

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE



“DISEÑO E IMPLEMENTACIÓN DE UN MODELO DE
MACHINE LEARNING APLICADO EN AGRICULTURA DE
PRECISIÓN, PARA IDENTIFICAR Y CLASIFICAR PALTOS
SEGÚN SU TAMAÑO”

MATÍAS FRANCIA CARRAMIÑANA

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Carlos Valle Vidal

Profesor Correferente: Ricardo Ñanculef

Abril - 2024

DEDICATORIA

Este trabajo de memoria se lo dedico a mi familia, pareja y amigos, así como a todos los que me apoyaron y me permitieron caminar a su lado.

AGRADECIMIENTOS

Quiero expresar mi profundo agradecimiento a mi profesor guía por su paciencia y apoyo constante a lo largo de esta memoria. Agradezco sinceramente a mis compañeros y amigos por su valiosa compañía. También quiero reconocer el apoyo incondicional de mi familia y seres queridos, quienes han estado a mi lado brindándome amor, comprensión y aliento durante todo este camino.

RESUMEN

Resumen— La detección precisa de la posición y el tamaño de las copas de árboles frutales es fundamental para el desarrollo de drones rociadores de pesticidas y el diagnóstico de las necesidades específicas de agua y nutrientes de los árboles. Aunque se han realizado investigaciones en varios cultivos, la atención dedicada a los paltos ha sido limitada, lo que ha restringido el avance de nuevas tecnologías en este campo. En este trabajo, se presenta un modelo de segmentación semántica de paltos basado en la arquitectura BEITv2, la cual se basa en aprendizaje generativo y emplea técnicas de Knowledge Distillation, Vector Quantization y Fine-tuning. Como resultado, se logró desarrollar un modelo que alcanzó una *overall class accuracy* de 92.8 %, una *mean accuracy* de 62.5 % y una *mean IoU* de 52.6 % sobre el conjunto de validación de los datos utilizados.

Palabras Clave— BEITv2; aprendizaje generativo; aprendizaje auto-supervisado; segmentación semántica; agricultura de precisión

ABSTRACT

Abstract— The precise detection of the position and size of fruit tree canopies is essential for the development of pesticide-spraying drones and for diagnosing the specific water and nutrient needs of trees. While research has been conducted on various crops, there has been limited attention given to avocados, hindering the advancement of new technologies in this area. This work introduces a semantic segmentation model for avocado tree canopies based on the BEITv2 architecture, leveraging generative learning and incorporating advanced Machine Learning techniques such as Knowledge Distillation, Vector Quantization, and Fine-tuning. The resulting model achieves an overall class accuracy of 92.8 %, a mean accuracy of 62.5 % and a mean IoU of 52.6 % over the validation split of the dataset used.

Keywords— BEITv2; generative Learning; self-supervised learning; semantic segmentation; precision agriculture

GLOSARIO

UAV: Unmanned Aerial Vehicles

ML: Machine Learning

DL: Deep Learning

RN: Redes Neuronales

RGB: Red Green Blue

LiDAR: Light Detection and Ranging

CNN: Convolutional Neural Network

FCN: Fully Convolutional Network

MIM: Masked Image Modeling

VQ: Vector Quantization

ÍNDICE DE CONTENIDOS

RESUMEN	IV
ABSTRACT	IV
GLOSARIO	V
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS	VIII
INTRODUCCIÓN	1
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	3
1.1 CONTEXTO	3
1.2 PROBLEMA DETECTADO	4
1.3 ACTORES INVOLUCRADOS	4
1.4 OBJETIVOS	5
1.5 CONTRIBUCIÓN	5
1.6 HIPÓTESIS	6
CAPÍTULO 2: ESTADO DEL ARTE	7
2.1 SEGMENTACIÓN SEMÁNTICA	7
2.1.1 ENFOQUES BASADOS EN DEEP LEARNING	10
2.1.2 U-NET	11
2.2 BEITv2: APRENDIZAJE GENERATIVO PARA TAREAS DE SEGMENTACIÓN	13
2.2.1 VQ-KD TOKENIZER	13
2.2.2 MASKED IMAGE MODELING FRAMEWORK	15
2.2.3 FINE-TUNING PARA TAREA DE SEGMENTACIÓN	16
2.3 SEGMENTACIÓN DE IMÁGENES DE CULTIVOS TOMADAS POR DRONES	16
2.3.1 UAV EN LA AGRICULTURA	17
2.3.2 DEEP LEARNING EN DETECCIÓN DE OBJETOS EN IMÁGENES TOMADAS POR UAV	17
2.3.3 ANÁLISIS DE TRABAJOS EN SEGMENTACIÓN DE CULTIVOS	20
2.3.4 DISCUSIÓN	23
2.4 DESCRIPCIÓN BASE DE DATOS	24
2.4.1 PREPROCESAMIENTO DE LOS DATOS	27
CAPÍTULO 3: SISTEMA PROPUESTO	31
3.1 METODOLOGÍA	31
3.2 PREPROCESAMIENTO DE LOS DATOS	31
3.3 ENTRENAMIENTO	32
3.3.1 HARDWARE	32
3.3.2 HIPERPARÁMETROS	33

3.3.3	MÉTRICAS	33
3.3.4	FUNCIÓN DE PÉRDIDA	34
CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN		36
4.1	RESULTADOS PARCIALES	36
4.1.1	U-Net	36
4.1.2	BEITv2	37
4.2	RESULTADOS MEJOR MODELO	38
4.3	COMPARACIÓN CON EL ESTADO DEL ARTE	39
CAPÍTULO 5: CONCLUSIONES		40
5.1	RESULTADOS	40
5.2	OBJETIVOS E HIPÓTESIS	41
5.3	LIMITACIONES	42
5.4	CONTRIBUCIONES	42
5.5	TRABAJO FUTURO	43
REFERENCIAS BIBLIOGRÁFICAS		44

ÍNDICE DE FIGURAS

1	Comparación entre tareas de procesamiento de imágenes.	6
2	Entradas y salidas de la segmentación semántica basada en textones	7
3	Descripción del flujo que realiza el sistema SuperParsing	8
4	Ejemplos de kernels con convoluciones dilatadas	11
5	Arquitectura U-Net	12
6	Estructura del tokenizador entrenado con Vector Quantization y Knowledge Distillation.	14
7	Estructura del <i>framework</i> MIM	16
8	Espectro electromagnético	19
9	Comparación del número y del ancho de las bandas en diferentes cámaras espectrales	20
10	Imagen del <i>dataset</i> con su respectiva máscara	24
11	Imagen del <i>dataset</i> con su respectiva máscara	25
12	Imagen del <i>dataset</i> con su respectiva máscara	25
13	Histograma de los píxeles de cada clase	26
14	Conjunto de imágenes georreferenciadas y proyecciones. Fuente: Elaboración propia.	30

ÍNDICE DE TABLAS

1	Parámetros del vuelo para capturar las imágenes.	26
2	Hiperparámetros utilizados en cada experimento para BEITv2. La configuración de modelo que obtuvo mejores resultados se encuentra destacada con *	33
3	Hiperparámetros utilizados en cada experimento para U-Net. La configuración de modelo que obtuvo mejores resultados se encuentra destacada con *	33
4	Resultados de los experimentos de U-Net sobre el conjunto de validación. El valor de la métrica con el mejor rendimiento entre los experimentos está resaltado en negrita , y el del mejor experimento está resaltado con (*).	37
5	Resultados de los experimentos de BEITv2 sobre el conjunto de validación. El valor de la métrica con el mejor rendimiento entre los experimentos está resaltado en negrita , y el del mejor experimento está resaltado con (*).	38
6	Resultados de las métricas por clase y métricas generales para los mejores experimentos de U-Net y BEITv2 sobre el conjunto de validación.	39
7	Resultados de la evaluación del modelo en comparación con otros trabajos del estado del arte. Las plantas se agrupan según su tipo: (A) Árbol, (B) Arbusto, (C) Cultivo. Notar que las unidades de medida de las columnas es porcentaje.	40

INTRODUCCIÓN

Un cultivo que ha ganado bastante importancia a nivel mundial es el palto (*Persea Americana*), no solo por la pulpa de sus frutos, sino que también por el uso de sus semillas y cáscaras, las cuales despertaron gran interés debido a su abundancia de componentes bioactivos, que incluyen ácidos grasos no saturados, vitamina C, B y E, fibra dietética, luteína y compuestos fenólicos, así como diversos pigmentos, como clorofila, antocianinas y carotenoides [Salazar-Reque *et al.*, 2023]. Por este motivo, la producción y el consumo global aumentó considerablemente en los últimos años, contribuyendo al desarrollo de nuevos productos comerciales, como helados, aceites, guacamole y productos de belleza [Colombo y Papetti, 2019].

Así como ha aumentado el consumo de ciertos productos como la palta en los últimos años, también se han desarrollado nuevas áreas de investigación, como lo es el caso de la agricultura de precisión. Esta área interdisciplinaria basada en la agronomía, la IA y la tecnología, se enfoca en hacer un diagnóstico de cada cultivo en particular. Este análisis incluye aspectos como la detección de enfermedades, la evaluación de la calidad del suelo y del agua de riego, así como la consideración de los factores climáticos del entorno. Un tema de gran interés en este campo es el uso de drones, los cuales permiten obtener rápidamente una visión general del cultivo, dando lugar al desarrollo de una gran cantidad de aplicaciones y modelos de aprendizaje automático, enfocados principalmente en el reconocimiento y segmentación de cultivos dentro de las imágenes que capturan, ya sean RGB de alta resolución [Diez *et al.*, 2021], multiespectrales [Adão *et al.*, 2017], o LiDAR [Reutebuch *et al.*, 2005].

A pesar de que existe una gran cantidad de trabajos enfocados en la tarea de segmentar árboles [Cong *et al.*, 2022] [Lu *et al.*, 2022] [Li *et al.*, 2024] [Ye *et al.*, 2022], no son muchos los enfocados en la segmentación de paltos. De este modo, los productores de paltas están en desventaja respecto a productores de otras plantas más estudiadas, ya que no tienen acceso a las mismas tecnologías, o no obtienen el mismo desempeño al utilizarlas. Por ello, resulta de particular interés trabajar en cómo mejorar las tecnologías de teledetección alrededor de los paltos. Por un lado, con la finalidad de obtener diagnósticos detallados de los nutrientes y agua que requiere cada sección del cultivo, tal como se propuso en [Salazar-Reque *et al.*, 2023]; y por otro, mejorar los sistemas de robots rociadores [Rosell y Sanz, 2012] [Narvaez *et al.*, 2017].

Una de las áreas de estudio más trabajadas en la agricultura de precisión es la segmentación semántica (Figura 1), ya que permite obtener una visión detallada de cada planta al interior de una imagen. En el caso particular de los árboles, se ha utilizado para extraer *features* que permiten reconocer deficiencias de nutrientes y agua en los cultivos. Así mismo, ha permitido el desarrollo de robots rociadores automatizados, especializados en la tarea de aplicar pesticidas y nutrientes de forma eficiente [Hejazipoor *et al.*, 2021] [Meshram *et al.*, 2022]. Por lo tanto, tal como se verá en la sección 1.4, el objetivo principal de esta investigación es diseñar y desarrollar un modelo de segmentación de árboles de palto en imágenes obteni-

das mediante drones, con el fin de mejorar las tecnologías de teledetección aplicadas a la gestión de cultivos de palto.

Dicho esto, se han hecho múltiples esfuerzos para mejorar la precisión de los modelos de segmentación de cultivos, utilizando diversas arquitecturas para lograrlo. Dentro de los modelos destacados por su desempeño, se encuentran: U-Net, Mask R-CNN, Mask Scoring R-CNN y Deeplab v3+ [Zhang *et al.*, 2020] [Kitano *et al.*, 2019] [Bouguettaya *et al.*, 2022] [Chen *et al.*, 2018] [Pang *et al.*, 2020], destacando U-Net [Ronneberger *et al.*, 2015], por su versatilidad y alta precisión en la segmentación de plantas bajo diversas condiciones ambientales y de iluminación [Zhang *et al.*, 2020] [Kitano *et al.*, 2019].

Este documento tiene la siguiente estructura: en el Capítulo 2 se muestra la revisión bibliográfica y el estado del arte del problema de segmentación semántica. En esta sección, se repasan brevemente los métodos que se han utilizado para la extracción de características en redes convolucionales, se realiza un análisis de algunos de los trabajos más relevantes para el problema de segmentación de cultivos, y se detalla el *dataset* utilizado para entrenar los modelos. En el Capítulo 3 se detalla la implementación del sistema propuesto, y se explica brevemente cómo se realizó la implementación de técnicas como *fine-tuning* y atención en la arquitectura usada. En el Capítulo 4 se muestra un análisis del rendimiento de los diversos modelos entrenados, y se realiza una comparación con los resultados obtenidos por otros trabajos e implementaciones. Por último, en el Capítulo 5 se detallan las conclusiones de la memoria.

CAPÍTULO 1

DEFINICIÓN DEL PROBLEMA

1.1. CONTEXTO

Tradicionalmente, la fumigación de árboles frutales se realizaba manualmente mediante máquinas llevadas en la espalda, una práctica ineficiente que consumía mucho tiempo, generaba altos costos y provocaba contaminación. Estos productos químicos, como pesticidas, fungicidas e insecticidas, se dispersaban en el aire y eran inhalados por los trabajadores, lo que causó daños importantes a su salud, provocando que incluso algunos de ellos perdiesen la vida [Meshram *et al.*, 2022].

Posteriormente, se comenzó a utilizar tractores para aplicar los químicos [Pandiselvam *et al.*, 2023]. Estos tractores estaban equipados con máquinas fumigadoras o pulverizadoras, lo que hacía que el proceso fuese más eficiente, especialmente en cultivos de baja altura, como la uva, las mandarinas o los duraznos. Sin embargo, no eran efectivos en árboles altos, como los paltos o los nogales. Para estos casos, se recurría frecuentemente al uso de avionetas [Randall, 1971], pero debido a la amplia dispersión del pesticida, se contaminaba el aire y se desperdiciaba una gran cantidad de producto.

En los últimos años, ha resultado de gran interés utilizar robots automatizados para realizar la fumigación de grandes cultivos, dando lugar a la colaboración entre agrónomos e informáticos en equipos interdisciplinarios que los desarrollen [Hejazipoor *et al.*, 2021] [Meshram *et al.*, 2022]. Sin embargo, para que estos robots puedan ajustar sus parámetros de acuerdo al cultivo en el que operan, es necesario que obtengan información detallada de los árboles. Esta información incluye datos geométricos como la densidad, el tamaño, la estructura y la posición de la corona de los árboles [Rosell y Sanz, 2012] [Narvaez *et al.*, 2017].

Las imágenes tomadas por drones a alturas relativamente bajas, son una de las formas más utilizadas para obtener parámetros de las coronas de un árbol y su follaje [Csillik *et al.*, 2018] [Malambo *et al.*, 2018]. Esto se debe a que comparado con las imágenes tomadas por satélites [Yin y Wang, 2019] o radares aéreos [Gomes *et al.*, 2018], las imágenes tomadas por UAV (Unmanned Aerial Vehicles) requieren menos procesamiento. De este modo, el primer paso para extraer los parámetros de la estructura y follaje de un árbol, es segmentar las fotografías, extrayendo la corona del fondo [Wu *et al.*, 2020] [McCarthy *et al.*, 2010] [Recio *et al.*, 2013].

Además de la caracterización geométrica para administrar eficientemente los pesticidas, esta información también es crucial para establecer estimaciones precisas de las necesidades de agua y nutrientes de los árboles, puesto que proporciona una base sólida para determinar dosis más sustentables de irrigación y fertilización [Rosell y Sanz, 2012] [Salazar-Reque *et al.*, 2023]. Tal como se observó en [Salazar-Reque *et al.*, 2023], las seg-

mentaciones de árboles en imágenes RGB, se pueden utilizar para obtener indicadores precisos del nivel de agua y nutrientes de cada región del árbol, lo que es de particular interés para los productores y agrónomos.

1.2. PROBLEMA DETECTADO

Si bien existe una gran cantidad de modelos de segmentación semántica para árboles frutales [Cong *et al.*, 2022] [Lu *et al.*, 2022] [Li *et al.*, 2024] [Ye *et al.*, 2022], en la práctica, obtener resultados “altamente precisos” es una tarea compleja. Esto se debe a que en entornos naturales, muchos factores pueden introducir ruido en las imágenes [Cong *et al.*, 2022], provocando que los modelos de segmentación no reconozcan bien las zonas de interés, y no obtengan un buen desempeño. Dicho esto, son muy pocos los trabajos que se han orientado al reconocimiento de paltos, lo que afecta negativamente al reconocimiento de los robots rociadores en estos cultivos.

1.3. ACTORES INVOLUCRADOS

Dentro de los actores que están involucrados en el desarrollo e implementación de esta memoria, se encuentran:

- **Agricultores:**

Son los usuarios directos de la tecnología propuesta, ya que se beneficiarían tanto del uso de robots que automaticen la aplicación de nutrientes y pesticidas, como del diagnóstico preciso del nivel de agua y de los nutrientes de cada árbol, lo que puede reducir considerablemente los costos asociados a la producción.

- **Organizaciones agrícolas:**

Desempeñan un papel importante en la difusión y promoción de estas nuevas tecnologías, y les pueden brindar apoyo técnico y capacitación a los agricultores interesados.

- **Proveedores de tecnología:**

Pueden aprovechar el desarrollo de estas nuevas formas de detección, para brindar servicios de mayor interés para los agricultores, lo que aumentaría el valor de su trabajo.

- **Investigadores:**

Los investigadores en el campo de la inteligencia artificial y la agricultura de precisión pueden utilizar como base este trabajo, para mejorar los sistemas de segmentación de paltos. Especialmente, aquellos que utilicen el dataset de [Amraoui *et al.*, 2022].

1.4. OBJETIVOS

Objetivo general: Consiste en diseñar e implementar un modelo que utilice técnicas de Machine Learning aplicadas a la agricultura de precisión, para identificar y clasificar paltos según su tamaño, en imágenes captadas por UAV (Unmanned Aerial Vehicle).

Mientras que **los objetivos específicos** son:

1. Reconocer las técnicas y los problemas que han surgido en el reconocimiento de imágenes de tipo *Bird View* (Vista de Pájaro) aplicado al rubro agrícola, así como la precisión y efectividad de los modelos que se han utilizado a la fecha.
2. Encontrar datos relevantes para el problema (datasets) que permitan entrenar correctamente un modelo de Machine Learning, para que pueda ser aplicado en un problema real.
3. Diseñar un modelo de reconocimiento de imágenes basado en técnicas de Machine Learning, tales como CNN, R-CNN o YOLO, para poder detectar y clasificar los árboles de paltas según su tamaño.
4. Evaluar la calidad de los resultados obtenidos, y compararlos con los del Estado del Arte.

De este modo, el alcance de esta memoria se limitará a la investigación, a la búsqueda de datos, y al diseño y evaluación de un modelo de Machine Learning. Este modelo se contrastará con los trabajos del Estado del Arte.

1.5. CONTRIBUCIÓN

Se propone el desarrollo de un modelo de segmentación semántica multiclase (Figura 1), en imágenes de palto capturadas por drones. Este modelo tiene como objetivo detectar los árboles individuales a partir de sus coronas, lo que facilita la creación de nuevos sistemas de diagnóstico de agua y nutrientes en cultivos, y permite mejorar el desempeño de robots rociadores en árboles frutales, especialmente en paltos.

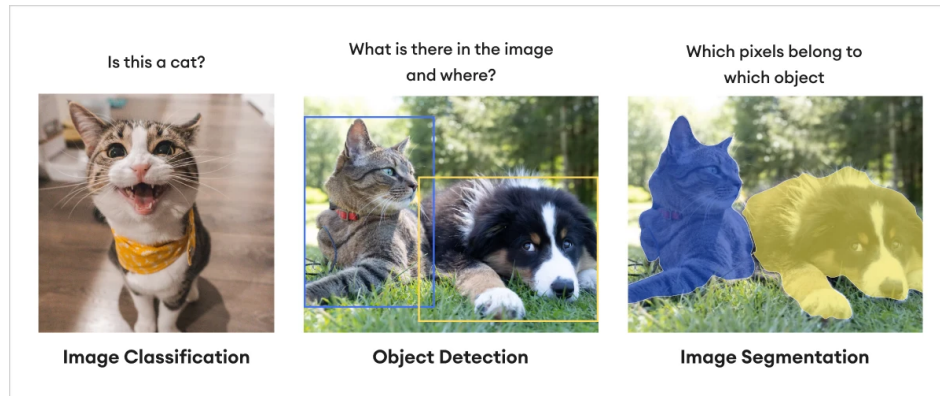


Figura 1: Comparación entre tareas de procesamiento de imágenes.
Fuente: SuperAnnotate

1.6. HIPÓTESIS

La hipótesis que se plantó en este trabajo es:

- Un método de aprendizaje generativo como BEITv2 permite mejoras en la tarea de segmentación semántica de árboles de palto por sobre un método convolucional como U-Net, en imágenes tomadas por drones.

CAPÍTULO 2

ESTADO DEL ARTE

En este capítulo se mostrará resumidamente el estado del arte del problema de segmentación semántica, haciendo énfasis en los métodos que trabajan con imágenes tomadas por UAV (*Unmanned Aerial Vehicles*) para el reconocimiento de árboles.

2.1. SEGMENTACIÓN SEMÁNTICA

Los primeros trabajos en obtener buenos resultados en segmentación semántica abordaron el problema utilizando textones, que son micro-estructuras en imágenes y vídeos, consideradas como los átomos de la percepción humana preatencional. Según la psicología, la percepción de formas se produce como un enfoque integrativo de características [Treisman, 1998], es decir, es un proceso que se desarrolla en etapas. La primera etapa, corresponde a la de procesamiento previo o preatencional, que consiste en un registro automático de características en el campo visual, y se realiza en forma paralela y simultánea en todos los objetos presentes en el campo visual. Estos textones se representan por un conjunto de pixels en las imágenes, de modo que algunos de los primeros enfoques en tener buenos resultados se basaron en la extracción de características *hand-crafted* en *patches* rectangulares de la imagen, los cuales fueron posteriormente etiquetados con diversos métodos de clasificación, como redes neuronales multicapa [He *et al.*, 2004], boosting [Shotton *et al.*, 2006] o random forest [Shotton *et al.*, 2008].

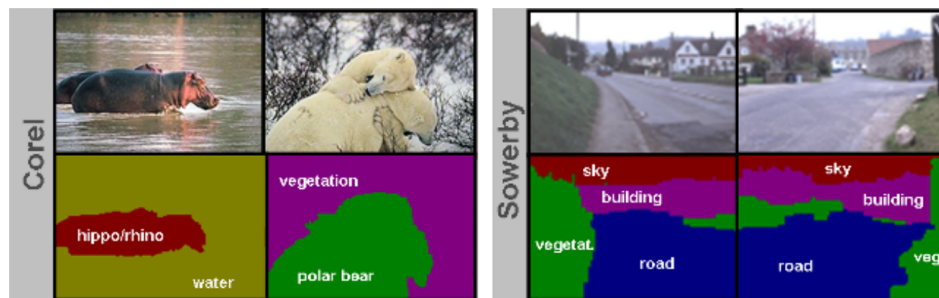


Figura 2: Entradas y salidas de la segmentación semántica basada en textones
Fuente: [Shotton *et al.*, 2006]

Posteriormente, para mejorar la etapa de clasificación, se utilizaron enfoques basados en *superpíxeles*. Los *superpíxeles* son grupos de píxeles que comparten propiedades similares, con los cuales se busca *sobresegmentar* una imagen. Esto permite obtener grupos de píxeles que contienen información semántica consistente con el sistema de reconocimiento visual humano, y que pueden servir de base para otros métodos de segmentación

[Lézoray *et al.*, 2017]. Son numerosos los trabajos que se realizaron con este enfoque, pero uno de los más destacados fue el realizado en [Tighe y Lazebnik, 2010]. En este trabajo, la imagen de *input* al sistema es relacionada con otras imágenes similares del *dataset* a través de diversos *features* globales. Luego, se divide la imagen en *superpíxeles* y se computa la verosimilitud de cada clase, en cada uno de ellos. De este modo, se usa la verosimilitud junto a un modelo que aporta contexto, para crear un etiquetado denso de la imagen original.

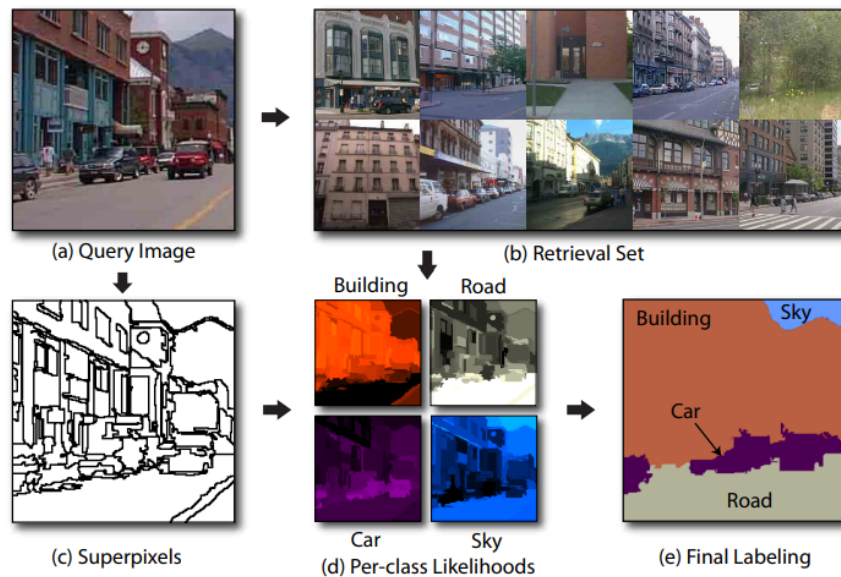


Figura 3: Descripción del flujo que realiza el sistema SuperParsing
Fuente: [Tighe y Lazebnik, 2010]

En lo que respecta al uso de características *hand-crafted*, se desarrollaron métodos que utilizaban aquellas que obtuvieron buenos resultados en el área de clasificación de imágenes, tales como: HOG (*Histogram of Oriented Gradients*) [Dalal y Triggs, 2005], que delimita contornos en la imagen utilizando los gradientes calculados con los valores de los píxeles de la imagen; SIFT (*Scale-invariant Feature Transform*) [Lowe, 2004], que encuentra *keypoints*, o segmentos comunes, entre diversas imágenes; y BOVW (*Bag of Visual Words*) [Csurka *et al.*, 2004], que se basa en el *Bag of Words*¹ utilizado en tareas NLP, para crear un histograma que cuente el número de ocurrencias de determinados patrones en un parche de la imagen, el cual se puede utilizar para clasificación o para encontrar elementos similares entre diversas imágenes.

En la búsqueda de características de mayor valor, se ha estudiado también el uso de aprendizaje no-supervisado para obtener más información de las imágenes. Sin embargo, ya que la segmentación no-supervisada no puede ser semántica, se utiliza principalmente para detectar regiones y bordes. Dentro de los métodos más populares que se han usado

¹BOW: se refiere a una representación que ignora el orden de las palabras en un texto, centrándose solo en su frecuencia de ocurrencia.

hasta la fecha, se encuentran los algoritmos de *clustering* (como *k-means* y *mean-shift*), el algoritmo Active Contour Models (ACM) [Kass *et al.*, 1988] y el algoritmo Watershed [Roerdink y Meijster, 2000].

De manera general, en procesamiento de imágenes, *k-means* se utiliza para agrupar píxeles alrededor de un centroide, y de este modo generar segmentos de la imagen. Así mismo, los algoritmos ACM segmentan las imágenes usando los bordes de las figuras presentes en ellas, y buscan bordes suaves que contengan los elementos de interés. Por otro lado, el algoritmo Watershed interpreta una imagen en escala de grises como un mapa de alturas, identificando cuencas hidrográficas como regiones de interés, basándose en las variaciones de colores en los píxeles. Este método utiliza los valores más bajos de la cuenca para establecer líneas divisorias (*watersheds*) entre cuencas adyacentes, y simula el llenado progresivo de las cuencas desde los puntos más bajos de la imagen, para que al conectar dos cuencas, se determine una línea divisoria.

A partir de 2012, la tendencia predominante en la segmentación semántica de imágenes ha sido la adopción de redes neuronales, impulsada por el éxito significativo del trabajo de Alex Krizhevsky en la clasificación de imágenes [Krizhevsky *et al.*, 2012]. Su contribución no solo sirvió como catalizador para nuevas investigaciones en esta área, sino también como punto de referencia para estudios en problemas afines, incluyendo la segmentación semántica. Los enfoques basados en Deep Learning para la clasificación de imágenes brindan robustez frente a variaciones y transformaciones en las imágenes de entrada. No obstante, esta fortaleza plantea un desafío en la segmentación semántica, al implicar una pérdida de información espacial producto de la reducción de dimensionalidad durante las capas de *subsampling*² de la arquitectura, la cual no es posible recuperar únicamente con el vector de salida de la red.

Entre los primeros trabajos que se realizaron en el área de segmentación semántica utilizando redes *deep*, y que obtuvieron buenos resultados, se encuentra [Farabet *et al.*, 2012]. En este trabajo proponen el uso de una red neuronal convolucional (CNN) con la misma arquitectura de aquellas redes utilizadas para clasificar imágenes, pero para detectar y etiquetar los diversos elementos de una imagen. Los resultados obtenidos fueron notables, ya que permitieron realizar un etiquetado denso de una imagen de 320x240 en menos de un segundo, incluyendo la extracción de características. Así mismo, se abordó el problema de la pérdida de información a través del uso de un entrenamiento supervisado, que compara las etiquetas predichas por el modelo con aquellas creadas manualmente, en secciones cuadradas de la imagen. Esto permitió obtener como salida de la red, un conjunto de características densas de la imagen de entrada.

El sistema propuesto en el trabajo de [Farabet *et al.*, 2012] se compone de diversas etapas, pero para entregar el etiquetado denso final cohesiona el resultado de las redes neuronales

²Técnica que implica la reducción de la resolución o dimensionalidad de una imagen al seleccionar un conjunto de píxeles representativos, con el objetivo de simplificar la cantidad de datos a procesar y extraer características relevantes.

convolucionales junto a las regiones generadas por superpíxeles, y utiliza CRF (*Conditional Random Fields*) para refinar el resultado de la segmentación, el cual es un modelo estocástico utilizado usualmente en Aprendizaje Automático para etiquetar y segmentar secuencias de datos o extraer información [Lafferty *et al.*, 2001].

Trabajos posteriores se enfocaron en crear redes *end-to-end* para crear inmediatamente un etiquetado semántico denso, sin la necesidad de usar otras técnicas de Machine Learning tradicional, como los superpíxeles o los CRFs.

2.1.1. ENFOQUES BASADOS EN DEEP LEARNING

Debido al gran éxito que tuvieron los enfoques basados en Deep Learning para clasificación de imágenes posterior al trabajo de [Krizhevsky *et al.*, 2012], se comenzó a utilizar cada vez más en problemas de segmentación semántica, tal y como se vio con el trabajo de [Farabet *et al.*, 2012], que utilizaba un enfoque basado en técnicas de aprendizaje profundo y de aprendizaje automático tradicional, para segmentar y etiquetar escenas. Sin embargo, aún existía el problema de pérdida de información espacial debido al *subsampling* por el que pasaba la imagen de entrada en las CNN, que eran las redes en las que estaba basado el trabajo de Krizhevsky. Si bien este comportamiento era esperado en la clasificación de imágenes, ya que permitía reducir la dimensionalidad del espacio de características y extraer las características más relevantes para la predicción de clase, en la segmentación semántica provocaba una pérdida irrecuperable de resolución de la imagen de entrada. Esto trajo consigo que se diese énfasis en crear nuevas arquitecturas de CNN que no dependiesen de la reducción de dimensionalidad para procesar las imágenes, surgiendo así, las Encoder-Decoder Networks [Ronneberger *et al.*, 2015] [Badrinarayanan *et al.*, 2017], las Fully Convolutional Networks (FCN) [Long *et al.*, 2015], y las redes basadas en Atrous Convolution [Chen *et al.*, 2017b].

Las primeras redes que propusieron un entrenamiento *end-to-end*³ para segmentación semántica mediante *deep learning*, fueron las FCN. Estas arquitecturas se basan en las arquitecturas de clasificación de imágenes, pero reemplazan las últimas capas *fully-connected* por capas convolucionales, lo que les permite conservar una mayor cantidad de información espacial que las redes de clasificación, a pesar del *subsampling* de las capas de *pooling*. Sin embargo, sólo haciendo esto sigue presente el problema. Es por esto, que el trabajo en que se propone, también plantea utilizar una capa de deconvolución al final de la red para recuperar las dimensiones originales, pero a pesar de esto, la resolución de salida sigue siendo baja.

Posteriormente, se propusieron las redes basadas en Atrous Convolution, también llamadas redes con Convoluciones Dilatadas, las cuales se basaron en el gran éxito que habían tenido las convoluciones *atrous* en problemas de segmentación semántica [Chen *et al.*, 2014]

³Se refiere a que realiza la tarea completa de principio a fin, desde que la imagen entra a la red, hasta que sale la segmentación.

[Chen *et al.*, 2017a] [Yu y Koltun, 2015] hasta la fecha de publicación del trabajo. Estas redes se proponen superar el problema de pérdida de información espacial a través del uso de convoluciones dilatadas, de forma muy similar a las FCN. Es decir, reemplazando las últimas capas de las redes entrenadas para clasificación de imágenes por capas de *upsampling*⁴, pero en este caso basadas en la inserción de espacios vacíos entre los valores de los kernels de la convolución, lo que permite controlar la resolución de la salida de cada capa, sin necesidad de aprender parámetros “extra”.

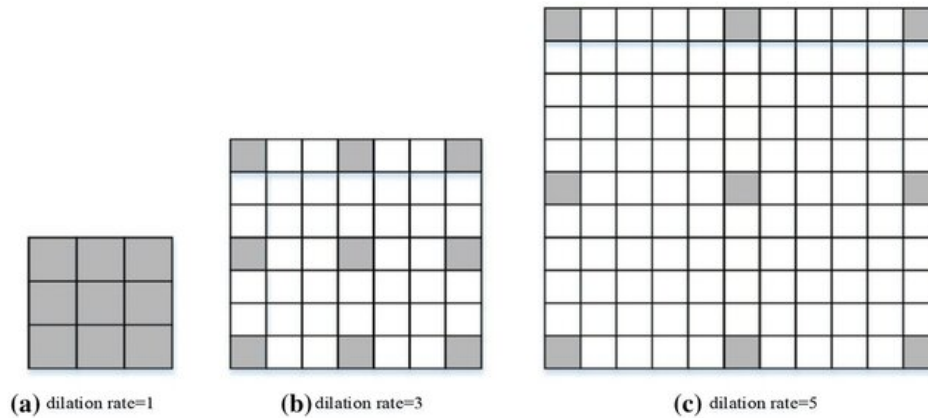


Figura 4: Ejemplos de kernels con convoluciones dilatadas

Fuente: [Zhang, 2022]

Así mismo, surgen las redes de tipo Encoder-Decoder como U-Net [Ronneberger *et al.*, 2015] o Segnet [Badrinarayanan *et al.*, 2017], las cuales se caracterizan por tener dos partes, como su nombre indica, un encoder, encargado de realizar un *subsampling* de la imagen de input; y un decoder, encargado de realizar un *upsampling* del *feature map*. A su vez, para no perder la información espacial durante las etapas de *pooling*, suelen utilizar *skip connections*, y han demostrado que es un enfoque muy eficaz para este tipo de problemas.

En particular, se revisará en mayor detalle la red U-Net, puesto que es la que se utiliza en esta memoria para obtener la segmentación semántica de los árboles de palto. Esta decisión se fundamenta en la gran cantidad de trabajos que se han realizado con esta arquitectura para segmentar imágenes, y en los buenos resultados que se han obtenido en las imágenes tomadas por drones.

2.1.2. U-NET

U-Net representa una de las primeras arquitecturas diseñadas para realizar un entrenamiento *end-to-end* específicamente orientado a la segmentación semántica. Este modelo, basado en las Fully Convolutional Networks (FCN) [Long *et al.*, 2015], ha sido adaptado para operar

⁴Son capas de la red que aumentan la resolución espacial de los mapas de *features* generados por la misma red

de manera eficiente con un conjunto reducido de imágenes de entrenamiento, produciendo segmentaciones más precisas. La idea principal consiste en complementar la sección de contracción de la red, afectada por capas de *pooling*, con una sección expansiva mediante la incorporación de operadores de *upsampling*, como las capas convolucionales traspuestas. Este enfoque permite aumentar la resolución de la salida, y la utilización de Skip Connections entre las capas de *downsampling* (Encoder) y *upsampling* (Decoder) asegura la retención de información espacial a lo largo de la red [Ronneberger et al., 2015].

El *encoder* sigue la estructura convencional de una red convolucional, con la aplicación repetida de convoluciones de 3x3 (sin *padding*), seguidas por unidades lineales rectificadas (ReLU) y operaciones de *pooling* de 2x2 con un *stride* de 2. De esta manera, tras cada capa de *pooling*, se duplica el número de canales de características.

Por otro lado, cada paso del *decoder* comprende un *upsampling* del mapa de características, seguido de una convolución de 2x2 que reduce a la mitad el número de canales, una concatenación con el mapa de características recortado correspondiente del *encoder*, y dos convoluciones de 3x3, cada una seguida por una activación ReLU. En la capa final, se emplea una convolución de 1x1 para asignar cada vector de características de 64 componentes al número deseado de clases.

Cabe recalcar que en esta arquitectura es esencial elegir un tamaño de imagen de entrada tal que el ancho y el alto sean divisibles por 32, asegurando que cada capa de *max-pooling* se aplique sobre un número par en los ejes *x* e *y*.

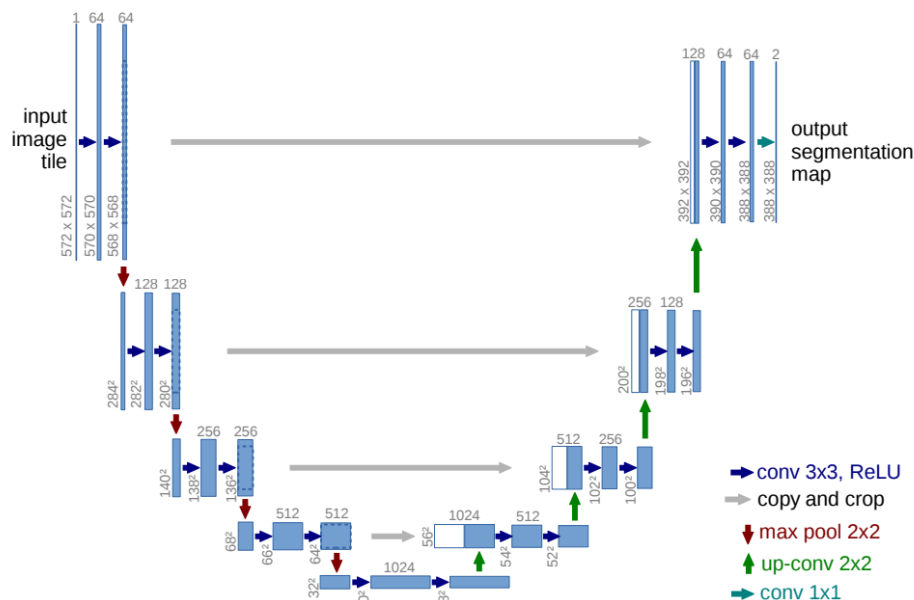


Figura 5: Arquitectura U-Net
Fuente: [Ronneberger et al., 2015]

2.2. BEITv2: APRENDIZAJE GENERATIVO PARA TAREAS DE SEGMENTACIÓN

Los modelos de aprendizaje generativo para procesamiento de imágenes, y específicamente para segmentación, nacen a partir del éxito que tuvieron estos métodos en el área de Procesamiento de Lenguaje Natural (NLP) y se basan en la estructura de los Autoencoders (AE). Si bien estos modelos suelen tener estructuras similares, han surgido muchas variaciones con el objetivo de mejorar los resultados tanto en tareas de clasificación y segmentación generales, agnósticas a un área en particular, como en tareas específicas.

A continuación, se describe de forma general el modelo BEITv2, que es sobre el cual se basa este trabajo de investigación.

BEITv2 es un modelo de aprendizaje generativo para procesamiento de imágenes, propuesto en el artículo [Peng *et al.*, 2022], que superó consistentemente al estado del arte en las tareas de clasificación de imágenes sobre el dataset ImageNet1K y en tareas de segmentación de imágenes sobre el dataset ADE20K [Gui *et al.*, 2024]. El modelo consta de tres componentes principales, las cuales deben ser entrenadas por separado y siguiendo una metodología particular, tal como se explicará a continuación.

2.2.1. VQ-KD TOKENIZER

Esta componente, como su nombre indica, es un *tokenizador* basado en Vector Quantization (VQ) y Knowledge Distillation (KD), el cual es entrenado de forma autosupervisada, sin requerir datos previamente etiquetados. Que este modelo utilice Vector Quantization quiere decir que las salidas que genera son *tokens* en un espacio discreto, los cuales son extraídos de un **codebook** de vectores predefinidos, en vez de generados en el espacio continuo. Por otro lado, que utilice Knowledge Distillation, o destilación del conocimiento, significa que para calcular la función de pérdida, utiliza la salida de otro modelo *teacher*, generalmente más grande y costoso, entrenado con más datos de los que uno dispone para entrenar el modelo *student*. Esto tiene como finalidad transferir parte del conocimiento del primero al segundo y reducir el costo de entrenamiento sin perder calidad en los resultados. En este trabajo, se siguieron las directrices del artículo original y se utilizó el modelo CLIP [Radford *et al.*, 2021] como *teacher*, dado que fue entrenado con *Representation Learning* y un gran conjunto de pares imagen-descripción que le permitieron generar un espacio de características de alta calidad.

En la Figura 6 se puede observar la estructura general del tokenizador. Los elementos principales son el Encoder, que corresponde a un modelo ViT [Dosovitskiy, 2020]; el *Codebook Embedding*, que es la estructura discreta que contiene vectores que sirven de *tokens*; y el Decoder, que es otro modelo ViT. El flujo de este componente es el siguiente. Una imagen entra al modelo y se divide en parches, que sirven de entrada para el ViT, el cual codifica cada uno de los parches en un vector en el espacio continuo. Luego, se calcula la similitud coseno entre cada parche codificado y cada vector del *codebook*, y el vector más cercano

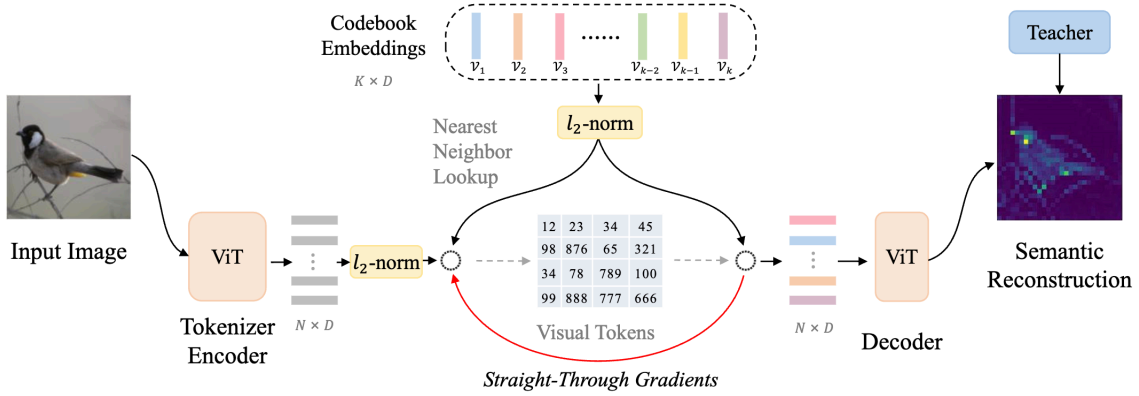


Figura 6: Estructura del tokenizador entrenado con Vector Quantization y Knowledge Distillation.

Fuente: [Peng et al., 2022]

reemplaza al vector original. De este modo, se obtiene un vector del conjunto discreto para cada parche de la imagen, que posteriormente se decodifica pasándolo a través del Decoder con el objetivo de generar una reconstrucción semántica de la imagen original. Para esto, se utiliza la salida del modelo *teacher*, al cual se le entrega la misma imagen como entrada, y se computa la pérdida del modelo *student*, cuyos gradientes fluyen hacia atrás mediante *backpropagation* y actualizan los pesos tanto del Decoder como del Encoder.

Como se puede observar, dado que el proceso de *quantization* no es diferenciable, los gradientes deben ser copiados desde la entrada del Decoder hacia la salida del Encoder. Cabe destacar que los vectores del *codebook* se inicializan con valores aleatorios y luego se actualizan durante el entrenamiento mediante medias móviles exponenciales para evitar el fenómeno de colapso del *codebook* y garantizar estabilidad durante el entrenamiento. Una vez finalizado el entrenamiento del modelo, se elimina la sección del Decoder y el tokenizador queda compuesto únicamente por el Encoder y el *Codebook Embedding*.

$$\max \sum_{x \in D} \sum_{i=1}^N \cos(\mathbf{o}_i, \mathbf{t}_i) - \|\text{sg}[\ell_2(\mathbf{h}_i)] - \ell_2(\mathbf{v}_{z_i})\|_2^2 - \|\ell_2(\mathbf{h}_i) - \text{sg}[\ell_2(\mathbf{v}_{z_i})]\|_2^2. \quad (1)$$

En la función de pérdida, el primer término penaliza que el *output* del Decoder del tokenizador sea diferente a la salida del modelo *teacher*. El segundo término penaliza que las salidas del Encoder sean distintas a los vectores del *codebook*, pero no actualiza los valores del Encoder, ya que utiliza *stop-gradient* (sg), es decir, “empuja” los valores del *codebook* hacia las salidas del Encoder. Por último, el tercer término también penaliza lo mismo que el segundo, pero utiliza el *stop-gradient* para mantener constante el valor de los vectores del *codebook*.

2.2.2. MASKED IMAGE MODELING FRAMEWORK

Luego de tener el tokenizador entrenado, hay que pre-entrenar una segunda componente de aprendizaje autosupervisado que corresponde al *framework* de Masked Image Modeling (MIM), el cual busca generar un espacio de características de buena calidad para tareas *downstream* de procesamiento de imágenes. Tal como su nombre indica, MIM se basa en dividir una imagen en parches y enmascarar aleatoriamente algún porcentaje de ellos, con la finalidad de que sean reconstruidos durante el entrenamiento. Hasta la fecha de confección de este trabajo, los enfoques MIM conforman el estado del arte para tareas de procesamiento de imágenes [Gui *et al.*, 2024].

En la Figura 7 se puede observar la estructura general del componente MIM de BEITv2. En el caso particular del trabajo [Peng *et al.*, 2022], se enmascara un 40 % de los parches extraídos al reemplazarlos por un *token* entrenable, el cual es el mismo para todos. Asimismo, se reemplazan los parches que no se enmascararán por sus correspondientes *tokens* entregados por el tokenizador, el cual, para la fase de inferencia, no cuenta con el Decoder. Además, se agrega un *token* [CLS] que se espera que contenga información global de la imagen, y no específica a los parches, y cuya finalidad es aportar información a la hora de reconstruir la imagen.

Una vez que se cuenta con todos los *tokens* de la imagen, se pasan como entrada a un *stack* de bloques ViT, los cuales sirven para generar representaciones de los datos en el espacio latente. Asimismo, a la salida del Encoder, se encuentra una capa completamente conectada que sirve para predecir los *tokens* de los parches faltantes a partir del *codebook* obtenido al entrenar el Tokenizador, y se computa la pérdida (1). Cabe recalcar que, durante la fase de entrenamiento de este componente, para poder agregar información global de la imagen en el parche [CLS], se crea un cuello de botella en el que se concatenan los parches codificados a la salida de la capa ℓ del Encoder con el *token* [CLS] que sale de la última capa. Esto permite que los *tokens* puedan ser pasados nuevamente por un *stack* de dos bloques ViT, y posteriormente por otro cabezal MIM (una capa completamente conectada), para computar la pérdida utilizando la misma función definida para el otro cabezal. Asimismo, los pesos entre los dos cabezales MIM son compartidos, y la pérdida total de esta fase del entrenamiento es la suma de $\mathcal{L}_{\text{MIM}}$ y $\mathcal{L}_{\text{MIM}}^{\text{C}}$.

Tal como se menciona en [Peng *et al.*, 2022], esto intuitivamente favorece que el *token* [CLS] agrupe la información global de la imagen, puesto que, para que el modelo disminuya la pérdida MIM adicional, este tiende a utilizar completamente los parámetros desde la capa $\ell + 1$ hasta la capa L . De este modo, el cuello de botella en el flujo de la información favorece que el *token* [CLS] contenga representaciones globales más confiables. Al igual que para el Tokenizador, luego del entrenamiento de este componente, se descarta el cuello de botella, el segundo cabezal MIM y la función de pérdida asociada.

$$\mathcal{L}_{\text{MIM}} = - \sum_{x \in \mathcal{D}} \sum_{i \in \mathcal{M}} \log p(\mathbf{z}_i | \mathbf{x}_i^{\mathcal{M}}), \quad (2)$$

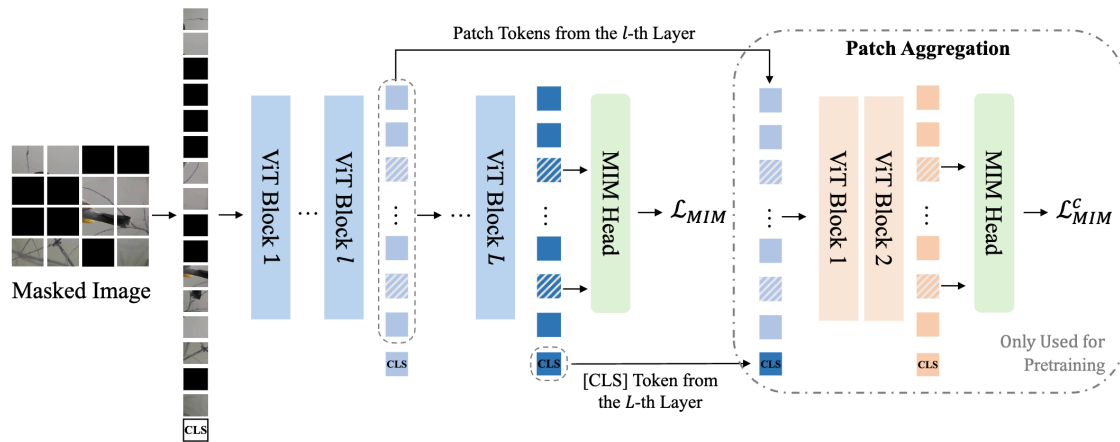


Figura 7: Estructura del *framework* MIM

Fuente: [Peng et al., 2022]

2.2.3. FINE-TUNING PARA TAREA DE SEGMENTACIÓN

Hasta ahora, al entrenar las componentes previas del modelo, es posible obtener un espacio de representación de alta calidad para tareas de procesamiento de imágenes, como se evidencia en los resultados de *linear probing* reportados en [Peng et al., 2022]. El *linear probing* consiste en agregar y entrenar una capa de clasificación luego del *encoder* del modelo, evaluando la calidad de los resultados de clasificación obtenidos. Este procedimiento permite analizar qué tan bien separados están los datos en el espacio de características y, por ende, determinar la calidad de los *features* generados.

Para realizar el *fine-tuning* en la tarea *downstream* de segmentación se utiliza el *framework* UPerNet [Xiao et al., 2018]. Este *framework* fue diseñado para entrenar modelos de segmentación semántica con etiquetas heterogéneas, permitiendo al modelo aprender a reconocer etiquetas semánticas en las imágenes. Esto es posible porque UPerNet soporta el uso de cualquier *backbone* de visión por computadora, el cual puede estar previamente entrenado. De este modo, para realizar el *fine-tuning* del modelo, es necesario entrenar únicamente las capas de UPerNet.

2.3. SEGMENTACIÓN DE IMÁGENES DE CULTIVOS TOMADAS POR DRONES

La introducción de los drones (UAVs) en la toma de imágenes y el desarrollo de la visión por computadora, sentaron las bases para el desarrollo de numerosos trabajos enfocados en procesar estas imágenes, para así obtener información valiosa del estado de la diversos

plantaciones, tal como la presencia de enfermedades [Shahi *et al.*, 2023] o la cuantificación de los árboles [Akca y Polat, 2022] en una determinada sección del cultivo.

2.3.1. UAV EN LA AGRICULTURA

Los drones, también conocidos como UAV (Vehículos Aéreos no Tripulados por sus siglas en inglés), han revolucionado la captura de imágenes en cultivos y bosques al poder estar equipados con una amplia variedad de sensores [Toro y Tsourdos, 2018], como cámaras de alta resolución [Diez *et al.*, 2021], cámaras multiespectrales [Adão *et al.*, 2017], o escáneres LiDAR ⁵, permitiendo obtener información detallada de la estructura y composición de los cultivos. Además, estos datos pueden ser utilizados para generar mapas precisos y de alta resolución de los terrenos fotografiados a un bajo costo relativo. Del mismo modo, los UAVs pueden sobrevolar territorios de acceso complicado y ajustar la resolución y escala [Grybas y Congalton, 2022] de los elementos capturados en las imágenes mediante el aumento o disminución de la altura de vuelo. Por ejemplo, para capturar imágenes de árboles individuales, se puede volar a baja altura, mientras que si se busca *mapear* una extensa área de terreno, se puede aumentar la altura de vuelo.

Los UAVs también son útiles para monitorear cambios en los terrenos o cultivos a lo largo del tiempo, ya que pueden capturar datos de forma regular para identificar áreas donde la vegetación está cambiando u observar cambios de interés en el terreno. Asimismo, los drones tienen la capacidad de volar a bajas altitudes con gran control y precisión, lo que resulta muy útil en la topografía [Jiménez-Jiménez *et al.*, 2021], permitiendo determinar la pendiente del suelo, así como la orientación y elevación de los árboles. Además, pueden utilizarse para crear modelos de riesgos naturales, como incendios forestales [Lewicki y Liu, 2021] o deslizamientos de tierra [Casagli *et al.*, 2017]. Sin embargo, si bien contribuyen significativamente al estado del arte de la cartografía de bosques y cultivos, los drones aún poseen un gran potencial de crecimiento debido a sus limitaciones, que incluyen una baja resistencia de vuelo, una capacidad de carga reducida y una gran sensibilidad a cambios en el entorno [Berra y Peppia, 2020].

2.3.2. DEEP LEARNING EN DETECCIÓN DE OBJETOS EN IMÁGENES TOMADAS POR UAV

Debido a la gran cantidad de imágenes generadas con el surgimiento y la comercialización de los drones, resultó fundamental desarrollar nuevas técnicas que permitieran procesarlas de manera más rápida y precisa. Es por esto que la llegada del Deep Learning (DL)⁶ resultó de particular interés, ya que ofreció métodos robustos para mejorar el mapeo de las superficies

⁵Una tecnología originalmente creada para *mapear* terrenos, que utiliza el tiempo que le toma a un láser ir a un objeto y regresar para reconocer la distancia a ellos y la posición [Reutebuch *et al.*, 2005].

⁶El DL se refiere a una red neuronal profunda, la cual es una configuración específica en la que las neuronas están organizadas en múltiples capas sucesivas [LeCun *et al.*, 2015].

cubiertas por drones. Esto se debe a que el aumento de las capas de una red neuronal mejora la potencia de expresión y el rendimiento de estos métodos, lo que podría producir un mayor nivel de abstracción [Ardila *et al.*, 2019]. Por otro lado, junto con el desarrollo del área del Deep Learning, también mejoraron los componentes de hardware y la captura de imágenes mediante drones, lo que resultó en un aumento de las aplicaciones de Deep Learning enfocadas en visión por computadora y procesamiento de imágenes, así como los *surveys* ⁷ de visión por computadora en imágenes tomadas por UAV [Osco *et al.*, 2021] [Yuan *et al.*, 2021] [Chehreh *et al.*, 2023].

En el *paper* [Osco *et al.*, 2021] se realizó un estudio cuantitativo y cualitativo de 250 *papers* que abordaron el problema de procesamiento de imágenes (ya sea de segmentación o de clasificación) tomadas por drones, mediante el uso de Deep Learning. En este trabajo se obtuvo que la mayor parte de los enfoques utilizados se basaba en la implementación de algún tipo de CNN (91,2 %); y a su vez, casi todos estos enfoques utilizaban arquitecturas ya establecidas, fueron pocos los *papers* que propusieron nuevos modelos y los compararon con las redes del estado del arte. A pesar de esto, también se encontraron algunas redes de tipo RNN (Recurrent Neural Networks) (8,8 %) compuestas por una arquitectura CNN-LSTM. Esto se debe a que los drones permiten obtener datos según la necesidad del usuario, es decir, que son capaces de tomar imágenes de una gran cantidad de fechas distintas, lo que puede ser trabajado como una serie de tiempo. Ejemplos de este tipo de problemas son el monitoreo de objetos estacionarios, como los ríos, la vegetación o las pendientes de un terreno.

En el campo específico de la segmentación de imágenes tomadas por UAV, existen diversos problemas que se suelen tratar, como el monitoreo ambiental [Blaschke *et al.*, 2000] [Yuan y Sarma, 2011], la detección y clasificación de cosechas [Kussul *et al.*, 2017] [Yang *et al.*, 2016] [Jadhav y Singh, 2018], la detección de especies en bosques [Dechesne *et al.*, 2017], y la clasificación de edificios y tipos de terrenos en suelos urbanos [Rottensteiner *et al.*, 2012] [Volpi y Ferrari, 2015] [Fang *et al.*, 2018].

Así mismo, son diversos los enfoques de Machine Learning utilizados en el campo de la segmentación de imágenes tomadas por UAV, pero como se mencionó anteriormente, la gran mayoría de ellos están fundamentados en redes neuronales convolucionales (CNN). Dentro de este tipo de redes, algunas arquitecturas han sido recurrentes a lo largo de numerosos trabajos, tanto en segmentación semántica en general, como en este campo específico de estudio. Esto se evidencia en [Yuan *et al.*, 2021], donde se lleva a cabo una revisión de los diferentes enfoques empleados para abordar el problema de segmentación de imágenes capturadas por drones, y se muestra que las arquitecturas más comunes han sido FCN, SegNet, U-Net, DeepLab y DenseNet.

Así como existen diversos enfoques de Machine Learning para abordar el problema de segmentación semántica, también existen diversos tipos de sensores que se pueden utilizar con

⁷ Son papers que recopilan y sintetizan información relevante de una serie de artículos científicos sobre un tema específico

los drones, los cuales capturan tipos distintos de información. Dentro de los más comunes en el área del mapeo de bosques y cultivos están las: cámaras RGB, cámaras multiespectrales, cámaras hiperespectrales, cámaras térmicas y los sensores LiDAR, y cada uno de estos puede ser usado individualmente o combinado. Revisando la literatura analizada por el *survey* [Chehreh *et al.*, 2023], uno puede notar que los sensores más utilizados fueron las cámaras RGB, seguidos por los sensores LiDAR y las cámaras multiespectrales.

Tal como se menciona en [Chehreh *et al.*, 2023], existen distintas estructuras en aquellos datos generados por los sensores, los cuales se agrupan en dos categorías, datos espectrales y datos estructurales. Los datos espectrales son aquellos generados por sensores térmicos, cámaras RGB, cámaras multiespectrales y cámaras hiperespectrales, los que son sensores pasivos, que dependen de la luminosidad externa y graban determinadas bandas del espectro electromagnético (8, 9). Mientras que los datos estructurales son aquellos datos generados por sensores LiDAR y que representan objetos mediante estructuras digitales tridimensionales, como las nubes de puntos; donde cada punto tiene coordenadas, intensidad, etiquetado semántico, y otros atributos relevantes.

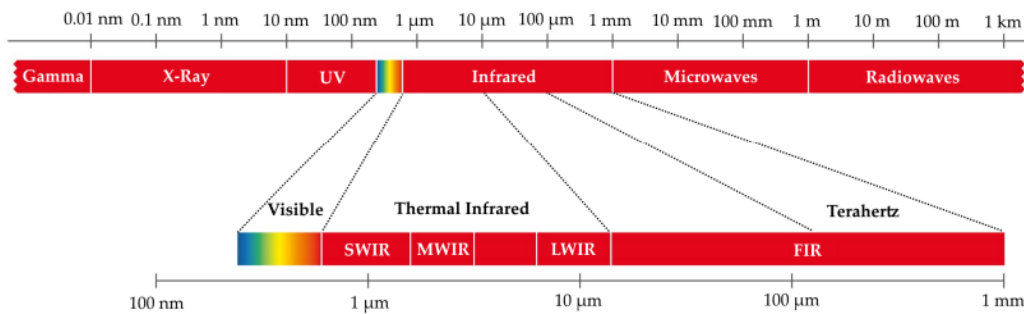


Figura 8: Espectro electromagnético
Fuente: [Chehreh *et al.*, 2023]

En el paper *survey* [Osco *et al.*, 2021] también se realiza un análisis cuantitativo de los sensores más utilizados en el área de reconocimiento de imágenes tomadas con drones usando Deep Learning, donde se menciona que el 52,4% de los trabajos revisados utilizaba cámaras RGB para obtener los datos, el 24,3% cámaras multiespectrales, el 17,8% cámaras hiperespectrales y el 5,5% sensores LiDAR. Lo que varía un poco con respecto a [Chehreh *et al.*, 2023], donde se menciona que si bien la mayor cantidad de *papers* revisados utilizaba RGB, el segundo lugar era ocupado por LiDAR. Esta diferencia se puede deber tanto a la diferencia del campo específico de estudio de cada *survey*, así como al año de publicación, puesto que la tecnología LiDAR ha ganado cada vez más terreno en el área de procesamiento de imágenes y modelado 3D.

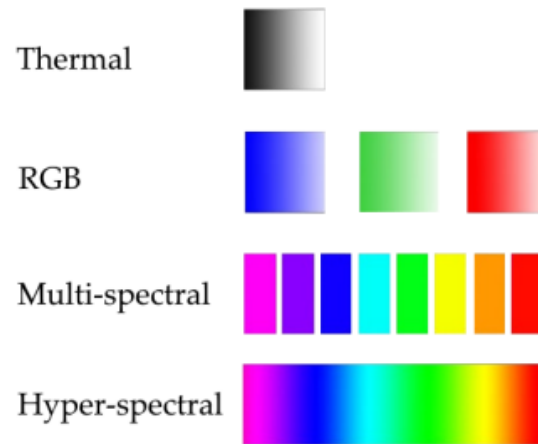


Figura 9: Comparación del número y del ancho de las bandas en diferentes cámaras espectrales

Fuente: [Chehreh *et al.*, 2023]

2.3.3. ANÁLISIS DE TRABAJOS EN SEGMENTACIÓN DE CULTIVOS

Es importante destacar que en el ámbito de la segmentación semántica de árboles, la obtención de datos es costosa y los conjuntos de datos disponibles son limitados, lo que dificulta la creación de un *benchmark* sólido para evaluar el rendimiento de los modelos diseñados para esta tarea. Como resultado, no es posible determinar con certeza qué modelos tienen el mejor rendimiento en general; más bien, deben ser evaluados de acuerdo a la tarea y al contexto específico en el que se aplican.

Teniendo esto en consideración, en el artículo [Bouguettaya *et al.*, 2022] se llevó a cabo una revisión de trabajos que informaron resultados en la segmentación semántica de árboles y plantas en imágenes capturadas por UAV. Para este análisis bibliográfico, se seleccionaron de [Bouguettaya *et al.*, 2022] los modelos que lograron los mejores puntajes de F1-Score/Accuracy en imágenes RGB. Estos modelos se describen brevemente a continuación.

En el estudio realizado en [La Rosa *et al.*, 2020], se aborda la necesidad de los productores de cítricos de monitorear sus huertos de manera frecuente, destacando la importancia de contar con herramientas automatizadas para analizar imágenes aéreas adquiridas por drones sobre las plantaciones. El trabajo propone un enfoque basado en Fully Convolutional Regression Networks y Multi-Task Learning para detectar árboles adultos, brotes de árboles y espacios vacíos en huertos de cítricos, con el objetivo de realizar un seguimiento del cultivo. Los resultados muestran que el método propuesto puede identificar árboles de naranja de ocho años con una precisión entre el 95 % y el 99 % en plantaciones comerciales de alta densidad donde las copas adyacentes se superponen. Sin embargo, se destaca que la detección de brotes de árboles y espacios vacíos sigue siendo un desafío, con Recall entre el 59 % y el 100 % para brotes y entre el 63 % y el 94 % para espacios vacíos.

En el trabajo realizado por [Kitano *et al.*, 2019], se aborda la automatización del proceso de contar plantas de maíz en imágenes tomadas por drones mediante el uso de Redes Neuronales y Aprendizaje Automático, con la finalidad de analizar el rendimiento del cultivo, ya que no es factible realizar un conteo manual. En el artículo se emplea la arquitectura de red U-Net para la segmentación de las imágenes, donde los elementos de interés son las plantas de maíz y el suelo. A pesar de que la metodología propuesta logró contar con éxito las plantas de maíz, se menciona que existen desafíos en la identificación correcta de otros objetos como malezas y paja, debido a formas y colores similares. El estudio concluye que la metodología propuesta puede ser una alternativa factible para contar plantas de maíz en imágenes tomadas a alrededor de 20 metros de altura, especialmente en campos con densidades de alrededor de 70,000 plantas/hectárea, y en etapas de crecimiento específicas, pero que sería ideal etiquetar también la paja y la maleza para que el modelo aprenda a reconocerlos. Bajo las condiciones mencionadas el modelo alcanzó una Accuracy total de hasta un 95.8 %.

En el trabajo propuesto por [Wu *et al.*, 2020] se presenta una técnica para la adquisición y análisis de datos en huertos de manzanos utilizando CNN. La técnica permite la detección automática y segmentación de árboles individuales, así como la medición de parámetros de la copa de los manzanos, tal como su perímetro, su ancho y el área de la proyección de la corona del árbol. Para entrenar y validar el modelo se recolectaron 50 imágenes de alta resolución durante la hibernación de los árboles, y se logró una Precisión y Recall del 91.1 % y 94.1 %, respectivamente, en la detección y conteo de los árboles. Además, se segmentaron las ramas con una precisión general del 97.1 % y se estimaron los parámetros de la copa con una precisión general superior al 92 %. El flujo del trabajo incluye el entrenamiento de un modelo de detección de objetos con imágenes tomadas por drones etiquetadas para identificar árboles individuales, la aplicación de algoritmos de segmentación semántica para segmentar los árboles de manzana del fondo, y la extracción de los parámetros de la copa de los árboles mediante estrategias de segmentación y cálculo automático. Además, debido a la poca disponibilidad de datos etiquetados, se utilizaron técnicas *data augmentation* para aumentar la cantidad de muestras etiquetadas y mejorar la precisión del modelo usado en la segmentación.

En la selva amazónica, uno de los ecosistemas más importantes es el aguajal de *Mauritia flexuosa*, cuya conservación se ve amenazada por la deforestación de la palma dominante, conocida como “aguaje”. Esta problemática es difícil de monitorear debido al acceso complicado a estos humedales, lo que motivó a que se realice el estudio [Morales *et al.*, 2018], en el cual se presenta el conjunto de datos MauFlex y se propone un método de segmentación y medición para áreas cubiertas por palmas de *Mauritia flexuosa* utilizando imágenes capturadas por drones. El método propuesto se compone de una CNN entrenada de forma *end-to-end* basada en la arquitectura Deeplab v3+ [Chen *et al.*, 2018], la cual fue entrenada con un conjunto representativo del dataset compuesto por 96 imágenes de 512 x 512, y que integra un Encoder, un módulo de Spatial Pyramid Pooling [He *et al.*, 2015] y un Decoder. Mientras que el conjunto de datos MauFlex contiene 25,248 parches de imagen de 512x512 píxeles y sus *ground truth* ⁸ correspondientes, obtenidos de imágenes adquiridas en

⁸Son las máscaras que contienen el etiquetado de las imágenes para tareas de segmentación semántica.

diferentes condiciones ambientales y lumínicas utilizando tres cámaras RGB distintas. Los resultados obtenidos muestran una Accuracy del 98.143 %, Precision del 96.599 %, y Recall del 95.556 %. En el *paper* se destaca también la capacidad del modelo propuesto para detectar tanto palmas completamente desarrolladas y aisladas, como palmas jóvenes o parcialmente cubiertas por otros tipos de vegetación.

El trabajo propuesto por [Machefer *et al.*, 2020] introduce un método diseñado para contar y dimensionar plantas de manera precisa y confiable en imágenes aéreas. El enfoque de investigación se centra en dos cultivos de baja densidad, papa y lechuga. El doble objetivo de contar y dimensionar lo logran mediante la detección y segmentación de plantas individuales utilizando la arquitectura pre-entrenada Mask R-CNN [He *et al.*, 2017] con *fine-tuning*. Además, el documento incluye una discusión detallada sobre la parametrización óptima para adaptar la arquitectura Mask R-CNN a la nueva tarea. Se concluye que el aprendizaje por transferencia puede ser utilizado de manera efectiva para reducir la cantidad de datos etiquetados requeridos, ya que se observó que la Mask R-CNN previamente entrenada en un cultivo de baja densidad mejoró su rendimiento después de ser entrenado en nuevos cultivos. Una vez entrenado para un cultivo específico, se demuestra que la solución Mask R-CNN supera a un algoritmo de visión por computadora ajustado manualmente. En términos de desempeño del modelo, se obtuvieron F1-score de 90.3 % para papas y 97.7 % para lechugas.

El conteo de plantas de maíz es una práctica común entre los agricultores para evaluar las condiciones de crecimiento de los cultivos y tomar decisiones de gestión durante la temporada. La metodología convencional implica una inspección manual que es lenta, laboriosa y limitada en términos de cobertura espacial. En el estudio propuesto por [Pang *et al.*, 2020], se desarrolló un sistema que emplea redes neuronales para detectar plantas de maíz en sus primeras etapas de crecimiento. En lugar de detectar individualmente cada brote, se procesa una fila completa de plantas, lo que reduce significativamente los requisitos de claridad en los campos de cultivo. El modelo propuesto se basa en una variante de Mask Scoring R-CNN [Huang *et al.*, 2019], que a su vez deriva de Mask R-CNN [He *et al.*, 2017]. Este modelo fue evaluado en datos recolectados durante varios años y en diferentes terrenos, logrando una precisión del 95.8 %. Además, se sugiere que debido a la rapidez de la inferencia del sistema, existe un potencial para desarrollar aplicaciones en tiempo real en el futuro.

La segmentación de hojas moradas en cultivos es esencial para monitorear el estrés de las plantas y contribuir al análisis fenotípico y a la estimación del rendimiento. Por ello, en el trabajo de [Zhang *et al.*, 2020], se utilizó un modelo U-Net para segmentar hojas moradas de colza (canola) durante las etapas tempranas del cultivo, basándose en imágenes RGB tomadas por drones. En este estudio, se implementó la arquitectura U-Net, pero los datos se utilizaron para entrenar diversos modelos: HSeg, Random Forest, SVM, SegNet y U-Net, y se realizó una comparación. De este modo, se obtuvo que los mejores resultados se alcanzaron con U-Net, logrando un F1-score del 90.29 % y una Intersección sobre Unión (IoU) del 82.41 %. Además, se evaluó el rendimiento del modelo U-Net en diferentes resoluciones y tamaños de parches, encontrando que los mejores resultados se alcanzaban con un tamaño de parche de 256 x 256 píxeles. El análisis de regresión reveló una fuerte correlación entre

las hojas moradas de la colza y el contenido de nitrógeno, sugiriendo el potencial de las hojas moradas como un método auxiliar para monitorear el estrés de nitrógeno en los cultivos. De este modo, en el *paper*, se plantea que los hallazgos respaldan la eficacia del modelo U-Net en la segmentación precisa de hojas moradas, proporcionando así un nuevo enfoque para el monitoreo del estrés de nitrógeno en los cultivos.

2.3.4. DISCUSIÓN

Los estudios revisados, como [Zhang *et al.*, 2020], [Bouguettaya *et al.*, 2022] [Chen *et al.*, 2018], y [Pang *et al.*, 2020], emplearon una variedad de modelos de segmentación, incluyendo U-Net, Mask R-CNN, Deeplab v3+, y variantes como Mask Scoring R-CNN, lo que muestra la diversidad de enfoques que se han utilizado para abordar la segmentación de cultivos. Además, se observó que los modelos como U-Net y sus variantes lograron altos niveles de precisión en la segmentación de plantas en diferentes condiciones ambientales y de iluminación [Zhang *et al.*, 2020] [Kitano *et al.*, 2019]. Sugiriendo que estos modelos podrían ser adecuados para la segmentación de coronas de árboles de palto, puesto que presentan desafíos similares en términos de complejidad y variabilidad.

Por otro lado, también se pudo observar que en el trabajo [Machefer *et al.*, 2020] se exploró la capacidad de transferencia de aprendizaje de los modelos, mostrando que los modelos pre-entrenados en diferentes tipos de cultivos pueden adaptarse con éxito a nuevas tareas de segmentación, lo que es de particular interés para el contexto de este trabajo, debido a que se cuenta con un conjunto de datos bastante limitado. Para esto, una opción sería transferir el conocimiento desde un modelo entrenado con el dataset propuesto en [Morales *et al.*, 2018], debido a su naturaleza y a la gran cantidad de datos con los que cuenta.

Se discutió además sobre el potencial de los modelos de segmentación para diversas aplicaciones en agricultura, como el monitoreo del crecimiento y brote de cultivos [Pang *et al.*, 2020] [Kitano *et al.*, 2019] [La Rosa *et al.*, 2020], la detección temprana de enfermedades o estrés en las plantas [Zhang *et al.*, 2020], y la optimización de prácticas de manejo agrícola [Wu *et al.*, 2020], aplicaciones de relevancia directa para el rubro agrícola.

Por último, hasta la fecha no son muchos los trabajos que se han enfocado en la segmentación de cultivos con aprendizaje generativo, y a la fecha de confección de este informe, el estado del arte sobre clasificación en el conjunto de datos ImageNet1K, y sobre segmentación en el conjunto ADE20K, son métodos de aprendizaje generativos [Gui *et al.*, 2024]. Esto podría hacer de este trabajo interesante para la comunidad.

2.4. DESCRIPCIÓN BASE DE DATOS

Los datos con los que se entrenará el modelo de segmentación semántica fueron obtenidos de [Amraoui *et al.*, 2022], el cual provee un conjunto de 984 imágenes tomadas por un dron DJI Phantom 4 en un cultivo de Marruecos. Siendo el *dataset* público de imágenes RGB de paltos tomadas por UAV, que cuenta con la mayor cantidad de datos etiquetados, alcanzando las 93 máscaras de 4864 x 3648 píxeles. Respecto a las clases presentes en las máscaras, hay 4: Fondo, Pequeño, Mediano y Grande. Siendo los 3 últimos referentes al tamaño de cada árbol.

En las Figuras 10 y 11 se pueden observar dos imágenes del *dataset* con sus respectivas máscaras. En ellas, se nota que existe traslape lateral y longitudinal, y en 1 se reporta que tienen valores de 65 % y 75 %, respectivamente. Además, se distinguen árboles de clase Pequeño (color rojo) y Grande (color azul). La segmentación de los árboles grandes no distingue por árbol individual, sino que por el conjunto de árboles, mientras que la de los árboles pequeños sí lo hace.

Luego, en la Figura 12 podemos observar que están presentes árboles de clase Grande y Mediano (color verde). Esta última si distingue al árbol en específico. Por otro lado, cabe recalcar que la calidad de las segmentaciones presentes en el *dataset* no son atentas al detalle, lo que podría dificultar el aprendizaje del modelo que se construya.

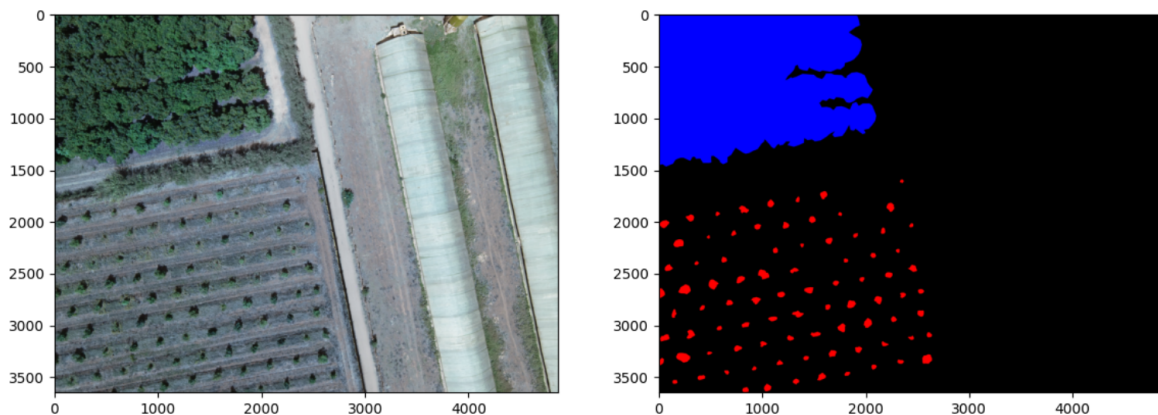


Figura 10: Imagen del *dataset* con su respectiva máscara
Fuente: Elaboración propia (basado en [Amraoui *et al.*, 2022])

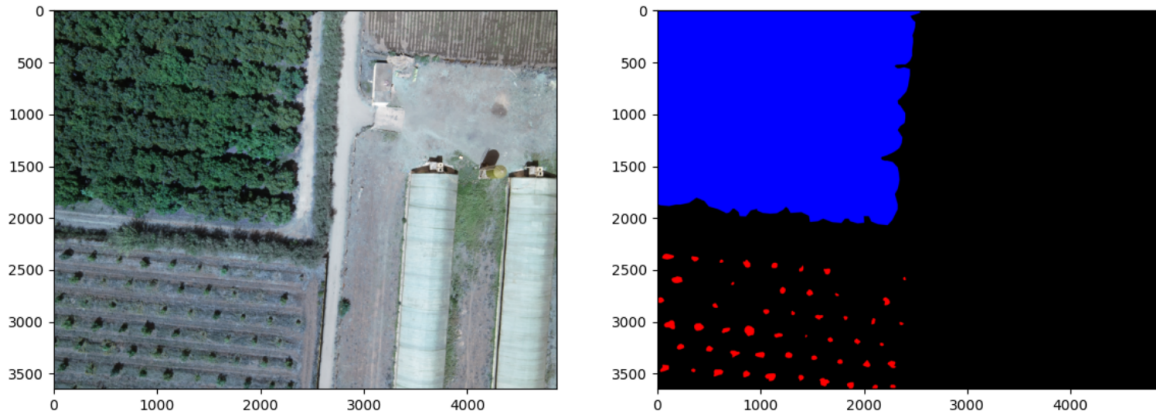


Figura 11: Imagen del *dataset* con su respectiva máscara
Fuente: Elaboración propia (basado en [Amraoui *et al.*, 2022])

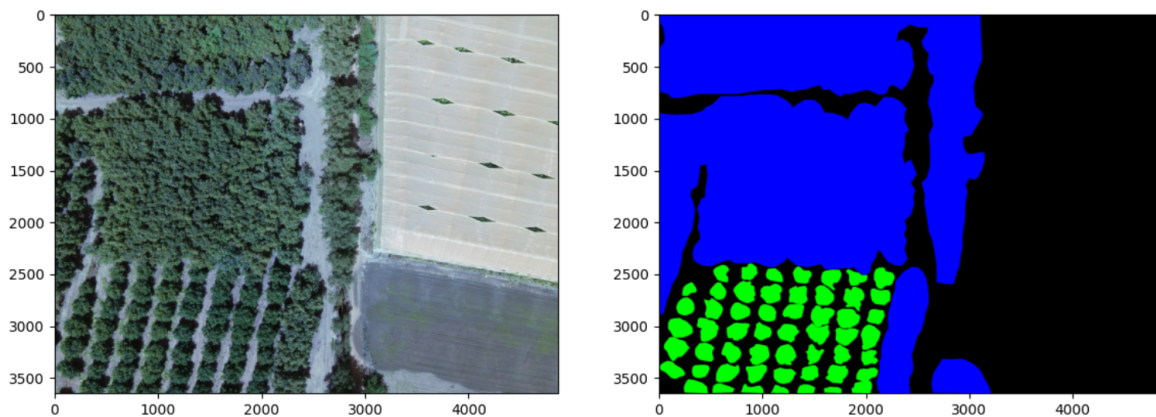


Figura 12: Imagen del *dataset* con su respectiva máscara
Fuente: Elaboración propia (basado en [Amraoui *et al.*, 2022])

En la Tabla 1 se presentan los parámetros del vuelo realizado.

Por otro lado, en la Figura 13 se realizó un histograma del *dataset*, en el que se muestra la cantidad de píxeles asociados a cada clase del conjunto. En ella, se puede apreciar que hay bastante desbalance de clases. Especialmente, se puede notar que las clases Medio y Pequeño tienen muy pocas ocurrencias dentro del conjunto de datos.

Parámetro	Valor
Altura de vuelo	90 m
Resolución estimada	2.7 cm/px
Superposición longitudinal	75 %
Superposición lateral	65 %
Dirección de vuelo	168°
Velocidad	9 m/s
Ángulo de la cámara	90°
Superficie	113 Ha
Tiempo de vuelo	50 min

Tabla 1: Parámetros del vuelo para capturar las imágenes.
Fuente: Elaboración propia

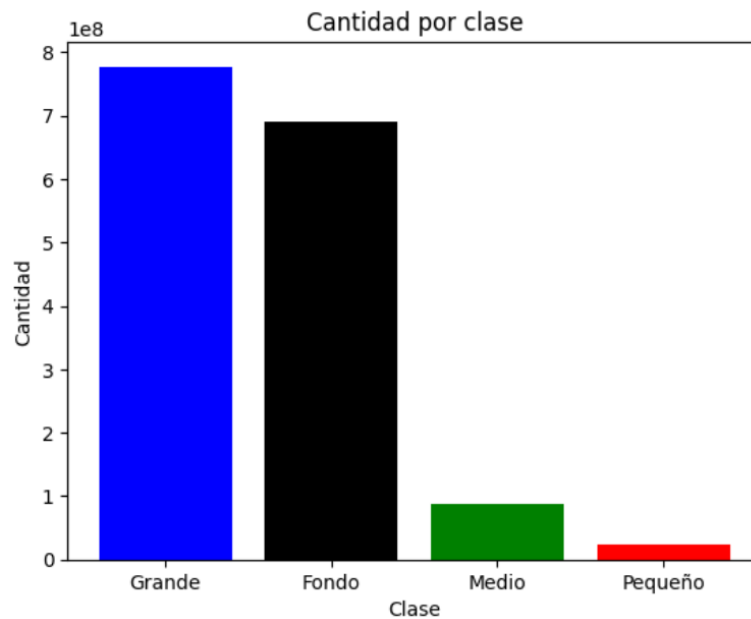


Figura 13: Histograma de los píxeles de cada clase
Fuente: Elaboración propia

2.4.1. PREPROCESAMIENTO DE LOS DATOS

Como se puede observar, en las imágenes originales del dataset hay mucha información repetida debido al traslape de unas con otras. Para ser precisos, cada imagen tiene alrededor de un 91.26 % de información repetida y un 8.74 % de información no repetida. La existencia de esta redundancia tiene especial importancia a la hora de entrenar un modelo de inteligencia artificial, específicamente de procesamiento de imágenes, puesto que conlleva un aumento de más del 90 % de los recursos computacionales o de tiempo a utilizar, y entrenar modelos de segmentación ya es una tarea cara. A su vez, también complejiza la separación del conjunto de datos en entrenamiento, validación y prueba, ya que si la redundancia de los datos entre distintos conjuntos es muy alta, es inevitable que el modelo sobreajuste y los valores de las métricas de evaluación no sean correctos.

Por esto, se volvió imprescindible utilizar alguna estrategia de preprocesamiento que permitiese reducir lo más posible la información repetida entre imágenes. Para poder diseñar esta estrategia se tuvo que analizar los datos en su contexto, es decir, en un mapa georreferenciado. Esto es posible debido a que las imágenes tomadas por drones suelen tener metadatos que contienen la latitud y longitud en las cuales fue tomada la imagen, específicamente de su centro. Mas esto no podía venir sin complicaciones, ya que si bien las imágenes en la realidad están prácticamente equiespaciadas (debido a la velocidad constante de vuelo), como se puede observar en las Figura 14a y 14b, la posición de las imágenes puede tener un error de hasta 10 metros, lo que se traduce en un error de alrededor del 50 % de la imagen, puesto que cada imagen cubre una superficie de alrededor de 20 metros a lo ancho.

Por lo tanto, se tuvo que abordar de alguna forma este problema primero. Para esto, se observó que si se agrupaban las imágenes por fila, se podría utilizar alguna técnica de reducción de error como mínimos cuadrados, o directamente equidistanciar los puntos en el **eje x**, tomando como referencia las imágenes de los bordes. Y a su vez, se podría reducir un poco el error en el **eje y**, si se tomaba el promedio de las alturas. Esto es posible debido a la premisa de que los datos se tomaron a velocidad constante, tal como se indica en la Tabla 1.

Sin embargo, agrupar las imágenes por fila no algo que se pudiese realizar directamente, debido a que como se puede observar en la Figura 14a, los puntos están rotados por un determinado ángulo, y además contienen un error en su posición, lo que dificultaba considerablemente la tarea. Así mismo, las posiciones estaban en ángulos respecto a los paralelos y meridianos de la tierra, por lo que no existía una claridad de qué sistema de coordenadas utilizar.

El enfoque que se utilizó para sobrellevar estos problema fue utilizar el realizar una proyección de los puntos en un plano, mediante el uso de una librería de Python para procesamiento de datos georreferenciados llamada *PyProj*, y luego se utilizó el ángulo de rotación que se puede ver en la Tabla 1, para rotar los puntos, mediante el uso de una matriz de rotación en dos dimensiones, dada por:

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix},$$

donde θ es el ángulo de rotación (en radianes).

Luego, para rotar un conjunto de puntos $\mathbf{P}$, la operación se define como:

$$\mathbf{P}_{\text{rotado}} = \mathbf{P} \cdot R(\theta)^T,$$

donde $R(\theta)^T$ se refiere a la matriz de rotación traspuesta.

Una vez que se obtuvo el conjunto de puntos rotados, se procedió a agrupar los puntos en *clusters* dados por las filas en las cuales se encontraban. Para esto, se definió empíricamente un error en la posición del **eje y**, y se iteró sobre todos los puntos, determinando a qué grupo pertenecían, hasta tener todos los puntos agrupados. Una vez hecho esto, se redujo el error en la posición de cada imagen tal como se mencionó anteriormente, y se procedió a generar la unión de las superficies, con la finalidad de generar un conjunto de datos sin información repetida.

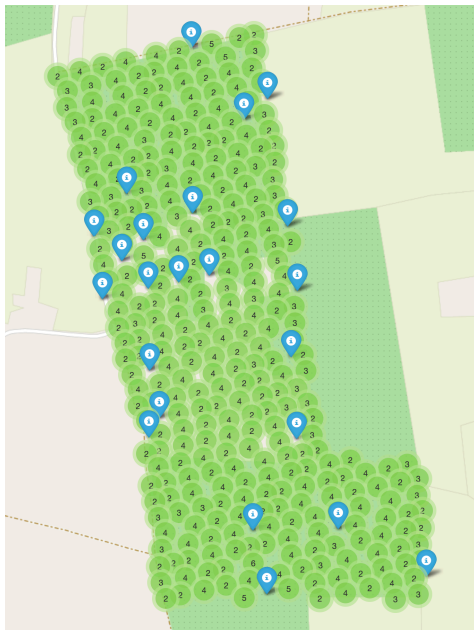
Para generar la unión de las superficies, se empleó un método de unión de áreas. Esto se debe a que cada imagen puede ser vista como un polígono de cuatro aristas (rectángulo), y se tiene tanto la posición de su centro, como la relación entre los píxeles y la superficie Tabla 1 (2.7 cm/px). Haciendo uso de esta información, se pudo computar la posición de cada arista de las imágenes, así como proyectarlas en un espacio cuya unidad de medida eran los metros, para tener la referencia de cuáles eran los límites de cada imagen en el espacio. Así, finalmente se computó la unión de las áreas.

Cabe notar que, debido a que se quería utilizar un modelo que utilizaba tanto aprendizaje no supervisado para generación de *tokens* y para entrenar un componente de aprendizaje generativo, como aprendizaje supervisado para realizar *fine-tuning* enfocado en la tarea de segmentación. Y dado que se contaba con tres conjuntos de imágenes separados y anotados en el *dataset* (para entrenamiento, validación y prueba), que contenían traslape y que no fueron anotados minuciosamente, se decidió hacer uso de la redundancia en esa información, y no eliminar la sobreposición de esas imágenes.

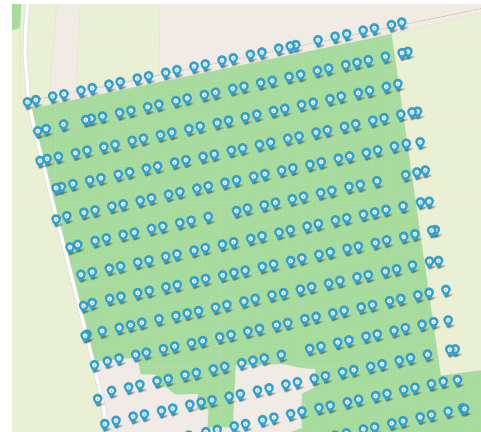
Para generar el conjunto de imágenes sin intersección se inicializó un conjunto con las uniones de las áreas de los conjuntos de entrenamiento, validación y prueba (es decir, todos los etiquetados), y se iteró sobre cada imagen del conjunto de datos sin anotaciones, revisando cuáles eran las secciones de la imagen que no intersectaban con lo que ya estaba en el conjunto, y se guardaron los límites. De este modo, se pudieron generar *bounding boxes* para cada imagen Figura 14d, las cuales fueron posteriormente utilizadas para extraer parches del tamaño necesario en el entrenamiento de los modelos, en este caso **224 píxeles**. Mientras que en las imágenes etiquetadas se sacaron parches de la imagen de tamaño completo, sin

importar la redundancia.

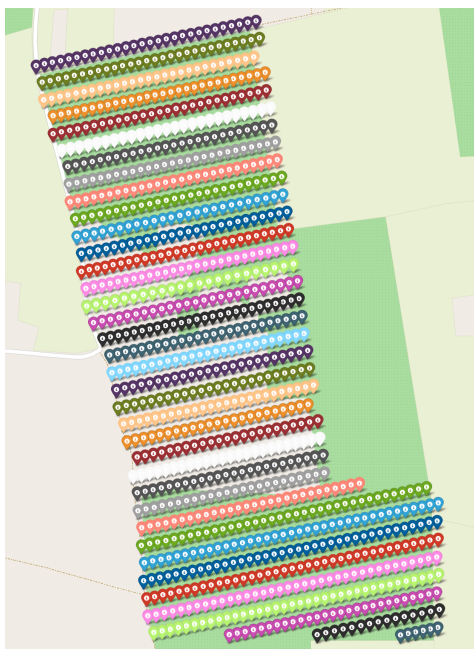
Tal y como se puede observar en la Figura 14d, la unión de las áreas de las imágenes, si bien es aproximada, cumple bastante bien con el propósito de eliminar información redundante en los datos, lo que satisface las necesidades planteadas inicialmente.



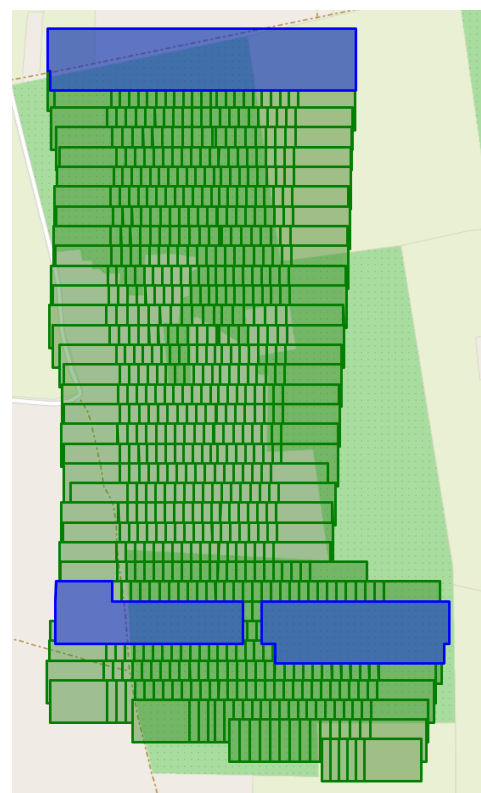
(a) Posiciones originales imágenes georreferenciadas



(b) Detalle posiciones originales



(c) Posiciones georreferenciadas corregidas y clusterizadas por filas



(d) Unión de la proyección polígonos que representan las imágenes y su posición

Figura 14: Conjunto de imágenes georreferenciadas y proyecciones. Fuente: Elaboración propia.

CAPÍTULO 3

SISTEMA PROPUESTO

A partir de la discusión llevada a cabo en la sección Estado del Arte, se propone usar el modelo BEITv2 [Peng *et al.*, 2022] para validar las hipótesis del trabajo. Esta decisión se sustenta en que el modelo superó consistentemente al estado del arte en las tareas de segmentación de imágenes en el dataset ADE20K y en la clasificación de imágenes sobre el dataset ImageNet1K [Gui *et al.*, 2024].

3.1. METODOLOGÍA

Para obtener el segmentador de imágenes requerido para este trabajo, se comenzó pre-procesando los datos y generando las particiones correspondientes para los diversos tipos de entrenamiento, tanto supervisado como no supervisado. Luego, se entrenó cada una de las componentes de BEITv2 siguiendo las indicaciones mencionadas en 2.1.2, partiendo con el Tokenizador, siguiendo con el pre-entrenamiento del modelo MIM, y terminando con el *fine-tuning* del modelo con el bloque de segmentación de UPerNet.

Para la evaluación del modelo y la validación de las hipótesis, se entrenó a su vez el modelo U-Net, con las mismas imágenes utilizadas para el entrenamiento supervisado de BEITv2, ya que U-Net no admite datos no etiquetados. Esto se realizó con la finalidad de comprobar si existían mejoras dadas por el uso de un método de aprendizaje generativo, por sobre uno de aprendizaje supervisado.

3.2. PREPROCESAMIENTO DE LOS DATOS

Tal como se mencionó en la sección de descripción del *dataset*, los datos utilizados para entrenar el modelo están compuestos por imágenes RGB de 4864×3648 píxeles, lo cual es demasiado grande en memoria como para ser utilizado directamente en el entrenamiento. Por este motivo, tal y como se describió en la sección 2.4, se realizó un preprocesamiento de las imágenes y sus respectivas máscaras, en el que cada imagen fue subdividida en parches de 224×224 píxeles.

Para las componentes del modelo que debían ser entrenadas con un enfoque no supervisado, se generó un conjunto de datos compuesto exclusivamente por imágenes no anotadas, a las cuales se les redujo el traslape siguiendo la metodología propuesta en la sección 2.4. Por otro lado, para las componentes que requerían entrenamiento supervisado, se utilizaron las imágenes anotadas tal y como estaban en el *dataset* original. Finalmente, el conjunto de datos de prueba se mantuvo inmaculado y solo se utilizó para evaluar el desempeño final de los

modelos. Debido a estos diferentes requerimientos, fue necesario generar varias particiones del conjunto de datos, las cuales se organizaron de la siguiente manera:

- **Carpeta *fine_tuning*:** Esta partición contiene las imágenes y anotaciones necesarias para el entrenamiento supervisado durante la etapa de *fine-tuning*. Está organizada en las siguientes subcarpetas:
 - **images:** Contiene las imágenes en formato RGB, divididas en dos subconjuntos:
 - **training:** Imágenes destinadas al entrenamiento supervisado.
 - **validation:** Imágenes utilizadas para la validación del modelo.
 - **annotations:** Contiene las máscaras correspondientes a las imágenes, organizadas en:
 - **training:** Máscaras utilizadas durante el entrenamiento.
 - **validation:** Máscaras utilizadas para la validación.
- **Carpeta *self_supervised*:** Esta partición incluye únicamente imágenes no anotadas, que fueron utilizadas para entrenar las componentes del modelo con enfoques no supervisados, como el VQ-KD Tokenizer y el preentrenamiento MIM. Todas las imágenes están almacenadas en la subcarpeta *images*.
- **Carpeta *test*:** Este conjunto de datos se mantuvo inmaculado y únicamente fue utilizado para evaluar el desempeño final del modelo. Está compuesto por:
 - **images:** Contiene las imágenes RGB del conjunto de prueba.
 - **masks:** Contiene las máscaras correspondientes a las imágenes de prueba.

3.3. ENTRENAMIENTO

La metodología utilizada para entrenar el modelo fue primero entrenar el Tokenizador con aprendizaje autosupervisado, con la finalidad de generar un *codebook* especializado para la tarea a realizar. Luego realizar inferencia sobre ese modelo sin el Decoder para generar los *tokens* con los que pre-entrenar el modelo MIM, y entrenar éste último con aprendizaje generativo, con el objetivo de generar un espacio de características de buena calidad para tareas *downstream*. Finalmente, hacer *fine-tuning* sobre el modelo UPerNet con el backbone de BEITv2 inicializado con los pesos previamente obtenidos, y evaluar los resultados.

3.3.1. HARDWARE

El entrenamiento y evaluación de los modelos se llevaron a cabo utilizando una máquina virtual alojada en la nube.

- **Tarjeta Gráfica:** NVIDIA RTX A100-80G-PCIe.
- **CPU:** 28 núcleos.
- **Memoria RAM:** 120 GB.

3.3.2. HIPERPARÁMETROS

Los hiperparámetros que se utilizaron para el entrenamiento de BEITv2 se encuentran en la Tabla 2, y fueron definidos mediante Random Search, al igual que los utilizados para U-Net, que se encuentran en la Tabla 3. En BEITv2, el Tokenizador y el Backbone MIM se pre-entrenaron por 100 épocas, y con 7 épocas de calentamiento. Mientras que el *fine-tuning* del modelo con el bloque de UPerNet fue realizado por 60K iteraciones. Por otro lado, U-Net también fue entrenado por 100 épocas, pero no tuvo épocas de calentamiento.

Tabla 2: Hiperparametros utilizados en cada experimento para BEITv2. La configuración de modelo que obtuvo mejores resultados se encuentra destacada con *.

Hiperparámetro	Experimento 1	Experimento 2	Experimento 3	Experimento 4*	Experimento 5	Experimento 6
<i>Tokenizador</i>						
Tamaño del Codebook	256	128	256	128	256	128
Tamaño del Embedding	32	16	64	32	16	64
Learning Rate	5e-5	1e-5	1e-4	1e-5	5e-5	1e-4
Learning Rate Mínimo	1e-5	1e-6	5e-5	5e-6	1e-5	5e-5
Tamaño del Batch	256	256	256	256	512	512
<i>MIM</i>						
Learning Rate	1.5e-3	1e-4	5e-5	1.5e-3	1e-4	5e-5
Clip	3.0	1.0	3.0	1.0	1.0	3.0
Scale	0.1	0.01	0.1	0.01	0.1	0.1
<i>Fine-tuning</i>						
Learning Rate	1e-4	5e-5	5e-6	5e-5	1e-4	5e-6
L ayer Decay	0.75	0.7	0.8	0.75	0.8	0.7

Tabla 3: Hiperparámetros utilizados en cada experimento para U-Net. La configuración de modelo que obtuvo mejores resultados se encuentra destacada con *.

Hiperparámetro	Experimento 1	Experimento 2	Experimento 3	Experimento 4*	Experimento 5	Experimento 6
Tamaño del Batch	32	32	16	16	8	8
Learning Rate	0.01	0.001	0.01	0.0005	0.01	0.001

3.3.3. MÉTRICAS

Las métricas que se utilizaron para evaluar el desempeño de los modelos en este trabajo fueron *accuracy (acc)* e *intersection over union (IoU)* por cada clase, y *mean IoU (mIoU)*,

mean accuracy (mAcc) y *accumulated accuracy (aAcc)* como métricas generales calculadas sobre todas las clases.

Las fórmulas de cada una de estas métricas son las siguientes:

1. IoU por clase (Intersection over Union)

$$\text{IoU}_i = \frac{\text{Área de Intersección}_i}{\text{Área de Unión}_i} = \frac{|\text{Predicción}_i \cap \text{Ground Truth}_i|}{|\text{Predicción}_i \cup \text{Ground Truth}_i|}. \quad (3)$$

2. Accuracy por Clase

$$\text{Acc}_i = \frac{\text{Número de píxeles correctamente clasificados para la clase } i}{\text{Número total de píxeles en la clase } i}. \quad (4)$$

3. Accuracy Global

$$\text{aAcc} = \frac{\text{Número total de píxeles correctamente clasificados}}{\text{Número total de píxeles en el conjunto de datos}}. \quad (5)$$

4. Promedio de IoU

$$\text{mIoU} = \frac{1}{N} \sum_{i=1}^N \text{IoU}_i, \quad (6)$$

donde N es el número de clases.

5. Promedio de Accuracy

$$\text{mAcc} = \frac{1}{N} \sum_{i=1}^N \text{Acc}_i, \quad (7)$$

donde N es el número de clases.

3.3.4. FUNCIÓN DE PÉRDIDA

Las funciones de pérdida que se utilizaron para entrenar el Tokenizador y el modelo de MIM fueron las mismas establecidas por [Peng *et al.*, 2022] y mencionadas en la Sección 2.1.2. Por otro lado, la función de pérdida utilizada para realizar el *fine-tuning* de UPerNet para la tarea de segmentación fue Cross Entropy Loss (8), al igual que la utilizada para el entrenamiento de U-Net.

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C y_{i,c} \log(p_{i,c}), \quad (8)$$

donde N es el número total de píxeles en la imagen (o batch), C es el número de clases, $y_{i,c}$ es un indicador binario que toma el valor 1 si el píxel i pertenece a la clase c , y 0 en caso contrario, y $p_{i,c}$ es la probabilidad predicha para el píxel i y la clase c , obtenida a través de una función *softmax*. Esta función de pérdida pondera las probabilidades logarítmicas de las clases verdaderas en cada píxel, sumándolas sobre todas las clases y píxeles, y normalizando el resultado por el número total de píxeles.

CAPÍTULO 4

VALIDACIÓN DE LA SOLUCIÓN

En esta sección se presentan los resultados obtenidos al evaluar los diversos modelos en su conjunto de validación, y se realizan comparaciones pertinentes para confirmar las hipótesis. Es importante notar que, si bien la clase fondo estaba presente en las anotaciones de las imágenes, ésta no fue considerada para el cálculo de las métricas.

4.1. RESULTADOS PARCIALES

En esta sección se presentarán los resultados sobre el conjunto de prueba de los modelos BEITv2 y U-Net, con la finalidad de comparar si BEITv2, un método que se basa principalmente aprendizaje generativo, es capaz de superar los resultados de U-Net sobre el *dataset* de interés. U-Net fue escogido como *baseline* debido a que es un modelo de redes convolucionales vástamente usado en el mundo de la segmentación semántica, y que pese a su simplicidad suele tener buenos resultados.

A continuación, se detallan los mejores resultados obtenidos en cada uno de los experimentos sobre el conjunto de validación. Estos fueron escogidos según la métrica mIoU más alta obtenida, puesto que en lugar de utilizar Early Stopping, se guardó el modelo con mejores resultados.

4.1.1. U-Net

En la tabla 4 se pueden observar los resultados que obtuvo U-Net para la tarea de segmentación multiclase, considerando la existencia de 3 clases: Small, Medium y Large. Acá se puede notar que el modelo predice muy bien sobre la clase Large, ya que las métricas IoU y Accuracy son bastante altas. Sin embargo, para varias configuraciones de hiperparámetros el modelo no es capaz de predecir la clase Small, y tiene muy malos resultados para la clase Medium. Esto se debe probablemente al desbalance de clases visto en la Figura 13, a que la clase Small es fácil de confundir con la maleza de la clase Background, y a que la clase Medium es difícil de diferenciar de la clase Large, la cual tiene una cantidad mucho mayor de píxeles asociados en el *dataset*. Además, a pesar de que los hiperparámetros modificados entre experimentos fueron sólo dos, se puede apreciar una alta variabilidad entre los resultados por clase, y se nota particularmente en la clase Small.

Tabla 4: Resultados de los experimentos de U-Net sobre el conjunto de validación. El valor de la métrica con el mejor rendimiento entre los experimentos está resaltado en **negrita**, y el del mejor experimento está resaltado con (*).

Métrica	Experimento 1	Experimento 2	Experimento 3 (*)	Experimento 4	Experimento 5	Experimento 6
<i>Métricas por Clase</i>						
IoU (Large)	94.52	95.78	96.41	94.15	92.67	96.24
IoU (Medium)	10.32	9.83	13.14	9.34	12.67	7.28
IoU (Small)	0.00	0.00	16.84	0.00	11.37	0.00
Accuracy (Large)	96.48	97.51	98.12	96.78	94.83	98.01
Accuracy (Medium)	34.78	27.35	23.57	26.84	44.16	20.54
Accuracy (Small)	0.00	0.00	21.89	0.00	13.79	0.00
<i>Resumen de Métricas</i>						
Mean IoU	5.32	4.23	14.82	4.62	12.87	3.14
Mean Accuracy	17.45	14.72	22.41	13.62	28.94	10.38
Overall Accuracy	94.72	94.18	96.32	94.78	92.65	96.14

4.1.2. BEITv2

En la Tabla 5 se pueden observar los resultados que obtuvo este modelo para la tarea de segmentación multiclase, considerando las mismas clases mencionadas anteriormente para U-Net. Acá se puede notar que el modelo es muy bueno prediciendo la clase Large, predice la clase Medium medianamente bien, y la clase Small en general no la puede predecir correctamente. Esto no es una sorpresa, puesto que tal como se ha mencionado en repetidas ocasiones en este trabajo, las clases están muy desbalanceadas en términos del número de píxeles, tal como se puede observar en 13, y la tarea de predecir los árboles pequeños es de mayor dificultad que la de predecir los más grandes. Por otro lado, a diferencia de U-Net, no se ve una variabilidad tan alta en los resultados obtenidos por clase para diversos experimentos. Sin embargo, si bien no es tan alta, también existe una alta varianza en los resultados de la clase Small, y nos indica que es sensible a la modificación de los experimentos.

Tabla 5: Resultados de los experimentos de BEITv2 sobre el conjunto de validación. El valor de la métrica con el mejor rendimiento entre los experimentos está resaltado en **negrita**, y el del mejor experimento está resaltado con (*).

Métrica	Experimento 1	Experimento 2	Experimento 3	Experimento 4 (*)	Experimento 5	Experimento 6
<i>Métricas por Clase</i>						
IoU (Large)	94.38	94.02	92.42	92.85	91.98	91.72
IoU (Medium)	38.97	44.44	45.32	50.02	40.88	47.10
IoU (Small)	13.72	4.55	15.17	15.00	15.21	12.35
Accuracy (Large)	98.35	98.78	96.00	96.51	95.43	95.42
Accuracy (Medium)	53.99	54.02	58.87	66.00	49.88	59.28
Accuracy (Small)	18.29	5.66	26.92	25.01	29.16	23.45
<i>Resumen de Métricas</i>						
Mean IoU	49.02	47.67	50.97	52.62	49.36	50.39
Mean Accuracy	56.88	52.82	60.60	62.51	58.15	59.38
Overall Accuracy	94.08	93.86	92.32	92.77	91.81	91.61

4.2. RESULTADOS MEJOR MODELO

Para poder escoger las configuraciones de los mejores modelos, la decisión se basó en las métricas **mIoU** y **mAcc**. La métrica IoU promedio (**mIoU**) tiene gran importancia en el ámbito de la segmentación semántica y es especialmente útil cuando se trabaja con conjuntos de datos desbalanceados, como lo es este caso. Esta métrica no solo favorece cuando se predice correctamente la clase de un píxel, sino que también penaliza los falsos positivos, lo que cumple de cierto modo el rol de la métrica F1-Score en otras tareas. A continuación, se presentan los mejores resultados obtenidos en los experimentos de los modelos U-Net y BEITv2.

Tal como se puede observar en la Tabla 6 los resultados para la clase Large fueron buenos en ambos modelos, aunque el modelo U-Net superó a BEITv2 por algunos puntos. Esto se justifica en que el modelo priorizó la información que le permitía reconocer los árboles grandes (Large), por sobre aquella necesaria para reconocer los medianos (Medium) y pequeños (Small). En cambio, si bien los resultados de BEITv2 no fueron igual de buenos que U-Net en la predicción de la clase Large, fue consistentemente mejor a la hora de predecir la clase Medium, y esto se refleja en las métricas generales, especialmente en Mean IoU y Mean Accuracy, que son considerablemente más altas que en el método convolucional. La clase Small no se podría decir que fue predicha mejor por un método u otro. Sin embargo, volviendo a los resultados de la Tabla 5 y la Tabla 4, se puede apreciar que los resultados para esta clase tienen una menor variabilidad en BEITv2, lo que es importante debido a que en cuatro de los seis experimentos, el modelo U-Net nunca (o casi nunca) predijo correctamente un píxel de la clase de árbol pequeño.

Tabla 6: Resultados de las métricas por clase y métricas generales para los mejores experimentos de U-Net y BEITv2 sobre el conjunto de validación.

Métrica	U-Net (Clase)			BEITv2 (Clase)		
Clase	Large	Medium	Small	Large	Medium	Small
IoU	96.41	13.14	16.84	92.85	50.02	15.00
Accuracy	98.12	23.57	21.89	96.51	66.00	25.01
Métrica General	U-Net			BEITv2		
Mean IoU	14.82			52.62		
Mean Accuracy	22.41			62.51		
Overall Accuracy	96.32			92.77		

4.3. COMPARACIÓN CON EL ESTADO DEL ARTE

Como se mencionó previamente, en el ámbito de estudio abordado en esta memoria, no existe un *benchmark* que permita comparar diferentes modelos y arquitecturas de forma estandarizada. Por este motivo, para evaluar el rendimiento del mejor modelo desarrollado en este trabajo, se realizará una comparación con los resultados reportados en estudios similares revisados en la literatura. Si bien esta comparación proporcionará una idea general del desempeño del sistema propuesto, es importante tener en cuenta que los modelos comparados no fueron evaluados con los mismos datos ni exactamente en la misma tarea. Esto último se debe a que la tarea realizada en este trabajo corresponde a una segmentación multiclase con clases altamente desbalanceadas. En cambio, los otros trabajos abordan segmentación binaria, lo que suele ser un problema de menor complejidad, al menos en el contexto de segmentación de cultivos. Asimismo, las métricas reportadas también difieren ligeramente, ya que en esos trabajos se priorizaron métricas de clasificación binaria, mientras que en este trabajo, al ser multiclase, se priorizó el reporte de métricas como Mean IoU o Mean Accuracy.

En la tabla 7 se presentan los resultados obtenidos por los estudios revisados en la literatura, junto con los resultados obtenidos en esta memoria.

Tabla 7: Resultados de la evaluación del modelo en comparación con otros trabajos del estado del arte. Las plantas se agrupan según su tipo: (A) Árbol, (B) Arbusto, (C) Cultivo. Notar que las unidades de medida de las columnas es porcentaje.

Referencia	Planta (Tipo)	Overall Acc.	Precision	Recall	F1-Score	Mean IoU
[La Rosa <i>et al.</i> , 2020]	Cítricos (A)	98.8	99.2	98.4	98.8	-
[Kitano <i>et al.</i> , 2019]	Maíz (C)	99.4	-	-	-	-
[Wu <i>et al.</i> , 2020]	Manzana (A)	97.1	84.2	84.5	84.2	-
[Morales <i>et al.</i> , 2018]	Mauritia Flexuosa (A)	98.0	96.7	95.6	96.1	-
[Machefer <i>et al.</i> , 2020]	Papa (C)	-	99.7	82.5	90.3	-
[Machefer <i>et al.</i> , 2020]	Lechuga (C)	-	100.0	95.4	97.7	-
[Pang <i>et al.</i> , 2020]	Maíz (C)	95.8	82.8	-	-	-
[Zhang <i>et al.</i> , 2020]	Canola (C)	-	94.0	90.0	91.56	-
BEITv2 (Este trabajo)	Paltas (A)	92.77	-	-	-	52.62

Nota: La métrica Overall Accuracy es equivalente para tareas de segmentación binaria y multiclase, ya que mide el porcentaje de píxeles correctamente clasificados en ambas configuraciones.

CAPÍTULO 5

CONCLUSIONES

En esta sección se presentarán las conclusiones del trabajo de memoria realizado, dividiéndose en varias secciones. En la primera, se detallarán brevemente los resultados obtenidos y las posibles causas de los problemas visualizados. En la segunda, se verificarán las hipótesis planteadas al inicio del documento. En la tercera, se mencionarán las limitaciones encontradas durante el desarrollo del trabajo. En la cuarta, se destacará brevemente la contribución del estudio. Finalmente, se concluirá con la quinta sección, donde se expondrá el trabajo futuro.

5.1. RESULTADOS

En este trabajo se implementó un modelo de segmentación semántica del Estado del Arte, el cual está basado en técnicas recientes que aprovechan al máximo los recursos computacionales de los nuevos componentes de *hardware*. Con este modelo, se lograron resultados que pudieron superar en gran medida a los obtenidos por el método convolucional U-Net. A pesar de esto, aún queda camino en esta línea de investigación, puesto que con la cali-

dad y distribución de los datos utilizados, no fue posible identificar satisfactoriamente los árboles de palto pequeños en las imágenes. A pesar de esto, sienta una buena base para el trabajo futuro, ya que incluso con la limitada cantidad de datos sobre árboles pequeños y medianos, y las diversas complejidades que tiene reconocer cada uno de ellos; se pudieron obtener resultados que indican que el modelo aprendió algunas de sus características. Por lo tanto, los resultados son prometedores en caso de mejorar el *dataset* o de implementar alguna metodología que permita sobrellevar el *gap* entre las clases.

5.2. OBJETIVOS E HIPÓTESIS

A continuación se referencia cada uno de los objetivos planteados al comienzo de este trabajo para comprobar que se cumplió:

1. En el Estado del Arte, se reconocieron las técnicas que se han utilizado y los problemas que han surgido en el reconocimiento de imágenes de tipo vista de pájaro (o *Bird View*) en el rubro agrícola. Asimismo, también se mencionaron la precisión y la efectividad de cada método, siendo medidos usando métricas específicas de esta área de investigación.
2. Se encontraron datos que permitieron entrenar un modelo de inteligencia artificial de segmentación de paltos en imágenes tomadas por drones, y este podría ser implementado sin problemas en aplicaciones del mundo real, tal como reconocimiento de árboles para rocío de pesticidas o nutrientes, o de estimación del crecimiento. Especialmente para árboles grandes y medianos.
3. Se diseñó e implementó un modelo de Machine Learning especializado en la tarea de segmentación de imágenes, el cual permite detectar las coronas de los árboles de paltos.
4. Se evaluaron los resultados del modelo implementado, y en la Tabla 7 se compararon con el Estado del Arte en tareas de segmentación de cultivos en imágenes tomadas por drones.

La hipótesis que se planteó al comienzo de este trabajo fue:

- Un método de aprendizaje generativo como BEITv2 permite mejoras en la tarea de segmentación semántica de árboles de palto por sobre un método convolucional como U-Net, en imágenes tomadas por drones.

Si bien hubo casos en los que el modelo U-Net tuvo mejores resultados que el modelo BEITv2, éste último fue, en términos generales, considerablemente mejor. Esta aseveración se basa

en las métricas Intersección sobre la Unión Promedio (**mIoU**) y Accuracy Promedio (**mAcc**), las cuales alcanzaron los valores **52.62** y **62.51**, respectivamente. En contraste con las obtenidas por U-Net, que fueron **14.82** y **22.41**. Además, si bien el método convolucional superó al generativo por alrededor de cuatro puntos en la Overall Accuracy, esto se debe únicamente a que la clase Large era predominante en el dataset, y no quiere decir que el modelo se desempeñe mejor en un entorno real, donde los árboles no necesariamente son tan grandes como los de esa clase.

5.3. LIMITACIONES

Las principales limitaciones encontradas durante la realización de esta memoria fueron tres. En primer lugar, la disponibilidad limitada de datos de calidad, ya que las anotaciones no eran muy precisas, lo que restringió el aprendizaje de patrones complejos por parte del modelo. En segundo lugar, la restricción de tiempo. Con más tiempo disponible, habría sido posible resegmentar los datos para obtener información más precisa y de mejor calidad, además de probar nuevas configuraciones de hiperparámetros y preprocesamiento de los datos, así como realizar mejoras al modelo base. Por último, la limitada disponibilidad de recursos computacionales. Durante gran parte de esta tesis no se contó con un alto poder de cómputo, lo cual limitó significativamente la cantidad de experimentos que se podían realizar, así como la complejidad de los modelos implementados. Al inicio del proyecto, no se disponía del *hardware* reportado en la Sección 3.3, lo que imposibilitó el uso de técnicas de aprendizaje no supervisado o generativo y restringió el trabajo al uso exclusivo de U-Net. Es más, incluso con el *hardware* reportado, el entrenamiento de BEITv2 en su totalidad toma alrededor de 12 horas, lo que vuelve complejo realizar pruebas. En esta misma línea, otra limitante fueron los recursos monetarios, ya que para poder realizar los experimentos reportados, se tuvo que contratar los servicios de la nube, los cuales son bastante costosos y no se contaba con asignación de recursos de parte de la Universidad.

5.4. CONTRIBUCIONES

En este trabajo se logró construir un modelo de segmentación semántica capaz de identificar con alta precisión árboles de palto de gran tamaño. Aunque aún queda trabajo por realizar para compensar la disparidad entre clases en el conjunto de datos de entrenamiento, el modelo muestra un alto potencial para alcanzar resultados satisfactorios con pequeñas modificaciones que favorezcan el aprendizaje de las clases menos representadas. Asimismo, hasta el momento, no existe una gran cantidad de investigaciones en el área de procesamiento de imágenes de cultivos mediante aprendizaje no supervisado, y son aún menos los trabajos que se enfocan específicamente en la segmentación de árboles de palto. De esta manera, esta investigación establece un *punto de partida* que permitirá a futuros trabajos en el área realizar comparaciones, fomentando así el desarrollo de esta línea de investigación.

5.5. TRABAJO FUTURO

Como se mencionó previamente, este trabajo se desarrolló con datos que presentaban un evidente desbalance de clases, lo cual provocó diversos problemas en las predicciones, principalmente que las clases *Small* y *Medium* no fueran del todo identificadas. Para mitigar este problema, sería posible emplear otras funciones de pérdida, como la *focal loss*, o incorporar términos de regularización que asignen mayor importancia a las clases minoritarias. Asimismo, se podrían aplicar técnicas de *data augmentation* que permitan equilibrar la disparidad en la cantidad de datos entre clases.

En cuanto al modelo en sí mismo, se podría considerar la incorporación de técnicas de *Self-Knowledge Distillation* al Tokenizador, permitiendo así aprovechar el *dark knowledge* de la red. Este término se refiere al conocimiento implícito del modelo reflejado en la predicción de una clase incorrecta, o en la asignación de una probabilidad mayor a 0 a una clase en escenarios de predicción multiclase, lo que indica que el modelo es capaz de reconocer ciertas similitudes entre las clases.

Es importante destacar que en Chile, debido a las regulaciones estatales, la mayoría de los cultivos de paltos también incluyen árboles nativos. Por lo tanto, al implementar nuevas tecnologías en el país, sería esencial contar con datos que permitan a los modelos de segmentación distinguir estas especies y evitar falsos positivos. Además, sería interesante considerar el entrenamiento del modelo con parches obtenidos de un ortomosaico, lo que podría lograrse mediante vuelos que capturen imágenes con un solapamiento del 90 %. Esto facilitaría el aprendizaje del modelo para segmentar con mayor detalle los contornos de los paltos y eventualmente distinguirlos de otros árboles.

REFERENCIAS BIBLIOGRÁFICAS

- [Adão *et al.*, 2017] Adão, T., Hruška, J., Pádua, L., Bessa, J., Peres, E., Morais, R., y Sousa, J. J. (2017). Hyperspectral imaging: A review on uav-based sensors, data processing and applications for agriculture and forestry. *Remote sensing*, 9(11):1110.
- [Akca y Polat, 2022] Akca, S. y Polat, N. (2022). Semantic segmentation and quantification of trees in an orchard using uav orthophoto. *Earth Science Informatics*, 15(4):2265–2274.
- [Amraoui *et al.*, 2022] Amraoui, K. E., Lghoul, M., Ezzaki, A., Masmoudi, L., Hadri, M., El-belrhiti, H., y Simo, A. A. (2022). Avo-airdb: An avocado uav database for agricultural image segmentation and classification. *Data in Brief*, 45:108738.
- [Ardila *et al.*, 2019] Ardila, Diego and Kiraly, Atilla P and Bharadwaj, Sujeeth and Choi, Bok-yung and Reicher, Joshua J and Peng, Lily and Tse, Daniel and Etemadi, Mozziyar and Ye, Wenxing and Corrado, Greg and others (2019). End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. *Nature medicine*, 25(6):954–961.
- [Badrinarayanan *et al.*, 2017] Badrinarayanan, V., Kendall, A., y Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495.
- [Berra y Peppia, 2020] Berra, E. y Peppia, M. (2020). Advances and challenges of uav sfm mvs photogrammetry and remote sensing: Short review. En *2020 IEEE Latin American GRSS & ISPRS Remote Sensing Conference (LAGIRS)*, pp. 533–538. IEEE.
- [Blaschke *et al.*, 2000] Blaschke, T., Lang, S., Lorup, E., Strobl, J., y Zeil, P. (2000). Object-oriented image processing in an integrated gis/remote sensing environment and perspectives for environmental applications. *Environmental information for planning, politics and the public*, 2(1995):555–570.
- [Bouguettaya *et al.*, 2022] Bouguettaya, A., Zarzour, H., Kechida, A., y Taberkit, A. M. (2022). Deep learning techniques to classify agricultural crops through uav imagery: A review. *Neural Computing and Applications*, 34(12):9511–9536.
- [Casagli *et al.*, 2017] Casagli, N., Frodella, W., Morelli, S., Tofani, V., Ciampalini, A., Intrieri, E., Raspini, F., Rossi, G., Tanteri, L., y Lu, P. (2017). Spaceborne, uav and ground-based remote sensing techniques for landslide mapping, monitoring and early warning. *Geoenvironmental Disasters*, 4:1–23.
- [Chehreh *et al.*, 2023] Chehreh, B., Moutinho, A., y Viegas, C. (2023). Latest trends on tree classification and segmentation using uav data—a review of agroforestry applications. *Remote Sensing*, 15(9):2263.

- [Chen et al., 2014] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., y Yuille, A. L. (2014). Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*.
- [Chen et al., 2017a] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., y Yuille, A. L. (2017a). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848.
- [Chen et al., 2017b] Chen, L.-C., Papandreou, G., Schroff, F., y Adam, H. (2017b). Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*.
- [Chen et al., 2018] Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., y Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. En *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818.
- [Colombo y Papetti, 2019] Colombo, R. y Papetti, A. (2019). Avocado (persea americana mill.) by-products and their impact: from bioactive compounds to biomass energy and sorbent material for removing contaminants. a review. *International Journal of Food Science & Technology*, 54(4):943–951.
- [Cong et al., 2022] Cong, P., Zhou, J., Li, S., Lv, K., y Feng, H. (2022). Citrus tree crown segmentation of orchard spraying robot based on rgb-d image and improved mask r-cnn. *Applied Sciences*, 13(1):164.
- [Csillik et al., 2018] Csillik, O., Cherbini, J., Johnson, R., Lyons, A., y Kelly, M. (2018). Identification of citrus trees from unmanned aerial vehicle imagery using convolutional neural networks. *Drones*, 2(4):39.
- [Csurka et al., 2004] Csurka, G., Dance, C., Fan, L., Willamowski, J., y Bray, C. (2004). Visual categorization with bags of keypoints. En *Workshop on statistical learning in computer vision, ECCV*, volumen 1, pp. 1–2. Prague.
- [Dalal y Triggs, 2005] Dalal, N. y Triggs, B. (2005). Histograms of oriented gradients for human detection. En *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volumen 1, pp. 886–893 vol. 1.
- [Dechesne et al., 2017] Dechesne, C., Mallet, C., Le Bris, A., y Gouet-Brunet, V. (2017). Semantic segmentation of forest stands of pure species as a global optimization problem. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-1/W1:141–148.
- [Diez et al., 2021] Diez, Y., Kentsch, S., Fukuda, M., Caceres, M. L. L., Moritake, K., y Cabezas, M. (2021). Deep learning in forestry using uav-acquired rgb data: A practical review. *Remote Sensing*, 13(14):2837.
- [Dosovitskiy, 2020] Dosovitskiy, A. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

- [Fang et al., 2018] Fang, F., Yuan, X., Wang, L., Liu, Y., y Luo, Z. (2018). Urban land-use classification from photographs. *IEEE Geoscience and Remote Sensing Letters*, 15(12):1927–1931.
- [Farabet et al., 2012] Farabet, C., Couprie, C., Najman, L., y LeCun, Y. (2012). Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1915–1929.
- [Gomes et al., 2018] Gomes, M. F., Maillard, P., y Deng, H. (2018). Individual tree crown detection in sub-meter satellite imagery using marked point processes and a geometrical-optical model. *Remote Sensing of Environment*, 211:184–195.
- [Grybas y Congalton, 2022] Grybas, H. y Congalton, R. G. (2022). Evaluating the impacts of flying height and forward overlap on tree height estimates using unmanned aerial systems. *Forests*, 13(9).
- [Gui et al., 2024] Gui, J., Chen, T., Zhang, J., Cao, Q., Sun, Z., Luo, H., y Tao, D. (2024). A survey on self-supervised learning: Algorithms, applications, and future trends. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- [He et al., 2017] He, K., Gkioxari, G., Dollár, P., y Girshick, R. (2017). Mask r-cnn. En *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969.
- [He et al., 2015] He, K., Zhang, X., Ren, S., y Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916.
- [He et al., 2004] He, X., Zemel, R., y Carreira-Perpinan, M. (2004). Multiscale conditional random fields for image labeling. 2:II–II.
- [Hejazipoor et al., 2021] Hejazipoor, H., Massah, J., Soryani, M., Vakilian, K. A., y Chegini, G. (2021). An intelligent spraying robot based on plant bulk volume. *Computers and Electronics in Agriculture*, 180:105859.
- [Huang et al., 2019] Huang, Z., Huang, L., Gong, Y., Huang, C., y Wang, X. (2019). Mask scoring r-cnn. En *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6409–6418.
- [Jadhav y Singh, 2018] Jadhav, J. K. y Singh, R. (2018). Automatic semantic segmentation and classification of remote sensing data for agriculture. *Mathematical Models in Engineering*, 4(2):112–137.
- [Jiménez-Jiménez et al., 2021] Jiménez-Jiménez, S. I., Ojeda-Bustamante, W., Marcial-Pablo, M. d. J., y Enciso, J. (2021). Digital terrain models generated with low-cost uav photogrammetry: Methodology and accuracy. *ISPRS International Journal of Geo-Information*, 10(5).

- [Kass *et al.*, 1988] Kass, M., Witkin, A., y Terzopoulos, D. (1988). Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331.
- [Kitano *et al.*, 2019] Kitano, B. T., Mendes, C. C., Geus, A. R., Oliveira, H. C., y Souza, J. R. (2019). Corn plant counting using deep learning and uav images. *IEEE Geoscience and Remote Sensing Letters*.
- [Krizhevsky *et al.*, 2012] Krizhevsky, A., Sutskever, I., y Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.
- [Kussul *et al.*, 2017] Kussul, N., Lavreniuk, M., Skakun, S., y Shelestov, A. (2017). Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 14(5):778–782.
- [La Rosa *et al.*, 2020] La Rosa, L. E. C., Zortea, M., Gemignani, B., Oliveira, D. A. B., y Feitosa, R. Q. (2020). Fcrn-based multi-task learning for automatic citrus tree detection from uav images. En *2020 IEEE Latin American GRSS & ISPRS Remote Sensing Conference (LAGIRS)*, pp. 403–408. IEEE.
- [Lafferty *et al.*, 2001] Lafferty, J., McCallum, A., y Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.
- [LeCun *et al.*, 2015] LeCun, Y., Bengio, Y., y Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.
- [Lewicki y Liu, 2021] Lewicki, T. y Liu, K. (2021). Multimodal wildfire surveillance with uav. En *2021 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6. IEEE.
- [Lézoray *et al.*, 2017] Lézoray, O., Meurie, C., y Celebi, E. (2017). Superpixels for image processing and computer vision. *Journal of Electronic Imaging*.
- [Li *et al.*, 2024] Li, Z., Deng, X., Lan, Y., Liu, C., y Qing, J. (2024). Fruit tree canopy segmentation from uav orthophoto maps based on a lightweight improved u-net. *Computers and Electronics in Agriculture*, 217:108538.
- [Long *et al.*, 2015] Long, J., Shelhamer, E., y Darrell, T. (2015). Fully convolutional networks for semantic segmentation. En *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440.
- [Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110.
- [Lu *et al.*, 2022] Lu, Z., Qi, L., Zhang, H., Wan, J., y Zhou, J. (2022). Image segmentation of uav fruit tree canopy in a natural illumination environment. *Agriculture*, 12(7):1039.
- [Machefer *et al.*, 2020] Machefer, M., Lemarchand, F., Bonnefond, V., Hitchins, A., y Sidiropoulos, P. (2020). Mask r-cnn refitting strategy for plant counting and sizing in uav imagery. *Remote Sensing*, 12(18):3015.

- [Malambo *et al.*, 2018] Malambo, Lonesome and Popescu, Sorin C and Murray, Seth C and Putman, E and Pugh, N Ace and Horne, David W and Richardson, G and Sheridan, R and Rooney, William L and Avant, R and others (2018). Multitemporal field-based plant height estimation using 3d point clouds generated from small unmanned aerial systems high-resolution imagery. *International Journal of Applied Earth Observation and Geoinformation*, 64:31–42.
- [McCarthy *et al.*, 2010] McCarthy, C. L., Hancock, N. H., y Raine, S. R. (2010). Applied machine vision of plants: a review with implications for field deployment in automated farming operations. *Intelligent Service Robotics*, 3:209–217.
- [Meshram *et al.*, 2022] Meshram, A. T., Vanalkar, A. V., Kalambe, K. B., y Badar, A. M. (2022). Pesticide spraying robot for precision agriculture: A categorical literature review and future trends. *Journal of Field Robotics*, 39(2):153–171.
- [Morales *et al.*, 2018] Morales, G., Kemper, G., Sevillano, G., Arteaga, D., Ortega, I., y Telles, J. (2018). Automatic segmentation of mauritia flexuosa in unmanned aerial vehicle (uav) imagery using deep learning. *Forests*, 9(12):736.
- [Narvaez *et al.*, 2017] Narvaez, F. Y., Reina, G., Torres-Torriti, M., Kantor, G., y Cheein, F. A. (2017). A survey of ranging and imaging techniques for precision agriculture phenotyping. *IEEE/ASME Transactions on Mechatronics*, 22(6):2428–2439.
- [Osco *et al.*, 2021] Osco, L. P., Junior, J. M., Ramos, A. P. M., de Castro Jorge, L. A., Fatholahi, S. N., de Andrade Silva, J., Matsubara, E. T., Pistori, H., Gonçalves, W. N., y Li, J. (2021). A review on deep learning in uav remote sensing. *International Journal of Applied Earth Observation and Geoinformation*, 102:102456.
- [Pandiselvam *et al.*, 2023] Pandiselvam, R., Mathew, A., Imran, S., Pandian, R. T. P., y Manikantan, M. (2023). Design, development and evaluation of a tractor mounted air blast sprayer for coconut and arecanut. *Science Progress*, 106(3):00368504231199927.
- [Pang *et al.*, 2020] Pang, Y., Shi, Y., Gao, S., Jiang, F., Veeranampalayam-Sivakumar, A.-N., Thompson, L., Luck, J., y Liu, C. (2020). Improved crop row detection with deep neural network for early-season maize stand count in uav imagery. *Computers and Electronics in Agriculture*, 178:105766.
- [Peng *et al.*, 2022] Peng, Z., Dong, L., Bao, H., Ye, Q., y Wei, F. (2022). Beit v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv preprint arXiv:2208.06366*.
- [Radford *et al.*, 2021] Radford, Alec and Kim, Jong Wook and Hallacy, Chris and Ramesh, Aditya and Goh, Gabriel and Agarwal, Sandhini and Sastry, Girish and Askell, Amanda and Mishkin, Pamela and Clark, Jack and others (2021). Learning transferable visual models from natural language supervision. En *International conference on machine learning*, pp. 8748–8763. PMLR.
- [Randall, 1971] Randall, J. (1971). The relationships between air volume and pressure on spray distribution in fruit trees. *Journal of Agricultural Engineering Research*, 16(1):1–31.

- [Recio *et al.*, 2013] Recio, J., Hermosilla, T., Ruiz, L., y Palomar, J. (2013). Automated extraction of tree and plot-based parameters in citrus orchards from aerial images. *Computers and electronics in agriculture*, 90:24–34.
- [Reutebuch *et al.*, 2005] Reutebuch, S. E., Andersen, H.-E., y McGaughey, R. J. (2005). Light detection and ranging (lidar): an emerging tool for multiple resource inventory. *Journal of forestry*, 103(6):286–292.
- [Roerdink y Meijster, 2000] Roerdink, J. B. y Meijster, A. (2000). The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta informaticae*, 41(1-2):187–228.
- [Ronneberger *et al.*, 2015] Ronneberger, O., Fischer, P., y Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. En *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III* 18, pp. 234–241. Springer.
- [Rosell y Sanz, 2012] Rosell, J. y Sanz, R. (2012). A review of methods and applications of the geometric characterization of tree crops in agricultural activities. *Computers and electronics in agriculture*, 81:124–141.
- [Rottensteiner *et al.*, 2012] Rottensteiner, F., Sohn, G., Jung, J., Gerke, M., Baillard, C., Benitez, S., y Breitkopf, U. (2012). The isprs benchmark on urban object classification and 3d building reconstruction. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences; I-3*, 1(1):293–298.
- [Salazar-Reque *et al.*, 2023] Salazar-Reque, I., Arteaga, D., Mendoza, F., Rojas, M. E., Soto, J., Huaman, S., y Kemper, G. (2023). Differentiating nutritional and water statuses in hass avocado plantations through a temporal analysis of vegetation indices computed from aerial rgb images. *Computers and Electronics in Agriculture*, 213:108246.
- [Shahi *et al.*, 2023] Shahi, T. B., Xu, C.-Y., Neupane, A., y Guo, W. (2023). Recent advances in crop disease detection using uav and deep learning techniques. *Remote Sensing*, 15(9):2450.
- [Shotton *et al.*, 2008] Shotton, J., Johnson, M., y Cipolla, R. (2008). Semantic texton forests for image categorization and segmentation. pp. 1–8.
- [Shotton *et al.*, 2006] Shotton, J., Winn, J., Rother, C., y Criminisi, A. (2006). Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. pp. 1–15.
- [Tighe y Lazebnik, 2010] Tighe, J. y Lazebnik, S. (2010). Superparsing: Scalable nonparametric image parsing with superpixels. En *Proceedings of the 11th European Conference on Computer Vision: Part V, ECCV'10*, p. 352–365, Berlin, Heidelberg. Springer-Verlag.
- [Toro y Tsourdos, 2018] Toro, F. G. y Tsourdos, A. (2018). *UAV sensors for environmental monitoring*. MDPI.

- [Treisman, 1998] Treisman, A. (1998). The perception of features and objects. *Visual attention*, 8:26–54.
- [Volpi y Ferrari, 2015] Volpi, M. y Ferrari, V. (2015). Semantic segmentation of urban scenes by learning local class interactions. En *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1–9.
- [Wu et al., 2020] Wu, J., Yang, G., Yang, H., Zhu, Y., Li, Z., Lei, L., y Zhao, C. (2020). Extracting apple tree crown information from remote imagery using deep learning. *Computers and electronics in agriculture*, 174:105504.
- [Xiao et al., 2018] Xiao, T., Liu, Y., Zhou, B., Jiang, Y., y Sun, J. (2018). Unified perceptual parsing for scene understanding. En *Proceedings of the European conference on computer vision (ECCV)*, pp. 418–434.
- [Yang et al., 2016] Yang, S., Chen, Q., Yuan, X., y Liu, X. (2016). Adaptive coherency matrix estimation for polarimetric sar imagery based on local heterogeneity coefficients. *IEEE Transactions on Geoscience and Remote Sensing*, 54(11):6732–6745.
- [Ye et al., 2022] Ye, Z., Wei, J., Lin, Y., Guo, Q., Zhang, J., Zhang, H., Deng, H., y Yang, K. (2022). Extraction of olive crown based on uav visible images and the u2-net deep learning model. *Remote Sensing*, 14(6):1523.
- [Yin y Wang, 2019] Yin, D. y Wang, L. (2019). Individual mangrove tree measurement using uav-based lidar data: Possibilities and challenges. *Remote Sensing of Environment*, 223:34–49.
- [Yu y Koltun, 2015] Yu, F. y Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*.
- [Yuan y Sarma, 2011] Yuan, X. y Sarma, V. (2011). Automatic urban water-body detection and segmentation from sparse alsm data via spatially constrained model-driven clustering. *IEEE Geoscience and Remote Sensing Letters*, 8(1):73 – 77.
- [Yuan et al., 2021] Yuan, X., Shi, J., y Gu, L. (2021). A review of deep learning methods for semantic segmentation of remote sensing imagery. *Expert Systems with Applications*, 169:114417.
- [Zhang et al., 2020] Zhang, J., Xie, T., Yang, C., Song, H., Jiang, Z., Zhou, G., Zhang, D., Feng, H., y Xie, J. (2020). Segmenting purple rapeseed leaves in the field from uav rgb imagery using deep learning as an auxiliary means for nitrogen stress detection. *Remote Sensing*, 12(9):1403.
- [Zhang, 2022] Zhang, Q. (2022). A novel resnet101 model based on dense dilated convolution for image classification. *SN Applied Sciences*, 4.