

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

**Aplicación de aprendizaje profundo para la detección y
clasificación automática de insectos agrícolas en trampas
pegantes**

Tesis para obtener el Título Profesional de Ingeniera Informática:

Autora:

Claudia Sofía Córdova Pérez

Asesor:

Dr. Edwin Rafael Villanueva Talavera

Lima, mayo de 2021

Resumen

La horticultura es una actividad que da trabajo a muchos peruanos en distintas zonas del país, sin embargo, gran parte de la producción de hortalizas es dañada por la alta incidencia de plagas de insectos. En la actualidad, un método efectivo para realizar el control de estas plagas es el uso de trampas pegantes, las cuales atraen y atrapan distintos tipos de insectos. Convencionalmente, las trampas pegantes son colocadas de forma que queden distanciadas uniformemente en el campo donde se realiza el cultivo y luego de varios días se realizan observaciones visuales por parte del personal entrenado en reconocimiento de insectos. No obstante, la información recopilada manualmente por el humano puede no ser tan exacta, pues existen diversos factores que pueden influir en la precisión de esta, por ejemplo, la habilidad de cada persona para detectar distintos tipos de insectos y la posible fatiga que puede ser consecuencia de haber realizado un trabajo manual por mucho tiempo y para una muestra grande de insectos.

Las soluciones que se encontraron en la revisión sistemática para tratar problemas de detección de insectos fueron algoritmos de segmentación con cambio de espacio de color, lo cual permite remover el fondo de una imagen y centrarse únicamente en el objeto de interés; también, se encontraron estudios que usaron modelos de detección, los cuales hacen uso de aprendizaje profundo con redes neuronales convolucionales para lograr la identificación de los insectos. Esta última técnica ha dado resultados óptimos en distintos problemas visión computacional, por lo que el presente proyecto de investigación propone usar los modelos de detección pre-entrenados Faster R-CNN y YOLOv4 y aplicarles aprendizaje por transferencia para ajustarlos al problema de detección de tres tipos de plagas de insectos: la mosca blanca, la mosca minadora y el pulgón verde del melocotonero en etapa de adulto alado.

Para ello, se debe contar con un corpus de imágenes de trampas pegantes con insectos plaga y, debido a la limitada disponibilidad de estas, se planteó construir un generador de imágenes realistas de trampas pegantes con insectos, el cual tiene en consideración factores realistas como la iluminación y el nivel de ruido en las imágenes, además, se usaron técnicas de segmentación y aumento de imágenes de modo que el corpus obtenido sea el adecuado para la fase de entrenamiento.

Finalmente, se midió la métrica mAP de ambos modelos para los tres tipos de insectos. El modelo Faster R-CNN obtuvo 94.06% y el modelo YOLOv4, 95.82%, donde se concluye que el desempeño de ambos detectores es aceptable.

Tabla de Contenido

Resumen.....	2
Índice de Figuras	5
Índice de Tablas.....	6
Capítulo 1. Generalidades.....	8
1.1 Problemática	8
1.1.1 Árbol de Problemas	8
1.1.2 Descripción	8
1.1.3 Problema seleccionado	10
1.2 Objetivos.....	10
1.2.1 Objetivo general.....	10
1.2.2 Objetivos específicos	10
1.2.3 Resultados esperados	10
1.2.4 Mapeo de objetivos, resultados y verificación.....	11
1.3 Métodos y Procedimientos.....	14
Capítulo 2. Marco Legal/Regulatorio/Conceptual/otros.....	23
2.1 Introducción.....	23
2.2 Desarrollo del marco	23
Capítulo 3. Estado del Arte	32
3.1 Introducción.....	32
3.2 Objetivos de revisión.....	32
3.3 Preguntas de revisión	32
3.4 Estrategia de búsqueda	32
3.4.1 Motores de búsqueda a usar.....	32
3.4.2 Cadenas de búsqueda a usar	32
3.4.3 Documentos encontrados.....	33
3.4.4 Criterios de inclusión/exclusión	33
3.5 Formulario de extracción de datos	33

3.6	Resultados de la revisión	35
3.6.1	Respuesta a pregunta: ¿Qué tipos de algoritmos se presentan para la segmentación de imágenes de insectos o relacionados y cuáles son sus métricas y resultados?	35
3.6.2	Respuesta a pregunta: ¿Qué arquitecturas de Deep Learning se vienen usando para la clasificación de insectos agrícolas?.....	37
3.6.3	Respuesta a pregunta: ¿Qué métricas son las usadas en los modelos de clasificación que usan Deep Learning para reconocimiento de imágenes de insectos y cuáles son sus resultados?	38
3.7	Conclusiones.....	39
Capítulo 4.	Construir un generador de imágenes realistas de trampas pegantes con insectos plaga de hortalizas	40
4.1	Introducción	40
4.2	Resultados alcanzados.....	40
4.3	Discusión.....	44
Capítulo 5.	Construir un modelo algorítmico para la identificación y clasificación de insectos plaga de hortalizas en imágenes de trampas pegantes.....	45
5.1	Introducción	45
5.2	Resultados alcanzados.....	45
5.3	Discusión.....	53
Capítulo 6.	Crear una herramienta que permita visualizar la identificación, clasificación, conteo y generación de reportes de poblaciones de plagas de insectos de relevancia en plantíos de hortalizas en Perú a partir del ingreso de fotografías de trampas pegantes.....	54
6.1	Introducción	54
6.2	Resultados alcanzados.....	54
6.3	Discusión.....	58
Capítulo 7.	Conclusiones y trabajos futuros	59
7.1	Conclusiones.....	59
7.2	Trabajos futuros	59
Referencias.....		61
Anexos		1

Índice de Figuras

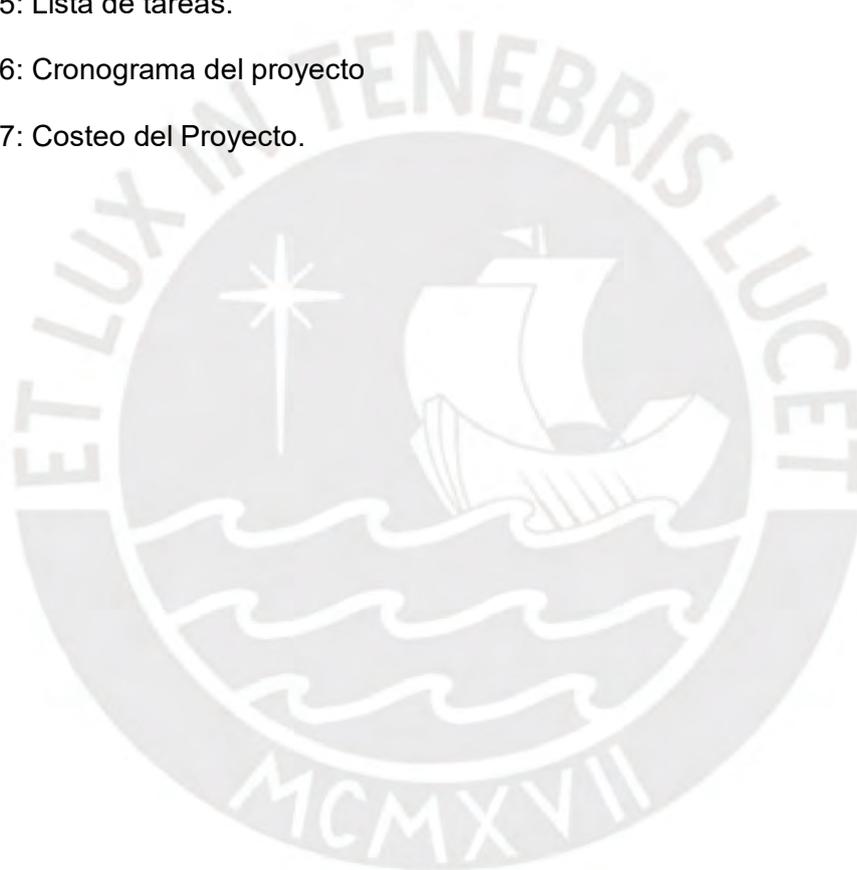
Figura 1: Árbol de problemas	8
Figura 2: Estructura del detector de objetos Faster R-CNN. Adaptado de Ren et al. (2016).	16
Figura 3: Estructura del detector de objetos YOLOv4. Adaptado de Bochkovskiy et al. (2020).	17
Figura 4: Forma adulta del pulgón verde del melocotonero o "Myzus persicae". Adaptado de Cañedo et al. (2011).	23
Figura 5: Daño directo del pulgón de la coliflor. Adaptado de Cañedo et al. (2011).	25
Figura 6: Daño directo de mosca minadora en apio. Adaptado de Cañedo et al. (2011).	25
Figura 7: Aplicación de insecticidas químicos sin protección adecuada. Adaptado de Cañedo et al. (2011).	26
Figura 8: Trampa pegante amarilla. Adaptado de (SENASA, 2006).	27
Figura 9: Lámpara de luz blanca fluorescente. Adaptado de Camborda et al. (2015).	27
Figura 10: Predador Hippodamia para el pulgón. Adaptado de Cañedo et al. (2011).	28
Figura 11: Identificación de la región de interés de la imagen. Adaptado de (ICAR-NBAIR, 2013).	29
Figura 12: Tipos de insectos con etiquetas de nombre. Adaptado de (ICAR-NBAIR, 2013).	30
Figura 13: CNN para clasificar una imagen. Adaptado de LeCun & Bengio (1998).	31
Figura 14: Distribución de algoritmos usados para segmentación de imagen.	36
Figura 15: Distribución de las arquitecturas de los modelos de Deep Learning	38
Figura 16: Muestra de imágenes recolectadas de la mosca blanca.	42
Figura 17: Muestra de imágenes recolectadas de la mosca minadora.	42
Figura 18: Imágenes recolectadas del pulgón verde del melocotonero.	42
Figura 19: Segmentación de una imagen de la mosca blanca.	43
Figura 20: Segmentación de una imagen de la mosca minadora.	43

Figura 21: Ejemplo de imagen de trampa pegante generada.	46
Figura 22: Ejemplo de las detecciones hechas por Faster R-CNN.	47
Figura 23: Average Precision por tipo de insecto y mAP del detector Faster R-CNN.	48
Figura 24: Ejemplo de las detecciones hechas por YOLOv4.	49
Figura 25: Average Precision por tipo de insecto y mAP del detector YOLOv4.	50
Figura 26: Gráficas de AP por clase del modelo Faster R-CNN.	51
Figura 27: Gráficas de AP por clase del modelo YOLOv4.	51
Figura 28: Ejemplos de FP de M1.	53
Figura 29: Pantalla de inicio de la herramienta web con imágenes subidas de trampas pegantes amarillas.	56
Figura 30: Identificación de los insectos en las imágenes subidas.	57
Figura 31: Reporte de resultados de la detección de insectos en las imágenes de trampas pegantes amarillas.	57
Figura 32: Diagrama de estructura de descomposición del trabajo (EDT).	7

Índice de Tablas

Tabla 1: Mapeo de objetivos, resultados e indicador objetivamente verificable.	11
Tabla 2: Matriz de contingencia para la clasificación de dos clases.	18
Tabla 3: Cuadro de objetivos y herramientas a usar.	20
Tabla 4: Principales plagas que originan daños económicos en el Perú. Adaptado de Cañedo et al. (2011).	24
Tabla 5: Documentos encontrados por cadena y motor de búsqueda	33
Tabla 6: Campos del formulario de extracción de datos para P1.	34
Tabla 7: Campos del formulario de extracción de datos para P2 y P3.	35
Tabla 8: Métricas y resultados de los algoritmos de segmentación.	36
Tabla 9: Métricas y resultados de los modelos de clasificación de insectos.	38

Tabla 10: Listado de requerimientos funcionales del generador de imágenes realistas de trampas pegantes.	40
Tabla 11: mAP de los modelos de detección.	52
Tabla 12: Modelos YOLO entrenados a comparar.	52
Tabla 13: Listado de requerimientos funcionales de la herramienta web.	54
Tabla 14: Mapeo de riesgos del proyecto.	6
Tabla 15: Lista de tareas.	7
Tabla 16: Cronograma del proyecto	8
Tabla 17: Costeo del Proyecto.	11



Capítulo 1. Generalidades

1.1 Problemática

1.1.1 Árbol de Problemas

		1	2	3
ARBOL DE PROBLEMAS	PROBLEMAS EFECTOS	Demora en la identificación manual de los insectos, cansancio y fatiga del personal que hace la identificación, alta demanda de horas-hombre, retrasos en la vigilancia y control y pérdidas económicas		
	PROBLEMA CENTRAL	No existe una herramienta que permita automatizar el reconocimiento de poblaciones de insectos plaga de relevancia en plantíos de hortalizas en Perú a partir de la ingesta de fotografías de trampas pegantes		
	PROBLEMAS CAUSAS	No hay disponible un dataset de imágenes de trampas pegantes con insectos plaga de relevancia en plantíos de hortalizas en Perú	No hay disponible un modelo algorítmico entrenado para detectar y clasificar insectos plaga de relevancia en plantíos de hortalizas en Perú a partir de imágenes de trampas pegantes	Falta de una herramienta que permita automatizar la identificación, clasificación, conteo y generación de reportes de poblaciones de insectos plaga de relevancia en plantíos de hortalizas en Perú a partir de la ingesta de fotografías de trampas pegantes

Figura 1: Árbol de problemas

1.1.2 Descripción

El Perú es considerado un país mega-biodiverso y presenta una diversidad de pisos ecológicos que constituyen la base de su gran variedad agrícola productiva (Escartín et al., 2012). La intensa demanda de hortalizas y la posibilidad de producir algunas todo el año muestran la importancia social de estos cultivos que constituyen una fuente de trabajo y de crecimiento económico en diversas zonas del país, teniendo Lima, Arequipa, Junín, La Libertad e Ica como las más importantes regiones productoras (Escartín et al., 2012).

Pese a que la horticultura ha crecido en las últimas décadas, y a pesar de los múltiples esfuerzos de innovación e inserción de tecnología, su rentabilidad no es muy alta debido a una diversidad de problemas a los que se enfrenta incluyendo su vulnerabilidad frente a la alta incidencia de plagas y enfermedades que las compromete (INIA, 2009).

Dentro de este contexto, es razonable considerar métodos de control y manejo de plagas que aseguren la productividad de los cultivos. Existen distintos métodos de control, entre

ellos tenemos el control etológico, en el cual se encuentran las trampas pegantes (Cañedo et al., 2011). Bravo-Portocarrero et al. (2020) reportó que las trampas pegantes son una excelente opción para la vigilancia de cultivos de hortalizas. Las de mayor eficiencia en hortalizas de hoja (repollo, acelga, lechuga y cilantro) para la captura de trips son las trampas pegantes celeste y blanco, mientras que la trampa pegante amarilla funciona mejor para la captura de pulgones y cigarritas. Otro método es el control químico, conformado principalmente por el uso de plaguicidas. Este método de control tiene una alta efectividad; sin embargo, el mal uso y abuso de estos productos, trae problemas a la salud de los productores, consumidores, animales y al medioambiente (Cañedo et al., 2011). Si comparamos estos dos métodos de control de plagas, el uso de trampas pegantes es una alternativa barata y amigable con el medio ambiente, a diferencia del control químico.

Con el fin de lograr un mejor control de plagas y el uso mínimo de pesticidas, se debe tener una adecuada estimación de las densidades de poblaciones de insectos plaga en el campo y así poder diseñar estrategias de manejo de plagas más adecuadas (Cho et al., 2007). Convencionalmente, para la identificación de los insectos plaga se colocan trampas pegantes y se realizan observaciones visuales por parte del personal entrenado en reconocimiento de insectos. Si se observa la presencia de insectos sospechosos, estos se colectan con un aspirador manual y se colocan en frascos viales con alcohol etílico de 70° (SENASA, 2013).

Sin embargo, la información obtenida por el juicio visual puede no ser tan exacta, pues depende de la habilidad de la persona para identificar los insectos, la posible fatiga en el muestreo continuo y el hecho de que esto requiere de mucho tiempo para contar una muestra grande de insectos (Cho et al., 2007).

Las técnicas actuales de Inteligencia Artificial han demostrado su idoneidad en diferentes problemas de visión computacional. Sin embargo, a pesar de dichos desarrollos, aún no se ha descrito en la literatura modelos algorítmicos para identificar y clasificar insectos plaga de relevancia en plantíos de hortalizas en Perú a partir de imágenes de trampas pegantes. Por consecuencia, no hay tampoco una herramienta que permita automatizar la identificación, clasificación, conteo y generación de reportes de poblaciones de estos insectos plaga.

1.1.3 Problema seleccionado

Por lo descrito previamente, el problema central elegido para el presente proyecto es la carencia de una herramienta que permita automatizar el reconocimiento de poblaciones de insectos plaga de relevancia en plantíos de hortalizas en Perú a partir del ingreso de fotografías de trampas pegantes.

1.2 Objetivos

1.2.1 Objetivo general

Construir una herramienta que permita la identificación y clasificación de los insectos plaga que generan pérdidas económicas en el Perú a través del ingreso de imágenes de trampas pegantes.

1.2.2 Objetivos específicos

- O 1. Construir un generador de imágenes realistas de trampas pegantes con insectos plaga de hortalizas.
- O 2. Construir un modelo algorítmico para la identificación y clasificación de insectos plaga de hortalizas en imágenes de trampas pegantes.
- O 3. Crear una herramienta que permita visualizar la identificación, clasificación, conteo y generación de reportes de poblaciones de plagas de insectos de relevancia en plantíos de hortalizas en Perú a partir del ingreso de fotografías de trampas pegantes.

1.2.3 Resultados esperados

- O 1. Construir un generador de imágenes realistas de trampas pegantes con insectos plaga de hortalizas.
 - R 1. Documento de requerimientos funcionales del generador de imágenes realistas.
 - R 2. Corpus de imágenes base de insectos plaga de hortalizas de relevancia en Perú.
 - R 3. Generador de imágenes de trampas pegantes con insectos plaga de hortalizas.

- O 2. Construir un modelo algorítmico para la identificación y clasificación de insectos plaga de hortalizas en imágenes de trampas.
 - R 4. Corpus de imágenes generadas de trampas pegantes con insectos identificados y etiquetados para el entrenamiento y evaluación.
 - R 5. Modelos algorítmicos para identificar y clasificar insectos plaga a base de imágenes de trampas pegantes (R4).
 - R 6. Reporte de comparación de los modelos de detección de insectos.
- O 3. Crear una herramienta que permita visualizar la identificación, clasificación, conteo y generación de reportes de poblaciones de plagas de insectos de relevancia en plantíos de hortalizas en Perú a partir del ingreso de fotografías de trampas pegantes.
 - R 7. Documento de requerimientos funcionales para la herramienta web.
 - R 8. Interfaz web que integre los modelos algorítmicos de identificación y clasificación de insectos plaga y genere reportes sobre poblaciones de insectos plaga de relevancia en plantíos de hortalizas en el Perú.

1.2.4 Mapeo de objetivos, resultados y verificación

Tabla 1: Mapeo de objetivos, resultados e indicador objetivamente verificable.

Objetivo: Construir un generador de imágenes realistas de trampas pegantes con insectos plaga de hortalizas.		
Resultado	Medio de verificación	Indicador objetivamente verificable
R1. Documento de requerimientos funcionales del generador de imágenes realistas.	- Documento con la lista de requerimientos funcionales del generador de imágenes realistas.	- Documento de requerimientos aprobado por un experto.

R2. Corpus de imágenes base de insectos plaga de hortalizas de relevancia en Perú.	- Carpetas con las imágenes organizadas por tipo de insecto.	- Imágenes base aprobadas por un experto. - La cantidad de imágenes por tipo de insecto es mayor o igual a 10.
R3. Generador de imágenes realistas de trampas pegantes con insectos plaga de hortalizas.	- Código fuente.	- Se cumple con 8 requerimientos de R1.
Objetivo: Construir un modelo algorítmico para la identificación y clasificación de insectos plaga de hortalizas en imágenes de trampas.		
Resultado	Medio de verificación	Indicador objetivamente verificable
R4. Corpus de imágenes generadas de trampas pegantes con insectos identificados y etiquetados para el entrenamiento y evaluación.	- Carpeta con las imágenes generadas.	- La cantidad de imágenes generadas es mayor o igual a 500.
R5. Modelos algorítmicos para identificar y clasificar insectos plaga a base de imágenes de	- Reporte de cómo fueron creados los modelos y cómo son sus estructuras.	- mAP de los modelos de detección mayor a 50%.

trampas pegantes (R4).		
R6. Reporte de comparación de los modelos de detección de insectos.	- Documento con la comparación de los modelos.	- Reporte de comparación aprobado por un experto.
<p>Objetivo: Crear una herramienta que permita visualizar la identificación, clasificación, conteo y generación de reportes de poblaciones de plagas de insectos de relevancia en plantíos de hortalizas en Perú a partir del ingreso de fotografías de trampas pegantes.</p>		
Resultado	Medio de verificación	Indicador objetivamente verificable
R7. Documento de requerimientos funcionales para la herramienta web.	- Documento con la lista de requerimientos funcionales del sistema web.	- Documento de requerimientos aprobado por un experto.
R8. Interfaz web que integre los modelos algorítmicos de identificación y clasificación de insectos plaga y genere reportes sobre poblaciones de insectos plaga de relevancia en plantíos de hortalizas en el Perú.	- Software y código fuente. - Plataforma web que consuma el API y permita interacción.	- Pruebas unitarias aprobadas al 100% - Pruebas funcionales aprobadas al 100%

1.3 Métodos y Procedimientos

1.3.1 Herramientas

Python

Es un lenguaje de programación orientado a objetos de alto nivel con semántica dinámica. Su intérprete y extensa biblioteca estándar fomentan la modularidad de los programas y la reutilización de código. (*What is Python?*, 2020).

Keras

Es un API que sigue las mejores prácticas para reducir la carga cognitiva, pues minimiza el número de acciones requeridas para casos de uso comunes, a la par que proporciona comentarios claros para posibles errores del usuario. Keras fue clasificado como el framework #1 para aprendizaje profundo (Team, 2015).

TensorFlow

Es una plataforma de código abierto para el aprendizaje automático. TensorFlow ofrece varios niveles de abstracción, cuenta con un ecosistema integral y flexible de herramientas y bibliotecas que permite implementar con facilidad aplicaciones de aprendizaje automático, usando la API de alto nivel de Keras. (*TensorFlow*, 2015).

Scikit-learn

Es una simple y eficiente herramienta para análisis de datos accesible para todos y de código abierto para proyectos en Python (Pedregosa et al., 2011).

LabelImg

Es una herramienta gráfica de anotación de imágenes y etiquetado de regiones de interés en imágenes (Lin, 2019).

OpenCV

Es una librería para software que contiene más de 2500 algoritmos optimizados, en los que se incluyen algoritmos de visión artificial y aprendizaje automático. Estos algoritmos se pueden usar para detectar y reconocer caras, identificar objetos, clasificar acciones humanas en videos, rastrear movimientos en cámara, etc. (Intel, 2020).

Flask

Es un micro marco de trabajo (micro framework) para Python que tiene como objetivo crear aplicaciones simples pero extensibles (Ronacher, 2010).

React Js

Es una biblioteca de JavaScript que permite crear interfaces de usuario interactivas de forma sencilla (Walke, 2013).

Unittest

Es un marco de trabajo utilizado para las pruebas unitarias inspirado en JUnit (*unittest — Unit testing framework — Python 3.9.0 documentation*, 2020).

Selenium

Es un conjunto de herramientas usado para pruebas y para automatizar navegadores web (*About Selenium*, 2013).

Faster R-CNN

Es un detector de objetos el cual integra Fast R-CNN con el algoritmo de propuesta de regiones (RPN), el cual le dice a la red dónde buscar, pues el componente RPN predice los límites de objetos y las puntuaciones de objetividad en cada posición. Esto permite compartir las características convolucionales de la imagen con la red de detección (Ren et al., 2016).

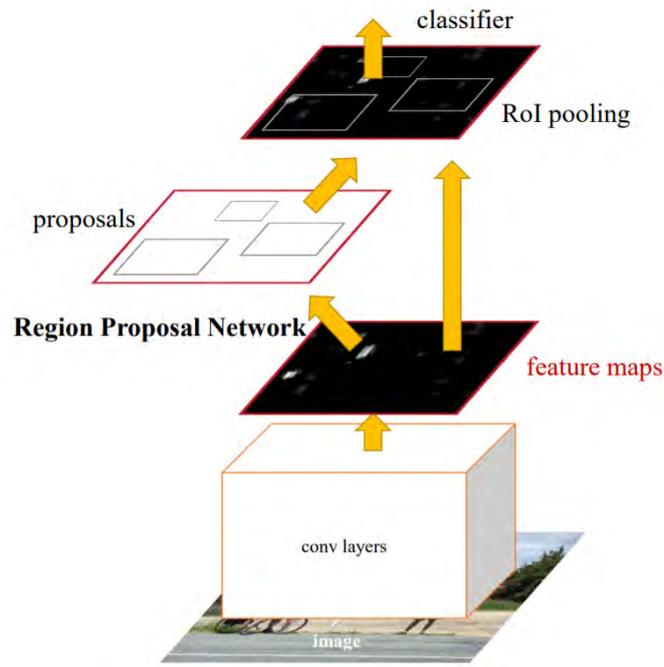


Figura 2: Estructura del detector de objetos Faster R-CNN. Adaptado de Ren et al. (2016).

YOLOv4

Es un detector de objetos el cual está compuesto por un backbone, que es la red neuronal pre-entrenada de imágenes que se usa para extraer las características principales de la imagen; un neck, que es una capa extra que se usa para extraer diferentes mapas de características de diferentes capas del backbone y, por último, un head que es una red encargada de la parte de detección (clasificación o regresión) de las regiones de interés (bounding boxes). Además, cuenta con un BoF (bag of freebies) que son métodos para obtener mejores resultados sin aumentar el costo de inferencia, un ejemplo de esto son las técnicas de data augmentation. También, se tiene un BoS (bag of specials) que son complementos como métodos de post-procesamiento que aumentan el costo de inferencia en una pequeña proporción, pero que pueden incrementar la exactitud del detector de objetos (Bochkovskiy et al., 2020).

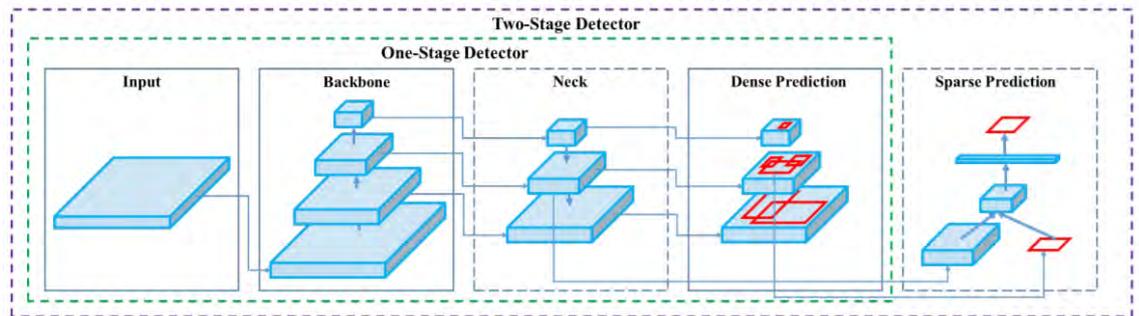


Figura 3: Estructura del detector de objetos YOLOv4. Adaptado de Bochkovskiy et al. (2020).

1.3.2 Métodos

Data augmentation

Son métodos que se usan para aprovechar al máximo los pocos ejemplos de datos de prueba y aumentar la precisión de modelos algorítmicos. Estas técnicas pueden ser escalar, rotar, reflejar horizontal o verticalmente, desplazar horizontal o verticalmente, hacer zoom, entre otros. Se espera que, con estas técnicas, el aumento de datos ayude a prevenir el sobreajuste (problema común cuando se tiene un conjunto de datos pequeño) y así el modelo mejore su capacidad para generalizar (Romero-Lopez et al., 2017).

Transferencia de conocimiento en modelos CNN

También conocido en inglés como transfer learning. Es una técnica que se emplea en modelos CNN pre-entrenados. Estos fueron entrenados con data masiva y lo que se quiere es ajustar las características aprendidas en los últimos niveles y añadir data objetivo para obtener el resultado deseado (Shin et al., 2016).

K-fold cross validation

Es una técnica que divide la muestra en K grupos, llamados pliegues, de igual tamaño (si fuera posible). Cada pliegue está constituido por distintos conjuntos de entrenamientos y conjuntos de pruebas. La idea es entrenar los K pliegues y analizar las métricas de estos modelos (Scikit-learn, 2011).

Métricas de evaluación:

Olson & Denle (2008) establece TP como los que fueron predichos como la clase deseada y eran verdaderamente de la clase deseada; FP, como los que fueron

predichos como la clase deseada, pero en realidad eran de otra clase; FN, como los que predijo que eran de otras clases, pero en realidad eran de la clase deseada y TN, como los que predijo como otras clases y, efectivamente, eran de otras clases. En la tabla 2, establece la matriz de contingencia para un problema de dos clases.

Tabla 2: Matriz de contingencia para la clasificación de dos clases.

		Clase verdadera	
		Positivo	Negativo
Clase predecida	Positivo	True Positive (TP)	False Positive (FP)
	Negativo	False Negative (FN)	True Negative (TN)

Precision

Es la proporción entre TP y la suma de TP con FP de una clase (Olson & Delen, 2008).

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

Accuracy

Es la proporción de la suma de TP y TN entre la suma de TP, TN, FP y FN de una clase (Olson & Delen, 2008).

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

Recall

Es la proporción entre TP y la suma de TP y FN de una clase (Olson & Delen, 2008).

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

IoU

Intersección sobre unión o Intersection over Union es una métrica popular usada para la segmentación de imágenes. Dado un objetivo y una predicción, la métrica se define como la proporción entre el objetivo intersectado con la predicción y el objetivo unión con la predicción (Nowozin, 2014).

$$IoU = \frac{\text{objetivo} \cap \text{predicción}}{\text{objetivo} \cup \text{predicción}}$$

AP

Average Precision es una popular medida que toma en consideración el precision y el recall al momento de comparar distintos algoritmos de detección. Se puede expresar el precision como una función del recall por $p(r)$ (M. Zhu, 2004). El average precision se puede calcular en un intervalo de $r=0$ a $r=1$ de la siguiente forma:

$$AP = \int_0^1 p(r) dr$$

Un método para estimar el valor del average precision es usando 11 Point Interpolation (Manning et al., 2008), el cual se calcula de la siguiente manera:

$$AP = \frac{1}{11} \sum_{r \in \{0.0, 0.1, 0.2, \dots, 1.0\}} AP_r$$

$$\text{donde } AP_r = \max_{\tilde{r} \geq r} p(\tilde{r})$$

mAP

Mean Average Precision es el promedio de AP para cada consulta o clase Q (Manning et al., 2008).

$$mAP(Q) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} p(r_{jk})$$

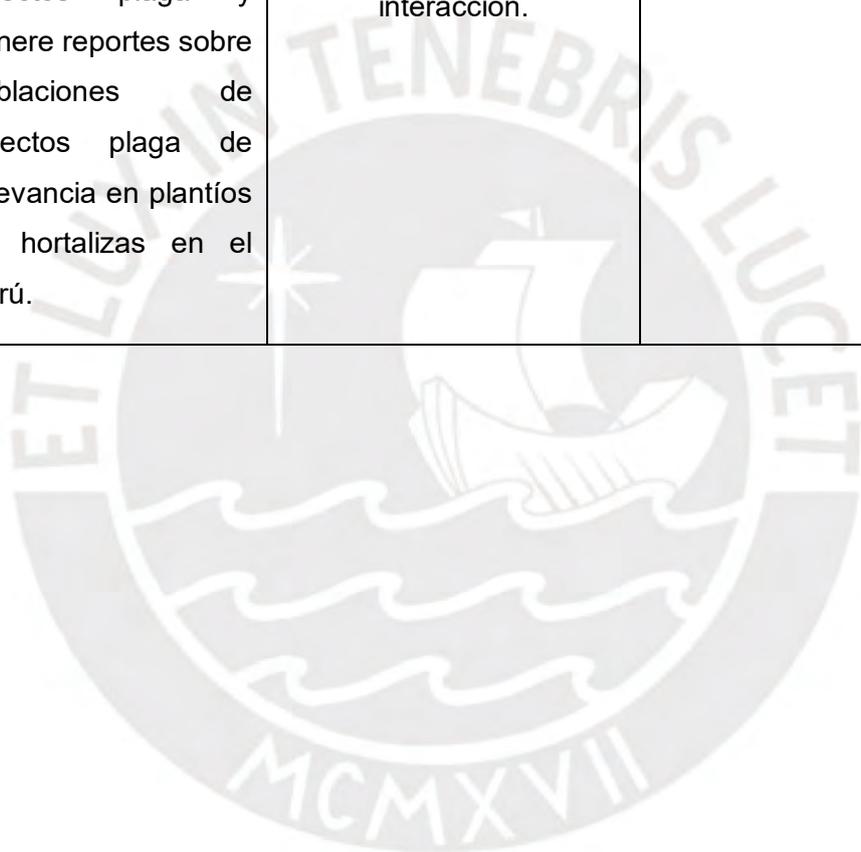
1.3.3 Cuadro de herramientas y métodos

Tabla 3: Cuadro de objetivos y herramientas a usar.

Objetivo: Construir un generador de imágenes realistas de trampas pegantes con insectos plaga de hortalizas.		
Resultado	Medio de verificación	Herramientas y métodos
R1. Documento de requerimientos funcionales del generador de imágenes realistas	- Documento con la lista de requerimientos funcionales del generador de imágenes realistas.	- Editor de texto.
R2. Corpus de imágenes base de insectos plaga de hortalizas de relevancia en Perú.	- Carpetas con las imágenes organizadas por tipo de insecto.	- Python. - Keras. - TensorFlow. - OpenCV. - Data augmentation.
R3. Generador de imágenes realistas de trampas pegantes con insectos plaga de hortalizas.	- Código fuente.	- Python. - Keras. - TensorFlow. - OpenCV.
Objetivo: Construir un modelo algorítmico para la identificación y clasificación de insectos plaga de hortalizas en imágenes de trampas.		
Resultado	Medio de verificación	Herramientas y métodos

R4. Corpus de imágenes generadas de trampas pegantes con insectos identificados y etiquetados para el entrenamiento y evaluación.	- Carpetas con las imágenes organizadas por tipo de insecto.	- Generador de imágenes realistas (R3).
R5. Modelos algorítmicos para identificar y clasificar insectos plaga a base de imágenes de trampas pegantes (R4).	- Reporte de cómo fueron creados los modelos y cómo son sus estructuras.	- Python. - Keras. - TensorFlow. - Faster R-CNN. - YOLOv4. - Transferencia de conocimiento.
R6. Reporte de comparación de los modelos de detección de insectos.	- Documento con la comparación de los modelos.	- Python. - Scikit-learn. - Métrica de evaluación mAP.
Objetivo: Crear una herramienta que permita visualizar la identificación, clasificación, conteo y generación de reportes de poblaciones de plagas de insectos de relevancia en plantíos de hortalizas en Perú a partir del ingreso de fotografías de trampas pegantes.		
Resultado	Medio de verificación	Herramientas y métodos
R7. Documento de requerimientos funcionales para la herramienta web.	- Documento con la lista de requerimientos	- Editor de texto.

	funcionales del sistema web.	
R8. Interfaz web que integre los modelos algorítmicos de identificación y clasificación de insectos plaga y genere reportes sobre poblaciones de insectos plaga de relevancia en plantíos de hortalizas en el Perú.	<ul style="list-style-type: none"> - Software y código fuente. - Plataforma web que consuma el API y permita interacción. 	<ul style="list-style-type: none"> - Flask. - React Js. - Unittest. - Selenium.



Capítulo 2. Marco Legal/Regulatorio/Conceptual/otros

2.1 Introducción

El objetivo del marco conceptual es presentar definiciones y conceptos relevantes que serán de ayuda para entender la problemática de las plagas de insectos en las plantaciones agrícolas y los métodos de detección, con especial énfasis en conceptos relacionados a cómo identificar y clasificar insectos en imágenes de trampas pegantes. Es necesario entonces, explicar los conceptos de aprendizaje automático en imágenes, como también los conceptos en el área de aplicación que es entomología agrícola y métodos de control de plagas.

2.2 Desarrollo del marco

2.2.1 Conceptos de entomología agrícola

2.2.1.1 Principales plagas de hortalizas en el Perú

En la producción de hortalizas se encuentran problemas fitosanitarios, los cuales implican el uso intensivo de plaguicidas. Estas plagas generalmente varían de acuerdo al tipo de hortaliza, zona de producción y clima (Cañedo et al., 2011). En la figura 4 se puede observar la forma adulta del pulgón verde del melocotonero como ejemplo de una de las principales plagas de hortalizas en el Perú.



Figura 4: Forma adulta del pulgón verde del melocotonero o “*Myzus persicae*”. Adaptado de Cañedo et al. (2011).

En la tabla 4 se muestran las principales plagas de hortalizas que generan grandes pérdidas económicas en el Perú, los cultivos que ataca y el color al que estos son atraídos.

Tabla 4: Principales plagas que originan daños económicos en el Perú. Adaptado de Cañedo et al. (2011).

Insecto plaga	Cultivos que ataca	Color atrayente
Mosquilla de los brotes/negrita del tomate (<i>Prodiplosis longifila</i>)	Espárrago, tomate y pimiento	No determinado azul/ amarillo
Mosca minadora (<i>Liriomyza huidobrensis</i>)	Guisantes, papa, frijoles, lechuga, apio, espinacas, brócoli y cebolla	Amarillo
Mosca Blanca (<i>Bemisia tabaci</i>)	Hortalizas en general	Amarillo
Pulgón de la col (<i>Brevicoryne brassicae</i>)	Coliflor, col y brócoli	Amarillo
Pulgón verde del melocotonero (<i>Myzus persicae</i>)	Coliflor, col y brócoli	Amarillo
Pulgón verde de las solanáceas (<i>Macrosiphum euphorbiae</i>)	Lechuga, zanahoria, apio y betarraga	Amarillo
Trips de hortalizas (<i>Trips tabaci</i>)	Hortalizas en general	Celeste

2.2.1.2 Efectos en las plantaciones

Existen diversos efectos que estas plagas pueden causar en la producción agrícola. Los pulgones generalmente hacen que las plantas atacadas se tornen amarillas o cloróticas, arrugadas o encrespadas y esto origina la muerte de las partes dañadas (Cañedo et al., 2011). Las larvas de la mosca minadora minan las hojas y tallos de ciertos vegetales, produciendo el marchitamiento de estos. La mosca minadora, junto con la mosca blanca y los pulgones, transmiten el patógeno fumagina (hongo negro) (Cañedo et al., 2011). En figura 5 se puede ver el daño causado por el pulgón de la col a la coliflor y en la figura 6, el daño al apio por la mosca minadora.



Figura 5: Daño directo del pulgón de la coliflor. Adaptado de Cañedo et al. (2011).



Figura 6: Daño directo de mosca minadora en apio. Adaptado de Cañedo et al. (2011).

2.2.1.3 Métodos de control de plagas

2.2.1.3.1 Control químico

Se hace uso de plaguicidas para matar o controlar las poblaciones de plagas. Estas pueden ser insecticidas, acaricidas, herbicidas, fungicidas, entre otros. Mayormente son sustancias de composición química sintética y son tóxicas. Si bien su uso ha resultado muy beneficioso en muchos casos, también ha resultado perjudicial por diversas razones, entre ellas se puede mencionar que estas sustancias son altamente tóxicas para los seres humanos y los animales, además, causan daños al ambiente, contaminación de aguas subterráneas y empobrecimiento de suelos. Todas estas consecuencias se deben tomar en consideración al momento de elegir un plaguicida, puesto que no se debe limitar

en el nivel de efectividad, sino en los posibles daños colaterales que puede ocasionar (Cañedo et al., 2011). En la figura 7 se puede ver la aplicación de insecticidas químicos sin uso de protección adecuada.



Figura 7: Aplicación de insecticidas químicos sin protección adecuada. Adaptado de Cañedo et al. (2011).

2.2.1.3.2 Control etológico

La etología se refiere al estudio del comportamiento de los animales, en este caso insectos, con el medio ambiente. En ese sentido, el control etológico es el control de plagas aprovechando los estímulos que se relacionan al comportamiento de los insectos y que sirve como atrayentes para estos. Por ejemplo, los atrayentes cromáticos, pues algunas especies de insectos se sienten atraídos por ciertos colores (Cañedo et al., 2011). En la tabla 4 se puede ver las principales plagas y el color que le atrae a cada una de estas.

2.2.1.3.2.1 Trampa pegante

Es un método de control etológico. Este puede ser de cartón, madera o metal, en la cual se impregna una sustancia a base de grasa (o mezcla de grasas), aceite y alcohol. Se usa para atrapar a los insectos que se posan o chocan contra esta cuando ellos se mueven a través de su hábitat natural (SENASA, 2006). Algunas de las plagas que controla son la mosca minadora, mosquitos o síldos, cigarritas y pulgones alados (Cañedo et al., 2011). En la figura 8 se puede observar un ejemplo de cómo está conformada la trampa pegante amarilla.

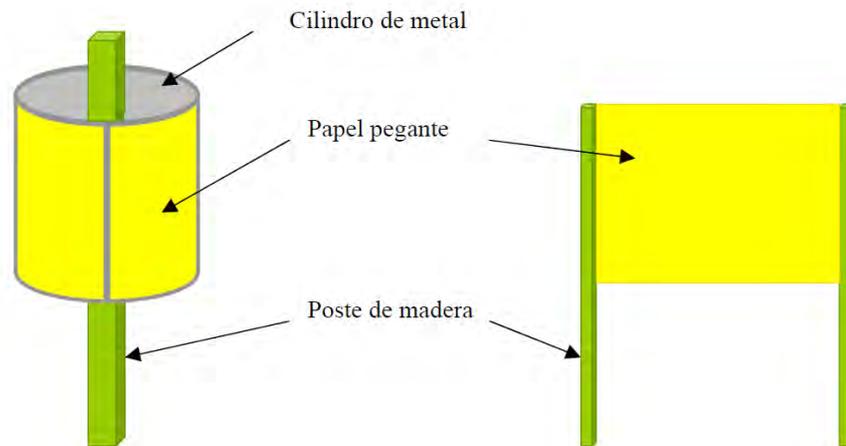


Figura 8: Trampa pegante amarilla. Adaptado de (SENASA, 2006).

2.2.1.3.2.2 Trampa de luz

Una de las más usadas para la captura de insectos es la trampa de luz, la cual puede ser de distintos colores incluyendo la luz negra y la ultravioleta. La captura se produce debido a que la alta iluminación de la trampa interfiere en la normal orientación fótica de un insecto nocturno, logrando así que este sea atraído por la fuente de luz (SENASA, 2006). En la figura 9 se observa un ejemplo de trampa de luz blanca fluorescente.



Figura 9: Lámpara de luz blanca fluorescente. Adaptado de Camborda et al. (2015).

2.2.1.3.3 Control biológico

Consiste en la represión de plagas mediante sus enemigos naturales o controladores biológicos, como pueden ser parasitoides, predadores o entomopatógenos. En la figura 10 se puede ver un caso de predadores.



Figura 10: Predador Hippodamia para el pulgón. Adaptado de Cañedo et al. (2011).

2.2.2 Conceptos de aprendizaje automático en imágenes

2.2.2.1 Aprendizaje automático

Aprendizaje automático o Machine Learning es un campo de estudio cuyo objeto de estudio incluye la automatización de la adquisición de nuevo conocimiento y desarrollo de habilidades cognitivas mediante la práctica, la organización de nuevo conocimiento en representaciones efectivas y el descubrimiento de nuevos hechos y teorías por medio de la observación y experimentación (Michalski et al., 1983).

2.2.2.1.1 Aprendizaje profundo en imágenes

Aprendizaje profundo o Deep Learning es una área del aprendizaje automático, a la cual se le añade un nivel de complejidad, pues la data es transformada usando distintas funciones que permiten su representación de forma jerárquica bajo distintos niveles de abstracción (Schmidhuber, 2015). Una ventaja del aprendizaje profundo es que nos permite extraer características importantes de la data y jerarquizarlas siendo las de mayor nivel jerárquico en abstracción formadas por las de menor nivel de abstracción (LeCun et al., 2015). Una posible aplicación del aprendizaje profundo es la construcción de clasificadores de imágenes de insectos, pues el sistema computarizado de procesamiento basado en la visión que utiliza el aprendizaje automático le permite identificar las características más importantes en las imágenes de insectos (Martineau et al., 2017).

2.2.2.1.1.1 Identificación de insecto agrícola en una imagen

La identificación, segmentación o detección del insecto es el proceso para encontrar en las coordenadas donde se encuentra un insecto en la imagen y así segmentar la región de interés (Thenmozhi & Srinivasulu Reddy, 2019). En la figura 11 se puede observar que la imagen ha pasado por el proceso de identificación de insecto y se ha encerrado en un cuadrilátero la región de interés, el cual encapsula al insecto y deja de lado el fondo color verde.

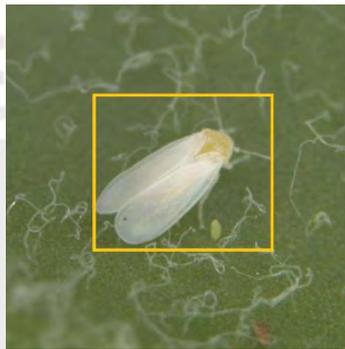


Figura 11: Identificación de la región de interés de la imagen. Adaptado de (ICAR-NBAIR, 2013).

2.2.2.1.1.2 Clasificación de insectos agrícolas

La clasificación de insectos agrícolas es el paso siguiente a la identificación de insectos. El proceso consiste en, dada la imagen con la región de interés, procesarla de manera que al final se le otorgue una etiqueta al insecto dentro de un conjunto de etiquetas pre-específicas, el cual terminará siendo el nombre de su especie. La clasificación de insectos es una tarea difícil debido a la complejidad de la estructura de los insectos y la similitud que existe entre distintas especies (Thenmozhi & Srinivasulu Reddy, 2019). En la figura 12 se puede distinguir distintos tipos de insectos los cuales serían las etiquetas del proceso de clasificación.

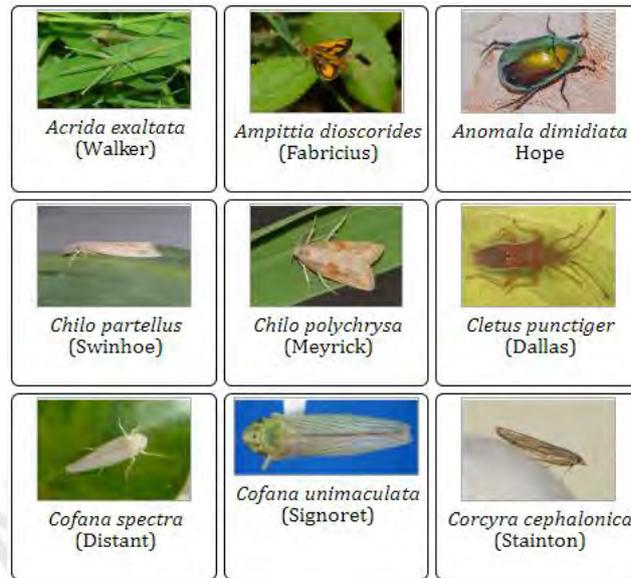


Figura 12: Tipos de insectos con etiquetas de nombre. Adaptado de (ICAR-NBAIR, 2013).

2.2.2.2 Red neuronal convolucional

Una red neuronal convolucional o Convolutional Neural Network (CNN) es un tipo de red neuronal, el cual combina tres técnicas arquitectónicas para asegurar cierto grado de cambio e invariancia en los niveles jerárquicos. Estos son los campos receptivos locales, la compartición de pesos y el submuestreo espacial o temporal (LeCun & Bengio, 1998). La estructura de un CNN incluye capas convolucionales, capas max pooling y capas fully connected. La capa convolucional actúa como filtro, pues su objetivo principal es extraer características de la imagen. La capa max pooling realiza un muestreo (subsampling) y retiene la información más importante de la imagen. Esta capa reduce el tamaño espacial de su representación y, por ende, el número de parámetros; asimismo, previene el sobreajuste, lo cual hace al modelo más eficiente. Por último, las capas fully connected usan una función de activación softmax para así obtener las características de alto nivel de la imagen (Thenmozhi & Srinivasulu Reddy, 2019). En la figura 13 se muestra una típica CNN para clasificar imágenes.

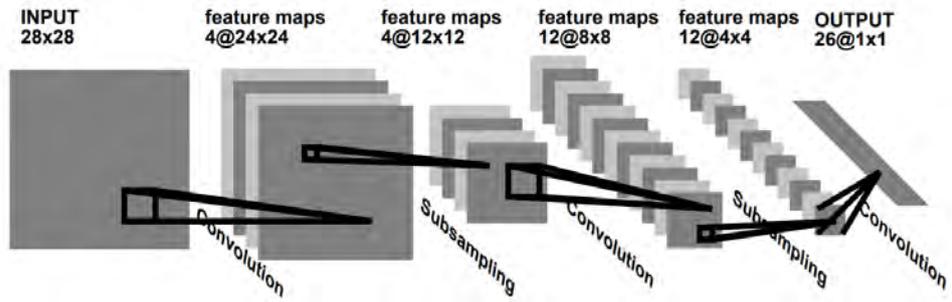


Figura 13: CNN para clasificar una imagen. Adaptado de LeCun & Bengio (1998).



Capítulo 3. Estado del Arte

3.1 Introducción

El objetivo del estado del arte es dar a conocer, mediante los objetivos de revisión, cómo es que se logra la identificación y clasificación de insectos a partir de imágenes de estos en trampas pegantes o en su hábitat natural.

3.2 Objetivos de revisión

- Conocer los conceptos en el área de Deep Learning para segmentación y reconocimiento de imágenes de insectos.
- Conocer a nivel conceptual las principales arquitecturas de los modelos de Deep Learning para detección y clasificación de imágenes.
- Conocer las similitudes que tienen los modelos que fueron entrenados con imágenes de insectos en trampas pegantes y otros que hayan sido entrenados con imágenes de insectos voladores en hojas u otras superficies.
- Conocer las métricas usadas para evaluar los modelos de clasificación de imágenes que usan aprendizaje profundo y sus resultados.

3.3 Preguntas de revisión

- P1. ¿Qué tipos de algoritmos se presentan para la segmentación de imágenes de insectos o relacionados y cuáles son sus métricas y resultados?
- P2. ¿Qué arquitecturas de Deep Learning se vienen usando para la clasificación de insectos agrícolas?
- P3. ¿Qué métricas son las usadas en los modelos de clasificación que usan Deep Learning para reconocimiento de imágenes de insectos y cuáles son sus resultados?

3.4 Estrategia de búsqueda

3.4.1 Motores de búsqueda a usar

Se usará Scopus y Web of Science.

3.4.2 Cadenas de búsqueda a usar

Para P1 se usará:

Cadena 1 en Scopus: TITLE-ABS-KEY ("image segmentation" AND insect)

Cadena 1 en Web of Science: TS=("image segmentation") AND TS=(insect)

Para P2 y P3 se usará:

Cadena 2 en Scopus: TITLE-ABS-KEY(("deep learning" OR "neural network") AND ("insect classification" OR "insect identification" OR "insect recognition"))

Cadena 2 en Web of Science: TS=("deep learning" OR "neural network") AND TS=("insect classification" OR "insect identification" OR "insect recognition")

3.4.3 Documentos encontrados

Tabla 5: Documentos encontrados por cadena y motor de búsqueda

Cadena	Motor de búsqueda	
	Scopus	Web of Science
Cadena 1	137	29
Cadena 2	38	13

3.4.4 Criterios de inclusión/exclusión

Los criterios de inclusión que se usaron en la revisión sistemática fueron los siguientes:

1. El estudio detalla un algoritmo de segmentación y/o clasificación de imagen de insecto.
2. El estudio presenta métricas que respalden los resultados del modelo de clasificación.
3. El estudio está escrito en inglés o español.

Adicionalmente, no se puso restricciones en el tipo de métricas usadas y no se puso límite en cuanto a la fecha de publicación de los estudios. Finalmente, se excluyeron los estudios que no trataban sobre la identificación de cualquier tipo de insecto.

3.5 Formulario de extracción de datos

En la tabla 6 se muestran los campos del formulario de extracción para P1, que es referente al algoritmo de segmentación de imagen y sus métricas y resultados.

Tabla 6: Campos del formulario de extracción de datos para P1.

Campo	Descripción	Pregunta
Id		General
Fecha de extracción		General
Autores		General
Título		General
Año de publicación		General
Número de veces citado		General
Insecto(s)	Nombre de insectos identificados	General
Tipo de superficie	Tipo de superficie de la imagen	General
Problemas abordados		General
Dataset		General
Tipo de algoritmo usado	Qué tipo de algoritmo fue usado	P1
Detalle de algoritmo	Descripción	P1
Métricas y resultados	Listado de métricas usadas con resultados	P1

En la tabla 7 se muestran los campos del formulario de extracción para P2 y P3, que es referente al modelo de Deep Learning de clasificación de imágenes de insectos y sus métricas y resultados.

Tabla 7: Campos del formulario de extracción de datos para P2 y P3.

Campo	Descripción	Pregunta
Id		General
Fecha de extracción		General
Autores		General
Título		General
Año de publicación		General
Número de veces citado		General
Insecto(s)	Nombre de insectos identificados	General
Tipo de arquitectura usada	Descripción del tipo de arquitectura de Deep Learning usada	P2
Dataset		General
Uso de transfer learning	Sí/No	General
Detalle del uso de transfer learning	Descripción	General
Métricas y resultados del modelo final	Listado de métricas usadas con resultados del modelo final	P3

Se puede visualizar la lista de estudios primarios que cumplieron con los criterios de inclusión/exclusión en el Anexo 1. Además, el formulario de extracción para P1 se encuentra en el Anexo 2 y para P2 y P3, en el Anexo 3.

3.6 Resultados de la revisión

3.6.1 Respuesta a pregunta: ¿Qué tipos de algoritmos se presentan para la segmentación de imágenes de insectos o relacionados y cuáles son sus métricas y resultados?

De la revisión realizada, se puede apreciar que los estudios utilizan distintos algoritmos para hacer la segmentación de imagen. Entre estos se encuentran la conversión de color a Luv, binary thresholding, morphological closing y blob detection (Rustia et al., 2020); conversión de color a Lab y el método de Otsu

(Espinoza et al., 2016); conversión de color a YCbCr y el algoritmo de watershed (Xia et al., 2015); conversión a grayscale y Canny Edge Detection (Thenmozhi & Srinivasulu Reddy, 2019); Otsu threshold segmentation (S. Zhu et al., 2019); binary thresholding con edge enhancement (Shah et al., 2019); conversión a HSI con watershed delineation (S. El-Faki et al., 2018); superpixel segmentation SLIC con DBSCAN cluster (J. Chen et al., 2017); análisis multifractal (Xia et al., 2012); conversión de color a HSI junto con k-means clustering (Cho et al., 2007); uso de Faster R-CNN (Nieuwenhuizen et al., 2018) y el uso de YOLO neural network (Zhong et al., 2018).

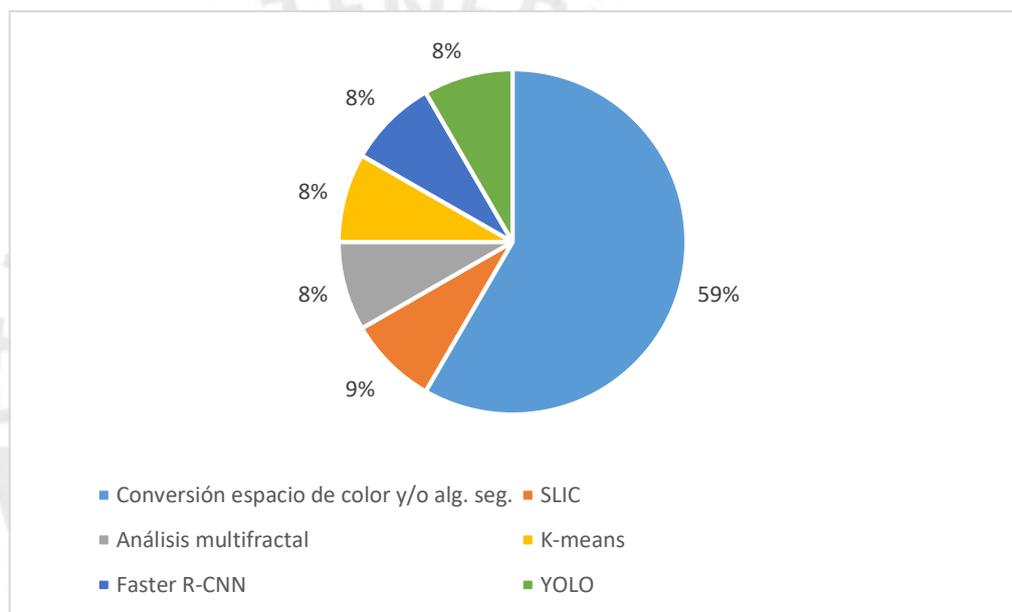


Figura 14: Distribución de algoritmos usados para segmentación de imagen.

En la figura 14 se muestra la distribución de algoritmos encontrados y se aprecia que más de la mitad de los estudios (59%) hicieron una conversión del espacio de color original de la imagen y/o usaron un algoritmo de segmentación.

Con respecto a las métricas y resultados de los algoritmos, se encontraron métricas de evaluación como Accuracy (J. Chen et al., 2017; Cho et al., 2007; Nieuwenhuizen et al., 2018; Rustia et al., 2020; S. El-Faki et al., 2018; Xia et al., 2015; Zhong et al., 2018) y F-score (Cho et al., 2007; Xia et al., 2012, 2015). En la tabla 8 se aprecia los resultados más significativos de los estudios que usaron un método para validar sus algoritmos y publicaron estos resultados.

Tabla 8: Métricas y resultados de los algoritmos de segmentación.

Estudio	Métrica de evaluación	Dataset	Insecto(s)	Resultado
E01	Accuracy	Propio	No específico	93%
E02	Accuracy	Propio	Empoasca flavescens	98.81%
E04	Accuracy	Propio	Whitefly (<i>Bemisia tabaci</i> Genn), aphids (<i>Aphis gossypii</i> Glover) y thrips (<i>Thrips tabaci</i> L.)	93%
E05	F-score	Propio	Whiteflies (Genus <i>Bemisia</i>)	96.50%
E09	Accuracy	Propio	Brown Planthopper	98%
E10	Accuracy	Propio	Huevos de <i>Cadra cautella</i>	91.50%
E11	Accuracy	ImageNet	7 tipos de insectos	92.50%

3.6.2 Respuesta a pregunta: ¿Qué arquitecturas de Deep Learning se vienen usando para la clasificación de insectos agrícolas?

De los estudios revisados, se encontraron diversas arquitecturas para los modelos. Entre estos se encuentran VGG-13 (Jian Chen et al., 2018), InceptionV3 (Chulu et al., 2019; Reza et al., 2019; Shen et al., 2018; Souza et al., 2019), VGG-16 (Wu et al., 2019), AlexNet (Lim et al., 2017), ResNet-50 (Kim et al., 2018), Inception ResNet V2 (Nieuwenhuizen et al., 2018) y dos estudios (Li & Yang, 2020; Thenmozhi & Srinivasulu Reddy, 2019) usaron arquitecturas personalizadas.

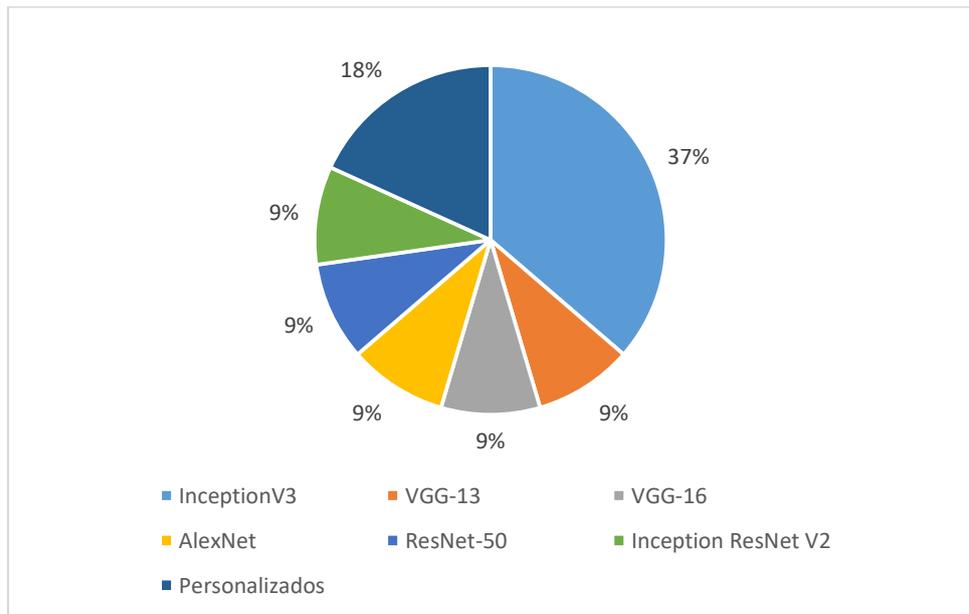


Figura 15: Distribución de las arquitecturas de los modelos de Deep Learning

En la figura 15 se observa la distribución de las arquitecturas los modelos de los estudios revisados. Se aprecia que el 37% de estos usaron la arquitectura InceptionV3.

3.6.3 Respuesta a pregunta: ¿Qué métricas son las usadas en los modelos de clasificación que usan Deep Learning para reconocimiento de imágenes de insectos y cuáles son sus resultados?

Se encontró que los estudios usaban métricas de evaluación como Accuracy (Chulu et al., 2019; Kim et al., 2018; Lim et al., 2017; Reza et al., 2019; Souza et al., 2019; Thenmozhi & Srinivasulu Reddy, 2019; Wu et al., 2019), F-score (Jian Chen et al., 2018) y mAP (mean average precision) (Shen et al., 2018). En la tabla 9 se detallan los resultados más significativos de los estudios revisados.

Tabla 9: Métricas y resultados de los modelos de clasificación de insectos.

Estudio	Métrica de evaluación	Dataset	Insecto(s)	Resultado
E06	Accuracy	ImageNet	24 tipos de insectos	97.47%
E12	F-score	Propio	Aphid nymphs	96.06%

E13	Accuracy	Propio	Fall Army Worm Moth	82%
E14	Accuracy	NBAIR	50 tipos de insectos	95.40%
E15	Accuracy	IP102	102 tipos de insectos	57.08%
E16	Accuracy	ImageNet	15 tipos de insectos	83.80%
E17	mAP	Propio	6 tipos de insectos	87.99%
E18	Accuracy	ImageNet	27 tipos de insectos	81.82%
E19	Accuracy	ImageNet	6 tipos de insectos	97%
E20	Accuracy	ImageNet	27 tipos de insectos	89.98%
E21	Accuracy	MS-COCO	Whitefly, macrolophus y nesidiocoris	87.4%

3.7 Conclusiones

Después de haber revisado los estudios más relevantes, se puede concluir lo siguiente:

- Los estudios sobre los algoritmos de segmentación de imágenes en el estado del arte demuestran que existe una gran variedad de métodos que se pueden usar y que obtienen buenos resultados.
- Los estudios sobre los modelos de Deep Learning para segmentación de imágenes de insectos señalan que es factible usar un algoritmo de segmentación de color, como también usar redes neuronales convolucionales. En ese sentido, se puede usar un modelo pre-entrenado de detección de imágenes y aplicar transfer learning para adecuarlo a un cierto problema de detección y clasificación de imágenes, en este caso, el de insectos agrícolas.
- Como se puede observar en la tabla 8 y en la tabla 9, solo tres estudios (E4, E5 y E21) tratan específicamente de la detección de la mosca blanca (insecto plaga de relevancia para los plantíos de hortalizas en Perú). El resto de estudios se enfoca en identificar y/o clasificar otros tipos de insectos plaga que no son relevantes en el contexto del proyecto de fin de carrera.

Capítulo 4. Construir un generador de imágenes realistas de trampas pegantes con insectos plaga de hortalizas

4.1 Introducción

En el presente capítulo se muestra el desarrollo del objetivo específico 1, el cual plantea construir un generador de imágenes realistas de trampas pegantes que sirvan para poder entrenar y evaluar los modelos de detección de insectos en situaciones próximas a la realidad. Este objetivo fue planteado por la limitada disponibilidad de imágenes reales de trampas pegantes con los insectos que se pretende reconocer. A continuación, se va a tratar como resultados alcanzados el documento de requerimientos funcionales del generador, el corpus de imágenes base de insectos plaga de hortalizas y el generador de imágenes realistas de trampas pegantes.

4.2 Resultados alcanzados

4.2.1 Documento de requerimientos funcionales del generador de imágenes realistas

Este documento describe las principales funciones que el generador debe tener para poder simular de forma controlada diversas condiciones que se presentan en campo, siendo esto factores como la variación de iluminación, ruido, presencia de otros insectos u objetos diversos, entre otros. En la tabla 10 se muestra el listado de los requerimientos funcionales del generador. Se tiene la columna prioridad, la cual es una escala del 1 al 5, donde 1 significa una prioridad muy alta y 5, una prioridad muy baja. Adicionalmente, se cuenta con la columna exigible, la cual muestra “E” si la funcionalidad es exigible y “D”, si es deseable.

Tabla 10: Listado de requerimientos funcionales del generador de imágenes realistas de trampas pegantes.

Nro.	Funcionalidades	Prioridad	Exigible
Generador de imágenes realistas de trampas pegantes			
RP01	El generador debe permitir al usuario especificar la cantidad de imágenes a generar.	1	E
RP02	El generador debe permitir al usuario especificar las dimensiones que tendrán todas las imágenes a generar.	1	E

RP03	El generador debe utilizar tres etiquetas para los tipos de insectos, estos son: “mosca blanca”, “mosca minadora” y “pulgón verde del melocotonero”.	1	E
RP04	El generador debe posicionar en coordenadas aleatorias una cantidad especificada por el usuario de insectos por tipo que aparecerán en todas las imágenes generadas.	1	E
RP05	El generador debe posicionar en coordenadas aleatorias algunas imágenes como otros insectos, hojas, tallos, ramas o flores, las cuales tendrán categorías “otros insectos” y “objetos varios” a fin de simular condiciones reales. La cantidad de muestras de ambas categorías será especificada por el usuario.	3	E
RP06	El generador debe permitir al usuario definir un rango de valores para escoger aleatoriamente el porcentaje de ruido que tendrá cada imagen simulada.	2	E
RP07	El generador debe permitir al usuario elegir si la imagen simulada tendrá un foco de luz en una coordenada aleatoria o no.	4	D
RP08	El generador debe permitir al usuario definir un rango de valores para escoger aleatoriamente el porcentaje de iluminación que se aplicará para todas las imágenes a generar.	2	E
RP09	El generador debe permitir exportar en formato .csv el detalle de cada región de interés generada. Esto es el identificador de la imagen generada en la que se encuentra, las coordenadas de la región de interés de cada insecto de interés agregado y la clase a la que este pertenece.	1	E

4.2.2 Corpus de imágenes base de insectos plaga de hortalizas de relevancia en Perú

Las imágenes base se obtuvieron de internet y fueron recortadas cuidadosamente para centrarse solo en el insecto y así evitar que el fondo sea un distractor del algoritmo de segmentación y también que pueda alterar el tamaño real de los insectos al momento de posicionarlos aleatoriamente en el fondo amarillo que simulará la trampa pegante amarilla. Estas imágenes base han sido validadas por un experto. A continuación, en las figuras 16, 17 y 18 se muestra una selección de imágenes por cada tipo de insecto.



Figura 16: Muestra de imágenes recolectadas de la mosca blanca.



Figura 17: Muestra de imágenes recolectadas de la mosca minadora.

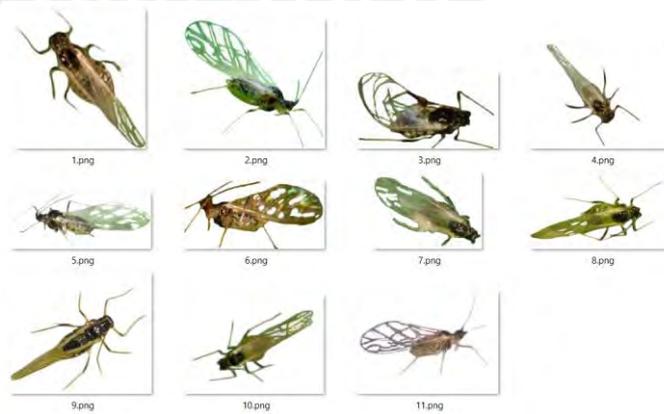


Figura 18: Imágenes recolectadas del pulgón verde del melocotonero.

4.2.3 Generador de imágenes de trampas pegantes con insectos plaga de hortalizas

El generador usa imágenes tipo “png” y “jpg”. Para el primero ya se cuenta con un canal para la transparencia, en cambio, para el segundo formato de imagen se usaron algoritmos de segmentación para así aislar el insecto del fondo, esto es encontrar una máscara, la cual pueda usarse para filtrar el insecto del fondo. Para el caso de la mosca blanca, se obtuvo la máscara convirtiendo la imagen original a escala de grises y luego se procedió a hacer una segmentación binaria. Para el caso de la mosca minadora, se hizo un cambio de espacio de color de BGR (blue-green-red o azul-verde-rojo) a HSV (hue-saturation-value o matiz-saturación-valor), luego se filtró el fondo de las imágenes, para lo cual se tuvo que establecer un límite superior e inferior de colores verde en espacio HSV y después la máscara se obtuvo con una segmentación binaria invertida. Finalmente, para el pulgón verde del melocotonero, se hizo una segmentación manual de las muestras y estas fueron guardadas en formato “png”. A todas estas imágenes sin fondo de los insectos de interés, junto con imágenes de otros insectos y otros objetos, las cuales también se encuentran en formato “png”, se le aplicó la técnica de aumentación de datos, las cuales consistieron en girar la imagen entre 0° y 360° y la reflexión horizontal y/o vertical de la imagen. Después, estas muestras fueron posicionadas en coordenadas aleatorias dentro del rango de las dimensiones de la trampa pegante. Asimismo, se le aplicó un filtro de ruido e iluminación a la imagen usando la librería OpenCV. En las figuras 19 y 20, se muestra la imagen original, la máscara y la imagen segmentada que usará el generador.



Figura 19: Segmentación de una imagen de la mosca blanca.



Figura 20: Segmentación de una imagen de la mosca minadora.

Finalmente, se exportó en formato “csv” el detalle de cada región de interés generada, esto es el nombre con el cual se guardó la imagen generada de trampa pegante, las coordenadas de la región de interés de cada insecto plaga agregado y la clase a la que este pertenece.

4.3 Discusión

Para la elaboración del generador de imágenes realistas fue necesario segmentar de manera manual las imágenes del pulgón verde del melocotonero, pues se contaban con pocas muestras y el fondo de estas imágenes recolectadas era de distintos colores que incluso se podrían confundir con el pulgón verde en sí.

Por otro lado, el código hecho para segmentar imágenes de la mosca blanca sirve, en general, para segmentar el color blanco de cualquier imagen, mientras que el código hecho para la segmentación de la mosca minadora, solo sirve para eliminar el color verde de imágenes. Además, el código hecho para importar imágenes en formato “png” puede servir para agregar cualquier otro tipo de insecto (en caso se tenga un corpus de imágenes del insecto sin fondo) y así poder generar un dataset de imágenes con distintos insectos plaga que afectan la producción de hortalizas.

Finalmente, se concluye que el objetivo propuesto cumple su propósito de poder generar imágenes realistas de trampas pegantes amarillas con insectos de tipo mosca blanca, mosca minadora y pulgón verde del melocotonero, ya que en la actualidad no se cuenta con un dataset de imágenes reales de estas trampas pegantes utilizadas en los plantíos de hortalizas.

Capítulo 5. Construir un modelo algorítmico para la identificación y clasificación de insectos plaga de hortalizas en imágenes de trampas pegantes

5.1 Introducción

En el presente capítulo se muestra el desarrollo del objetivo específico 2, el cual plantea construir un modelo algorítmico para la identificación y clasificación de insectos plaga de hortalizas en imágenes de trampas pegantes. A continuación, se va a tratar como resultados alcanzados el corpus de imágenes generadas de trampas pegantes, los modelos de aprendizaje profundo de detección que se usarán y finalmente, se realizará un reporte de comparación de los modelos que determinará cuál es el más óptimo a usar.

5.2 Resultados alcanzados

5.2.1 Corpus de imágenes generadas de trampas pegantes con insectos identificados y etiquetados para el entrenamiento y evaluación

El corpus fue generado usando el resultado esperado N°3. Contiene 500 imágenes de trampas pegantes y un archivo resumen el cual indica las coordenadas de los insectos de interés. Cada imagen tiene una altura y anchura de 600 pixeles, también posee una cantidad de insectos de relevancia que varía en un rango del 10 al 15, un porcentaje de iluminación que varía entre 0.80 y 1.00 y un porcentaje de ruido que varía entre 0.010 y 0.030. En la figura 21 se muestra un ejemplo de una imagen de trampa generada.



Figura 21: Ejemplo de imagen de trampa pegante generada.

5.2.2 Modelos algorítmicos para identificar y clasificar insectos plaga a base de imágenes de trampas pegantes (R4)

Los modelos de detección a usar fueron Faster R-CNN (Ren et al., 2016) y YOLOv4 (Bochkovskiy et al., 2020, p. 4). El primero se obtuvo de la librería torchvision, la cual posee el modelo Faster R-CNN con la arquitectura de la red neuronal ResNet50-FPN como backbone. Luego, se hizo un cambio en la capa de clasificación y se extrajeron las características más importantes. Se usó el método de optimización de gradiente descendente con el optimizador Momentum, cuyo valor fue de 0.9; también, se usó un manual stepping con un learning rate inicial de 0.0003, una secuencia de learning rate de 0.00003 y límites de step de 900000 y 1200000. Estos valores son los que provee Tensorflow por defecto. Se usaron 400 imágenes para el entrenamiento y 100 para prueba. El batch size usado para el

entrenamiento fue de 1 para 130 épocas. En la figura 22 se puede ver las detecciones que se obtuvo en la imagen de prueba de trampa pegante (figura 21) con el modelo Faster R-CNN.

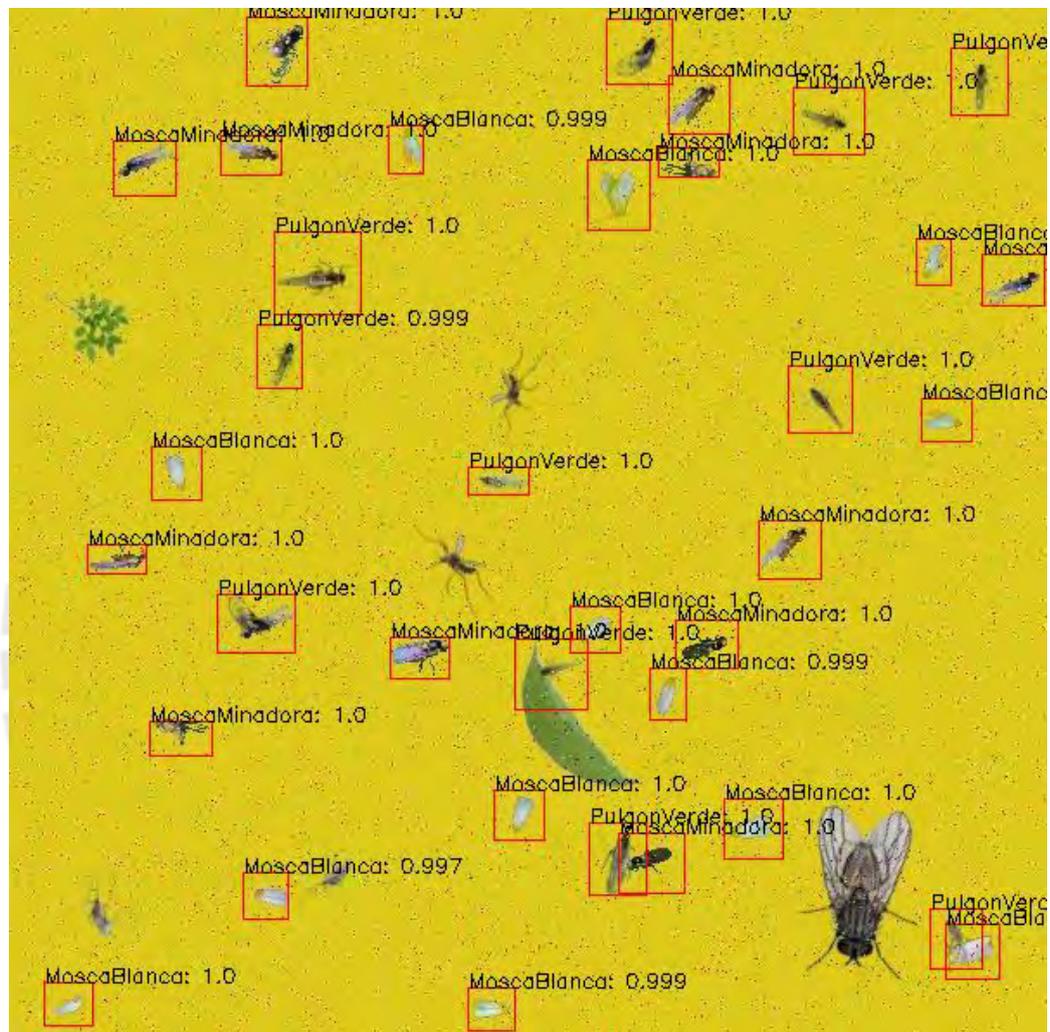


Figura 22: Ejemplo de las detecciones hechas por Faster R-CNN.

Se usaron las 100 imágenes de validación del dataset y se obtuvo la métrica mAP (mean average precision) calculando el AP de cada clase con el método 11 point interpolation, en donde se consideró una detección como True Positive si el IoU es mayor a 50%, caso contrario, la detección sería considerada como False Positive. En la figura 23 se muestra el Average Precision de cada tipo de insecto, junto con el mAP logrado de 94.06%.

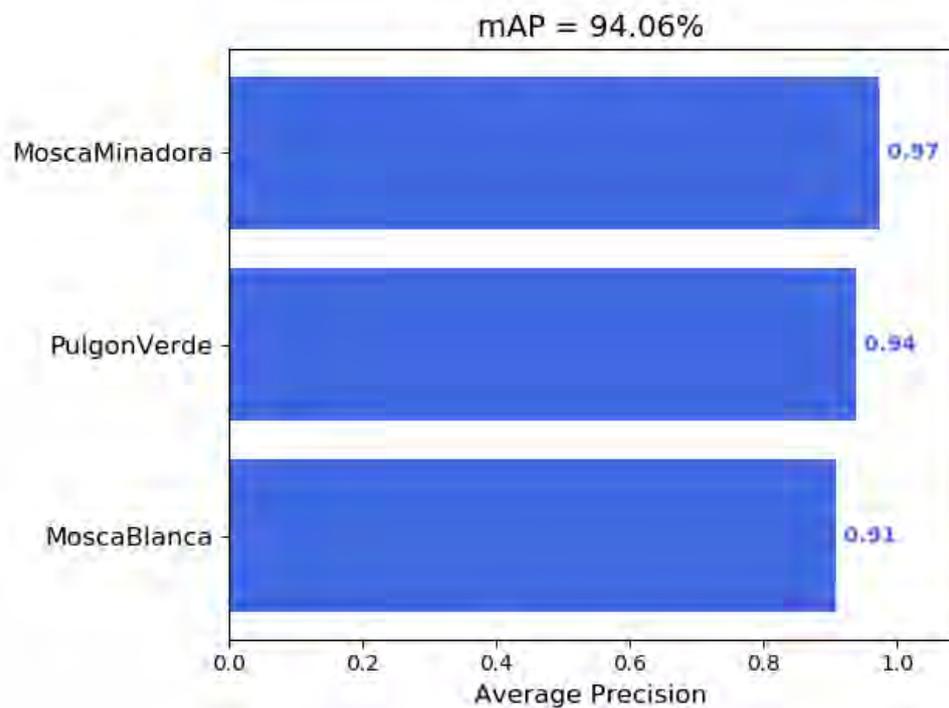


Figura 23: Average Precision por tipo de insecto y mAP del detector Faster R-CNN.

Para el detector YOLOv4, se descargó la arquitectura de la red junto con los pesos del detector de objetos y los archivos de configuración para el entrenamiento. Se utilizó un optimizador Adam con un learning rate de 0.001, momentum de 0.949, weigh_decay de 0.0005. De manera similar al primer modelo, se usaron las mismas 400 imágenes para el entrenamiento y las 100 restantes para prueba. El batch size usado para el entrenamiento fue de 64 para 1600 épocas. En la figura 24 se puede ver las detecciones que se obtuvo en la imagen de prueba de trampa pegante (figura 21) con el modelo YOLOv4.

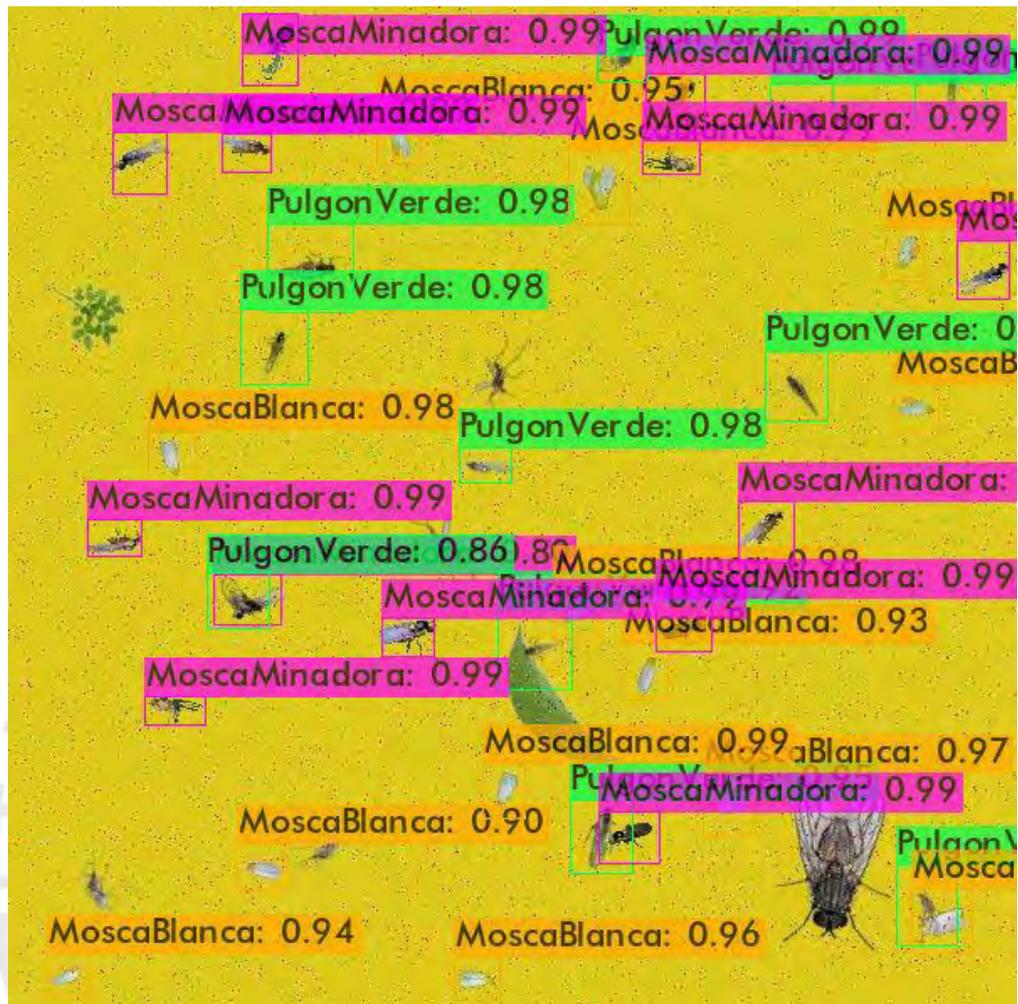


Figura 24: Ejemplo de las detecciones hechas por YOLOv4.

De igual forma, se usaron las 100 imágenes de validación y se obtuvo el mAP del detector con las tres clases. En la figura 25 se muestra el Average Precision de cada tipo de insecto, junto con el mAP logrado de 95.82%.

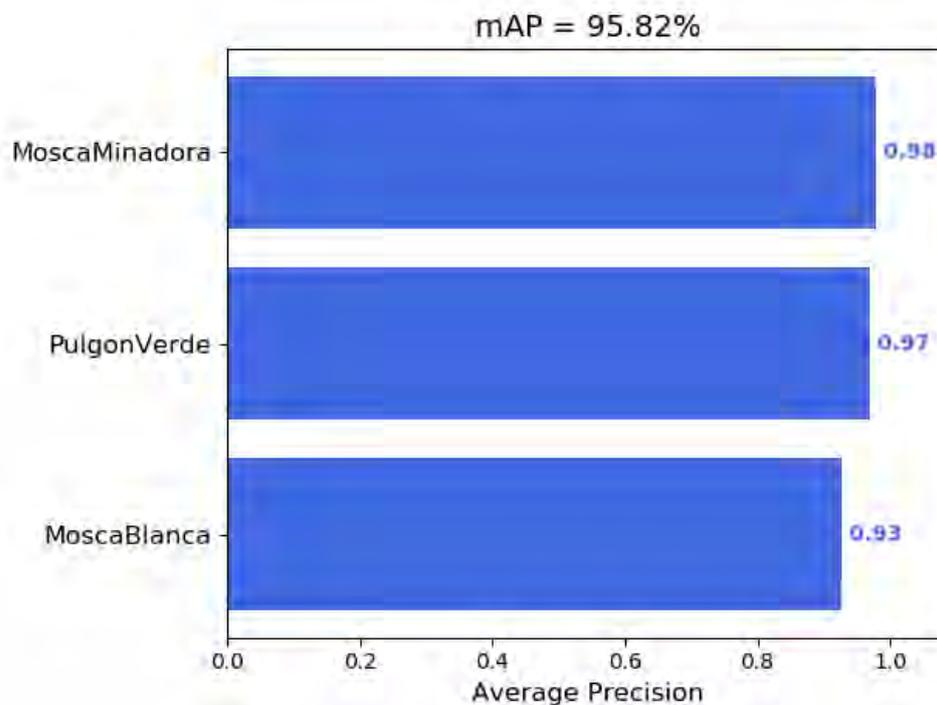


Figura 25: Average Precision por tipo de insecto y mAP del detector YOLOv4.

5.2.3 Reporte de comparación de los modelos de detección de insectos

En primer lugar, se hizo la comparación de los modelos Faster R-CNN y YOLOv4 descritos en el resultado N° 5, para esto, se hizo una prueba de detección en el corpus generado de validación a una confianza de 50% y se hicieron gráficas de Precision vs. Recall por cada clase para cada modelo. En la figura 26 se muestran estas gráficas para el modelo Faster R-CNN, donde el AP de la mosca blanca fue igual a 90.95%, el de la mosca minadora fue de 97.31% y el del pulgón verde del melocotonero fue de 93.92%. En este caso, el mAP, es decir, el promedio de estos valores fue igual a 94.06%. En la figura 27 se muestran las gráficas para el modelo YOLOv4, donde el AP de la mosca blanca fue 92.78%, el de la mosca minadora fue 97.78% y el del pulgón verde del melocotonero fue 96.89%. También se calculó su mAP, el cual fue de 95.82%.

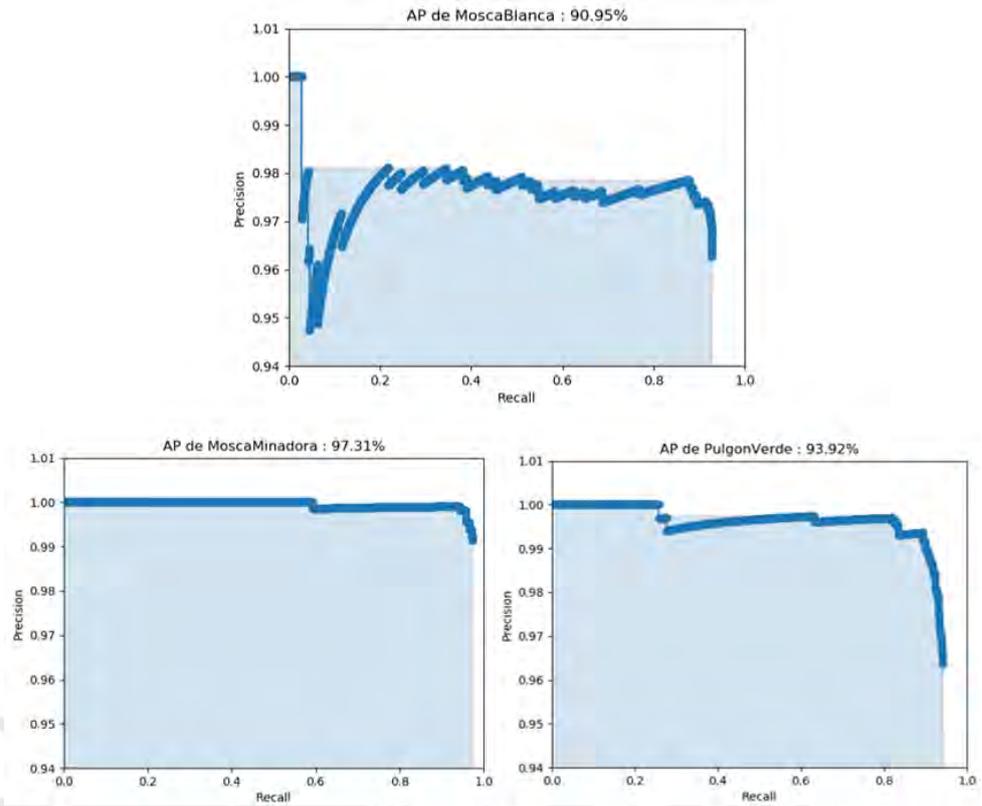


Figura 26: Gráficas de AP por clase del modelo Faster R-CNN.

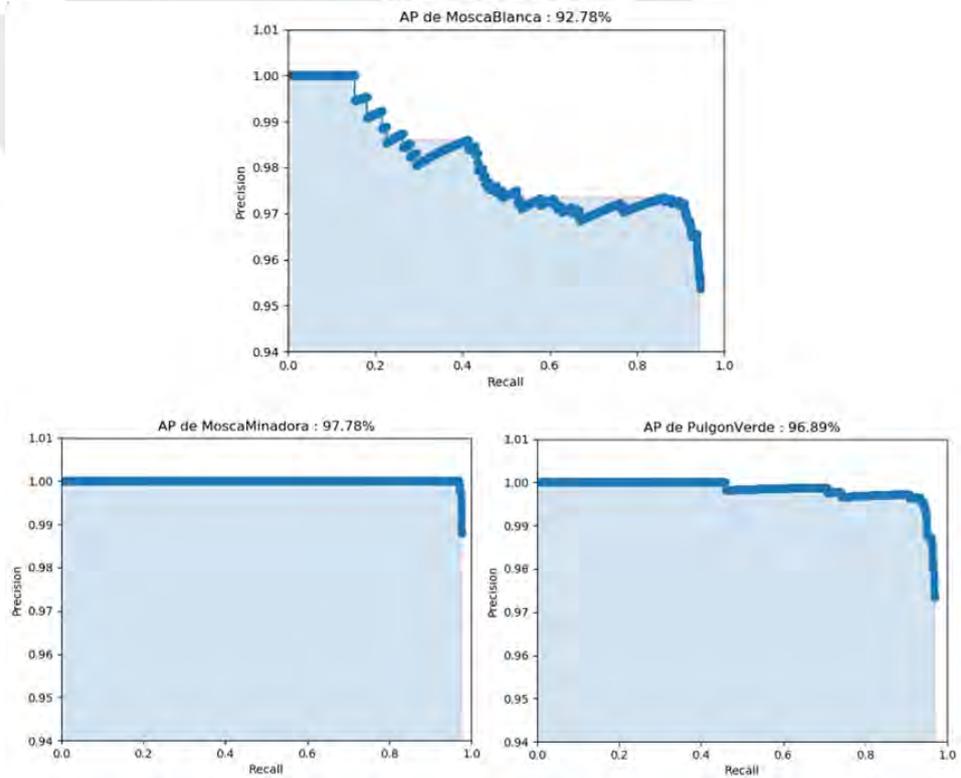


Figura 27: Gráficas de AP por clase del modelo YOLOv4.

En la tabla 11 se puede ver el resumen de la métrica mAP de los modelos para la data de validación.

Tabla 11: mAP de los modelos de detección.

Modelo	% mAP Validación (confianza 50%)
Faster R-CNN	94.06
YOLOv4	95.82

Por otro lado, se realizaron cinco experimentos en los cuales se entrenó modelos YOLOv4 con las imágenes generadas de trampas pegantes y con imágenes reales de trampas pegantes con insectos del tipo mosca blanca (solo para esta especie de insecto se tiene disponible imágenes de trampas pegantes reales). En la tabla 12 se especifica la composición de los datos de entrenamiento usados por cada modelo, la cantidad de imágenes artificiales y reales usadas en el entrenamiento, el porcentaje de AP de la clase “mosca blanca” que obtuvo el modelo en el corpus de validación con imágenes reales de trampas pegantes con detecciones a un nivel de confianza del 50%, la cantidad de TP y FP en la detección de cada modelo, la cantidad real de insectos identificados en el corpus y el porcentaje de aciertos por cada modelo.

Tabla 12: Modelos YOLO entrenados a comparar.

N° Modelo	Datos de entrenamiento	Cantidad de imágenes generadas usadas en set de entrenamiento	Cantidad de imágenes reales usadas en set de entrenamiento	% AP validación de mosca blanca en data real (confianza 50%)	TP	FP	Cantidad real de mosca blanca en validación	% Aciertos de mosca blanca en validación
M0.5	1/2 artificial	200	0	0.00	0	16	44	0.00
M1	Artificial	400	0	0.00	0	15	44	0.00
M2	1/2 artificial y 1/2 real	200	75	17.05	9	3	44	20.45
M3	Real	0	150	35.47	16	3	44	36.36
M4	Artificial y real	400	150	60.60	28	3	44	63.64

5.3 Discusión

En los resultados mostrados del reporte de comparación de los modelos de detección Faster R-CNN y YOLOv4 se concluye que ambos modelos tienen desempeños aceptables, pues el mAP de cada uno es mayor al 90%. Se optó por usar YOLOv4 para los posteriores análisis debido a que este modelo obtuvo desempeños de detección mAP ligeramente mayores a los mostrados por Faster R-CNN, sin embargo, se podría realizar los experimentos entrenando modelos Faster R-CNN, pues quedó demostrado que este es un buen detector que ofrece desempeños de detecciones competitivos.

En los experimentos con data real, se concluye que un modelo de redes neuronales puede mejorar su predicción en una clase si, además de entrenarlo con data real, se le entrena con data generada artificialmente, pues el modelo M3 (entrenado con únicamente data real) obtuvo 16 predicciones correctas de un total de 44 y el modelo M4 (entrenado con la misma data real más las imágenes artificiales hechas por el generador desarrollado en este proyecto) obtuvo 28 de un total de 44 insectos del tipo mosca blanca etiquetados. Cabe mencionar que se investigó la razón por la que el modelo M1 (modelo entrenado solo con data artificial) obtuvo 0% de AP detectando mosca blanca en imágenes reales y esto es debido a que las predicciones se hicieron a un nivel de confianza de 50%. Se verificó que el modelo M1 sí reconocía las características de la mosca blanca en data real, pues detectaba al insecto a un nivel de confianza de 10%, valor que tal vez podría aumentar si se realizara el experimento con Faster R-CNN. Además de esto, el modelo le asignaba un bounding box más grande al anotado y, debido a que la métrica AP se calcula con un IoU de 50%, esta detección era calificada como un FP. En la figura 28 se puede ver unos ejemplos de lo ocurrido. En este caso, el rectángulo rojo es la predicción de M1 a un nivel de confianza de 10% y el rectángulo rosa es el bounding box anotado.



Figura 28: Ejemplos de FP de M1.

Capítulo 6. Crear una herramienta que permita visualizar la identificación, clasificación, conteo y generación de reportes de poblaciones de plagas de insectos de relevancia en plantíos de hortalizas en Perú a partir del ingreso de fotografías de trampas pegantes

6.1 Introducción

En el presente capítulo se muestra el desarrollo de una herramienta web, el cual tiene como finalidad ser el medio por el cual las personas que trabajen en control de plagas puedan hacer un análisis (identificación, clasificación y conteo) sobre las imágenes de trampas pegantes que el usuario pueda subir para controlar la presencia de plagas de relevancia sus plantíos. A continuación, se va a tratar como resultados alcanzados el documento de requerimientos funcionales de la herramienta web y la interfaz web que integre el detector de insectos plaga y genere reportes sobre poblaciones de insectos plaga a partir de imágenes de trampas pegantes.

6.2 Resultados alcanzados

6.2.1 Documento de requerimientos funcionales para la herramienta web

Este documento describe las principales funciones que la herramienta web debe tener para que sea lo más sencillo posible de usar por los trabajadores de control de plagas. En la tabla 13 se muestra el listado de los requerimientos funcionales del generador. Se tiene la columna prioridad, la cual es una escala del 1 al 5, donde 1 significa una prioridad muy alta y 5, una prioridad muy baja. Adicionalmente, se cuenta con la columna exigible, la cual muestra “E” si la funcionalidad es exigible y “D”, si es deseable.

Tabla 13: Listado de requerimientos funcionales de la herramienta web.

Nro.	Funcionalidades	Prioridad	Exigible
Herramienta web			
RP01	El sistema debe permitir al usuario subir una o más imágenes para que sean analizadas por el modelo algorítmico de identificación y clasificación.	1	E

RP02	El sistema debe mostrar las imágenes segmentadas y anotadas indicando la probabilidad con la que se clasificó cada insecto, siendo las posibles clases: “mosca blanca”, “mosca minadora”, “pulgón verde del melocotonero”.	1	E
RP03	El sistema debe indicar la cantidad de insectos identificados por especie para cada imagen subida por el usuario.	1	E
RP04	El sistema debe generar un reporte resumen de todas las imágenes subidas por el usuario que indique la distribución de los insectos detectados.	1	E

6.2.2 Interfaz web que integre los modelos algorítmicos de identificación y clasificación de insectos plaga y genere reportes sobre poblaciones de insectos plaga de relevancia en plantíos de hortalizas en el Perú

Se utilizó React Js para la vista del usuario y Flask para el servidor backend. La herramienta web cumple con todas las especificaciones del resultado esperado N°7. En la figura 29 se muestra la pantalla de inicio del sistema web con ocho imágenes subidas de trampas pegantes analizar. En la figura 30 se observa un carrusel de imágenes que tiene todas las imágenes subidas por el usuario con las respectivas detecciones de insectos relevantes y la probabilidad con la que estos fueron clasificados. Finalmente, en la figura 31 se muestra el reporte de resultados. Se utilizó una gráfica de barras para la distribución de cada insecto en cada imagen y un gráfico circular para el resumen del total de insectos identificados en todas las imágenes subidas de trampas pegantes amarillas.

Las pruebas unitarias fueron realizadas haciendo uso de la librería “unittest”, en donde se probó la función de detección de imágenes, la cual tiene como único parámetro una lista de imágenes en base64 (codificación en la que vienen las imágenes desde la vista del usuario). La respuesta esperada de la función de detección es un JSON que contenga una lista de las imágenes detectadas y la cantidad de insectos identificados por el modelo para cada insecto relevante. Se hizo la prueba con un arreglo vacío y con arreglos de tamaño 1 y 2 que contengan imágenes en base64. El test duró 94.816 segundos y finalizó todos los casos de prueba de manera satisfactoria.

Para las pruebas funcionales, se usó Selenium con el web driver Chrome. Se probó ingresar a la página web por su URL, luego se probó la funcionalidad del botón de subida de imágenes y finalmente, el hacer click al botón analizar, el cual manda la

lista de imágenes en base64 al servidor backend para que sean analizadas y anotadas. El test duró 19.511 segundos y finalizó el test de manera satisfactoria.

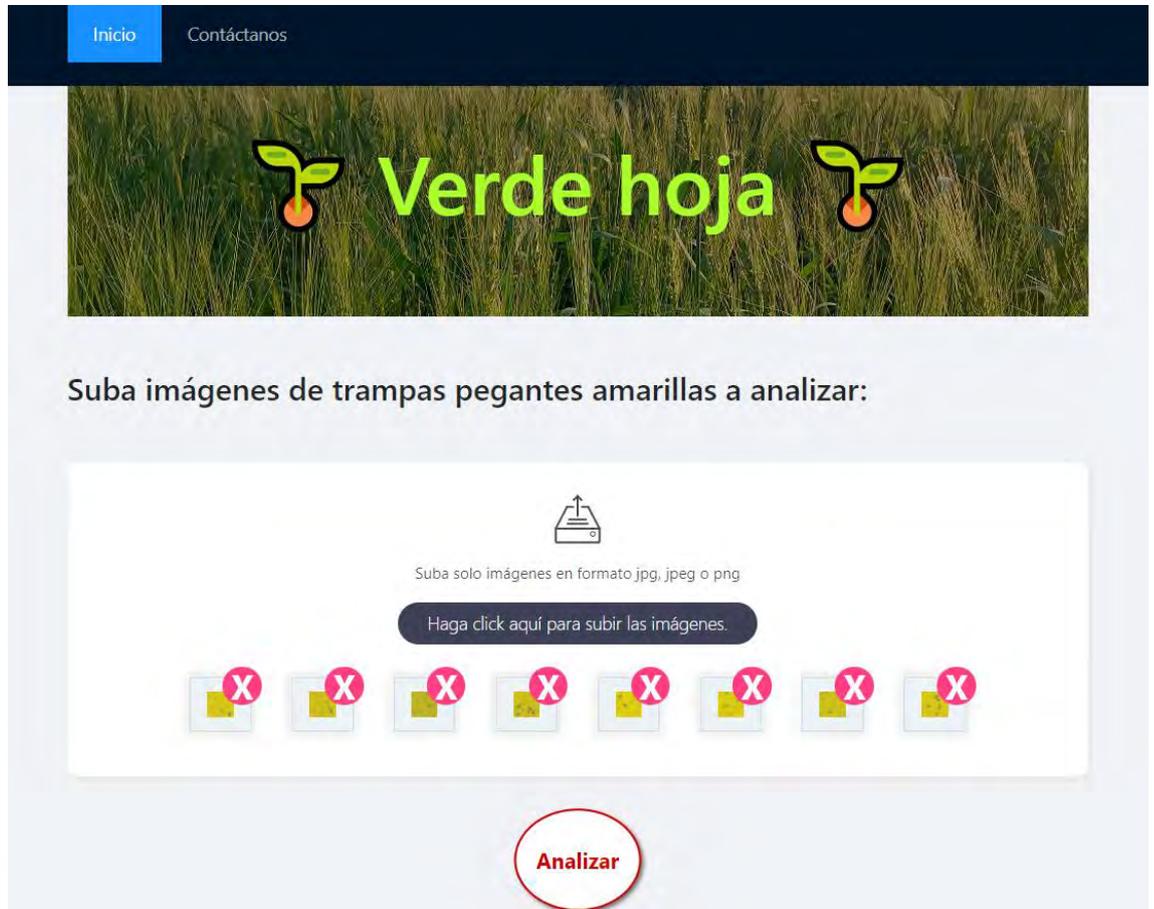


Figura 29: Pantalla de inicio de la herramienta web con imágenes subidas de trampas pegantes amarillas.

Resultados del análisis de imágenes:

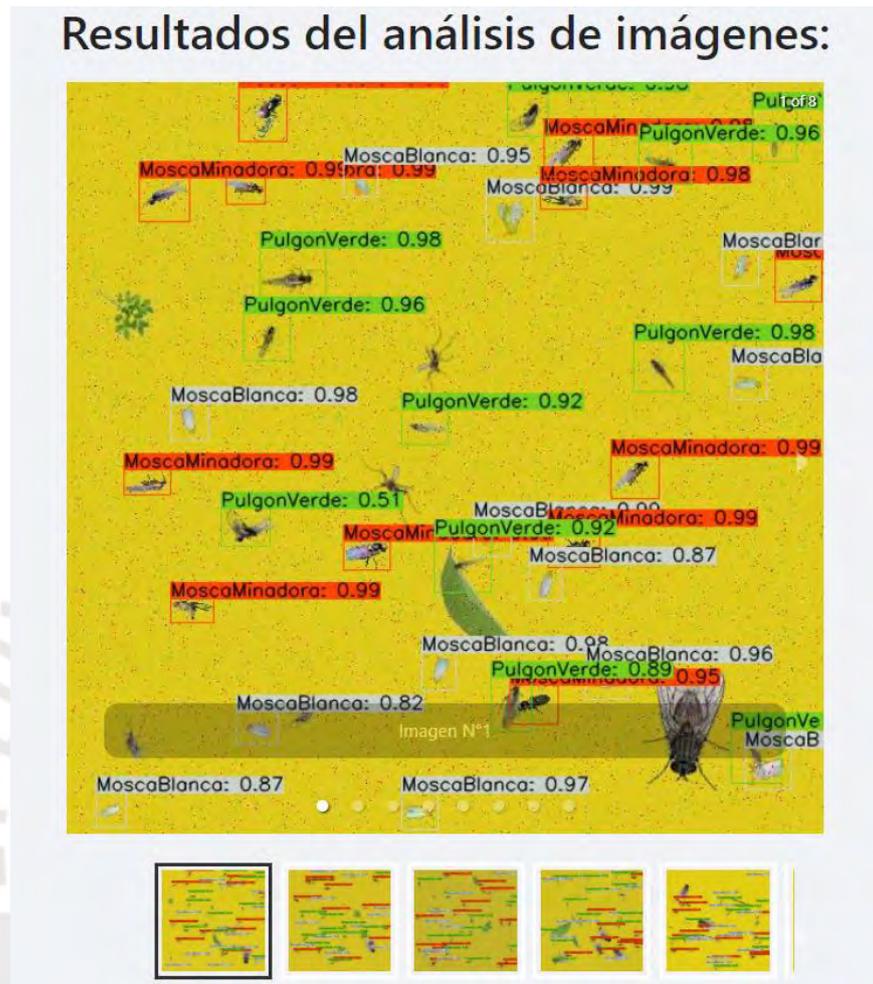


Figura 30: Identificación de los insectos en las imágenes subidas.



Figura 31: Reporte de resultados de la detección de insectos en las imágenes de trampas pegantes amarillas.

6.3 Discusión

El documento de requerimientos funcionales de la herramienta web fue planteado para poder saber qué es lo que el usuario final espera del software, por lo que se pensó en la simplicidad del uso, esto es que en la pantalla principal se muestren las funcionalidades de subida de archivos y de análisis de imágenes subidas, lo cual fue plasmado en la herramienta web, de forma tal que el usuario pueda obtener, en tan solo dos pasos, las imágenes segmentadas y anotadas de insectos plaga de relevancia en sus imágenes y un reporte de poblaciones de insectos.

Las pruebas unitarias y funcionales mencionadas en el resultado esperado N° 8 demuestran la efectividad de la integración backend y frontend. Los tiempos de ejecución presentados son referenciales, en el caso de las pruebas funcionales, se ponía al proceso a dormir por unos segundos entre cada click y envío de información. Finalmente, se podría realizar casos de prueba más complejos, por ejemplo, un test en el que se tenga una cantidad de insectos mínimos en cierta imagen subida para que la función de detección sea considerada como correcta y pase el test; sin embargo, este tipo de prueba un poco más específica no está siendo contemplada y se está tomando como incluida en el caso de prueba general.

Capítulo 7. Conclusiones y trabajos futuros

En el presente capítulo se mencionan las conclusiones del proyecto de fin de carrera y se propone ciertos trabajos futuros con la finalidad de dar propuestas de solución al problema central.

7.1 Conclusiones

En primer lugar, el presente proyecto plantea construir un generador de imágenes realistas de trampas pegantes con insectos plaga de hortalizas, pues se quiere abarcar un problema multiclase y en la actualidad solo se cuenta con data real de la mosca blanca, por lo que este resultado alcanzado se considera fundamental para los siguientes pasos de la investigación. Por otra parte, se planteó construir un modelo algorítmico para la identificación y clasificación de insectos plaga de hortalizas en imágenes de trampas pegantes, lo cual fue logrado al construir dos modelos de detección (Faster R-CNN y YOLOv4) y entrenar ambos modelos con la data generada. Después, se realizaron experimentos al entrenar distintos modelos con data real y artificial, donde se llegó a la conclusión de que la data artificial ayuda a la predicción en data real y esta aumenta significativamente si se entrena con ambos (data real y artificial). Finalmente, se planteó crear una herramienta que permita visualizar las detecciones de insectos plaga realizadas por el modelo en imágenes de trampas pegantes amarillas y que genere un reporte de las poblaciones de insectos plaga. Esto fue logrado mediante una plataforma web, en la que los distintos usuarios pueden subir sus imágenes para que sean analizadas por el detector y, posteriormente, genere gráficos de barras con la distribución de los insectos detectados por imagen y un gráfico circular para el total de ellos a manera de resumen. Con todo lo expuesto anteriormente se estaría resolviendo el problema central escogido, el cual es la carencia de una herramienta informática que permita automatizar el reconocimiento de poblaciones de insectos plaga de relevancia en plantíos de hortalizas en Perú a partir del ingreso de imágenes de trampas pegantes amarillas.

7.2 Trabajos futuros

A partir de lo desarrollado en el presente proyecto, se propone como trabajo futuro el entrenar un modelo Faster R-CNN con data real y artificial de la mosca blanca y comparar los resultados con los obtenidos por YOLOv4, pues sería interesante y relevante ver qué modelo aprende mejor si es entrenado con data real y generada para una clase. También, se propone recolectar más muestras del pulgón verde del melocotonero y de la mosca

minadora para generar mayor variedad al generar data artificial con la que se pueda entrenar un modelo de detección; además, se podría conseguir imágenes reales de estos insectos capturados en trampas pegantes y así entrenar el modelo con data generada y real. Adicionalmente, se podría entrenar un modelo Faster R-CNN con data artificial, similar a lo trabajado, pero con diferente backbone, pues existe la posibilidad de que ResNet50 no sea el más adecuado para obtener las características de los insectos. Finalmente, sería interesante la aplicación de modelos generativos adversariales (GAN) para la generación de imágenes artificiales de insectos, pues ello viene siendo realizado con relativo éxito en diferentes contextos y podría ser de utilidad en el problema abordado.



Referencias

About Selenium. (2013). <https://www.selenium.dev/about/>

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv:2004.10934 [cs, eess]*.
<http://arxiv.org/abs/2004.10934>

Bravo-Portocarrero, R., Zela, K., & Lima-Medina, I. (2020). Eficiencia de trampas pegantes de colores en la captura de insectos de hortalizas de hoja. *Scientia Agropecuaria*, *11*(1), 61–66. <https://doi.org/10.17268/sci.agropecu.2020.01.07>

Camborda, F., Castillo V., J., & Rodríguez Q., S. (2015). TRAMPAS DE LUZ CON PANEL PEGANTE PARA LA CAPTURA DE ADULTOS DE *Prodiplosis longifila* Gagné (Diptera: Cecidomyiidae) EN EL CULTIVO DE ESPÁRRAGO. *Ecología Aplicada*, *14*(1–2), 139. <https://doi.org/10.21704/rea.v14i1-2.90>

Cañedo, V., Alfaro-Tapia, A., & Kroschel, J. (2011). *Manejo Integrado de plagas de insectos en hortalizas Principios y referencias técnicas para la Sierra Central de Perú*.

Chen, J., Zhu, Q., Huang, M., & Guo, Y. (2017). *Research on recognition of *Empoasca flavescens* Based on machine vision*.
<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85035333847&doi=10.13031%2faim.201700460&partnerID=40&md5=f01146b821e41b2a94412063323fdb53>

- Chen, Jian, Fan, Y., Wang, T., Zhang, C., Qiu, Z., & He, Y. (2018). Automatic Segmentation and Counting of Aphid Nymphs on Leaves Using Convolutional Neural Networks. *Agronomy*, 8(8), 129. <https://doi.org/10.3390/agronomy8080129>
- Cho, J., Choi, J., Qiao, M., Ji, C. W., Kim, H. Y., Uhm, K. B., & Chon, T.-S. (2007). Automatic identification of whiteflies, aphids and thrips in greenhouse based on image analysis. *International Journal of Mathematics and Computers in Simulation*, 1, 46–53.
- Chulu, F., Phiri, J., Nkunika, P. O. Y., Nyirenda, M., Kabemba, M. M., & Sohati, P. H. (2019). *A convolutional neural network for automatic identification and classification of Fall Army Worm Moth*. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85070083253&partnerID=40&md5=b1520225e59c18c0682b0e57acd22cd8>
- Escartín, E., León, G., Arce, B., Winkler, R., Silvestre, A., Gómez, M., Rivera-Jacinto, M., Rodríguez Ulloa, C., Sobol, R., de los Santos-Villalobos, S., Vera-Nuñez, J., Peña Cabriales, J., Lopes Filho, J., Souza, Z., Lovo, I., Espínola, J., Montanez, A., Cardona, G., Davicino, R., & Cavallini, J. (2012). *RIESGOS MICROBIANOS EN LA PRODUCCIÓN DE ALIMENTOS FRESCOS EN ÁREAS URBANAS Y PERIURBANAS DE AMÉRICA LATINA Editado por*.
- Espinoza, K., Valera, D. L., Torres, J. A., López, A., & Molina-Aiz, F. D. (2016). *Combination of image processing and artificial neural networks as a novel approach for the identification of Bemisia tabaci and Frankliniella occidentalis on sticky traps in greenhouse agriculture*. <https://www.scopus.com/inward/record.uri?eid=2-s2.0->

84978880903&doi=10.1016%2fj.compag.2016.07.008&partnerID=40&md5=de202b4e356b02c88409df81fe3569c2

ICAR-NBAIR. (2013). *Insect Pests*.

<https://www.nbair.res.in/Databases/insectpests/index.php>

INIA. (2009). *Cultivo de hortalizas experiencias exitosas*.

https://www.minagri.gob.pe/portal/download/pdf/direccionesyoficinas/oficina_apoyo_enlace/cultivo_de_hortalizas_inia.pdf

Intel. (2020). *About*. <https://opencv.org/about/>

Kim, S., Park, S., Lim, S., & Kim, D. (2018). Classification Performance Analysis of Weight Update Method Applied to Various ConvNet Models. *2018 International Conference on Control and Robots (ICCR)*, 78–83.

<https://doi.org/10.1109/ICCR.2018.8534487>

LeCun, Y., & Bengio, Y. (1998). Convolutional networks for images, speech, and time series. En *The handbook of brain theory and neural networks* (pp. 255–258). MIT Press.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

<https://doi.org/10.1038/nature14539>

Li, Y., & Yang, J. (2020). Few-shot cotton pest recognition and terminal realization.

Computers and Electronics in Agriculture, *169*, 105240.

<https://doi.org/10.1016/j.compag.2020.105240>

Lim, S., Kim, S., & Kim, D. (2017). Performance effect analysis for insect classification using convolutional neural network. *2017 7th IEEE International Conference on*

Control System, Computing and Engineering (ICCSCE), 210–215.

<https://doi.org/10.1109/ICCSCE.2017.8284406>

Lin, T. (2019). *labelImg: LabelImg is a graphical image annotation tool and label object bounding boxes in images* (1.8.3) [Python]. <https://github.com/tzutalin/labelImg>

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

<https://doi.org/10.1017/CBO9780511809071>

Martineau, M., Conte, D., Raveaux, R., Arnault, I., Munier, D., & Venturini, G. (2017). A survey on image-based insect classification. *Pattern Recognition*, 65, 273–284.

<https://doi.org/10.1016/j.patcog.2016.12.020>

Michalski, R., Carbonell, J., & Mitchell, T. (1983). *Machine learning. An Artificial Intelligence approach. Volume 2: Vol. Vol. II*. <https://doi.org/10.1007/978-3-662-12405-5>

Nieuwenhuizen, A., Hemming, J., & Suh, H. K. (2018). *Detection and classification of insects on stick-traps in a tomato crop using Faster R-CNN*.

Nowozin, S. (2014). Optimal Decisions from Probabilistic Models: The Intersection-over-Union Case. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 548–555. <https://doi.org/10.1109/CVPR.2014.77>

Olson, D. L., & Delen, D. (2008). *Advanced data mining techniques* (pp. 1–180).

www.scopus.com

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A.,

- Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv:1506.01497 [cs]*.
<http://arxiv.org/abs/1506.01497>
- Reza, M. T., Mehedi, N., Tasneem, N. A., & Ashraful Alam, M. (2019). Identification of Crop Consuming Insect Pest from Visual Imagery Using Transfer Learning and Data Augmentation on Deep Neural Network. *2019 22nd International Conference on Computer and Information Technology (ICCIT)*, 1–6.
<https://doi.org/10.1109/ICCIT48885.2019.9038450>
- Romero-Lopez, A., Giro-i-Nieto, X., Burdick, J., & Marques, O. (2017). Skin Lesion Classification from Dermoscopic Images Using Deep Learning Techniques. *Biomedical Engineering*. Biomedical Engineering, Innsbruck, Austria.
<https://doi.org/10.2316/P.2017.852-053>
- Ronacher, A. (2010). *Foreword—Flask Documentation (1.1.x)*.
<https://flask.palletsprojects.com/en/1.1.x/foreword/#what-does-micro-mean>
- Rustia, D. J. A., Lin, C. E., Chung, J.-Y., Zhuang, Y.-J., Hsu, J.-C., & Lin, T.-T. (2020). *Application of an image and environmental sensor network for automated greenhouse insect pest monitoring*.
<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85075710698&doi=10.1016%2fj.aspen.2019.11.006&partnerID=40&md5=37dac3d571b63b35c693132217bd5627>

- S. El-Faki, M., Song, Y., Zhang, N., A. El-Shafie, H., Xin, P., 1. Date Palm Research Center of Excellence, King Faisal University, Al-Ahsa, Kingdom of Saudi Arabia, 2. Department of Biological and Agricultural Engineering, College of Engineering, Kansas State University, Manhattan, KS 66506, USA, & 3. College of Computer and Information Engineering, Inner Mongolia Agricultural University, Hohhot 010018, Inner Mongolia, China. (2018). Automated detection of parasitized *Cadra cautella* eggs by *Trichogramma bourarachae* using machine vision. *International Journal of Agricultural and Biological Engineering*, 11(3), 94–101.
<https://doi.org/10.25165/j.ijabe.20181103.2895>
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61, 85–117. <https://doi.org/10.1016/j.neunet.2014.09.003>
- Scikit-learn. (2011). 3.1. Cross-validation: Evaluating estimator performance—Scikit-learn 0.23.1 documentation. https://scikit-learn.org/stable/modules/cross_validation.html#cross-validation-and-model-selection
- SENASA. (2006). *Manual de procedimientos para la inspección fitosanitaria de predios seleccionados*.
https://www.senasa.gob.pe/senasa/descargasarchivos/2014/12/Manual_Procedimientos_Inspeccion_Fitosanitaria_Predios_Seleccionados.pdf
- SENASA. (2013). *Procedimiento para la detección del Huanglongbing de los cítricos y sus vectores Diaphorina citri y Trioza erytraea*.
https://www.senasa.gob.pe/senasa/descargasarchivos/2014/12/PRO-SARVF-05_-HLB_vigilancia-OCT-2013-firmado.pdf

- Shah, N., Osman, M., Othman, N., Ahmad, F., & Ahmad, A. R. (2019). Identification and Counting of Brown Planthopper in Paddy Field using Image Processing Techniques. *Procedia Computer Science, 163*, 580–590.
<https://doi.org/10.1016/j.procs.2019.12.140>
- Shen, Y., Zhou, H., Li, J., Jian, F., & Jayas, D. S. (2018). Detection of stored-grain insects using deep learning. *Computers and Electronics in Agriculture, 145*, 319–325.
<https://doi.org/10.1016/j.compag.2017.11.039>
- Shin, H.-C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., & Summers, R. M. (2016). Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. *IEEE Transactions on Medical Imaging, 35*(5), 1285–1298.
<https://doi.org/10.1109/TMI.2016.2528162>
- Souza, W. S. R., Alves, A. N., & Borges, D. L. (2019). A Deep Learning Model for Recognition of Pest Insects in Maize Plantations. *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2285–2290.
<https://doi.org/10.1109/SMC.2019.8914428>
- Team, K. (2015). *Keras documentation: Why choose Keras?* https://keras.io/why_keras/
- TensorFlow. (2015). TensorFlow. <https://www.tensorflow.org/?hl=es-419>
- Thenmozhi, K., & Srinivasulu Reddy, U. (2019). Crop pest classification based on deep convolutional neural network and transfer learning. *Computers and Electronics in Agriculture, 164*, 104906. <https://doi.org/10.1016/j.compag.2019.104906>

Unittest—Unit testing framework—Python 3.9.0 documentation. (2020).

<https://docs.python.org/3/library/unittest.html>

Walke, J. (2013). *React – A JavaScript library for building user interfaces.*

<https://reactjs.org/>

What is Python? Executive Summary. (2020). Python.org.

<https://www.python.org/doc/essays/blurb/>

Wu, L., Liu, Z., Bera, T., Ding, H., Langley, D. A., Jenkins-Barnes, A., Furlanello, C.,

Maggio, V., Tong, W., & Xu, J. (2019). A deep learning model to recognize food contaminating beetle species based on elytra fragments. *Computers and Electronics in Agriculture*, 166, 105002. <https://doi.org/10.1016/j.compag.2019.105002>

Xia, C., Chon, T.-S., Ren, Z., & Lee, J.-M. (2015). *Automatic identification and counting of small size pests in greenhouse conditions with low computational cost.*

<https://www.scopus.com/inward/record.uri?eid=2-s2.0-84940791674&doi=10.1016%2fj.ecoinf.2014.09.006&partnerID=40&md5=e87833d9533e0e9e5254a468728ed88a>

Xia, C., Lee, J.-M., Li, Y., Chung, B.-K., & Chon, T.-S. (2012). *In situ detection of small-size insect pests sampled on traps using multifractal analysis.*

<https://www.scopus.com/inward/record.uri?eid=2-s2.0-84871097944&doi=10.1117%2f1.OE.51.2.027001&partnerID=40&md5=9cf2eaff500b644dd57018f4ea71b223>

Zhong, Y., Gao, J., Lei, Q., & Zhou, Y. (2018). A Vision-Based Counting and Recognition System for Flying Insects in Intelligent Agriculture. *Sensors*, 18(5), 1489.

<https://doi.org/10.3390/s18051489>

Zhu, M. (2004). *Recall, precision and average precision*.

Zhu, S., Zhang, J., Lin, X., & Liu, D. (2019). Classification of rice planthoppers based on shape descriptors. *The Journal of Engineering*, 2019(22), 8378–8382.

<https://doi.org/10.1049/joe.2019.1085>





Anexos

Anexo 1: Estudios primarios

ID	Tipo	Año de publicación	Autor	Título
E01	journalArticle	2020	Rustia, D.J.A.; Lin, C.E.; Chung, J.-Y.; Zhuang, Y.-J.; Hsu, J.-C.; Lin, T.-T.	Application of an image and environmental sensor network for automated greenhouse insect pest monitoring
E02	conferencePaper	2017	Chen, J.; Zhu, Q.; Huang, M.; Guo, Y.	Research on recognition of <i>Empoasca flavescens</i> Based on machine vision
E03	journalArticle	2016	Espinoza, K.; Valera, D.L.; Torres, J.A.; López, A.; Molina-Aiz, F.D.	Combination of image processing and artificial neural networks as a novel approach for the identification of <i>Bemisia tabaci</i> and <i>Frankliniella occidentalis</i> on sticky traps in greenhouse agriculture
E04	journalArticle	2015	Xia, C.; Chon, T.-S.; Ren, Z.; Lee, J.-M.	Automatic identification and counting of small size pests in greenhouse conditions with low computational cost
E05	journalArticle	2012	Xia, C.; Lee, J.-M.; Li, Y.; Chung, B.-K.; Chon, T.-S.	In situ detection of small-size insect pests sampled on traps using multifractal analysis
E06	journalArticle	2019	Thenmozhi, K.; Srinivasulu Reddy, U.	Crop pest classification based on deep convolutional neural network and transfer learning
E07	journalArticle	2019	Zhu, Saihua; Zhang, Junyuan; Lin, Xiangze; Liu, Deying	Classification of rice planthoppers based on shape descriptors
E08	journalArticle	2019	Shah, Nur; Osman, Muhammad; Othman, Nor; Ahmad, Fadzil; Ahmad, Abdul Rahim	Identification and Counting of Brown Planthopper in Paddy Field using Image Processing Techniques
E09	journalArticle	2007	Cho, J.; Choi, J.; Qiao, M.; Ji, C.W.; Kim, H.Y.; Uhm, K.B.; Chon, Tae-Soo	Automatic identification of whiteflies, aphids and thrips in greenhouse based on image analysis
E10	journalArticle	2018	S. El-Faki, Mohammed; Song, Yuqi; Zhang, Naiqian; A. El-Shafie, Hamadttu; Xin, Pan; 1. Date Palm Research Center of Excellence, King Faisal University, Al-Ahsa, Kingdom of Saudi Arabia; 2. Department of Biological and Agricultural Engineering, College of Engineering, Kansas State University, Manhattan, KS 66506, USA; 3. College of Computer and Information Engineering, Inner Mongolia Agricultural University, Hohhot 010018, Inner Mongolia, China	Automated detection of parasitized <i>Cadra cautella</i> eggs by <i>Trichogramma bourarachae</i> using machine vision
E11	journalArticle	2018	Zhong, Yuanhong; Gao, Junyuan; Lei, Qilun; Zhou, Yao	A Vision-Based Counting and Recognition System for Flying Insects in Intelligent Agriculture
E12	journalArticle	2018	Chen, Jian; Fan, Yangyang; Wang, Tao; Zhang, Chu; Qiu, Zhengjun; He, Yong	Automatic Segmentation and Counting of Aphid Nymphs on Leaves Using Convolutional Neural Networks
E13	journalArticle	2019	Chulu, F.; Phiri, J.; Nkunika, P.O.Y.; Nyirenda, M.; Kabemba, M.M.; Sohati, P.H.	A convolutional neural network for automatic identification and classification of Fall Army Worm Moth
E14	journalArticle	2020	Li, Yang; Yang, Jiachen	Few-shot cotton pest recognition and terminal realization
E15	conferencePaper	2019	Reza, Md Tanzim; Mehedi, Nayeem; Tasneem, Nazifa Afroza; Ashraful Alam, Md	Identification of Crop Consuming Insect Pest from Visual Imagery Using Transfer Learning and Data Augmentation on Deep Neural Network
E16	journalArticle	2019	Wu, Leihong; Liu, Zhichao; Bera, Tanmay; Ding, Hongjian; Langley, Darryl A.; Jenkins-Barnes, Amy; Furlanello, Cesare; Maggio, Valerio; Tong, Weida; Xu, Joshua	A deep learning model to recognize food contaminating beetle species based on elytra fragments
E17	journalArticle	2018	Shen, Yufeng; Zhou, Huiling; Li, Jiangtao; Jian, Fujii; Jayas, Digvir S.	Detection of stored-grain insects using deep learning
E18	conferencePaper	2017	Lim, Suchang; Kim, Seunghyun; Kim, Doyeon	Performance effect analysis for insect classification using convolutional neural network
E19	conferencePaper	2019	Souza, Witenberg S. R.; Alves, Adao Nunes; Borges, Dibio Leandro	A Deep Learning Model for Recognition of Pest Insects in Maize Plantations
E20	conferencePaper	2018	Kim, Seunghyun; Park, Sungwook; Lim, Suchang; Kim, Doyeon	Classification Performance Analysis of Weight Update Method Applied to Various ConvNet Models
E21	journalArticle	2018	Nieuwenhuizen Ard; Hemming J.; Suh H.K.	Detection and classification of insects on stick-traps in a tomato crop using Faster R-CNN

Anexo 2: Formulario de extracción (P1)

Id	Fecha de extracción	Autores	Título	Año de publicación	Número de veces citado	Insecto(s)	Tipo de superficie	Problemas abordados	Dataset	Tipo de algoritmo usado	Detalle de algoritmo	Métricas y resultados
E01	1/05/2020	Rustia D.J.A., Lin C.E., Chung J.Y., Zhuang Y.-J., Hsu J.-C., Lin T.-T.	Application of an image and environmental sensor network for automated greenhouse insect pest monitoring	2020	0	No específico	Trampa pegante amarilla	Plaga de insectos que afecta la producción de tomate en Taiwan	Propio	RGB-to-LUV, static binary thresholding, morphological closing y blob detection	Conversión de color RGB a LUV, luego usa static binary thresholding para segmentar el color amarillo del fondo de la trampa pegante, después aplica morphological closing a la imagen y finalmente hace blob detection para separar los distintos objetos en la imagen	Accuracy: 93% aprox
E02	1/05/2020	Chen J., Zhu Q., Huang M., Guo Y.	Research on recognition of Empoasca flavescens Based on machine vision	2017	0	Empoasca flavescens	Trampa pegante amarilla	Plaga de insectos que afectan la producción de té	Propio	Superpixel segmentation SLIC y DBSCAN cluster	SLIC fue usado para segmentar la imagen en pequeños clusters y así separar el fondo de la imagen y los diferentes tipos de insectos. Luego, DBSCAN fue usado para juntar los superpíxeles, en caso este último no fuera muy pequeño	Accuracy: 98.81%
E03	1/05/2020	Espinoza K., Valera D.L., Torres J.A., López A., Molina-Aiz F.D.	Combination of image processing and artificial neural networks as a novel approach for the identification of Bemisia tabaci and Frankliniella occidentalis on sticky traps in greenhouse agriculture	2016	24	Whitefly (Bemisia tabaci) and thrip (Frankliniella occidentalis)	Trampa pegante amarilla y azul	Plaga de insectos (mosca blanca y trips de las flores) afecta producción agrícola	Propio	RGB-to-Lab, Otsu Method, RGB-to-HSV	Para trampa amarilla, conversión de color RGB a Lab, donde el a-channel corresponde al cuerpo y el b-channel a las alas. En ambos canales aplica filtro de contraste y con el Otsu Method hace binary thresholding y luego hace una operación de conjunción lógica. Finalmente hace una selección de objetos con mayor área. Para trampa azul es lo mismo solo que al inicio convierte de RGB a HSV	No menciona para segmentación de imagen
E04	1/05/2020	Xia C., Chon T.-S., Ren Z., Lee J.-M.	Automatic identification and counting of small size pests in greenhouse conditions with low computational cost	2015	22	Whitefly (Bemisia tabaci Genn), aphids (Aphis gossypii Glover) and thrips (Thrips tabaci L.)	Trampa pegante amarilla	Plaga de insectos afecta producción agrícola	Propio	RGB-to-YCbCr y Watershed algorithm	Conversión de color a YCbCr y luego aplicar el algoritmo watershed	Accuracy: 93% aprox F-score: max de 96%
E05	1/05/2020	Xia C., Lee J.-M., Li Y., Chung B.K., Chon T.-S.	In situ detection of small-size insect pests sampled on traps using multifractal analysis	2012	14	Whiteflies (Genus Bemisia)	Trampa pegante amarilla	Plaga de insectos afecta producción agrícola	Propio	Multifractal analysis	Aplicar multifractal análisis a la imagen y usar regional mínima para separar insectos del fondo	F-score: 96.5%
E06	1/05/2020	Thenmozhi, K., Srinivasulu Reddy, U.	Crop pest classification based on deep convolutional neural network and transfer learning	2019	7	104 tipos de insectos	Hojas	Plaga de insectos afecta producción agrícola	ImageNet	RGB-to-grayScale, Canny Edge Detection	Conversión de color RGB a escala de grises y luego aplicar Canny Edge Detection	No menciona para segmentación de imagen
E07	1/05/2020	Zhu, S., Zhang, J., Lin, X., & Liu, D.	Classification of rice planthoppers based on shape descriptors	2019	0	Rice planthopper	White screen	Plaga de insectos afecta producción de arroz	Propio	OTSU threshold segmentation	Usa OTSU threshold segmentation para hacer binary thresholding y luego usa un algoritmo para quedarse con el contorno del insecto	No menciona para segmentación de imagen
E08	1/05/2020	Jongman Cho, Junghyeon Choi, Mu Qiao, Chang-woo Ji, Hwang-young Kim, Ki-baik Uhm, and Tae-soo Chon	Automatic identification of whiteflies, aphids and thrips in greenhouse based on image analysis	2007	74	Whiteflies, aphids y thrips	Trampa pegante amarilla	Plaga de insectos afecta la producción agrícola	Propio	Binary thresholding y edge enhancement	Binary thresholding y edge enhancement con filtros de contraste y erosión	No menciona para segmentación de imagen
E09	1/05/2020	Nur Atiqah Nasser Shaha, Muhammad Khusairi Osmana, Nor Azlan Othmana, Fadzil Ahmada, Abdul Rahim Ahmada	Identification and Counting of Brown Planthopper in Paddy Field using Image Processing Techniques	2019	0	Brown Planthopper	Trampa pegante	Plaga de insectos afecta la producción agrícola	Propio	RGB-to-HSI y k-means clustering	Conversión de color RGB a HSI y luego hace segmentación de imágenes usando k-means con k=2 (fondo e insecto). Finalmente remueve el ruido haciendo image binarization y también remueve las pequeñas regiones	Precision: 97%aprox Recall: 96%aprox Accuracy: 98%aprox F-score: 96%aprox
E10	1/05/2020	El-Faki M., Song Y., Zhang N., El Shafie H., Xin, P.	Automated detection of parasitized Cadra cautella eggs by Trichogramma bourarachae using machine vision	2018	2	Huevos de Cadra cautella	No específico	Huevos de Cadra cautella afecta la producción agrícola	Propio	RGB-to-HSI y watershed delineation	Conversión de color RGB a HSI, luego remueve el fondo aplicando binary thresholding y finalmente usa el algoritmo watershed	Accuracy: 91.5%
E11	8/05/2020	Zhong, Yuanhong; Gao, Junyuan; Lei, Qilun; Zhou, Yao	A Vision-Based Counting and Recognition System for Flying Insects in Intelligent Agriculture	2018	18	7 tipos de insectos	Trampa pegante amarilla	Plaga de insectos afecta la producción agrícola	ImageNet	YOLO neural network	Usó el modelo YOLO para predecir la región de interés (bounding box)	Accuracy: 92.50%aprox
E21	12/07/2020	Nieuwenhuizen Ard; Hemming J.; Suh H.K.	Detection and classification of insects on stick-traps in a tomato crop using Faster R-CNN	2018	3	Whitefly, macrolophus y nesidocoris	Trampa pegante amarilla	Plaga de insectos afecta la producción de tomate	MS-COCO	Faster R-CNN	Usó el modelo Faster RCNN entrenado con el dataset MS-COCO y luego usó transfer learning para ajustar el modelo a las clases de insectos de interés	Accuracy: 87.4%aprox

Anexo 3: Formulario de extracción (P2 y P3)

Id	Fecha de extracción	Autores	Título	Año de publicación	Número de veces citado	Insecto(s)	Tipo de arquitectura usada	Dataset	Uso de transfer learning	Detalle del uso de transfer learning	Métricas y resultados
E06	1/05/2020	Thenmozhi, K.; Srinivasulu Reddy, U.	Crop pest classification based on deep convolutional neural network and transfer learning	2019	7	24 tipos de insectos	6 convolutional layers, 5 max pooling layers, 1 fully connected layer y output layer con Softmax	ImageNet	Sí	Probó con modelos AlexNet, ResNet, GoogLeNet y VGGNet. Sin embargo, estos no fueron parte del modelo final	Accuracy: 97.47%
E12	1/05/2020	Chen, Jian; Fan, Yangyang; Wang, Tao; Zhang, Chu; Qiu, Zhengjun; He, Yong	Automatic Segmentation and Counting of Aphid Nymphs on Leaves Using Convolutional Neural Networks	2018	0	Aphid nymphs	Arquitectura de VGG-13	Propio	Sí	Usó VGG-13 network y le cambió los pesos del último layer	Dice score: 82.07% Precision: 95.63% Recall: 96.50% F-Score: 96.06%
E13	1/05/2020	Chulu, F.; Phiri, J.; Nkunika, P.O.Y.; Nyirenda, M.; Kabemba, M.M.; Sohati, P.H.	A convolutional neural network for automatic identification and classification of Fall Army Worm Moth	2019	0	Fall Army Worm Moth	Arquitectura del InceptionV3	Propio	Sí	Usó InceptionV3 y cambió el último softmax layer	Accuracy: 82% aprox
E14	1/05/2020	Li, Yang; Yang, Jiachen	Few-shot cotton pest recognition and terminal realization	2020	0	50 tipos de insectos	2 convolutional layers, 2 max pooling layers, 2 dense layers	NBAIR	No	No usó	Accuracy: 95.4%
E15	1/05/2020	Reza, Md Tanzim; Mehedi, Nayeem; Tasneem, Nazifa Afroza; Ashrafu Alam, Md	Identification of Crop Consuming Insect Pest from Visual Imagery Using Transfer Learning and Data Augmentation on Deep Neural Network	2019	0	102 tipos de insectos	Arquitectura del InceptionV3	IP102	Sí	Usó VGG19, InceptionV3 y ResNet50 con data augmentation.	Accuracy: 57.08% por parte de InceptionV3
E16	1/05/2020	Wu, Leihong; Liu, Zhichao; Bera, Tanmay; Ding, Hongjian; Langley, Darryl A.; Jenkins-Barnes, Amy; Furlanello, Cesare; Maggio, Valerio; Tong, Weida; Xu, Joshua	A deep learning model to recognize food contaminating beetle species based on elytra fragments	2019	0	15 tipos de insectos	Arquitectura VGG16 + 4 fully connected layers y 1 activation layer	ImageNet	Sí	Usó pretrained VGG16 y le agregó 4 fully connected layers y 1 activation layer	Accuracy: 83.8%
E17	1/05/2020	Shen, Yufeng; Zhou, Huiling; Li, Jiangtao; Jian, Fuji; Jayas, Digvir S.	Detection of stored-grain insects using deep learning	2018	19	6 tipos de insectos	Arquitectura "inception"	Propio	No	No usó	mAP: 87.99%
E18	1/05/2020	Lim, Suchang; Kim, Seunghyun; Kim, Doyeon	Performance effect analysis for insect classification using convolutional neural network	2017	3	27 tipos de insectos	Arquitectura de AlexNet	ImageNet	Sí	Aumentó el número de convolution kernels	Accuracy: 81.82%
E19	1/05/2020	Souza, Witenberg S. R.; Alves, Adao Nunes; Borges, Dibio Leandro	A Deep Learning Model for Recognition of Pest Insects in Maize Plantations	2019	0	6 tipos de insectos	Arquitectura de Inception-V3 + Batch normalization y Dropouts	ImageNet	Sí	Usó Inception-V3 y le agregó batch normalization y dropouts con diferentes tasas para mejorar el accuracy	Accuracy: 97%
E20	1/05/2020	Kim, Seunghyun; Park, Sungwook; Lim, Suchang; Kim, Doyeon	Classification Performance Analysis of Weight Update Method Applied to Various ConvNet Models	2018	0	27 tipos de insectos	Arquitectura de ResNet-50	ImageNet	Sí	Usó ResNet-50 y AdaDelta como método para actualizar los weights	Accuracy: 89.98% aprox
E21	12/07/2020	Nieuwenhuizen Ard; Hemming J.; Suh H.K.	Detection and classification of insects on stick-traps in a tomato crop using Faster R-CNN	2018	3	Whitefly, macrolophus y nesidiocoris	Faster RCNN	MS-COCO	Sí	Usó el modelo Faster RCNN entrenado con el dataset MS-COCO y luego usó transfer learning para ajustar el modelo a las clases de insectos de interés	Accuracy: 87.4%aprox

Anexo 4: Plan de Proyecto

- **Justificación**

Según lo expuesto en este proyecto de fin de carrera, existe un sobreesfuerzo del personal de vigilancia en campos agrícolas al momento de detectar plagas de insectos, por lo que el proyecto busca implementar una herramienta informática que permita que estas personas puedan hacer su trabajo de control y vigilancia de los campos de hortalizas de una forma más eficiente. Esta herramienta busca solucionar el problema haciendo uso del aprendizaje profundo en imágenes con un modelo algorítmico de detección y clasificación de insectos, pues en la actualidad no se tienen modelos para clasificar a los insectos de relevancia que aparecen en los plantíos de hortalizas en Perú.

- **Viabilidad**

- ✓ **Viabilidad temporal:** En el cronograma de actividades (tabla 13) se verifica la viabilidad temporal del proyecto, pues se planea terminar con este en el tiempo establecido por el curso.
- ✓ **Viabilidad técnica:** Las herramientas y lenguajes de programación a utilizar en el desarrollo del proyecto de fin de carrera son de acceso libre. Además, se cuenta con el apoyo del grupo de Inteligencia Artificial (IA-PUCP) para el uso de una máquina con un GPU potente.
- ✓ **Viabilidad económica:** El proyecto es económicamente viable por lo siguiente: Los costos asociados a las horas del tesista serán asumidos por el mismo. Las horas de asesoría son asumidas por el servicio educativo de la universidad. Las horas del uso de servidor con GPU para entrenamiento de modelos será donación del grupo de Inteligencia Artificial (IA-PUCP), sujeto a disponibilidad del servidor.

- **Alcance**

Este proyecto de fin de carrera pertenece al área de ciencias de la computación. En él, se propone desarrollar pre-procesamiento de imágenes y algoritmos de segmentación y clasificación para imágenes de insectos agrícolas en trampas pegantes.

El modelo algorítmico de detección y clasificación será integrado a una herramienta web con interfaz simple que permita al usuario subir una imagen de una trampa pegante con insectos y elaborar, a partir de esa imagen, un reporte en el que se indique la cantidad de insectos identificados por especie, la imagen de la trampa segmentada y anotada indicando la probabilidad con la que se clasificó cada insecto.

Se considerará a tres especies de insectos para el modelo de clasificación, los cuales son la mosca blanca (*Bemisia Tabaci*), la mosca minadora (*Liriomyza Huidobrensis*) y el pulgón verde del melocotonero en etapa de adulto alado (*Myzus Persicae*), pues estos son de los insectos que más daño ocasionan a los campos de hortalizas en Perú (Cañedo et al., 2011). Finalmente, solo se considerará imágenes de trampas pegantes amarillas con iluminación de día como entrada para el modelo algorítmico.

- **Restricciones**

Para la realización de este proyecto se ha identificado una restricción referente al poder computacional que se tiene disponible en la universidad para la creación de los modelos, pues estos requieren de un hardware especializado para que el tiempo de entrenamiento sea lo más corto posible.

- **Identificación de los riesgos del proyecto**

Para el trabajo de investigación, se definió una escala de probabilidad e impacto que va del 0 al 1, donde 0 es muy bajo y 1 es muy alto. La severidad se calculó como el producto de la probabilidad y el impacto. En la tabla 14 se muestran los riesgos identificados en el proyecto.

Tabla 14: Mapeo de riesgos del proyecto.

Descripción	Síntomas	Probabilidad	Impacto	Severidad	Mitigación	Contingencia
El corpus de imágenes de insectos en trampas pegantes no es suficiente para lograr el desempeño planteado.	Bajos resultados en la métrica de evaluación.	0.3	0.8	0.24	Se usará un modelo ya entrenado en objetos similares.	Recopilar más imágenes de los insectos requeridos.
El costo computacional que requiere el entrenamiento de los modelos es mayor al que se posee.	Demora en la etapa de entrenamiento de los modelos.	0.5	0.7	0.35	Usar recursos de la nube.	Pedir acceso a IA-PUCP para usar sus máquinas Lambda con tarjetas gráficas GPU.

- **Estructura de descomposición del trabajo (EDT)**

En la figura 32 se muestra el EDT del proyecto de investigación.

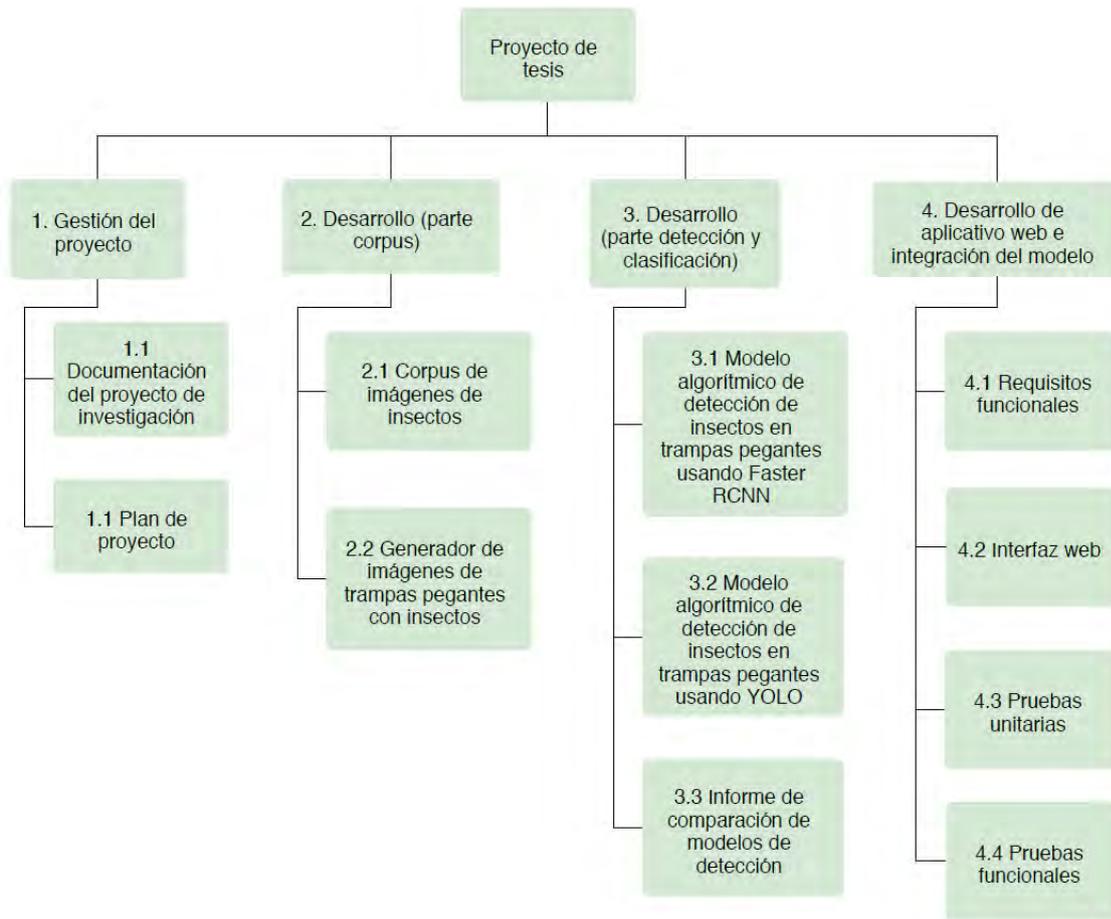


Figura 32: Diagrama de estructura de descomposición del trabajo (EDT).

● **Lista de tareas**

En la tabla 15 se muestra la lista de tareas del proyecto de investigación, las cuales abarcan la planificación, el desarrollo y las pruebas del proyecto.

Tabla 15: Lista de tareas.

Tarea		Duración estimada (días)	Esfuerzo asociado (horas-persona)	Costo estimado
Gestión del proyecto y documentación.				
1.1	Investigación de algoritmos usados en problemas de detección y clasificación de insectos.	24	60	600
1.2	Investigación de métricas usadas en problemas de detección y clasificación.	22	55	550

1.3	Investigación sobre la problemática del proyecto de tesis.	9	22.5	225
1.4	Definición del objetivo general, objetivos y resultados específicos.	9	22.5	225
1.5	Análisis de viabilidad del proyecto.	10	25	250
1.6	Corrección de observaciones realizadas al documento de investigación.	2	5	50
O1. Construir un generador de imágenes realistas de trampas pegantes con insectos plaga de hortalizas				
2.1	Elaboración de la lista de requerimientos del generador de imágenes realistas.	4	10	100
2.2	Obtención de imágenes base de insectos.	4	10	100
2.3	Desarrollo del código para hacer data augmentation a las imágenes base de insectos.	4	10	100
2.4	Creación del generador de imágenes de trampas pegantes con insectos.	11	27.5	275
O2. Construir un modelo algorítmico para la identificación y clasificación de insectos plaga de hortalizas en imágenes de trampas pegantes.				
3.1	Creación del corpus usando el generador de imágenes realistas de trampas pegantes con insectos.	4	10	100
3.2	Entrenar el detector de objetos Faster R-CNN con el corpus generado.	11	27.5	275
3.3	Entrenar el detector de objetos YOLO con el corpus generado.	11	27.5	275
3.4	Elaboración del reporte de comparación de modelos algorítmicos de clasificación de insectos.	4	10	100
O3. Crear una herramienta que permita visualizar la identificación, clasificación, conteo y generación de reportes de poblaciones de plagas de insectos de relevancia en plantíos de hortalizas en Perú a partir del ingreso de fotografías de trampas pegantes.				
4.1	Elaboración del documento de requerimientos funcionales de la herramienta web.	4	10	100
4.2	Elaboración de la interfaz web que integre los modelos de identificación y clasificación.	11	27.5	275
4.3	Elaboración de las pruebas unitarias y funcionales para la interfaz web.	11	27.5	275
4.4	Elaboración y presentación del informe final con resultados y lecciones aprendidas.	39	97.5	975

- **Cronograma del proyecto**

En la tabla 16 se muestra el cronograma del proyecto de investigación.

Tabla 16: Cronograma del proyecto

Tarea	Fecha de inicio	Fecha de fin	Sem 01	Sem 02	Sem 03	Sem 04	Sem 05	Sem 06	Sem 07	Sem 08	Sem 09	Sem 10	Sem 11	Sem 12	Sem 13	Sem 14	Sem 15	Sem 16	Sem 17	Sem 18	Sem 19	
Gestión del proyecto y documentación.																						
1.1	Investigación de algoritmos usados en problemas de detección y clasificación de insectos.	13/04/2020	6/05/2020																			
1.2	Investigación de métricas usadas en problemas de detección y clasificación.	28/04/2020	19/05/2020																			
1.3	Investigación sobre la problemática del proyecto de tesis.	20/05/2020	28/05/2020																			
1.4	Definición del objetivo general, objetivos y resultados específicos.	1/06/2020	9/06/2020																			
1.5	Análisis de viabilidad del proyecto.	28/06/2020	7/07/2020																			
1.6	Corrección de observaciones realizadas al documento de investigación.	11/07/2020	12/07/2020																			
O1. Construir un generador de imágenes realistas de trampas pegantes con insectos plaga de hortalizas.																						
2.1	Elaboración de la lista de requerimientos del generador de imágenes realistas.	31/08/2020	3/09/2020																			
2.2	Obtención de imágenes base de insectos.	31/08/2020	3/09/2020																			
2.3	Desarrollo del código para hacer data augmentation a las imágenes base de insectos.	31/08/2020	3/09/2020																			
2.4	Creación del generador de imágenes de trampas pegantes con insectos.	7/09/2020	17/09/2020																			
O2. Construir un modelo algorítmico para la identificación y clasificación de insectos plaga de hortalizas en imágenes de trampas pegantes.																						
3.1	Creación del corpus usando el generador de imágenes realistas de trampas pegantes con insectos.	14/09/2020	17/09/2020																			
3.2	Entrenar el detector de objetos Faster R-CNN con el corpus generado.	21/09/2020	1/10/2020																			
3.3	Entrenar el detector de objetos YOLO con el corpus generado.	28/09/2020	8/10/2020																			
3.4	Elaboración del reporte de comparación de modelos algorítmicos de clasificación de insectos.	12/10/2020	15/10/2020																			
O3. Crear una herramienta que permita visualizar la identificación, clasificación, conteo y generación de reportes de poblaciones de plagas de insectos de relevancia en plantíos de hortalizas en Perú a partir del ingreso de fotografías de trampas pegantes.																						
4.1	Elaboración del documento de requerimientos funcionales de la herramienta web.	31/08/2020	3/09/2020																			
4.2	Elaboración de la interfaz web que integre los modelos de identificación y clasificación.	19/10/2020	29/10/2020																			
4.3	Elaboración de las pruebas unitarias y funcionales para la interfaz web.	2/11/2020	12/11/2020																			
4.4	Elaboración y presentación del informe final con resultados y lecciones aprendidas.	2/11/2020	10/12/2020																			

Tarea	Fecha de inicio	Fecha de fin	Sem 20	Sem 21	Sem 22	Sem 23	Sem 24	Sem 25	Sem 26	Sem 27	Sem 28	Sem 29	Sem 30	Sem 31	Sem 32	Sem 33	Sem 34	Sem 35	Sem 36
Gestión del proyecto y documentación.																			
1.1	Investigación de algoritmos usados en problemas de detección y clasificación de insectos.	13/04/2020	6/05/2020																
1.2	Investigación de métricas usadas en problemas de detección y clasificación.	28/04/2020	19/05/2020																
1.3	Investigación sobre la problemática del proyecto de tesis.	20/05/2020	28/05/2020																
1.4	Definición del objetivo general, objetivos y resultados específicos.	1/06/2020	9/06/2020																
1.5	Análisis de viabilidad del proyecto.	28/06/2020	7/07/2020																
1.6	Corrección de observaciones realizadas al documento de investigación.	11/07/2020	12/07/2020																
O1. Construir un generador de imágenes realistas de trampas pegantes con insectos plaga de hortalizas.																			
2.1	Elaboración de la lista de requerimientos del generador de imágenes realistas.	31/08/2020	3/09/2020																
2.2	Oblención de imágenes base de insectos.	31/08/2020	3/09/2020																
2.3	Desarrollo del código para hacer data augmentation a las imágenes base de insectos.	31/08/2020	3/09/2020																
2.4	Creación del generador de imágenes de trampas pegantes con insectos.	7/09/2020	17/09/2020																
O2. Construir un modelo algorítmico para la identificación y clasificación de insectos plaga de hortalizas en imágenes de trampas pegantes.																			
3.1	Creación del corpus usando el generador de imágenes realistas de trampas pegantes con insectos.	14/09/2020	17/09/2020																
3.2	Entrenar el detector de objetos Faster R-CNN con el corpus generado.	21/09/2020	1/10/2020																
3.3	Entrenar el detector de objetos YOLO con el corpus generado.	28/09/2020	8/10/2020																
3.4	Elaboración del reporte de comparación de modelos algorítmicos de clasificación de insectos.	12/10/2020	15/10/2020																
O3. Crear una herramienta que permita visualizar la identificación, clasificación, conteo y generación de reportes de poblaciones de plagas de insectos de relevancia en plantíos de hortalizas en Perú a partir del ingreso de fotografías de trampas pegantes.																			
4.1	Elaboración del documento de requerimientos funcionales de la herramienta web.	31/08/2020	3/09/2020																
4.2	Elaboración de la interfaz web que integre los modelos de identificación y clasificación.	19/10/2020	29/10/2020																
4.3	Elaboración de las pruebas unitarias y funcionales para la interfaz web.	2/11/2020	12/11/2020																
4.4	Elaboración y presentación del informe final con resultados y lecciones aprendidas.	2/11/2020	10/12/2020																

- **Lista de recursos**

- **Personas involucradas y necesidades de capacitación**

En el proyecto de investigación se tiene una persona involucrada, que es el entomólogo que verificará las imágenes de insectos para la creación de los corpus de imágenes.

- **Materiales requeridos para el proyecto**

Se necesitará de computadoras, mesas, sillas y materiales de escritorio para la realización del proyecto.

- **Equipamiento requerido**

Se necesitará de computadoras con GPU para el proyecto de investigación.

- **Herramientas requeridas**

Para el proyecto de investigación se hará uso de las siguientes herramientas: Python, Keras, TensorFlow, Scikit-learn, Labellmg, OpenCV, Flask, React Js, Selenium, Faster R-CNN y YOLOv4.

- **Costeo del Proyecto**

En la tabla 17 se muestra el costeo del proyecto.

Tabla 17: Costeo del Proyecto.

Ítem	Descripción	Unidad	Cantidad	Valor Unidad (S/.)	Monto Parcial (S/.)	Monto Total (S/.)
0	Costo total del proyecto	---	---	---	---	7,740
1.	Tesista Claudia Córdova	---	---	---	---	930
1.1	Análisis	Horas	10	10	100	
1.2	Capacitación	Horas	20	10	200	
1.3	Documentación	Horas	25	10	250	
1.4	Diseño	Horas	8	10	80	
1.5	Implementación	Horas	25	10	250	
1.6	Pruebas	Horas	5	10	50	
2.	Otros participantes	---	---	---	---	750
2.1	Asesor Edwin Villanueva	Horas	20	30	600	
2.2	Entomólogo	Horas	5	30	150	
3.	Servicios y consultoría	---	---	---	---	---
4.	Materiales e insumos	---	---	---	---	3,110
4.1	Energía	Valor			1,440	
4.2	Internet	Valor			1,170	
4.3	Materiales de escritorio	Valor			500	
5.	Bienes y equipos	---	---	---	---	2,950
5.1	Computadora	Valor			2,500	
5.2	Mesas	Valor			250	
5.3	Sillas	Valor			200	
6.	Pasajes y viáticos	---	---	---	---	---