

Reconocimiento automático de imágenes para reconstrucción de redes planta-polinizador



Universitat Oberta
de Catalunya

**Andrés Merino
Toapanta**

Máster Universitario en
Ciencias de Datos

Área: Machine Learning

Tutor de TF:

Albert Solé-Ribalta, Pau
Enric Serra, Javier Borge
Holthoefer

**Profesor responsable
de la asignatura:**

Albert Solé-Ribalta
Enero 2024



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada 3.0 España de Creative Commons

Ficha del Trabajo Final

Título del trabajo:	Reconocimiento automático de imágenes para reconstrucción de redes planta-polinizador
Nombre del autor:	Andrés Merino Toapanta
Nombre del Tutor de TF:	Albert Solé-Ribalta, Pau Enric Serra, Javier Borge Holthoefer
Fecha de entrega:	Enero 2024
Titulación o programa:	Máster Universitario en Ciencias de Datos
Área del Trabajo Final:	Machine Learning
Idioma del trabajo:	Castellano
Palabras clave:	Reconocimiento de imágenes, red planta-polinizador, red neuronal convolucional

*La Ciencia de Datos
es un invento de los matemáticos
para vender más Matemática.*

Agradecimientos

A mi familia, por sus enseñanzas y los valores que me supieron inculcar. Por las alegrías y problemas que me han llevado a donde ahora estoy. A mis padres, mis hermanas y mi cuñado por el amor y el cariño que han compartido conmigo.

A mis tutores: Albert Solé-Ribalta, Pau Enric Serra, Javier Borge Holthoefer, por proponer el problema de este trabajo, por su apoyo y guía durante el desarrollo de este trabajo.

A Janeth y Mario, quienes me han acompañado en este camino, me motivaron a iniciarla y me han brindado su apoyo y cariño.

A mis alumnos, en especial a Melani, Sofía, Martín y Pancho, quienes me ayudan a seguir descubriendo los hermoso del camino de aprender y enseñar.

A todos mis amigos (Priss, Eve, Cristian, David, Pau, Mishu,...), que cerca o lejos, siempre están ahí para compartir un momento, una charla, un café, una cerveza, una sonrisa, un abrazo, un consejo, una vida.

Abstract

The present work focuses on the construction and training of a convolutional neural network (CNN) with the main objective of detecting pollinators in flower images, which will serve as a basis for the reconstruction of plant-pollinator interaction networks. Specific objectives include the collection and labeling of around 5,000 images of pollinators in two study habitats in Cabrera, Balearic Islands; evaluation of two convolutional neural network architectures (YOLOv5 and EfficientNET) for accurate pollinator detection (in functional groups), training of the selected network, reconstruction of a pollination network from the detections, and performance evaluation using standard network metrics. With the results of this work, we seek to automate the collection and classification of interactions, directly delivering the plant-pollinator interaction network, which offers valuable information both for new research and for decision-making regarding agriculture and conservation.

Keywords: Image recognition, plant-pollinator network, convolutional neural network.

Resumen

El presente trabajo se centra en la construcción y entrenamiento de una red neuronal convolucional (CNN) con el objetivo principal de detectar polinizadores en imágenes de flores, lo que servirá como base para la reconstrucción de redes de interacción planta-polinizador. Los objetivos específicos incluyen la recopilación y etiquetado de alrededor de 5000 imágenes de polinizadores en dos hábitats de estudio en Cabrera, Islas Baleares; la evaluación de dos arquitecturas de redes neuronales convolucionales (YOLOv5 y EfficientNET) para la detección precisa de polinizadores (en grupos funcionales), el entrenamiento de la red seleccionada, la reconstrucción de una red de polinización a partir de las detecciones y la evaluación del desempeño utilizando métricas estándar sobre la red. Con los resultados de este trabajo, se busca poder automatizar la recolección y clasificación interacciones, entregando directamente la red de interacción planta-polinizador, que ofrezca información valiosa tanto para nuevas investigaciones como para la toma de decisiones en lo referente a agricultura y conservación.

Palabras clave: Reconocimiento de imágenes, red planta-polinizador, red neuronal convolucional.

Índice general

Abstract	VI
Resumen	VII
Índice	VIII
Listado de Figuras	X
Listado de Tablas	XII
1 Introducción	1
1.1 Contexto y justificación del trabajo	1
1.2 Motivación	2
1.3 Objetivos	3
1.3.1 Objetivo principal	3
1.3.2 Objetivos específicos	3
1.4 Metodología	3
1.5 Planificación	5
1.6 Impacto en sostenibilidad, ético-social y de diversidad	7
2 Estado del arte	8
2.1 Reconocimiento de imágenes	8
2.2 Origen de las redes neuronales convolucionales	8
2.2.1 Arquitectura y conjuntos de entrenamiento	9
2.2.2 YOLOv5	10
2.2.3 EfficientNet	11
2.2.4 Otras arquitecturas	13
2.2.5 Posibles dificultades	14
2.3 Preprocesamiento de imágenes	14
2.4 Redes de polinización	15
3 Implementación del trabajo	17

3.1 Entorno de trabajo	17
3.2 Procesamiento de datos	18
3.2.1 Aumento de datos	22
3.3 Aplicación de modelos	23
3.3.1 YOLOv5	23
3.3.2 EfficientNet	32
3.4 Reconstrucción de red de polinizadores	33
3.4.1 Red planta-polinizador real	35
3.4.2 Red planta-polinizador estimada	37
3.5 Comparación de redes	38
4 Conclusiones y trabajo futuro	41
A Anexos	43
A.1 Repositorio GitHub	43
A.2 Repositorio de pesos de modelos	44
Bibliografía	44

Listado de Figuras

1.1	Ubicación de Cabrera.	4
1.2	Diagrama Gantt de la planificación del proyecto.	6
2.1	Arquitectura de la red YOLOv5.	11
2.2	Ejemplos de imágenes de insectos utilizadas en [6].	12
2.3	Arquitectura de la red EfficientNet.	13
2.4	Ejemplos de imágenes de insectos utilizadas en [20].	13
3.1	Distribución del alto de las imágenes, en escala logarítmica.	19
3.2	Ejemplo de cada una de las especies de polinizadores.	20
3.3	Distribución de las especies en las imágenes, en escala logarítmica.	21
3.4	Submuestreo de imágenes de la categoría «Bee» y «Lepidoptera».	21
3.5	Ejemplo de reflexiones realizadas.	22
3.6	Distribución de las especies en las imágenes luego del aumento de datos.	22
3.7	Curvas de precisión-confianza modelo YOLOv5 en las imágenes de validación.	25
3.8	Curvas de exhaustividad-confianza modelo YOLOv5 en las imágenes de validación.	25
3.9	Matriz de confusión del modelo YOLOv5.	26
3.10	Comparación entre detecciones realizadas por humanos y por el modelo YOLOv5.	27
3.11	Matriz de confusión del modelo YOLOv5 en el conjunto de prueba.	28
3.12	Matriz de confusión del modelo YOLOv5 en el conjunto de prueba, con un área mínima del 2.5 % del área total de la imagen.	30
3.13	Curvas de aprendizaje del modelo EfficientNet.	33
3.14	Distribución de los insectos en las imágenes.	34
3.15	Distribución de las familias de plantas en las imágenes, en escala logarítmica.	34
3.16	Matriz de interacciones de la red real.	35
3.17	Matriz de probabilidades de la red real.	36

3.18 Red planta-polinizador real.	36
3.19 Matriz de interacciones de la red estimada.	37
3.20 Matriz de probabilidades de la red estimada.	37
3.21 Red estimada.	38
3.22 Comparación de las redes real y estimada.	38

Listado de Tablas

2.1 Arquitecturas utilizadas en la detección de insectos en imágenes.	14
3.1 Distribución de los datos en los conjuntos de entrenamiento, validación y prueba.	23
3.2 Objetos detectados por el modelo YOLOv5.	23
3.3 Métricas del modelo YOLOv5.	28
3.4 Precisión general modelo YOLOv5 en función del área de detección. .	29
3.5 Métricas del modelo YOLOv5, con un área mínima del 2.5 % del área total de la imagen.	30
3.6 Métricas del modelo YOLOv5, con validación cruzada.	31
3.7 Objetos detectados por el modelo EfficientNet.	32
3.8 Métricas de las redes.	39
3.9 Distancia entre las redes.	40

Capítulo 1

Introducción

1.1. Contexto y justificación del trabajo

La polinización es un proceso crucial para la reproducción de muchas especies vegetales, incluyendo cultivos agrícolas de importancia económica, de hecho, tres cuartos (75%) de los 111 cultivos agrícolas más importantes en todo el mundo dependen, en mayor o menor medida, de la polinización realizada por animales [1].

Los polinizadores, como las abejas, mariposas, pájaros y murciélagos, desempeñan un papel esencial en la transferencia de polen de una flor a otra, permitiendo la fertilización y la producción de frutas y semillas. De todos estos animales, los insectos son los polinizadores más importantes, ya que son responsables de la polinización de más del 67% de las especies de plantas con flores en todo el mundo [2].

Sin embargo, en las últimas décadas, ha habido una disminución preocupante en la población de polinizadores debido a factores como la pérdida de hábitat, el uso de pesticidas y el cambio climático [3]. Esto plantea una amenaza significativa para la seguridad alimentaria y la biodiversidad global.

Para abordar esta problemática, es crucial comprender y monitorear la actividad de los polinizadores en diferentes entornos. La observación directa y el seguimiento manual de los polinizadores pueden ser costosos y laboriosos. Aquí es donde entra en juego la tecnología de visión por computadora y las redes neuronales convolucionales.

La detección de polinizadores a partir de imágenes utilizando redes neuronales convolucionales ofrece numerosos beneficios como por ejemplo:

- **Eficiencia y Escalabilidad:** Las redes neuronales convolucionales permiten procesar grandes conjuntos de datos de imágenes de manera eficiente

y automatizada. Esto significa que es posible recolectar y analizar todas las interacciones en una cantidad significativa de datos en un corto período de tiempo, lo que facilita la monitorización a largo plazo.

- **Precisión:** Las CNN son conocidas por su capacidad para detectar patrones y características en imágenes con alta precisión. Esto garantiza que las detecciones de polinizadores sean confiables y consistentes.
- **Aplicabilidad:** Este enfoque puede ser aplicado en una variedad de entornos, desde campos agrícolas hasta áreas naturales, lo que lo hace relevante tanto para la investigación científica como para la toma de decisiones en agricultura y conservación.

El objetivo de este trabajo es utilizar redes neuronales convolucionales para construir un modelo que habilite la detección de polinizadores y los clasifique en grupos funcionales, con lo que se pueda reconstruir una red de polinización. De esta manera, se podrá monitorear la actividad de los polinizadores en diferentes entornos y se podrá identificar tendencias en las poblaciones de polinizadores. Esto brindará información valiosa para las instituciones de investigación como para los diferentes departamentos gubernamentales y no gubernamentales en la toma de decisiones para la agricultura y la conservación.

1.2. Motivación

Mi trayectoria como docente e investigador en la Facultad de Ciencias Exactas y Naturales de la Pontificia Universidad Católica del Ecuador se ha enriquecido significativamente gracias a la colaboración interdisciplinaria con colegas de la Escuela de Biología. Estas sinergias han abierto oportunidades únicas para integrar mi especialización en matemáticas y ciencia de datos con el estudio y la preservación de la biodiversidad ecuatoriana, un campo de vital importancia en un país que se distingue por ser el sexto más megadiverso del mundo.

La interacción con estas diversas disciplinas me ha permitido acompañar a estudiantes tanto de Biología como de Ciencia de Datos en salidas de campo, donde la teoría se encuentra con la práctica. Estas experiencias en el terreno han sido fundamentales para comprender la complejidad y la riqueza de las especies de nuestro país, muchas de las cuales aún están por descubrir. Esta inmersión directa en el estudio de la biodiversidad ha fortalecido mi convicción de que la conservación efectiva requiere de enfoques innovadores y técnicas avanzadas, como el reconocimiento automático de imágenes, para enfrentar los desafíos

emergentes como la deforestación, la caza furtiva, la contaminación y el cambio climático.

A través de este trabajo, se busca contribuir con un enfoque que permita la reconstrucción de redes de polinización de manera más eficiente y no invasiva. La meta es proporcionar herramientas que mejoren nuestra capacidad de monitoreo y protección de las especies, particularmente en extensas áreas donde la aparición de nuevas especies requiere una rápida inclusión en los planes de conservación. Este proyecto representa la confluencia de mi experiencia en el aula y en el campo, mi colaboración con expertos en Biología y mi formación en Ciencia de Datos, todo dirigido hacia un objetivo común: proteger y entender mejor el tesoro natural que Ecuador posee.

1.3. Objetivos

1.3.1. Objetivo principal

Construir y entrenar una red neuronal convolucional para la detección de polinizadores en imágenes de flores, que sirva de manera universal para la reconstrucción de redes de polinización.

1.3.2. Objetivos específicos

- Depurar un conjunto de datos de imágenes de polinizadores.
- Seleccionar un modelo de red neuronal convolucional para la detección de polinizadores en imágenes.
- Entrenar la red neuronal convolucional sobre el conjunto de datos de imágenes de polinizadores construida.
- Reconstruir una red de polinización a partir de las detecciones de la red neuronal convolucional.
- Evaluar los resultados analizando las métricas de la red de polinizadores reconstruida.

1.4. Metodología

Como es usual en los proyectos de Ciencia de Datos, el proyecto se divide en las siguientes etapas: recopilación y preparación de datos, creación de mo-

dicho a aplicarse sobre los datos y evaluación contextualizada de los resultados obtenidos.

En lo que respecta a la recopilación y preparación de datos, se utilizará un conjunto de alrededor de cinco mil imágenes (5000) registradas en dos hábitats y cuatro lugares de estudio en Cabrera (Islas Baleares, ver Figura 1.1). Los polinizadores están identificados por expertos y etiquetados utilizando las siguientes categorías: «Bee», «Diptera», «Coleoptera», «Hoverfly», «Lepidoptera», «Wasp» y «Others».



Figura 1.1: Ubicación de Cabrera. Tomado de Bing [4]

Para realizar la identificación de las plantas que se hallan en el fondo de las imágenes, se utilizará la herramienta *My PlantNet API* [5]. Se planteará la identificación a nivel de Familia, para, de esta manera, obtener los grupos de plantas que cada insecto poliniza.

Esta será la etapa de partida para la construcción del conjunto de datos de imágenes de polinizadores que será utilizado para el entrenamiento y evaluación de las etapas posteriores. Finalmente, de darse el caso, se utilizarán técnicas tradicionales de aumento de imágenes para incrementar el tamaño del conjunto de datos; entre estas, se encuentran la rotación, el recorte, el volteo, el cambio de color, etc.

Por el lado del diseño de una red neuronal convolucional para la detección de polinizadores en imágenes, se analizarán dos alternativas: la arquitectura de red neuronal convolucional YOLOv5 que ha demostrado dar buenos resultados en la detección de objetos en imágenes, en especial, de insectos [6, 7]; y la arquitectura de red neuronal convolucional U-Net que es conocida por su eficiencia y precisión en la segmentación de imágenes médicas [8].

tura de red neuronal convolucional *EfficientNET* que ha obtenido resultados de precisión altos cuando se trata de pocas clases [8].

En lo relacionado con la evaluación contextualizada de los resultados obtenidos, en primera instancia, se utilizarán las métricas de precisión, sensibilidad y especificidad para evaluar el desempeño de la red neuronal convolucional. Además, a partir de los resultados, se reconstruirá una red de polinización bajo la metodología descrita en [9] y se analizarán las métricas de la misma.

Finalmente, en lo que respecta a la documentación del proyecto, se utilizará el lenguaje de programación Python y el entorno de desarrollo Jupyter Notebook. En específico, se utilizará la librería *Tensorflow* y *Pytorch* para la construcción de la red neuronal convolucional y la librería *NetworkX* para el análisis y visualización de la red de polinización. El código fuente del proyecto se encontrará disponible en el repositorio de GitHub (ver Anexo A.1). Adicionalmente, los pesos de los modelos entrenados se encontrarán disponibles en el repositorio particular de OneDrive del autor (ver Anexo A.2).

1.5. Planificación

En la Figura 1.2 se muestra el diagrama Gantt de la propuesta de planificación de las diferentes actividades del proyecto.

Las actividades del proyecto se agrupan en cuatro etapas: Definición y planificación del trabajo, Elaboración del estado del arte, Diseño e implementación del trabajo, Redacción de la memoria. Adicionalmente, se incluye una etapa de Defensa del trabajo.

Para la primera etapa, se estima una duración de 2 semanas. En esta etapa se realizará la definición de la propuesta, los objetivos y la metodología. Para la segunda etapa, se estima una duración de 3 semanas. En esta etapa se realizará la recolección de información y la redacción del estado del arte. Para la tercera etapa, se estima una duración de 8 semanas. En esta etapa se realizará la construcción del conjunto de datos, el diseño de la red neuronal, el entrenamiento de la red neuronal y la reconstrucción de la red de polinización; se finaliza con la evaluación de los resultados. Para la cuarta etapa, se estima una duración de 4 semanas. En esta etapa se realizará la redacción de la memoria. Finalmente, para la quinta etapa, se estima una duración de 1 semana. En esta etapa se realizará la defensa del trabajo.

De manera específica, en la Figura 1.2 se aprecian la distribución de las actividades en el tiempo. Los rombos representan las fechas de entrega de avances del proyecto.

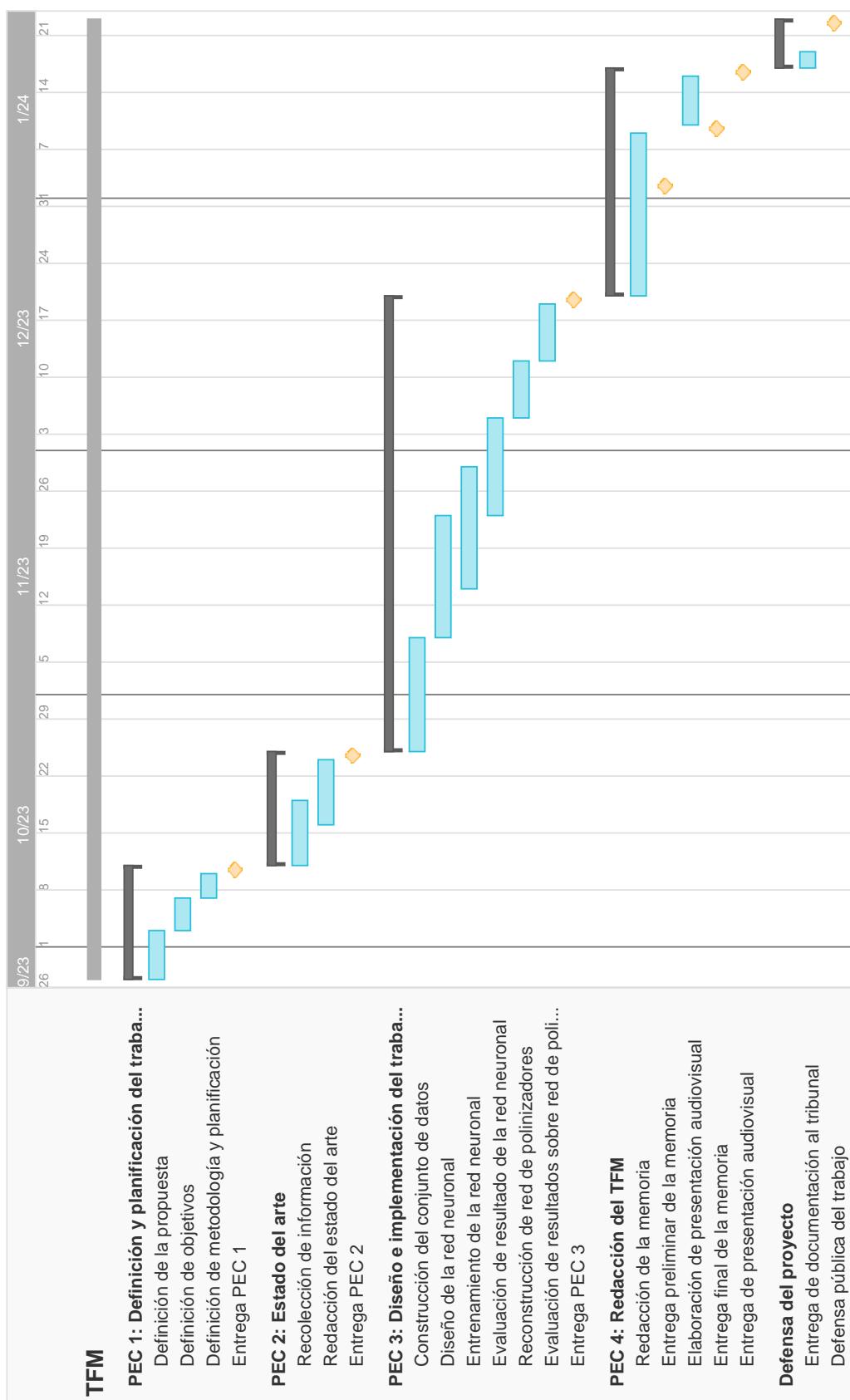


Figura 1.2: Diagrama Gantt de la planificación del proyecto.

1.6. Impacto en sostenibilidad, ético-social y de diversidad

Sostenibilidad: Este proyecto contribuye al ODS 15 (Vida de Ecosistemas Terrestres), ya que promueve la sostenibilidad de los ecosistemas terrestres. Al facilitar la monitorización y el estudio de los polinizadores, contribuye a la conservación de la biodiversidad y al mantenimiento de ecosistemas saludables, esenciales para la seguridad alimentaria y el equilibrio natural. Además, al emplear tecnologías de visión por computadora, reduce la necesidad de métodos de monitoreo invasivos y laboriosos, alineándose con prácticas sostenibles y de bajo impacto ambiental.

Comportamiento Ético y Responsabilidad Social: Al mejorar la gestión de ecosistemas de polinizadores, este trabajo apoya indirectamente el ODS 2 (Hambre cero). Una identificación precisa de los polinizadores puede impulsar estrategias para optimizar prácticas agrícolas, aumentando la eficiencia de la polinización y mejorando los rendimientos de cultivos. Esto no solo fortalece la seguridad alimentaria, sino que también apoya prácticas agrícolas más éticas y sostenibles, contribuyendo al bienestar de las comunidades dependientes de la agricultura.

Diversidad, Género y Derechos Humanos: Este proyecto contribuye al ODS 10 (Reducción de las desigualdades). La implementación de tecnologías de detección automática de polinizadores permite a países con recursos limitados acceder a información vital para la conservación de sus ecosistemas, equiparando su capacidad de cuidado y estudio de la biodiversidad con la de países de mayores recursos. Esto promueve una distribución más equitativa del conocimiento y las herramientas necesarias para la protección ambiental global.

Capítulo 2

Estado del arte

En este capítulo se presentan los conceptos básicos de las redes neuronales convolucionales, así como los trabajos más relevantes en el campo de la detección de polinizadores a partir de imágenes utilizando este tipo de redes. Luego de esto, se expone lo referente al preprocesamiento de las imágenes. Posteriormente, se resume una metodología para realizar la reconstrucción de redes de polinización a partir de datos observacionales. Estas son las técnicas fundamentales que se van a usar en lo largo del proyecto.

2.1. Reconocimiento de imágenes

Se entiende por «reconocimiento de imágenes» a la tarea de identificar y localizar objetos en una imagen. Esta tarea se puede dividir en dos sub-tareas: clasificación y detección. La clasificación consiste en identificar el objeto en la imagen, mientras que la detección consiste en identificar su ubicación en la imagen (*bounding box*). Para el caso de la clasificación, se tiene como entrada una imagen y se obtiene como salida una etiqueta que identifica el objeto en la imagen. Estas etiquetas pertenecen a un conjunto finito de categorías preestablecidas.

2.2. Origen de las redes neuronales convolucionales

Las redes neuronales convolucionales (CNN por sus siglas en inglés *Convolutional Neural Network*) son un tipo de red neuronal artificial que se utiliza principalmente para el procesamiento de imágenes. Las CNN se inspiran en la corteza visual de los animales, específicamente en el sistema visual primario de los gatos, descrito por Hubel y Wiesel en 1962 [10]. En este sistema, las neuronas

individuales responden a estímulos en regiones restringidas del campo visual conocidas como campos receptivos. Estas neuronas se organizan en capas, donde cada capa se compone de un conjunto de neuronas que responden a un campo receptivo específico. Las neuronas de la primera capa responden a estímulos simples, como líneas y bordes, mientras que las neuronas de capas posteriores responden a estímulos más complejos, como formas y objetos. Las neuronas de capas posteriores se activan cuando se activan las neuronas de capas anteriores, lo que permite la detección de características cada vez más complejas. Este proceso se conoce como extracción de características.

El inicio de las CNN se remonta a 1982, cuando Fukushima propuso la arquitectura Neocognitron [11], en la cual se presentaban células simples y complejas que sería el análogo a lo que actualmente se conoce como las capas de una CNN. No fue hasta 1989 cuando LeCun et al. [12] propusieron la arquitectura LeNet-5, que se considera la primera CNN moderna. Desde entonces, las CNN han sido ampliamente utilizadas en el campo del procesamiento de imágenes y en el reconocimiento de objetos.

El inicio de las CNN se remonta a 1982, cuando Fukushima propuso la arquitectura Neocognitron [11], sentando las bases para lo que más tarde serían las capas de una CNN. La evolución continuó en 1989 con LeCun et al. [12] y la introducción de la arquitectura LeNet-5, considerada la primera CNN moderna. Sin embargo, el verdadero salto hacia el Aprendizaje Profundo (*Deep Learning*) se produjo en la última década, impulsado por varios factores clave.

El desarrollo de hardware más potente, especialmente GPUs, permitió el entrenamiento de redes neuronales más profundas y complejas [13]. Paralelamente, avances en algoritmos y técnicas de entrenamiento, como la función de activación ReLU y la regularización mediante Dropout, optimizaron el proceso de aprendizaje para redes más profundas [14]. La aparición de arquitecturas revolucionarias, como AlexNet en 2012, demostró mejoras significativas en tareas de procesamiento de imágenes y reconocimiento de objetos [15].

2.2.1. Arquitectura y conjuntos de entrenamiento

En la actualidad, existen varias arquitecturas que han dado buenos resultados en el campo del reconocimiento de objetos en imágenes. Algunas de estas son: AlexNet [15], VGGNet [16], GoogLeNet [17], ResNet [18], entre otras. La mayoría de estas arquitecturas son entrenadas y evaluadas en el conjunto de datos ImageNet [19], que contiene más de 14 millones de imágenes. Este conjunto de datos brinda una amplia variedad de objetos en las imágenes y, por lo tanto, se tienen resultados generales en lo que respecta al reconocimiento. Sin embargo,

para lograr el reconocimiento de insectos, es necesario contar con conjuntos de datos más específicos y también arquitecturas enfocadas en este tipo de imágenes.

En este trabajo, se han seleccionado las arquitecturas YOLOv5 y EfficientNet por su equilibrio único entre velocidad, precisión y eficiencia, esenciales para el análisis de grandes conjuntos de datos de imágenes de insectos. YOLOv5 sobresale en la detección rápida y precisa, vital para el análisis en tiempo real, y ha mostrado resultados prometedores en conjuntos de datos similares [6]. Por su parte, EfficientNet destaca por su escalabilidad y alto rendimiento en la clasificación de imágenes, junto con un uso eficiente de recursos computacionales [20], lo cual es crucial para la identificación precisa de especies en redes planta-polinizador. Esta combinación de características hace que ambas arquitecturas sean particularmente adecuadas para abordar los desafíos específicos del proyecto.

2.2.2. YOLOv5

YOLOv5 es una arquitectura de red neuronal convolucional para el reconocimiento de objetos en imágenes. Esta arquitectura se basa en la arquitectura YOLO (*You Only Look Once*) [21], la cual se caracteriza por ser un modelo de detección de objetos en tiempo real. El modelo toma como entradas las imágenes y produce como salida la detección de objetos, de manera específica: *bounding box*, clase del objeto identificado, nivel de confianza.

De manera específica, la arquitectura de la red se detalla en [22] y se presenta el diseño de la misma en la Figura 2.1. En esta figura, se puede observar que la arquitectura se compone de una serie de bloques convolucionales denominada *backbone*, la cual se encarga de realizar la detección de objetos en la imagen; aquí, Conv representa una capa convolucional, C3 son tres capas convolucionales y un módulo en cascada con varios cuellos de botella y SPP es una capa de agrupación que se utiliza para que el modelo pueda identificar objetos de diferentes tamaños. Posteriormente, se tiene un grupo de capas denominado *Neck*, la cual se encarga de realizar la fusión de características de las capas anteriores; en este grupo se tienen también capas Upsample (para aumentar el muestreo) y Concat (utilizada para cortar la capa anterior). Finalmente, se tiene un grupo de capas denominado *Head*, el cual se encarga de realizar la detección de objetos en la imagen; en este grupo se tienen capas Conv2d (módulos de detección).

En [22] se indica que la arquitectura YOLOv5, al entrenarse sobre el conjunto de datos denominado DOTA (conjunto de imágenes aéreas), alcanza una precisión de 0.707 y un F1 Score de 0.655 en la tarea de reconocer y ubicar objetos en

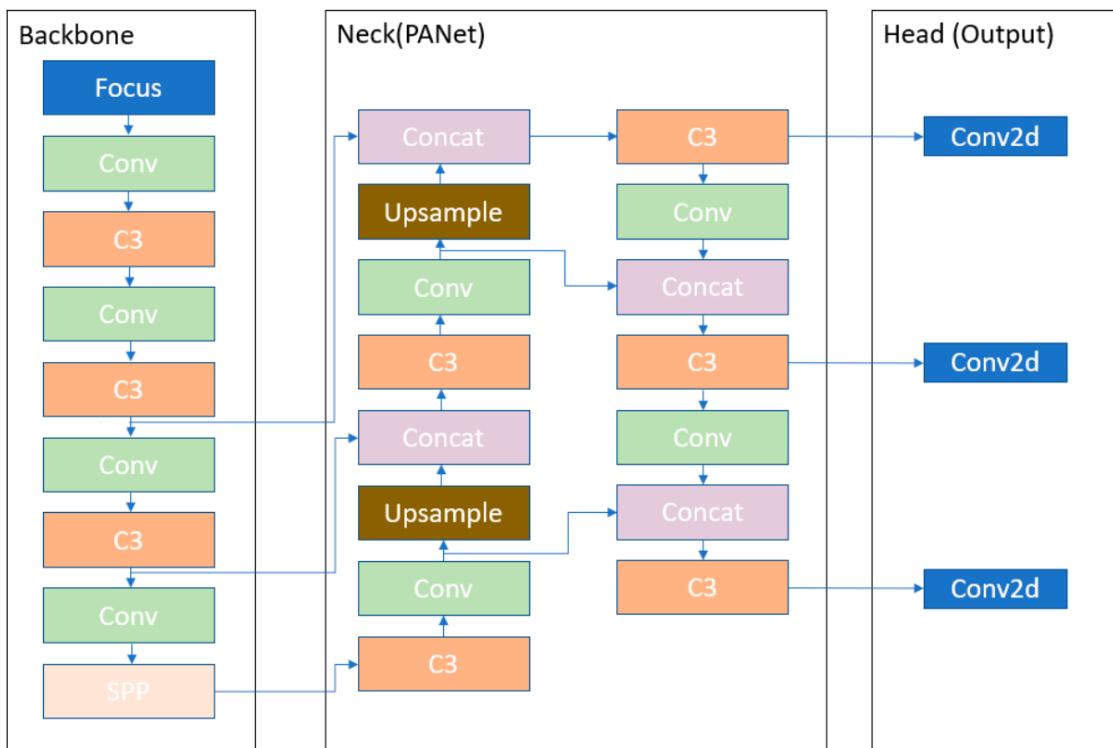


Figura 2.1: Arquitectura de la red YOLOv5. Fuente: [22].

imágenes. En [6], se trabajó con esta arquitectura y se entrenó con un conjunto de datos de 7046 imágenes de insectos tomadas de internet (algunos ejemplos pueden verse en la Figura 2.2). De igual manera, ser enfocó en la tarea de reconocer insectos en una imagen, en este caso, se obtuvo una precisión de entre el 0.7453 y el 0.9450 (con variaciones menores en la arquitectura); por el lado del F1 Score, se obtuvo entre 0.77 y 0.96. Por otro lado, en [7], se utilizó un conjunto de 25028 imágenes obtenidas por los autores del estudio; se obtuvo una precisión de 0.927 y un F1 Score de 0.932.

2.2.3. EfficientNet

EfficientNet [23] es una arquitectura de red neuronal convolucional que se caracteriza por ser eficiente en términos de cómputo y parámetros. El modelo toma como entradas las imágenes y produce como salida la clasificación de objetos en la imagen, de manera específica: clase del objeto identificado y nivel de confianza. Cabe indicar que, a diferencia de YOLOv5, este modelo no realiza la localización del objeto.

En la Figura 2.3 se presenta la arquitectura de la red. Se aprecia que esta red tiene una estructura simple, pues es una capa convolucional (Conv), seguida de

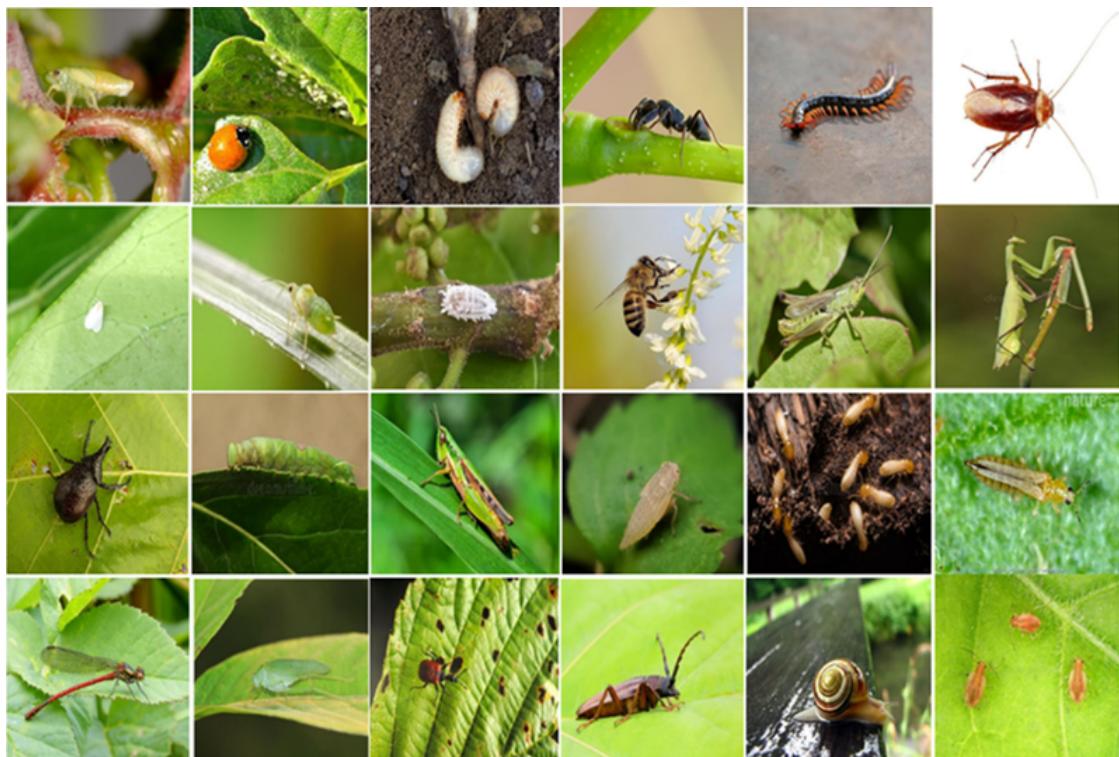


Figura 2.2: Ejemplos de imágenes de insectos utilizadas en [6]. Fuente: [6].

múltiples capas MBconv (capa convolucional móvil con cuello de botella invertido [24]). A cada una de estas capas se aplicó una optimización «apretar y excitar» que tiene el objetivo de mejorar la capacidad de atención de la red, lo cual implica un incremento en la capacidad de discriminación de la información de la imagen.

Inicialmente, esta red se entrenó usando el conjunto de datos ImageNet, donde obtuvo una precisión de 0.933 (el artículo no especifica el F1 score alcanzado) [23] en la tarea de reconocimiento de objetos en imágenes. Adicionalmente, el artículo destaca que esta precisión es alcanzada entre 5.7 y 6.1 veces más rápido que otras arquitecturas, como ResNet.

En [20], se utilizó esta red con un conjunto de datos de 1622 imágenes de insectos. Estas imágenes fueron tomadas de [26], donde se tenían 24 especies diferentes de insectos (algunos ejemplos pueden verse en la Figura 2.4). Se obtuvo una precisión de entre 0.9095 y 0.9751 (dependiendo de la cantidad de clases utilizadas).

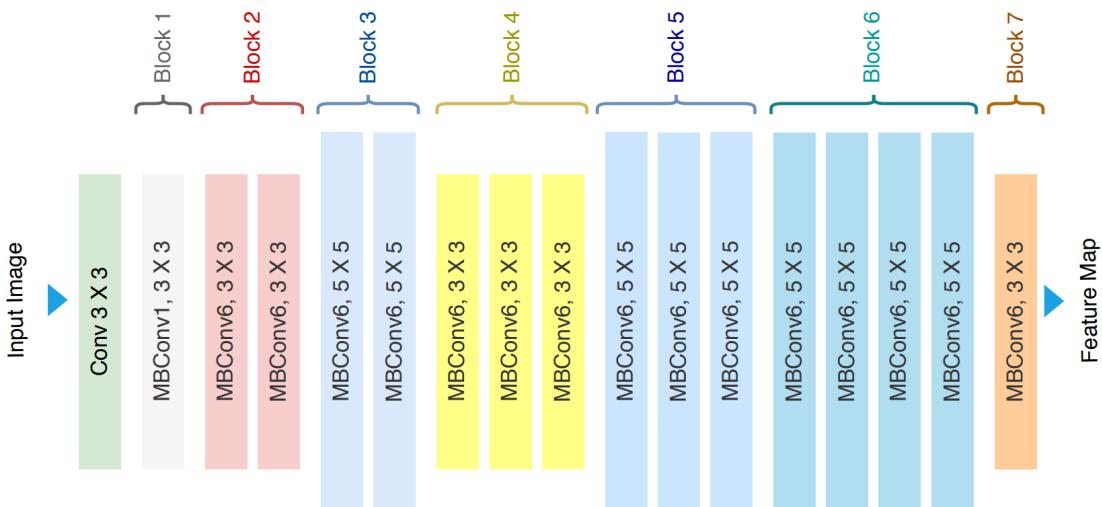


Figura 2.3: Arquitectura de la red EfficientNet. Fuente: [25].



Figura 2.4: Ejemplos de imágenes de insectos utilizadas en [20]. Fuente: [20].

2.2.4. Otras arquitecturas

Cabe indicar que, YOLOv5 y EfficientNet no son las únicas arquitecturas que han sido sometidas a conjuntos específicos de insectos. En [27] se cuenta con un análisis detallado de modelos usados en la detección de insectos en imágenes. En la Tabla 2.1 se presenta un resumen de los resultados obtenidos con los modelos más relevantes.

Modelo	Imágenes	Categorías	Precisión
Small-sample learning [28]	11000	220	0.5765
Tiny YOLOv3 [29]	400	5	0.9800
Fine-tuned ResNet-50 [30]	3549	10	0.9501
Ensemble CNNs [31]	1774	3	0.9904
EfficientNet-B6 [32]	170000	1300	0.8650
GoogLeNet [33]	5629	10	0.9461
ResNet-101 [34]	550	10	0.9867
AlexNet [35]	5000	12	0.9510

Tabla 2.1: Resumen de arquitecturas utilizadas en la detección de insectos en imágenes. Fuente [27].

2.2.5. Posibles dificultades

Li, et al. [27] indican que, a pesar de que los modelos de detección de objetos en imágenes presenta buenos resultados, en el caso de realizar reconocimiento de insectos sobre plantas, se pueden presentar algunas dificultades. Las más relevantes para este trabajo son: la complejidad del fondo de la imagen, los diversos tamaños de los insectos y el desbalance de clases. Con respecto a los dos primeros problemas, se tiene que arquitecturas de la familia YOLO presentan buenos resultados en este aspecto [36].

Para solventar el desbalance de clases, se aplicarán técnicas de aumento de datos (*data augmentation*), las mismas que serán descritas en la siguiente sección.

2.3. Preprocesamiento de imágenes

Dado un conjunto de imágenes, el primer paso es realizar un preprocesamiento de las mismas antes de ser utilizadas en un modelo. Este preprocesamiento va desde el etiquetado hasta el aumento de datos.

A pesar de los múltiples conjuntos de datos sobre insectos (ver [27], Tabla 3), en este trabajo se usará un conjunto de datos propio. Por lo tanto, se debe realizar el etiquetado de las imágenes; para esto, existen herramientas como *LabelImg* [37], orientadas tanto al etiquetado de objetos como a su localización.

Posteriormente, se debe realizar una redimensión de las imágenes, de tal manera que todas tengan el mismo tamaño. Esto es necesario para que el modelo pueda procesar las imágenes de manera eficiente. A continuación, para solventar problemas de limitación de variedad de clases y localizaciones, se puede realizar un aumento de datos, el cual consiste en generar nuevas imágenes a partir de las imágenes originales. Para esto, se pueden utilizar técnicas como el

volteo horizontal y vertical, el recorte de imágenes, el cambio de brillo, entre otras. Adicionalmente, al tratarse de imágenes de insectos, es usual que no se tenga un enfoque apropiado en el objetivo, para solventar esto, a partir de imágenes con enfoque adecuado, se puede agregar ruido Gaussiano para simular el desenfoque [31].

2.4. Redes de polinización

Las redes de polinización son redes de interacción entre especies de plantas y polinizadores. Contar con este tipo de redes de un determinado ecosistema es esencial para entender la dinámica de las interacciones entre las especies, lo cual es importante para la estabilidad de los ecosistemas y, por ende, para la conservación de la biodiversidad.

Estas redes tradicionalmente se construyen mediante observación y trabajo de campo que normalmente son incompletos y, por esto, hace falta aplicar técnicas estadísticas. Para esto, se puede utilizar la metodología propuesta por Young et al. [9], la cual se resume a continuación.

Se consideran n_p especies de plantas y n_a especies de polinizadores, se tiene la matriz $M = (M_{ij})$, de orden $n_p \times n_a$, donde M_{ij} es el número de veces que se ha detectado una interacción de la especie j con la planta i . Se tiene que M podría representar un grafo bipartito ponderado (matriz de incidencia). El objetivo es, a partir de este grafo, generar otro donde se tenga una arista de un polinizador a una planta si el polinizador prefiere a dicha planta; a la representación matricial de este grafo se la denomina B . De esta manera, se tiene que B_{ij} es igual a 1 si la especie j es un polinizador de la planta i y 0 en caso contrario.

Con esto, Young et al. [9] proponen que B_{ij} representa una variable aleatoria de la cual, al tener la matriz M , se puede calcular su probabilidad condicional. Para esto, deducen que

$$P(B_{ij} = 1|M) = \sum_B \int B_{ij} P(B, \theta|M) d\theta$$

donde la suma corre sobre todas las posibles matrices de incidencia y θ representa todos los parámetros del modelo planteado: cambio al número promedio de visitas (r), el efecto del muestreo (C) y abundancia de plantas (σ) e insectos (τ). A pesar de que, con este enfoque, se tiene una distribución de probabilidad para B , esta no es computacionalmente efectiva, por lo tanto, se puede utilizar técnicas de Monte Carlo para aproximar la distribución de probabilidad.

Para esto, se tiene que el número de visitas M_{ij} es una variable aleatoria de

Poisson con media

$$\mu_{ij} = C\sigma_i\tau_j(1 + rB_{ij}).$$

De esta manera, si se considera que la probabilidad de que un insecto sea polinizador de una planta como una variable uniforme de probabilidad ρ , se tiene que

$$P(B, \theta|M) \propto P(\theta) \prod_{ij} (1 - \rho)^{1-B_{ij}} \rho^{B_{ij}} \frac{\mu_{ij}^{M_{ij}}}{M_{ij}!} e^{-\mu_{ij}}.$$

Con esto, se puede utilizar un algoritmo de maximización de la esperanza para obtener los parámetros del modelo y, de esta manera, obtener la distribución de probabilidad de B .

Esta metodología proporcionó resultados coherentes con las propiedades de la distribución de conexión que fue propuesta como un invariante en este tipo de redes en [38].

Capítulo 3

Implementación del trabajo

Durante el desarrollo de este trabajo, se ha seguido un ciclo de trabajo iterativo, en el que se han ido realizando pruebas y modificaciones sobre el código, hasta obtener los resultados deseados. En este capítulo se describirán los pasos seguidos para la implementación del trabajo, así como las herramientas utilizadas para ello. Este proceso se puede dividir en las siguientes etapas:

- **Preprocesamiento de datos:** en esta etapa se ha realizado manipulación y limpieza de datos, de tal forma que estos se encuentren en un formato adecuado para ser procesados por los modelos. A la par, se ha realizado un análisis exploratorio de los datos, con el fin de obtener información relevante sobre los mismos.
- **Aplicación de un modelo basado en YOLOv5:** en esta etapa se ha entrenado modelo basado en YOLOv5, con el fin de realizar la detección de insectos en las imágenes.
- **Aplicación de un modelo basado en EfficientNet:** en esta etapa se hace lo análogo a la etapa anterior, pero con un modelo basado en EfficientNet.
- **Evaluación de los modelos:** en esta etapa se evalúan los modelos, con el fin de determinar cuál de ellos es el más adecuado para el problema.
- **Reconstrucción de red de polinizadores:** en esta etapa, utilizando el modelo seleccionado, se reconstruye la red de polinizadores.

3.1. Entorno de trabajo

En general, se utilizó el lenguaje de programación Python para el desarrollo de este trabajo. En específico, se desarrolló el código en el entorno Jupyter No-

tebook. Para el modelo basado en YOLOv5, se utilizó la librería *Pytorch*, mientras que para el modelo basado en EfficientNet, se utilizó la librería *Tensorflow*. Finalmente, para la reconstrucción, análisis y visualización de la red de polinizadores, se utilizó la librería *NetworkX*.

En cuanto al hardware utilizado, se emplearon dos instancias. Para el procesamiento de datos y el entrenamiento del modelo basado en EfficientNet, se utilizó un computador de escritorio con procesador AMD Rayzen 7 de 8 núcleos y 40 GB de memoria RAM, adicionalmente, se utilizó una tarjeta gráfica NVIDIA GeForce RTX 3060 de 12 GB de memoria dedicada. Sin embargo, esto no fue suficiente para los modelos basados en YOLOv5, por lo cual se utilizó Google Colab, con una GPU Nvidia A100 de 40 GB de memoria dedicada.

3.2. Procesamiento de datos

Los datos para este estudio se obtuvieron mediante un sistema de monitoreo automático desarrollado en [39], que consistió en un ensamblaje de cámara controlado por un ordenador de placa única, específicamente una Raspberry Pi 4. Esta configuración fue elegida por su óptima resolución óptica para objetos pequeños como insectos y flores, y su capacidad para programar ciclos de vídeo en el campo. El sistema incluía una cámara Module V2 de 5MP, que proporcionaba videos de 1080p a 30 fotogramas por segundo. Para la autonomía, se utilizó una batería portátil de 1000mAh Li-polímero, que permitía hasta seis horas de funcionamiento, y los videos se almacenaban en un USB de 32 GB. Todo el conjunto estaba protegido por una caja de plástico y montado en un trípode, diseñado para registrar interacciones planta-polinizador.

Para la recopilación de datos, se programó la Raspberry Pi 4 para automatizar la grabación de videos en ciclos de 10 minutos durante períodos de una hora y media, usando un archivo '.ssh' y Python 3.0. Además de los videos capturados por este sistema, se desarrolló una biblioteca fotográfica de 12000 imágenes de insectos polinizadores. Los objetos en los fotogramas de los videos se etiquetaron manualmente utilizando el programa LabelImg.

De manera específica, para este proyecto, se seleccionó 5445 imágenes de flores con polinizadores, las cuales fueron seleccionadas por el autor de [39] como las de mejor calidad. Estas imágenes contaban con diferentes dimensiones, siendo mayoritario el tamaño de 1280×962 píxeles. El ancho de las imágenes era fijo, y la distribución del alto de las imágenes, en escala logarítmica, se puede apreciar en la Figura 3.1.

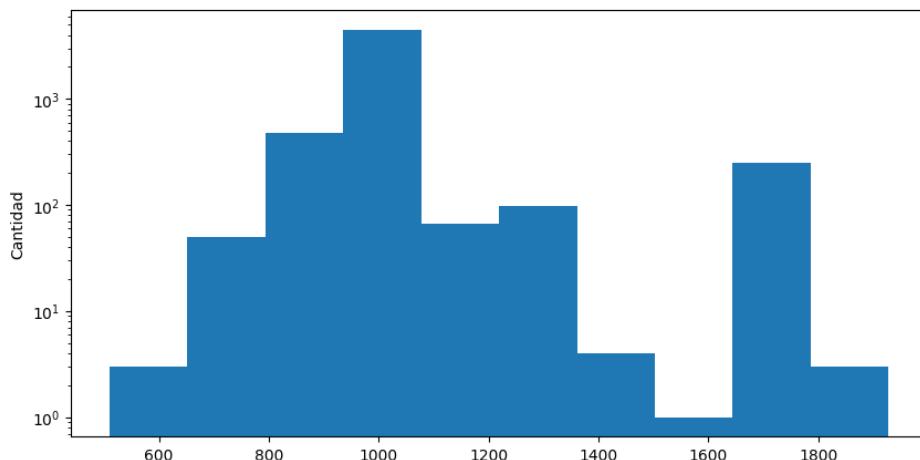


Figura 3.1: Distribución del alto de las imágenes, en escala logarítmica.

Dado que los modelos a utilizarse trabajan con imágenes del mismo tamaño, se debe estandarizar los altos de las imágenes. Para esto, se tomó como medida estándar la moda de las alturas. En las imágenes con un alto menor a este valor, se completó con un fondo blanco, mientras que en las que poseían un alto mayor a esto, se las recortó.

De cada imagen se contaba con un archivo XML, que contenía la información de las coordenadas de los polinizadores en la imagen, así como la especie de cada uno de ellos. Un ejemplo de este archivo se puede apreciar en el Código 3.1.

```
1  <annotation>
2  <folder>fotos transformades</folder>
3  <filename>P1050344.JPG</filename>
4  <path>F:\polinizadores\fotos insectes\fotos transformades\P1050344.JPG</
   path>
5  <source>
6    <database>Unknown</database>
7  </source>
8  <size>
9    <width>1280</width>
10   <height>960</height>
11   <depth>3</depth>
12 </size>
13 <segmented>0</segmented>
14 <object>
15   <name>Bee</name>
16   <pose>Unspecified</pose>
17   <truncated>0</truncated>
18   <difficult>0</difficult>
19   <bndbox>
20     <xmin>344</xmin>
```

```
21      <ymin>243</ymin>
22      <xmax>569</xmax>
23      <ymax>429</ymax>
24    </bndbox>
25  </object>
26 </annotation>
```

Código 3.1: Ejemplo de archivo XML de información de imágenes.

De la información de las imágenes se tomó el nombre de la imagen (filename), el nombre del objeto (object, name) y la posición (object, bndbox). A partir del nombre del objeto, se realizó una clasificación de los polinizadores en 7 categorías: Bee, Lepidoptera, Diptera, Coleoptera, Others, Wasp y Hoverfly. Cabe señalar que, en algunas imágenes, se encontraban más de un polinizador. En la Figura 3.2 se puede apreciar un ejemplo de cada una de estas categorías.

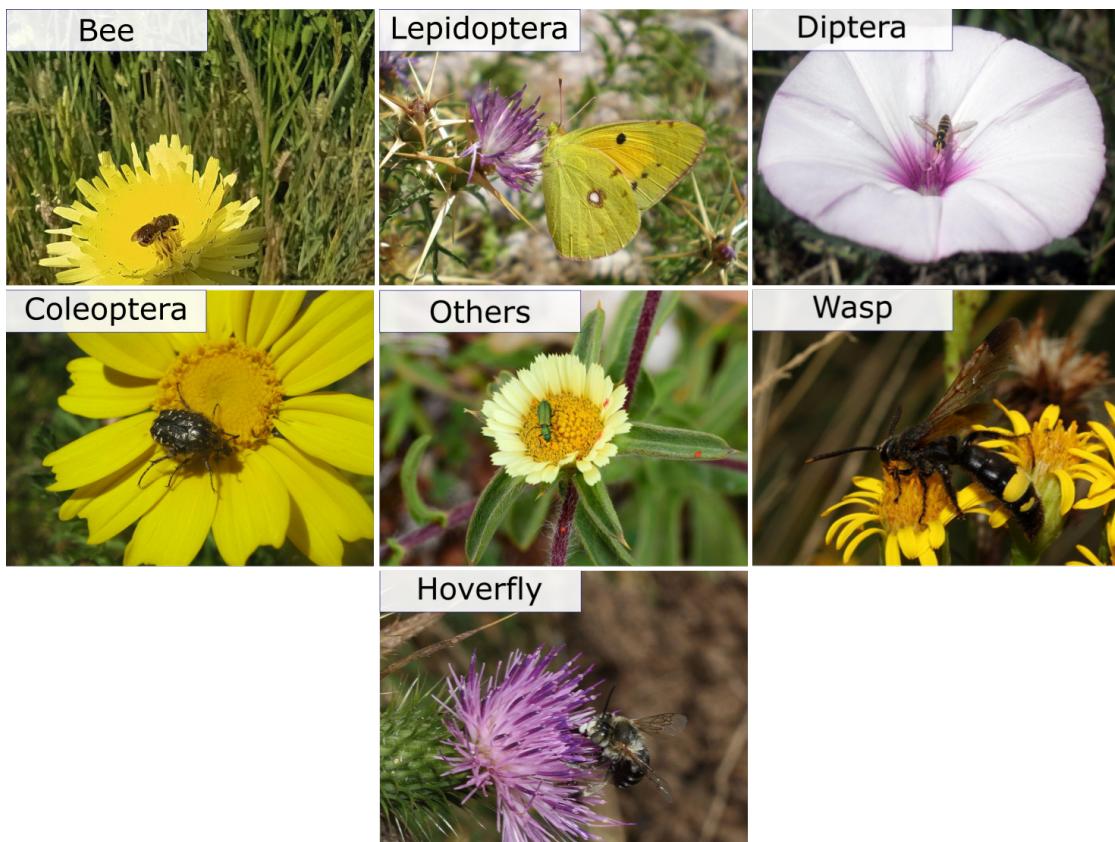


Figura 3.2: Ejemplo de cada una de las especies de polinizadores.

La distribución de las especies, por categoría, en las imágenes se puede apreciar en la Figura 3.3, en escala logarítmica. Se observa que la especie más abundante es «Bee», con 5189, seguida de «Lepidoptera», con 510, y «Diptera», con 305. Por otro lado, las especies menos abundantes son «Wasp», con 109, y «Ho-

verfly», con 81. En total, en las 5445 imágenes se tienen 6681 polinizadores, por lo cual, para el modelo, se tiene en la práctica 6681 imágenes etiquetadas.

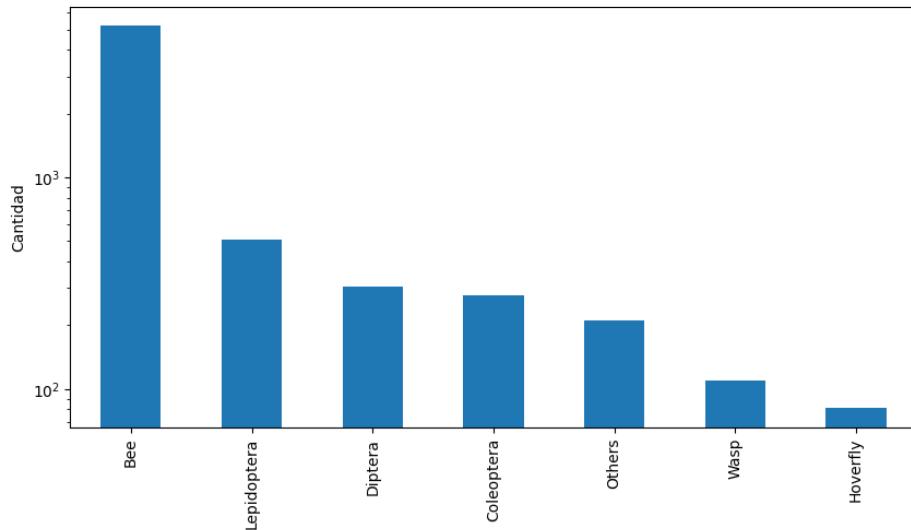


Figura 3.3: Distribución de las especies en las imágenes, en escala logarítmica.

Dada la abundante cantidad de imágenes de la especie «Bee», se decidió realizar un submuestreo de estas imágenes, con el fin de balancear la cantidad de imágenes de cada especie. Para ello, se descartaron, mediante inspección directa, imágenes similares, de una misma planta, que contenga la especie «Bee» o «Lepidoptera». El resultado de este proceso se puede apreciar en la Figura 3.4. De esta manera, el conjunto de datos con el que se trabajará cuenta con 2121 imágenes.

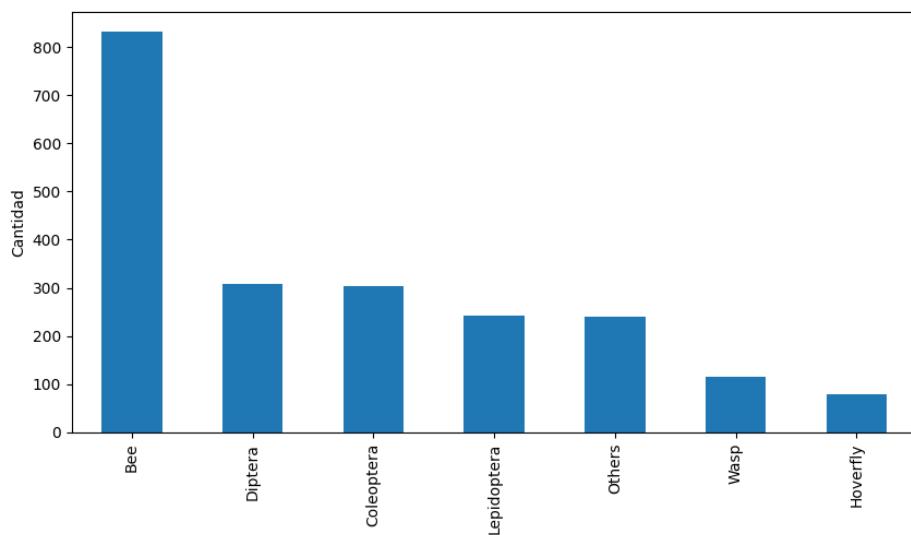


Figura 3.4: Submuestreo de imágenes de la categoría «Bee» y «Lepidoptera».

3.2.1. Aumento de datos

Para el aumento de datos, se procedió a realizar reflexiones. En las dos clases menos abundantes, «Wasp» y «Hoverfly», se realizó reflexión horizontal y vertical, mientras que en las demás clases, salvo «Bee», se realizó reflexión horizontal. En la Figura 3.5 se puede apreciar un ejemplo de este proceso.

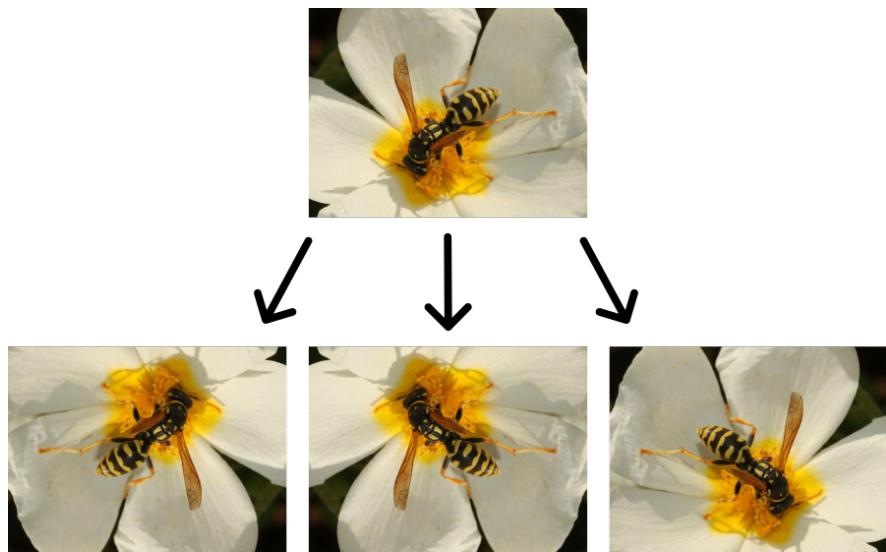


Figura 3.5: Ejemplo de reflexiones realizadas.

Con esto, se obtuvieron 3415 imágenes y se mejoró la distribución de las mismas en las diferentes categorías. Esta distribución se puede apreciar en la Figura 3.6.

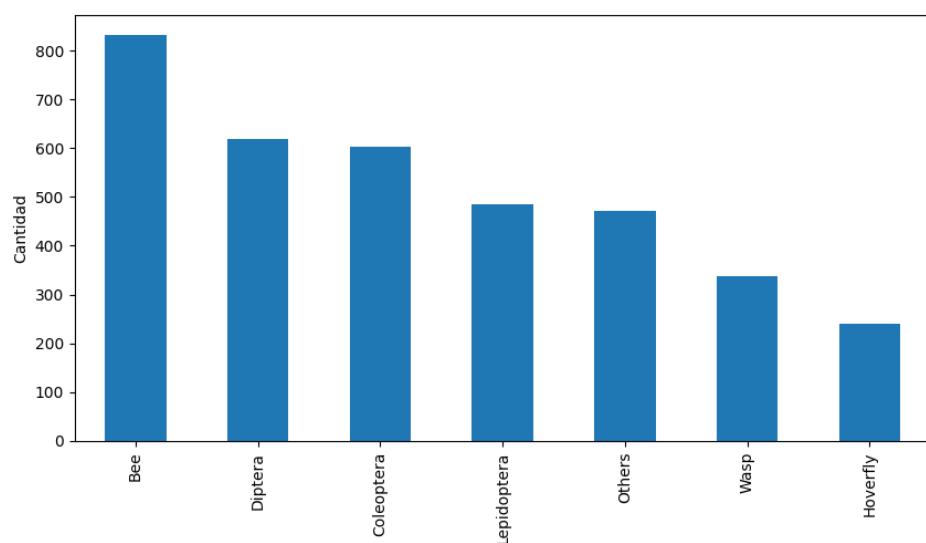


Figura 3.6: Distribución de las especies en las imágenes luego del aumento de datos.

Con estos resultados, se dividió los datos en los conjuntos de entrenamiento, validación y prueba. La distribución de los datos en estos conjuntos muestra en la Tabla 3.1.

	Entrenamiento	Validación	Prueba
Bee	184	46	581
Lepidoptera	184	46	364
Diptera	184	46	322
Coleoptera	184	46	240
Others	184	46	191
Wasp	184	46	97
Hoverfly	184	46	10
Total	1288	322	1805

Tabla 3.1: Distribución de los datos en los conjuntos de entrenamiento, validación y prueba.

3.3. Aplicación de modelos

3.3.1. YOLOv5

Para iniciar el análisis de los modelos, se procedió a realizar una prueba con el modelo YOLOv5, con el fin de determinar si este era capaz de detectar los polinizadores en las imágenes. Para ello, se utilizó el conjunto de prueba para determinar las detecciones realizadas por el modelo; cabe recalcar que, el modelo YOLOv5 está entrenado con el conjunto COCO, el cual no cuenta con polinizadores. Por esto, se obtuvieron diversos objetos detectados, los primeros diez objetos más detectados se pueden apreciar en la Tabla 3.2.

Objeto	Cantidad
no detection	853
bird	322
banana	109
dining table	98
apple	67
potted plant	25
microwave	19
orange	19
broccoli	18
chair	16

Tabla 3.2: Objetos detectados por el modelo YOLOv5.

Cabe destacar que, en las detecciones, se observan pocas referentes a plantas y ninguna referente a insectos. Adicionalmente, en los casos de detección

de plantas, no se encuentran relacionados con las especies que se tienen en las imágenes.

Con este resultado base, se procedió a entrenar el modelo YOLOv5 con los datos de polinizadores. Para ello, se utilizó el conjunto de entrenamiento, y se validó con el conjunto de validación. El entrenamiento se realizó con 100 épocas, con un tamaño de lote de 32 imágenes, y con un tamaño de imagen de 1280 píxeles de ancho.

Para evaluar el rendimiento del modelo YOLOv5 entrenado, se emplearon dos tipos de gráficos basados en el conjunto de validación: las curvas de precisión-confianza y de exhaustividad-confianza.

- **Curvas de Precisión-Confianza (*precision-confidence*):** Estas curvas ilustran la relación entre la precisión del modelo y el nivel de confianza (o probabilidad) asignado a sus predicciones. La precisión se define como la proporción de identificaciones correctas (verdaderos positivos) entre todas las identificaciones realizadas (verdaderos positivos más falsos positivos). En este gráfico, el eje horizontal representa la confianza del modelo en sus predicciones, mientras que el eje vertical muestra la precisión correspondiente a cada nivel de confianza. Un modelo ideal tendría una curva que se mantiene alta a medida que aumenta el nivel de confianza, indicando que puede mantener una alta precisión incluso cuando está muy seguro de sus predicciones.
- **Curvas de Exhaustividad-Confianza (*recall-confidence*):** Estas curvas muestran cómo la exhaustividad (o recuperación) del modelo varía con diferentes niveles de confianza. La exhaustividad se refiere a la proporción de casos positivos reales (verdaderos positivos) que el modelo ha identificado correctamente, respecto al total de casos positivos reales (verdaderos positivos más falsos negativos). En estas curvas, el eje horizontal representa la confianza, y el eje vertical muestra la exhaustividad. Una curva ideal sería aquella que se mantenga alta en todos los niveles de confianza, lo que indica que el modelo es capaz de identificar la mayoría de los casos positivos reales incluso con un alto nivel de confianza en sus predicciones.

En lo que respecta a la curva de precisión-confianza (Figura 3.7), se tiene que la precisión óptima para todas las clases se alcanza a un umbral de confianza de aproximadamente 0.948, indicando un alto grado de fiabilidad del modelo en este punto. Sin embargo, las clases individuales muestran variabilidad en la precisión; «Bee» mantienen valores altos de precisión incluso a niveles bajos de confianza, sugiriendo un reconocimiento robusto, mientras que otras clases,

particularmente las agrupadas en «Others», exhiben una precisión más baja, lo que puede sugerir una mayor dificultad en su detección.

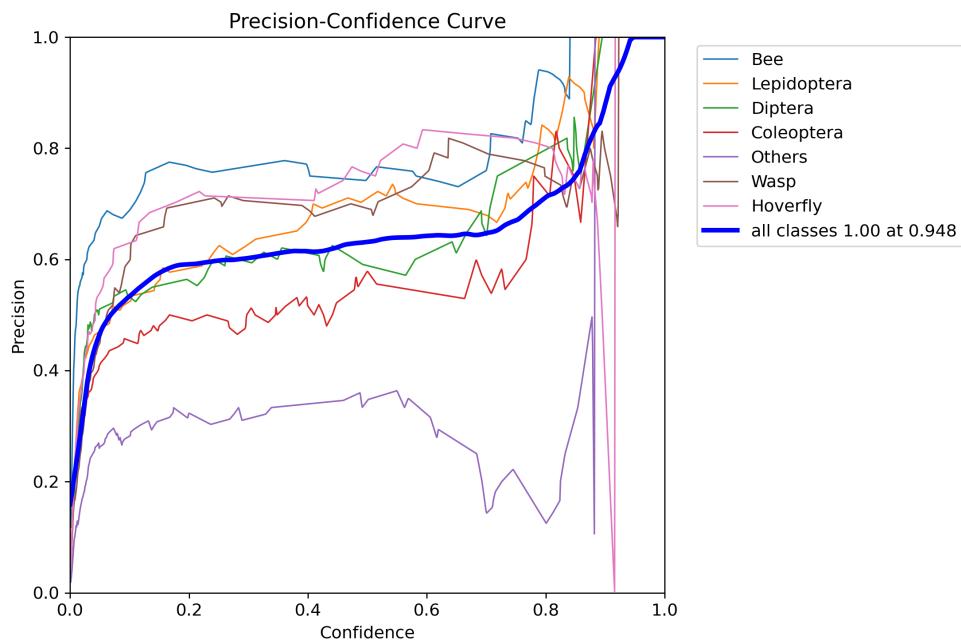


Figura 3.7: Curvas de precisión-confianza modelo YOLOv5 en las imágenes de validación.

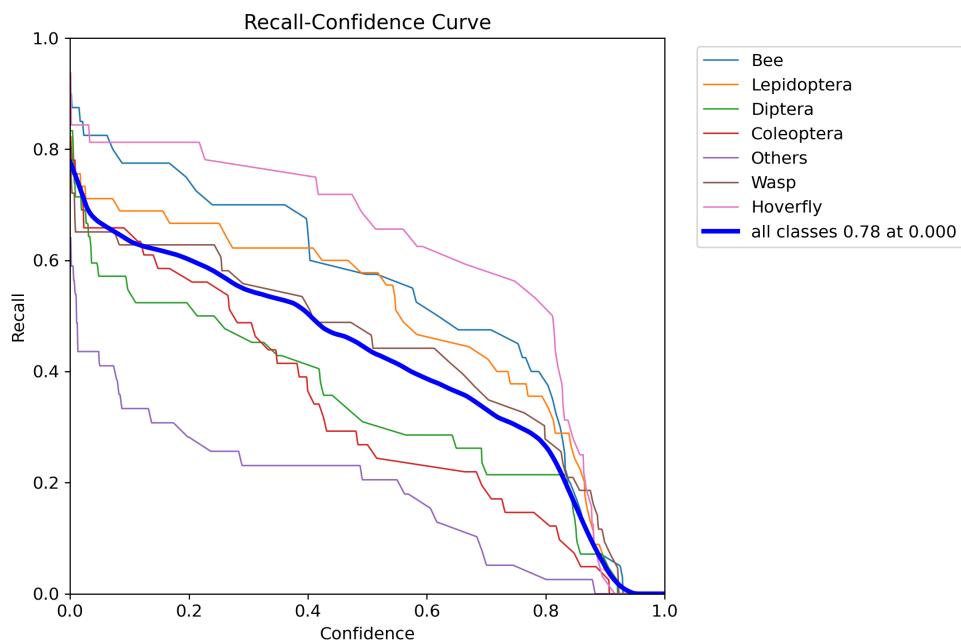


Figura 3.8: Curvas de exhaustividad-confianza modelo YOLOv5 en las imágenes de validación.

Por el lado de la curva exhaustividad-confianza (Figura 3.8), demuestra que la exhaustividad general para la detección de insectos se mantiene en un nivel notable de 0.78 incluso a un umbral de confianza cero, subrayando la habilidad del modelo para detectar una gran proporción de las instancias positivas desde el inicio. Sin embargo, la exhaustividad disminuye conforme se incrementa el umbral, un comportamiento esperado al elevar la rigurosidad para la aceptación de las predicciones. Las distintas clases de insectos exhiben divergencias en la exhaustividad a lo largo de los umbrales, lo que refleja diferencias en la facilidad de detección o en la representación en el conjunto de datos.

Adicionalmente, se obtuvo la matriz de confusión con las predicciones realizadas sobre el conjunto de validación (Figura 3.9). Se observa un alto grado de precisión en la detección de «Bee» y «Hoverfly», con valores de 70% y 78%, respectivamente, indicando un reconocimiento efectivo de estas clases. No obstante, la confusión entre clases de insectos y el fondo, como se evidencia en los «Diptera» y «Coleoptera» mal clasificados como fondo en un 45% y 45% de los casos, respectivamente, sugiere un reto en la discriminación de características entre insectos y su entorno. Por otro lado, es remarcable el hecho de que casi no existe confusión entre las diferentes clases (valores fuera de la diagonal), llegando solo al 2%.



Figura 3.9: Matriz de confusión del modelo YOLOv5.

Por otro lado, se puede contrastar las detecciones realizadas por el modelo con las detecciones realizadas por los humanos. En la Figura 3.10, se puede apreciar un ejemplo de los resultados en validación. Los colores indican la clase y el valor indica la confianza de la clasificación.



Figura 3.10: Comparación entre detecciones realizadas por humanos (arriba) y por el modelo YOLOv5 (abajo); el color muestra la categoría y el valor indica la confianza de la clasificación.

La Figura 3.10 muestra una comparación entre la anotación manual (arriba) y las detecciones automatizadas realizadas por el modelo YOLOv5 (abajo). Es notable que en la categoría «Others», el modelo no logró detectar el insecto que sí fue identificado por un observador humano. En el caso de «Bee», el modelo proporciona un nivel de confianza relativamente bajo de 0.4, lo que podría indicar una menor certeza en su clasificación o una posible ambigüedad visual en la imagen. Sin embargo, en el caso de «Diptera», no solo la clasificación es correcta sino que la confianza es más alta (0.7), y el recuadro de detección parece más ajustado alrededor del insecto en comparación con la anotación humana.

Finalmente, contando ya con el modelo entrenado, se procedió a realizar las detecciones en el conjunto de prueba. La matriz de confusión se puede apreciar en la Figura 3.11. Se muestra una tendencia predominante hacia la no detección de insectos, más que hacia la confusión entre clases. Los valores significativos en la columna «Background» sugieren que el modelo frecuentemente clasifica regiones que contienen insectos como parte del fondo, particularmente en las categorías de «Coleoptera» y «Others». Sin embargo, en la confusión entre clases, se obtiene valores por debajo del 2%.

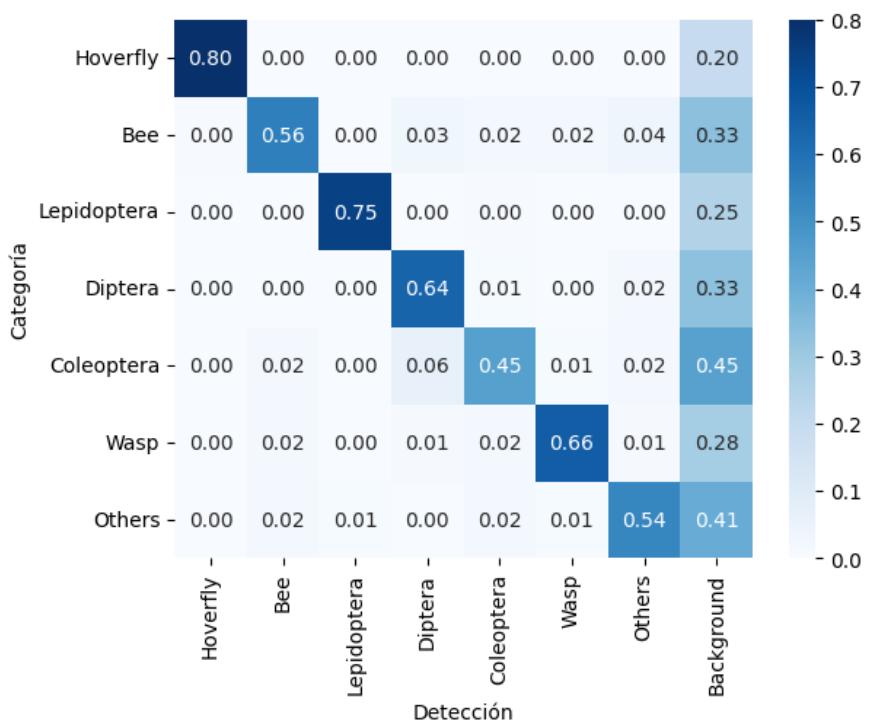


Figura 3.11: Matriz de confusión del modelo YOLOv5 en el conjunto de prueba.

Con esto, se calculó la precisión y exhaustividad por clase, los resultados pueden verse en la Tabla 3.3. La precisión se mantiene alta a través de todas las categorías, sobresaliendo en «Lepidoptera» con un 0.98, lo que refleja una fuerte capacidad del modelo para identificar correctamente las instancias positivas. No obstante, la exhaustividad presenta una variabilidad significativa; categorías como «Hoverfly» exhiben una alta concordancia entre precisión y exhaustividad (0.80), sugiriendo una detección equilibrada, mientras que otras como «Coleoptera» muestran una exhaustividad de solo 0.45, implicando una proporción considerable de falsos negativos.

Categoría	Precisión	Exhaustividad
Bee	0.96	0.56
Coleoptera	0.90	0.45
Diptera	0.85	0.64
Hoverfly	0.80	0.80
Lepidoptera	0.98	0.75
Others	0.79	0.54
Wasp	0.82	0.66

Tabla 3.3: Métricas del modelo YOLOv5.

Adicionalmente, la exactitud global del modelo (*accuracy*) es del 58 %, lo cual refleja la proporción de predicciones correctas en relación con el total de predicciones realizadas. Este número es influido por todas las clases, por lo que la baja exhaustividad en algunas categorías puede afectar negativamente la exactitud general.

Dado que la deficiencia del modelo se encuentra en la falta de detección de insectos en algunas imágenes, esto puede deberse a que los insectos son demasiado pequeños para ser detectados, por esta razón, se recalcó la precisión general del modelo, filtrando por el tamaño de los insectos. El cálculo de tamaño de los insectos se lo realizó tomando en cuenta el área de detección del insecto, medido en pixeles. Con esto, se obtuvieron los resultados que se presentan en la Tabla 3.4.

Área mínima (%)	Cantidad de imágenes	Exactitud general
0.0	1891	0.58
0.1	1795	0.61
0.5	1489	0.66
1.0	1233	0.67
2.5	812	0.71
5.0	502	0.72
10.0	271	0.74

Tabla 3.4: Precisión general modelo YOLOv5 en función del área de detección.

Con estos resultados se observa que, el tamaño del insecto incide de manera significativa en la precisión general de la clasificación. Un posible valor para tomar es que el insecto ocupe un espacio superior al 2.5 % del área total de la imagen, en la cual se obtiene una exactitud general de 0.71. Con esto, se recalcó la matriz de confusión, la cual se puede apreciar en la Figura 3.12. Adicionalmente, se recalcó la precisión y exhaustividad por clase, los resultados pueden verse en la Tabla 3.4.

La nueva matriz de confusión (Figura 3.12) tras el ajuste en el tamaño de los insectos detectados refleja una mejora notable en la detección de la categoría «*Hoverfly*», alcanzando una precisión del 100 %. Adicionalmente, la todas las categorías, salvo «*Others*», muestran una mejora en los resultados, superando en todas el valor de 0.6. Aunque la no detección ha disminuido todas las categorías, persiste como un problema significativo, especialmente en «*Others*».

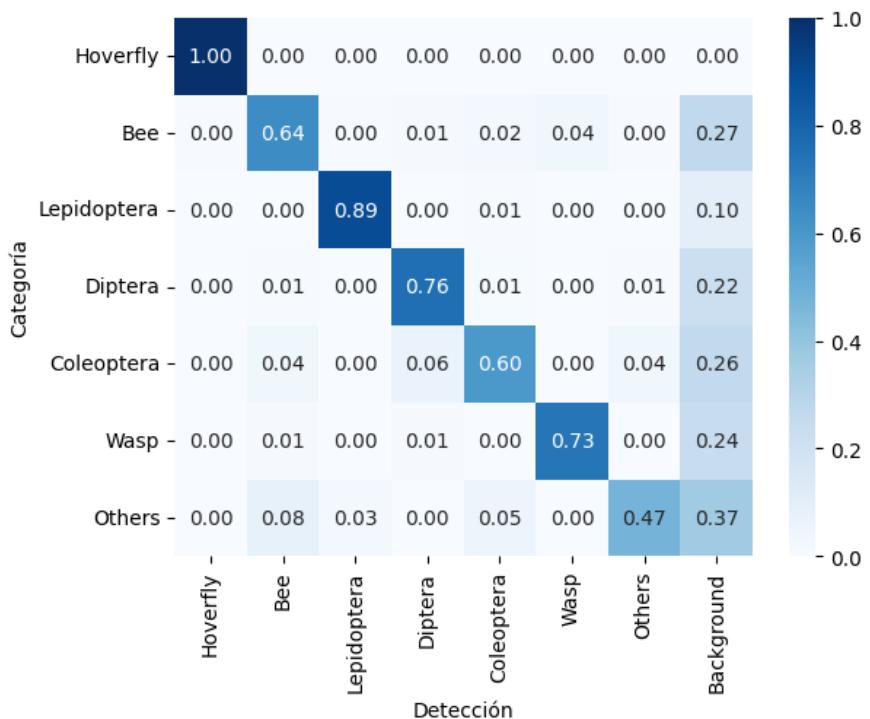


Figura 3.12: Matriz de confusión del modelo YOLOv5 en el conjunto de prueba, con un área mínima del 2.5 % del área total de la imagen.

Por otro lado, el modelo muestra una mejora en la precisión para la mayoría de las categorías (ver Tabla 3.5), con «Lepidoptera» alcanzando una precisión sobresaliente de 0.98 y una exhaustividad de 0.87. Las categorías «Bee», «Coleoptera», y «Wasp» muestran una precisión alta, pero persiste un nivel de exhaustividad bajo, lo que indica que aún se pasan por alto instancias reales. La categoría «Others» continúa siendo la más desafiante para el modelo, con las métricas más bajas en precisión y exhaustividad, reflejando dificultades tanto en la detección como en la clasificación correcta.

Categoría	Precisión	Exhaustividad
Bee	0.94	0.64
Coleoptera	0.88	0.60
Diptera	0.92	0.76
Hoverfly	0.88	1.00
Lepidoptera	0.98	0.87
Others	0.72	0.47
Wasp	0.86	0.73

Tabla 3.5: Métricas del modelo YOLOv5, con un área mínima del 2.5 % del área total de la imagen.

Validación cruzada

Se implementó un enfoque de validación cruzada con 5 *folds* para evaluar la generalización del modelo YOLOv5 en la tarea de clasificación. Este método consiste en dividir el conjunto de datos de forma aleatoria en cinco partes (o *folds*), utilizando cuatro de ellas para el entrenamiento y la restante para la validación, repitiendo este proceso cinco veces de manera que cada *folds* sirva una vez como conjunto de validación.

Para integrar los resultados de los cinco modelos generados en la validación cruzada, se adoptó el criterio de unificarlos bajo la máxima media del nivel de confianza de las predicciones de cada modelo. Esto significa que, para cada observación, se seleccionó la predicción con la mayor confianza promedio entre los diferentes modelos. Este enfoque maximiza la certeza en las predicciones finales y contribuye a una evaluación más precisa de la estabilidad del modelo.

Con esto, se calculó la precisión y exhaustividad por clase, los resultados pueden verse en la Tabla 3.6.

Categoría	Precisión	Exhaustividad
Bee	0.89	0.69
Coleoptera	0.91	0.55
Diptera	0.85	0.70
Hoverfly	0.32	0.70
Lepidoptera	0.96	0.78
Others	0.68	0.57
Wasp	0.75	0.77

Tabla 3.6: Métricas del modelo YOLOv5, con validación cruzada.

Se presentan variaciones interesantes en comparación con los resultados originales. En la categoría «Bee», se observa una disminución en la precisión (de 0.96 a 0.89) pero un aumento en la exhaustividad (de 0.56 a 0.69), lo que indica una mejor identificación de esta categoría a costa de una precisión ligeramente reducida. Similarmente, en «Coleoptera» y «Diptera», la precisión se mantiene alta, con un notable aumento en la exhaustividad para «Diptera». Sin embargo, la categoría «Hoverfly», aunque muestra un aumento en la exhaustividad, tiene una precisión significativamente baja, posiblemente debido al número limitado de muestras. En contraste, la categoría «Others» muestra un descenso tanto en precisión como en exhaustividad.

La exactitud global del modelo mejoró notablemente, pasando de 0.58 en los resultados originales a 0.67 en la validación cruzada. Este incremento es un indicador positivo de que el modelo generaliza mejor sus predicciones a través de diferentes conjuntos de datos, lo que sugiere una robustez incrementada. Las

fluctuaciones en precisión y exhaustividad, aunque son comunes en el proceso de validación y pruebas, indican que el modelo mantiene una buena capacidad de generalización. La mejora en la exactitud global, junto con los ajustes observados en otras métricas, sugiere que el modelo es confiable y efectivo, con un nivel de *overfitting* controlado.

3.3.2. EfficientNet

Se siguió la misma metodología que en el modelo anterior. Se realizó una prueba con el modelo EfficientNetB7, con el fin de determinar si este era capaz de detectar los polinizadores en las imágenes. Para ello, se utilizó el conjunto de prueba para determinar las detecciones realizadas por el modelo; cabe recalcar que, el modelo EfficientNet [23] está entrenado con el conjunto ImageNet, el cual no cuenta con polinizadores. Por esto, se obtuvieron diversos objetos detectados, los primeros diez objetos más detectados se pueden apreciar en la Tabla 3.7.

Objeto	Cantidad
matchstick	779
jack-o'-lantern	233
nematode	181
daisy	80
abaya	66
spotlight	54
velvet	49
feather boa	31
candle	22
sea urchin	21

Tabla 3.7: Objetos detectados por el modelo EfficientNet.

Con este resultado base, se procedió a entrenar el modelo EfficientNet con los datos de polinizadores. Para ello, se utilizó el conjunto de entrenamiento, y se validó con el conjunto de validación. El entrenamiento se realizó con 50 épocas, con un tamaño de lote de 32 imágenes, y con un tamaño de imagen de 600 píxeles, para esto, se realizó un reescalado de las imágenes. Adicionalmente, se agregó una capa *softmax* al final del modelo, con el fin de realizar la clasificación de las detecciones.

Las curvas de aprendizaje obtenidas se pueden apreciar en la Figura 3.13. Estas curvas muestran una precisión de entrenamiento y validación inusualmente bajas, lo que sugiere un posible subajuste y la necesidad de una optimización en el proceso de entrenamiento. La baja precisión, con valores fluctuantes entre

0.11 y 0.18, indica que el modelo tiene dificultades para captar las características distintivas de los datos.

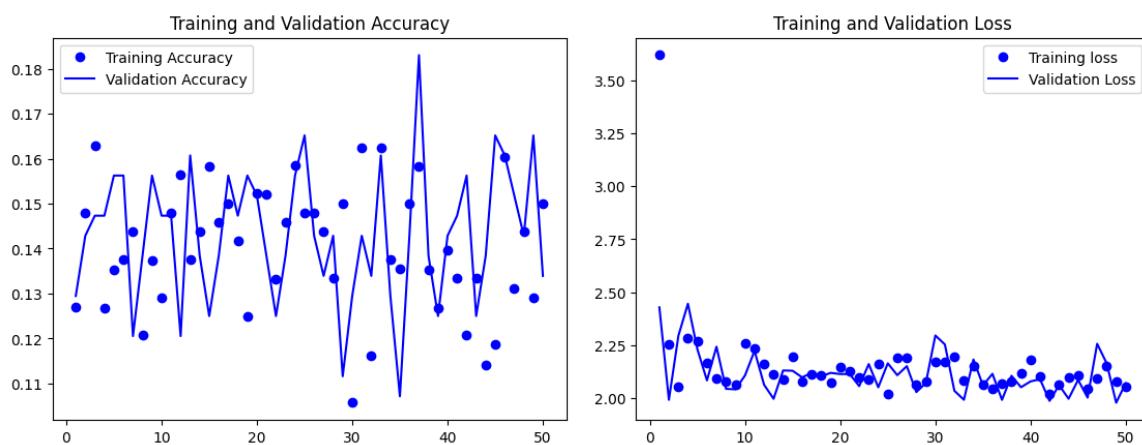


Figura 3.13: Curvas de aprendizaje del modelo EfficientNet.

Se puede apreciar que no existe un aprendizaje, esto puede deberse a que EfficientNet no toma en cuenta el bounding box de las imágenes, por lo que realiza una detección en general de la imagen, y no de los polinizadores en específico. Finalmente, contando ya con el modelo entrenado, se procedió a realizar las detecciones en el conjunto de prueba, sin resultados satisfactorios.

3.4. Reconstrucción de red de polinizadores

Con el modelo YOLOv5 entrenado, se procedió a realizar las detecciones en el conjunto de prueba. En este punto, solo se tomaron las imágenes originales, sin sus reflexiones. Se tiene un total de 1110 imágenes. Dentro de estas, el modelo detectó insectos en 701 imágenes. La distribución de los insectos en las imágenes se puede apreciar en la Figura 3.14. Se observa que se tiene una distribución similar entre las observaciones reales y las estimaciones del modelo YOLOv5 para la mayoría de las categorías de insectos, lo que valida el buen funcionamiento del método aplicado. Sin embargo, es notable la discrepancia en la categoría «Bee», donde el modelo subestima claramente su presencia.

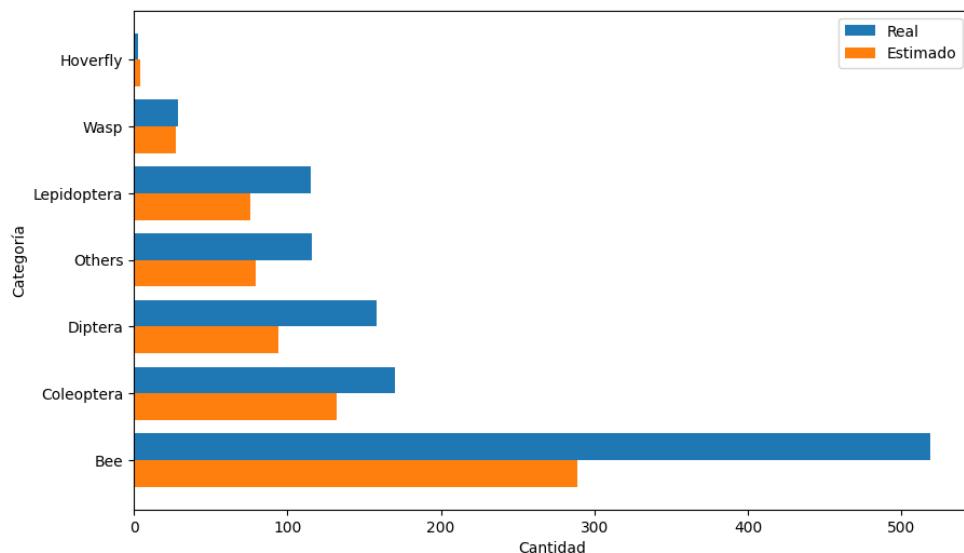


Figura 3.14: Distribución de los insectos en las imágenes.

Por el lado de las plantas, se utilizó la herramienta *My PlantNet API* [5] para la identificación de las plantas en las imágenes. A partir de los datos proporcionados por la API, se tomó la Familia de cada planta. En total, se detectaron 40 familias de plantas, la distribución de estas se puede apreciar en la Figura 3.15. Se puede apreciar que, la familia más abundante es Asteraceae, con más de 300 imágenes. Por otro lado, existen algunas familias con una sola imagen.

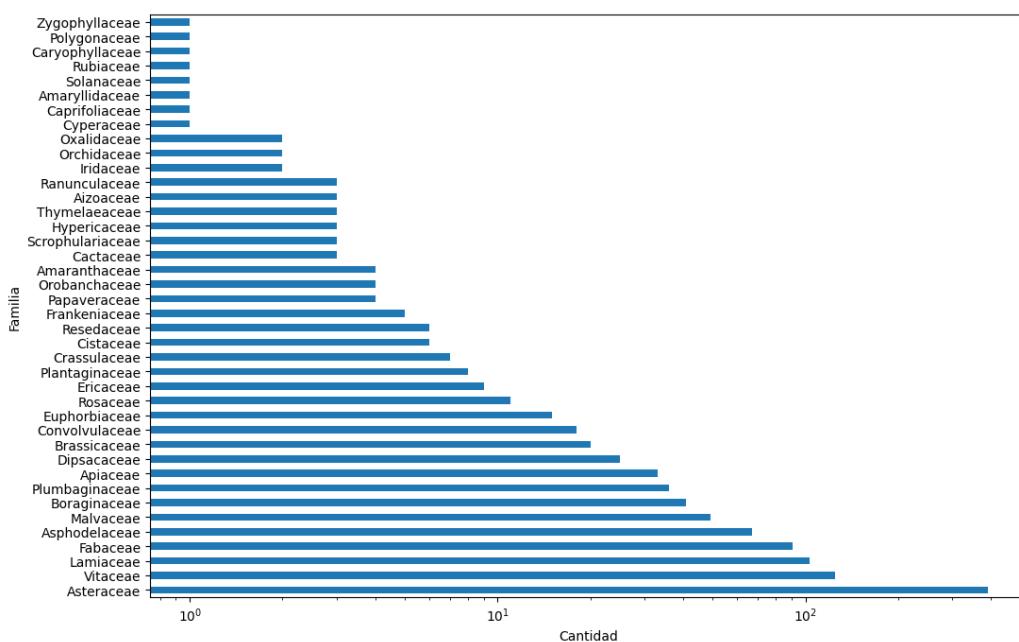


Figura 3.15: Distribución de las familias de plantas en las imágenes, en escala logarítmica.

Con esto, se llevó a cabo la construcción de dos redes planta-polinizador distintas. La primera red se derivó de las categorías identificadas a través de la observación humana, la cual se asume como el punto de referencia o red «real». Esta red proporciona una base para comparar la efectividad de las detecciones automáticas. La segunda red se generó a partir de las detecciones realizadas por el modelo YOLOv5. Al contrastar las dos redes, se puede evaluar no solo la capacidad de detección del modelo sino también su aplicabilidad en estudios ecológicos reales.

3.4.1. Red planta-polinizador real

La matriz de interacciones de la red real se puede apreciar en la Figura 3.16. Se puede observar que, la mayoría de las especies interactúan con la especie «Bee», mientras que las especies «Hoverfly» y «Wasp» tienen pocas interacciones.

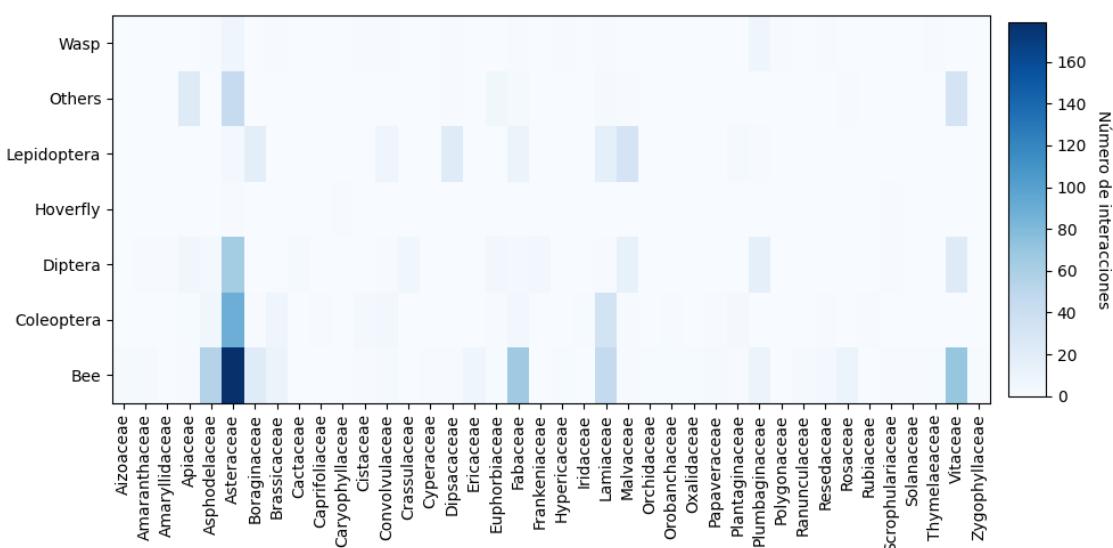


Figura 3.16: Matriz de interacciones de la red real.

A partir de esta matriz, con la metodología expuesta en el Capítulo 2, perteneciente a Young et al. [9], se procedió a estimar la red real. El resultado de aplicar esta metodología es una matriz que contiene la probabilidad de conexión en el grafo, es decir, la probabilidad de que un insecto sea un polinizador de una planta. Esta matriz se puede apreciar en la Figura 3.17.

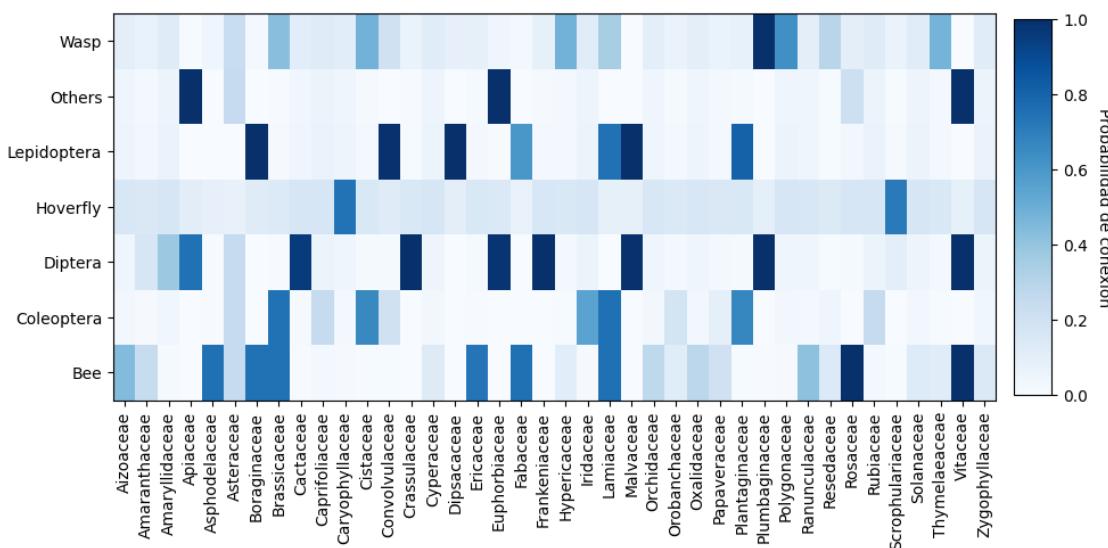


Figura 3.17: Matriz de probabilidades de la red real.

Seleccionando un umbral de confianza del 70 %, se construyó la red planta-polinizador considerada como la red real. Este valor se eligió estratégicamente por ser el más alto que garantiza que cada especie de insecto conserva al menos una interacción con una planta, asegurando así la integridad biológica de la red. La visualización del resultado de esta estimación se presenta en la Figura 3.18. Con esto, se tiene que los insectos son polinizadores de únicamente 20 familias de plantas.

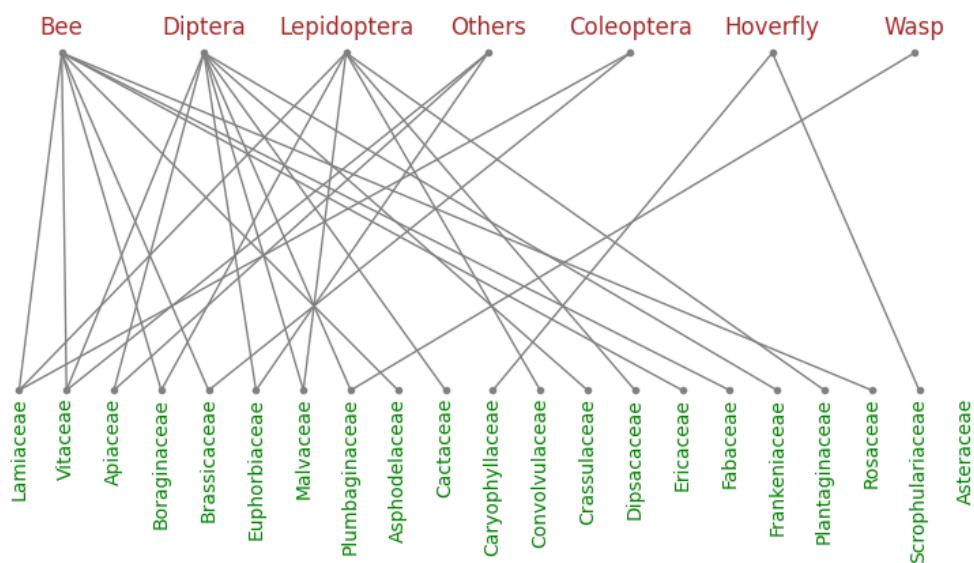


Figura 3.18: Red planta-polinizador real.

3.4.2. Red planta-polinizador estimada

La matriz de interacciones de la red estimada se puede apreciar en la Figura 3.19. Se puede apreciar que tiene las mismas características que la matriz de interacciones de la red real, con la diferencia de que, en esta matriz, la clase «Coleoptera» posee menos interacciones que en la matriz de la red real.

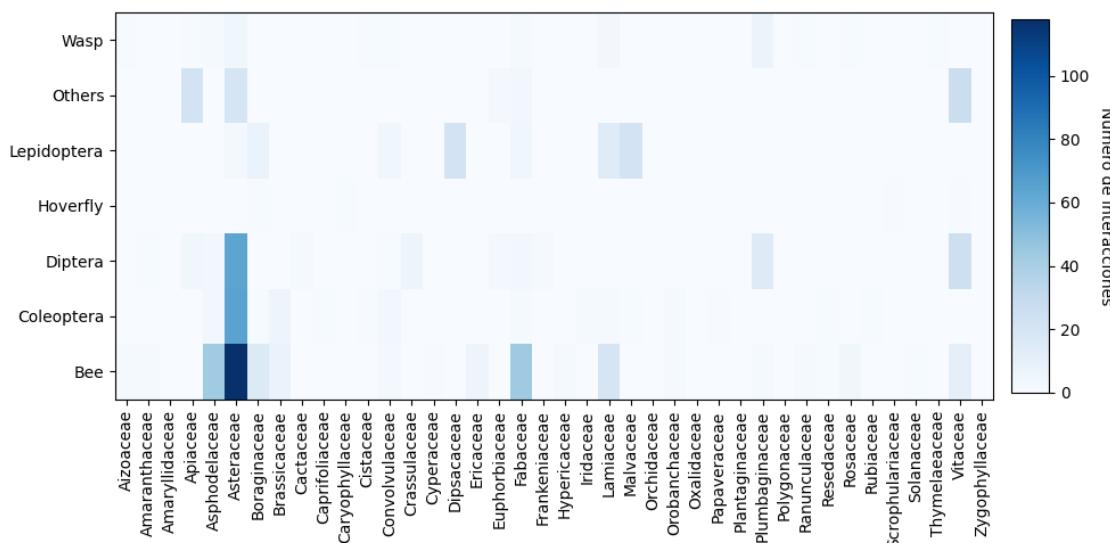


Figura 3.19: Matriz de interacciones de la red estimada.

A partir de esta matriz, al igual de lo descrito en la sección anterior, se procedió a estimar la red estimada. Con esto, se obtuvo la matriz de probabilidades de la red estimada, la cual se puede apreciar en la Figura 3.20.

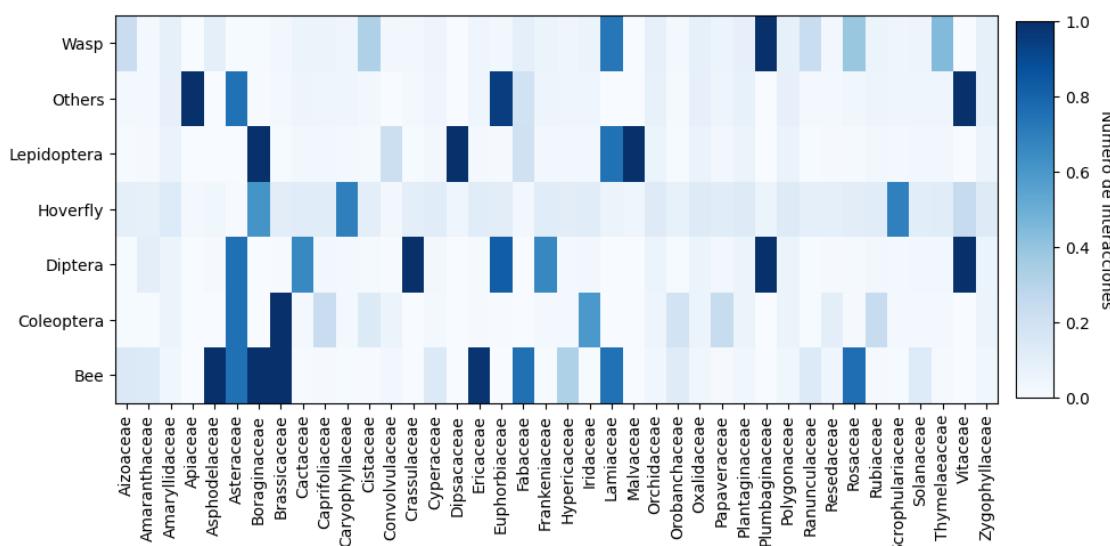


Figura 3.20: Matriz de probabilidades de la red estimada.

Tomando un nivel de confianza del 70 %, se procedió a generar la red estimada. El resultado de esta estimación se puede apreciar en la Figura 3.21.

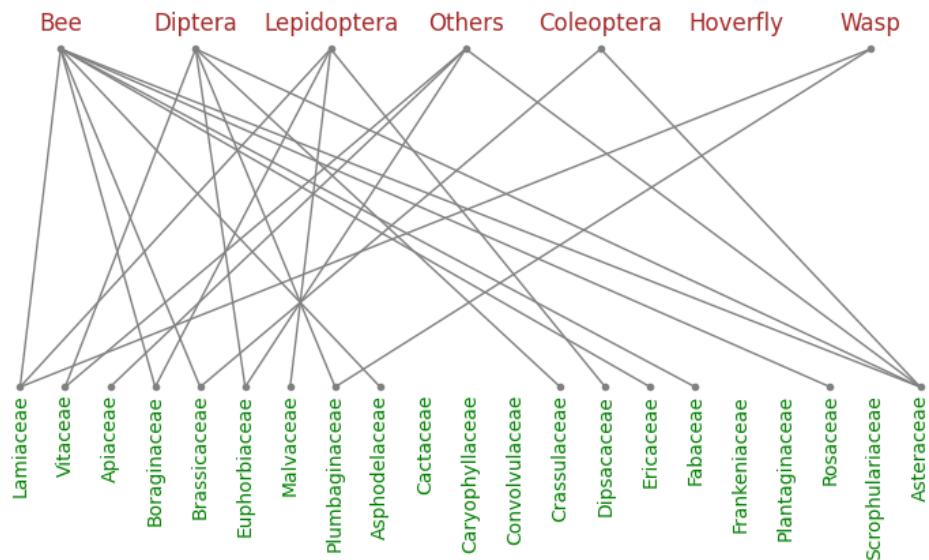


Figura 3.21: Red estimada.

3.5. Comparación de redes

Para la comparación de las redes, se tomará la componente conexa máximas de cada red. Para una comparación visual, se presentan la gráfica de ambas redes en la Figura 3.22.

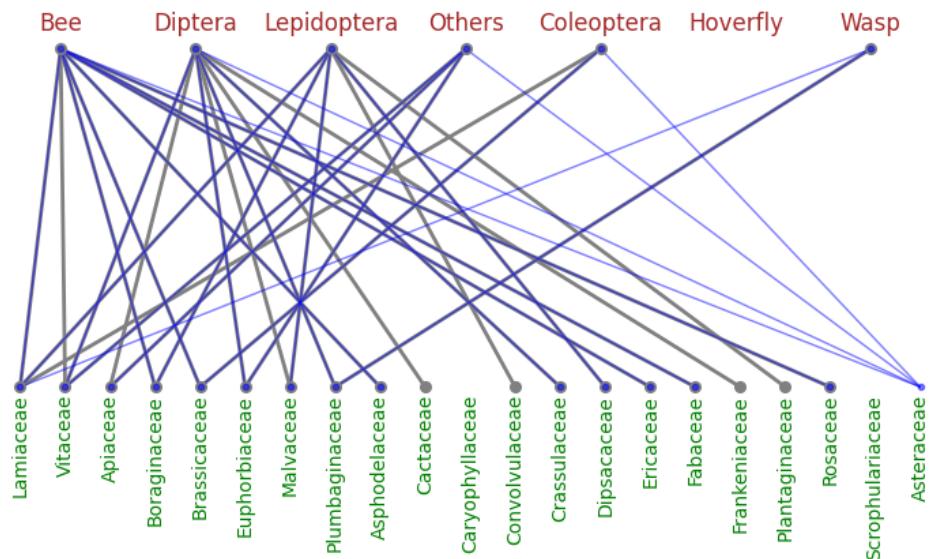


Figura 3.22: Comparación de las redes, en gris la red real y en azul la red estimada.

Se puede apreciar que, la red estimada posee una mayor cantidad de nodos y aristas que la red real. Por otro lado, se observa que, en la red estimada, la familia Asteraceae tiene una presencia mayor que en la red real.

Para comenzar el análisis cuantitativo de las redes planta-polinizador, se examinan los datos básicos de la red real y la red estimada. En la Tabla 3.8 se presentan las métricas de ambas redes. La red real cuenta con 24 nodos y 28 aristas, lo que sugiere una estructura de red con una variedad de especies y conexiones. Por otro lado, la red estimada presenta una ligera reducción con 21 nodos y 25 aristas, lo que podría indicar que el modelo de estimación no capturó todas las interacciones o especies presentes en la red real.

Métrica	Red real	Red estimada
Nodos	24	21
Aristas	28	25
Densidad	0.101	0.119
Grado medio	2.33	2.38

Tabla 3.8: Métricas de las redes.

La densidad de la red, que proporciona una idea de cuán conectada está la red, es ligeramente mayor en la red estimada (0.119) en comparación con la red real (0.101). Esto implica que, proporcionalmente, la red estimada tiene una mayor propensión de conexiones entre sus nodos respecto al número total de conexiones posibles. El grado medio, que es el número promedio de conexiones por nodo, es similar en ambas redes, con 2.33 para la red real y 2.38 para la red estimada, lo que indica que, en promedio, cada especie en ambas redes tiende a interactuar con un número comparable de otras especies.

Para analizar la similitud entre los grafos de las redes real y estimada, se procedió a calcular el distancias de *error de desplazamiento* (MDE) y la *similitud del coseno*, entre las filas de las matrices de adyacencia de los grafos. Estas dos métricas que ofrecen perspectivas distintas sobre la comparación de las matrices de adyacencia de los grafos. El error de desplazamiento mide la diferencia promedio en las conexiones entre dos grafos; un valor bajo indica que las posiciones de las conexiones en un grafo son muy similares a las del otro. Se calcula tomando la diferencia absoluta entre las matrices de adyacencia de los dos grafos, elemento por elemento, y luego obteniendo la media de estas diferencias. Por otro lado, la similitud del coseno mide el coseno del ángulo entre dos vectores en un espacio multidimensional que, en este caso, son las filas de las matrices de adyacencia de los grafos. Esta métrica varía de -1 a 1, donde 1 indica que los dos vectores son idénticos en orientación, y un valor cercano a 1 sugiere una alta similitud

entre los patrones de conexión de los grafos. Los resultados de estas distancias se pueden apreciar en la Tabla 3.9.

Distancia	Valor
Error de desplazamiento	0.053
Similitud del coseno	0.972

Tabla 3.9: Distancia entre las redes.

Los resultados obtenidos de estas medidas indican una alta similitud entre los grafos de las redes real y estimada. Un error de desplazamiento medio de aproximadamente 0.053 sugiere que las diferencias promedio en las conexiones entre los dos grafos son mínimas, lo que implica que la ubicación de las conexiones en la red estimada corresponde estrechamente a la red real. Además, una similitud del coseno media de aproximadamente 0.972 confirma esta observación, indicando que la orientación de los vectores (o las filas de las matrices de adyacencia) es muy similar.

Capítulo 4

Conclusiones y trabajo futuro

Este trabajo ha demostrado con éxito el desarrollo y la aplicación de una red neuronal convolucional (YOLOv5) para el reconocimiento automático de polinizadores en imágenes de flores, un paso significativo en la reconstrucción de redes planta-polinizador. Se ha procesado y analizado un conjunto de datos compuesto por 5445 imágenes, con especial atención en la estandarización del tamaño de las imágenes y la extracción precisa de información relevante mediante archivos XML.

Los resultados obtenidos del modelo YOLOv5 destacan una notable capacidad para detectar y clasificar polinizadores, aunque con ciertas limitaciones en la detección precisa de algunas especies y la diferenciación entre insectos y su entorno. Sin embargo, el uso de un tamaño mínimo de detección ha mejorado significativamente la precisión del modelo.

En lo que respecta a la validación cruzada para el modelo YOLOv5, los resultados obtenidos son alentadores y reflejan la robustez y la capacidad de generalización del modelo. A pesar de ciertas variaciones en la precisión y la exhaustividad entre las categorías, estas fluctuaciones son normales y esperadas en el proceso de validación.

Por otro lado, el modelo EfficientNet no logró un rendimiento satisfactorio, sugiriendo la necesidad de un enfoque más especializado para este tipo de datos.

En cuanto a la reconstrucción de redes de polinizadores, el análisis comparativo entre la red real y la estimada por el modelo ha proporcionado una visión valiosa sobre la eficacia de estas técnicas en la representación precisa de las interacciones en ecosistemas.

Las métricas calculadas sobre las redes muestran que el modelo ha hecho un trabajo notable al reproducir la estructura de la red planta-polinizador y que las diferencias entre las redes real y estimada son, en promedio, insignificantes. Esto sugiere que el modelo puede ser una herramienta confiable para la estimación

de redes ecológicas.

Para trabajos futuros, se recomienda:

- Exploración de Nuevas Técnicas y Modelos: Investigar otras arquitecturas de redes neuronales y métodos de procesamiento de imágenes que puedan mejorar la precisión en la detección y clasificación de polinizadores, especialmente en casos difíciles.
- Optimización de Modelos Existentes: Mejorar el entrenamiento y ajuste de modelos como EfficientNet para aumentar su eficacia en la detección de polinizadores.
- Ampliación de la Base de Datos: Incluir más variedades de polinizadores y plantas para enriquecer el conjunto de datos y permitir una generalización más robusta de los modelos.
- Aplicaciones en Conservación de la Biodiversidad: Explorar cómo estos modelos pueden aplicarse en proyectos de conservación y estudios de biodiversidad para comprender mejor las interacciones planta-polinizador.
- Generar protocolos para generar la detección de imágenes y la construcción de redes planta-polinizador a partir del trabajo de campo.

Apéndice A

Anexos

A.1. Repositorio GitHub

En el siguiente enlace, se puede acceder al repositorio de GitHub donde se puede encontrar los notebooks con el código necesario para replicar los resultados presentados en este proyecto.

<https://github.com/andres-merino/TFM-UOC-CienciaDeDatos>

De manera específica, los *notebooks* que se pueden encontrar son los siguientes:

- **EfficientNET.ipynb**: *notebook* con el código necesario para realizar el entrenamiento del modelo EfficientNet.
- **YOLOv5-entrenamiento.ipynb**: *notebook* con el código necesario para realizar el entrenamiento del modelo YOLOv5, este fue ejecutado en Google Colab.
- **YOLOv5-prediccion.ipynb**: *notebook* con el código necesario para realizar la predicción del modelo YOLOv5.
- **YOLOv5-prediccion-CrossVal.ipynb**: *notebook* con el código necesario para realizar la predicción del modelo YOLOv5 con validación cruzada.
- **EstimacionRedPolinizadores.ipynb**: *notebook* con el código necesario para realizar la estimación de la red de polinizadores.
- **RedPolinizadores.ipynb.ipynb**: *notebook* con el código necesario para realizar el análisis de la red de polinizadores.

A.2. Repositorio de pesos de modelos

En el siguiente enlace, se puede acceder al repositorio particular de OneDrive del autor donde se puede encontrar los pesos de los modelos entrenados en este proyecto.

<https://1drv.ms/f/s!AuK-ajP1UCYfoPF4K2pF0xNCGibnJg?e=RXUcK4>

De manera específica, los pesos que se pueden encontrar son los siguientes:

- `efficientnetb7_weights.h5`: pesos del modelo EfficientNet.
- `yolov5m_v01.pt`: pesos del modelo YOLOv5, en su versión `m`.
- `yolov5l_v01.pt`: pesos del modelo YOLOv5 en su versión `l`.
- `yolov5m_foldn.pt`: pesos del modelo YOLOv5 en el *fold* `n` de la validación cruzada.

Bibliografía

- [1] Alexandra-Maria Klein, Bernard Vaissière, James H. Cane, Ingolf Steffan-Dewenter, Saul A. Cunningham, Claire Kremen, and Teja Tscharntke. Importance of pollinators in changing landscapes for world crops. *Proceedings of The Royal Society B: Biological Sciences*, 274(1608):303–313, 2006.
- [2] Innovatione AgroFood Design. Polinización: tipos y polinizadores en el campo- Innovatione, 2021.
- [3] Fundación Unam. Alertan crisis por disminución de polinizadores a nivel mundial, 6 2019. <https://www.fundacionunam.org.mx/ecopuma/alertan-crisis-por-disminucion-de-polinizadores-a-nivel-mundial/>.
- [4] Bing Maps. Cabrera, Islas Baleares, España. Map, 2024. <https://www.bing.com/maps?osid=a92987d3-57e1-4b18-9d35-3a1b7f6ddc96&cp=38.975843~3.035423&lvl=11&imgid=e8d6c895-2362-4fdd-9dba-b2dc5ac7cf8e&v=2&sV=2&form=S00027>.
- [5] Pl@ntNet. Pl@ntNet API for developers. <https://my.plantnet.org/>.
- [6] Iftikhar Ahmad, Yayun Yang, Yi Yao, Hongzhi Wang, Muhammad Hassan, Xi Cheng, Yunzhi Wu, and Youhua Zhang. Deep learning based detector YOLOV5 for identifying insect pests. *Applied sciences*, 12(19):10167, 2022.
- [7] Fang Qi, Yuxiang Wang, Zhe Tang, and Shuhong Chen. Real-time and effective detection of agricultural pest using an improved YOLOv5 network. *Journal of Real-time Image Processing*, 20(2), 2023.
- [8] Joel Bharat Monis, Rishab Sarkar, Sn Nagavarun, and Jayati Bhadra. Efficient NET: identification of crop insects using convolutional neural networks. 2022 *International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*, 2022.

- [9] Jean-Gabriel Young, Fernanda S. Valdovinos, and M. E. J. Newman. Reconstruction of plant–pollinator networks from observational data. *Nature Communications*, 12(1), 2021.
- [10] David H. Hubel and Torsten N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160(1):106–154, 1962.
- [11] Kunihiko Fukushima and Sei Miyake. Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. *Pattern Recognition*, 15(6):455–469, 1982.
- [12] Yann LeCun, Bernhard E. Boser, John S. Denker, D. Henderson, Richard Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten Zip code recognition. *Neural Computation*, 1(4):541–551, 1989.
- [13] Laith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ayad Q. Al-Dujaili, Ye Duan, Omran Al-Shamma, José Santamaría, Mohammed A. Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1), 3 2021.
- [14] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Liyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, and Tsuhan Chen. Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354–377, 2018.
- [15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Siying Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Feifei Li. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 4 2015.
- [16] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [17] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.
- [18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

- [19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.
- [20] Kim Bjerge, Jamie Alison, Mads Dyrmann, Carsten Eie Frigaard, Hjalte M. R. Mann, and Toke T. Høye. Accurate detection and identification of insects from camera trap images with deep learning. *PLOS sustainability and transformation*, 2(3):e0000051, 2023.
- [21] Joseph Redmon, D. Hari Hara Santosh, Girshick Ross, and Ali Farhadi. You only look once: Unified, Real-Time Object Detection. *arXiv (Cornell University)*, 2015.
- [22] Upesh Nepal and Hossein Eslamiat. Comparing YOLOV3, YOLOV4 and YOLOV5 for autonomous landing spot detection in faulty UAVs. *Sensors*, 22(2):464, 2022.
- [23] Mingxing Tan and Quoc V. Le. EfficientNet: Rethinking model scaling for convolutional neural networks. *arXiv (Cornell University)*, 2019.
- [24] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. *arXiv (Cornell University)*, 2019.
- [25] Tashin Ahmed and Noor Hossain Nuri Sabab. Classification and understanding of cloud structures via satellite images with efficientunet. *arXiv (Cornell University)*, 2021.
- [26] Chengjun Xie, Jie Zhang, Rui Li, Jinyan Li, Peilin Hong, Junfeng Xia, and Peng Chen. Automatic classification for field crop insects via multiple-task sparse representation and multiple-kernel learning. *Computers and Electronics in Agriculture*, 119:123–132, 2015.
- [27] Wenyong Li, Tengfei Zheng, Zhankui Yang, Ming Li, Cheng Sun, and Xiting Yang. Classification and detection of insects from field images using deep learning for smart pest management: A systematic review. *Ecological Informatics*, 66:101460, 2021.
- [28] Zhankui Yang, Xiting Yang, Ming Li, and Wenyong Li. Small-sample learning with salient-region detection and center neighbor loss for insect recognition in real-world complex scenarios. *Computers and Electronics in Agriculture*, 185:106122, 6 2021.

- [29] Dan Jeric Arcega Rustia, Chen-Yi Lu, Jun-Jee Chao, Ya-Fang Wu, Jui-Yung Chung, Ju-Chun Hsu, and Ta-Te Lin. Online semi-supervised learning applied to an automated insect pest monitoring system. *Biosystems Engineering*, 208:28–44, 8 2021.
- [30] V. Malathi and M. P. Gopinath. Classification of pest detection in paddy crop based on transfer learning approach. *Acta Agriculturae Scandinavica Section B-soil and Plant Science*, 71(7):552–559, 2 2021.
- [31] Morteza Khanramaki, Ezzatollah Askari Asli-Ardeh, and Ehsan Kozegar. Citrus pests classification using an ensemble of deep learning models. *Computers and Electronics in Agriculture*, 186:106192, 7 2021.
- [32] V.K. Sagar, R. Gengaje Sachin, K. Chandrashekara, and K. N. Ganeshiah. Identification of Indian Butterflies and Moths with Deep Convolutional Neural Networks x. *Current Science*, 118(9):1456, 5 2020.
- [33] Yanfen Li, Hanxiang Wang, L. Minh Dang, Abolghasem Sadeghi-Niaraki, and Hyeyoon Moon. Crop pest recognition in natural scenes using convolutional neural networks. *Computers and Electronics in Agriculture*, 169:105174, 2 2020.
- [34] Xi Cheng, Youhua Zhang, Yiqiong Chen, Yunzhi Wu, and Yi Yue. Pest identification via deep residual learning in complex background. *Computers and Electronics in Agriculture*, 141:351–356, 9 2017.
- [35] Ziyi Liu, Junfeng Gao, Guoguo Yang, Huan Zhang, and Yong He. Localization and Classification of Paddy Field Pests using a Saliency Map and Deep Convolutional Neural Network. *Scientific Reports*, 6(1), 2 2016.
- [36] Yue He, Zhiyan Zhou, Luhong Tian, Youfu Liu, and Xiwen Luo. Brown rice planthopper (*Nilaparvata lugens* Stål) detection based on deep learning. *Precision Agriculture*, 21(6):1385–1402, 2020.
- [37] D. Tzutalin. LabelImg (GitHub), 2015. <https://github.com/HumanSignal/labelImg>.
- [38] Pedro Jordano, Jordi Bascompte, and Jens M. Olesen. Invariant properties in coevolutionary networks of plant-animal interactions. *Ecology Letters*, 6(1):69–81, 2002.
- [39] Pau Serra. Automated pollination monitoring: a method that provides complementary information in community-level studies, 2022.