IDENTIFICACIÓN Y CONTEO DE INSECTOS (MOSCA BLANCA, TRIPS Y MINADOR DE HOJA) CAPTURADOS EN TRAMPA ADHESIVA MEDIANTE VISIÓN E INTELIGENCIA ARTIFICIAL. CASO FLORES EL TRIGAL

Modalidad: Exploratorio

NATALIA LÓPEZ GRISALES

Trabajo de grado para optar al título de Ingeniera Mecatrónica

Director
Alejandro Martínez Osorio, MEng

UNIVERSIDAD EIA
INGENIERÍA MECATRÓNICA
ENVIGADO
2020
La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
LISTA DE TABLAS

Tabla 1 Información resumen de artículo antecedente 1 .................................................. 4
Tabla 2 Información resumen de artículo antecedente 2 .................................................. 5
Tabla 3 Información resumen de artículo antecedente 3 .................................................. 5
Tabla 4 Información resumen de artículo antecedente 4 .................................................. 6
Tabla 5 Información resumen de artículo antecedente 5 .................................................. 6
Tabla 6 Información resumen de artículo antecedente 6 .................................................. 7
Tabla 7 Requerimientos de usuario .................................................................................. 26
Tabla 8 Requerimientos técnicos ..................................................................................... 27
Tabla 9 Matriz morfológica ............................................................................................... 31
Tabla 10 Descripción de conceptos propuestos en la matriz morfológica ...................... 32
Tabla 11 Parámetros modelos YOLO ................................................................................. 41
Tabla 12 Métricas de desempeño de modelos de inteligencia artificial .......................... 44
Tabla 13 Cotización desarrollo de aplicativo ................................................................... 47
Tabla 14 Cumplimiento de los requerimientos de usuario ............................................. 48
Tabla 15 Pautas generales para diseñar un experimento ................................................. 51

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
LISTA DE FIGURAS

Figura 1. Trips, Frankliniella bispinosa................................................................. 9
Figura 2. Daño causado por presencia de Trips .......................................................... 9
Figura 3. Mosca blanca, Aleurotrachelus trachoides..................................................... 10
Figura 4. Minador de hoja, Liriomyza trifolii............................................................. 11
Figura 5. Daño causado por presencia de Minador de hoja ........................................... 11
Figura 6. Secuencia de pasos del procesamiento digital de imágenes.............................. 12
Figura 7. Procedimiento para detección de objetos ..................................................... 22
Figura 8. Arquitectura de la red neuronal usada por YOLO ......................................... 22
Figura 9. Arquitectura funcional del proyecto ............................................................ 28
Figura 10. Descomposición funcional de la aplicación móvil ....................................... 29
Figura 11. Diagrama de Procesamiento Digital de Imágenes. ...................................... 33
Figura 12. Selección de color para el procesamiento digital de imágenes........................ 34
Figura 13. Procesamiento digital hasta efecto “Zoom” .................................................. 35
Figura 14. Procesamiento digital hasta identificación de insectos .................................... 36
Figura 15. Resultados generales del modelo creado en Custom Vision.......................... 37
Figura 16. Resultados por etiqueta del modelo creado en Custom Vision ....................... 38
Figura 17. Interfaz de usuario Custom Vision, sección de prueba ................................. 38
Figura 18. Esquema de procedimiento para entrenamiento YOLO ................................ 39
Figura 19. Interfaz de etiquetado de imágenes mediante Python ................................... 40
Figura 20. Prueba de modelos YOLO en la misma imagen .......................................... 42

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
RESUMEN

En los cultivos de flores, la presencia de plagas puede generar grandes pérdidas y afectar la producción de las empresas, por esto, es fundamental realizar un monitoreo constante en busca de insectos, enfermedades o daños en las plantas que indiquen la presencia de plagas. Esta es una labor que se torna complicada debido a que el comportamiento de algunos insectos puede ser más activo en la noche o en la ausencia de los operarios.

Actualmente, para contrarrestar esta problemática, se implementa el uso de trampas adhesivas como método de monitoreo indirecto, con las cuales se capturan insectos para luego analizar el crecimiento de su población. La información recolectada en la lectura de las trampas es clave para la toma de decisiones, pero la magnitud de los cultivos y el tamaño de los insectos hacen que sea una tarea que demanda tiempo, esfuerzo y conocimiento especializado. Por esto, con el desarrollo de este proyecto se busca automatizar el proceso, mediante la identificación automática de tres clases de insectos: Minador de hoja, Trips y Mosca Blanca, a través de una aplicación móvil, dando alivio a la persona encargada de realizar esta actividad.

En este documento se presenta un estudio de diferentes métodos computacionales para la clasificación y conteo de insectos, mediante la metodología Design Thinking de Herbert Simon (1969), la cual propone 7 etapas: definición, investigación, ideación, prototipado, selección, implementación y aprendizaje. Esta metodología puede ejecutarse de forma no lineal e iterativa.

Los métodos computacionales estudiados son: Procesamiento Digital de Imágenes, Machine Learning a través de la plataforma Custom Vision de Microsoft Azure y Deep Learning con el modelo de detección de objetos YOLO (You Only Look Once). Luego de diseñar y probar un prototipo por cada método, se elige el modelo YOLO para su implementación, ya que obtuvo el mejor desempeño en la clasificación de los insectos.

El modelo seleccionado se implementa en una aplicación móvil desarrollada en el software Android Studio, con la cual es posible clasificar y cuantificar los insectos capturados en la trampa en tiempo real, lo que reduce el tiempo que toma realizar el monitoreo manual, de 1 minuto por placa aproximadamente a un par de segundos que le toma a la persona enfocar la trampa con el dispositivo móvil; suprime la necesidad de capacitar a las personas en identificación de plagas y ofrece a la empresa la oportunidad de almacenar la información digitalmente, dando un paso importante en la transformación digital, la cultura del dato y el uso de la información.

Además, se presenta la cotización de una empresa de software para el desarrollo del aplicativo, con la cual se calcula el retorno de la inversión del proyecto, basándose en las horas ahorradas con la automatización.

Palabras clave: Android Studio, Machine Learning, Deep Learning.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
ABSTRACT

In flower crops, the presence of pests can generate great losses and affect the production of the companies. For this reason, it is essential to carry out constant monitoring in search of insects, diseases or damage to plants that indicate the presence of pests. This is a task that becomes complicated because the behavior of some insects can be more active at night or in the absence of operators.

Currently, to counteract this problem, the use of adhesive traps is implemented as a method of indirect monitoring, with which insects are captured and then analyzed for population growth. The information collected in the reading of the traps is key for decision making, but the magnitude of the crops and the size of the insects make it a task that demands time, effort and specialized knowledge. For this reason, with the development of this project we seek to automate the process, through the automatic identification of three classes of insects: Leaf miner, Trips and White fly, through a mobile application, giving relief to the person in charge of this activity.

This document presents a study of different computational methods for the classification and counting of insects, through the Design Thinking methodology by Herbert Simon (1969), which proposes 7 stages: definition, research, ideation, prototyping, selection, implementation and learning. This methodology can be executed in a non-linear and iterative way.

The computational methods studied are: Digital Image Processing, Machine Learning through Microsoft Azure's Custom Vision platform and Deep Learning with the YOLO (You Only Look Once) object detection model. After designing and testing a prototype for each method, the YOLO model is chosen for implementation, since it obtained the best performance in the classification of insects.

The selected model is implemented in a mobile application developed in the Android Studio software, with which it is possible to classify and quantify the insects captured in the trap in real time, which reduces the time it takes to perform manual monitoring, from 1 minute per plate approximately to a couple of seconds that it takes the person to focus the trap with the mobile device; it eliminates the need to train people in pest identification and offers the company the opportunity to store the information digitally, taking an important step in the digital transformation, data culture and use of information.

In addition, it presents the quotation of a software company for the development of the application, with which the return on investment of the project is calculated, based on the hours saved with the automation.

Keywords: Android Studio, machine learning, deep learning.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
INTRODUCCIÓN

El gremio floricultor en Colombia genera 140.000 empleos en 60 municipios del país, de los cuales el 65% son ocupados por madres cabeza de hogar (ProColombia, 2019). En el 2018 este sector alcanzó los US$ 1.460 millones en exportaciones de flor, lo que representa el 3.5% de las exportaciones totales del país y el 9.6% de las No Minero Energéticas (Mincomercio, 2019).

Para mantener la productividad y los indicadores de desempeño en un alto nivel es fundamental velar por la salud de los cultivos, la cual puede verse afectada por la presencia de plagas o enfermedades.

Actualmente, en los cultivos de flores de la empresa Flores El Trigal, se realiza un Manejo Integrado de Plagas y Enfermedades (MIPE) que permite mantener controlada la incidencia de problemas fitosanitarios. Una de las técnicas utilizadas es el monitoreo indirecto mediante trampas adhesivas, en las cuales se capturan insectos y luego son inspeccionadas por un operario capacitado en la identificación de estos.

La labor de lectura de trampas demanda una gran cantidad de tiempo y esfuerzo visual por parte del operario, ya que en el cultivo puede haber entre 834 y 1666 trampas instaladas y por cada una se tarda aproximadamente un minuto en su lectura según la cantidad de insectos atrapados. Por esto, se plantea realizar el conteo y clasificación de los insectos por medio de métodos computacionales, con la intención de lograr que un operario sin capacitación en monitoreo de plagas pueda realizar la labor de manera eficiente y eficaz, además de disminuir significativamente el tiempo de lectura de las trampas y de tener un registro histórico y digital de la información para su posterior análisis.

Los métodos computacionales explorados en el desarrollo del proyecto inician con Visión Artificial mediante el procesamiento digital de imágenes, seguido de Machine Learning a través de la plataforma Custom Vision de Microsoft Azure y finalmente Deep Learning con el sistema de detección de objetos en tiempo real YOLO (You Only Look Once).

Este trabajo se desarrolló bajo la metodología Design Thinking de Herbert Simon (1969), la cual plantea las etapas de definición, investigación, ideación, prototipado, selección, implementación y aprendizaje como la ruta para alcanzar los objetivos establecidos.
1. PRELIMINARES

1.1 PLANTEAMIENTO DEL PROBLEMA

El sector floricultor es hoy en día el más sobresaliente del gremio agrícola en Colombia. Su aporte en generación de empleos, exportaciones y desarrollo rural lo consolidan como un sector estratégico para la economía del país (Villamizar et al., 2017).

El cambio climático y la reducción de personal o mano de obra en los cultivos son algunos de los grandes retos que enfrenta la floricultura colombiana (Syngenta, 2017). Según Carlos Andrés Plazas, agrónomo que trabaja con Syngenta, el cambio climático afecta directamente la producción de flores ya que genera problemas fitosanitarios y de calidad, pues cuando hace mucho frío las flores no germinan y cuando hace mucho calor los botones de las flores no se desarrollan adecuadamente. Adicionalmente, Plazas afirma que, bajo estas condiciones climáticas, los cultivos se ven expuestos a blancos biológicos, plagas y enfermedades que damnifican la producción (Syngenta, 2017).

Las plagas presentes en las plantas, además de disminuir la productividad de los cultivos, pueden llegar a generar daños que impacten económicamente las empresas floricultoras, por esto, se recomienda adoptar estrategias como el monitoreo directo o indirecto (County of Santa Clara, 2020). La información obtenida durante el monitoreo de cultivos ayuda a determinar si la población del insecto se encuentra por encima o debajo del umbral económico, el cual indica el punto a partir del cual la plaga comienza a ser perjudicial para el cultivo. Además, al conocer el tipo de insectos presentes y su evolución en el tiempo, es posible aplicar técnicas de control más específicas para lograr una intervención más eficiente, reduciendo el impacto en el medio ambiente y en el cultivo (Liptak Clare & Motis Timothy, 2017).

En algunos casos, el monitoreo directo del cultivo para la identificación de plagas no resulta muy eficiente, pues muchos insectos actúan en la noche cuando no hay buena visibilidad o el tamaño del cultivo hace que esta labor sea mucho más complicada. Por este motivo resulta conveniente recurrir a técnicas como el uso de trampas de monitoreo, que están activas durante el día y la noche capturando insectos (Liptak Clare & Motis Timothy, 2017).

El uso de trampas de monitoreo requiere de personal capacitado que realice la inspección, determinando la cantidad y el tipo de insectos atrapados, lo cual, actualmente en los cultivos se realiza manualmente. Una persona capacitada recorre el campo examinando las trampas y registra la información recolectada en planillas impresas. Esta actividad requiere de tiempo y esfuerzo visual por parte de la persona encargada, ya que, en ocasiones, es necesario el uso de lupa debido al tamaño de los insectos. Además, el tamaño de los cultivos es extenso, demandando el trabajo de más de una persona para realizar esta tarea. Debido a lo anterior surge la problemática: ¿Cómo clasificar y contar los insectos atrapados

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
sobre una placa de color sólido, presente en una cama piloto de la finca Trigal Caribe, de una manera automatizada?

El desarrollo de este proyecto busca simplificar el trabajo que se realiza hoy en día en las empresas floriculatoras a la hora de monitorear la presencia de plagas. Pues al identificar de manera automática la presencia de insectos y el comportamiento de su población en un cultivo, se suprime la necesidad de personal especializado para realizar dicha actividad, además de reducir el tiempo que le toma llevarla a cabo.

1.2 OBJETIVOS DEL PROYECTO

1.2.1 Objetivo General

Cuantificar la viabilidad de automatizar el proceso de conteo y caracterización de insectos atrapados en trampas sólidas.

1.2.2 Objetivos Específicos

- Implementar un método computacional para la caracterización y cuantificación del número de insectos capturados en trampas sólidas.

- Comparar el tiempo y la precisión de la identificación de los insectos atrapados en las trampas con el método computacional implementado y de manera manual.

- Cuantificar el retorno de la inversión del desarrollo de una aplicación Android para dispositivos móviles que cuente y caracterice los insectos atrapados por trampas sólidas de manera automática.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
1.3 MARCO DE REFERENCIA

1.3.1 Marco contextual

Para el desarrollo del proyecto se tomarán datos y se realizarán pruebas en los cultivos de la empresa Flores El Trigal S.A.S, la cual se dedica a la producción de flores de corte para exportación y tiene sus instalaciones distribuidas en el oriente del departamento de Antioquia. El territorio del oriente antioqueño, con una altura de 2.015 metros sobre el nivel del mar y una temperatura promedio de 17°C (Rojas Pérez, 2014) presenta condiciones favorables para el proceso de cultivo de flores (Chica Toro & Correa Londoño, 2005). Pero, es también este clima versátil y cambiante el que intensifica la propagación de plagas e insectos (Grupo técnico ProCaucho, 2012), demandando un constante monitoreo del cultivo para mantenerlo sano y controlado.

Las plagas y enfermedades son una problemática que afecta diferentes tipos de cultivos alrededor del mundo, por lo cual, múltiples autores han contribuido con métodos, herramientas o investigaciones a su solución. A continuación, se presentan algunas tablas resumen de múltiples iniciativas que se han llevado a cabo. Se registra en cada tabla el título de la investigación, el método utilizado para la clasificación o conteo automáticos, el dispositivo utilizado para la adquisición de las imágenes, los insectos clasificados junto con su porcentaje de precisión alcanzado y el enlace del artículo del cual se extraja la información. Los artículos elegidos se asemejan a la solución que se desea lograr con el desarrollo del proyecto, aportando información valiosa al respecto y permitiendo aprender de la experiencia de los autores.

<table>
<thead>
<tr>
<th>Tabla 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Información resumen de artículo antecedente 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Título del artículo</th>
<th>Un algoritmo de visión inteligente para contar moscas blancas y trips en trampas adhesivas utilizando el espectro de transformada de Fourier bidimensional.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Método utilizado</td>
<td>Algoritmo de visión inteligente basado en el espectro de transformación de Fourier bidimensional (2DFT) que trata los insectos como ruido en la imagen.</td>
</tr>
<tr>
<td>Adquisición de imágenes</td>
<td>Cámara digital (Nikon Coolpix S9200, 4.7 Mega pixels and 24 bits pixel^-1)</td>
</tr>
<tr>
<td>Insectos</td>
<td>% Clasificación</td>
</tr>
<tr>
<td>moscas blancas de batata</td>
<td>99.94 %</td>
</tr>
<tr>
<td>trips de flores occidentales</td>
<td>99.89 %</td>
</tr>
<tr>
<td>Enlace</td>
<td><a href="https://doi.org/10.1016/J.BIOSYSTEMSENG.2016.11.001">https://doi.org/10.1016/J.BIOSYSTEMSENG.2016.11.001</a></td>
</tr>
</tbody>
</table>

Fuente: Elaboración propia

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
<table>
<thead>
<tr>
<th>Título del artículo</th>
<th>Enfoque de características invariables de escala para el monitoreo de insectos.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Método utilizado</td>
<td>Versión ampliada del algoritmo LOSS para una identificación inicial seguido de un procesamiento digital de imágenes basado en la transformación de características invariantes de escala (SIFT).</td>
</tr>
<tr>
<td>Adquisición de imágenes</td>
<td>Cámara digital de alta resolución (Panasonic Lumix).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Insectos</th>
<th>% Clasificación</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thrips (Thrips tabaci L.)</td>
<td>99,75%</td>
</tr>
<tr>
<td>Mosca blanca (Bemisia tabaci Genn.)</td>
<td>98,27%</td>
</tr>
<tr>
<td>Áfidos (Aphis gossypii Genn.)</td>
<td>97,60%</td>
</tr>
<tr>
<td>Diabrotica (Coleoptera: Chrysomelidae)</td>
<td>97,44%</td>
</tr>
<tr>
<td>Lacewings spp.</td>
<td>96,28%</td>
</tr>
</tbody>
</table>

Enlace: https://doi.org/10.1016/J.COMPAG.2010.10.001
Fuente: Elaboración propia

<table>
<thead>
<tr>
<th>Título del artículo</th>
<th>Algoritmo de visión artificial para la exploración de moscas blancas (Bemisia tabaci Genn.) en ambiente de invernadero.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Método utilizado</td>
<td>Algoritmo de visión artificial basado en la extracción de características de excentricidad y área de las proyecciones del insecto permite diferenciarlo de otros insectos y de otros elementos presentes en las trampas adhesivas.</td>
</tr>
<tr>
<td>Adquisición de imágenes</td>
<td>Cámara digital de alta resolución (Panasonic Lumix).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Insectos</th>
<th>% Clasificación</th>
</tr>
</thead>
<tbody>
<tr>
<td>moscas blancas (Bemisia tabacín)</td>
<td>96,95%</td>
</tr>
</tbody>
</table>

Enlace: https://doi.org/10.1111/j.1439-0418.2009.01400.x
Fuente: Elaboración propia

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

### Tabla 4
**Información resumen de artículo antecedente 4**

<table>
<thead>
<tr>
<th>Título del artículo</th>
<th>Método utilizado</th>
</tr>
</thead>
<tbody>
<tr>
<td>Un sistema de reconocimiento y conteo para insectos voladores en la agricultura inteligente basado en la visión.</td>
<td>Se usa inicialmente un sistema de detección y conteo aproximado basado en YOLO (You Only Look Once), posteriormente se aplica el método de clasificación y conteo fino SVM (Support Vector Machine) basado en características globales.</td>
</tr>
</tbody>
</table>

| Adquisición de imágenes | Módulo de cámara V2 Sony IMX219 de alta definición. |

<table>
<thead>
<tr>
<th>Insectos</th>
<th>% Clasificación</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abeja</td>
<td>91,89%</td>
</tr>
<tr>
<td>Mosca</td>
<td>88,65%</td>
</tr>
<tr>
<td>Zancudo</td>
<td>90,23%</td>
</tr>
<tr>
<td>Polilla</td>
<td>92,31%</td>
</tr>
<tr>
<td>Escarabajo</td>
<td>87,69%</td>
</tr>
<tr>
<td>Mosca de fruta</td>
<td>91,25%</td>
</tr>
</tbody>
</table>

**Enlace**

https://doi.org/10.3390/s18051489

Fuente: Elaboración propia

### Tabla 5
**Información resumen de artículo antecedente 5**

<table>
<thead>
<tr>
<th>Título del artículo</th>
<th>Método utilizado</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimación de la densidad de Bemisia tabaci (Hemiptera: Aleyrodidae) en un invernadero utilizando trampas adhesivas junto con un sistema de procesamiento de imágenes.</td>
<td>Se usan algoritmos simples de procesamiento digital de imágenes, basados en las características morfológicas de los insectos, para estimar la densidad poblacional de insectos.</td>
</tr>
</tbody>
</table>

| Adquisición de imágenes | Plataforma de escáner de imágenes (HP Scanjet G3010®) |

<table>
<thead>
<tr>
<th>Insectos</th>
<th>% Clasificación</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mosca blanca Bemisia tabaci (Hemiptera: Aleyrodidae)</td>
<td>97,56%</td>
</tr>
</tbody>
</table>

**Enlace**

https://doi.org/10.1016/J.ASPEN.2008.03.002

Fuente: Elaboración propia
Tabla 6
Información resumen de artículo antecedente 6

<table>
<thead>
<tr>
<th>Título del artículo</th>
<th>Identificación de la mosca de la fruta en trampas inteligentes utilizando técnicas de procesamiento digital de imágenes y aprendizaje automático.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Método utilizado</td>
<td>Se aplica procesamiento digital de imágenes, se extrae un vector de características y se aplican métodos de aprendizaje automático: support vector machine (SVM), k-nearest neighbors (KNN), decision tree (DT) y Gaussian Naive Bayes (GNB).</td>
</tr>
<tr>
<td>Adquisición de imágenes</td>
<td>Módulo de cámara 5-megapixel Raspberry Pi.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Insectos</th>
<th>% Clasificación</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mosca mediterránea de la fruta (Ceratitis capitata)</td>
<td>85 %</td>
</tr>
<tr>
<td>Mosca suramericana de la fruta (Anastrepha fraterculus)</td>
<td>84 %</td>
</tr>
</tbody>
</table>

| Enlace                        | https://doi.org/10.1145/3167132.3167155 |

Fuente: Elaboración propia

Marco teórico

Los conocimientos teóricos relacionados con el desarrollo de este proyecto se dividen principalmente en dos ramas: la agricultura y las técnicas computacionales. La primera permite contextualizar y entender la problemática desde conceptos como el monitoreo de cultivos y la identificación de plagas, mientras la segunda, orienta hacia las posibles soluciones a implementar.

Agricultura

La identificación temprana de plagas en un cultivo da paso al uso oportuno de herramientas o insumos para su erradicación, pues para algunos casos, existe una población o una edad específica en los insectos que es ideal para fumigarlos (Liptak Clare & Motis Timothy, 2017).

Según la Organización de las Naciones Unidas para la Alimentación y la Agricultura (FAO), el Manejo Integrado de Plagas (MIP) consiste en la integración de técnicas de control y monitoreo de plagas, con la posterior aplicación de medidas que mitiguen el desarrollo de sus poblaciones. Lo anterior, manteniendo el uso justificado de pesticidas en niveles
mínimos, reduciendo los riesgos en la salud humana y en el impacto al medio ambiente. El MIP promueve el uso de mecanismos naturales para el control de plagas (FAO, 2020).

Monitoreo de cultivos floricultores

El monitoreo de cultivos es una técnica de MIP, la cual consiste en la inspección regular de las plantas con el objetivo de encontrar enfermedades, insectos o malezas (Liptak Clare & Motis Timothy, 2017). El monitoreo es posible realizarlo de manera directa o indirecta. La primera consiste en recorrer la finca o el campo examinando directamente algunas plantas al azar, con el fin de identificar plagas o daños en las plantas que indiquen su presencia; mientras la segunda, consiste en usar instrumentos como trampas para capturar insectos y luego realizar una lectura de ellas.

El uso de trampas de monitoreo tiene grandes ventajas, ya que están activas todo el tiempo, pues muchos insectos actúan en la noche o en momentos en los cuales la persona encargada de monitorear no está presente.

Las trampas consisten en un mecanismo de atracción o cebo que incita a los insectos a acercarse para luego capturarlos. Actualmente, existen diferentes tipos de trampas de monitoreo que pueden ser fabricadas artesanalmente o adquiridas en tiendas dedicadas al sector agrícola. (Liptak Clare & Motis Timothy, 2017)

Este proyecto se centrará en cuantificar y caracterizar los insectos atrapados en trampas solidas adhesivas de color amarillo y azul, ya que son estos dos colores los principales señuelos para las especies a tratar (Ekrem ATAKAN & Serkan PEHLIVAN, 2015).

Plagas

En los cultivos de flores, comúnmente se encuentran múltiples insectos, pero no todos son considerados plagas, ya que, para pertenecer a este grupo, el animal debe causar daño o afectar negativamente la planta (RAE, 2019). Según lo anterior, hay insectos presentes en los cultivos que no perjudican la cosecha, y que por el contrario, son beneficios para ésta ya que actúan como control biológico alimentándose de las plagas (Doğan et al., 2019).

La identificación de los insectos es fundamental, pues de esta manera se determina si las plantas están expuestas ante alguna plaga y en qué proporción, según su población. Para el desarrollo de este proyecto se prioriza la identificación de tres tipos de insectos, clasificados como plagas para los cultivos a tratar, estos son: trips, mosca blanca y minador de hoja.

Trips

Los trips (Frankliniella bispinosa) son insectos pequeños y delgados. Cuando son adultos miden aproximadamente 1.3 mm y tienen alas, mientras que, cuando son larvas, no poseen alas como puede verse en la Figura 1. Su color puede variar desde blanco translúcido o

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
amarillo a marrón oscuro o negro según su especie. Se alimentan perforando la capa externa del tejido de la planta para succionar su contenido celular. La alimentación del trips produce múltiples efectos negativos en la planta huésped, como lo son: manchas decoloradas en la superficie de las hojas, formación de cicatrices y crecimiento distorsionado, como se observa en la Figura 2. Mientras que en las flores, puede presentar rayas en los pétalos o provocar que el brote no se abra correctamente (Moritz et al., 2009).

![Trips, Frankliniella bispinosa](image)

**Figura 1.** Trips, Frankliniella bispinosa  
 a) Trips adulto. b) Larva de trips.  
Fuente: (Arthurs et al., 2015)

![Daño causado por presencia de Trips](image)

**Figura 2.** Daño causado por presencia de Trips  
 En la parte superior izquierda de la figura se observa la formación de cicatrices, mientras en la parte inferior derecha se presenta una mancha decolorada. Ambos efectos se deben a la presencia de Trips.  
Fuente: Elaboración propia

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

**Mosca blanca**

La mosca blanca (*Aleurotrachelus trachoides*) es un pequeño insecto que recibe su nombre debido a la cera blanca y harinosa que cubre las alas y el cuerpo del adulto, presentando además coloración amarillenta y cuatro alas blanquecinas como se muestra en la Figura 3.a. La mosca blanca se reproduce rápidamente en climas cálidos (Vivek Kumar et al., 2016).

Este insecto succiona la savia del floema (tejidos que conducen los alimentos en los tallos y en las hojas de las plantas), además, como se observa en la Figura 3.b, excretan un líquido azucarado llamado melaza, el cual hace que las hojas se vuelvan pegajosas, se tornen amarillentas y se caigan. La presencia de melaza en las hojas también puede causar que estas se llenen de hormigas afectando los procesos de producción. Las pérdidas en las plantas se dan cuando la población de las moscas es suficientemente significativa (Vivek Kumar et al., 2016).

![Mosca blanca](image1) ![Hoja infestada de melaza debido a la presencia de mosca blanca](image2)

**Figura 3.** Mosca blanca, *Aleurotrachelus trachoides*

a) Mosca blanca. b) Hoja infestada de melaza debido a la presencia de mosca blanca

Fuente: (Vivek Kumar et al., 2016)

**Minador**

El minador de hoja (*Liriomyza trifolii*) inicia su proceso de desarrollo cuando la hembra perfora la superficie de la hoja para depositar los huevos, los cuales son claros al principio y luego pasan a ser blanco cremoso. Posteriormente, estos se convierten en larvas incoloras con un par de espiráculos en un extremo y a medida que maduran adquieren un color amarillento como se presenta en la Figura 4.a. Más tarde esta larva sale de la mina creada en la hoja y cae al suelo para convertirse en puparium, el cual, como se muestra en la Figura 4.b, inicialmente es amarillo, luego marrón dorado y con el tiempo se vuelve marrón oscuro, es ovalado y mide alrededor de 1.3 a 2.3mm de largo. Por último, en la Figura 4.c se observa el minador en la edad adulta midiendo aproximadamente 2mm, donde desarrollan alas de 1.25 a 1.9mm. Se caracterizan por tener tórax y abdomen, en su mayoría, grises y negros; patas, cabeza y superficie ventral de color amarillo y ojos rojos (Capinera & Castner, 2017).

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

Figura 4. Minador de hoja, Liriomyza trifolii
a) Larva de minador. b) Puparium de minador. c) Minador adulto
Fuente: (Capinera & Castner, 2017)

Aunque la hembra durante la oviposición y los procesos de alimentación puede causar una apariencia punteada en el follaje (Parrella et al., 1985), el principal daño de esta especie es el minado de la hoja causado por la larva como se muestra en la Figura 5, que termina en la destrucción del mesófilo de las hojas reduciendo en gran medida el área fotosintética de la planta (Capinera & Castner, 2017).

Figura 5. Daño causado por presencia de Minador de hoja
a) Hoja punzada por la oviposición. b) Galería generada por larva. En el extremo señalado se observa la sombra amarillenta de la larva dentro de la mina. c) Larva extraída de su galería.
Fuente: Elaboración propia

Técnicas computacionales

Para identificar los insectos de interés en las trampas, se evaluarán diferentes métodos computacionales. Inicialmente se analizará la técnica de procesamiento digital de imágenes, posteriormente se explorará la plataforma Custom Vision de Microsoft y finalmente el algoritmo YOLO.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
Procesamiento digital de imágenes

Esta es una técnica utilizada por la Visión Artificial o Computacional en la extracción de características de una imagen para su posterior descripción e interpretación (Florence Ysiquio, 2004).

El procesamiento digital de imágenes consiste en la manipulación de imágenes digitalmente para describir, reconocer o transformar su contenido (Pitas, 2000). Generalmente, este proceso se divide en cinco pasos: (1) *Adquisición de imagen* o su conversión a formato digital; (2) *Pre-procesamiento* son las operaciones que mejoran la imagen, manteniendo las mismas dimensiones que la original; (3) *Segmentación* consiste en la división la imagen en regiones disjuntas que no se superponen; (4) *Extracción de características* es la definición de atributos de los objetos como tamaño, forma, color, textura, etc.; y (5) *Clasificación* es la identificación de los objetos y su separación en diferentes grupos (Mendoza & Lu, 2015). En la *Figura 6* se muestra gráficamente la secuencia de pasos y los resultados representativos de cada etapa.

*Figura 6. Secuencia de pasos del procesamiento digital de imágenes*

Fuente: Tomado de (Mendoza & Lu, 2015)

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
Adquisición de la imagen

La etapa inicial del proceso es de suma importancia, ya que de esta depende que se obtengan los resultados esperados en las siguientes etapas. Para esto, se deben considerar los factores descritos a continuación:

a) Cámara: Es el primer paso para la adquisición de las imágenes. Para su correcta elección se revisan múltiples características. A continuación, se mencionan algunas de ellas:

- Cámara de color o monocromática: Para decidir cuál es la cámara más apropiada, además del presupuesto, se debe tener en cuenta cuál es el proceso que se va a realizar, ya que es dependiendo de este que se determina si es necesario poseer información del color de los objetos o si es suficiente una imagen monocromática. También es importante tener en cuenta el tiempo de cómputo requerido por una imagen de color y por una imagen monocromática, así como también su demanda de procesamiento.

- Velocidad del obturador: Se debe tener claro si las imágenes se van a obtener en movimiento o estáticamente.

- Resolución: A mayor resolución se pueden observar objetos y detalles más pequeños, pero también requerirá de mayor tiempo de procesamiento y mayor costo de la cámara.

- Distancia focal: La distancia focal es la distancia desde el centro de la lente hasta el punto de imagen (plano focal) donde se recoge la luz para la imagen. Las diferentes distancias focales crean diferentes niveles de aumento y cambian el ángulo de visión de la fotografía resultante (Panasonic Corporation, 2018).

- Diáfragma: La apertura del diáfragma se modifica para controlar la cantidad de luz que pasa a través de la lente. Al alterar la configuración de apertura, la profundidad de campo y la nitidez de la imagen se puede controlar una mayor expresividad fotográfica (Panasonic Corporation, 2018).

b) El host y el software utilizados: Un proceso de visión artificial requiere de una gran cantidad de cálculos para el procesamiento de las imágenes y más aún cuando se requiere que el sistema actúe en tiempo real. Debido a esto, es importante elegir un equipo adecuado según las condiciones ambientales donde operará y según el procesador o procesadores requeridos para el tratamiento de las imágenes. Además, el software a utilizar se elige según el lenguaje de programación que se empleó para la captura y procesamiento de las imágenes.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
c) Técnicas de iluminación: Un entorno previamente adecuado para la adquisición de las imágenes mejora notoriamente la calidad y la distinción de los detalles. Existen diversas técnicas de iluminación para la adquisición de las imágenes:

- Iluminación frontal: Para superficies con pocos reflejos
- Iluminación lateral: Para resaltar detalles específicos en los objetos que solo son visibles orientando la luz de forma lateral al eje de la cámara.
- Iluminación por campo oscuro: Es ideal para resaltar texturas superficiales como grietas, caracteres o códigos grabados.
- Iluminación por contraste: Esta técnica se utiliza para identificar la silueta de los objetos y permite realizar mediciones precisas. Es utilizada en especial en materiales translucidos o transparentes para la identificación de manchas e imperfecciones.
- Iluminación coaxial: Se usa para la iluminación de objetos reflectantes.
- Iluminación difusa: Se usa para instrumentos médicos, espejos y otras superficies especulares complejas. No produce sombras (DACHS-SOLUTIONS, 2019).

**Preprocesamiento**

En esta etapa se busca eliminar el ruido capturado, ajustar el nivel de brillo y contraste en la imagen, con la intención de mejorar la apariencia de esta. Algunas técnicas de preprocesamiento se mencionan a continuación:

a) Binarización: Consiste en convertir en cero (0) los píxeles que estén por debajo de un umbral definido y en uno (1) los píxeles que estén por encima de ese umbral. Es importante aclarar que la imagen debe estar en escala de grises antes de ser binarizada.

b) Transformaciones geométricas: Estas técnicas, generalmente, consisten en redistribuir o modificar las coordenadas de los píxeles. Algunas transformaciones son:

- Escalado: Permite reducir o aumentar el tamaño de la imagen, así como también, permite realizar un acercamiento en determinadas zonas.
- Traslación o desplazamiento: Se utiliza para cambiar la ubicación de un objeto detectado dentro de la imagen o de la imagen completa.
- Rotación: Se utiliza para girar una imagen sobre su centro.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
- Espejo: Permite cambiar la apariencia de la imagen invirtiéndola respecto al eje horizontal, vertical o diagonal.

c) Filtrado: las técnicas de filtrado permiten eliminar componentes no deseados dentro de una imagen. Existen diferentes tipos de filtro según lo que se desee eliminar o según lo que se desee mejorar dentro de la imagen:

- Filtros lineales: Reemplazan cada pixel por una combinación lineal de los pixeles vecinos. Usan un kernel convolucional como fórmula para la combinación lineal.
  
  o Media móvil o cuadro de desenfoque: Filtro espacial, pasa bajas, con kernel cuadrado. Se usa para suavizar la imagen y reducir el ruido (Blackledget, 2006).
  
  o Ventana de Hann: Filtro en el dominio de la frecuencia, pasa bajas. Elimina el ruido de la imagen suavizando los picos de frecuencia alta (Dartmouth Research Computing, 2020).
  
  o Desenfoque Gaussiano: Filtro frecuencial, pasa bajas no uniforme. Reduce el ruido de la imagen y los detalles despreciables (Misra & Wu, 2020).

- Filtros no lineales: Suavizan el ruido en la imagen preservando los bordes.
  
  o Filtro de la mediana: filtro digital. Ordena los valores de los pixeles vecinos, incluyendo el valor central, en un vector y toma el valor del medio (mediana) y se lo asigna al pixel central (Chandel & Gupta, 2013).
  
  o Transformaciones morfológicas binarias: Conjunto de operaciones no lineales relacionadas con las características de forma o morfológicas de una imagen. Usan un elemento estructural o Kernel para modificar la imágenes, pues modificando el tamaño y la forma de este elemento se ajusta la salida deseada (Beyerer et al., 2015). Las principales transformaciones morfológicas son:

  - Erosión: Hace que los espacios entre regiones se vuelvan más grandes. Los pequeños detalles de la imagen y algunos bordes son eliminados (Beyerer et al., 2015).
  
  - Dilatación: Aumenta el tamaño de los bordes de las regiones, a la vez que cierra espacios entre regiones. Los pequeños detalles intrusos dentro de las regiones son rellenados (Beyerer et al., 2015).

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
Existen otras operaciones morfológicas derivadas de la combinación de la erosión y la dilatación como el cierre, la apertura, la extracción de bordes, etc.

- **Filtro del máximo/mínimo**: Son casos especiales de los filtros morfológicos erosión y dilatación, por lo cual la imagen resultante es similar a la de estos (Tulleken, 2011).

- **Transformaciones morfológicas en escala de grises**: Es una generalización de imágenes de un bit por pixel a imágenes de múltiples bits por pixel, donde las operaciones Máximo y Mínimo son usadas en lugar de las operaciones Or y And respectivamente (Chandel & Gupta, 2013).

**Segmentación**

Consiste en dividir la imagen en regiones según un criterio establecido (Cheng et al., 2001). Actualmente existen múltiples técnicas de segmentación, las cuales han sido abordadas por diferentes autores. A continuación, se mencionan algunas de ellas:

a) **Segmentación basada en regiones**: Se fundamenta en el principio de homogeneidad, considerando que los pixeles vecinos dentro de una misma región poseen características similares, las cuales son diferentes a las de los pixeles de otras regiones (Khan & Srisha, 2013). Los criterios usados para la segmentación por regiones pueden ser el color, la intensidad, objeto, etc. (Ramya et al., 2015).

b) **Segmentación basada en bordes**: Consiste en identificar cambios repentinos en la intensidad de pixeles vecinos, lo cual indica el límite en una región. El reconocimiento de estas discontinuidades en la imagen permite identificar los bordes de las áreas de interés (Khan & Srisha, 2013).

c) **Segmentación basada en umbral (Thresholding)**: Esta técnica es la más simple y fácil de aplicar ya que solo requiere la determinación de un umbral para cumplir su función de segmentación, mientras las demás técnicas requieren de más parámetros o acciones. Se basa en los niveles de intensidad de los pixeles para segmentar la imagen en dos clases, la primera es a la cual pertenecen los pixeles con un nivel de intensidad por debajo (o por encima, según se requiera) del umbral establecido y la segunda es a la que pertenecen el resto de los pixeles. Esta técnica es comúnmente usada para separar objetos del fondo usando una partición binaria para segmentar la imagen (Khan & Srisha, 2013).

d) **Segmentación basada en clusters**: El objetivo de esta técnica es agrupar pixeles con atributos semejantes, segmentando los datos de la imagen en clusters de pixeles con información similar (Khan & Srisha, 2013).

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
**Extracción de características**

En esta sección se determina numéricamente el valor de los atributos de interés de una imagen, lo que conlleva a una gran reducción en la cantidad de datos necesarios para representarla, ya que se seleccionan una características relevantes y sus valores son los que la representarán en lugar de toda la información de la imagen original (Khan & Srisha, 2013).

Son múltiples los atributos mediante los cuales puede describirse una imagen. A continuación, se describirán tres de los principales: color, textura y geometría o morfología.

a) **Análisis de color**: La identificación de colores, aunque en la cotidianidad se realiza mediante etiquetas o nombres, en el procesamiento de imágenes y en el mundo digital no es muy conveniente hacerlo de esta manera ya que los colores pueden medirse y evaluarse fisiológicamente, convirtiéndolos en números. Por esto, deben representarse de forma que puedan ser adecuadamente almacenados, mostrados y procesados (Koschan & Abidi, 2008).

La identificación de colores en el procesamiento digital de imágenes puede realizarse de diferentes maneras según el objetivo del procesamiento, estas son conocidas como *espacios de color*, los cuales indican el sistema coordenado de color en el cual los valores de la imagen están representados (Koschan & Abidi, 2008). A continuación, se mencionan algunos de los espacios de color más utilizados:

- **Modelo de color RGB**: Representa una mezcla aditiva de los colores primarios rojo, verde y azul. El principal propósito de este modelo es sensar, representar y mostrar imágenes en sistemas electrónicos como televisores y computadores, sin embargo, también ha sido usado para fotografía convencional. Este modelo se representa mediante un cubo ubicado en el origen de un sistema coordenado. Las dimensiones del cubo son 1x1x1, lo que indica que no hay valores negativos en este espacio de color y que cada componente está en un rango de 0-1 (Celebi & Schaefer, 2013). Este modelo es diferente del modelo de color BGR, el cual es usado por la librería OpenCV. Al cambiar el orden de los colores rojo y azul se varía el área de importancia de cada color.

- **Modelo de color HSV**: Representa el tono (hue), la saturación (saturation) y el valor (value). También se conoce como HSI (I, intensidad), HSB (B, brillo), HSL (L, luminosidad). Este modelo de color es una representación en sistema coordenado cilíndrico del modelo RGB, reorganizando la geometría del cubo para ser más intuitivo. Fue desarrollado en 1970 para aplicaciones gráficas de computador, herramientas de modificación de color en software de edición de imágenes y para análisis de imágenes y visión artificial (Celebi & Schaefer, 2013).

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
• Modelo de color CMYK: Es un modelo de color sustractivo utilizado para la impresión a color. CMYK hace referencia a los cuatro colores o tintas usadas en este proceso: cian (cyan), magenta (magenta), amarillo (yellow) y negro (key) (Celebi & Schaefer, 2013).

b) Análisis de textura: La textura, en el procesamiento digital de imágenes, se define como la distribución espacial de variaciones de intensidad de los píxeles en una imagen o, también, la repetición de un patrón de variaciones locales de intensidad. La textura no puede ser definida por un punto o píxel (Mendoza & Lu, 2015).

c) Análisis morfológico o geométrico: Una vez se ha identificado la región de interés se pueden describir las propiedades geométricas de esta mediante un análisis morfológico. El tamaño y la forma son características comunes de este análisis, las cuales son fáciles de medir usando técnicas de procesamiento de imágenes. Otros atributos morfológicos usados para este análisis son: el conteo de partículas, huecos, gránulos, entre otras, así como la definición de su posición, rugosidad superficial y orientación dentro de la imagen (Mendoza & Lu, 2015).

La medición del tamaño de partículas u otros objetos en una imagen, la mayoría de veces, no es suficiente para realizar un análisis, por esto existen otras características relacionadas con la geometría de los objetos que permiten profundizar en la descripción de imágenes como: relación de aspecto, área, circularidad, compacidad, diámetro, excentricidad, redondez, perímetro, elongación, etc. (Mendoza & Lu, 2015).

Clasificación

La detección de objetos hace referencia al reconocimiento de elementos dentro de una imagen y la determinación de su posición y orientación (Beyerer et al., 2015). En un estudio realizado en la Universidad Carnegie Mellon en Pittsburgh, Pensilvania, se demostró experimentalmente que mediante la optimización de combinaciones lineales de filtros, dentro de un rango de parámetros, es posible lograr un buen desempeño en la detección de objetos sin recurrir a experimentos ensayo-error o a entrenamientos de modelos (Carmichael et al., 2002).

Machine Learning con Azure Custom Vision

El aprendizaje automático o machine learning consiste en programar computadoras con el fin de optimizar un criterio de rendimiento usando experiencias pasadas o datos de ejemplo. El aprendizaje se basa en modelos que pueden ser predictivos, para conocer posibles comportamientos futuros, o descriptivo para obtener información de los datos, o ambos. El

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
El rol de la ciencia computacional en el aprendizaje automático cumple una doble funcionalidad: inicialmente se debe entrenar el modelo, por medio de algoritmos para resolver el problema de optimización y para almacenar y procesar la cantidad de datos; en segundo lugar, una vez entenado el modelo, este debe ser igualmente eficiente para obtener una buena precisión predictiva (Thomas Dietterich et al., 2010).

Custom Vision es un servicio de reconocimiento de imágenes ofrecido por Microsoft Azure, el cual permite crear e implementar identificadores de imágenes y detectores de objetos personalizados a través de una interfaz web (Microsoft, 2020a).

Este servicio usa un algoritmo de aprendizaje automático (machine learning) para analizar las imágenes cargadas por el usuario, las cuales serán posteriormente clasificadas o etiquetadas. El algoritmo se entrena con base en estos datos y calcula sus medidas de desempeño realizando pruebas con esas mismas imágenes de entrenamiento. Una vez que el algoritmo finaliza su entrenamiento, se pasa a la fase de prueba donde se cargan nuevas imágenes para luego entrenar el modelo nuevamente o implementarlo en la aplicación de reconocimiento de imágenes, según sean los resultados obtenidos. Este servicio también ofrece la opción de exportar el modelo entrenado para usarlo sin conexión (Microsoft, 2020a).

Custom Vision se divide en dos funcionalidades, la clasificación de imágenes y la detección de objetos. La primera le asigna una o varias etiquetas a una imagen, mientras la segunda, además de la etiqueta, entrega las coordenadas o la localización de el o los objetos identificados dentro de la imagen (Microsoft, 2020a).

La optimización de este servicio permite la identificación rápida de las principales diferencias entre imágenes, empezando a crear el modelo con una pequeña cantidad de datos, aunque se sugiere un mínimo de cincuenta imágenes por etiqueta (Microsoft, 2020a).

Debido al enfoque del proyecto, se hará énfasis en el servicio de detección de objetos. Los prerrequisitos para hacer uso de este servicio son: una cuenta en Azure y un conjunto de imágenes para entrena el modelo. Microsoft recomienda que el conjunto de imágenes a utilizar sea de variedad visual, es decir, que tengan diferentes ángulos de cámara, iluminación, fondo, tamaño, etc., además de cumplir los siguiente criterios: el formato debe ser .jpg, .png, .bmp o .gif; el tamaño de las imágenes de entrenamiento no debe superar los 6 MB y para las de predicción los 4 MB; el borde más corte de la imagen debe ser superior a los 256 píxeles (si la imagen tiene un borde menor, Custom Vision la escalará verticalmente y de forma automática) (Microsoft, 2020b).

Una vez cargadas las imágenes, estas aparecerán en la sección “Sin Etiqueta” en la interfaz de usuario. El siguiente paso es etiquetarlas manualmente señalando los objetos que quiere enseñar al modelo en cada imagen. La interfaz web es intuitiva y amigable con el usuario en este proceso, ya que le sugiere etiquetas y objetos a etiquetar. Cuando se finaliza el proceso de etiquetado, se procede a entrena el modelo, lo cual solo toma unos minutos.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
Posteriormente, se muestran los indicadores de rendimiento calculados para el modelo entrenado, los cuales representan la eficacia del detector de objetos.

La **precisión** y la **coincidencia** son dos de las medidas de eficacia calculadas. La primera indica la fracción de las identificaciones que fueron correctas, mientras la segunda indica la fracción de las clasificaciones reales que se identificaron correctamente (Microsoft, 2020b).

En la interfaz web hay una pestaña llamada “Performance”, en la cual se muestran los indicadores de desempeño del modelo entrenado. En esta, también se presentan unos controles deslizantes que permiten al usuario modificar parámetros que alteran el rendimiento del modelo. Uno de ellos es el Umbral de Probabilidad, que representa el nivel de confianza que debe tener una predicción para considerarse correcta.

Luego de realizar ajustes mediante los parámetros variables, se procede a probar y reentrenar el modelo, de ser necesario. Cuando se reentrena, se crea una iteración nueva con sus propias métricas de rendimiento. Estas iteraciones se muestran en la interfaz de usuario y pueden ser usadas o eliminadas en el momento que el usuario lo desee.

**Deep Learning con YOLO**

El aprendizaje profundo es una rama del machine learning que entrena un modelo para realizar tareas similares a las realizadas por un ser humano, como lo son el reconocimiento de sonidos o imágenes o realizar predicciones. Este procedimiento, en lugar de ordenar los datos para que se ejecuten a través de ecuaciones predefinidas, configura parámetros básicos acerca de los datos para que el computador, por su propia cuenta, reconozca patrones mediante el uso de varias capaz de procesamiento (SAS: Software y Soluciones de Analítica, 2019).

You Only Look Once (YOLO) es un sistema de detección de objetos en tiempo real, propuesto por Joseph Redmon, Santosh Divvala, Ross Girshick y Ali Farhadi en 2016, el cual consiste en una sola red neuronal convolucional que predice simultáneamente múltiples recuadros delimitadores y las probabilidades de clasificación para dichos recuadros (Redmon et al., 2016).

YOLO es un modelo que se posiciona por encima de las técnicas tradicionales de detección de objetos debido a su rapidez, ya que enmarca la detección como una regresión; a su método de procesar la imagen de manera general para hacer predicciones, lo que le permite obtener información contextual de las clases y su apariencia; y a su generalización de las representaciones de los objetos, que lo vuelven menos propenso a fallar cuando recibe entradas inesperadas (Redmon et al., 2016).

**Detección de objetos**

Como se explica en el estudio realizado por Joseph Redmon (2016), el sistema divide la imagen de entrada en una malla de \( S \times S \). Si el centro del objeto queda dentro de una celda...
La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

**Figura 7. Procedimiento para detección de objetos**
Representación de procedimiento, previamente descrito, que realiza el sistema en la imagen de entrada para detectar o clasificar los objetos.
Fuente: Tomada de (Redmon et al., 2016)

*Red neuronal*

En el estudio realizado por Joseph Redmon también se presenta la arquitectura de red usada en el modelo YOLO, indicando que las capas convolucionales iniciales de la red extraen características de la imagen, mientras las capas completamente conectadas predicen las probabilidades de salida y las coordenadas. Es una red neuronal basada en el modelo de clasificación GoogLeNet. Tiene 24 capas convolucionales seguidas de 2 capas completamente conectadas. La salida final de la red es un tensor de predicciones de 7x7x30. La arquitectura completa de la red neuronal se muestra en la **Figura 8**.

*Figura 8. Arquitectura de la red neuronal usada por YOLO*
Fuente: Tomada de (Redmon et al., 2016)

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
La red está diseñada con 4 capas convolucionales y 2 capas completamente conectadas con pesos aleatoriamente inicializados. La capa final predice tanto probabilidades como recuadros y coordenadas usando una función de activación lineal, mientras las demás capas usan la función de activación rectificada presentada en la función (4):

$$\varphi(x) = \begin{cases} 
    x, & \text{if } x > 0 \\
    0.1x, & \text{de otra manera}
\end{cases} \quad (4)$$

Junto con el modelo YOLO, se propuso una versión rápida de este, la cual usa una red neuronal con menos capas convolucionales (9 en lugar de 24) y menos filtros en dichas capas, de resto todos los parámetros con los mismos que la versión general de YOLO. Esta versión de YOLO es el modelo más rápido para detección de objetos (Redmon et al., 2016).

Para la implementación de la red del sistema YOLO se usa el marco de red neuronal de código abierto Darknet, el cual se escribe en C y soporta tanto computación CPU como GPU (Redmon, 2016).

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
2. METODOLOGÍA

Para dar solución a la problemática planteada y alcanzar los objetivos propuestos, se adopta la metodología Design Thinking de Herbert Simon (Warner & Simon, 1969), la cual se define como un modelo no lineal, iterativo (es posible repetirlo en varias ocasiones y devolverse a través de las etapas). Esta metodología plantea 7 etapas a través de las cuales se puede llegar a una solución del problema establecido (Ambrose & Harris, 2010).

A continuación, se presenta la estructura metodológica que guiará el desarrollo del presente proyecto, la cual se determinó con base en el libro Basics Design 08: Design Thinking de los autores Ambrose & Harris:

ETAPA 1: Definición

Se identifican las características de la población objetivo del proyecto; se da respuesta a preguntas como: ¿quién usará el prototipo propuesto?, ¿cuál es el nivel de escolaridad de dicho usuario?, ¿cuál es su cultura, edad? ¿Cuál es la afinidad o destreza con desarrollos tecnológicos semejantes al propuesto?

En esta etapa se define qué espera el cliente obtener al final del proyecto, cuáles son los requerimientos del usuario, cómo se logrará satisfacer dichos necesidades, cuáles son los atributos de la calidad o requerimientos técnicos, se realiza la House of Quality (HoQ), la arquitectura y descomposición funcionales (caja transparente).

ETAPA 2: Investigación

Con base en los antecedentes, se evalúa qué información facilitará o impulsará el progreso del proyecto y cuáles serían los posibles obstáculos que se presentarán a lo largo del desarrollo.

Se busca información que pueda ampliar conocimientos relacionados con los temas del proyecto y que generen o aporten información de interés, en relación tanto con el software y la programación como al hardware y las herramientas físicas.

ETAPA 3: Ideación

Se proponen múltiples alternativas que den solución al problema planteado, para esto se usan herramientas que permitan organizar la información y las ideas, con el fin de generar la mayor cantidad de alternativas posibles.

En esta etapa se usan herramientas de ideación cómo matriz morfológica.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
ETAPA 4: Prototipado

Se construyen o materializan los prototipos de las alternativas propuestas para evaluar su funcionalidad. Se debe tener en cuenta cuales son las variables o métricas a partir de las cuales se compararán las alternativas ya que, en los prototipos, estas deben poder medirse efectivamente.

ETAPA 5: Selección

En esta etapa se evalúan los prototipos desarrollados y se comparan entre ellos para seleccionar el que tenga un mejor desempeño, según las métricas establecidas. A su vez, el prototipo seleccionado se evalúa respecto a los requerimientos de usuario planteados al inicio del proyecto. La verificación del cumplimiento de las métricas del prototipo se hace a través de un diseño de experimento en el cual se puedan evaluar las variables deseadas.

En caso de que esto último no cumpla lo establecido se debe volver a las etapas iniciales, sea para corregir el prototipo o para crear uno nuevo.

ETAPA 6: Implementación

Se lleva el prototipo a una versión que cumpla con las especificaciones visuales y estéticas que impuso el cliente. En esta etapa se obtiene el producto terminado del proyecto que debe cumplir con todos los requerimientos y expectativas del usuario. Se cuantifica el retorno de la inversión del proyecto verificando las mejoras consecuentes de éste.

ETAPA 7: Aprendizaje

En la última etapa del procedimiento se realiza una retroalimentación con el cliente en la cual se expresan los aspectos a mejorar en el transcurso de todas y cada una de las etapas. Se indaga por la satisfacción generada por el producto final y por los ademanes que se presentaron en el desarrollo. Este aprendizaje será de utilidad para futuros proyectos.
3. DESARROLLO DEL PROYECTO

3.1 ETAPA 1: DEFINICIÓN

El entregable final del proyecto está dirigido a los operarios agrícolas encargados de realizar el monitoreo de las trampas para plagas en el cultivo. En Colombia, el 59% de los trabajadores del sector floricultor son mujeres, el 80% está en un rango de edad de 35 a 57 años, el 60% son cabeza de hogar y solo el 27% ha culminado la educación secundaria (ASOCOLFLORES, 2019).

Según estos datos, se infiere que la persona que utilizará el producto a desarrollar no posee conocimientos acerca de aplicaciones móviles o dispositivos tecnológicos.

3.1.1 Requerimientos de usuario

A continuación, se presentan los requerimientos de usuario identificados a partir de la información obtenida. A cada uno se le asigna un peso según su importancia en un rango de 1 a 5, siendo 1 una baja importancia y 5 una alta importancia:

<table>
<thead>
<tr>
<th>REQUERIMIENTOS DE USUARIO</th>
<th>IMPORTANCIA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requerimientos básicos</td>
<td></td>
</tr>
<tr>
<td>Alta precisión (mayor a 85%)</td>
<td>5</td>
</tr>
<tr>
<td>Compatibilidad con múltiples versiones de Android</td>
<td>4</td>
</tr>
<tr>
<td>Requerimientos lineales</td>
<td></td>
</tr>
<tr>
<td>Facilidad de manejo de la aplicación móvil</td>
<td>4</td>
</tr>
<tr>
<td>Seguridad de la información</td>
<td>3</td>
</tr>
<tr>
<td>Alta velocidad de respuesta</td>
<td>4</td>
</tr>
<tr>
<td>Bajo consumo de memoria y espacio de almacenamiento</td>
<td>3</td>
</tr>
<tr>
<td>Requerimientos atractivos</td>
<td></td>
</tr>
<tr>
<td>Estética y orden en la interfaz gráfica</td>
<td>3</td>
</tr>
<tr>
<td>Orden en la presentación de la información generada</td>
<td>4</td>
</tr>
</tbody>
</table>

Fuente: Elaboración propia

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
Los requerimientos básicos hacen referencia a los requisitos fundamentales para que el producto cumpla el objetivo o función para la cual fue diseñado. Los requerimientos lineales son requisitos esperados por el cliente pero que, en caso de no cumplirse, no se va a ver afectado el funcionamiento o correcto desempeño del producto. Por último, están los requerimientos atractivos que son aquellos que el cliente no espera encontrar pero que le generarán un alto nivel de satisfacción.

### 3.1.2 Requerimientos técnicos

Teniendo en cuenta los requerimientos de usuario planteados previamente, se definen los requerimientos técnicos que contribuyen a satisfacer dichas necesidades de los usuarios. También se describen las unidades o propiedades de medida de cada requerimiento técnico y su dirección de mejora. Dentro de los requerimientos se encuentran inicialmente la precisión y la exactitud, con los cuales se podrá medir el desempeño del modelo de clasificación; la adaptabilidad da a conocer si será posible ejecutar la aplicación en múltiples versiones de Android; el tiempo de respuesta y la utilización de recursos hacen referencia al desempeño de la aplicación móvil en el dispositivo; la seguridad de la información hace referencia a la protección que se le da a la información generada; la densidad de objetos por pantalla y el uso de gráficos e indicadores hacen referencia a la interfaz gráfica de usuario de la aplicación y a la forma de presentar la información generada:

<table>
<thead>
<tr>
<th>REQUERIMIENTOS TÉCNICOS</th>
<th>UNIDAD DE MEDIDA</th>
<th>DIRECCIÓN DE MEJORA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precisión</td>
<td>[%]</td>
<td>▲</td>
</tr>
<tr>
<td>Exactitud</td>
<td>[%]</td>
<td>▲</td>
</tr>
<tr>
<td>Adaptabilidad</td>
<td>No aplica</td>
<td>No aplica</td>
</tr>
<tr>
<td>Tiempo de respuesta</td>
<td>[s]</td>
<td>▼</td>
</tr>
<tr>
<td>Utilización de recursos</td>
<td>[MB]</td>
<td>▼</td>
</tr>
<tr>
<td>Sistema de seguridad</td>
<td>No aplica</td>
<td>No aplica</td>
</tr>
<tr>
<td>Densidad de objetos por pantalla</td>
<td>[Cantidad objetos/área de pantalla]</td>
<td>▼</td>
</tr>
<tr>
<td>Uso de gráficos e indicadores</td>
<td>No aplica</td>
<td>No aplica</td>
</tr>
</tbody>
</table>

Fuente: Elaboración propia

La dirección de mejora en la Tabla 8 indica cual sería el mejor escenario para cada requerimiento (▲: Aumento, ▼: Reducción, ◇: No aplica).

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
3.1.3 Casa de la calidad

Usando los requerimientos de usuario y los requerimientos técnicos se construye la casa de la calidad que se presenta en el ANEXO A. Analizando los puntajes de importancia técnica presentados en la parte inferior de la casa de la calidad, se puede observar como el tiempo de respuesta y la utilización de recursos tienen los más altos puntajes, esto debido a que son requerimientos técnicos que permiten medir el desempeño de aspectos que dan respuesta a los requerimientos de usuario con mayor importancia. Por otro lado, el sistema de seguridad presenta el más bajo puntaje, indicando su baja influencia en el cumplimiento de los requerimientos exigidos por el usuario. También se muestra la matriz de relaciones entre requerimientos técnicos y requerimientos de usuario y la matriz de correlaciones de los requerimientos técnicos.

3.1.4 Arquitectura funcional

Se despliegan las funciones, procesos y actividades que se llevarán a cabo para alcanzar los objetivos propuestos.

Figura 9. Arquitectura funcional del proyecto
Fuente: Elaboración propia

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
3.1.5 Diagrama de caja transparente

A partir de la descomposición funcional presentada en la Figura 9 se construye el diagrama de caja transparente para la aplicación móvil, donde se puede observar la relación entre los diferentes procesos. Cada función está interconectada con las demás por medio de líneas de señal, energía o materia. En el diagrama se describe gráficamente el funcionamiento del producto a desarrollar a partir de unas entradas y salidas al sistema.

Figura 10. Descomposición funcional de la aplicación móvil
Fuente: Elaboración propia

3.2 ETAPA 2: INVESTIGACIÓN

Inicialmente, para el desarrollo del proyecto, se requiere obtener fotografías de las trampas instaladas en los cultivos de flores. Para lo cual, es importante evaluar las condiciones ambientales de la ubicación, ya que, de esta manera, se define si es necesaria una

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

La adecuación del espacio donde se realizará la adquisición de imágenes. También, es de suma importancia definir el dispositivo con el cual se tomarán las fotografías, teniendo en cuenta la calidad y la resolución de las imágenes, el precio del dispositivo, la forma como se enviará la imagen al host de la aplicación móvil y otros factores determinantes.

Para esto, se realizó una visita a los cultivos de flores de la empresa Flores El Trigal, sede Olas, donde se tuvo contacto directo con las trampas que se trabajarán y se pudo evaluar las condiciones ambientales de las mismas. Según lo anterior, se definió que no se requiere sistema de iluminación artificial ya que las trampas se encuentran ubicadas en campo bajo techo plástico transparente, lo que permite el paso de la luz natural, además se tienen algunas trampas ubicadas al aire libre.

En la visita realizada se llevaron múltiples dispositivos electrónicos para obtener capturas fotográficas de las trampas y de esa manera evaluar cuál sería la opción más adecuada para la aplicación requerida. Entre los dispositivos se encontraban: cámara industrial, cámara USB, cámara profesional y cámara de celular.

Posteriormente, analizando los antecedentes descritos en el marco contextual, donde se presentan trabajos que han tratado proyectos similares, se evalúan las métricas de precisión obtenidas y los insectos trabajados con el fin de definir cuál sería el procedimiento más optimo a seguir para lograr los objetivos establecidos.

En el marco teórico se presenta información relacionada con los posibles modelos de aprendizaje automático y aprendizaje profundo a implementar, así como también, información sobre las plagas a identificar.

3.3 ETAPA 3: IDEACIÓN

Con la información obtenida en la etapa de investigación se construye la matriz morfológica. En esta se presentan las posibles soluciones para cada función del proyecto y las alternativas de combinaciones entre ellas.

Para la primera función de la matriz morfológica, presentada en la Tabla 9, solo se incluyeron tres opciones de dispositivo para tomar la fotografía, ya que, luego de examinar las imágenes obtenidas en la visita al cultivo, se pudo evidenciar que las fotografías capturadas mediante estos tres dispositivos tenían mejor calidad.

En la cuarta función de la matriz, que corresponde a la clasificación de los insectos, se proponen tres soluciones que satisfacen la función planteada. El criterio de selección de estas soluciones es la precisión de clasificación de cada modelo, por lo que es necesario probar cada uno de ellos para medir su desempeño. Por esto, cada concepto tiene las tres opciones dentro de su combinación de soluciones.

Para la creación de la aplicación móvil se propone solo un entorno de desarrollo.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
3.3.1 Matriz Morfológica

Tabla 9
Matriz morfológica

<table>
<thead>
<tr>
<th>FUNCIÓN</th>
<th>SOLUCIÓN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Tomar fotografía</td>
<td>(ALKOSTO, 2020)</td>
</tr>
<tr>
<td></td>
<td>(LINIO, 2020)</td>
</tr>
<tr>
<td></td>
<td>(B&amp;H, 2020)</td>
</tr>
<tr>
<td>2 Cargar fotografía a la aplicación móvil</td>
<td>(ALKOSTO, 2020)</td>
</tr>
<tr>
<td></td>
<td>(ALKOMPRAR, 2020)</td>
</tr>
<tr>
<td>3 Pre-procesar la imagen</td>
<td>(Fiverr, 2020)</td>
</tr>
<tr>
<td>4 Clasificar insectos</td>
<td>Procesamiento digital (Fiverr, 2020)</td>
</tr>
<tr>
<td></td>
<td>Custom Vision (BlueGranite, 2018)</td>
</tr>
<tr>
<td></td>
<td>YOLO (Medium, 2018)</td>
</tr>
<tr>
<td>5 Mostrar información</td>
<td>(Quora, 2018)</td>
</tr>
<tr>
<td></td>
<td>(ALKOMPRAR, 2020)</td>
</tr>
</tbody>
</table>

Fuente: Elaboración propia

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
Tabla 10
Descripción de conceptos propuestos en la matriz morfológica

<table>
<thead>
<tr>
<th>CONCEPTO</th>
<th>DESCRIPCIÓN</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Se toma la fotografía por medio de un celular el cual tiene instalada la aplicación, por lo que se carga directamente la fotografía en esta. Se realiza el preprocesamiento de la imagen mediante la librería openCV, la cual está cargada en la aplicación. Posteriormente se realiza la clasificación mediante la técnica computacional con mejor desempeño y finalmente se muestra la información a través de una interfaz gráfica creada en Android Studio.</td>
</tr>
<tr>
<td>B</td>
<td>Se toma la fotografía por medio de una cámara profesional, luego se envía a un computador que desempeña el rol de host. Se realiza el preprocesamiento de la imagen mediante un algoritmo desarrollado en Python haciendo uso de la librería openCV, el cual está guardado en el host. Posteriormente se realiza la clasificación mediante la técnica computacional con mejor desempeño y finalmente se muestra la información directamente en el computador.</td>
</tr>
<tr>
<td>C</td>
<td>Se toma la fotografía por medio de una cámara industrial, posteriormente se envía a un equipo de cómputo, en el cual se realiza el preprocesamiento de la imagen usando un script creado en Python aplicando la librería openCV. Luego se realiza la clasificación mediante la técnica computacional que presenta el mayor desempeño y finalmente se muestra la información directamente en el computador.</td>
</tr>
</tbody>
</table>

Para elegir uno de los conceptos propuestos se deben tener en cuenta los requerimientos planteados por el usuario, eligiendo el que de una mayor satisfacción a sus necesidades. Para esto, es importante conocer la precisión de clasificación, el costo de implementación y otros criterios que permitirán tomar una decisión acertada.

Debido a lo anterior, se creará el algoritmo de preprocesamiento, se prototiparán las 3 opciones de solución de la cuarta función presentada en la Tabla 9 y se realizará una comparación para definir cual presenta un mejor desempeño.

Luego de tener la información completa de los criterios de selección se procederá a elegir el concepto que se implementará para dar solución a la problemática planteada.

Fuente: Elaboración propia

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
3.4 ETAPA 4: PROTOTIPADO

A continuación, se describe detalladamente el proceso llevado a cabo para el desarrollo de cada alternativa propuesta:

3.4.1 Procesamiento Digital de Imágenes

El procedimiento se desarrolló en el lenguaje de programación de código abierto Python, usando librerías como Numpy y OpenCV. El código creado recibe como entrada una o varias imágenes y la selección del color de la placa a trabajar, posteriormente realiza una serie de funciones que modifican la imagen, entregando como salidas: la cantidad de insectos identificados y una imagen en la cual estos son señalados.

El código se divide en tres secciones principales, inicia con la importación de librerías: numpy, cv2 y glob, luego la definición de funciones y, por último, el main o programa principal del código, el cual contiene la secuencia de pasos, donde se invocan las funciones o actividades que se llevarán a cabo. El código es posible ejecutarlo para una o varias imágenes (esta última con la librería Glob), a continuación, se describe el proceso para el tratamiento de una sola imagen, este también se presenta en el diagrama de la Figura 11.

![Diagrama de Procesamiento Digital de Imágenes](image)

**Figura 11.** Diagrama de Procesamiento Digital de Imágenes.
Fuente: Elaboración propia

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
Inicialmente, como se muestra en la Figura 12, se solicita a través de Python Shell la selección del color de la trampa con la cual se trabajará.

**Menú principal:**
1. Placa amarilla
2. Placa azul

¿Qué color de placa trabajará? (Escriba el número):

Figura 12. Selección de color para el procesamiento digital de imágenes
Fuente: Elaboración propia

Seguido se esto se carga la imagen a procesar con el método `cv2.imread`. La imagen cargada junto con el color seleccionado se usa como parámetros de entrada de la primera función, en la cual, se cambia el espacio de color de la imagen de BGR a HSV con la instrucción `cv2.cvtColor` para luego realizar una binarización mediante `cv2.inRange`. En este último se debe ingresar un vector con el límite inferior de color y otro con el límite superior, los cuales se usarán como umbrales para la binarización.

La siguiente función consiste en eliminar el ruido enmarcado por la trampa, el cual se genera debido a los insectos capturados. Esto con el fin de obtener una máscara con la forma y el tamaño de la placa para delimitar el área en la cual se trabajará. A la función le ingresa como parámetro la imagen binarizada, en esta se encuentran los contornos de los insectos ubicados en el centro de la placa y se “rellenan”. Lo anterior se realiza mediante los métodos: `cv2.findContours` y `cv2.boundingRect`.

A la imagen obtenida se le aplica el filtro de la mediana mediante `cv2.medianBlur` con el objetivo de eliminar el ruido restante en los bordes de la placa, pues, hasta el momento, las operaciones de esta función solo han actuado en el sector central de la trampa. Una vez aplicado el filtro se obtiene la máscara deseada, la cual se utiliza para recortar la imagen con determinado desfase para luego escalarla a las dimensiones originales, lo que genera un efecto “Zoom”. En la Figura 13 se muestra el proceso realizado en cada paso mediante las imágenes de salida de cada función descrita.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

La siguiente función consiste en eliminar los bordes que quedan en la imagen luego de hacer el acercamiento. Para esto, se realiza el mismo efecto “Zoom” descrito anteriormente a la imagen que contiene la máscara, la cual se muestra en la Figura 13c. A esta se le aplica el filtro de la mediana hasta que se tenga una sola contorno (la trampa) en la imagen para eliminar posibles puntos que generen ruido. Luego esta imagen es usada como máscara para la imagen a la que se le desean eliminar los bordes, la cual se presenta en la Figura 13d.

Las funciones anteriores se enfocan en procesar la imagen para obtener la región de interés donde se identificarán los insectos, ya que, al momento de tomar la fotografía, esta puede contener un fondo que interfiera con la correcta identificación.

 Una vez obtenida la región de interés, se procesa a cambiar el espacio de color de la imagen de BGR a GRAY para posteriormente realizar una nueva binarización usando cv2.threshold. A la imagen binarizada se le aplican transformaciones morfológicas para eliminar el ruido generado por partículas de polvo que pueden confundirse con insectos. Inicialmente se aplica erosión, mediante cv2.erode, para anular los detalles pequeños y posteriormente se aplica dilatación, mediante cv2.dilate, para aumentar el tamaño de las ventanas que corresponden a los insectos identificados. Para cada transformación morfológica se utiliza un kernel y un número de iteraciones distinto. En las trampas amarillas, inicialmente se usa una erosión con un kernel elíptico de dimensiones (3,4) el cual se aplica 3 iteraciones seguidas, posteriormente se aplica otra erosión, esta vez las dimensiones del kernel son de (3,2) con 2 iteraciones y finalmente se aplica una dilatación con kernel elíptico de dimensiones (5,5) y 15 iteraciones. Por otro lado, para las trampas azules, se usan dos erosiones iniciales, la primera con dimensiones de kernel elíptico iguales a (2,3) y la segunda con (4,2), y por último una dilatación con kernel de (4,3) y 7 iteraciones.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
Finalmente, se tiene una función que dibuja los contornos de los insectos con el método `cv2.drawContours`, señalando su ubicación dentro de la imagen. Esta función tiene como salidas la cantidad de insectos identificados y una imagen con los contornos de los insectos dibujados, la cual se muestra en la Figura 14d.

![Figura 14](image)

**Figura 14.** Procesamiento digital hasta identificación de insectos

a) Imagen acercada sin bordes  
   b) Imagen acercada sin bordes en espacio de color GRAY  
   c) Imagen binarizada después de aplicar transformaciones morfológicas  
   d) Imagen con insectos identificados señalados con contorno rojo.

Fuente: Elaboración propia

En los modelos de inteligencia artificial se usan dos conjuntos de datos, uno para entrenamiento y otro para prueba. El objetivo de este proceso es generar una predicción en los datos de prueba con un alto porcentaje de precisión. Para lograr esto último, se debe evitar ajustar demasiado el modelo a los datos de entrenamiento, ya que existe el riesgo de que el modelo se ajuste al ruido en estos datos, memorizando particularidades, en lugar de encontrar una regla general de predicción. Este fenómeno se conoce como “Overfitting” (Tom Dietterich, 1995).

Con el fin de evitar este efecto, se capturaron más fotografías para el entrenamiento de los modelos, 75 de trampas amarillas y 39 de trampas azules. Además, se realizó un incremento en el conjunto de datos mediante la modificación de las imágenes originales. Las transformaciones consistieron en girar cada fotografía una cantidad de grados específica, obteniendo un total de 828 imágenes, 522 de trampas amarillas y 276 de trampas azules. El desequilibrio en la cantidad de imágenes por tipo de trampa se debe a que, en las trampas azules, en las cuales se capturan Trips, al momento de la adquisición de imágenes, la población capturada de estos insectos fue muy baja. Posteriormente, se eliminaron las imágenes no comprensibles y las imágenes de baja calidad.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
3.4.2 Custom Vision

Antes de iniciar el proceso con la herramienta de Microsoft Azure, se realizó un redimensionamiento de las imágenes obtenidas mediante un script en Python usando la función `cv2.resize`, ya que, en promedio, el tamaño de estas es de 22MB y la plataforma no permite imágenes de entrenamiento que superen los 6MB.

Una vez redimensionadas las imágenes, estas se cargaron en el sistema y se inició el proceso de etiquetado, para el cual, además de las etiquetas “Trips”, “MoscaBlanca” y “Minador”, se usaron otras etiquetas para identificar elementos comunes presentes en las trampas como “Mosca Tigre”, “Afido” y “Gota”. Es importante tener en cuenta que la Mosca Blanca y el Minador solo se etiquetaron sobre trampas amarillas y de la misma manera el Trips solo se etiquetó sobre trampas azules. Esto debido a que, en caso de identificarse uno de estos insectos en la trampa contraria, no se cuenta como lectura según lo especificó la persona encargada del monitoreo de plagas en los cultivos de Flores El Trigal. Al etiquetar estos insectos sólo sobre uno de los tipos de trampa, el modelo relacionará el color de fondo del escenario con el insecto etiquetado, prediciendo su presencia solo sobre la trampa relacionada.

Luego de realizar el etiquetado se procede a entrenar el modelo, lo cual solo toma un par de minutos. Posteriormente, la plataforma arroja los resultados obtenidos, los cuales se muestran en la Figura 15 de manera general y en la Figura 16 se muestran los resultados por etiqueta, además, se habilita la pestaña Predicciones para probar la precisión alcanzada en imágenes de prueba. En la Figura 17 se puede observar la interfaz de prueba de Custom Vision donde se señalan los insectos identificados sobre la trampa. En esta se tiene un control deslizante con el cual es posible variar el valor del umbral de confianza de los insectos detectados.

![Figura 15. Resultados generales del modelo creado en Custom Vision](image)

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
3.4.3 YOLO

Para el proceso de etiquetado de este modelo, se usa un script creado en Python, el cual genera un archivo de texto por cada imagen, con la información de cada etiqueta. Luego se copian todas las imágenes tanto de trampas amarillas como azules en una carpeta específica dentro del directorio de Darknet creado en el equipo, al igual que con los archivos de texto generados para cada imagen. Seguido de esto, se crear un archivo .txt con las direcciones de todas las imágenes etiquetadas; un archivo .data con información relevante para el proceso de entrenamiento como el número de clases, la dirección del archivo .txt mencionado anteriormente, la dirección de la carpeta backup donde se guardarán los pesos calculados cada 1000 iteraciones, etc.; y un archivo .names con los nombres de las etiquetas a utilizar. Además, se descargan dos archivos de internet: un archivo .weights con los pesos pre-entrenados para las capas convolucionales del modelo y un archivo .cfg con la configuración específica para el modelo a entrenar. El procedimiento descrito se presenta en el esquema de la Figura 18. En la Figura 19 se muestra la interfaz para el etiquetado ocasionada por el script.

Figura 18. Esquema de procedimiento para entrenamiento YOLO
Fuente: Elaboración propia

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
Para el proyecto se entrenaron dos modelos YOLO, usando las mismas imágenes etiquetadas, pero con diferentes archivos .cfg. La modificación de los archivos .cfg se realizó con base en el repositorio de Alexey AB en GitHub (AlexeyAB, 2020).

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
<table>
<thead>
<tr>
<th>Parámetro</th>
<th>Modelo 1 YOLO</th>
<th>Modelo 2 YOLO</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parámetros de configuración</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>batch</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>subdivisions</td>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>width</td>
<td>416</td>
<td>512</td>
</tr>
<tr>
<td>height</td>
<td>416</td>
<td>512</td>
</tr>
<tr>
<td>channels</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>momentum</td>
<td>0.9</td>
<td>0.9</td>
</tr>
<tr>
<td>decay</td>
<td>0.0005</td>
<td>0.0005</td>
</tr>
<tr>
<td>angle</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>saturation</td>
<td>1.5</td>
<td>1.2</td>
</tr>
<tr>
<td>exposure</td>
<td>1.5</td>
<td>1.2</td>
</tr>
<tr>
<td>hue</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>learning_rate</td>
<td>0.00261</td>
<td>0.001</td>
</tr>
<tr>
<td>burn_in</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>max_batches</td>
<td>8000</td>
<td>13100</td>
</tr>
<tr>
<td>policy</td>
<td>steps steps</td>
<td>steps steps</td>
</tr>
<tr>
<td>steps</td>
<td>6400,7200</td>
<td>4000,7000,10000</td>
</tr>
<tr>
<td>scales</td>
<td>.1,.1</td>
<td>.1,.1,.1</td>
</tr>
</tbody>
</table>

| **Filtro en la capa convolucional precedente a las capas de salida YOLO** |
|---------------------------|-------------|-------------|
| filter                    | 33          | 55          |

| **Capas de salida YOLO** |
|--------------------------|-------------|
| **Capa 1 YOLO**          |
| mask                     | 3,4,5       | 0,1,2,3,4   |
| anchors                  | 7, 10, 11, 11, 9, 15, 12, 18, 15, 15, 19, 24 | 9,13, 15,19, 21,25, 27,32, 33,38 |

<table>
<thead>
<tr>
<th><strong>Capa 2 YOLO</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>mask</td>
</tr>
<tr>
<td>anchors</td>
</tr>
</tbody>
</table>

Fuente: Elaboración propia

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

En la Tabla 11 se presentan algunos parámetros de configuración del archivo .cfg modificados para cada modelo. En el Modelo 2 se ajustan parámetros respecto al Modelo 1, teniendo en cuenta el tamaño reducido de los objetos a identificar y el tamaño de las imágenes. Además, el segundo modelo se entrena en un equipo con mayor capacidad, por lo cual también se ajustan algunos parámetros de configuración relacionados.

Una vez finalizado el entrenamiento se prueban los modelos mediante un script de Python o mediante la consola del computador.

Figura 20. Prueba de modelos YOLO en la misma imagen
a) Prueba de Modelo 1 YOLO en trampa amarilla b) Prueba de modelo 2 YOLO en la misma trampa amarilla.
Fuente: Elaboración propia

Para cada modelo YOLO entrenado se obtienen las métricas de desempeño generales Precisión, Recall y mAP, y específicas por cada etiqueta AP. En el Código 1 y Código 2 se presentan los resultados del Modelo 1 YOLO y del Modelo 2 YOLO respectivamente.

Los resultados mostrados por cada modelo dan información sobre las capas de la red neuronal en las cuales se encuentra la función YOLO (Detection Layer), sobre la precisión promedio (AP) por cada clase etiquetada en las imágenes, las predicciones verdaderas (TP) y predicciones falsas (FP) por clase, etc.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
3.5 ETAPA 5: SELECCIÓN

En esta etapa se realiza la comparación de los resultados obtenidos en cada prototipo propuesto, con el objetivo de seleccionar el de mejor desempeño. Este último pasará a la etapa de implementación.

El Procesamiento Digital de Imágenes es el primer prototipo propuesto, con el cual puede obtenerse una aproximación de la cantidad de insectos capturados en las trampas adhesivas. El prototipo realiza el conteo basándose en el color y tamaño de los insectos, lo cual no permite realizar una diferenciación entre plagas de interés como el Minador de hoja y otros insectos que puedan presentar características morfológicas similares. Debido a esto, se descarta el prototipo como alternativa solución.

Para los demás prototipos, se realiza una comparación basada en las métricas Precision, Recall y mAP (mean Average Precision), la cual se muestra en la Tabla 12, donde la Precision se define como la porción de insectos identificados correctamente del total de insectos encontrados en la imagen; por otro lado, el Recall indica, del total de insectos capturados en la trampa, que porción de ellos fue identificada digitalmente. Y, por último, el mAP representa el comportamiento general del modelo a través de todas las etiquetas. Para definir cuál es el modelo de mejor desempeño, se debe definir cuál es la métrica más representativa acorde a los objetivos del proyecto.

Tabla 12
<table>
<thead>
<tr>
<th>Modelo  \ Métrica</th>
<th>Precision (%)</th>
<th>Recall (%)</th>
<th>mAP (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom Vision</td>
<td>85.10</td>
<td>21.50</td>
<td>51.70</td>
</tr>
<tr>
<td>Modelo 1 YOLO</td>
<td>85.00</td>
<td>84.00</td>
<td>83.62</td>
</tr>
<tr>
<td>Modelo 2 YOLO</td>
<td>46.00</td>
<td>70.00</td>
<td>45.88</td>
</tr>
</tbody>
</table>

Fuente: Elaboración propia

Además de las métricas de desempeño, es importante presentar una comparación en el tiempo que toma entrenar cada modelo. Por un lado, respecto al etiquetado de las imágenes, tanto para YOLO como para Custom Vision se dedicó el mismo tiempo en esta tarea, mientras que, respecto al tiempo de entrenamiento, Custom Vision se toma solo un par de minutos (entre 5 – 10 minutos) mientras YOLO se toma varios días (dependiendo de los parámetros dados en el archivo .cfg y las especificaciones del equipo en el cual se entrene el modelo).

Teniendo en cuenta que lo que se busca es automatizar el proceso de identificación y conteo de insectos, suprimiendo la necesidad de capacitar el personal en monitoreo de plagas, se determina que se superpone el Recall sobre las otras métricas de desempeño, ya que, prevalece la correcta identificación de la mayoría de los insectos capturados en la trampa sobre la correcta identificación de los insectos detectados digitalmente por el

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
modelo, sin perder de vista el comportamiento general de este último. Según lo anterior, se elige el Modelo 1 YOLO para continuar a la etapa de implementación.

3.6 ETAPA 6: IMPLEMENTACIÓN

Para la implementación del desarrollo se crea una aplicación móvil en el software Android Studio para que el usuario pueda ejecutar, en el lugar de identificación del cultivo y en tiempo real, el modelo seleccionado en la etapa anterior.

3.6.1 Desarrollo de aplicación móvil

La aplicación consiste en una interfaz donde el usuario puede visualizar, en la imagen transmitida por la cámara, los insectos identificados enmarcados en un color diferente según su clasificación. Además, en la parte inferior de la pantalla se muestra la cantidad de insectos identificada por cada clase.

En la aplicación creada solo se está mostrando la información de interés para el usuario, pero esta no está siendo almacenada, ya que se hizo a manera de muestra representativa, pero esto si debe estar contemplado en la creación de la aplicación oficial. En el diagrama de la Figura 21 se muestra el procedimiento realizado por uno de los códigos de la aplicación, el cual se encarga de la ejecución del modelo seleccionado. Para dar inicio a la identificación se debe presionar el botón “DETECTAR” que aparece en pantalla. La interfaz gráfica de usuario se muestra en la Figura 22.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

3.6.2 Retorno de la inversión

Para el caso específico de Flores El Trigal sede Olas, actualmente, en el proceso de monitoreo de plagas, a la persona encargada le toma más o menos 1 minuto la lectura de cada placa, dependiendo de la cantidad de insectos capturados en esta. Las placas se leen directamente en el lugar donde se encuentran ubicadas en el campo. En este cultivo se tienen 50 bloques y en cada uno puede haber de 100 a 200 camas. Las trampas se ubican por pares (una amarilla y una azul) cada 12 de camas. Lo que indica que cada vez que se monitorea se deben leer entre 834 y 1666 trampas. El monitoreo en este cultivo se realiza una vez por semana.

Según lo anterior, la empresa está dedicando de 56 a 111 horas mensuales al monitoreo de plagas, teniendo en cuenta solo el tiempo de lectura de las trampas y no el tiempo que le toma a la persona desplazarse entre la ubicación de una trampa y otra. Mientras que, con el uso de la aplicación, puede reducirse drásticamente el tiempo de lectura ya que identifica y cuantifica los insectos en tiempo real, tardando solo unos segundos que le toman a la persona enfocar la trampa. Para el cálculo del retorno de la inversión, se usará el valor

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
de la horas ahorradas en esta labor como el ingreso y como costos se usará el valor obtenido al cotizar el desarrollo con una empresa de software. El ingreso (ahorro) generado por el desarrollo o entregable del proyecto, se calculó usando el valor de la hora laboral correspondiente al salario mínimo legal vigente en Colombia que equivale a 3.658 pesos. Por otro lado, para el costo, se cotizó con una empresa de software el desarrollo de un aplicativo que cumpla con todos los requerimientos de usuario a cabalidad. El valor de dicho aplicativo junto con el detalle de sus rubros se presenta en la Tabla 13.

### Tabla 13
Cotización desarrollo de aplicativo

<table>
<thead>
<tr>
<th>Perfil Profesional</th>
<th>Actividades</th>
<th>Precio Hora</th>
<th>Horas Dedicadas</th>
<th>Valor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Desarrollador Junior</td>
<td>Frontend</td>
<td>$ 24.000</td>
<td>142</td>
<td>$ 3.408.000</td>
</tr>
<tr>
<td>Desarrollador Senior</td>
<td>Backend</td>
<td>$ 38.000</td>
<td>142</td>
<td>$ 5.396.000</td>
</tr>
<tr>
<td>Especialista AI</td>
<td>Diseño, entrenamiento y validación de modelos</td>
<td>$ 80.000</td>
<td>48</td>
<td>$ 3.840.000</td>
</tr>
<tr>
<td>Etiquetadores x 2</td>
<td>Etiquetado de imágenes</td>
<td>$ 8.000</td>
<td>48</td>
<td>$ 384.000</td>
</tr>
<tr>
<td>Arquitecto de Software</td>
<td>Revisar arquitectura, código y despliegue</td>
<td>$ 54.000</td>
<td>42,6</td>
<td>$ 2.300.400</td>
</tr>
<tr>
<td>Validador de Apps</td>
<td>Validar funcionamiento del front y el back</td>
<td>$ 34.000</td>
<td>28,4</td>
<td>$ 965.600</td>
</tr>
<tr>
<td>Director</td>
<td>Coordinar actividades y reuniones con el cliente final</td>
<td>$ 48.000</td>
<td>12</td>
<td>$ 576.000</td>
</tr>
</tbody>
</table>

**Subtotal** $ 16.870.000

**Imprevistos** $ 3.374.000

**Administración** $ 1.687.000

**Total** $ 21.931.000

Fuente: Tomado de Cotización enviada por empresa de Software

Según los cálculos y valores mencionados anteriormente se define que, a la empresa Flores El Trigal le costaría $ 21.931.000 el desarrollo del aplicativo propuesto y le tomaría 5.9 años aproximadamente recuperar el valor de la inversión realizada. Esto, teniendo en cuenta solo los ingresos (ahorros) obtenidos debido al uso de la aplicación. Además, se ahorraría entre 56 y 111 horas mensuales dedicadas al monitoreo, que ahora pueden ser dedicadas a otras labores que generen valor para la empresa.

### 3.7 ETAPA 7: APRENDIZAJE

Se presenta en el Capítulo 4 y Capítulo 5.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
4. DISCUSIÓN DE RESULTADOS

Analizando los requerimientos de usuario planteados inicialmente en comparación con el prototipo desarrollado, como se muestra en la Tabla 14, se determina que se da cumplimiento al 100% de los requerimientos de usuario. Sin embargo, el requerimiento “Alta precisión”, aunque se logró el desempeño esperado del modelo al momento de ejecutarse, se considera que requiere una mejora en su nivel de precisión, lo cual podría lograrse modificando parámetros de entrenamiento o las imágenes y etiquetas utilizadas.

Por otro lado, si en el aplicativo se incluye la opción de almacenar la información generada en la nube, se requeriría un sistema de seguridad especializado para proteger la información.

Tabla 14
Cumplimiento de los requerimientos de usuario

<table>
<thead>
<tr>
<th>REQUERIMIENTOS DE USUARIO</th>
<th>PROTOTIPO DESARROLLADO</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Requerimientos básicos</strong></td>
<td></td>
</tr>
<tr>
<td>Alta precisión (mayor a 85%)</td>
<td>Cumple (Requiere mejora)</td>
</tr>
<tr>
<td>Compatibilidad con múltiples versiones de Android</td>
<td>Cumple</td>
</tr>
<tr>
<td><strong>Requerimientos lineales</strong></td>
<td></td>
</tr>
<tr>
<td>Facilidad de manejo de la aplicación móvil</td>
<td>Cumple</td>
</tr>
<tr>
<td>Seguridad de la información</td>
<td>Cumple</td>
</tr>
<tr>
<td>Alta velocidad de respuesta</td>
<td>Cumple</td>
</tr>
<tr>
<td>Bajo consumo de memoria y espacio de almacenamiento</td>
<td>Cumple</td>
</tr>
<tr>
<td><strong>Requerimientos atractivos</strong></td>
<td></td>
</tr>
<tr>
<td>Estética y orden en la interfaz gráfica</td>
<td>Cumple</td>
</tr>
<tr>
<td>Orden en la presentación de la información generada</td>
<td>Cumple</td>
</tr>
</tbody>
</table>

Fuente: Elaboración propia

La deficiencia en la precisión puede darse debido a múltiples factores. Uno de ellos es que, al momento de etiquetar las imágenes, alrededor de los insectos enmarcados, queda mucho ruido correspondiente a polvo, restos de plantas, otros insectos, etc., lo que hace que el modelo no converja de la mejor manera. Esto puede evidenciarse en la Figura 23, donde se identificaron manualmente 2 Trips, los cuales están señalados con círculos rojos, alrededor de estos hay gran cantidad de ruido, lo cual dificulta su identificación de manera automática en la imagen.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

Figura 23. Ruido en trampa azul
a) En la imagen se señalan dos Trips identificados de forma manual, alrededor de los cuales se presenta gran cantidad de ruido debido a basuras llevadas por el viento hasta la trampa.

b) Acercamiento de la imagen a) enfocando Trips señalados.

Fuente: Elaboración propia

Otro factor que puede influir en la precisión de identificación es la calidad de las imágenes, ya que, al ser adquiridas mediante la lente de un celular de gama media, no tienen la mejor calidad. Esto hace que en algunas imágenes no pueda discriminarse entre un insecto y otro, debido a que las características físicas que ayudan en su diferenciación no son perceptibles. En la Figura 24 se observa un caso en el cual no es posible discriminar los insectos de la parte izquierda de la imagen debido a su calidad. Este efecto puede darse por cuestiones del enfoque establecido en el celular al momento de tomar las fotografías.

Estos factores asociados a las imágenes utilizadas para el etiquetado y entrenamiento de los modelos son los que requieren de atención para mejorar el porcentaje de precisión alcanzado.
Por otro lado, es importante destacar algunas características del prototipo desarrollado, las cuales se mencionan a continuación:

- Al momento de desarrollar la aplicación se definió Android 9 (Nivel API 28) como plataforma objetivo del desarrollo. Sin embargo, la aplicación es funcional en dispositivos con Android 4.3 (Nivel API 15) en adelante. Lo cual incluye todas las versiones iguales o superiores a Jelly Bean (versiones de Android entre 4.1 y 4.3).

- Aunque en el desarrollo de la aplicación se determinó enmarcar las clases “Minador”, “Trips” y “Mosca Blanca” en recuadros de color, el aplicativo también está en la capacidad de detectar “Mosca Tigre”, “Afidos” y “Gotas”, debido a que fueron elementos etiquetados durante el proceso de entrenamiento de los modelos. Esto con el fin de que se identificaran otros elementos diferentes a los insectos objetivos y se encontrara diferencias entre ellos para luego no generar confusiones al momento de detectarlos en tiempo real.

- La aplicación desarrollada no está publicada en ninguna AppStore. Para su instalación en el dispositivo, se debe descargar el archivo APK y ejecutarlo. Además, se debe tener en cuenta que, para su correcto funcionamiento, se debe crear una carpeta en la memoria interna del dispositivo nombrada “

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
4.1 DISEÑO DE EXPERIMENTO

Cómo se mencionó anteriormente, la calidad de las imágenes es un factor clave en la precisión de predicción del modelo implementado, por esto, se sugiere llevar a cabo un diseño de experimento para determinar cuál es la manera adecuada de adquirir las imágenes, tanto para validar el modelo como para entrenar uno nuevo. Douglas C. Montgomery en su libro Diseño y análisis de experimentos (Montgomery, 2005) plantea unas pautas generales para el diseño de experimentos, las cuales se presentan en la Tabla 15.

| Tabla 15 |
Pautas generales para diseñar un experimento |

1. Identificación y exposición del problema.

2. Elección de los factores, los niveles y los rangos

3. Selección de la variable de respuesta

4. Elección del diseño experimental.

5. Realización del experimento.

6. Análisis estadístico de los datos.

7. Conclusiones y recomendaciones.

Fuente: Tomado del libro Diseño y análisis de experimentos (Montgomery, 2005)

4.1.1 Identificación y exposición del problema

Debido a la sensibilidad del sistema implementado ante la calidad de las imágenes de entrada, se ha determinado que es de suma importancia llevar a cabo un experimento que permita definir cuáles son los factores que tienen un mayor impacto en la calidad de las imágenes al momento de su adquisición. El hecho de que las fotografías sean adquiridas en campo, en un ambiente no controlado, da paso a que la calidad de las imágenes obtenidas pueda ser influenciada por diversos factores, tanto del entorno donde se encuentran ubicadas las trampas como de la persona encargada de adquirir las fotografías de estas.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
4.1.2 Elección de los factores, los niveles y los rangos

De los factores que pueden influir en el desempeño del sistema se definen como factores potenciales de diseño: la iluminación y la distancia entre el lente y la trampa adhesiva. Para estos factores se definen a continuación los niveles específicos con los que se realizarán las corridas del experimento, así como los rangos en los que variarán estos factores.

Iluminación

Como se presentó en el marco teórico, la técnica de iluminación al momento de adquirir las imágenes puede mejorar la calidad, así como la distinción de los detalles de esta. De las técnicas allí propuestas se define la iluminación coaxial como la técnica adecuada para este caso debido al material de las trampas, ya que presentan una superficie reflectante. Esta técnica de iluminación consiste en emitir la luz de manera lateral sobre un espejo semitransparente que desvía los haces de luz en la dirección del eje de la lente como se muestra en la Figura 25.

![Figura 25. Sistema de iluminación coaxial](image)

Fuente: Tomado y modificado de (DACHS-SOLUTIONS, 2019)

Para la implementación de esta técnica de iluminación se propone diseñar un sistema que pueda adaptarse al dispositivo móvil con el cual se van a adquirir las imágenes.

Los niveles definidos para este factor son: sistema de iluminación artificial e iluminación natural.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
Distancia

Las diferentes distancias focales, como se mencionó en el marco teórico, crean diferentes niveles de aumento y cambian el ángulo de visión de la fotografía resultante, esto debido a que varían la profundidad de campo. La distancia focal depende del lente del dispositivo con el cual se capturan las imágenes, pero la profundidad de campo puede modificarse también mediante la distancia de enfoque (GCF Global, 2020), la cual representa la distancia entre el lente y el objeto a fotografiar.

Para llevar a cabo el experimento se proponen dos niveles para este factor, una distancia de enfoque corta y una más prolongada. Los rangos para estas distancias son 10cm para el nivel corto y 15cm para el nivel prolongado, están son distancias aproximadas ya que este factor va a depender de la distancia focal del dispositivo móvil. Los rangos propuestos se definen con base en las imágenes capturadas con un dispositivo móvil gama media, en la cual en el nivel corto se captura la trampa de manera ajustada, mientras en el nivel prolongado se captura parte del fondo alrededor de la trampa.

4.1.3 Selección de la variable de respuesta

Al modificar los factores de diseño propuestos para el experimento, se pretende determinar cómo estos influyen en la calidad de las imágenes. Para medir la calidad de las imágenes se han propuesto múltiples métodos tanto objetivos como subjetivos. Los métodos subjetivos requieren personal que examine las imágenes para determinar su calidad, lo cual es un proceso lento y costoso (Thung & Raveendran, 2009). Por otro lado, los métodos objetivos computan las imágenes obteniendo su métrica de calidad automáticamente (Thung & Raveendran, 2009). Para llevar a cabo un análisis de calidad de imágenes objetivo se debe tener una referencia que le indique al algoritmo lo que se considera una imagen de alta calidad y este realizará una comparación de las demás imágenes con la referencia, arrojando un valor entre 0 y 1, siendo 0 una mala calidad en comparación con la referencia y 1 una alta calidad.

En el pasado se han desarrollado múltiples métodos objetivos basados en el sistema de visión humano (HVS), similitud estructural (SSIM), proporción máxima de señal a ruido (PSNR), etc. Debido a la naturaleza de las imágenes tratadas en este proyecto y al objetivo del experimento, se busca un método objetivo que permita medir la calidad de las imágenes según el enfoque de los detalles, permitiendo diferenciar las imágenes que tengan un enfoque adecuado y alta nitidez de las que presentan distorsión o contengan zonas difuminadas. Para esto, se define el método de similitud estructural basado en bordes (ESSIM), el cual es un método basado en SSIM mejorando la respuesta ante imágenes difusas (Chen et al., 2006). El resultado obtenido de este algoritmo será la variable de respuesta del experimento.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
4.1.4 Elección del diseño experimental

Los diseños factoriales son los más eficientes para experimentos en los cuales intervienen los efectos de dos o más factores (Montgomery, 2005), como es este caso. Un diseño factorial es requerido cuando se presentan interacciones entre los factores del experimento, con el fin de evitar llegar a conclusiones incorrectas o sesgadas. Estos diseños permiten analizar los efectos de un factor con varios niveles de los factores restantes, generando conclusiones que son válidas para un rango de condiciones experimentales.

Debido a que se proponen dos factores para el experimento y a que cada uno tiene dos niveles, se define el diseño factorial de dos factores o diseño factorial $2^2$ como diseño experimental. En este paso del diseño experimental es importante definir el tamaño de la muestra (número de repicas), la selección de un orden de corridas y la determinación de algunas configuraciones como la formación de bloques y las restricciones de aleatorización.

Para determinar el número de muestra, Montgomery sugiere apoyarse en las curvas de operación característica que presenta en su libro antes mencionado. Para usar estas curvas se debe hallar previamente el valor de algunos parámetros que dependen de las medias de dos tratamientos, los niveles de cada factor, los efectos de interacción entre factores, etc. Una vez se aplican estos parámetros en las curvas, se puede obtener el número de repicas y su riesgo o probabilidad de rechazar la hipótesis nula.

Debido a cuestiones de tiempo, no será posible llevar a cabo el desarrollo del experimento en este proyecto, por esto se deja planteado para posibles futuros trabajos. A continuación, se mencionan los pasos siguientes para el diseño del experimento.

4.1.5 Realización del experimento

Al momento de llevar a cabo el experimento es fundamental monitorear todo el proceso con el fin de verificar de que se esté realizando conforme a lo planeado. Montgomery sugiere que antes de iniciar el experimento se lleven a cabo algunas pruebas para validar la consistencia del material experimental, el sistema de medición, etc.

4.1.6 Análisis estadístico de los datos

Para analizar los datos obtenidos en el experimento se deben usar métodos estadísticos que agreguen objetividad a los resultados, las conclusiones y las decisiones tomadas a partir de estos.

4.1.7 Conclusiones y recomendaciones

Una vez analizados los datos, se debe definir el curso de acción, además de realizar algunas corridas de seguimiento o pruebas de confirmación que permitan validar las conclusiones del experimento.

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
5. CONCLUSIONES Y CONSIDERACIONES FINALES

El uso de la plataforma Custom Vision, aunque simplifica una gran parte del proceso de entrenamiento de modelos, presenta desventajas frente a estos al no tener acceso al código, limitando las posibilidades de configurar parámetros y variables dependientes de las imágenes y de los objetos de interés dentro de ellas, lo que impide realizar mejoras específicas y personalizadas en el modelo entrenado. Para usar la versión gratuita de Custom Vision se debe registrar una tarjeta de crédito lo cual es una desventaja ya que no todas las personas cuentan con una, además esta versión presenta algunas limitaciones como la cantidad de imágenes de entrenamiento permitidas y su resolución.

El proceso de entrenamiento de modelos mediante YOLO permite a los desarrolladores variar múltiples parámetros ajustándolos a las necesidades y requerimientos de cada proyecto, por lo cual es posible obtener mejores resultados, reflejándose en las métricas obtenidas, siendo un sistema totalmente gratuito y de código abierto. Sin embargo, es un proceso que demanda más tiempo y conocimientos para programar los algoritmos necesarios, además de capacidades elevadas en redes neuronales para configurar los parámetros adecuadamente según las entradas del modelo y según lo que se desee obtener a la salida, tardando varios días en su entrenamiento.

Una muestra de las propiedades de sistema YOLO mencionadas anteriormente, es cómo el Modelo 2 YOLO, el cual tuvo modificaciones en los parámetros respecto al Modelo 1 YOLO, presentó resultados inferiores luego de que se ajustó más específicamente al tamaño de las imágenes de entrada y a las características de los objetos a detectar. La variación en el rendimiento del modelo se da debido a que también se realizaron modificaciones en las capas de la red neuronal. Además, es importante mencionar que el modelo que tuvo mejores resultados se configuró con una ventana menos específica, lo que le permitió una mayor generalización de las imágenes de entrada, presentando un mejor desempeño en la detección.

Mediante el uso del aplicativo propuesto, se logra una reducción en el tiempo que le toma a la persona encargada leer las trampas, ya que la aplicación permite hacerlo en tiempo real. Además, la empresa ya no tendría que capacitar estas personas para realizar dicha actividad. Otro beneficio que ofrece este desarrollo es la posibilidad de almacenar la información directamente en la nube, lo que le permitirá a la empresa dar un paso importante en la transformación digital y la cultura del dato, dos temas que se encuentran en auge debido a la cuarta revolución industrial. Esto abre también la posibilidad de llevar un registro histórico digital, de crear indicadores que ayuden a la toma de decisiones de la empresa y a tener documentado el comportamiento de las poblaciones de los insectos de interés tanto en el cultivo completo como por camas o por bloques.

En comparación con los estudios presentados en el marco contextual, se observa que la precisión de predicción alcanzada se posiciona aproximadamente entre 0% y 14% por debajo, lo cual puede deberse a múltiples factores como la calidad de las imágenes

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.
ANEXOS

ANEXO A. Casa de la calidad

<table>
<thead>
<tr>
<th>Flja.</th>
<th>Gráfica de peso</th>
<th>Peso relativo</th>
<th>Importancia</th>
<th>Relación máxima</th>
<th>Correlaciones</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16,7%</td>
<td>5</td>
<td>9</td>
<td>Alta precisión (máyor a 89%)</td>
<td>●</td>
</tr>
<tr>
<td>2</td>
<td>13,3%</td>
<td>4</td>
<td>9</td>
<td>Compabilidad con múltiples versiones de Android</td>
<td>●</td>
</tr>
<tr>
<td>3</td>
<td>13,3%</td>
<td>4</td>
<td>3</td>
<td>Facilidad de manejo de la aplicación móvil</td>
<td>○</td>
</tr>
<tr>
<td>4</td>
<td>10,0%</td>
<td>3</td>
<td>9</td>
<td>Seguridad de la información</td>
<td>○</td>
</tr>
<tr>
<td>5</td>
<td>13,3%</td>
<td>4</td>
<td>9</td>
<td>Alta velocidad de respuesta</td>
<td>○ ○ ○ ● ● ○ ●</td>
</tr>
<tr>
<td>6</td>
<td>10,0%</td>
<td>3</td>
<td>9</td>
<td>Bajo consumo de memoria y espacio de almacenamiento</td>
<td>○ ○ ○ ○ ○ ○ ○ ○</td>
</tr>
<tr>
<td>7</td>
<td>10%</td>
<td>3</td>
<td>9</td>
<td>Estética y orden en la interfaz gráfica</td>
<td>○ ○ ○ ● ●</td>
</tr>
<tr>
<td>8</td>
<td>13%</td>
<td>4</td>
<td>9</td>
<td>Orden en la presentación de la información generada</td>
<td>○ ○ ● ●</td>
</tr>
</tbody>
</table>

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.