

TRABAJO FIN DE GRADO

INGENIERÍA Y SISTEMAS DE DATOS

TÍTULO: Sistema basado en Edge-AI para detección de anomalías en cultivos

AUTOR: Alberto Sánchez Sánchez

TUTOR: Vicente Hernández Díaz

DEPARTAMENTO: Ingeniería Telemática y Electrónica

VºBº TUTOR

Miembros del Tribunal Calificador:

PRESIDENTE: Francisco Aznar Ballesta

TUTOR: Vicente Hernández Díaz

SECRETARIO: Néstor Lucas Martínez

Fecha de lectura:

Calificación:

El secretario

Agradecimientos

Este trabajo va dedicado a todas aquellas personas que me han apoyado durante estos cuatro años, en los que ha habido tanto momentos complicados como momentos felices.

Dedico este trabajo, en primer lugar, a mis padres y a mi hermana, por su apoyo incondicional, su compromiso con mi desarrollo educativo, su apoyo en los momentos complicados y la confianza que han depositado en mí durante todo este tiempo.

A quienes me conocen mejor que yo mismo y nunca han dejado de estar: mis amigos y Laura, quienes han estado conmigo mucho antes de empezar esta etapa y siempre han sabido sacar lo mejor de mí y hacer que no me rinda. Gracias por ser un apoyo incondicional constante y por seguir a mi lado.

A mis amigos de la universidad, por las largas horas juntos y el apoyo constante que me habéis mostrado, y por hacer los momentos complicados más amenos.

Gracias a ti, Vicente, por tu apoyo y directrices tanto durante la carrera como en la elaboración de este trabajo, por ayudarme a sacar la versión más crítica y reflexiva de mí mismo, y por tu paciencia y dedicación en todo momento.

Sé que sin todos vosotros, esto no hubiera sido posible.

Resumen

El TFG realizado, denominado "**Sistema basado en Edge-AI para detección de anomalías en cultivos**", tiene como objetivo el desarrollar un sistema de detección de maleza en cultivos de zanahorias, aplicando técnicas de visión artificial e inteligencia artificial en el contexto de la agricultura de precisión. La finalidad del estudio es optimizar el uso de herbicidas, reduciendo su uso innecesario y al mismo tiempo mejorar la calidad de cultivos y el impacto ambiental asociado.

Uno de los objetivos clave del presente trabajo es demostrar la viabilidad y eficiencia del uso de **Edge-AI** en dispositivos de bajo consumo energético, concretamente la **Nvidia Jetson Nano**, dentro del contexto de agricultura de precisión. Aunque se ha tomado como caso un estudio de detección de maleza en cultivos de zanahoria, el foco del estudio radica en validar una arquitectura basada en Edge Computing capaz de ejecutar algoritmos de inteligencia artificial en tiempo real, con baja latencia, mayor privacidad de datos e independiente a localizaciones con baja conectividad.

La metodología seguida consiste en utilizar un algoritmo de segmentación semántica, debido a la segmentación a nivel de píxeles de imágenes de cultivos de zanahoria. Se han comparado distintos modelos de redes convolucionales, tomando como solución el algoritmo de **U-Net** debido a su equilibrio en resultados tanto de entrenamiento como de validación, basándose en la métrica **mIoU**, y a su rendimiento en estos dispositivos de bajo rendimiento. El algoritmo final, además ha sido optimizado utilizando **TensorRT** garantizando el correcto funcionamiento en este dispositivo de borde.

También, se ha desarrollado un sistema de almacenamiento que permite monitorizar la evolución temporal de la presencia de maleza en cultivos de zanahoria, a través de la visualización del porcentaje de superficie afectada en cada momento. Esto proporciona al agricultor o agricultora una herramienta automática de seguimiento de detección de maleza y aplicación de herbicida, garantizando así un control y trazabilidad sobre el tratamiento aplicado.

En conclusión, el sistema desarrollado permite avanzar hacia una solución práctica y sostenible para el control de maleza, contribuyendo a la optimización del uso de herbicida, garantizando una reducción de la huella ecológica y una mayor productividad agrícola, suponiendo una mejor compensación económica también al agricultor o agricultora. Se plantea como una base fiable para ampliaciones o adaptaciones a otros tipos de cultivo con un caso de uso similar.

Abstract

The Final Degree Project titled '**Edge-AI-based system for the detection of crop anomalies**', aims to develop a weed detection system for carrot crops by applying computer vision and artificial intelligence techniques in the context of precision agriculture. The main goal of the study is to optimize herbicide use, minimize unnecessary applications while improving crop quality and reducing associated environmental impact.

One of the key objectives of this work is to demonstrate the feasibility and efficiency of using **Edge-AI** on low-power devices, specifically the **Nvidia Jetson Nano**, within the context of precision agriculture. Although a weed detection system in carrot crops has been used as a case study, the main focus of the project is to validate an Edge Computing architecture capable of running artificial intelligence algorithms in real time, with low latency, enhanced data privacy, and independence from network connectivity.

The proposed methodology is based on the use of a semantic segmentation algorithm, due to the need for pixel-level classification of images from carrot crops. Various convolutional neural network models were compared, and U-Net was selected as the final solution, offering balanced performance in both training and validation based on the **mIoU** metric, as well as efficient execution on low-resource hardware. Furthermore, the algorithm was optimized using **TensorRT** to ensure proper performance on the edge device.

A storage system was also developed to monitor the temporal evolution of the presence of weeds in the carrot fields by visualizing the percentage of affected surfaces over time. This provides farmers with an effective monitoring tool for the automated weed detection and herbicide application system, ensuring better control and traceability of the treatment process.

In conclusion, the developed system offers a practical and sustainable solution for weed control, contributing to the optimization of herbicide use, reducing the ecological footprint, and improving agricultural productivity. It also represents a solid foundation for future extensions or adaptations to other types of crops with similar use cases.

Índice de contenidos

| | |
|---|-----------|
| Agradecimientos | 1 |
| Resumen | 2 |
| Abstract | 4 |
| Índice de Figuras | 11 |
| Índice de Tablas | 13 |
| Acrónimos | 15 |
| 1. Introducción | 17 |
| 2. Resumen Ejecutivo | 19 |
| 3. Estado del arte | 21 |
| 3.1. Edge AI y sus dispositivos | 21 |
| 3.1.1. Edge Computing | 21 |
| 3.1.2. Edge AI | 22 |
| 3.1.3. Dispositivos | 23 |
| 3.1.3.1. Raspberry Pi | 23 |
| 3.1.3.2. Jetson Nano | 24 |
| 3.2. Redes Convolucionales | 26 |
| 3.3. Clasificación de imágenes | 27 |
| 3.3.1. FlexiGrobots | 27 |
| 3.3.2. CropDeep | 29 |
| 3.3.3. Enfermedad trigo | 31 |
| 3.4. Segmentación Semántica de imágenes | 31 |
| 3.4.1. Aplicación Control de Maleza en el Borde | 31 |
| 3.5. Optimización IA | 33 |

| | |
|---|-----------|
| 3.6. Resumen Estado del Arte | 33 |
| 4. Especificaciones técnicas y restricciones de diseño | 35 |
| 4.1. Especificaciones del diseño | 35 |
| 4.2. Restricciones de diseño | 36 |
| 5. Descripción de la propuesta | 37 |
| 5.1. Descripción del caso de uso | 37 |
| 5.2. Modelo caso de uso | 37 |
| 5.2.1. Detectar Maleza - Caso de uso | 40 |
| 5.2.2. Generar estadísticas - Caso de uso | 41 |
| 5.3. Diseño de la arquitectura | 42 |
| 5.4. Diagrama de despliegue | 44 |
| 5.5. Implementación | 46 |
| 5.5.1. Estrategia de Despliegue del Modelo de IA | 46 |
| 5.5.2. Detección de Maleza | 47 |
| 5.5.3. Generar Estadísticas y Almacenamiento | 49 |
| 5.5.4. Monitorización del cultivo | 51 |
| 6. Pruebas y Resultados | 53 |
| 6.1. Pruebas algoritmos IA | 53 |
| 6.2. Validación Resultados | 54 |
| 6.3. Validación Optimización IA | 56 |
| 7. Presupuesto | 57 |
| 7.1. Costes de Equipamiento y Software | 57 |
| 7.2. Coste Desarrollo y Mano de Obra | 57 |
| 7.3. Costes Totales | 58 |
| 8. Análisis de impacto | 59 |
| 8.1. Impacto tecnológico | 59 |
| 8.2. Impacto económico | 59 |
| 8.3. Impacto social | 60 |
| 8.4. Impacto ambiental | 60 |
| 8.5. Impacto en Objetivos de Desarrollo Sostenible | 61 |

| | |
|---|-----------|
| 9. Conclusiones y trabajos futuros | 63 |
| 9.1. Conclusiones | 63 |
| 9.2. Trabajos futuros | 64 |
| 10. Referencias | 65 |
| A. Anexo CWFID Dataset | 67 |
| B. Monitorización Cultivo | 69 |
| C. Anexo Scripts y Ejecución | 71 |

Índice de Figuras

| | | |
|-----|------------------------------------|----|
| 1. | Arquitectura Edge AI [2] | 22 |
| 2. | Raspberry Pi 3 B+ [5] | 24 |
| 3. | Jetson Nano Developer Kit E [7] | 25 |
| 4. | Arquitectura CNN (VGG16) [8] | 27 |
| 5. | Detección ramos uva FlexiGrobot[9] | 28 |
| 6. | Caso de Uso Modelo | 37 |
| 7. | Diagrama de secuencia general | 39 |
| 8. | Detectar anomalía Caso de uso | 40 |
| 9. | Componentes Arquitectura | 42 |
| 10. | Diagrama de Despliegue | 44 |
| 11. | Cultivo capturado | 47 |
| 12. | Arquitectura Unet [16] | 48 |
| 13. | Imagen predicha | 48 |
| 14. | Base de Datos I | 49 |
| 15. | Base de Datos II | 49 |
| 16. | Gráfica Monitoreo | 51 |
| 17. | Pérdidas y Métricas U-Net | 55 |
| 18. | Imágenes Monitorización | 69 |

Índice de Tablas

| | | |
|-----|--|----|
| 1. | Comparación Raspberry Pi vs Jetson Nano | 24 |
| 2. | Comparativa entre MLP y CNN para procesamiento de imágenes | 26 |
| 3. | Modelos y y Métricas FlexiGrobots [9] | 28 |
| 4. | Resultados CropDeep [10] | 30 |
| 5. | Resultados Transfer Learning Control Maleza | 32 |
| 6. | Requisitos funcionales | 35 |
| 7. | Requisitos funcionales | 35 |
| 8. | Restricciones de diseño | 36 |
| 9. | Detectar Maleza - Caso de Uso | 40 |
| 10. | Generar estadísticas - Caso de uso | 41 |
| 11. | Resultados Entrenamiento Modelo IA | 54 |
| 12. | Resultados Validación Modelo IA | 55 |
| 13. | Resultados Validación Modelo IA | 56 |
| 14. | Costes de Equipamiento y Software | 57 |
| 15. | Costes Laborales | 57 |
| 16. | PM | 58 |
| 17. | Coste Total | 58 |

Acrónimos

| | |
|----------------|--|
| ATVs | Vehículos Todo Terrenos. |
| CNN | Red Neuronal Convolucional. |
| Edge-AI | Edge Computing para IA. |
| GPU | General Process Unit. |
| HDF5 | Formato de Datos Jerárquico versión 5. |
| HTTP | Protocolo de Transferencia de Hipertexto. |
| HTTPS | Protocolo Seguro de Transferencia de Hipertexto. |
| IA | Inteligencia Artificial. |
| IoU | Intersección sobre Unión. |
| mAP | Precision Promedio Media. |
| mIoU | Intersección Media sobre Unión. |
| ML | Machine Learning. |
| MLP | Perceptrón Multi Capa. |
| ODS | Objetivos de Desarrollo Sostenible. |
| OS | Paso de Salida. |
| RGPD | Reglamento General de Protección de Datos. |
| UART | Transmisor-Receptor Asíncrono Universal. |
| UAV | Vehículos Aéreos No Tripulados. |
| UGV | Vehículos Terrestres No Tripulados. |
| UML | Lenguaje Uificado de Modelado. |

1. Introducción

En la actualidad, la sociedad y la tecnología evolucionan de forma conjunta, avanzando hacia sistemas cada vez más autónomos y capaces de tomar decisiones inteligentes sin intervención humana directa. Esta toma de decisiones se vuelve especialmente crítica en contextos donde la rapidez de respuesta, la eficiencia energética y la optimización de recursos son factores determinantes para garantizar un funcionamiento adecuado y sostenible del sistema [1].

El desarrollo de tecnologías para el procesamiento de datos mediante algoritmos de Inteligencia Artificial (IA) ha dado lugar a múltiples arquitecturas, desde soluciones centralizadas como Cloud Computing, hasta modelos más distribuidos como el Mobile Cloud Computing (MCC), Fog Computing, Cloudlet, Micro Data Centers o redes MEC Edge. Estos últimos permiten acercar el procesamiento al punto de origen de los datos, reduciendo la latencia y mejorando la eficiencia del sistema.

Entre todas estas alternativas, destaca el paradigma del **Edge AI**, que consiste en integrar algoritmos de IA directamente en los dispositivos donde se capturan los datos [2]. Esta arquitectura no solo permite un procesamiento local más rápido, sino que también incrementa la privacidad, reduce la carga en las redes de comunicación y disminuye el impacto energético y económico del sistema, haciéndolo especialmente adecuado para entornos donde el tiempo de respuesta y la autonomía son clave.

El presente proyecto propone aplicar este enfoque de Edge AI al ámbito de la agricultura de precisión, desarrollando un sistema inteligente que, mediante el tratamiento de imágenes capturadas en campo, sea capaz de detectar de forma automática la presencia de maleza en cultivos de zanahoria y activar mecanismos de aplicación localizada de herbicidas. La agricultura de precisión representa una evolución tecnológica del sector agrícola, orientada a la optimización de los insumos, la eficiencia productiva y la sostenibilidad ambiental[3]. Según estudios realizados en 2016, su implementación puede reducir hasta un 30 % los costes de producción, mientras que, de acuerdo con Carlos Palomar [4], director de la Asociación Empresarial para la Protección de las Plantas (Aepla), también permite ahorrar un 10 % en fertilizantes, aumentar en un 10 % la productividad y alcanzar hasta un 20 % de incremento en la rentabilidad.

2. Resumen Ejecutivo

Este documento ofrece una solución a una problemática dentro del sector de agricultura de precisión, como es la detección de maleza en cultivos, más concretamente en cultivos de zanahorias. La solución planteada consiste en un algoritmo de IA de segmentación semántica para la detección de maleza junto con el monitoreo y la automatización del uso de herbicida. Esta inferencia está basada en el procesamiento en el borde o Edge Computing aplicando así un enfoque Edge AI. La finalidad es lograr una producción sostenible, optimizando el uso de herbicida, reduciendo el impacto ambiental y aumentando la productividad agrícola.

El documento se organiza en varias secciones, donde se expone desde el análisis a la resolución para el éxito del estudio:

- **Estado del arte:** en primer lugar, se presenta el concepto de procesamiento en el borde (Edge Computing) junto con la descripción del dispositivo final utilizado para inferencias locales. A continuación, se analizan distintas tecnologías y estudios relacionados con el procesamiento de imágenes de cultivos aplicando algoritmos de IA.
- **Especificaciones técnicas y restricciones de diseño:** se explican detalladamente las especificaciones de diseño como requisitos funcionales y no funcionales, hasta las restricciones de diseño.
- **Descripción de la propuesta:** en esta sección se explica a fondo el caso de uso utilizando diagramas de secuencia. También se explica los diferentes componentes que forman parte de la arquitectura de la solución planteada y un diagrama de despliegue.
- **Implementación:** se exponen los pasos de la solución planteada con detalle, desde la extracción de la fuente de datos, el algoritmo de IA usado, la generación de estadísticas, almacenamiento de información, y la monitorización del cultivo.
- **Pruebas y Resultados:** explicación de los diferentes algoritmos de IA probados y la explicación de la elección del algoritmo resultante, mediante la comparación de distintas métricas referidas al modelo y tiempos de inferencia.
- **Presupuesto:** explicación detalla del precio de mano de obra, material utilizado. Al final de este capítulo se encuentra el coste total del proyecto.
- **Análisis del impacto:** se analiza el impacto potencial del sistema desarrollado, tanto en términos tecnológicos, económicos, sociales, ambientales como los ODS que aborda.
- **Conclusiones del trabajo:** esta sección se sintetiza el algoritmo de IA usado con las capacidades que presenta para su despliegue en un sistema real, confirmando la fiabilidad y precisión de la solución final para abordar el problema planteado. También se expresan trabajos futuros y futuras funcionalidades a adoptar basándose en el estudio.

3. Estado del arte

A día de hoy, existen varias tecnologías capaces de identificar diferentes anomalías en el campo de la agricultura de precisión. Estas tecnologías son capaces de abordar la problemática mediante el concepto de visión artificial, el cual, contiene características como reconocimiento de imágenes, segmentación de clases, clasificación de imágenes, detección de objetos o reconocimiento de patrones.

Haciendo uso de estas tecnologías, es posible tratar el problema de detección de anomalías en cultivos mediante su integración con funcionalidades adicionales de automatización y monitoreo para garantizar un sistema fiable, efectivo y ventajoso, teniendo en cuenta las capacidades del dispositivo final. Estos desarrollan el concepto de **Edge AI** (*Edge Computing para IA*), siendo el dispositivo donde se realiza tanto la captura de datos como el procesamiento.

3.1. Edge AI y sus dispositivos

3.1.1. Edge Computing

Edge Computing o computación en el borde [2], es una técnica de computación que consiste en procesar los datos lo más cercano posible al lugar donde se han generado/capturado, es decir, en los dispositivos periféricos o nodos de borde, en lugar de enviarlos a servidores externos o la nube, siguiendo una estrategia Cloud Computing. Estos nodos, pueden ser desde cámaras, drones hasta dispositivos IoT.

Entre a las ventajas del Edge Computing se encuentran:

- **Reducción de latencia:** la latencia es el tiempo, en procesos de comunicación, que tarda en ir un dato capturado en origen a destino. Al realizarse tanto la captura como el procesamiento en el mismo dispositivo (o dispositivos muy cercanos en el espacio) la latencia es prácticamente nula o muy baja, haciendo posible una inferencia en tiempo efectivo.
- **Reducción ancho de banda:** alineado con la latencia, al no haber que transmitir datos grandes como pueden ser imágenes o vídeos, se transmiten únicamente posibles etiquetas tras aplicar algún tipo de algoritmo.
- **Costes operativos bajos:** enviando menos datos se disminuye el consumo de datos satelitales o móviles (cuando se usan en zonas rurales) y menores costes de almacenamiento y transferencia. Se presenta como una alternativa energéticamente eficiente y respetuosa con el medioambiente.
- **Escalabilidad:** no depende de un sistema centralizado, por lo que la escalabilidad tanto horizontal como vertical permite una adaptación sencilla sin comprometer la dificultad

de integración, garantizando así el correcto funcionamiento del sistema.

- **Privacidad:** los datos sensibles como pueden ser imágenes de cultivos o información geolocalizada, no viajan, se procesan localmente, garantizando el cumplimiento de la RGPD (*Reglamento General de Protección de Datos*), al reducir el riesgo de exposición o un mayor control por parte del usuario.

De esta manera, en cuanto a la **agricultura de precisión**, donde los cultivos se encuentran en zonas de comunicación limitada o intermitente y se mejora la rapidez de detección de anomalías, garantizando así una reducción en impacto económico y ambiental, es conveniente adoptar el enfoque de Edge Computing

3.1.2. Edge AI

Edge AI, consiste en aplicar algoritmos de IA en la arquitectura de Edge Computing, es decir, en los dispositivos finales. Algunos ejemplos de algoritmos de IA más conocidos son desde ML (*Machine Learning*) o Aprendizaje Máquina, algoritmos de DeepLearning o Aprendizaje Profundo, redes bayesianas o algoritmos genéticos. El algoritmo en cuestión, debe ser capaz de ejecutarse en el dispositivo final elegido, por lo que hay que tener en cuenta las capacidades de procesamiento.

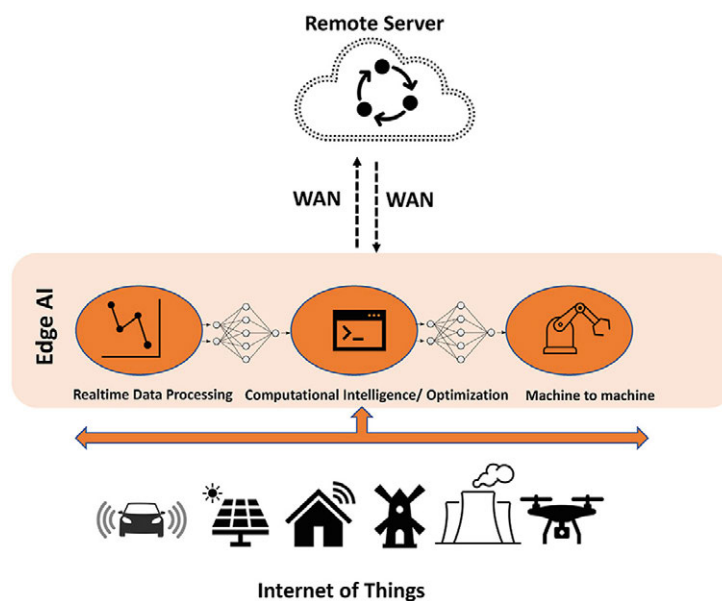


Figura 1. Arquitectura Edge AI [2]

La Figura 1 representa una aproximación de la arquitectura de Edge AI, donde se tienen los dispositivos finales, el procesamiento de los datos utilizando algoritmos de IA en la parte de Edge AI y la última capa, donde se puede generar una conexión con un servidor remoto en caso de que se desee transmitir muy poca cantidad de información, como alguna etiqueta (en problemas de clasificación) o una alerta, siendo este un dato pequeño.

Los algoritmos de IA deben ser optimizados, debido a las propiedades de procesamiento de los dispositivos finales. Estos dispositivos finales normalmente cuentan con menos capacidad de

cómputo: RAM, CPU o un consumo energético limitado. Estas optimizaciones tienen que velar por el equilibrio con respecto a la precisión del modelo aplicado, para no perder la efectividad en la tarea correspondiente.

3.1.3. Dispositivos

Los dispositivos finales son el elemento núcleo de una arquitectura basada en Edge Computing. También conocidos como *end devices*, deben ser capaces de ejecutar los algoritmos de IA correspondientes a la tarea designada. Se tratan de dispositivos ubicados en los extremos de la red, encargados de la captura de los datos y de su procesamiento local o en dispositivos auxiliares cercanos en el espacio.

Estos dispositivos incluyen sensores, cámaras, actuadores y microcontroladores que recolectan información relevante. Son para la monitorización en tiempo real y la automatización, ya que constituyen la interfaz directa con el entorno físico, permitiendo que sistemas inteligentes tomen decisiones informadas basadas en datos actualizados.

Centrándose en los microcontroladores y microordenadores que se encargan de la parte de procesamiento en la actualidad hay dos que destacan en el mercado:

3.1.3.1. Raspberry Pi

La Raspberry Pi **Raspberry** es un microordenador de placa única de bajo coste y alto rendimiento, capaz de ejecutar modelos de inferencia en el borde gracias a su procesador de cuatro núcleos, soporte de bibliotecas de IA como Tensorflow Lite y conectividad para integrar sensores ambientales o cámaras. Se utiliza en campos como IoT, Edge Computing, domótica, robótica y agricultura de precisión. Hay varios modelos, cada uno con mayor capacidad de procesamiento y menor coste. Algunas de las características que contiene son:

- **Procesador y GPU:** el procesador se encarga de ejecutar las tareas principales del sistema, desde la gestión de sensores hasta el análisis de los datos. Permite ejecutar modelos de IA ligeros directamente en el dispositivo, mientras que la GPU puede acelerar el procesamiento de las imágenes.
- **Memoria y almacenamiento:** permite ejecutar aplicaciones y modelos en paralelo de forma eficiente. Se encarga de guardar el sistema operativo, los datos capturados y los resultados procesados. Es esencial para mantener la autonomía del sistema.
- **Conectividad:** permite la comunicación con otros dispositivos o servidores a través de redes cableadas o inalámbricas. Útil para la sincronización de los datos, recibir instrucciones remotas o enviar alertas.
- **Interfaces para expansión:** facilitan la conexión con cámaras, actuadores o pantallas de dispositivos, haciendo que sea altamente configurable y escalable.

La Figura 2 expone una imagen de la placa Raspberry Pi 3 B+, uno de los modelos más conocidos.

La Raspberry Pi es ideal para proyectos sencillos, económicos y de bajo consumo que no requieran de mucho procesamiento de IA, como control de sensores o tareas básicas



Figura 2. Raspberry Pi 3 B+ [5]

de inferencia como pueden ser algoritmos menos pesados como los basadas en entradas numéricas (ML). Los entornos donde mejor responde es en aquellos energéticamente limitados.

La Raspberry Pi tiene pocos recursos para ejecutar modelos de IA, pero existe la posibilidad de conectarle/añadirle componentes como la Hailo8 que incorpora un hardware especializado y optimizado para ejecutar modelos de IA, aunque aún está en fase de estudio.

3.1.3.2. Jetson Nano

La **Nvidia Jetson Nano** [6] es una computadora de desarrollo de bajo costo y alto rendimiento diseñada para aplicar IA y aprendizaje profundo a dispositivos de computación de borde. Su diseño es único y está dirigido a estudiantes y desarrolladores que desean implementar soluciones mediante visión artificial, redes neuronales y robótica sin necesidad de hardware costoso.

Sus principales aplicaciones son proyectos relacionados con procesamiento local inteligente como agricultura de precisión, robots autónomos, cámaras inteligentes y sistemas de seguridad. Al igual que las Raspberry Pi tiene las características mencionadas anteriormente aunque se destaca por ligeras diferencias.

Tabla 1. Comparación Raspberry Pi vs Jetson Nano

| Funcionalidades | Raspberry Pi | Jetson Nano |
|--------------------------------|---|---|
| Procesamiento general | Tareas ligeras (control sensores y lógica básica) | Tareas potentes (complejas y procesamiento en paralelo) |
| Procesamiento IA GPU | Modelos ligeros No orientada a IA | Modelos más complejos Orientada a acelerar con CUDA |
| Conectividad | Si | Si |
| Compatibilidad sensores | GPIO y buses I2C, SPI | GPIO y soporte cámaras CSI |

Las funcionalidades son parecidas, aunque la Jetson Nano es mejor opción para optar por modelos de visión artificial que usan imágenes, al tener una mayor capacidad de soporte de modelos complejos y poder realizar una optimización del modelo, al estar orientada específicamente a utilizar IA. Con esta optimización, se obtienen resultados mejores en cuanto a tiempo de inferencia.

Debido a las altas prestaciones de la **Jetson Nano** se ha optado por esta solución al soportar modelos más complejos, ya que se está trabajando con imágenes y se necesita mayor capacidad de cómputo, además de posibles adaptaciones a nuevas soluciones, garantizando flexibilidad y adaptabilidad.

En concreto se ha utilizado para el desarrollo la **Waveshare Jetson Nano**, más concretamente el **Development Kit E**.



Figura 3. Jetson Nano Developer Kit E [7]

Este tipo de microordenador es capaz de ejecutar modelos de IA relacionados con clasificación de imágenes, detección de objetos, segmentación semántica, y procesamiento de voz. Las características hardware son las siguientes:

- **Fabricante:** Waveshare
- **GPU:** 128 núcleos CUDA Maxwell
- **CPU:** Cuatro núcleos ARM Cortex-A57
- **RAM:** 4 GB LPDDR4
- **Puertos:** HDMI, USB 3.0, USB 2.0, GPIOs
- **Almacenamiento:** microSD
- **Consumo energético:** 5 W – 10 W
- **Número de procesadores:** 4

- **Dimensiones:** 6,9 × 4,5 × 4,5 cm
- **Peso:** 1,19 kg
- **Conectividad:** Bluetooth, Wi-Fi y Ethernet
- **Refrigeración:** Ventilador incluido
- **Dispositivos compatibles:** Cámara estéreo (visión 3D o detección de profundidad), sensores, ordenador, smartphone, teclado, ratón, almacenamiento externo, pantallas.

3.2. Redes Convolucionales

Para el procesamiento de imágenes en modelos de IA se utilizan redes convolucionales o también conocidas por sus siglas como **CNN**. En comparación con otros modelos, como Redes Neuronales donde la entrada al modelo son un conjunto de datos numéricos, la entrada son un conjunto de imágenes. Ese tipo de red aprende a detectar patrones como bordes, formas, colores etc. Se compone de **capas convolucionales**, encargadas de aplicar filtros pequeños para extraer las características (a su salida genera una nueva imagen llamado mapa de características) y **capas de reducción**, conocidas como pooling para eliminar información poco importante y así disminuir la dimensionalidad para que el modelo sea más rápido y evite errores por ruido. Cuanto mayor es la profundidad está es capaz de aprender patrones más complejos, pero también más tarda en inferir.

Tabla 2. Comparativa entre MLP y CNN para procesamiento de imágenes

| Característica | MLP (Perceptrón Multicapa) | CNN |
|------------------------------------|--|---|
| Arquitectura | Totalmente conectada (cada neurona con todas las entradas) | Conectividad local (usa filtros sobre regiones pequeñas) |
| Entrada | Vector plano (requiere aplanar la imagen) | Imagen 2D o 3D (mantiene estructura espacial) |
| Parámetros | Muchísimos (rápido crecimiento con el tamaño de la imagen) | Menos (usa los mismos filtros en toda la imagen) |
| Coste computacional | Alto para imágenes grandes | Más eficiente gracias al uso de filtros compartidos |
| Reconocimiento de patrones locales | No es eficiente para detectar estructuras espaciales | Muy bueno para detectar bordes, texturas, formas |
| Generalización | Más propenso a sobreajuste con imágenes grandes | Mejor capacidad de generalizar en visión por computador |
| Aplicaciones típicas | Datos tabulares o secuenciales | Clasificación de imágenes, segmentación, detección de objetos |

Es decir, la CNN es una mejor opción para el procesamiento de imágenes ya que tiene en cuenta las relaciones espaciales entre los píxeles y no solo sus valores numéricos. La Figura 4 representa la arquitectura de las etapas de convolución y reducción es la siguiente:

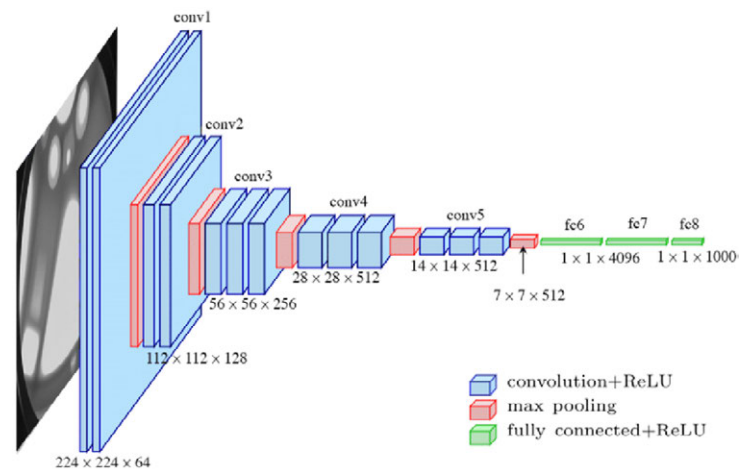


Figura 4. Arquitectura CNN (VGG16) [8]

3.3. Clasificación de imágenes

La clasificación de imágenes constituye una tarea fundamental en el campo de la visión artificial. Este proceso consiste en utilizar un sistema de IA (Inteligencia Artificial) para analizar una imagen y determinar a qué clase o categoría pertenece. La clasificación se realiza en consecuencia a una imagen, es decir, a la imagen resultante se le asigna una única clase.

A continuación se exponen una serie de proyectos relacionados con la agricultura de precisión que han utilizado clasificación de imágenes con diferentes propósitos.

3.3.1. FlexiGrobots

FlexiGrobot [9] es un proyecto de agricultura de precisión de la Unión Europea financiado por la Comisión Europea en el marco del programa Horizonte 2020. Es el resultado de un amplio estudio en el que se utilizó un algoritmo de IA para detectar ramas de uva no saludables, por culpa del hongo *Botrytis*, en diferentes viñedos. Se utilizaron vehículos aéreos no tripulados (UAV) y terrestres no tripulados (UGV). El proceso consta de cuatro pasos:

1. **Exploración aérea:** Los drones realizan inspecciones aéreas y predicen donde hay una alta probabilidad de *Botrytis* en los viñedos. Esta predicción se realizó mediante un algoritmo *Random Forest*, algoritmo de IA supervisado, cuyos datos de entrada fueron la altura de la copa del árbol (CHM), el índice de diferencia normalizada de vegetación (NDVI) y el índice de área foliar.
2. **Exploración terrestre:** Los vehículos terrestres no tripulados (UGVs) recorren las áreas predichas por los UAVs y toman fotografías del viñedo correspondiente. Se realiza la predicción con el algoritmo de IA.
3. **Tratamiento:** Los robots recorren las zonas afectadas por esta enfermedad y aplican productos fitosanitarios según el mapa o en tiempo real durante la exploración.
4. **Asistencia:** Un conjunto de UGV también ayuda a los operadores en la cosecha manual del cultivo, creando un entorno de trabajo cooperativo para transportar las cestas.

En resumen, el modelo de IA utilizado para detectar el hongo es el usado por los UGVs. Tras realizar la foto, se realiza una identificación de los ramos de uva que hay en la foto, obteniendo una imagen con un único ramo de uva (conjunto de uvas). La detección de estos racimos de uvas fue mediante **bounding boxes** o **cajas delimitadoras**, las cuales consisten en rectángulos que rodean un objeto de interés dentro de una imagen.

El modelo de IA consiste en una **CNN** (*Red Neuronal Convolutacional*), que extrae las características y dibuja los cuadros delimitadores. La capa final, el clasificador, asigna los racimos de 5 uvas como sanos o no sanos. La Jetson Nano fue utilizada y equipada en los UGVs y la cámara empleada, una Canon EOS 7D, compatible con el módulo del Kit de Desarrollo E empleado en este estudio. La clasificación de las uvas debería verse así, como muestra la **Figura 5**, en las que se muestran las cajas delimitadoras.

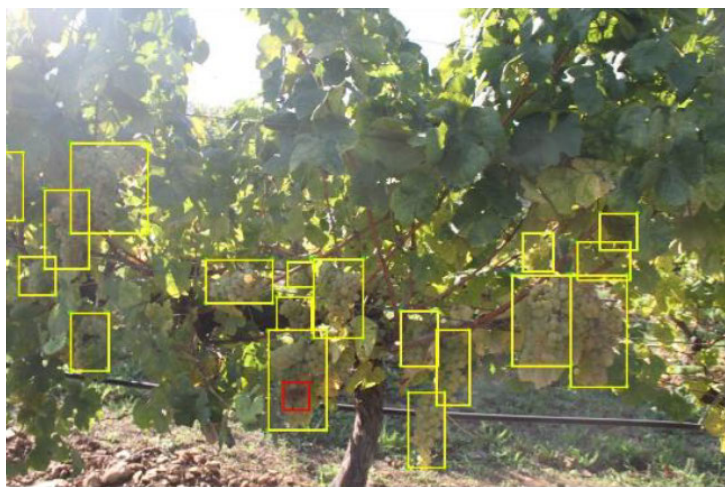


Figura 5. Detección ramos uva FlexiGrobot[9]

Finalmente, la capa de clasificación **MLP** (*Perceptrón Multi Capa*), asignaba la etiqueta RD (clase 1) al racimo con el hongo y RS (clase 0) al racimo sano. Los diferentes algoritmo y resultados con respecto a las métricas utilizadas fueron las siguientes:

Tabla 3. Modelos y y Métricas FlexiGrobots [9]

| Arquitectura | Red base | Tamaño Entrada | Pasos | mAP @ 0.5IoU |
|--------------|---------------|----------------|-------|--------------|
| FRCNN | ResNet50 | 640x640 | 43k | 0.86 |
| | ResNet101 | 640x640 | 35k | 0.73 |
| | ResNet152 | 640x640 | 20k | 0.75 |
| EfficientNet | EfficientNet | 512x512 | 54k | 0.88 |
| | EfficientNet | 640x640 | 27k | 0.81 |
| | EfficientNet | 768x768 | 38k | 0.83 |
| | EfficientNet | 768x768 | 25k | 0.80 |
| CenterNet | ResNet50 | 512x512 | 5k | 0.75 |
| | Hourglass 104 | 512x512 | 8k | 0.78 |
| SSD | MobileNet V1 | 640x640 | 7k | 0.71 |

Donde la **red base** representa la red principal encargada de extraer características básicas de

la imagen (detector de patrones básicos) como pueden ser bordes, esquinas, texturas, colores planos o cortonos simples. La **arquitectura** o estructura interna de capas para detección de objetos mediante cajas delimitadoras, **tamaño de entrada** o dimensiones de la imagen de entrada, los **pasos** o número de iteraciones del modelo en el entrenamiento y **mAP @ 0.5IoU**, siendo la métrica que determina que una caja ha sido correctamente predicha cuando su IoU (*Intersección sobre Unión*) con respecto a la caja verdadera es superior a 0.5.

Para poder entrenar correctamente un modelo de detección de objetos, es imprescindible que los datos de entrenamiento están dentro de cajas delimitadoras (bounding boxes).

3.3.2. CropDeep

Parte del estudio [10] se basa en la clasificación de cultivos en ciertas etapas de maduración y distinción de especies, utilizando algoritmos de IA. Presenta una revisión exhaustiva sobre la integración de tecnologías avanzadas de IA en ATVs (*Vehículos Todo Terrenos*) para la revolución de la agricultura de precisión. El conjunto de datos está formado por más de 31 clases, las cuales contienen alrededor de 31.000 imágenes. Utilizaron diferentes modelos y se basaron en la métrica **mAP** (*Precisión Promedio Media*) para realizar la clasificación de imágenes. Este estudio no aborda la detección de anomalías, sin embargo con imágenes etiquetadas de frutos sanos y frutos anómalos se podrían emplear estas arquitecturas para realizar la clasificación en cuestión. Los estudios fueron los siguientes:

Tabla 4. Resultados CropDeep [10]

| Arquitecturas | Faster R-CNN | SSD | RFB | YOLOv2 | YOLOv3 | RetNet |
|-------------------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Extractor | VGG-16 | VGG-16 | VGG-19 | Darknet-19 | Darknet-53 | ResNet50 |
| Características | | | | | | |
| Tomato | 90.82 | 90.91 | 90.45 | 97.28 | 97.51 | 98.82 |
| Unripe tomato | 88.76 | 89.18 | 90.18 | 93.3 | 92.34 | 97.62 |
| Tomato early-blossom | 84.16 | 79.82 | 89.26 | 88.42 | 88.42 | 91.09 |
| Tomato full-blossom | 90.13 | 87.11 | 89.79 | 91.28 | 91.9 | 85.1 |
| Cucumber | 79.87 | 80.34 | 80.37 | 88.41 | 88.53 | 83.21 |
| Cucumber blossom | 76.12 | 75.74 | 84.76 | 86.4 | 88.05 | 94.95 |
| Unripe cucumber | 75.47 | 82.05 | 86.17 | 91.88 | 90.84 | 98.5 |
| Winter squash | 91.48 | 90.91 | 90.8 | 96.06 | 96.06 | 92.73 |
| Fingered citron | 94.6 | 99.18 | 95.19 | 99.38 | 98.29 | 100 |
| Pawpaw | 67.17 | 78.89 | 81.38 | 97.52 | 97.36 | 98.61 |
| Head lettuce | 95.76 | 97.02 | 95.42 | 96.82 | 96.21 | 98.38 |
| Endive | 89.02 | 87.92 | 91.82 | 91.81 | 88.78 | 95.4 |
| Butter lettuce | 88.43 | 90.46 | 90.57 | 96.03 | 96.09 | 99.15 |
| Rutabaga | 87.71 | 88.08 | 95.43 | 96.32 | 98.56 | 99.15 |
| Purple cabbage | 88.79 | 89.15 | 87.99 | 95.3 | 94.47 | 97.61 |
| Luosheng lettuce | 95.29 | 100 | 100 | 97.44 | 97.71 | 97.33 |
| Celery | 75.74 | 88.24 | 81.15 | 76.62 | 78.99 | 87.61 |
| Wolfberry | 90.33 | 98.42 | 99.86 | 95.84 | 99.35 | 100 |
| Lemon | 55.44 | 66.52 | 52.94 | 64.06 | 58.39 | 74.21 |
| Persimmon | 90.46 | 80.22 | 89.02 | 82.54 | 85.65 | 89.63 |
| Iceberg lettuce | 95.65 | 90.91 | 90.91 | 98.61 | 99.14 | 100 |
| Chinese cabbage | 100 | 100 | 100 | 100 | 100 | 100 |
| Turnip | 43.53 | 80.37 | 21.76 | 100 | 100 | 100 |
| Green turnip | 91.31 | 96.97 | 90.7 | 99.97 | 99.97 | 100 |
| Spinach | 63.64 | 63.64 | 65.43 | 62.12 | 63.6 | 66.25 |
| Scallion | 73.02 | 72.73 | 81.06 | 76.42 | 82.54 | 85.62 |
| Watermelon | 80.01 | 79.81 | 84.5 | 90.44 | 89.13 | 92.13 |
| Muskmelon | 90.17 | 86.58 | 94.98 | 94.02 | 92.3 | 87.36 |
| Chili pepper | 88.73 | 87.62 | 88.41 | 94.92 | 96.47 | 99.1 |
| Pumpkin | 83.26 | 88.27 | 92.38 | 96.15 | 97.54 | 97.54 |
| Person | 84.71 | 90.91 | 92.62 | 97.63 | 97.88 | 99.3 |
| Average | 83.53 | 86.19 | 85.23 | 90.78 | 91.44 | 92.79 |

3.3.3. Enfermedad trigo

Un artículo publicado en el MDPI [11], aborda la investigación del impacto adverso de las enfermedades de producción y rendimiento de cultivos. La enfermedad del trigo a menudo se detecta en etapas tardías y consume mucho tiempo, por lo que su detección precoz genera múltiples ventajas.

En el artículo se indica que el procesamiento se realiza usando la Jetson Nano, por lo que la ejecución se adapta al procesamiento local. Los modelos que se utilizaron fueron Inceptionv3, Resnet50, VGG16, VGG19, destacando un **99,38 %** de precisión por parte de **VGG19** siendo la más alta y un 81,57 % de precisión por la más baja en ResNet50.

También se muestra la relación entre la diferencia de tiempos de inferencia al utilizar procesamiento con CPU (Core i7) frente a GPU (Nvidia Tesla T4 y Nvidia K80).

3.4. Segmentación Semántica de imágenes

La **segmentación semántica** es una tarea dentro del campo de la visión artificial que consiste en clasificar cada píxel de una imagen a una categoría o clase, de esta manera, se genera un mapa de clases píxel a píxel. En cuanto a la agricultura de precisión, cada píxel se puede etiquetar por ejemplo perteneciente al suelo, hoja, fruto, etc.

Algunas de las funcionalidades de esta solución puede ser distinguir entre hojas sanas y hojas enfermas, separar el fondo del cultivo, o identificar frutos en distintas etapas de maduración.

3.4.1. Aplicación Control de Maleza en el Borde

Un estudio llamado "**Real-Time Weed Control Application Using a Jetson Nano Edge Device and a Spray Mechanism**" [12] utilizó la Jetson Nano con las mismas capacidades para realizar un sistema capaz de detectar maleza en cultivos y su posterior aplicación localizada de herbicida. La maleza, o también conocida como hierba mala, trata de un conjunto de plantas no deseadas que crecen junto a los cultivos agrícolas. Estas, compiten con los cultivos por recursos como el agua, la luz solar, los nutrientes o incluso el espacio, lo que afecta negativamente al cultivo. Alguna de las características de la maleza son:

- Crecen de forma rápida y resistente.
- Interfieren en la cosecha o dificultan labores agrícolas.
- Son invasivas o incluso tóxicas para animales y humanos.

Por lo que aplicar herbicida únicamente sobre la maleza y no sobre el cultivo reduce costes económicos, daños ambientales y evita resistencia en plantas.

La metodología y recursos del modelo asociado es un modelo de IA, donde se aplicó el modelo **DeepLabV3**, que es un modelo CNN, con una red base de **MobileNetV2**. Las ventajas de usar MobileNet en dispositivos en computación en el borde son:

- Diseñado para dispositivos de bajo poder computacional.

- Usa operaciones convolucionales que reducen drásticamente el número de parámetros y operaciones necesarias, por lo que el coste computacional es menor.
- Menor consumo de memoria y energía.
- Buen rendimiento con algoritmos de transfer learning.
- Compatible con herramientas Nvidia de optimización como TensorRT.

De esta manera, **MobileNet** es una opción ideal para la Jetson Nano, porque logra un equilibrio entre precisión y eficiencia, permitiendo ejecutar modelos de visión artificial en tiempo real en un dispositivo compacto y económico. Este red tiene una parte de **codificador** y **decodificador**, que se encargan de la extracción de las características relevantes de la imagen de entrada reduciendo la dimensionalidad (variables latentes), y de la restauración de los mapas de características a la resolución original, creando así la clasificación de los píxeles a partir de las variables latentes.

En este estudio, se usó la técnica de **Transfer Learning** la cual consiste en utilizar modelos preentrenados para una nueva tarea. Se tratan de modelos convolucionales profundos que ya han aprendido a reconocer patrones, características y estructuras de alto nivel en imágenes (parte correspondiente al codificador). Esta técnica se realiza cuando se disponen de pocos datos de entrenamiento disponibles y se quiere construir un modelo que generalice correctamente. También se opta por esta solución cuando se desea reducir el tiempo de entrenamiento o mejorar el rendimiento en tareas similares sin realizar el entrenamiento desde cero. Por ello se utilizan datasets que favorecen esta transferencia de conocimiento para alcanzar mejores resultados. En el estudio realizaron estos experimentos basándose en la métrica de evaluación **mIoU**(*Intersección Media sobre Unión*).

Tabla 5. Resultados Transfer Learning Control Maleza

| Data | OS = 8 | OS = 16 | OS = 32 |
|--------------------------|--------|---------|---------|
| Pre-Entrenamiento | | | |
| PASCAL | 75 % | 72 % | 66 % |
| ADE20k | 76 % | 74 % | 68 % |
| CityScape | 78 % | 74 % | 68 % |

Donde **OS** (*Paso de Salida*) es un hiperparámetro del modelo que define la relación entre el tamaño de la imagen de entrada y el tamaño del último mapa de características de salida en el codificador.

3.5. Optimización IA

El Edge AI como se ha expuesto, consiste en ejecutar modelos de IA directamente en dispositivos locales. Estos dispositivos tienen recursos limitados, por lo que optimizar al máximo el modelo permite una respuesta más rápida y una eficiencia de recursos mayor. Los frameworks más comunes en el desarrollo de la IA son **Tensorflow** y **Pytorch**. Algunas bibliotecas de estos marcos de trabajo, han sido desarrolladas específicamente para permitir la optimización:

- **TensorRT[13]**: biblioteca de optimización e inferencia acelerada desarrollado por Nvidia. Se integra con Tensorflow y se usa cuando el modelo está entrenado (parte de inferencia). Algunas de las técnicas que aplica son fusión de capas, cuantización, eliminación de nodos redundantes y ejecución de batch optimizado.
- **TensorFlow Lite[14]**: versión ligera de Tensorflow diseñada específicamente para móviles y sistemas embebidos. TensorRt está enfocado en dispositivos de GPU de Nvidia, mientras que TensorFlow Lite es más global.
- **PytorchScript[15]**: forma de convertir modelos de Pytorch (dinámicos y estáticos en Python) a otro ejecutable más eficiente y sin necesidad de Python. Esto se debe a que en dispositivos hardware a veces no es compatible Python, por lo que se debe de ejecutar en otros lenguajes como puede ser C.

3.6. Resumen Estado del Arte

En resumen, para procesar imágenes utilizando algoritmo de IA es preferible utilizar redes convolucionales o **CNN** debido a su ventaja en cuanto a la consideración de las relaciones espaciales de los píxeles. Para asignar un etiquetado a la imagen completa, se deben aplicar algoritmos de **clasificación de imágenes**, mientras si el etiquetado se requiere a nivel de píxel, se usan algoritmos de **segmentación semántica**. Por último, debido al procesamiento en el borde, se requieren soluciones optimizadas para poder ejecutar estos algoritmos.

En cuanto al dispositivo final del Edge Computing, la **Jetson Nano** es más flexible y contiene mayor capacidad de procesamiento, además de estar orientada a la IA, por lo que es una opción más adecuada para el Edge AI que la Raspberry Pi.

Debido a que la **solución propuesta** es identificar parte de maleza en hojas de cultivo, se debe realizar una clasificación a nivel de píxel, por lo que se deben optar por soluciones de **segmentación semántica**. Estos modelos deben ser compatibles con su ejecución con el software de la Jetson Nano.

4. Especificaciones técnicas y restricciones de diseño

Este capítulo pretende definir los requisitos funcionales y no funcionales, además de las restricciones de diseño, que deben ser considerados durante el desarrollo de la solución propuesta. Estos aspectos resultan fundamentales para garantizar que el sistema cumpla con las expectativas del usuario final y con los criterios técnicos definidos más adelante, en la descripción de la propuesta.

4.1. Especificaciones del diseño

- **Requisitos funcionales:** Los requisitos funcionales que el sistema de detección de anomalías en cultivos basado en Edge AI son:

Tabla 6. Requisitos funcionales

| Requisito | Descripción |
|-----------|--|
| RF1 | Captura e inferencia de datos en el borde del sistema. |
| RF2 | Detectar anomalías en cultivos mediante cámara. |
| RF3 | Mantener un registro temporal de las anomalías. |
| RF4 | Generar estadísticas o reportes de los registros temporales. |
| RF5 | Interfaz de consulta de estadísticas o reportes. |

- **Requisitos no funcionales:** Los requisitos no funcionales del sistema de detección de anomalías en cultivos basado en Edge AI son:

Tabla 7. Requisitos funcionales

| Requisito | Descripción |
|-----------|---|
| RNF1 | Tiempo inferencia en el borde menor que en la nube (u otras arquitecturas). |
| RNF2 | Capacidad de soporte de procesamiento por la Jetson Nano. |
| RNF3 | Disponibilidad del sistema el 95 % de la jornada laboral del agricultor. |
| RNF4 | Almacenamiento de imágenes y estadísticas local y acceso autorizado. |
| RNF5 | Software del microordenador debe permitir actualizaciones remotas sin necesidad de intervención física. |
| RNF6 | Escalabilidad y flexibilidad del sistema a integración con nuevos sensores. |

4.2. Restricciones de diseño

Las restricciones de diseño que determinan las decisiones técnicas son las representadas en la **Tabla 8**.

Tabla 8. Restricciones de diseño

| Restricción | Tipo |
|---|-------------|
| El sistema debe operar entre temperaturas de -10 y 50°C. | Ambiental |
| El sistema debe ser resistente a salpicaduras de herbicida | Ambiental |
| El sistema debe poder funcionar en superficies irregulares y con presencia de obstáculos. | Ambiental |
| La solución debe cumplir con el estándar IEEE 802.11n. | Normativa |
| Solo se pueden utilizar bibliotecas, frameworks y herramientas compatibles con el software Jetpack 4.6 de la Jetson Nano. | Técnico |
| El modelos debe estar optimizado para ejecutarse con un uso inferior a 4GB RAM. | Técnico |
| La cámara debe tener un resolución suficiente para capturar adecuadamente el cultivo. | Técnico |
| El sistema de herbicida debe tener una precisión óptima para su aplicación. | Técnico |

5. Descripción de la propuesta

5.1. Descripción del caso de uso

La solución propuesta se ha basado en el estudio de "Aplicación de Control de Maleza en el Borde" descrito en el apartado 2.4.1 del documento, aunque con funcionalidades añadidas y una arquitectura distinta debido a las características de la Jetson Nano Developer Kit E. La anomalía en cuestión, como el nombre del estudio indica, es la detección de la maleza a través de la IA mediante procesamiento de imágenes en la Jetson Nano, incorporada en un UGV con un sistema automático de pulverización de herbicida. De esta manera, el agricultor/a obtiene las ventajas de la propuesta mediante la monitorización continua del cultivo.

5.2. Modelo caso de uso

El Caso de Uso de la **Figura 6** representa el núcleo de las funcionalidad del sistema de detección de anomalías y las interacciones con los dos principales actores: **UGV** y **Agricultor/a**.

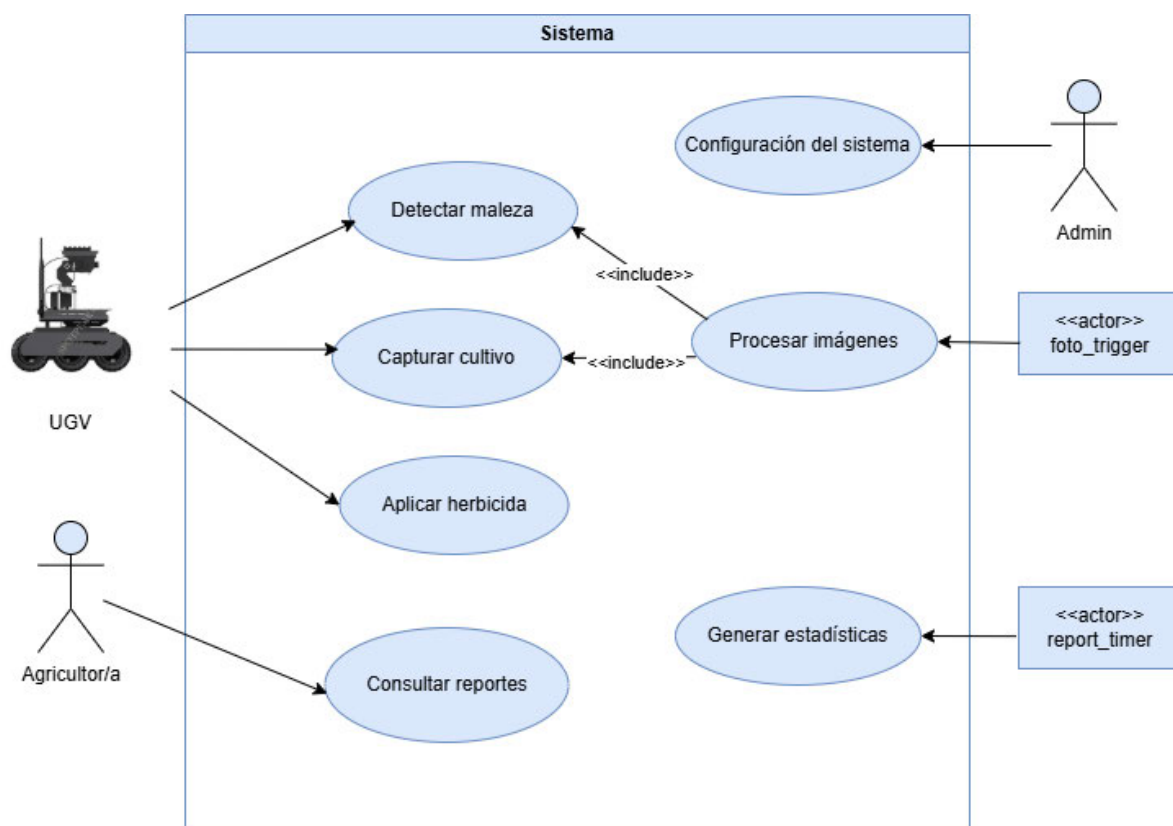


Figura 6. Caso de Uso Modelo

1. **Admin:** El actor Admin es responsable de la configuración del sistema, que incluye configurar la Jetson Nano, la configuración del modelo de IA, de la base de datos y la comunicación con la aplicación que genera los reportes. Este rol garantiza que el sistema funcione correctamente ajustando la configuración inicial y administrando las configuraciones del sistema según sea necesario.
2. **UGV:** Actor responsable de realizar la foto al cultivo y de ejecutar el modelo de IA que permita la detección de maleza a través de una imagen de entrada. Estas funcionalidades se corresponden las del dispositivo final, la Jetson Nano, incorporada en el UGV. Aparte el UGV debe aplicar herbicida automáticamente a la parte de la planta correspondiente a donde se encuentra la anomalía.
3. **Agricultor/a:** Usuario final que puede consultar las estadísticas generadas a partir del procesamiento de las imágenes de su cultivo. Con esto se garantiza la monitorización del mismo, permitiendo al agricultor/a aprovecharse de las ventajas del sistema.
4. **foto_trigger** y **report_timer:** Estos dos actores representan tanto el mecanismo de procesamiento de imágenes como el de generación de estadísticas asociadas a la imagen de salida del algoritmo de IA, respectivamente.
 - **foto_trigger:** trigger o evento que sucede cuando se toma un foto del cultivo por parte del UGV. Este actor, está compuesto del actor UGV. Tras la foto, se procesa la imagen para su introducción en el algoritmo de IA.
 - **report_timer:** timer o temporizador encargado de calcular las estadísticas asociadas a las predicciones.

El límite del sistema en el modelo de caso de uso define claramente el alcance de cada funcionalidad. Tanto el UGV como el/la Agricultor/a actúan a través de roles específicos, lo que garantiza el acceso a las herramientas necesarias para desempeñar sus funciones. Al tener únicamente el/la agricultor/a acceso a sus datos del cultivo y el almacenamiento de los mismos en su propio dispositivo final, se garantiza la privacidad de los datos. Este enfoque proporciona una solución eficiente y segura.

Para una mejor comprensión del sistema general se representa el diagrama de secuencia de la **Figura 8**, que representa la interacción de todos los componentes entre ellos para detectar la maleza en el cultivo con su consecuente monitoreo. Representa la descripción del caso uso en el apartado **4.1** del documento.

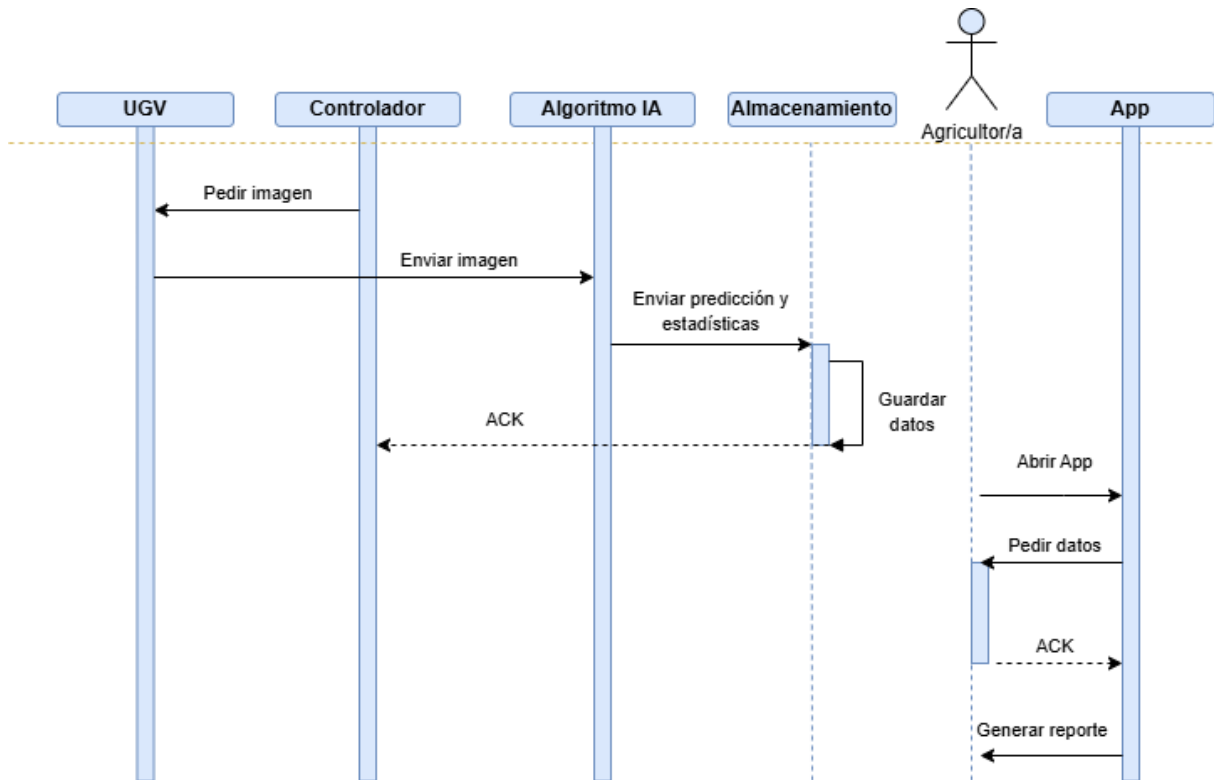


Figura 7. Diagrama de secuencia general

El componente **Controlador** se encarga del control de procesos desde la captura de datos, procesos internos de sincronización y eventos del sistema, inferencia, almacenamiento de los datos y monitorización de los mismos.

La **App** por su lado se encarga de la visualización de la evolución del estado de la maleza en los cultivos, mediante una interfaz que se conecta a la base de datos. Esta base de datos es el **Almacenamiento**, donde los datos son guardados de manera tabular, con capacidad de recuperación mediante secuencias SQL.

Las flechas representan las acciones a tomar entre las interacciones de dos o más componentes del sistema y las flechas de ACK corresponden a confirmaciones de comunicación entre estos componentes, para la verificación correcta de la comunicación y funcionamiento del sistema.

5.2.1. Detectar Maleza - Caso de uso

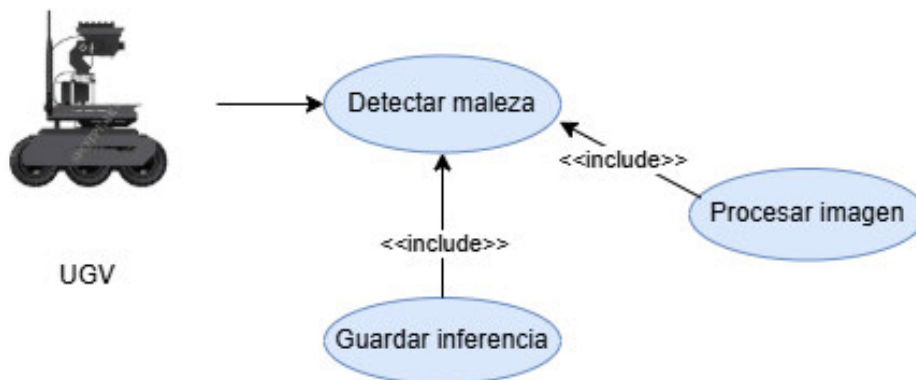


Figura 8. Detectar anomalía Caso de uso

La **Tabla 9** da la explicación detallada del caso de uso "Detectar Maleza".

Tabla 9. Detectar Maleza - Caso de Uso

| Caso de Uso 01 - Detectar Maleza |
|--|
| Actor: foto_trigger |
| Descripción: Procesamiento de la imagen de cultivo para detectar maleza |
| Pre-condiciones: Sistema configurado y foto capturada por la cámara |
| Pasos: <ol style="list-style-type: none">1. Cuando el UGV realiza la foto, salta el trigger del sistema2. La imagen pasa como argumento de entrada al sistema de procesamiento (algoritmo de IA)3. Inferencia del modelo y salida de predicción4. Conexión con la base de datos5. Se guarda la imagen predicha con la clasificación de cada pixel |
| Post-condiciones: Se ha realizado la inferencia y almacenamiento de la predicción del cultivo |

5.2.2. Generar estadísticas - Caso de uso

Tabla 10. Generar estadísticas - Caso de uso

| Caso de Uso 02 - Generar estadísticas |
|--|
| Actor: report_timer |
| Descripción: Generación de estadísticas |
| Pre-condiciones: Predicción de la maleza |
| Pasos: <ol style="list-style-type: none">1. Calcular el porcentaje de cada clase en la imagen2. Abrir conexión con la base de datos3. Guardar datos en formato tabular4. Conexión con la aplicación de visualización de reportes |
| Post-condiciones: Se han almacenado las estadísticas y se han generado los reportes del cultivo |

5.3. Diseño de la arquitectura

En esta sección, se explica la arquitectura del sistema. La arquitectura es esencial para entender como interactúan los subsistemas y componentes. La arquitectura general del sistema ha sido diseñada de manera modular, permitiendo una separación clara de responsabilidades y facilitando futuras ampliaciones o mantenimientos. En la **Figura 9** se desglosa en subsistemas la arquitectura general de la solución propuesta.

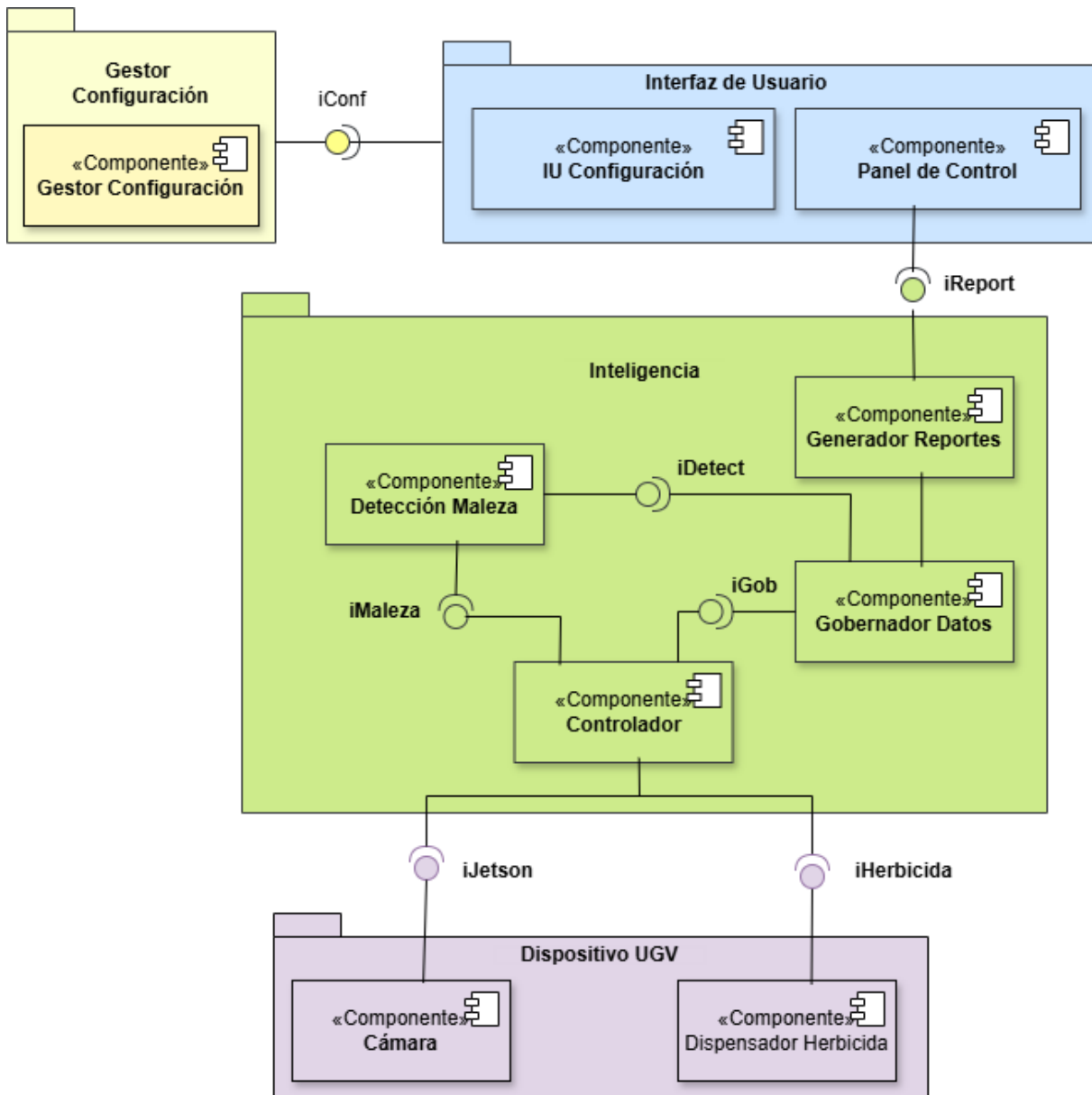


Figura 9. Componentes Arquitectura

Los subsistemas que componen la arquitectura son:

- **Dispositivos UGV:** Este subsistema agrupa los sensores/actuadores del sistema, es decir, los dispositivos hardware. Estos componentes son la **Cámara** que se encarga de capturar las imágenes del cultivo. Tras su captura, se comunica con el Controlador del Sistema a través de la interfaz de comunicación llamada **iJetson**. El otro componente trata del **Dispensador de Herbicida**, siendo un actuador que echa la herbicida a la parte de la hoja del cultivo donde detecta maleza. Esta última acción la realiza gracias a las órdenes dirigidas por el Controlador a través de la interfaz **iHerbicida**.
- **Inteligencia:** Define la parte de inteligencia del sistema completo, es decir, la parte de control de datos y ejecución del algoritmo de IA. Ese subsistema se compone de cuatro componentes distintos:
 - **Controlador:** Se encarga de la comunicación de las acciones entre los componentes de la capa de Dispositivos y los componentes de Inteligencia. Este componente, se comunica con los dispositivos a través de las interfaces iJetson e iHerbicida. Por otro lado, para la comunicación con el algoritmo de IA que predice la maleza, se conecta con el componente **Detección de Maleza** a través de la interfaz **iMaleza** para proporcionarle la imagen capturada.
 - **Detección de Maleza:** Ejecuta el algoritmo de predicción de maleza tras recibir la imagen de entrada, a través de la interfaz **iMaleza**. La predicción en cuestión se la envía al **Gobernador Datos** a través de la interfaz **iDetect**. La salida generada por el componente Detección de Maleza es un conjunto de coordenadas donde se detecta la maleza.
 - **Gobernador Datos:** Encargado de recolectar todos los datos que se capturan y generan en el sistema. Estos datos llegan al **Generador de Reportes** para que pueda realizar el cálculo de estadísticas y reportes y al **Controlador** para que envíe las ordenes de los puntos espaciales donde se debe dispersar la herbicida.
 - **Generador Reportes:** Componente encargado del cálculo de las estadísticas y reportes a través de los datos proporcionados por el **Gobernador Datos**.
- **Intefaz de Usuario:** Herramienta o aplicación utilizada para la visualización de los reportes generador por la capa de Inteligencia. Los dos componentes que contiene son **IU Configuración** el cual contiene los ajustes de la interfaz de usuario y el **Panel de Control**, siendo esta última la responsable de poder visualizar los datos. El/la **Agricultor/a** accede a este componente para realizar la monitorización del cultivo
- **Gestor de Configuración:** El subsistema responsable de la configuración del sistema, incluyendo inicialización de la **Jetson Nano** además de la configuración de la herramienta de visualización. Este componente interactúa con la **Interfaz de Usuario** a través de la interfaz **iConfig**.

5.4. Diagrama de despliegue

La figura del **Diagrama de despliegue** representa la arquitectura física del sistema, es decir, que componentes del software se ejecutan en cada nodo del hardware y los protocolos de comunicación entre estos. Se detalla la integración de la solución en el mundo real para garantizar la funcionalidad y el flujo de datos que sigue la solución propuesta.

Gracias a esta visualización, el desarrollo se realiza de manera más efectiva abogando por la estabilidad del sistema. La **Figura 10** representa el diagrama en cuestión.

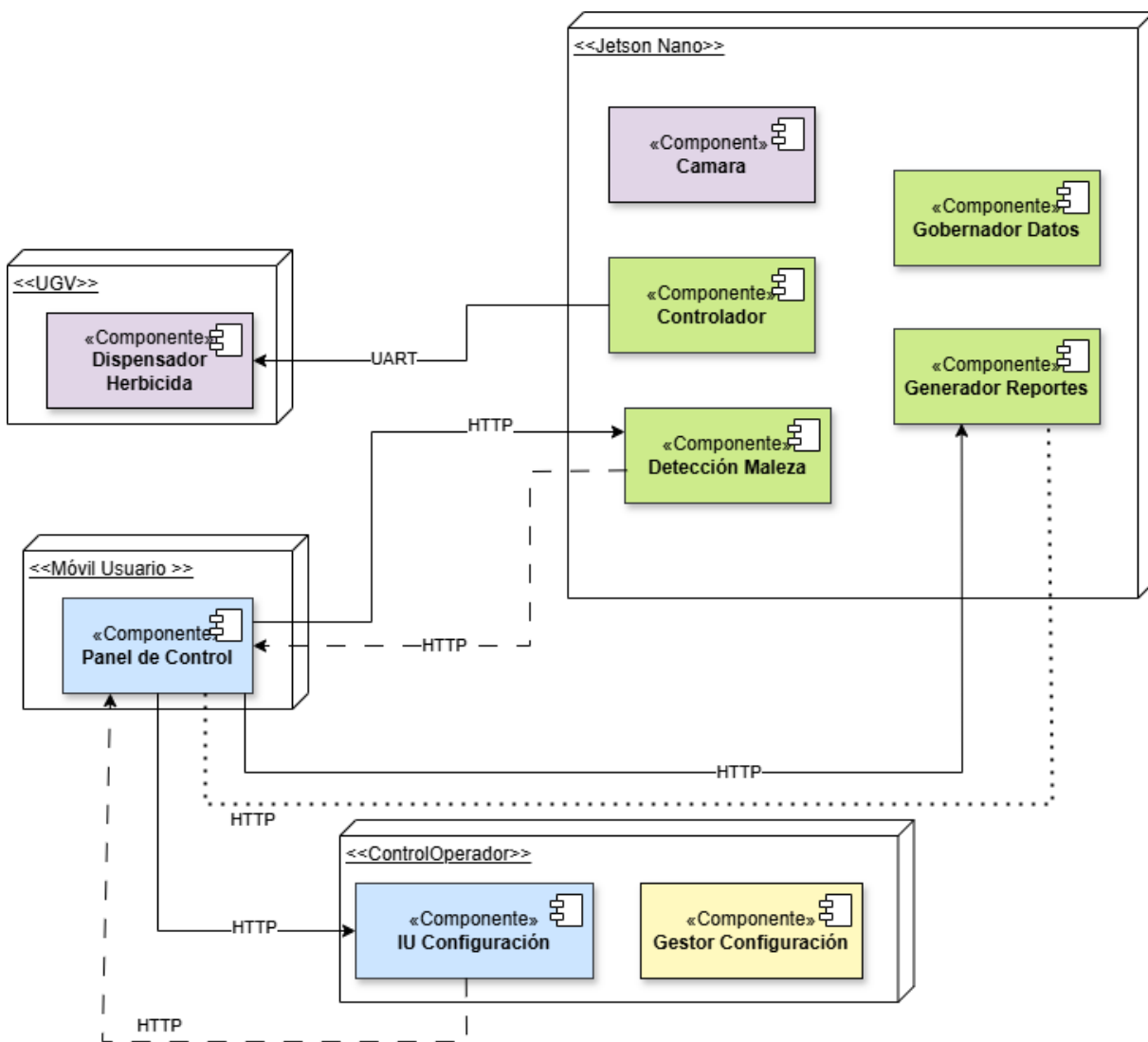


Figura 10. Diagrama de Despliegue

- **UGV (Dispensador Herbicida):** El *Dispensador de Herbicida* recibe por parte del *Controlador* las coordenadas de la posición del cultivo en cuestión donde debe dispersar el herbicida. Al tomar la imagen, se inicia el procesado de la misma, se determina en qué coordenadas hay que aplicar herbicida y se envía la orden y las coordenadas al dispensador de herbicida. Es importante tener en cuenta el tiempo de procesamiento y velocidad del UGV para realizar la última acción.

El protocolo de comunicación es **UART** (*Transmisor-Receptor Asíncrono Universal*) ya que los sistemas de herbicidas suelen ser sistemas menos complejos, como microcontroladores (Arduino, STM32, etc.), por lo que no tienen capacidad de recibir grandes datos como pueden ser imágenes. Por ello se envían las coordenadas (x,y). Este tipo de protocolo de comunicación está pensado para enviar datos pequeños, sin una confirmación de llegada de datos.

- **Jetson Nano:** El *Controlador* como se ha mencionado anteriormente se comunica con el *Dispensador de Herbicida*, para indicar donde debe actuar el herbicida. El *Controlador* como anteriormente se ha comentado, recibe la imagen capturada por la *Cámara* del cultivo.

Al estar la *Cámara* conectada directamente con la Jetson Nano, no hace falta un protocolo de comunicación para la emisión de estos datos. Seguidamente, está el algoritmo de IA, *Detección de Maleza*, y su predicción junto con la generación de estadísticas, se realiza todo dentro de la Jetson Nano, en *Generador de Reportes*.

- **Móvil Usuario:** Consiste en la herramienta de visualización que utilizará el/la Agricultor/a a través de su dispositivo móvil. De tal forma, se garantiza la accesibilidad, ya que hoy en día todo el mundo dispone de un teléfono móvil. Este protocolo de comunicación se debe realizar mediante **HTTP** (*Protocolo de Transferencia de Hipertexto*) sin necesidad de **HTTPS** (*Protocolo Seguro de Transferencia de Hipertexto*) ya que los datos de una planta no son datos sensibles, por lo que no están sujetas a amenazas de seguridad.

HTTP es posible ya que la Jetson Nano contiene módulos WIFI para comunicación y la mayoría de sistemas de dispersión de herbicida también lo soportan. Este protocolo es adecuado, y es esencial una confirmación, de ahí las flechas discontinuas que representan la confirmación del paquete de información.

- **Control Operador:** La *IU Configuración* proporciona una comunicación por parte de petición del Agricultor/a para cambiar la personalización de la visión de los reportes generador. Esto se realiza también mediante el protocolo HTTP. Por último, se encuentra el *Gestor Configuración* encargado de la configuración completa del sistema en términos generales.

5.5. Implementación

En esta sección, se expone de manera detallada el comportamiento del sistema de manera textual y visual para ambos casos de uso, *Detección de Maleza* y *Generar estadísticas*. Por último, se expone la simulación de la monitorización de la evolución del estado del cultivo.

5.5.1. Estrategia de Despliegue del Modelo de IA

La estrategia tomada para el despliegue del modelo de IA utilizado para esta solución se ha dividido en dos fases:

- **Entrenamiento del modelo:** se ha entrenado en un ordenador portátil a modo de dispositivo centralizado para ajustar el modelo antes de su exportación a la Jetson Nano. Se ha realizado de esta manera por la capacidad reducida de procesamiento y la limitación de memoria de la Jetson Nano, por lo que de esta manera el proceso de entrenamiento es mucho más rápido.
- **Inferencia del modelo:** esta parte se realiza en la Jetson Nano, gracias a la exportación del modelo entrenado en el paso anterior. Esto es posible gracias a un archivo con extensión **.h5**, siendo un archivo HDF5 (*Formato de Datos Jerárquico versión 5*) para el almacenamiento de datos estructurados de manera eficiente. Gracias a este formato de archivo se puede almacenar:
 - **Arquitectura del modelo:** organización de las capas y conexiones entre ellas.
 - **Pesos entrenamiento:** valores de los pesos aprendidos durante el entrenamiento.
 - **Configuración de entrenamiento:** que optimizador, función de pérdidas y métricas se obtuvieron durante el entrenamiento.

Es decir, este formato de archivo permite de manera eficiente la migración del modelo entre dispositivos para la correcta inferencia con los valores aprendidos en la etapa de entrenamiento y poder realizar una tarea eficiente de detección de maleza.

5.5.2. Detección de Maleza

La detección de maleza supone el principal caso de uso del sistema completo. En la solución planteada las imágenes de los cultivos son capturados por una cámara integrada en el UGV, sin embargo, para este estudio se han simulado los datos utilizados a partir de un dataset de libre uso. Este dataset, llamado "**CWFID (Crop/Weed Field Dataset)**" contiene imágenes etiquetadas de la parte del cultivo sano y de la maleza. Para detalles más específicos del dataset, consulte el Anexo llamado "**CWFID Dataset**".

1. **Captura imagen y envío:** En esta primera fase, el sistema toma fotos mediante la cámara situada en el UGV y seguidamente realiza su envío al **Controlador** del sistema.



Figura 11. Cultivo capturado

2. **Algoritmo de IA:** Se encarga de la parte de inteligencia del sistema. La imagen de salida del paso anterior, es la entrada de este segundo paso. La arquitectura utilizada es una red de segmentación semántica, ya que se desea la clasificación a nivel de píxel y no de imagen. Se extraen las características a nivel espacial, textura u otros patrones que diferencien ambas clases para entrenar el algoritmo.

Se realiza la técnica de Transfer-Learning debido a los pocos datos disponibles y el algoritmo utilizado es una CNN. La CNN utilizada es U-Net, donde la Figura 12 representa su arquitectura.

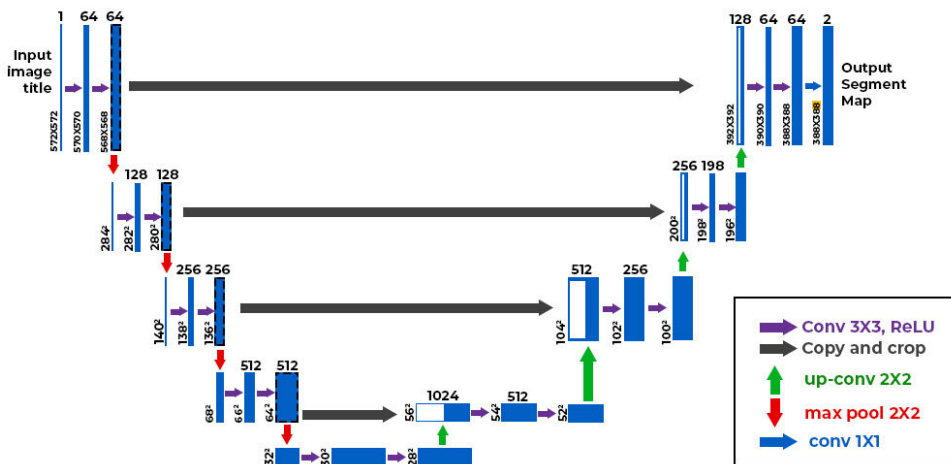


Figura 12. Arquitectura Unet [16]

El nombre U-Net se debe a la forma de U que tiene. La salida de este algoritmo de IA consiste en la generación de una imagen con el conjunto de colores asociados a tres clases:

- **Clase Fondo:** representa el fondo de la imagen, es decir, lo que no supone la hoja del cultivo. Se representa con color rojo.
- **Clase Verde:** representa la parte del cultivo común, es decir lo que no es maleza. Se representa con color verde.
- **Clase Naranja:** representa la maleza de la parte del cultivo, es decir, donde se debe echar la herbicida. Se representa en color naranja.

La **Figura 13** representa la predicción realizada por el algoritmo de IA representando con los colores respectivos las diferentes clases segmentadas.

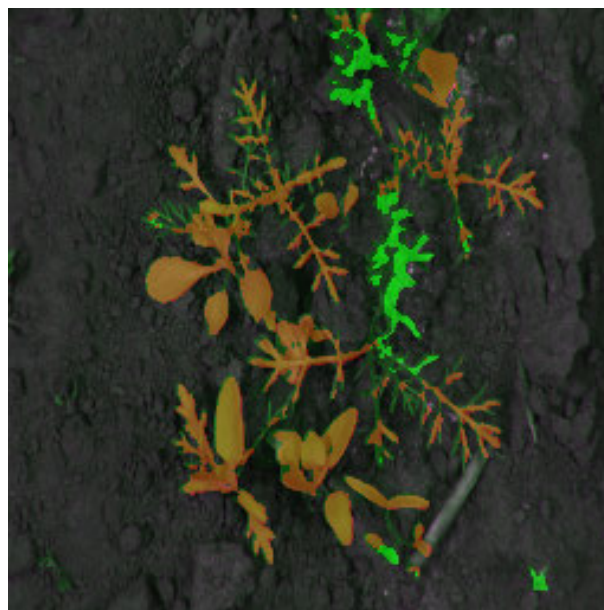


Figura 13. Imagen predicha

5.5.3. Generar Estadísticas y Almacenamiento

La generación de estadísticas se corresponde con la parte de monitoreo del sistema para corroborar la funcionalidad del sistema. El parámetro de entrada es la imagen predicha, la Figura 12, a la salida del algoritmo de IA. Una vez se tenga la imagen, se dispone a contar el número de píxeles asociados a cada clase y calcular el porcentaje de cada uno de ellos, es decir;

- **(%) Fondo:** porcentaje de píxeles asociados al fondo, es decir, que tengan color negro, sobre el total de píxeles en la imagen.
- **(%) Verde:** porcentaje de píxeles asociados a la parte de la hoja del cultivo real, es decir, que tengan color verde, sobre el total de píxeles en la imagen.
- **(%) Rojo:** porcentaje de píxeles asociados a la maleza, es decir, que tengan color naranja, sobre el total de píxeles en la imagen de la predicción. Se etiqueta como rojo ya que la máscara de la predicción utiliza este color, sin embargo, para su visualización es naranja, debido a que se ha hecho una combinación con la imagen capturada para una representación más visual.

Una vez se calculan estos parámetros se procede al almacenamiento de los datos en la base de datos. Debido a la naturaleza tabular de la información, se ha utilizado una base de datos **relacional**. Más concretamente, esta base de datos, se encuentra dentro de la Jetson Nano, debido a que la información a almacenar no ocupa mucha cantidad de datos. La Figura 14 y la Figura 15 representa una visualización de los datos guardados.

```
mysql> SELECT * FROM Carrot_Herbicide;
```

| id | timestamp | img_path | pred_path |
|-------|---------------------|--|---------------|
| 001 | 2025-01-02 10:00:00 | /home/jetson/Desktop/CWFID/dataset-1.0/images/001_images.png | /home/jetson/ |
| 001_1 | 2025-01-17 10:00:00 | /home/jetson/Desktop/CWFID/dataset-1.0/images/001_1_images.png | /home/jetson/ |
| 001_2 | 2025-02-01 10:00:00 | /home/jetson/Desktop/CWFID/dataset-1.0/images/001_2_images.png | /home/jetson/ |
| 001_3 | 2025-02-16 10:00:00 | /home/jetson/Desktop/CWFID/dataset-1.0/images/001_3_images.png | /home/jetson/ |
| 001_4 | 2025-03-03 10:00:00 | /home/jetson/Desktop/CWFID/dataset-1.0/images/001_4_images.png | /home/jetson/ |
| 001_5 | 2025-03-18 10:00:00 | /home/jetson/Desktop/CWFID/dataset-1.0/images/001_5_images.png | /home/jetson/ |
| 001_6 | 2025-04-02 10:00:00 | /home/jetson/Desktop/CWFID/dataset-1.0/images/001_6_images.png | /home/jetson/ |
| 001_7 | 2025-04-17 10:00:00 | /home/jetson/Desktop/CWFID/dataset-1.0/images/001_7_images.png | /home/jetson/ |
| 001_8 | 2025-05-02 10:00:00 | /home/jetson/Desktop/CWFID/dataset-1.0/images/001_8_images.png | /home/jetson/ |
| 001_9 | 2025-05-17 10:00:00 | /home/jetson/Desktop/CWFID/dataset-1.0/images/001_9_images.png | /home/jetson/ |

Figura 14. Base de Datos I

| | green | red | background |
|-------------------|-------|-------|------------|
| ge_Jetson.png | 4.65 | 10.09 | 85.26 |
| ge_Jetson_t+1.png | 5.9 | 8.84 | 85.26 |
| ge_Jetson_t+2.png | 6.52 | 8.22 | 85.26 |
| ge_Jetson_t+3.png | 7.15 | 7.59 | 85.26 |
| ge_Jetson_t+4.png | 7.77 | 6.97 | 85.26 |
| ge_Jetson_t+5.png | 8.4 | 6.34 | 85.26 |
| ge_Jetson_t+6.png | 9.02 | 5.72 | 85.26 |
| ge_Jetson_t+7.png | 9.65 | 5.09 | 85.26 |
| ge_Jetson_t+8.png | 10.27 | 4.47 | 85.26 |
| ge_Jetson_t+9.png | 10.9 | 3.85 | 85.26 |

Figura 15. Base de Datos II

La tabla almacenada en la base de datos consta de además los porcentajes comentados anteriormente, estos nuevos campos:

- **id:** Clave Primaria (**PK**) de la tabla que identifica inequívocamente a cualquier imagen del cultivo. La primera identifica la cuadrícula, la planta, que se está monitoreando, por lo que,

$$\text{id1} \in \{x \in \mathbb{Z} \mid 1 \leq x \leq 60\}$$

refiriéndose a la primera parte del id. La segunda parte del identificador es un número de secuencia que permite saber en que orden se tomaron las diferentes imágenes de la misma planta, de la misma cuadrícula (cada número indica el número de veces que ha actuado la herbicida), es decir, la segunda parte corresponde con,

$$\text{id2} \in \{x \in \mathbb{Z} \mid 0 \leq x \leq n\}, \quad n \in \mathbb{N}_0$$

En resumen, id da información de que cultivo y con que número de imagen corresponde con,

$$\text{id} = \text{id1_id2}$$

Una aproximación puede ser que el terreno total de cultivo esté numerado en parcelas, y cada parcela sea el id1 asociado a cada planta, pudiendo de esta forma identificar en el espacio que planta ha sido capturada, ya que distinguir el mismo cultivo a partir de diferentes imágenes es complicado y costoso.

- **timestamp:** marca de tiempo, que indica el momento en el que se ha capturado la foto.
- **img_path:** ruta de la imagen captura donde ha sido guardada dentro del sistema de la Jetson Nano.
- **pred_path:** ruta de la imagen predicha donde ha sido guardada dentro del sistema de la Jetson Nano.

Como se puede observar en la Figura 15, los porcentajes de verde y naranja van aumentando/disminuyendo a medida que están asociados a instantes de tiempo distintos, simulando la obtención de muestras de la misma imagen, tras haber aplicado herbicida un número distinto de veces.

Este sistema de almacenamiento es escalable ya que, se pueden almacenar dentro de la misma base de datos, diferentes tablas, cada una de ellas referenciando a diferentes terrenos de cultivos que puedan estar en distintas zonas geográficas.

5.5.4. Monitorización del cultivo

Como se ha comentado en la arquitectura, la parte de monitorización del cultivo es esencial para que el Agricultor/a tenga un control sobre el dispensador de herbicida sobre sus plantas.

Para su monitoreo se pueden utilizar herramientas que acceden a los datos de la base de datos y generan estas gráficas para facilitar esta información. Algunas de las herramientas más conocidas son **Tableau**, **Power BI** o **Grafana**. Se trata de herramientas de visualización de datos que permiten crear informes, paneles y gráficos interactivos a partir de diversas fuentes de datos. Tableau y Power BI se enfocan en Business Intelligence, mientras que Grafana está más orientada al monitoreo en tiempo real de sistemas e infraestructuras.

En este caso, al tratarse de un sistema no muy crítico, no es necesario un monitoreo explícitamente en tiempo real, por lo que cualquiera de las tres herramientas sería una buena opción. La gráfica a continuación ha sido creada utilizando la biblioteca **Matplotlib** de Python. La Figura 16 representa la evolución del porcentaje de maleza (**color rojo**) y de la parte de la zanahoria verdadera (**color verde**).

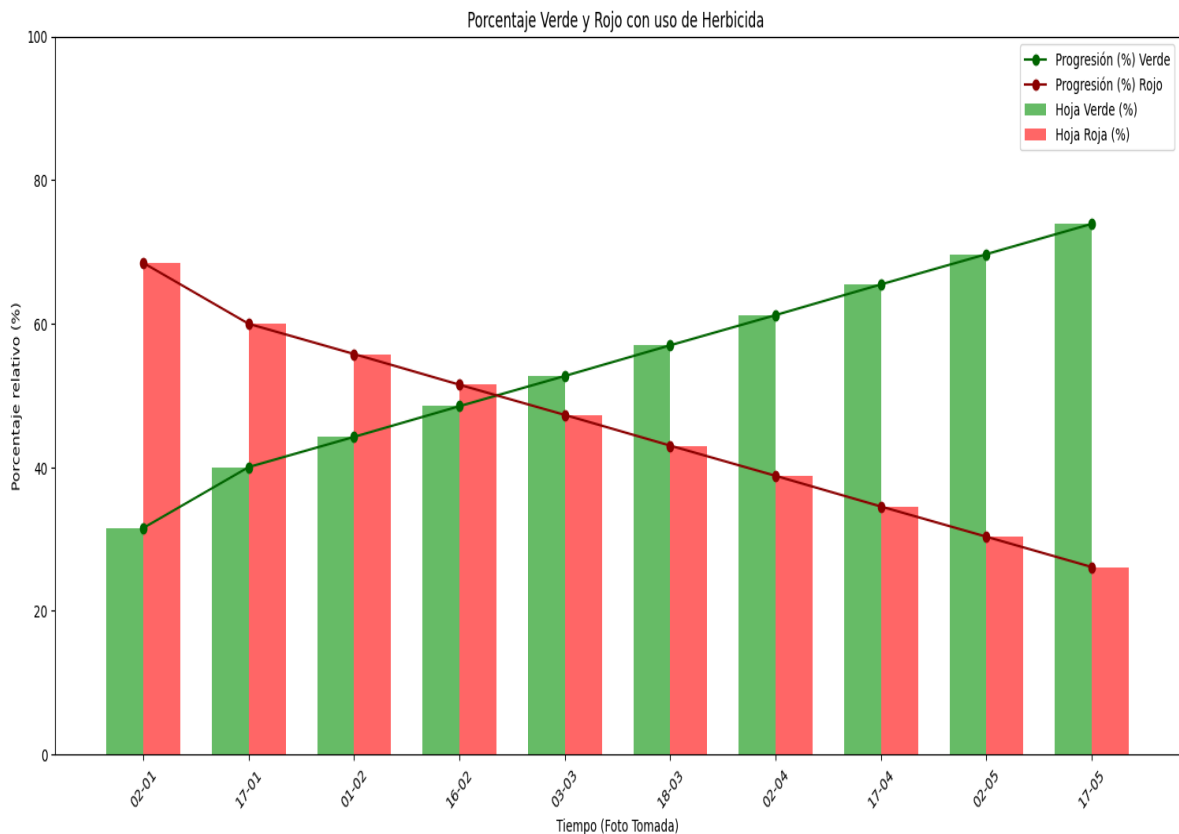


Figura 16. Gráfica Monitoreo

El **eje X** representa el instante de tiempo en el que se toma la foto y en el **eje Y** los valores de los porcentajes relativos de maleza y zanahoria sin tener en cuenta el fondo (sobre la misma imagen tiene el mismo valor). La evolución del cultivo que se muestra en la figura anterior, correspondería a una situación en la que la herbicida va anulando la maleza progresivamente.

6. Pruebas y Resultados

Con el objetivo de garantizar la robustez, precisión y fiabilidad del sistema desarrollado, se ha llevado a cabo una exhaustiva fase de validación experimental. En esta etapa, se han analizado de forma sistemática los distintos algoritmos de Inteligencia Artificial aplicados a la detección de maleza, evaluando su comportamiento en escenarios simulados que replican condiciones reales de operación en campo.

Las pruebas contemplan tanto el análisis cuantitativo mediante métricas objetivas como la evaluación cualitativa mediante análisis visual. Este enfoque combinado permite contrastar la efectividad de los distintos modelos implementados y facilitar la selección de la arquitectura más adecuada en términos de equilibrio de rendimiento computacional y precisión en la tarea de segmentación semántica.

6.1. Pruebas algoritmos IA

Como bien se ha comentado en el caso de uso, la clasificación se corresponde con la detección de maleza, por lo que al necesitar una clasificación a nivel píxel, se usan algoritmos de segmentación semántica.

Dado que el desarrollo del sistema se ha llevado a cabo utilizando la librería de Python **Tensorflow**, como framework base, y su interfaz de alto nivel **Keras**, las pruebas realizadas se han llevado a cabo con arquitecturas compatibles con Keras. Esto permite una implementación más eficiente, reproducible y evita complicaciones derivadas de la incompatibilidad con la Jetson Nano, que es donde se debe ejecutar el algoritmo.

En el estudio relacionado con el capítulo 2.4.1, se utilizó una arquitectura **DeepLab**. Esta arquitectura ofrece resultados ligeramente mejores que la arquitectura final planteada, sin embargo, su tiempo de inferencia es mayor que la planteada, además de su no incompatibilidad con Keras. Esta incompatibilidad produce problemas con librerías y su alta complejidad para resolverlo descartó esta opción, ya que la idea es realizar una solución compatible con gran parte de dispositivos hardware, haciéndola además una solución estable y escalable.

Los algoritmos probados durante las pruebas de entrenamiento, que son soportados por Keras y cumplen con los requisitos de ejecución en la Jetson Nano, han sido los siguientes:

- **U-Net:** red neuronal convolucional especialmente eficaz en tareas biomédicas. Su estructura tiene formato de U.
- **FPN:** especializada en detección multiescala con la captura tanto de detalles finos como información contextual.
- **LinkNet:** utiliza conexiones directas entre capas correspondientes, permitiendo una reconstrucción rápida y precisa con bajo coste computacional.

Se ha utilizado la técnica de **Fine-Tuning** al disponer de pocos datos en el data set. Usar pesos preentrenados obtenidos mediante el entrenamiento de redes neuronales convolucionales (CNN) con una gran cantidad de imágenes permite detectar de forma efectiva las características esenciales. En este caso, el conjunto de datos utilizados fue ImageNet, que contiene una gran cantidad de imágenes de plantas.

La Tabla 11 muestra los resultados del entrenamiento medidos por la métrica mIOU. La selección de los mejores parámetros para cada modelo ha sido realizada mediante la técnica de **GridSearch**, la cual es una técnica de combinación en malla, que elige las mejores combinación de hiperparámetros en base al resultado de la métrica.

Tabla 11. Resultados Entrenamiento Modelo IA

| Arquitectura | Data Pre-Entrenamiento | mIoU | Etapa |
|--------------|---------------------------|------|---------------|
| U-Net | ImageNet | 68 % | Entrenamiento |
| FPN | ImageNet | 79 % | Entrenamiento |
| LinkNet | ImageNet | 50 % | Entrenamiento |

La primera idea fue estudiar un indicador como la precisión, que mide la proporción del total de píxeles bien clasificados sobre el total de píxeles. En la primera implementación, la precisión es del 98 %, sin embargo, la mayor parte de la imagen es fondo, y al ser más fácil diferenciarlo que la maleza frente a la parte de la planta real, es más fácil de antemano tener un mejor indicador. Por lo tanto, se utilizó el **mIOU**, que considera el número de píxeles bien clasificados, conociendo el total de cada clase.

$$mIoU = \frac{1}{C} \sum_{i=1}^C \frac{TP_i}{TP_i + FP_i + FN_i}$$

Con esta aproximación se realiza una evaluación mucho más precisa del correcto funcionamiento del algoritmo teniendo en cuenta la proporción de cada clase.

6.2. Validación Resultados

Una vez presentados los valores de entrenamiento, es fundamental identificar los valores asociados a la parte de validación. La parte de validación es la inferencia sobre el modelo de datos nunca antes vistos, por lo que representa una aproximación de como se comporta el modelo frente a datos nunca antes vistos (simulación de nuevos datos capturados después del despliegue). Un modelo puede mostrar métricas muy buenas en el conjunto de entrenamiento, pero si hay gran diferencia con las métricas de evaluación, indica que el modelo está sobreajustado, lo que significa que no es capaz de generalizar y los patrones los ha aprendido de memoria en vez de buscar relaciones.

La Tabla 12 muestra los resultados en la parte de validación con la misma métrica que en la fase de entrenamiento.

Tabla 12. Resultados Validación Modelo IA

| Arquitectura | Data Pre-Entrenamiento | mIoU | Etapas |
|--------------|------------------------|------|------------|
| U-Net | ImageNet | 50 % | Validación |
| FPN | ImageNet | 40 % | Validación |
| LinkNet | ImageNet | 43 % | Validación |

Para un sistema real, al disponer de pocos datos específicos del cultivo de zanahorias, se opta por una solución más generalizable. Por lo que la **arquitectura final** optada es la **U-Net**. U-Net presenta un **menor tiempo de inferencia** en comparación con FPN. Esto lo convierte en una opción más eficiente para entornos con recursos limitados, como la **Jetson Nano**, donde es crucial optimizar tanto el rendimiento como el uso de memoria. Los valores de hiperparámetros asociados a la solución final son: **épocas = 15**, **tamaño_batch = 2**, **tasa aprendizaje = 0.0001**, **momentum = 0.9** y **epsilon = 1e-8**

La Figura 17 representa la evolución de las gráficas de la función de pérdidas, valor de la métrica durante el proceso de entrenamiento y validación.

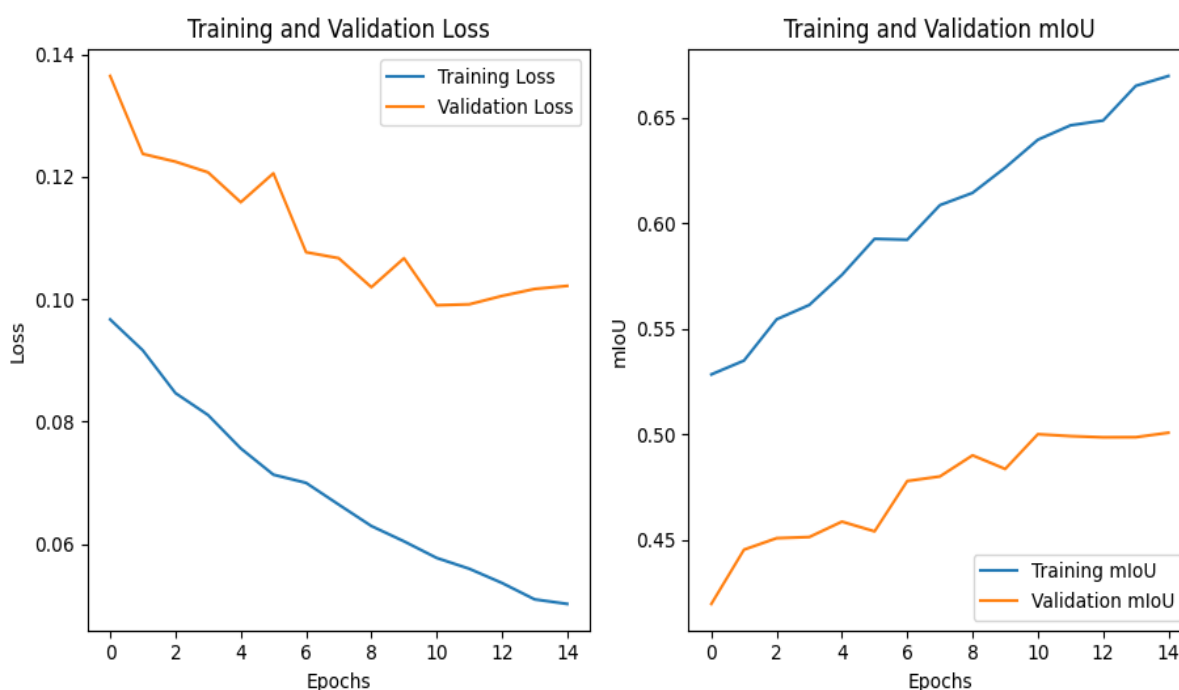


Figura 17. Pérdidas y Métricas U-Net

Loss corresponde a la función de pérdida, *Training* a la parte de entrenamiento, *Validation* a la validación y *Epochs* al número de épocas.

6.3. Validación Optimización IA

Como se define en el capítulo 2.5, es necesario optimizar el algoritmo de IA para poder optar por soluciones más rápidas y de menor capacidad de cómputo. **TensorRT** es la biblioteca utilizada, desarrollada por Nvidia, orienta a IA y especializada en la Nvidia Jetson Nano. Tras aplicar la optimización usando parámetros como la precisión de bits entre otros, la Tabla 13 muestra los tiempos de inferencia anterior y posterior a la optimización.

Tabla 13. Resultados Validación Modelo IA

| Precisión bits | sin TensorRT (s) | con TensorRT (s) | mIoU validación |
|----------------|------------------|------------------|-----------------|
| 8 bits | 0,120 s | 0,080 s | 25 % |
| 16 bits | 0,350 s | 0,180 s | 40 % |
| 32 bits | 0,679 s | 0,220 s | 49 % |

Estos resultados de tiempo, se han calculado usando la **GPU** de la Jetson Nano, que en estimación es 10 veces más rápido que usando la CPU.

A medida que se utilizan menos bits, el tiempo de inferencia es menor, sin embargo, el equilibrio entre precisión y tiempo es esencial. Al no ser un sistema tan crítico desde el punto de vista de respuesta rápida, como puede ser por ejemplo un vehículo autónomo, se aboga por la métrica de validación, por ello se ha optado por utilizar una **precisión de 32 bits**. Estos bits son los mismos utilizados en la parte de entrenamiento, por lo que la optimización en ese campo no ha cambiado, sin embargo TensorRT optimiza automáticamente la fusión de capas, la eliminación de redundancias o la reordenación de operaciones.

7. Presupuesto

Dentro de este capítulo se detalla la estimación de costes asociados al desarrollo del proyecto considerando tanto los materiales como los recursos operativos.

7.1. Costes de Equipamiento y Software

El valor de estos costes se ha calculado con una amortización proporcional al tiempo de uso dentro del proyecto. La fórmula que se aplica es:

$$C_{\text{proyecto}} = \frac{C_{\text{total}}}{\text{vida útil}} \times \text{meses de uso}$$

La Tabla 14 representa el coste de equipamiento desglosado. Hay que destacar que el software utilizado para la codificación ha sido Python, que al ser de código abierto (no se necesita licencia comercial) no supone ningún coste.

Tabla 14. Costes de Equipamiento y Software

| Concepto | Coste Total (€) | Vida Útil (meses) | Meses Usados | Coste (€) |
|--------------------|-----------------|-------------------|--------------|--------------|
| Ordenador portátil | 620 | 60 | 5 | 51,60 |
| Nvidia Jetson Nano | 267 | 48 | 4 | 22,25 |
| Total | | | | 73,85 |

7.2. Coste Desarrollo y Mano de Obra

El coste de personal se estima utilizando el coste de mano de obra, que incluye el coste laboral. La Tabla 15 representa el coste personal desglosado.

Tabla 15. Costes Laborales

| Categoría | Horas | €/hora | €/mes | Coste Total (€) |
|---------------|-------|--------|--------|------------------|
| Data Engineer | 351 | 37,61 | 3.300€ | 13.201,11 |
| Total | | | | 13.201,11 |

7.3. Costes Totales

Se expone a continuación los PM asociados a la vida del proyecto durante su realización. Se ha tomado como base que 1 PM equivale a 1700h anuales. Se trata de un parámetro que indica el esfuerzo de personal que se necesita. En resumen, es una forma de medir el esfuerzo necesario.

Tabla 16. PM

| Job Position | PMs | Monthly PM | Total PM |
|---------------|-----|------------|----------|
| Data Engineer | 1 | 0.0515 | 0.206 |

Con toda la información anterior, se procede a exponer el coste total del proyecto, desglosado en costes directos e indirectos. Recaltar que los costes indirectos se ha calculado siguiendo la aproximación siguiente:

$$C_{indirectos} = 0,15 \times (C_{equipamiento} + C_{mano_obra})$$

Tabla 17. Coste Total

| Coste Directo Personal/€ | Otros Costes Directos/€ | Coste directo por apoyo financiero a terceros/€ | Contribuciones | | Costes Indirectos (€) | Presupuesto Total (€) |
|--------------------------|-------------------------|---|--|--------------------------------------|-----------------------|-----------------------|
| | | | en especie no usadas en instalaciones del beneficiario/€ | Costes directos de subcontratación/€ | | |
| 13.201,11€ | 73,85€ | 0€ | 0€ | 0€ | 1.991,24€ | 15.266,20€ |

Es resumen, el coste total del proyecto es de **14.860,70 €**

8. Análisis de impacto

Se exponen a continuación los distintos impactos, tanto tecnológicos, económicos, sociales, ambientales y los relacionados con los ODS (*Objetivos de Desarrollo Sostenible*). Este análisis permite valorar el alcance real del trabajo y su proyección más allá del entorno.

8.1. Impacto tecnológico

El proyecto contribuye al avance de la tecnología mediante:

- Mejora y extensión de funcionalidades en soluciones existentes mediante la implementación optimizada de modelos de segmentación semántica para dispositivos embebidos.
- Desarrollo de un enfoque técnico con funcionalidades adicionales con un enfoque que puede aplicarse a casos similares.
- Compatibilidad con estándares reconocidos como IEEE 802.11n y protocolos de comunicación serie UART, facilitando la interoperabilidad con otros sistemas y dispositivos embebidos.
- Posibilidad de integración en productos y procesos industriales a corto y medio plazo, especialmente en sectores como la agricultura de precisión, vehículos autónomos, robótica de servicio y sistemas IoT
- Contribución a la mejora de la autonomía y eficiencia de sistemas embebidos, al permitir procesamiento local de datos críticos para la toma de decisiones en tiempo real.

8.2. Impacto económico

Desde el punto de vista de la viabilidad económica:

- Se ha realizado una evaluación preliminar de costes (ver capítulo 6), indicando que el sistema es implementable dentro de un presupuesto razonable.
- Reducción de costos operativos al disminuir la dependencia de procesamiento en la nube, reduciendo el ancho de banda y servidores remotos.
- Aceleración en los procesos de toma de decisiones gracias al procesamiento en el borde, lo que mejora la eficiencia productiva y reduce tiempos de inactividad.
- Generación de nuevas oportunidades de negocio en sectores como agricultura inteligente impulsando el desarrollo económico local, regional o global.

8.3. Impacto social

Desde el punto de vista de la importancia social:

- Mejora en la calidad de vida al facilitar soluciones tecnológicas accesibles para monitoreo y automatización, incrementando la seguridad y bienestar en entornos rurales o industriales.
- Generación de empleo especializado impulsando la creación de puestos de trabajo en áreas de ingeniería, desarrollo de software, inteligencia artificial y mantenimiento tecnológico, fomentando la capacitación profesional.
- Acceso a tecnología avanzada acercando las herramientas de IA y Edge Computing a comunidades y sectores tradicionalmente alejados de la innovación.
- Apoyo a la educación y formación haciendo posibles programas educativos basados en el proyecto que dote de facultades a las personas para su adaptabilidad a las nuevas demandas del mercado laboral.
- Se han considerado aspectos éticos como la privacidad de los datos, el consentimiento del usuario, la transparencia de los algoritmos o el uso responsable de la inteligencia artificial.

8.4. Impacto ambiental

Desde el punto de vista del impacto ambiental:

- Reducción del coste energético al implementar modelos de Edge AI eficientes, disminuyendo el consumo eléctrico en comparación con soluciones en la nube, favoreciendo un uso más sostenible de la energía.
- Optimización en el uso de recursos como el agua, herbicida a través de la monitorización, reduciendo el impacto ambiental.
- Minimización de emisiones contaminantes al utilizar soluciones automatizadas, pudiendo disminuir la necesidad de maquinaria pesada y un uso indiscriminado, ayudando a reducir emisiones de gases contaminantes.
- Fomentación de soluciones sostenibles al integrar sistemas inteligentes fomentando la adopción de métodos más ecológicos y responsables en la industria o agricultura.

8.5. Impacto en Objetivos de Desarrollo Sostenible

Los ODS constituyen una referencia internacional para el desarrollo de sostenible. El proyecto contribuye a los siguientes objetivos:

- **ODS 2 - Hambre Cero:** se mejora la eficiencia agrícola y se reduce pérdidas en los cultivos.
- **ODS 9 Industria, Innovación e Infraestructura:** al fomentar soluciones tecnológicas avanzadas como Edge AI, promoviendo una industria más inteligente y sostenible.
- **ODS 12 - Producción y Consumo Responsable:** al mejorar la precisión en el uso de recursos como herbicidas, favoreciendo un uso responsable de los recursos.
- **ODS 13 - Acción por el Clima:** al reducir emisiones mediante la automatización eficiente y evitar un posible sobreuso de maquinaria pesada.
- **ODS 15 - Vida de Ecosistemas Terrestres:** al mejorar el uso de agroquímicos, se protege la biodiversidad.

9. Conclusiones y trabajos futuros

9.1. Conclusiones

El proyecto de detección de maleza en cultivos de zanahoria, con la automatización de herbicida aborda una de las problemáticas en el campo de la agricultura de precisión como es la optimización de recursos y la mejora de la calidad el cultivo con su consecuente reducción de pérdidas de alimentos no aprovechados.

La solución planteada, basada en una arquitectura en el **Edge Computing**, utilizando **inteligencia artificial** y **visión artificial**, permite distinguir entre cultivo y maleza con alta fiabilidad, reduciendo así el uso indiscriminado de herbicidas. Con esta automatización no solo se mejora la eficiencia operativa, si no que también se avanza hacia una agricultura más sostenible, responsable, de calidad y avanzada tecnológicamente. El dispositivo final de procesamiento utilizado es la **Nvidia Jetson Nano**, construyendo un modelo con las capacidades de procesamiento adaptadas al procesamiento local.

El sistema emplea tecnologías relacionadas con el estado del arte, adaptadas específicamente a los desafíos articulares del entorno agrícola, más concretamente de los cultivos de zanahoria. Las soluciones propuestas han sido diseñadas para mantener la escalabilidad, fácil integración con dispositivos hardware existentes y manteniendo el equilibrio entre respuesta en tiempo real y precisión del sistema.

A lo largo del trabajo se han evaluado distintos algoritmos de segmentación semántica, destacando **U-Net** con una red base **MobileNetV2** adoptada como la solución final, debido a su equilibrio en términos de precisión y generalización, basándose en la métrica *mIoU*, con valores de **0.69** y **0.50** en entrenamiento y validación respectivamente, suponiendo una segmentación de alta calidad en proyectos de agricultura. También se ha logrado optimizar el modelo, utilizando **TensorRT**, manteniendo constante la predicción y a la reducción de tiempos de inferencia, tocando parámetros como la precisión de bits del modelo. Adicionalmente, se comprueba que el uso de TensorRT permite reducir drásticamente los tiempos de inferencia, incluso manteniendo la precisión en **32 bits**, convirtiéndose en una adaptación clave para su implementación en dispositivos de recursos limitados como la Jetson Nano.

Por último, se ha diseñado una conexión de almacenamiento con una base de datos interna en el dispositivo local de procesamiento, que guarda las estadísticas relacionadas con el porcentaje de parte de maleza y parte de la hoja de zanahoria como información primordial, visualizando de manera gráfica la evolución de la calidad del cultivo a medida que se dispersa herbicida.

9.2. Trabajos futuros

Los pasos futuros del estudio incluyen la integración del sistema con cámaras reales para la captura de imágenes en campo, ya que en esta etapa se trabajó exclusivamente con un conjunto de datos previamente recolectado. Además, se plantea la conexión directa con el sistema automatizado de aplicación de herbicida, permitiendo una intervención en tiempo real y precisa.

Por otro lado, se contempla la integración con herramientas de visualización como Tableau, Power BI o Grafana, con el fin de desarrollar una aplicación personalizada que permita al agricultor o agricultora realizar una monitorización continua del estado de su cultivo desde un dispositivo móvil. Esto abriría la puerta a una solución completa, inteligente y accesible que combina visión artificial, automatización agrícola y análisis de datos, orientada a una agricultura más eficiente y sostenible.

Este estudio puede servir de base para la generalización a otros tipos de plantas o cultivos, no solo al de zanahorias con respectivas adaptación como dispersión automática de otros recursos como pesticidas o agua, lo que acomplia significativamente su aplicabilidad y potencial impacto en distintos sistemas agrícolas.

10. Referencias

- [1] E. G. Álvarez, «¿Qué problemas resolverá la agricultura de precisión?», ene. 2022. [En línea]. Disponible en: <https://farmable.tech/es/que-problemas-resolvera-la-agricultura-de-precision/>. [Accedido: 14 nov. 2025].
- [2] D. Situnayake y J. Plunkett, *AI at the Edge*. Sebastopol, CA: O'Reilly Media, 2023. [E-book]. [En línea]. Disponible en: O'Reilly Online Learning.
- [3] Agroinformación, «La agricultura de precisión ahorra un 30% en costes de producción al agricultor», nov. 2016. [En línea]. Disponible en: <https://agroinformacion.com/la-agricultura-precision-ahorra-30-costes-produccion-al-agricultor/>. [Accedido: 16 feb. 2025].
- [4] Diario Siglo XXI, «La agricultura de precisión hará el campo 'más rentable y sostenible'», nov. 2023. [En línea]. Disponible en: <https://www.diariosigloxxi.com/texto-s/mostrar/524642/agricultura-precision-hara-campo-rentable-sostenible>. [Accedido: 15 feb. 2025].
- [5] Raspberry Pi Foundation, *Raspberry Pi 3 Model B+*, 2025. [En línea]. Disponible en: <https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/>. [Accedido: 19 jun. 2025].
- [6] Waveshare, *JETSON-NANO-DEV-KIT*, 2025. [En línea]. Disponible en: <https://www.waveshare.com/wiki/JETSON-NANO-DEV-KIT>. [Accedido: 19 jun. 2025].
- [7] *JETSON-NANO-DEV-KIT*, 2025. [En línea]. Disponible en: <https://www.waveshare.com/wiki/JETSON-NANO-DEV-KIT>. [Accedido: 19 jun. 2025].
- [8] D. Pérez Aguilar, R. Risco y L. Casaverde-Pacherrez, «Transfer learning en la clasificación binaria de imágenes térmicas», *Ingenius*, pp. 71-86, jun. 2021. DOI: 10.17163/ings.n26.2021.07.
- [9] S. Á. Fernández, D. P. López y A. Ribeiro, «D4.1 Pilot 1 objectives, requirements and design», *Cordis - EU*, dic. 2023. [En línea]. Disponible en: <https://cordis.europa.eu/project/id/101017111/results>. [Accedido: 25 mar. 2025].

- [10] Y.-Y. Zheng, J.-L. Kong, X.-B. Jin et al., «CropDeep: The Crop Vision Dataset for Deep-Learning-Based Classification and Detection in Precision Agriculture», MPID, mar. 2019. DOI: 10.3390/s19051058. [Accedido: 28 mar. 2025].
- [11] T. Aboneh, A. Rorissa, R. Srinivasagan y A. Gemechu, «Computer Vision Framework for Wheat Disease Identification and Classification Using Jetson GPU Infrastructure», MPID, vol. 9, no. 3, p. 18, jul. 2021. DOI: 10.3390/technologies9030047. [En línea]. Disponible en: <https://doi.org/10.3390/technologies9030047>. [Accedido: 28 mar. 2025].
- [12] E. Assunção, P. D. Gaspar, R. Mesquita et al., «Real-Time Weed Control Application Using a Jetson Nano Edge Device and a Spray Mechanism», MIP, 2022. [Accedido: 3 abr. 2025].
- [13] NVIDIA, NVIDIA TensorRT Documentation, 2025. [En línea]. Disponible en: <https://docs.nvidia.com/deeplearning/tensorrt/latest/index.html>. [Accedido: 18 abr. 2025].
- [14] TensorFlow, TensorFlow Lite Guide, 2025. [En línea]. Disponible en: <https://www.tensorflow.org/lite/guide?hl=es-419>. [Accedido: 18 abr. 2025].
- [15] PyTorch, TorchScript — PyTorch Documentation, 2025. [En línea]. Disponible en: <https://docs.pytorch.org/docs/stable/jit.html>. [Accedido: 18 abr. 2025].
- [16] DataScientest, U-Net: Lo que tienes que saber, 2025. [En línea]. Disponible en: <https://datascientest.com/es/u-net-lo-que-tienes-que-saber>. [Accedido: 15 abr. 2025].
- [17] R. Singh y S. S. Gill, «Edge-AI: A survey», KeAi Chinese Roots Global Impact, vol. 3, pp. 71-92, 2003.
- [18] Raspberry Pi Foundation, Raspberry Pi Documentation, 2025. [En línea]. Disponible en: <https://www.raspberrypi.com/documentation/>. [Accedido: 17 jun. 2025].
- [19] Organización de las Naciones Unidas, Objetivos de Desarrollo Sostenible (ODS), 2025. [En línea]. Disponible en: <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>. [Accedido: 16 jun. 2025].
- [20] M. V. P. Núñez y J. Achachi, «Impacto de la Inteligencia Artificial en las Actividades Humanas: Un Análisis de las Consecuencias», Revista Científica Conectividad, vol. 6, no. 1, ene. 2025. [En línea]. Disponible en: <http://revista.ister.edu.ec/ojs/index.php/ISTER/article/view/238/277>. [Accedido: 15 feb. 2025].

A. Anexo CWFID Dataset

Este anexo explica detalladamente el conjunto de datos utilizado para el algoritmo de Inteligencia Artificial, tanto para su entrenamiento como para la validación.

El conjunto de datos **CWFID** (*Crop Weed Field Image Dataset*) representa un conjunto de imágenes asociadas a cultivos como puede ser remolacha, maíz, etc. En este caso, el conjunto de datos obtenido en el enlace únicamente contiene imágenes de zanahorias. El conjunto de características que aborda este conjunto de datos son:

- **Propósito:** segmentación de imágenes para distinguir entre plantas de cultivo (crop) y maleza (weed).
- **Contenido:** imágenes reales tomadas en campo.
- **Etiquetas:** anotaciones a nivel de píxel o segmento.

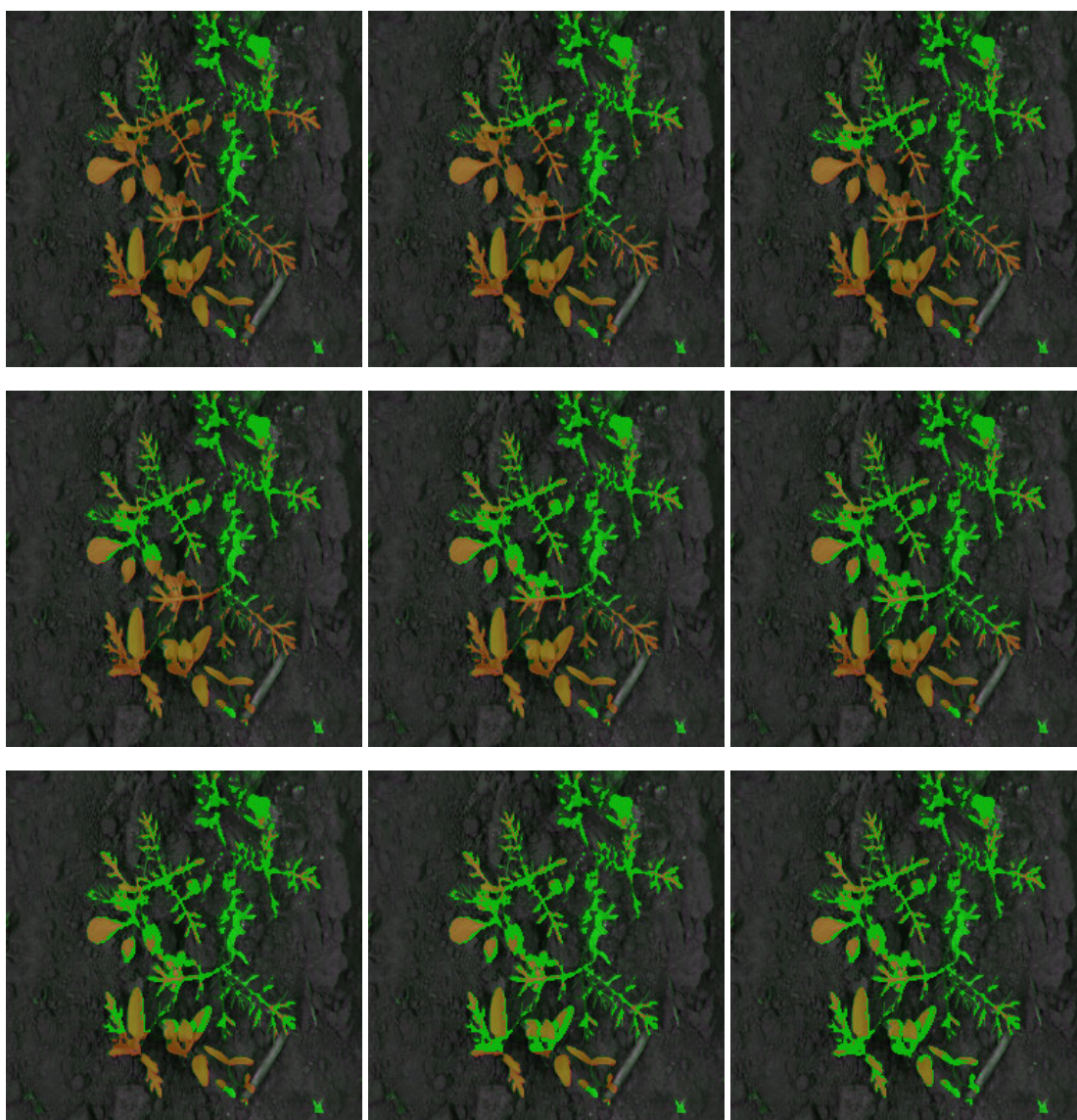
Más concretamente, el conjunto de imágenes de zanahorias utilizadas son 60, etiquetadas desde el número 1 al 60. El conjunto de imágenes se divide en dos partes:

- **Imágenes Entrada:** representan las imágenes reales capturadas por una cámara en terrenos de campo reales. Son imágenes en RGB, es decir, a color.
- **Imágenes Etiquetadas:** se representan como máscaras etiquetadas, es decir, parte de la imagen de entrada y la convierten a una escala de únicamente tres colores. Destacar que estas imágenes han sido etiquetadas por expertos asociados al campo de la agricultura de precisión.
 - **Negro:** representa el fondo de la imagen (parte de la imagen que no corresponde a la planta).
 - **Rojo:** píxeles asociados a la maleza.
 - **Verde:** píxeles asociados a la parte de la planta sana.

B. Monitorización Cultivo

Se expone a continuación el conjunto de imágenes que se han simulado a partir de las cuales se han extraído las estadísticas para la "Monitorización del cultivo" en el apartado de "Implementación".

Figura 18. Imágenes Monitorización



La evolución de las imágenes muestran que la aplicación de herbicida es exitosa, ya que cada vez el porcentaje de rojo es menor. Lo ideal es que finalmente el porcentaje de rojo sea nulo, lo que supone que ya no hay maleza en el cultivo.

C. Anexo Scripts y Ejecución

Este anexo explica el conjunto de scripts utilizado tanto para modelar el sistema como para realizar el entrenamiento y el sistema final ejecutable que se localiza en el dispositivo de procesamiento del borde. Este es el repositorio donde se localiza el código en [Github](#). La organización del repositorio se basa en dos carpetas:

- **System:** se trata del conjunto de scripts necesarios para la ejecución del sistema. Estos scripts se encuentran dentro de la **Jetson Nano** y son:
 - **UNet_TR.py:** Extrae la imagen, la almacena en Jetson Nano, realiza la predicción de IA y calcula el porcentaje de plantas sanas y de maleza. Consiste en el modelo ya optimizado con TensorRT.
 - **Load_DB.py:** carga en la base de datos MySQL las estadísticas y rutas de la imagen original y predicha una vez que finaliza el algoritmo de IA.
 - **Main.py:** controlador de todo el sistema responsable de ejecutar cada parte sistemáticamente.
 - **run_unet.sh:** ejecuta main.py usando argumentos de la CLI para su ejecución automática. Se requiere una carga de framework específico para el correcto funcionamiento del modelo de IA.
- **Train:** corresponde a la parte de los algoritmos utilizados para el entrenamiento del algoritmo.
 - **AI_algorithm.py:** algoritmo utilizado para realizar el entrenamiento con diferentes metodologías como búsqueda de hiperparámetros y entrenamiento del codificador y decodificador.
 - **Trt_model.py:** código que realiza la optimización del modelo común al optimizado en TensorRT.
 - **extract.py:** módulo que realiza la relación entre el nombre de imagen de entrada y la ruta donde se encuentra. Recordemos que el algoritmo se simula con un dataset previo.
- **Nvidia_Jetson_Nano.pdf:** manual que explica en detalle los componentes y puesta en marcha de la Jetson Nano.

Github disponible en "https://github.com/albertosanchez33/TFG_CD"

Los archivos se encuentran distribuidos dentro de la Jetson Nano en este directorio.

```
CWFID/  
dataset/  
model/  
saved_model/  
Trt_model.py  
Unet_TRT.py  
run_unetRT.sh
```

Para la ejecución de una simulación de como funcionaría el sistema se debe ejecutar dentro de la **Jetson Nano** el siguiente comando en el CLI, es decir, en la línea de comando Linux asociada.

```
.\run_unetRT.py ruta_imagen_original NUM
```

Donde **ruta_imagen_original** representa la ruta donde se encuentra almacenada la imagen original captura y **NUM** representa el número asociado a la imagen correspondiente a una parcela (seguimiento del número de veces que se ha dispensado herbicida), simulando el funcionamiento real del sistema.