	0.417	3.605	17.188	8	113.576	8.005	7
rs41-2p_region3.png	0.5	224x224	0.434	3.686	18.578	6	116.345	8.098	11
rs41-2p_region4.png	0.5	224x224	0.403	3.545	17.320	8	109.938	8.016	9
rs42-1p_region1.png	0.5	224x224	0.489	3.715	11.524	5	114.361	7.006	11
rs42-1p_region2.png	0.5	224x224	0.463	3.766	15.255	6	104.744	7.860	8
rs42-1p_region3.png	0.5	224x224	0.469	3.672	13.618	8	107.089	7.415	5
rs42-1p_region4.png	0.5	224x224	0.442	3.703	14.818	8	110.700	7.333	8
rs51-1p_region1.png	0.5	224x224	0.486	3.558	14.262	7	109.446	7.549	5
rs51-1p_region2.png	0.5	224x224	0.463	3.655	16.890	5	111.316	7.604	5
rs51-1p_region3.png	0.5	224x224	0.475	3.652	0.0	0	109.429	7.181	13
rs51-1p_region5.png	0.5	224x224	0.493	3.666	18.415	4	110.506	7.492	5
rs52-2p_region1.png	0.5	224x224	0.433	3.772	22.535	4	111.000	9.317	9
rs52-2p_region2.png	0.5	224x224	0.372	3.644	24.958	5	110.000	10.602	6
rs52-2p_region3.png	0.5	224x224	0.384	3.690	22.981	4	109.405	12.017	8
rs52-2p_region4.png	0.5	224x224	0.365	3.647	23.470	6	106.455	10.978	8
rs53-0p_region1.png	0.5	224x224	0.408	3.487	16.735	2	109.345	9.876	9
rs53-0p_region2.png	0.5	224x224	0.390	3.552	0.0	0	108.441	8.126	15
rs53-0p_region3.png	0.5	224x224	0.404	3.513	17.314	3	105.789	10.983	6
rs53-0p_region4.png	0.5	224x224	0.398	3.522	13.959	2	107.155	11.174	10
rs61-1p_region1.png	0.5	224x224	0.390	3.635	20.330	7	111.061	7.793	7
rs61-1p_region2.png	0.5	224x224	0.423	3.681	17.789	7	115.714	7.886	7
rs61-1p_region3.png	0.5	224x224	0.391	3.726	17.856	8	112.442	9.078	6
rs61-1p_region4.png	0.5	224x224	0.421	3.361	16.933	2	111.346	8.030	8
rs62-0p_region1.png	0.5	224x224	0.441	3.792	24.084	5	104.211	9.758	6
rs62-0p_region3.png	0.5	224x224	0.417	3.559	21.291	5	107.045	10.368	8
rs63-2p_region1.png	0.5	224x224	0.513	3.819	14.287	6	108.789	7.619	10
rs63-2p_region2.png	0.5	224x224	0.527	3.725	14.478	7	102.818	7.236	8
rs63-2p_region3.png	0.5	224x224	0.505	3.526	11.595	12	104.706	7.760	5
rs63-2p_region4.png	0.5	224x224	0.489	3.829	16.170	6	111.150	6.569	9
rs71-0p_region1.png	0.5	224x224	0.417	3.535	19.709	6	104.229	8.593	2
rs71-0p_region2.png	0.5	224x224	0.357	3.373	22.945	2	108.254	8.297	10
rs71-0p_region3.png	0.5	224x224	0.433	3.700	19.501	4	106.812	10.382	6
rs71-0p_region4.png	0.5	224x224	0.411	3.742	22.884	2	108.720	7.448	11
rs710-1p_region1.png	0.5	224x224	0.391	3.700	17.526	4	116.667	8.751	9
rs710-1p_region2.png	0.5	224x224	0.357	3.604	21.052	5	111.863	10.330	8
rs710-1p_region3.png	0.5	224x224	0.341	3.646	19.619	5	108.618	9.232	6
rs710-1p_region4.png	0.5	224x224	0.365	3.612	22.448	4	107.664	9.683	11
rs72-1p_region1.png	0.5	224x224	0.380	3.562	17.709	3	112.875	7.108	14
rs72-1p_region2.png	0.5	224x224	0.362	3.464	10.764	1	108.338	8.874	11
rs72-1p_region3.png	0.5	224x224	0.361	3.570	17.951	3	113.333	7.873	17
rs72-1p_region5.png	0.5	224x224	0.370	3.664	20.105	4	107.854	8.088	10
rs74-0p_region1.png	0.5	224x224	0.466	3.786	20.395	5	108.960	8.757	5
rs74-0p_region2.png	0.5	224x224	0.490	3.829	19.306	5	102.084	8.506	6
rs74-0p_region3.png	0.5	224x224	0.458	3.722	15.956	5	107.767	9.192	5

..... (continúa en la página siguiente) .....

APÉNDICE A. RESULTADOS DE ENTRENAMIENTO

---

Tabla B.2 – (*viene de la página anterior*)

Imagen	Umbral	Tamaño	VD	VT	AS	NA	BA	VVD	MV
rs74-0p_region4.png	0.5	224x224	0.519	3.864	17.603	4	106.385	7.474	9
rs76-2p_region1.png	0.5	224x224	0.437	3.705	16.979	6	111.124	8.329	5
rs76-2p_region2.png	0.5	224x224	0.461	3.699	13.619	8	107.556	9.415	5
rs76-2p_region3.png	0.5	224x224	0.431	3.629	15.196	6	108.662	8.188	11
rs76-2p_region4.png	0.5	224x224	0.418	3.538	17.225	4	109.180	8.164	7
rs77-2p_region1.png	0.5	224x224	0.537	3.770	13.509	5	110.286	6.876	4
rs77-2p_region2.png	0.5	224x224	0.542	3.803	16.200	2	103.586	7.659	7
rs77-2p_region3.png	0.5	224x224	0.519	3.694	14.339	3	111.545	6.129	8
rs77-2p_region4.png	0.5	224x224	0.510	3.713	10.382	3	106.432	7.555	5
rs78-1p_region1.png	0.5	224x224	0.464	3.743	13.801	8	104.968	8.253	7
rs78-1p_region2.png	0.5	224x224	0.467	3.746	17.025	5	106.382	9.322	9
rs78-1p_region3.png	0.5	224x224	0.473	4.023	17.491	5	109.884	7.291	6
rs78-1p_region4.png	0.5	224x224	0.470	3.816	14.204	6	106.762	7.782	8
rs79-2p_region1.png	0.5	224x224	0.449	3.610	7.399	1	107.036	7.067	9
rs79-2p_region2.png	0.5	224x224	0.490	3.554	16.926	1	102.527	6.193	9
rs79-2p_region3.png	0.5	224x224	0.441	3.650	0.0	0	107.891	7.994	11
rs79-2p_region4.png	0.5	224x224	0.420	3.621	0.0	0	111.273	6.921	10
rs81-1p_region1.png	0.5	224x224	0.402	3.559	20.110	6	105.385	9.153	4
rs81-1p_region2.png	0.5	224x224	0.420	3.704	16.469	4	109.071	8.287	8
rs81-1p_region3.png	0.5	224x224	0.447	3.586	15.812	4	110.630	8.293	8
rs81-1p_region4.png	0.5	224x224	0.373	3.584	20.018	3	115.000	8.669	6
rs83-0p_region4.png	0.5	224x224	0.551	3.816	9.097	1	112.216	6.593	3
rs84-2p_region1.png	0.5	224x224	0.401	3.706	14.374	3	108.714	7.955	16
rs84-2p_region2.png	0.5	224x224	0.460	3.607	17.012	2	105.929	7.902	8
rs84-2p_region3.png	0.5	224x224	0.454	3.473	17.517	1	100.583	6.860	9
rs84-2p_region4.png	0.5	224x224	0.473	3.735	16.565	3	111.900	7.173	5
rs85-0p_region1.png	0.5	224x224	0.375	3.568	0.0	0	114.737	8.723	13
rs85-0p_region2.png	0.5	224x224	0.375	3.778	20.806	1	114.762	8.569	9
rs85-0p_region3.png	0.5	224x224	0.362	3.678	26.391	1	113.276	8.343	12
rs85-0p_region4.png	0.5	224x224	0.341	3.591	25.650	2	109.525	9.751	7
rs86-2p_region1.png	0.5	224x224	0.508	3.536	11.711	7	110.755	7.063	10
rs86-2p_region2.png	0.5	224x224	0.449	3.687	13.263	4	107.937	7.335	10
rs86-2p_region3.png	0.5	224x224	0.437	3.540	12.014	6	107.510	8.050	4
rs86-2p_region4.png	0.5	224x224	0.452	3.648	14.971	6	108.000	7.866	5
rs87-2p_region3.png	0.5	224x224	0.786	4.236	5.178	4	105.548	6.564	1
rs88-1p_region1.png	0.5	224x224	0.364	3.790	17.846	3	112.417	7.951	6
rs88-1p_region2.png	0.5	224x224	0.381	3.694	17.152	4	107.797	8.348	4

## Apéndice C

# Gráficas de Entrenamiento

---

En este anexo se muestran las curvas de evolución de la función de pérdida durante el entrenamiento de los modelos. Cada gráfica muestra la evolución tanto en el conjunto de entrenamiento como en el de validación.

## C.1 Experimento 1

En este experimento se evalúa el efecto de la función de pérdida utilizada en el entrenamiento de la red, variando únicamente la función de pérdida entre BCE, BCE + Dice, Dice y Focal.

Las Figuras [C.1](#), [C.2](#) y [C.3](#) muestran la evolución de los resultados del entrenamiento para los tres folds, usando UNet como arquitectura y mobilenet\_v2 como encoder.

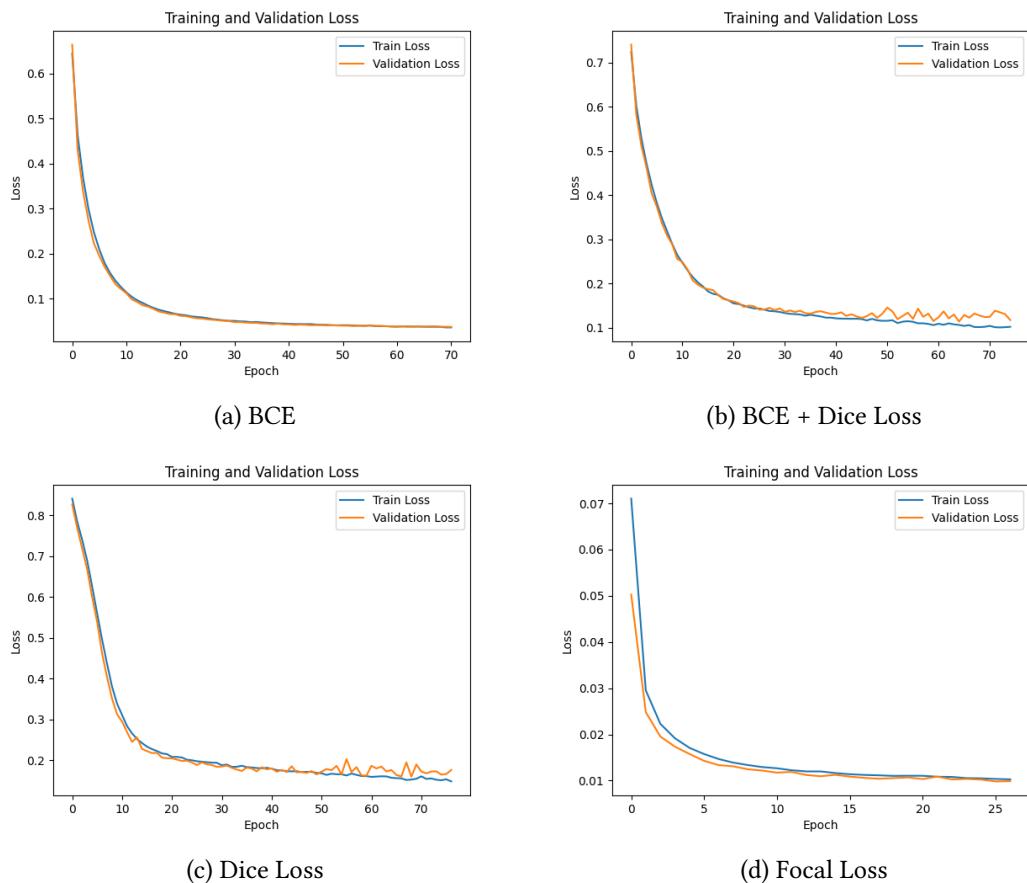


Figura C.1: Evolución de las funciones de pérdida para el Fold 1 en el Experimento 1

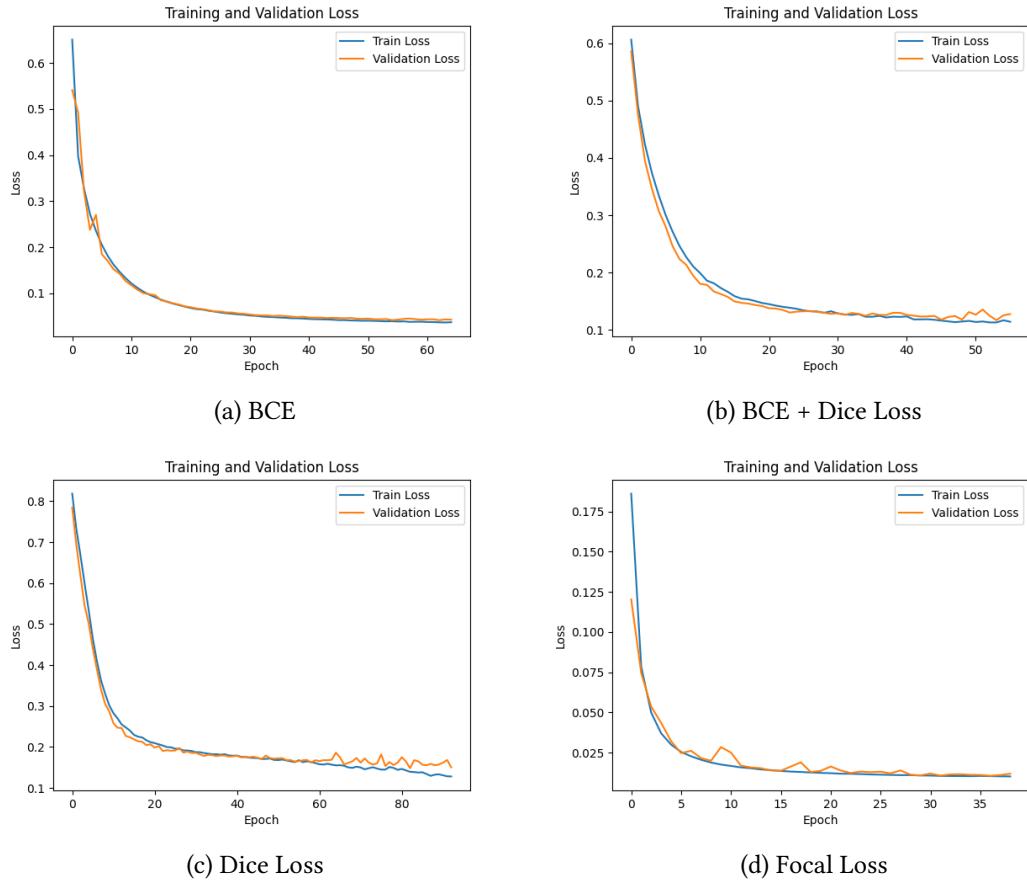


Figura C.2: Evolución de las funciones de pérdida para el Fold 2 en el Experimento 1

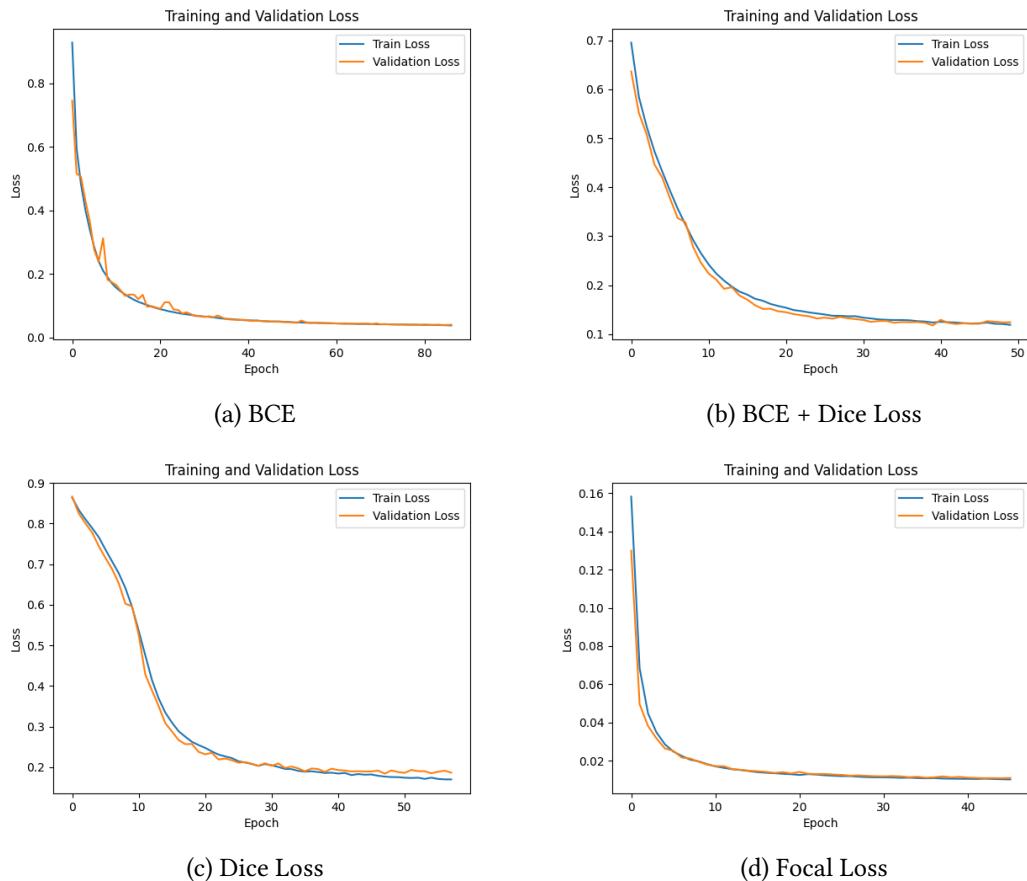


Figura C.3: Evolución de las funciones de pérdida para el Fold 3 en el Experimento 1

## C.2 Experimento 2

En este experimento se analiza el efecto del encoder en la calidad del entrenamiento, variando únicamente el encoder entre EfficientNet-b0, MobileNet\_V2, ResNet34 y ResNet50.

Las Figuras [C.4](#), [C.5](#) y [C.6](#) muestran la evolución de los resultados del entrenamiento para los tres folds, usando UNet como arquitectura y Dice como función de pérdida.

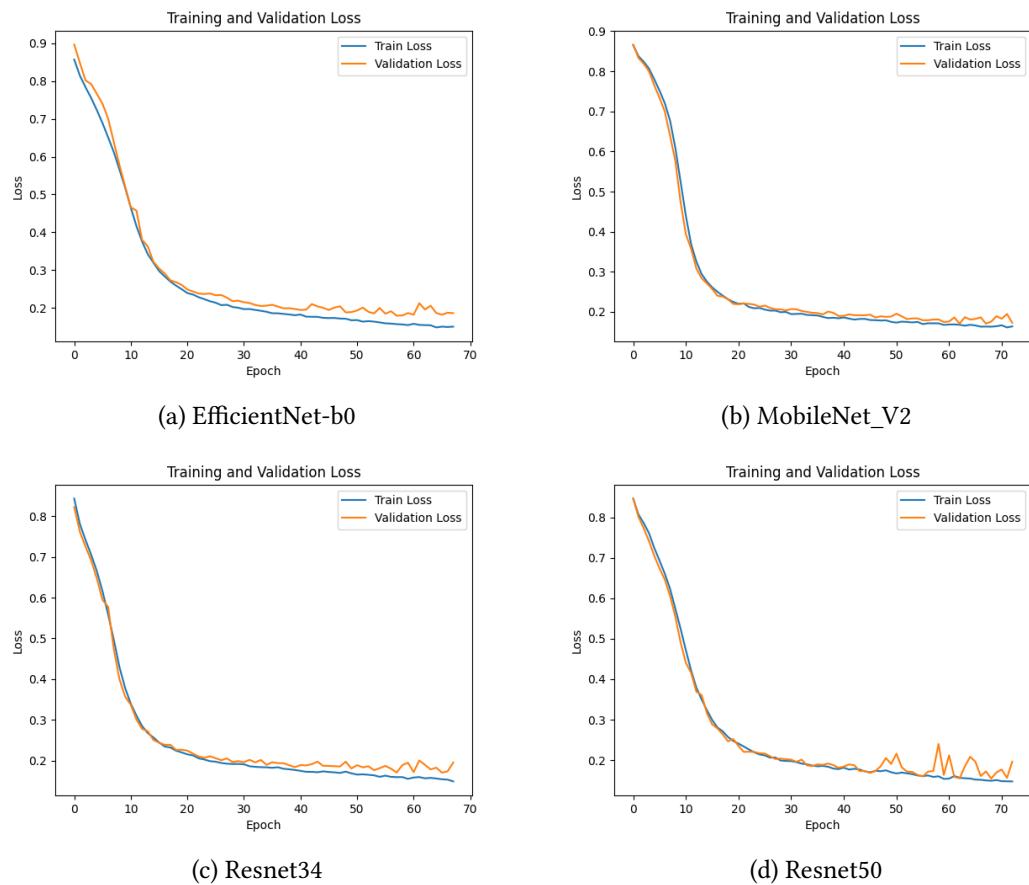


Figura C.4: Evolución de las funciones de pérdida para el Fold 1 del Experimento 2

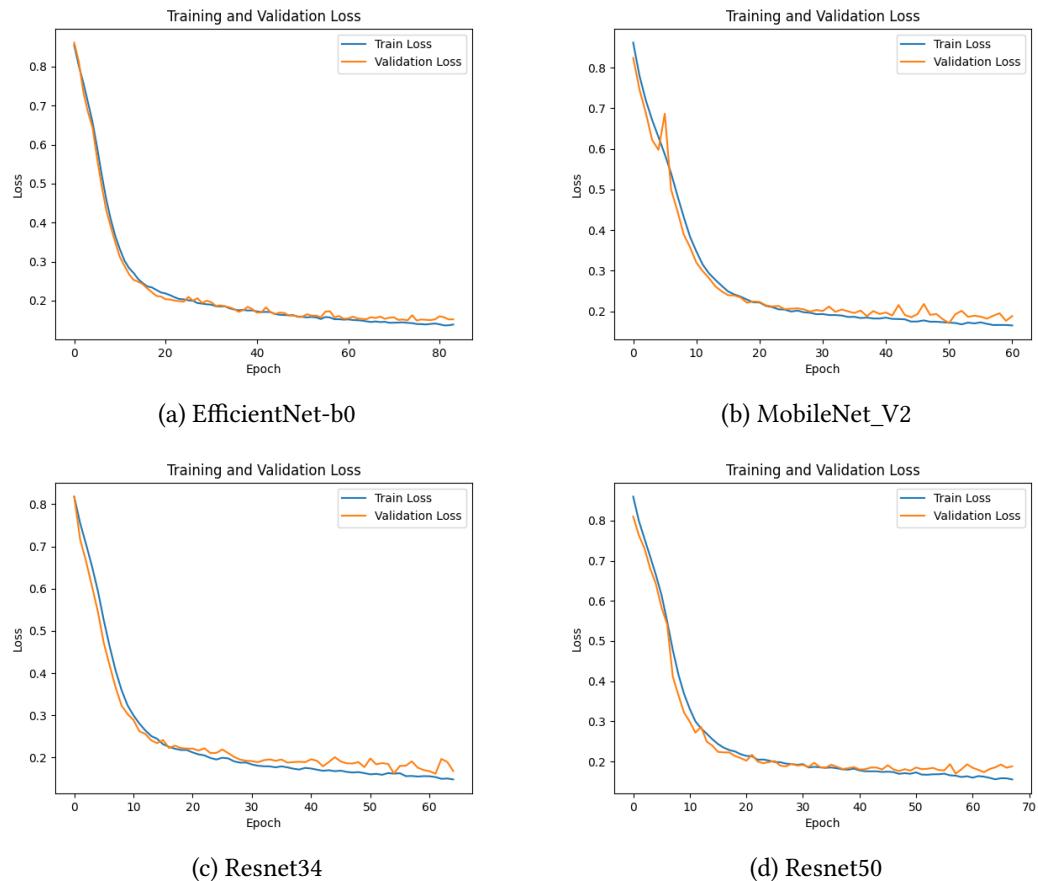


Figura C.5: Evolución de las funciones de pérdida para el Fold 2 del Experimento 2

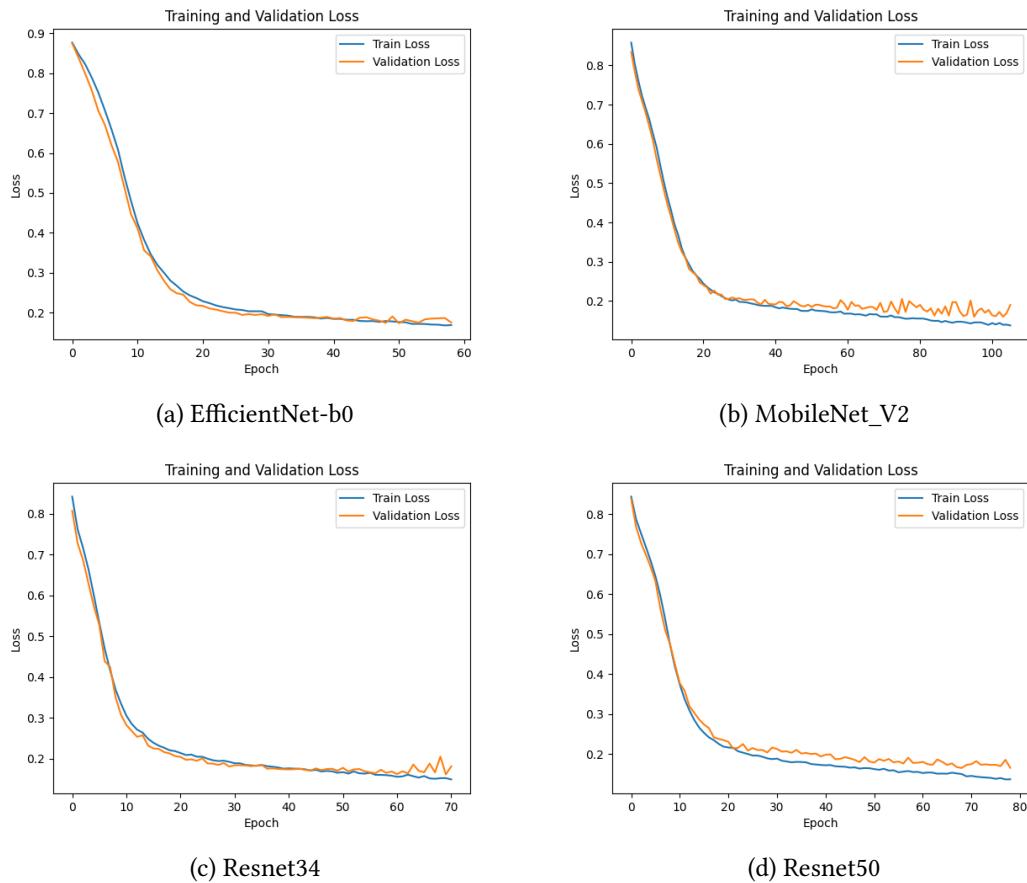


Figura C.6: Evolución de las funciones de pérdida para el Fold 3 del Experimento 2

### C.3 Experimento 3

En este experimento se estudia el efecto de la arquitectura de la red en el entrenamiento, variando la arquitectura entre U-net, FPN, PSPNet y DeepLab.

Las Figuras C.7, C.8 y C.9 muestran la evolución de los resultados del entrenamiento para los tres folds, usando mobilenet\_v2 como encoder y Dice como función de pérdida.

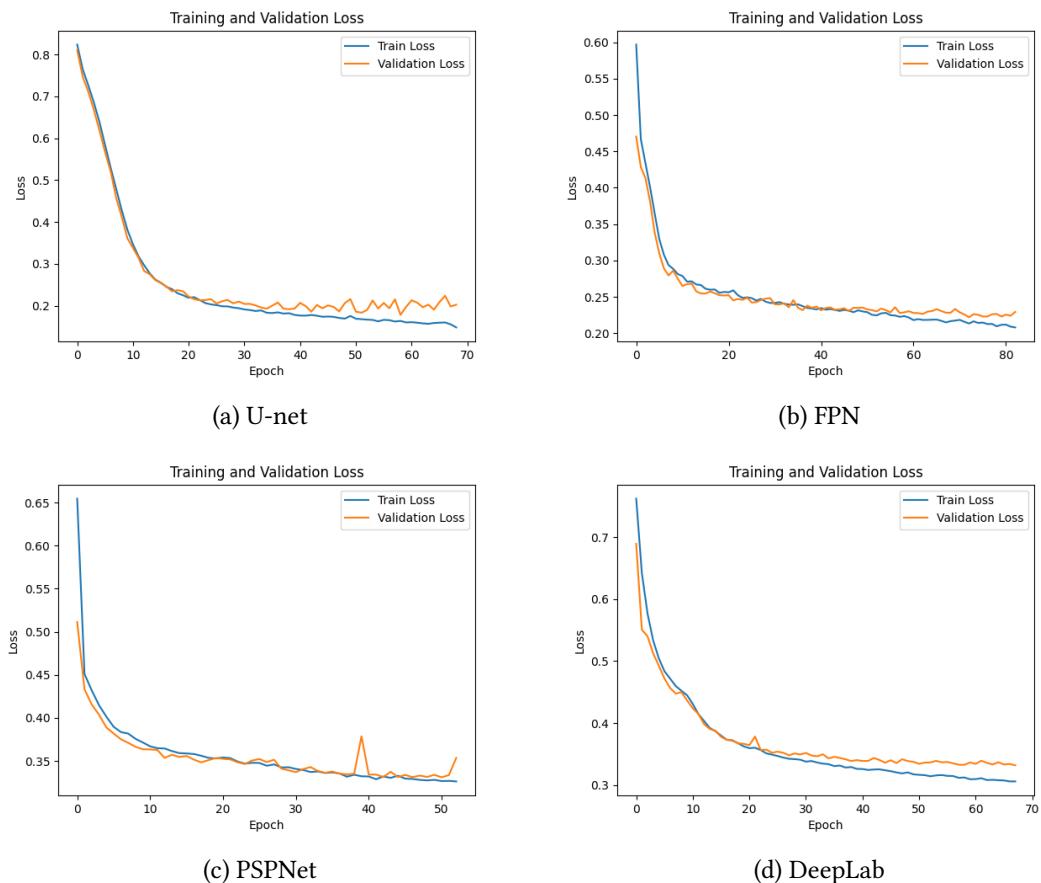


Figura C.7: Evolución de las funciones de pérdida para el Fold 1 del Experimento 3

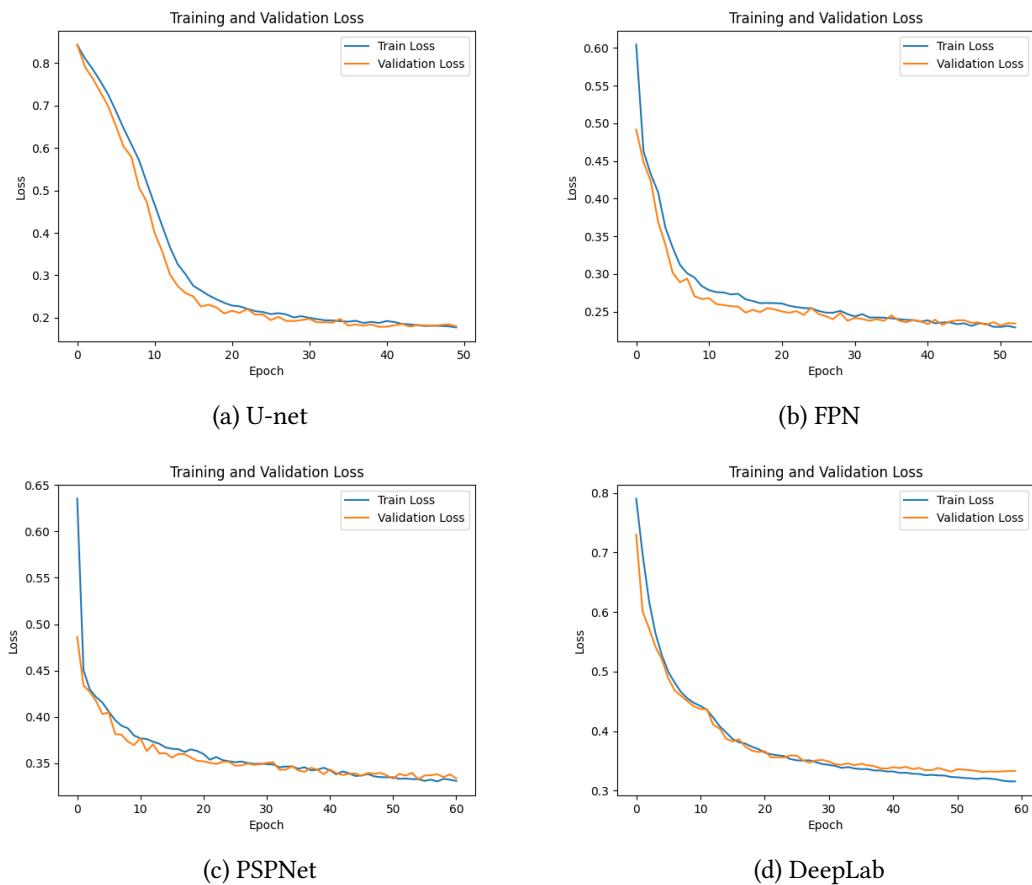


Figura C.8: Evolución de las funciones de pérdida para el Fold 2 del Experimento 3

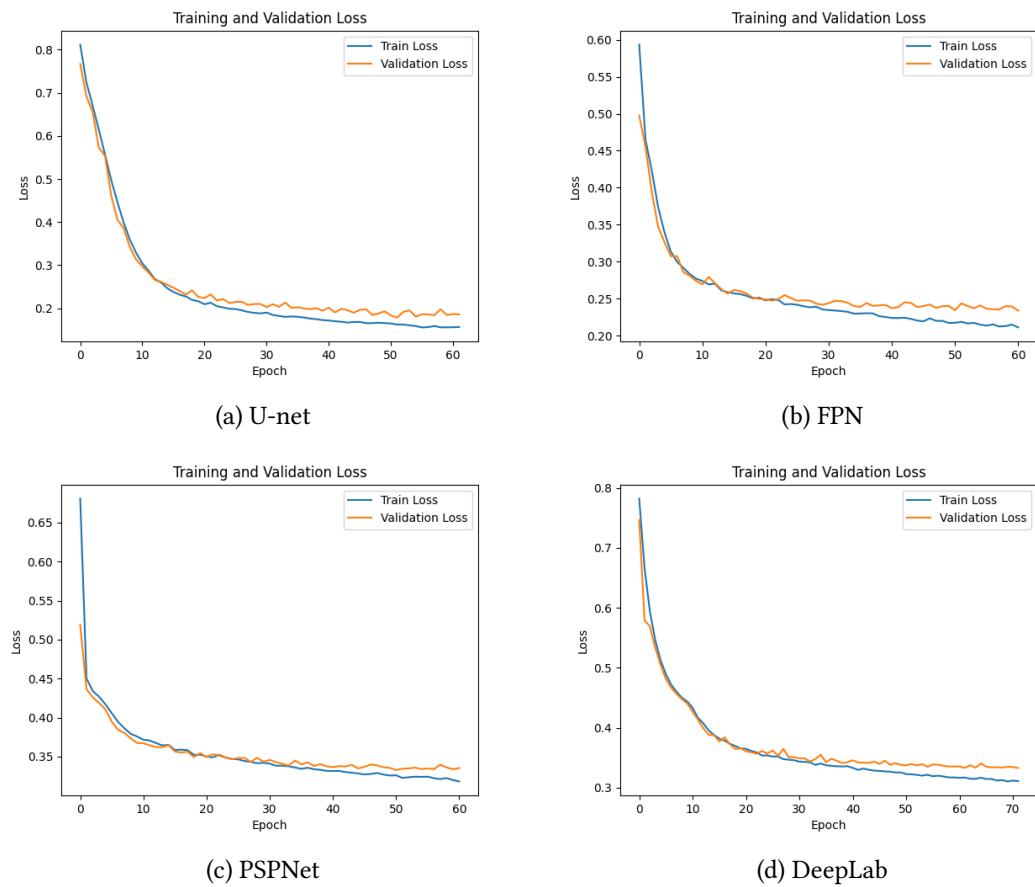


Figura C.9: Evolución de las funciones de pérdida para el Fold 3 del Experimento 3



# Bibliografía

---

- [1] C. A. Price, O. Symonova, Y. Mileyko, T. Hille, and J. S. Weitz, “Leafj: an imagej plugin for semi-automated leaf shape measurement,” *American Journal of Botany*, vol. 98, no. 6, pp. 1029–1036, 2011.
- [2] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [3] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.
- [4] F. Meyer, “Topographic distance and watershed lines,” *Signal Processing*, vol. 38, no. 1, pp. 113–125, 1994.
- [5] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [6] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. Sánchez, “A survey on deep learning in medical image analysis,” *Medical Image Analysis*, vol. 42, pp. 60–88, 2017.
- [7] F. Milletari, N. Navab, and S. A. Ahmadi, “V-net: Fully convolutional neural networks for volumetric medical image segmentation,” in *2016 Fourth International Conference on 3D Vision (3DV)*, 2016, pp. 565–571.
- [8] N. Abraham and N. M. Khan, “A novel focal tversky loss function with improved attention u-net for lesion segmentation,” in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. IEEE, 2019, pp. 683–687.
- [9] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

- [10] A. A. Taha and A. Hanbury, “Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool,” *BMC Medical Imaging*, vol. 15, no. 1, p. 29, 2015.
- [11] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015, pp. 234–241.
- [12] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2117–2125.
- [13] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2881–2890.
- [14] L. C. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking atrous convolution for semantic image segmentation,” *arXiv preprint arXiv:1706.05587*, 2017. [En línea]. Disponible en: <https://arxiv.org/abs/1706.05587>
- [15] Python Software Foundation, *Python Language Reference, version 3.11.2*, 2023. [En línea]. Disponible en: <https://www.python.org/>
- [16] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” *Advances in Neural Information Processing Systems*, vol. 32, pp. 8024–8035, 2019.
- [17] P. Iakubovskii, “Segmentation models pytorch (smp),” 2019. [En línea]. Disponible en: [https://github.com/qubvel/segmentation\\_models.pytorch](https://github.com/qubvel/segmentation_models.pytorch)
- [18] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors, “scikit-image: image processing in python,” *PeerJ*, vol. 2, p. e453, 2014.
- [19] G. Bradski, “The opencv library,” *Dr. Dobb’s Journal of Software Tools*, 2000.
- [20] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, “Array programming with NumPy,” *Nature*, vol. 585, pp. 357–362, 2020.

- [21] Python Software Foundation, *Tkinter 8.6 Reference: A GUI for Python*, 2019. [En línea]. Disponible en: <https://docs.python.org/3/library/tkinter.html>
- [22] Intel Corporation, “Cvat: Computer vision annotation tool,” 2019, herramienta de anotación de imágenes. [En línea]. Disponible en: <https://github.com/opencv/cvat>
- [23] Microsoft Corporation, “Visual studio code,” <https://code.visualstudio.com/>, 2025.
- [24] Overleaf, Inc., “Overleaf, online latex editor,” <https://www.overleaf.com/>, 2025.
- [25] L. Lamport, *LaTeX: A Document Preparation System*. Reading, MA, USA: Addison-Wesley, 1994.
- [26] K. Schwaber and J. Sutherland, *The Scrum Guide*. Scrum.org, 2020. [En línea]. Disponible en: <https://scrumguides.org/>
- [27] J. Sutherland, *Scrum: The Art of Doing Twice the Work in Half the Time*. Currency, 2014.
- [28] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, “Image segmentation using deep learning: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3523–3542, 2021.
- [29] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [30] M. Tan and Q. V. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *International Conference on Machine Learning (ICML)*, 2019, pp. 6105–6114.
- [31] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510–4520.
- [32] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [33] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso, “Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations,” in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, 2017, pp. 240–248.
- [34] C. A. Price, B. J. Enquist, and V. M. Savage, “Global patterns in leaf vein anatomy and implications for functional ecology,” *Ecology Letters*, vol. 14, no. 4, pp. 357–365, 2011.

- [35] J. Rolland, K. Knebel, and L. Sack, “Leaf vein network architecture and its relation to leaf hydraulic efficiency in diverse plant species,” *New Phytologist*, vol. 205, no. 4, pp. 1517–1529, 2015.
- [36] scikit-image contributors, “Skeletonize — scikit-image documentation,” [https://scikit-image.org/docs/0.25.x/auto\\_examples/edges/plot\\_skeleton.html](https://scikit-image.org/docs/0.25.x/auto_examples/edges/plot_skeleton.html).
- [37] R. A. Slattery, E. A. Ainsworth, and D. R. Ort, “Leafj: An imagej plugin for semi-automated measurement of leaf traits,” *The Plant Journal*, vol. 93, no. 3, pp. 566–576, 2018.
- [38] B. Blonder, C. Vialle, L. P. Bentley, and B. J. Enquist, “Venation networks and the origin of the leaf economics spectrum,” *Ecology Letters*, vol. 14, no. 2, pp. 91–100, 2011.
- [39] E. Katifori, G. J. Szöllősi, and M. O. Magnasco, “Damage and fluctuations induce loops in optimal transport networks,” *Physical Review Letters*, vol. 104, no. 4, p. 048704, 2010.
- [40] L. P. Ronse De Craene *et al.*, “Leaf venation network analysis dataset and scripts,” 2023. [En línea]. Disponible en: <https://doi.org/10.5281/zenodo.8272938>
- [41] D. C. Montgomery, *Design and Analysis of Experiments*, 9th ed. Hoboken, NJ: John Wiley & Sons, 2017.
- [42] J. W. Tukey, “Comparing individual means in the analysis of variance,” *Biometrics*, vol. 5, no. 2, pp. 99–114, 1949.