
**Diseño de un vuelo fotogramétrico con un vehículo
aéreo no tripulado (UAV) para Agricultura de
Precisión en terrenos inclinados**



Andrés Alejandro Moreno Sánchez

**Universidad Escuela Colombiana de Ingeniería Julio Garavito
Maestría en Ingeniería Electrónica
Bogotá, Colombia
2024**

**Diseño de un vuelo fotogramétrico con un vehículo aéreo
no tripulado (UAV) para Agricultura de Precisión en terrenos
inclinados**

Andrés Alejandro Moreno Sánchez

**Trabajo de grado para optar al título de Magíster en Ingeniería
Electrónica con énfasis en Automatización y Control Industrial**

Director del trabajo de grado

Ing. Alexander Pérez Ruiz MSc. PhD.

Universidad Escuela Colombiana de Ingeniería Julio Garavito

Maestría en Ingeniería Electrónica

Bogotá, Colombia

2024

Jurado:

Jurado:

Nota de Aceptación:

El trabajo de grado titulado “Diseño de un vuelo fotogramétrico con UAV para Agricultura de Precisión en terrenos Irregulares” presentada por Andrés Alejandro Moreno Sánchez, cumple con los requisitos establecidos para optar al título de Magíster en Ingeniería Electrónica con énfasis en Automatización Industrial.

Jurado

Jurado

Director

Bogotá D.C., mayo del 2024

Agradecimientos

Quiero dedicar unas palabras de gratitud a las personas que han estado a mi lado durante la realización de esta maestría, ofreciéndome su apoyo y amor incondicional. Mi madre y mi hermana merecen una mención especial. Ellas han sido no solo mi refugio, sino también una fuente constante de motivación y aliento. Las largas horas de estudio y trabajo fueron siempre acompañadas por su fe inquebrantable en mi potencial y su disposición inagotable para escucharme y animarme a seguir adelante. Este documento es un reflejo de su amor y sacrificio, y espero que se sientan tan orgullosas de él como yo lo estoy de ellas.

También quiero agradecer profundamente a mi compañero de maestría, Daniel Palomino. Su colaboración, al prestarme la tarjeta Pixhawk 2, fue crucial para realizar pruebas de hardware-in-the-loop que enriquecieron significativamente mi proyecto. Daniel, tu generosidad y espíritu de equipo han dejado una huella imborrable en mi experiencia académica y personal.

Este viaje no habría sido posible sin el apoyo emocional y logístico de todas las personas que creyeron en mí y en mi proyecto. A cada uno de ustedes, les estoy eternamente agradecido. Juntos, hemos logrado no solo cumplir un requisito académico, sino también hacer una contribución valiosa a nuestra área de estudio.

Finalmente, quiero dedicar esta tesis a todos aquellos que buscan el conocimiento con pasión y perseverancia. Espero que este trabajo sirva como un pequeño ladrillo en la construcción de un futuro más innovador y colaborativo.

Resumen

En la actualidad, la agricultura de precisión emplea diferentes tecnologías para aprovechar y gestionar los recursos en diferentes tipos de cultivos, consiguiendo así un alto grado de automatización y optimización en los cultivos. Una de las herramientas usadas en esta industria son los vuelos **fotogramétricos** que asumen, que el terreno es horizontal y cuando se utiliza en terrenos inclinados se genera un problema en el traslape entre las imágenes. En estos casos se debe utilizar información topográfica detallada de la elevación del terreno previa para poder realizar una ruta de vuelo. Considerando que Colombia posee un relieve montañoso, se hace necesario adaptar una metodología que automatice el proceso de planeamiento de la ruta de vuelo en terrenos inclinados sin datos topográficos previos. Este proyecto diseñó un algoritmo que realiza la **planeación de rutas** de un **UAV** (Unmanned Aerial Vehicle) para vuelos fotogramétrico en terrenos montañosos de una extensión de 20 hectáreas utilizando como solución el cambio de altura de vuelo del dron manteniendo el **GSD** constante. En la realización de este trabajo se propuso un nuevo método que permite la clasificación del tipo de terreno midiendo el porcentaje de fondo, también se propuso un método para la medida de la distancia del dron con respecto al suelo y la medición de la inclinación de terreno, por último, se propuso un algoritmo que distingue entre praderas, laderas y valles para crear una ruta de vuelo que permite seguir las curvas de nivel en el terreno. Este proceso se implementó utilizando la computadora **Nvidia Jetson TX2** como computadora de compañía. Este computador SBC utiliza el sistema operativo Linux y el *framework* **ROS** (Robot Operating System) como estándar para que sea escalable a otros proyectos de robótica. Por último, se utilizó un controlador de vuelo **Pixhawk 2** en modo de ejecución **hardware-in-the-loop** (Palomino, 2020) para validar todo el proceso. En el análisis de resultados se encontró que el algoritmo propuesto permitió realizar una ruta de vuelo con una desviación en el GSD menor al 20% manteniendo los parámetros de traslape y **GSD** (Ground Sample Distance) constantes durante el vuelo.

Abstract

Currently, precision agriculture utilizes various technologies to harness and manage resources in different crop types, achieving a high degree of automation and optimization in farming. One of the tools used in this industry is **photogrammetric** flights, which assume that the terrain is horizontal. However, when used on sloped terrains, an issue arises in the overlap between images. In such cases, detailed topographic information on the terrain's elevation must be used beforehand to plan a flight route. Given that Colombia has mountainous terrain, it is necessary to adapt a methodology that automates the flight route planning process on inclined terrains without prior topographic data. This project designed an algorithm that **plans the routes** of a **UAV** (Unmanned Aerial Vehicle) for photogrammetric flights over 20 hectares of mountainous terrain, using the solution of changing the drone's flight altitude to maintain a constant **GSD**. In this work, a new method was proposed that allows for the classification of terrain type by measuring the percentage of background. Additionally, a method was proposed for measuring the distance of the drone from the ground and the measurement of terrain inclination. Finally, an algorithm was proposed that distinguishes between meadows, slopes, and valleys to create a flight route that follows the terrain's contour lines. This process was implemented using the **Nvidia Jetson TX2** as a companion computer. This SBC computer uses the Linux operating system and the **ROS** (Robot Operating System) framework as a standard to be scalable to other robotics projects. Lastly, a **Pixhawk 2** flight controller was used in **hardware-in-the-loop** mode (Palomino, 2020) to validate the entire process. In the results analysis, it was found that the proposed algorithm allowed for a flight route with a deviation in the GSD of less than 20%, maintaining constant overlap and **GSD** (Ground Sample Distance) parameters during the flight.

Índice General

Agradecimientos.....	4
Resumen	5
Abstract	6
Índice General.....	7
Índice de Tablas	9
Índice de Figuras.....	10
1. Introducción.....	15
1.1. Pregunta problema	19
2. Objetivos	20
2.1. Objetivos General.....	20
2.2. Objetivos Específicos	20
3. Marco Teórico	21
3.1. Agricultura de precisión y Fotogrametría	21
3.2. Fotogrametría en terrenos irregulares.....	22
3.3. Sistemas de referencia.....	24
3.4. Algoritmos de planificación de trayectoria.....	27
3.5. Características de un vuelo fotogramétrico.....	28
3.6. Sistema Operativo Robótico (ROS).....	30
3.7. Redes neuronales artificiales y <i>deep learning</i>	31
3.8. Modelo de cámaras estéreo.....	36
3.9. Nubes de puntos.....	38
3.10. Regresión lineal múltiple	39
3.11. Propagación de errores.....	39
4. Metodología	41
4.1. Medición de características del terreno circundante.....	42

4.2.	Clasificación del terreno.	44
4.3.	Medición de la inclinación y la altura.....	47
4.4.	Planeación de ruta de vuelo.....	51
5.	Resultados y Análisis.....	59
5.1.	Resultado en la medición de características del terreno.....	59
5.2.	Entrenamiento y selección de red neuronal para la clasificación del terreno.....	61
5.3.	Precisión de la medición de altura del terreno.....	65
5.4.	Análisis de los resultados de la planeación de rutas de vuelo.....	66
5.5.	Resultados de Integración en controladores Pixhawk 2 y Nvidia Jetson TX2.....	72
6.	Conclusiones y Trabajos Futuros.....	74
6.1.	Conclusiones.....	74
6.2.	Trabajos futuros.....	75
	Referencias Bibliográficas.....	77
	Anexo 1. Instalación del entorno de desarrollo.....	82
	Anexo 2. Selección del vehículo y adaptaciones realizadas.....	86
	Anexo 3. Descripción y modelado de los terrenos utilizados para las pruebas.....	88
	Anexo 4. Evaluación de métodos de clasificación tradicionales.....	104

Índice de Tablas

Tabla 1. Resultados del entrenamiento de las arquitecturas evaluadas.	62
Tabla 2. Parámetros de selección para los 15 modelos de clasificación.	63
Tabla 3. Métricas de validación modelo ANN2_64.	65
Tabla 4. Exactitud y precisión de medidas con cámara estéreo simulada.	66
Tabla 5. Análisis de regresión del GSD efectuado en un vuelo a altura constante.	70
Tabla 6. Análisis de regresión del GSD efectuado con la planeación de ruta propuesta.	71
Tabla 7. Comparación del RMSPE del GSD entre línea base y algoritmo propuesto para la planeación de rutas de vuelo.	72
Tabla 8. Tiempos de ejecución de algoritmo de planeación de rutas de vuelo en hardware in the loop.	72
Tabla 9. Tiempos de ejecución promedios por tipo de terreno en hardware in the loop.	73

Índice de Figuras

Figura 1. Disminución del traslape debido al cambio de la altura del terreno. Fuente: Pérez (2001).	22
Figura 2. Vuelo fotogramétrico basado en tolerancia del terreno. Fuente: Pérez (2001).....	23
Figura 3. Vuelo fotogramétrico con variaciones de altura. Fuente: Pérez (2001).	24
Figura 4. Frames: Geodésico, ECEF y Local (ENU). Fuente: Mike1024 (2010).	26
Figura 5. Frames: Local, Body y Camera. Fuente Palomino (2020).	27
Figura 6. Huella de la cámara. Fuente: Riaño (2018).....	28
Figura 7. Separación entre líneas de vuelo. Fuente: Riaño (2018).	29
Figura 8. Base en el aire. Fuente: Riaño (2018).	29
Figura 9. Límites de vuelo y distancia de giro.....	30
Figura 10. Modelo de Cámara estéreo Fuente (Karunanayake, 2016).	36
Figura 11. Estructura del algoritmo general para el vuelo fotogramétrico en terrenos inclinados.	42
Figura 12. Algoritmo de medición de características del terreno.....	43
Figura 13. Curvas paramétricas resultantes de la medición de una pradera y una ladera.	44
Figura 14. Curva paramétrica resultantes de la medición de un valle.....	44
Figura 15. Arquitectura de clasificación ANN.	46
Figura 16. Arquitectura de clasificación ANN2.	46
Figura 17. Arquitectura de clasificación 1D-CNN.	47
Figura 18. Algoritmo para medición de altura e inclinación del terreno.....	49

Figura 19. Cálculo para el punto de medición de altura e inclinación del terreno.	49
Figura 20. Tablero de ajedrez de 22x14 metros usado en el proceso de calibración.	50
Figura 21. Angulo de entrada de 30° en vuelo fotogramétrico.	52
Figura 22. Cálculo de ángulo de entrada en laderas.	53
Figura 23. Cálculo de ángulo de entrada en valles.	53
Figura 24. Punto de entrada en trayectoria de vuelo.	54
Figura 25. Primera recta definida para ruta de vuelo.	55
Figura 26. Recta trasladada y primer punto de ruta de vuelo.	55
Figura 27. Puntos en ruta de vuelo sobre recta con separación entre ellos de BA.	56
Figura 28. Punto en nueva recta trasladada de ruta de vuelo.	56
Figura 29. Fin de planeación de ruta de vuelo.	57
Figura 30. Corrección de alturas en laderas.	58
Figura 31. Corrección de alturas en valle.	58
Figura 32. Imágenes tomadas durante el proceso de medición de características del terreno.	59
Figura 33. Imágenes de disparidad calculadas durante el proceso de medición de características del terreno.	60
Figura 34. Porcentaje de fondo representado en naranja sobre las imágenes de disparidad. ...	60
Figura 35, Vista frontal y vista lateral izquierda de curva paramétrica medida sobre el terreno en rviz.	61
Figura 36. Vista isométrica de curva paramétrica medida sobre el terreno en rviz.	61

Figura 37. Diagrama de cajas y bigotes del entrenamiento de los modelos de clasificación propuestos.....	62
Figura 38. Numero de parámetros Vs Tiempo de ejecución de modelos de clasificación.	64
Figura 39. Matriz de confusión modelo ANN2_64.	65
Figura 40. Nube de puntos recreada por cámara estéreo simulada.	66
Figura 41. Rutas de vuelo generadas para los 5 terrenos de pradera.	67
Figura 42. Rutas de vuelo generadas para los 5 terrenos de ladera.....	68
Figura 43. Rutas de vuelo generadas para los 5 terrenos de valle.	69
Figura 44. Entorno de simulación gazebo y rviz.	83
Figura 45. Entorno de simulación gazebo y rviz para terreno ladera 1.	84
Figura 46. Interfaz para modificación de parámetros de rqt reconfigure	85
Figura 47. UAV con carga útil de cámara gimbal usado en el proyecto.....	86
Figura 48. Interfaz en QGroundControl para calibrar controladores PID del UAV.	87
Figura 49. Software in the loop (SITL) vs Hardware in the loop (HITL). Fuente: PX4. (2024). ..	87
Figura 50. Curvas de nivel con resolución de 10 metros para el terreno ladera 1.	89
Figura 51. Vista de superior, frontal, lateral derecha e isométrica del terreno ladera 1.....	89
Figura 52. Curvas de nivel con resolución de 10 metros para el terreno ladera 2.	90
Figura 53. Vista superior, posterior, lateral derecha e isométrica del terreno ladera 2.	90
Figura 54. Curvas de nivel con resolución de 10 metros para el terreno ladera 3.	91
Figura 55. Vista superior, frontal, lateral derecha e isométrica del terreno ladera 3.	91

Figura 56. Curvas de nivel con resolución de 10 metros para el terreno ladera 4.	92
Figura 57. Vista superior, posterior, lateral derecha e isométrica del terreno ladera 4.	92
Figura 58. Curvas de nivel con resolución de 10 metros para el terreno ladera 5.	93
Figura 59. Vista superior, posterior, lateral izquierda e isométrica del terreno ladera 4.	93
Figura 60. Curvas de nivel con resolución de 10 metros para el terreno pradera 1.....	94
Figura 61. Vista superior e isométrica del terreno pradera 1.....	94
Figura 62. Curvas de nivel con resolución de 10 metros para el terreno pradera 2.....	95
Figura 63. Vista superior e isométrica del terreno pradera 2.....	95
Figura 64. Curvas de nivel con resolución de 10 metros para el terreno pradera 3.....	96
Figura 65. Vista superior e isométrica del terreno pradera 3.....	96
Figura 66. Curvas de nivel con resolución de 10 metros para el terreno pradera 4.....	97
Figura 67. Vista superior e isométrica del terreno pradera 4.....	97
Figura 68. Curvas de nivel con resolución de 10 metros para el terreno pradera 5.....	98
Figura 69. Vista superior e isométrica del terreno pradera 5.....	98
Figura 70. Curvas de nivel con resolución de 10 metros para el terreno valle 1.....	99
Figura 71. Vista superior e isométrica del terreno valle 1.....	99
Figura 72. Curvas de nivel con resolución de 10 metros para el terreno valle 2.....	100
Figura 73. Vista superior e isométrica del terreno valle 2.....	100
Figura 74. Curvas de nivel con resolución de 10 metros para el terreno valle 3.....	101
Figura 75. Vista superior e isométrica del terreno valle	101

Figura 76. Curvas de nivel con resolución de 10 metros para el terreno valle 4.....	102
Figura 77. Vista superior e isométrica del terreno valle 4.....	102
Figura 78. Curvas de nivel con resolución de 10 metros para el terreno valle 5.....	103
Figura 79. Vista superior e isométrica del terreno valle 5.....	103
Figura 80. Diagrama de cajas y bigotes del entrenamiento de los modelos de clasificación tradicionales.	105
Figura 81. Numero de parámetros Vs Tiempo de ejecución de modelos de clasificación clásicos.	106

1. Introducción

La agricultura de precisión ha revolucionado el manejo de los recursos agrícolas mediante la integración de tecnologías avanzadas, optimizando las prácticas de cultivo de manera significativa. Sin embargo, la aplicación de la fotogrametría de baja altura en terrenos irregulares sigue presentando desafíos importantes, especialmente en regiones montañosas de cultivo como se dan en Colombia, donde las variaciones del terreno complican la captura de datos consistentes y precisos. Este problema se agrava por los sistemas de fotogrametría diseñados predominantemente para terrenos planos, que no consideran las fluctuaciones de altitud que caracterizan a los paisajes montañosos, resultando en dificultades para mantener adecuadamente el traslape y el *Ground Sample Distance (GSD)* necesarios para obtener mapas detallados y precisos (Pérez, 2001).

En respuesta a estos retos, este proyecto propuso una metodología para la automatización de la planificación de la ruta de vuelo que se aparta de las soluciones convencionales que requieren de información de un *digital elevation model (DEM)*. Estas soluciones tradicionales han demostrado requerir un procesamiento detallado de la elevación del terreno previo a realizar la planificación del vuelo, como lo demuestra la investigación sobre la optimización de misiones de vuelo en terrenos empinados (Kozmus et al., 2020). En su lugar, esta investigación propuso una planificación basada en un vuelo de reconocimiento automático el cual en tiempo real determina parámetros de elevación e inclinación del terreno sin necesidad de un (DEM). Este vuelo inicial, se realiza para optimizar la ruta basada en un análisis detallado del terreno, permitiendo ajustes en los parámetros de vuelo de la misión.

Los algoritmos de clasificación con *Deep learning* analizan las imágenes y los datos recogidos, identificando características del terreno para la planificación de la ruta en tiempo real. Este enfoque se complementa con la implementación de un modelo de cámaras estéreo y tecnologías de nubes de puntos que permiten la recopilación de datos tridimensionales del terreno con una sola cámara,

proporcionando una base detallada para análisis más complejos y mejorando la eficacia de la fotogrametría en terrenos irregulares (Hsieh et al., 2023).

El proyecto utilizó el Sistema Operativo Robótico (ROS) para integrar y gestionar eficazmente los componentes tecnológicos en una plataforma cohesiva y modular, lo que permite integrar hardware y software y proporcionar herramientas robustas para la simulación y prueba de vuelos en un entorno simulado. La capacidad para adaptar la altitud del vuelo en respuesta a las variaciones del terreno, manteniendo un traslape y una GSD constantes, aborda de manera efectiva los problemas asociados con la variabilidad en estos parámetros en condiciones de terreno inclinado, una solución respaldada por la literatura reciente en diseño dinámico de redes 3D para fotogrametría en terrenos montañosos (Gargari et al., 2022).

La evaluación de la metodología propuesta se hizo mediante pruebas tanto en simulación como en dispositivos de vuelo reales para validar la efectividad de esta aproximación, comparando la precisión y estabilidad del GSD obtenidos con vuelos programados a una altura constante. Esta investigación permitió avanzar en la práctica de la fotogrametría para la agricultura de precisión en terrenos irregulares, también ofrece un enfoque más adaptable y eficiente que las metodologías tradicionales, abriendo nuevas posibilidades para la mejora continua en este campo.

El desafío de implementar vuelos fotogramétricos efectivos en terrenos montañosos no se limita únicamente a la recopilación de datos topográficos; también abarca la necesidad de una integración precisa y funcional de tecnologías que puedan procesar y analizar grandes volúmenes de datos en tiempo real. La utilización del modelo de cámaras estéreo y tecnologías de nubes de puntos es crucial, ya que proporcionan la capacidad de capturar información detallada del terreno en tres dimensiones, facilitando una planificación más precisa de la misión fotogramétrica (Li et al., 2021).

Esta metodología se apoya en la robustez del Sistema Operativo para Robots (ROS), que juega un papel fundamental en la gestión de los procesos y operaciones.

ROS permite la integración de múltiples sensores y sistemas de una manera modular, permitiendo una comunicación fluida entre los componentes del hardware y el software, ofreciendo una plataforma flexible y adaptable para el desarrollo y pruebas de nuevas funcionalidades. Esta capacidad es indispensable para realizar simulaciones detalladas de los vuelos, que predican el comportamiento del UAV bajo diferentes condiciones ambientales y topográficas. Estas condiciones descritas, aseguran que cada aspecto de la misión esté diseñado para enfrentar las variaciones del terreno (Singh et al., 2023).

Un aporte del proyecto es la planificación del vuelo sin un DEM, esto se logró realizando un vuelo de reconocimiento automatizado el cual determina la cantidad de inclinaciones del terreno y a partir de esto, un número de planos que representaran la forma del terreno. A diferencia de los sistemas tradicionales que requieren una intervención manual significativa para ajustar los parámetros de vuelo, esta metodología utiliza algoritmos para modificar automáticamente la altitud de vuelo en función de las características del terreno identificadas mediante los planos. Esto no solo mejora la eficacia del proceso fotogramétrico, sino que también aumenta la precisión de los datos recogidos, asegurando que el traslape y la GSD se mantengan dentro de los parámetros definidos a lo largo de toda la misión (Kozmus et al., 2020).

Además, la capacidad de realizar ajustes en la planificación de vuelo, basado en un análisis en tiempo real durante el vuelo de reconocimiento, es particularmente valiosa en entornos montañosos, donde las variaciones del terreno pueden ser extremadamente heterogéneas e impredecibles. Como menciona Gargari et al., (2022) en donde presenta un enfoque dinámico con el uso de un DEM, para la gestión de vuelos en terrenos inclinados. Este enfoque se alinea con las mejores prácticas en la modelación de terrenos y la gestión de datos espaciales, proporcionando un marco metodológico que puede adaptarse a otros proyectos de fotogrametría y robótica.

La adaptabilidad del vuelo fotogramétrico se convierte en un factor crítico para asegurar la recolección de datos fiables y de alta calidad lo cual se consigue al ajustar la altitud del UAV en respuesta a las variaciones topográficas detectadas.

Este enfoque de adaptación de vuelo, fundamentado en la literatura, refleja un avance significativo sobre los métodos de automatización convencionales. En investigaciones anteriores, Hsieh et. al. (2023) ha dependido del uso de un DEM para hacer los cambios de altura de una ruta de vuelo en un terreno de ladera. En contraste, la estrategia propuesta en este proyecto es usar un vuelo de reconocimiento automatizado en tiempo real.

Adicionalmente, el proyecto se apoya en los últimos avances en *deep learning* para la detección de características como imágenes de disparidad mediante una sola cámara. Estos algoritmos de inteligencia artificial no solo permiten la implementación con menos sensores como carga útil, sino que también reducen el tiempo necesario para el procesamiento de datos, permitiendo una respuesta en tiempo real al tipo de terreno y una mayor adaptabilidad en la planificación de vuelos futuros.

A medida que se profundiza en la metodología y tecnología detrás de la planificación de vuelos fotogramétricos, es esencial considerar también el aspecto del análisis de datos y cómo se integra en el proceso general. La capacidad para procesar y analizar eficientemente los datos recopilados durante los vuelos es tan crucial como la recogida de datos en sí. Utilizando técnicas de procesamiento de imágenes y algoritmos de *deep learning*, el proyecto no solo identifica y clasifica las características del terreno, sino que también mejora la forma en la que se realiza la planeación de las rutas de vuelo en terrenos inclinados.

El proyecto busco avanzar en el conocimiento de automatización para la fotogrametría en terrenos inclinados y exploro nuevas fronteras con la tecnología y metodologías probadas en este proyecto, futuras investigaciones podrían extenderse a terrenos con un mayor número de inclinaciones y a terrenos que

presenten topografías variadas en su extensión, demostrando la adaptabilidad y la escalabilidad de las soluciones desarrolladas. La continua evolución de las tecnologías de UAV y *deep learning* promete aún más mejoras, con el potencial de automatizar completamente el proceso de fotogrametría, reduciendo los costos operativos y aumentando la accesibilidad para usuarios en todo el mundo.

Este proyecto no solo abordó una necesidad latente en el campo de la agricultura de precisión, sino que también contribuyó a ampliar el conocimiento en el uso de la fotogrametría y la robótica. La investigación aquí presentada puede servir como base para futuros avances, haciendo que las metodologías y tecnologías propuestas puedan tener un impacto duradero en la forma en que se comprenden y gestionan los paisajes naturales y cultivados.

1.1. Pregunta problema

¿Cómo automatizar el proceso de planeación del vuelo fotogramétrico en terrenos inclinados, sin información previa, para que de forma autónoma un UAV ejecute la ruta y el cambio de alturas?

2. Objetivos

2.1. Objetivos General

Automatizar la planificación de la ruta de vuelo de un drone multirrotor para el desarrollo de una misión de vuelo fotogramétrico en condiciones de terreno inclinado con una extensión inferior a 20 hectáreas, utilizando como estrategia el cambio de altitud durante el vuelo para mantener constante el traslape y la distancia real sobre el terreno entre pixeles adyacentes (GSD).

2.2. Objetivos Específicos

- Establecer una metodología para la clasificación de los terrenos entre praderas, valle o laderas y determinar la dirección y grado de inclinación del terreno utilizando técnicas de reconocimiento de imágenes.
- Planificar el vuelo fotogramétrico que mantenga una GSD con un cambio menor al 20% utilizando la dirección y el grado de inclinación del terreno.
- Evaluar el funcionamiento del algoritmo utilizando el framework de desarrollo ROS y la metodología *hardware-in-the-loop* utilizando un controlador de vuelo Pixhawk y una Nvidia Jetson TX2 para validar la correcta toma de las fotografías en los modelos digitales.
- Comparar la precisión y estabilidad del GSD obtenidos con el vuelo automático y con un vuelo preprogramado a una altura constante.

3. Marco Teórico

3.1. Agricultura de precisión y Fotogrametría

La agricultura de precisión (AP) se define como *“una práctica de gestión que ha sido posible gracias al advenimiento de tecnologías de información adecuadas y proporciona un marco dentro del cual los administradores de cultivos comprenden y controlan con mayor precisión lo que sucede en sus granjas”* Blackmore, (1994). Este campo de trabajo ha ido desarrollando múltiples formas para poder censar y monitorizar grandes terrenos de cultivos, se han usado tecnologías como monitoreo satelital, teledetección con vehículos terrestres no tripulados (UGV) y tomas aéreas con vehículos aéreos no tripulados (UAV), entre muchas otras.

En esta investigación se centró en la fotogrametría que es la ciencia y la tecnología de obtener información confiable sobre los objetos físicos y el medio circundante, a través del proceso de grabación, medición e interpretación de imágenes fotográficas y patrones de energía electromagnética radiante y otros fenómenos. El desarrollo de esta técnica ha sido una pieza clave en múltiples campos, desde la cartografía hasta la agricultura de precisión, permitiendo el análisis detallado y la gestión eficaz de grandes extensiones de tierra.

Históricamente, la fotogrametría ha evolucionado significativamente desde sus primeras aplicaciones en la cartografía del siglo XIX, donde se utilizaban fotografías aéreas tomadas desde globos aerostáticos. En la actualidad, la tecnología ha avanzado al punto de integrar cámaras de alta resolución en drones, permitiendo capturas aéreas más precisas y a un menor costo. Este progreso tecnológico ha facilitado la rápida expansión de sus aplicaciones, no solo en la geografía y la cartografía, sino también en campos como la ingeniería civil, la arqueología y, por supuesto, la agricultura de precisión (Puce et al., 2023).

En el contexto de la agricultura de precisión, la fotogrametría juega un rol crucial al proporcionar datos detallados sobre el estado y la salud de los cultivos, la topografía del terreno y otras variables ambientales esenciales para una gestión

agrícola efectiva. Las imágenes procesadas ayudan a los agricultores a tomar decisiones informadas sobre la irrigación, la aplicación de fertilizantes y pesticidas, y la optimización general de los recursos.

Además, la fotogrametría se emplea ampliamente en la planificación urbana y rural, en la gestión de recursos naturales y en la monitorización del medio ambiente. La capacidad de esta técnica para generar modelos digitales de superficie (MDS) y modelos digitales de terreno (MDT) ha sido fundamental para el análisis espacial y la toma de decisiones basada en evidencia geográfica precisa. Sin embargo, el proceso es complejo y sujeto a varias limitaciones técnicas como objetos en movimiento, condiciones de poca luz, variaciones topográficas, u objetos parcialmente obstruidos, que pueden complicar el proceso de medición, proporcionar datos insuficientes o contradictorios que pueden complicar el proceso de medición (Nebel et al., 2020).

3.2. Fotogrametría en terrenos irregulares

El problema de la fotogrametría para agricultura de precisión en terrenos irregulares se encuentra en dos factores: el cambio del traslape que se produce con el cambio de altitud del terreno, y el cambio del GSD. El primero genera que al procesar las imágenes en un computador no se puedan encontrar puntos en común (Pérez, 2001). El segundo, que la resolución en los resultados varíe entre una sección y otra. El problema del traslape se puede observar en la Figura 1.

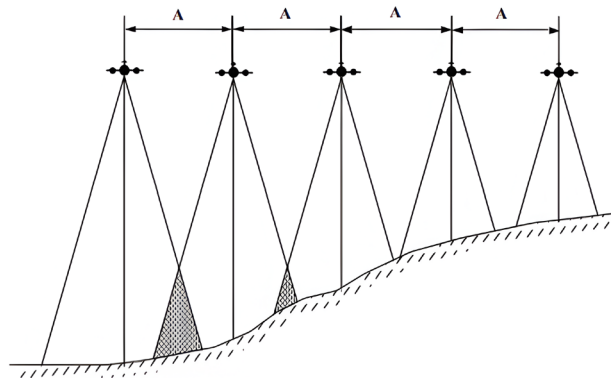


Figura 1. Disminución del traslape debido al cambio de la altura del terreno. Fuente: Pérez (2001).

Este problema afecta mayormente a países que realizan su agricultura en terrenos montañosos, como en Colombia. Debido a que la mayoría de la agricultura de precisión que se realiza es en terreno plano, como en las grandes planicies de EE.UU. y otros países desarrollados, no se ha tenido en cuenta incluir esta característica en los planificadores de vuelo existentes. Esto genera la necesidad de aplicar nuevos métodos y protocolos para poder realizar la misma labor en este territorio inclinado.

La solución que más se ha utilizado en topografía (Pachas, 2009) se da mediante tolerancias en los cambios de terreno, lo que aumenta el traslape original que se tenía previsto usar, como se ve en la Figura 2. Esto genera que el vuelo programado sea más largo, dado que se programa con unas ventanas muy pequeñas generando un sobre muestreo en la información y en el caso de tolerancias pequeñas, genera que se tenga que realizar otro vuelo ya que las imágenes no cuentan con el traslape necesario, además que no se mantiene el GSD constante durante el vuelo.

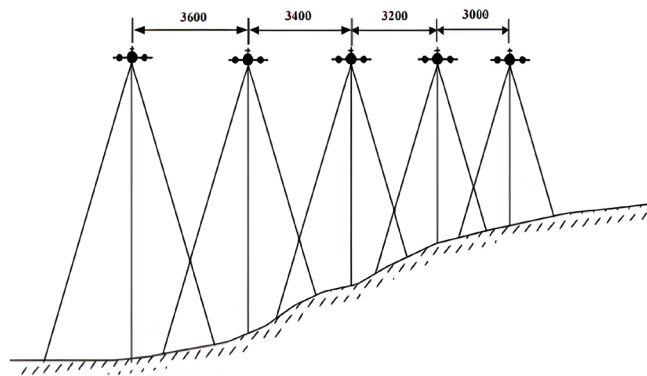


Figura 2. Vuelo fotogramétrico basado en tolerancia del terreno. Fuente: Pérez (2001).

En este contexto la teoría de la fotogrametría plantea otra solución, esta se basa en ir cambiando la altitud en la que se toman las fotos, con esto se logra solucionar tanto el problema del traslape como mantener una GSD más uniforme durante el vuelo, como se ve en la Figura 3. No obstante, no es un proceso que se encuentre automatizado (Palomino, 2020).

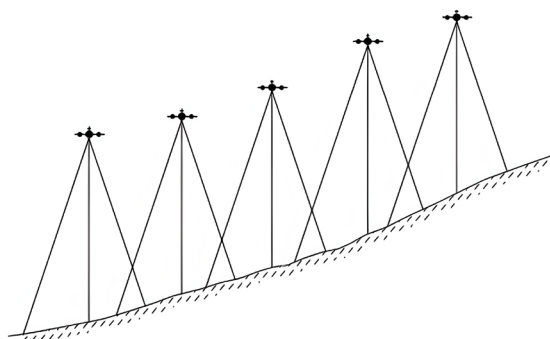


Figura 3. Vuelo fotogramétrico con variaciones de altura. Fuente: Pérez (2001).

Por esta razón resulto viable la generación de un método que permita mejorar el grado de automatización de este proceso, de forma que la calidad y el tiempo de toma de fotos pueda mejorarse.

3.3. Sistemas de referencia

En el ámbito del diseño, navegación y control de trayectorias de vehículos, se utilizan diversos sistemas de coordenadas, a menudo llamados *frames*, particularmente en aplicaciones con vehículos aéreos. Estos sistemas están diseñados para especificar la ubicación de un objeto dentro de un entorno determinado, como podría ser el planeta Tierra. Los vehículos UAV recurren a una variedad de sistemas de referencia, que son esencialmente agrupaciones de normas y conceptos teóricos creados para identificar la posición y orientación de un objeto en el espacio. Algunos de los sistemas de referencia más utilizados en vehículos UAV se enumeran a continuación:

3.3.1. Sistema Geodésico:

Este sistema considera la forma geoide (cercano a una esfera) y define la ubicación de un punto sobre la superficie de la tierra a partir de dos ángulos latitud (λ), longitud (φ) y la distancia radial de la altura (H). Basándose en un elipsoide de referencia, busca representar con precisión la superficie terrestre. La latitud se mide por el ángulo entre el meridiano de referencia y el punto de medición, mientras que

la longitud es el ángulo formado por el plano ecuatorial y la perpendicular al elipsoide en el punto de interés. La altitud se mide desde la superficie del elipsoide (Walter et al., 2018).

3.3.2. Sistema ECEF (Earth-Centered, Earth-Fixed):

Este sistema de coordenadas cartesianas, sincronizado con la rotación terrestre, tiene su origen en el centro de la Tierra. Representa un punto con coordenadas $(X_{ecef}, Y_{ecef}, Z_{ecef})$, con el eje z alineado con el eje de rotación terrestre y el eje x a lo largo del meridiano principal. La precisión en la determinación de estos ejes es fundamental para la efectividad del sistema (Walter & Blanch, 2017).

3.3.3. Sistema Local:

Este es un sistema de coordenadas cartesianas que se adapta a la geometría terrestre. Este sistema, cuyo plano XY es tangencial a la superficie de la Tierra en el punto de tangencia, se sincroniza tanto con el marco de mapeo como con el de navegación. En este contexto, un objeto se define por sus coordenadas tridimensionales $(X_{navf}, Y_{navf}, Z_{navf})$.

Existen dos enfoques principales para establecer los ejes de este sistema coordinado. Primero, la notación ENU (East, North, Up), donde el eje y apunta hacia el norte, el eje x hacia el este y el eje z se extiende perpendicularmente hacia arriba desde la superficie terrestre. Segundo, la notación NED (North, East, Down), preferida en aplicaciones con vehículos aéreos, sitúa el norte a lo largo del eje x , el este a lo largo del eje y , y el eje z se dirige hacia abajo, hacia el centro de la Tierra.

El punto de origen de este sistema de coordenadas se establece de forma arbitraria, y generalmente se selecciona el lugar de despegue o el punto donde se activan los sistemas de sensores. Esta elección del origen es crucial, ya que proporciona un punto de referencia constante para todas las operaciones de navegación y mapeo en la robótica (Walter et al., 2018).

En la Figura 4 se muestra el sistema geodésico, el sistema ECEF y el sistema local con notación ENU anteriormente descritos.

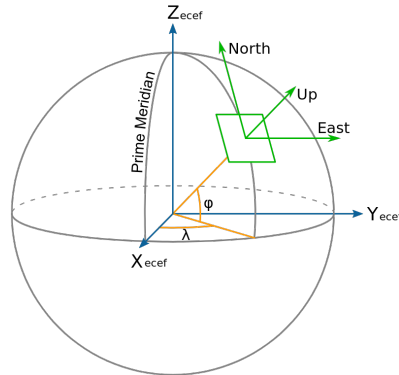


Figura 4. Frames: Geodésico, ECEF y Local (ENU). Fuente: Mike1024 (2010).

3.3.4. Sistema de referencia del cuerpo del vehículo (Body):

Centrado en el centro de gravedad del vehículo, este sistema cartesiano utiliza el centro de masa como origen. El eje x se alinea con la dirección de movimiento, y los otros ejes pueden variar según la aplicación, en vehículos aéreos el eje z se alinea con la rotación yaw, y el eje y con la rotación pitch (Walter & Blanch, 2017).

3.3.5. Sistema de referencia de la cámara:

Con origen en el punto principal de la cámara, este sistema organiza sus ejes de manera que el eje z sea perpendicular a la superficie del sensor, y el eje x apunte a la derecha de la cámara. La precisión en la definición de estos ejes es vital para la calidad de la imagen capturada y su posterior (Walter et al., 2018).

En la Figura 5 se muestra el sistema local con notación NED, el sistema de referencia del cuerpo del vehículo (*Body*) y el sistema de referencia cámara anteriormente descritos.

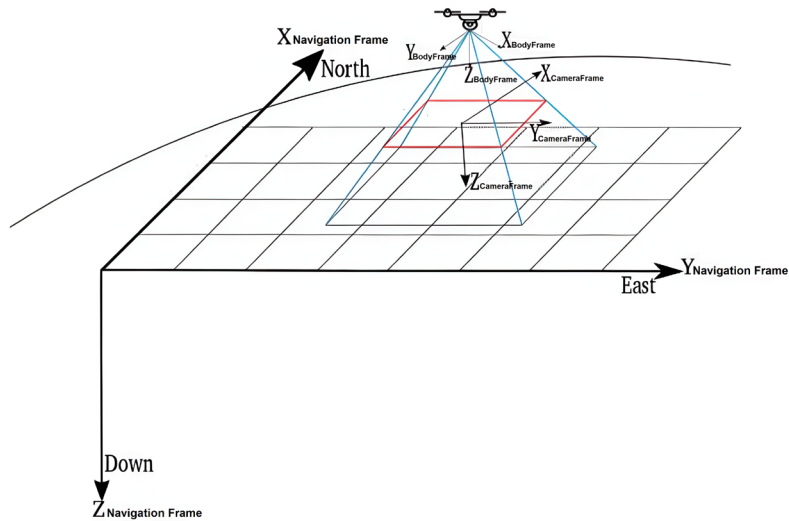


Figura 5. Frames: Local, Body y Camera. Fuente Palomino (2020).

3.4. Algoritmos de planificación de trayectoria

La planificación de trayectorias para vehículos aéreos no tripulados (UAV) enfrenta desafíos debido a la incertidumbre y al desconocimiento parcial o total del entorno, lo que requiere algoritmos capaces de funcionar en condiciones donde el conocimiento del entorno puede ser completo (problema estático) o incompleto (problema dinámico) (Mir et al., 2022). Los UAV requieren algoritmos que no solo los guíen a través del entorno, sino que también se adapten y respondan a los elementos cambiantes y las incertidumbres inherentes, como se discute en estudios recientes en la planificación de trayectorias (Jin et al., 2024; Li et al., 2024).

Los algoritmos de planificación de trayectorias se clasifican en dos categorías principales según su implementación y tiempo de ejecución: *Online* u *On-Board* y *Offline* u *Off-Board*. Los algoritmos *Online* u *On-Board* se ejecutan en tiempo real en el hardware del UAV, lo que le da la capacidad para responder dinámicamente a los cambios inmediatos en el entorno. Esta modalidad se usa para misiones en entornos no estructurados o donde la comunicación externa es limitada o inexistente, permitiendo al UAV mantener una autonomía completa y una capacidad de reacción instantánea (Palomino, 2020).

Por otro lado, la planificación *Offline* u *Off-Board* implica algoritmos que se ejecutan antes de la operación del UAV, dando la posibilidad de hacer el procesamiento de la ruta dentro o fuera del vehículo. Esta modalidad es adecuada para misiones que pueden ser planificadas de antemano y se llevan a cabo en entornos conocidos o menos dinámicos.

3.5. Características de un vuelo fotogramétrico

Para calcular los parámetros de vuelo, y la ruta a seguir en un vuelo fotogramétrico, se encontró en Riaño (2018) y en Palomino (2020) las siguientes ecuaciones para el cálculo de la GSD, ancho de huella, alto de huella, separación de líneas de vuelo y base en el aire. Estas ecuaciones hacen uso de las características de la cámara: ancho del sensor, distancia focal, ancho de imagen en píxeles, alto de imagen en píxeles. Y los parámetros del diseño del vuelo que son, altura de vuelo, el porcentaje de recubrimiento lateral y el porcentaje de recubrimiento longitudinal. En la Figura 6 se ve el ancho de huella y la altura de la huella. En la Figura 7 se ve la separación entre las líneas de vuelo. En la Figura 8 se ve la distancia entre las fotos en la dirección de la ruta de vuelo:

$$GSD(en\ cm/pix) = \frac{ancho\ del\ sensor\ (en\ mm) * H\ de\ vuelo * 100}{distancia\ focal * ancho\ imagen\ en\ pixeles} \quad (1)$$

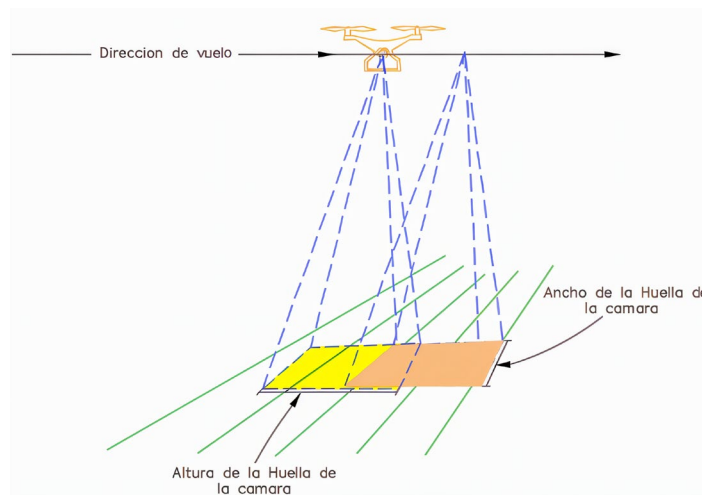


Figura 6. Huella de la cámara. Fuente: Riaño (2018).

$$\text{Ancho huella (en m)} = \frac{\text{GSD} * \text{Ancho imagen (en Píx)}}{100} \quad (2)$$

$$\text{Alto huella (en m)} = \frac{\text{GSD} * \text{Alto imagen (en Píx)}}{100} \quad (3)$$

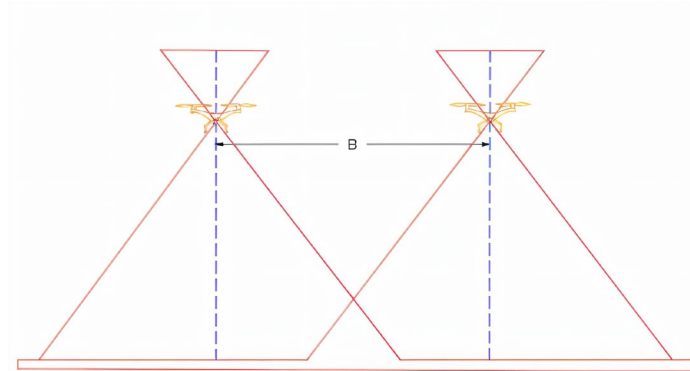


Figura 7. Separación entre líneas de vuelo. Fuente: Riaño (2018).

$$\text{Separación líneas de vuelo} = \text{Ancho huella (m)} * \left(1 - \frac{\text{Recubrimiento lateral en \%}}{100}\right) \quad (4)$$

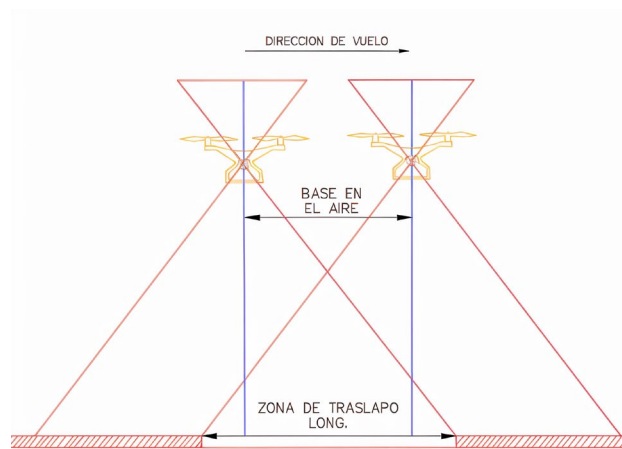


Figura 8. Base en el aire. Fuente: Riaño (2018).

$$\text{Base en el aire} = \text{alto de huella (m)} * \left(1 - \frac{\text{Recubrimiento longitudinal en \%}}{100}\right) \quad (5)$$

La definición de la zona de vuelo a fotografiar se puede hacer mediante cuadriláteros, circunferencias o polígonos, los cuales deben ser definidos por el usuario, mediante dos puntos que hacen de vértices opuestos para un cuadrilátero,

un punto y el radio para una circunferencia y un polígono mediante una serie de puntos que definan un circuito cerrado, En esta investigación se utilizó una definición de la zona de vuelo mediante un cuadrilátero.

Para que la zona de vuelo sea fotografiada en su totalidad y se tengan puntos de coincidencias en los bordes, se utilizo el parámetro de *Turnaround dist.* o distancia de giro, el cual es una medida en metros que agranda la cerca de la zona de vuelo definida, como se puede ver en Figura 9, en donde la cerca interna está en color rojo y la cerca externa en color azul.

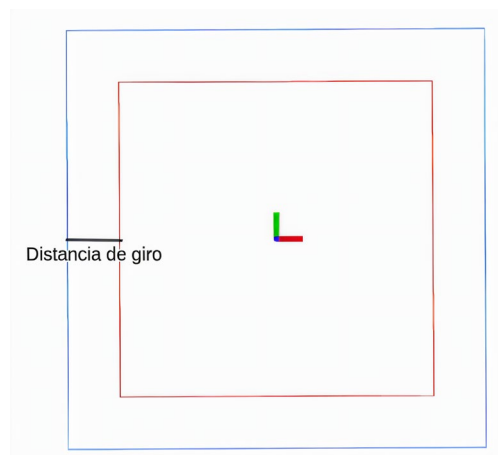


Figura 9. Límites de vuelo y distancia de giro.

3.6. Sistema Operativo Robótico (ROS).

El *framework* ROS (Robot Operating System) es el estándar de facto para la implementación de sistemas robóticos en el mundo. Se utilizo para desarrollar el sistema de comunicaciones entre el piloto automático, el computador de compañía y el simulador del entorno de pruebas. ROS es definido en Quigley et al (2009) como un sistema operativo *open source* para robots, el cual no es un sistema operativo tradicional de gestión y programación de procesos, sino que por el contrario proporciona una capa de comunicaciones estructurada por encima del sistema operativo Linux posibilitando un sistema de cómputo heterogéneo.

En ROS, un nodo es un proceso ejecutable que forma parte de la red del sistema operativo robótico. Los nodos son responsables de una amplia variedad de tareas, desde la adquisición de datos hasta su procesamiento o una actuación sobre el sistema físico. Estos procesos se diseñan para funcionar de forma independiente, pero pueden interactuar entre sí mediante diversas formas de comunicación para formar sistemas robóticos complejos.

Para la comunicación entre los diferentes nodos existen 3 formas de interacción: la primera son los mensajes, que se parecen a las comunicaciones UDP en donde varios nodos pueden transmitir bajo un nombre de mensaje y tener varios receptores que escuchan al tiempo. Los servicios, los cuales son una comunicación uno a uno, en el que un nodo cliente solicita un servicio y queda en espera a que el nodo servidor lo reciba, realice la operación, y responda a su petición. Y por último los *ActionLibs*, la cual es una comunicación uno a uno, en donde un nodo cliente solicita una acción a otro nodo servidor y continua la ejecución de sus procesos internos, mientras el nodo receptor procesa la petición, y envía mensajes de actualización del estado en la que se encuentra la tarea.

3.7. Redes neuronales artificiales y *deep learning*

Otras herramientas utilizadas en la investigación son los métodos de clasificación basados en redes neuronales artificiales y *deep learning*, que se usaron para clasificar el tipo de terreno entre laderas, valles y prados.

Las redes neuronales artificiales (ANN), inspiradas en el procesamiento de información del cerebro humano, se han consolidado como una herramienta útil en el campo del aprendizaje automático. Estas redes aprenden de grandes cantidades de datos, lo que les permite modelar sistemas complejos que son inaccesibles para otros tipos de algoritmos más tradicionales (Taye, 2023). El aprendizaje profundo o *deep learning*, que es una evolución directa de las ANN, emplea múltiples capas de procesamiento para extraer características progresivamente más abstractas de los datos de entrada. Esta técnica ha revolucionado la manera en que las máquinas

reconocen patrones y realizan clasificaciones complejas, alcanzando en algunos casos un desempeño que supera al humano, especialmente en tareas como el reconocimiento de imágenes y el procesamiento de lenguaje natural (Taye, 2023).

La eficacia del *deep learning* en la clasificación y procesamiento de imágenes es notable, gracias a la utilización de redes neuronales convolucionales (CNN). Estas redes son especialmente aptas para manejar datos visuales gracias a su estructura, que preserva las relaciones espaciales entre píxeles y permite una robusta extracción de características. Además, las CNN han sido fundamentales en el avance de técnicas que requieren la interpretación precisa de contenido visual, como la detección de objetos y la segmentación semántica (Srinivasu et al., 2021).

El rápido crecimiento del campo del *deep learning* ha sido impulsado por el desarrollo de algoritmos más eficientes y la disponibilidad de grandes conjuntos de datos y poder de cómputo. Este avance ha permitido explorar nuevas arquitecturas y modelos que pueden aprender de manera más eficiente y efectiva. Sin embargo, estos modelos requieren una considerable cantidad de recursos computacionales y energéticos, lo que plantea desafíos en términos de sostenibilidad y accesibilidad (Taye, 2023). Además, la interpretación de los modelos de deep learning sigue siendo una tarea compleja, ya que la naturaleza 'caja negra' de estas redes puede dificultar la comprensión de cómo llegan a ciertas decisiones o clasificaciones (Taye, 2023).

Un modelo de redes neuronales de vanguardia que destaca es MiDaS, el cual realiza estimaciones de profundidad utilizando una cámara monocular. En la revisión bibliográfica a la fecha de realización del proyecto, se encuentra que es el primer modelo que se encuentra implementado y es fácilmente accesible a través del repositorio de TensorFlow 2. Desarrollado por Ranftl et al. (2022), MiDaS ha sido entrenado con múltiples conjuntos de datos para facilitar la transferencia de aprendizaje entre diversos contextos sin la necesidad de ejemplos previos (zero-shot). Este modelo puede generar imágenes de disparidad a partir de imágenes únicas, convirtiéndolo en una herramienta para la percepción de profundidad en

aplicaciones de visión por computadora. Sin embargo, este modelo presenta limitaciones en la precisión de los valores medidos, los cuales no son completamente fiables. Esto restringe su uso principalmente a la identificación de la forma de los objetos en las imágenes, sin proporcionar una escala precisa de las distancias.

3.7.1. Capas de una Red Neuronal Artificial (ANN) y Red Neuronal Convolutiva (CNN)

Las redes neuronales artificiales (ANN) y las redes neuronales convolucionales (CNN) están compuestas por diferentes tipos de capas, cada una con funciones específicas para el procesamiento de datos. En este contexto, se describen brevemente las capas más relevantes usadas en este proyecto como lo hace TensorFlow (2024):

La capa **Conv1D** en una red neuronal convolutiva (CNN) es para el procesamiento de datos secuenciales, como sonidos o series temporales. Esta capa utiliza filtros o *kernels* que se desplazan a lo largo del eje temporal de los datos de entrada, aplicando una convolución matemática en cada posición. El resultado es un tensor de salida que representa características localizadas de la entrada, lo que permite a la red detectar patrones temporales en cualquier parte de los datos de entrada.

Las capas **Dense**, también conocidas como capas completamente conectadas, son el tipo más común de capas en las redes neuronales. En estas capas, cada neurona recibe entradas de todas las neuronas de la capa anterior, lo que permite a la red aprender relaciones complejas entre los atributos de los datos. Son esenciales para integrar la información aprendida por la red en las capas anteriores y para producir la salida final, dependiendo de la tarea específica, como la clasificación o la regresión.

Una capa **Dropout** es una técnica para reducir el sobreajuste en las redes neuronales mediante la eliminación aleatoria de conexiones entre las neuronas

durante el entrenamiento. En cada paso del entrenamiento, cierto porcentaje de las neuronas se "desactiva" aleatoriamente, lo que significa que sus contribuciones a la activación de neuronas posteriores temporalmente se eliminan y no se actualizan durante el paso de retro propagación. Esto evita que el modelo dependa demasiado de cualquier entrada individual y promueve la generalización.

La capa **Flatten** se utiliza para convertir la estructura de datos multidimensional que sale de las capas convolucionales en un formato unidimensional. Esta transformación es necesaria para que los datos puedan ser procesados por capas **Dense**, que requieren entradas lineales.

En cuanto a las **funciones de activación**, la función **ReLU** (Rectified Linear Unit) es una de las más populares debido a su simplicidad y eficacia, proporcionando una salida no lineal que es cero para todos los valores negativos y lineal para todos los valores positivos. La función **Softmax** se utiliza en la última capa de una red para clasificación, donde convierte los valores no normalizados de las neuronas, en probabilidades facilitando la interpretación de la salida como probabilidades de clase.

3.7.2. Entrenamiento de una Red Neuronal Artificial (ANN) o Red Neuronal Convolucional (CNN)

El proceso de entrenamiento de una red neuronal ya sea convolucional (CNN) o artificial (ANN), es donde se aprenden los pesos óptimos de las conexiones neuronales a partir de los datos proporcionados. Los siguientes parámetros y conceptos descritos en TensorFlow (2024) se utilizan en este proceso:

El **learning rate** determina la magnitud de los pasos que se toman en la dirección del gradiente de la función de pérdida durante la optimización. Un **learning rate** demasiado alto puede hacer que el entrenamiento oscile sin encontrar un mínimo, mientras que uno demasiado bajo puede resultar en un proceso de aprendizaje muy lento o en la detención prematura en un mínimo local.

Las **épocas** se refieren a las veces que el algoritmo de entrenamiento trabaja a través de todos los datos. Cuantas más épocas, más veces el modelo tiene la oportunidad de aprender y ajustar sus pesos a partir de los datos, aunque también existe el riesgo de sobreajuste si el número de épocas es excesivamente alto sin una adecuada regularización.

El **batch size** influye directamente en la precisión de la estimación del gradiente en cada paso del entrenamiento. Un *batch size* más grande proporciona una estimación más precisa, pero también requiere más memoria y puede ser más lento por iteración. Un tamaño más pequeño puede hacer que el entrenamiento sea más rápido y también puede introducir un útil elemento de ruido en el entrenamiento, a veces ayudando a evitar mínimos locales.

la **función de pérdida *sparse categorical cross entropy*** es ideal para tareas de clasificación donde las etiquetas de clase están en formato de enteros. Esta función compara la distribución de probabilidad predicha por el modelo frente a la distribución real de las etiquetas, penalizando las predicciones en función de su divergencia de los valores verdaderos, lo que permite una eficiente retroalimentación y ajuste del modelo durante el entrenamiento.

El **optimizador Adam** es un método de optimización utilizado para actualizar los pesos en redes neuronales durante el entrenamiento, basándose en estimaciones adaptativas de momentos de primer y segundo orden. Combina características de dos técnicas anteriores, *AdaGrad* y *RMSProp*. *AdaGrad* adapta la tasa de aprendizaje para cada parámetro individualmente, mejorando el rendimiento en problemas con datos dispersos, mientras que *RMSProp* ajusta la actualización de *AdaGrad* para acumular solo gradientes recientes, evitando la desaceleración del aprendizaje en etapas avanzadas.

3.8. Modelo de cámaras estéreo

Para calcular la altura del dron con respecto al terreno se utilizó el modelo de una cámara estéreo para que, mediante una sola cámara y moviendo el UAV en diferentes posiciones, se pueda medir la profundidad.

Una cámara estéreo consiste en un sistema que integra dos cámaras dispuestas para imitar la visión binocular humana. Cada cámara captura una imagen desde una perspectiva ligeramente diferente, permitiendo así calcular la profundidad del campo visualizado a través de la diferencia entre las imágenes, conocida como disparidad. Este sistema es fundamental en aplicaciones que requieren percepción de profundidad, como la robótica autónoma y la modelización 3D (OpenCV, 2021).

El modelo matemático detrás de las cámaras estéreo se basa en la geometría epipolar, que describe cómo se relacionan los puntos de una imagen con los de la otra a través de líneas epipolares. Esto implica que, para cada punto en una imagen, su correspondiente en la otra imagen debe estar en la misma línea epipolar. La intersección de estas líneas con las imágenes de ambas cámaras permite determinar la posición tridimensional de los puntos observados (Wei et al., 2023).

Una cámara estéreo puede ser representada como se describe en la Figura 10:

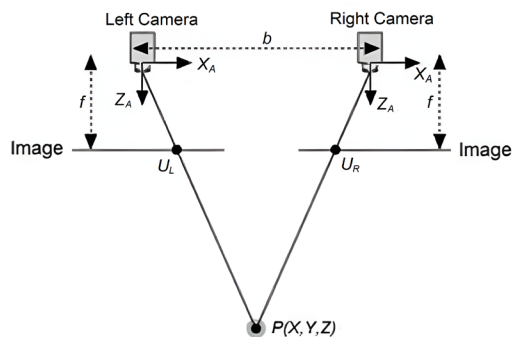


Figura 10. Modelo de Cámara estéreo Fuente (Karunanayake, 2016).

En donde, b representa la línea base o la distancia entre las dos cámaras, izquierda y derecha. f es la longitud focal de las cámaras. X_A es el eje X de una

cámara. Z_A es el eje óptico de una cámara. P es el punto físico definido por las coordenadas X, Y y Z . U_L es la proyección del punto físico real P en una imagen capturada por la cámara izquierda. U_R es la proyección del punto físico real P en una imagen capturada por la cámara derecha.

En Karunanayake (2016) se describe que la disparidad se calcula como:

$$d = U_L - U_R \quad (6)$$

Donde la profundidad de un punto es.

$$Z = b * \frac{f}{d} \quad (7)$$

Para que esta cámara estéreo simulada pueda dar mediciones correctas es indispensable realizar una calibración para obtener información precisa de profundidad a partir de las imágenes capturadas.

Este proceso se realiza para determinar los parámetros internos y externos de las cámaras, que incluyen las distancias focales, el tamaño de los píxeles, las coordenadas del centro de la imagen y la orientación y posición relativas de las cámaras en el sistema estéreo (Zhao & Xu, 2023; School of Geomatics et al., 2022).

La calibración de la cámara estéreo suele implicar un patrón de tablero de ajedrez, usado como objeto de calibración conocido. Las esquinas del tablero de ajedrez se detectan en las imágenes y se utilizan para estimar los parámetros de la cámara. La geometría del tablero de ajedrez (es decir, la ubicación tridimensional de las esquinas en un sistema de coordenadas del mundo) se conoce, y esto permite que se establezca una correspondencia entre los puntos de objeto en el mundo 3D y los puntos de imagen en el plano de la imagen 2D. A través de esta correspondencia, se pueden estimar los parámetros de la cámara.

En términos matemáticos, la calibración de la cámara estéreo implica resolver un sistema de ecuaciones que relaciona los puntos del mundo 3D con los puntos de

la imagen 2D a través de la matriz de proyección de la cámara. Esta matriz de proyección se compone de los parámetros intrínsecos y extrínsecos de la cámara.

Los parámetros intrínsecos se refieren a las propiedades de la cámara que no cambian con la orientación o la ubicación de la cámara, como la distancia focal y el centro de la imagen. Los parámetros extrínsecos se refieren a la orientación y la posición de la cámara en el espacio tridimensional.

3.9. Nubes de puntos

Para obtener una nube de puntos a partir de las imágenes de la cámara estéreo, se usa el modelo matemático anteriormente descrito. La precisión de este proceso depende en gran medida de la precisión de los parámetros de la cámara obtenidos a través de la calibración.

En primer lugar, se capturan dos imágenes distintas con la cámara en dos ubicaciones separadas, lo que proporciona perspectivas ligeramente diferentes de la misma escena. Luego, se rectifican las imágenes para transformarlas de manera que las líneas epipolares se alineen. Esto simplifica el problema de encontrar correspondencias entre las dos imágenes.

A partir de las imágenes rectificadas, se calcula la disparidad entre los píxeles correspondientes en las dos imágenes. La disparidad es simplemente la diferencia en la posición x (horizontal) de un punto en las dos imágenes como se describe en la ecuación (6). Esta disparidad se puede utilizar para calcular la profundidad o la distancia al punto en el mundo 3D utilizando la geometría epipolar y la distancia conocida entre las dos cámaras.

Finalmente, se puede crear una nube de puntos 3D mapeando cada píxel (o un subconjunto de píxeles) y su disparidad correspondiente a una ubicación en el espacio 3D. Este proceso se conoce como reconstrucción 3D y resulta en una nube de puntos que representa la escena original en tres dimensiones.

3.10. Regresión lineal múltiple

Para procesar la nube de puntos se propone utilizar un modelo de regresión lineal múltiple, el cual analiza la influencia de una variable explicativa X en los valores que toma otra variable denominada dependiente (Y). Con el fin de poder encontrar una ecuación de un plano que resuma las características como la altura y la inclinación de la nube de puntos. La ecuación de este modelo es:

$$y = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \dots + \beta_k * x_k + \epsilon \quad (8)$$

Donde $\beta_0, \beta_1, \dots, \beta_k$ son los coeficientes que se estiman mediante la minimización de la suma de los cuadrados de los residuos, y ϵ representa el error aleatorio asociado con las predicciones del modelo, (Breiman y Friedman, 2002),

En la regresión lineal múltiple, los coeficientes son elegidos de forma que la suma de cuadrados entre los valores observados y los pronosticados sea mínima, es decir, que se va a minimizar la varianza residual (Dorta-González, 2023)

3.11. Propagación de errores.

Para comprender cómo los errores de posición pueden afectar las mediciones realizadas con un UAV, es esencial entender la propagación de errores a través de un modelo matemático. La propagación de errores, un concepto fundamental en la experimentación científica describe cómo las incertidumbres en las mediciones de entrada de un sistema o proceso influyen en la incertidumbre de una cantidad derivada. Este análisis es indispensable cuando se utilizan las mediciones para calcular otras cantidades o resultados mediante una función matemática.

Si se tienen varias variables (x_1, x_2, \dots, x_z) medidas con incertidumbres $(\delta x_1, \delta x_2, \dots, \delta x_z)$ respectivamente y estas variables se emplean para computar una función $(q(x_1, x_2, \dots, x_z))$. Si las incertidumbres en (x_1, x_2, \dots, x_z) son independientes y aleatorias, entonces la incertidumbre en (q) puede calcularse utilizando la fórmula de propagación de errores (Taylor, 1997):

$$\delta q = \sqrt{\left(\frac{\partial q}{\partial x_1} \delta x_1\right)^2 + \left(\frac{\partial q}{\partial x_2} \delta x_2\right)^2 + \dots + \left(\frac{\partial q}{\partial x_z} \delta x_z\right)^2} \quad (9)$$

Esta ecuación demuestra que la incertidumbre en (q) es la raíz cuadrada de la suma de los cuadrados de las incertidumbres parciales, donde cada término es el producto de la derivada parcial de (q) con respecto a cada variable y la incertidumbre correspondiente.

Además, la incertidumbre total nunca supera la suma ordinaria de las incertidumbres individuales:

$$\delta q \leq \left|\frac{\partial q}{\partial x_1}\right| \delta x_1 + \left|\frac{\partial q}{\partial x_2}\right| \delta x_2 + \dots + \left|\frac{\partial q}{\partial x_z}\right| \delta x_z \quad (10)$$

Finalmente, para observar de manera más efectiva el impacto del error en la medida, resulta útil considerarlo como un error porcentual o error fraccional. La fórmula para calcularlo sería:

$$\frac{\delta q}{q} = \frac{\sqrt{\left(\frac{\partial q}{\partial x_1} \delta x_1\right)^2 + \left(\frac{\partial q}{\partial x_2} \delta x_2\right)^2 + \dots + \left(\frac{\partial q}{\partial x_z} \delta x_z\right)^2}}{q} \quad (11)$$

Este análisis permitió minimizar el impacto de las desviaciones en la posición del UAV, permitiendo una toma de medidas con una precisión adecuada para este proyecto.

4. Metodología

Para dar respuesta al problema planteado se siguió una metodología de Investigación Aplicada con el diseño e implementación de un algoritmo para calcular sin información previa del terreno, la trayectoria para un vuelo fotogramétrico para praderas, laderas o valles, manteniendo el GSD con variaciones menores al 20%.

En esta sección, se discutirán los métodos empleados en el desarrollo del algoritmo, los principales desafíos encontrados y las estrategias implementadas para resolverlos. Inicialmente, es fundamental establecer de manera explícita los procedimientos secuenciales que se desean ejecutar para que el Vehículo Aéreo No Tripulado (UAV) logre planificar eficazmente la ruta de vuelo.

El diseño del algoritmo se estructuró en dos etapas fundamentales: el vuelo de reconocimiento y el cálculo de la ruta de vuelo.

Para el vuelo de reconocimiento, el principal objetivo fue el de reducir su duración con una planificación en tiempo real, evitando la necesidad de contar con un modelo digital de elevación (DEM). Esto se consigue identificando el tipo de terreno que se va a fotografiar y para esto se clasifican de acuerdo con el número de pendientes que presenta, ya que un terreno será una pradera si no se encuentran puntos de elevación, una ladera si contiene una o un valle si contiene dos puntos de elevación., Una vez determinado el tipo de terreno se mide la altura con respecto al suelo y se construye una nube de puntos que permite identificar la inclinación del terreno.

El siguiente paso es el cálculo de la ruta de vuelo que depende de la clasificación del terreno y que se realiza en la dirección perpendicular al gradiente del plano que representa la inclinación del terreno, que resulta ser muy cercano a las curvas de nivel que lo representan.

El algoritmo se evaluó creando 15 entornos de simulación que replican el comportamiento del UAV en diferentes terrenos inclinados de praderas, valles y

laderas que son descritos en el Anexo 3. Dentro de estos entornos se buscó que la ruta de vuelo mantenga el GSD dentro del 20% de desviación.

Con base a lo anterior se presenta una definición formal del algoritmo general que debe de seguir el UAV durante cada vuelo en la Figura 11.

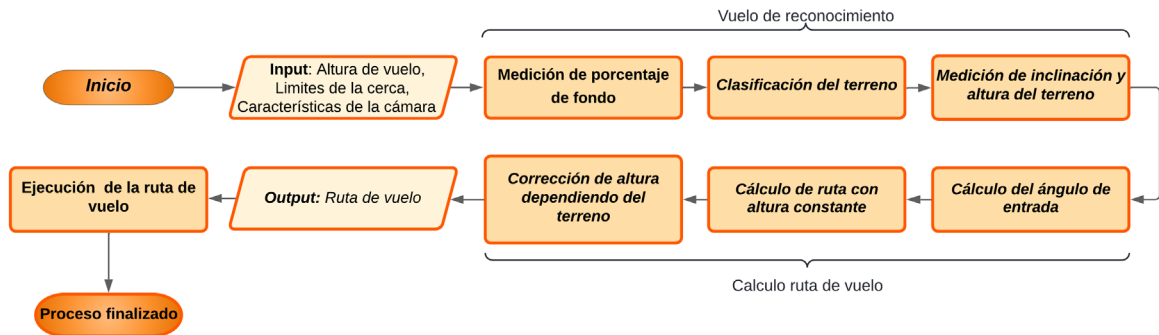


Figura 11. Estructura del algoritmo general para el vuelo fotogramétrico en terrenos inclinados.

A continuación, se hace una descripción detallada de cada etapa del algoritmo general.

4.1. Medición de características del terreno circundante

La primera parte consiste en realizar mediciones del horizonte visible sobre un punto del terreno a clasificar, con el objetivo de identificar elevaciones. Este método consiste en elevar el UAV a 10 metros del suelo con el fin de evitar oclusiones por estructuras cercanas y tomar fotos en orientación horizontal al terreno, realizando una rotación total a la derecha sobre el eje vertical del UAV en pasos de 15°, consiguiendo un traslape entre las tomas del 75% ya que la cámara tiene un campo de visión horizontal (FOV) de 65°.

En cada una de las fotografías tomadas se calcula el porcentaje de fondo (PF) con la CNN MiDas (sección 3.7), El PF se define como el porcentaje de píxeles de una imagen de disparidad que se encuentran a una distancia superior al 80% de la distancia máxima de la imagen y se expresa así:

$$\text{Porcentaje de fondo} = \left(\frac{N_{dist > (0.8 \times D_{max})}}{N_{total}} \right) \times 100 \quad (12)$$

Donde, $N_{dist > (0.8 \times D_{max})}$ es el número de píxeles en los que la distancia es superior a 80% de la distancia máxima, D_{max} es la distancia máxima encontrada en la imagen, y N_{total} es el número total de píxeles en la imagen.

Por cada imagen capturada se obtiene una medición del PF, con lo que se cuentan 24 mediciones utilizadas para clasificar el terreno según el número de elevaciones. Este proceso fue implementado en el UAV siguiendo el algoritmo mostrado en la Figura 12.

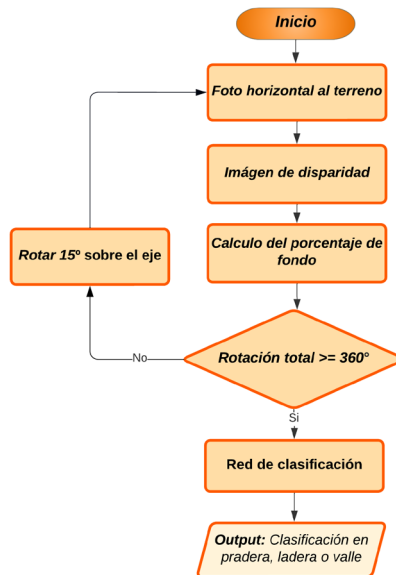


Figura 12. Algoritmo de medición de características del terreno.

En la Figura 13 y en la Figura 14 se pueden observar los patrones que se forman al graficar los datos en coordenadas cilíndricas tomando como altura el valor de $(100 - PF)$ y un radio constante.

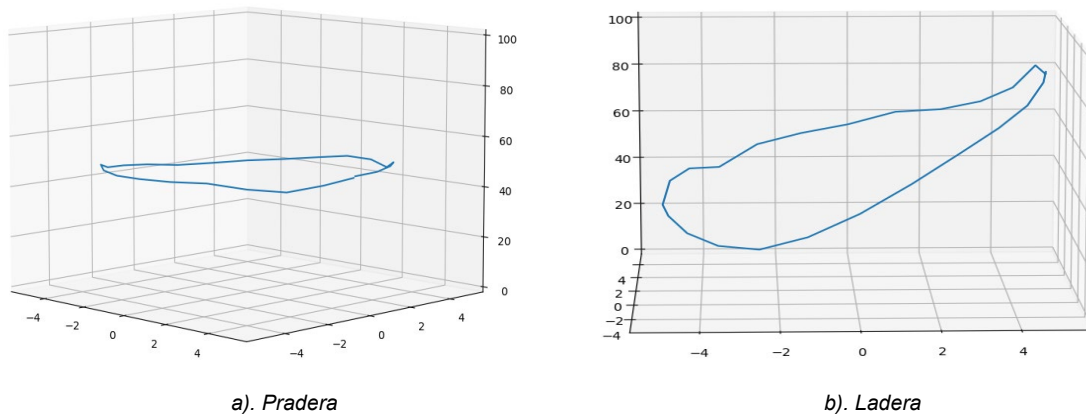


Figura 13. Curvas paramétricas resultantes de la medición de una pradera y una ladera.

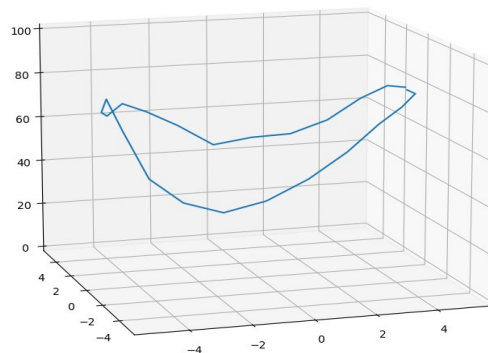


Figura 14. Curva paramétrica resultantes de la medición de un valle.

Conociendo la orientación inicial de la cámara en el terreno se puede determinar la orientación en la cual se detecta una elevación. Aquí se ha determinado que una ladera tendrá una única elevación, un valle tendrá dos y una pradera ninguna.

4.2. Clasificación del terreno.

La clasificación del terreno se hizo a partir de la medición de las características del terreno, mediante redes neuronales artificiales ANN, y mediante redes convolucionales de una dimensión CNN.

A partir de la literatura existente se probaron diferentes arquitecturas con varios niveles de complejidad y robustez, para determinar el nivel de dificultad en la tarea y encontrar la mejor arquitectura para resolverlo.

4.2.1. Creación de base de datos

La generación de la base de datos se hizo tomando medidas de las características del terreno circundante en 376 vuelos, los cuales se tomaron a lo largo de los 15 terrenos en diferentes puntos de partidas y diferentes orientaciones tomando de forma equitativa las tres clases de terrenos.

Cada muestra de la base de datos es un vector de 24 valores obtenidos del procedimiento anterior y se obtendrá como salida el tipo de terreno (pradera, ladera o valle).

Esta base de datos se usó para el entrenamiento de los modelos de clasificación, en donde se dividió de forma aleatoria, dejando 80% de los datos como entrenamiento y el 20% de los datos como validación.

4.2.2. Arquitecturas evaluadas

Se usaron tres (3) arquitecturas de redes neuronales que responden con diferentes niveles de complejidad: dos ANN y una CNN-1D.

Cada arquitectura se evaluó con una variación de las neuronas N de la primera capa oculta dentro de los siguientes valores (8, 16, 32, 64, 128) y el mismo patrón de entrenamiento: un optimizador Adam con un *learning rate* de 0.0001, una función de pérdida de *sparse categorical crossentropy*, 30 épocas y un *batch size* de 3.

La primera arquitectura fue una *Artificial Neural Network* (ANN) en la que se configuraron 3 capas. 1 capa oculta, 1 capa de entrada y 1 capa de salida. En la capa oculta, se utilizan N número de neuronas con una función de activación tipo *Relu* y en la capa de salida se tiene 3 neuronas con una función de activación tipo *Softmax*. En este modelo se varía el número de neuronas en la capa oculta capa, como se muestra en la Figura 15.

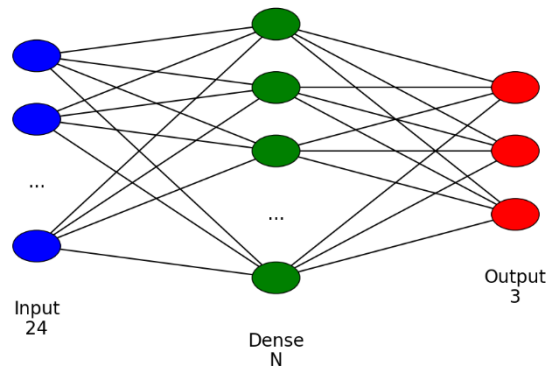


Figura 15. Arquitectura de clasificación ANN.

Para el segundo modelo tipo ANN se tuvieron 4 capas ocultas, en la capa oculta, se utilizan N número de neuronas con una función de activación tipo *Relu*, 1 capa *Dropout* que reduce los datos a la mitad, 1 capa *Flatten*, 1 capas *dense* con un numero de neuronas de la mitad del valor de N con una función de activación tipo *Relu* y en la capa de salida se tiene 3 neuronas con una función de activación tipo *Softmax*. Como se ve en la Figura 16.

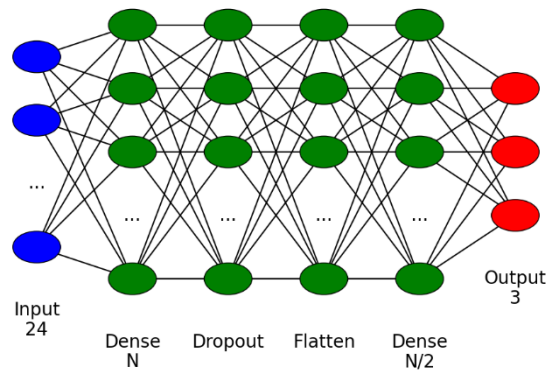


Figura 16. Arquitectura de clasificación ANN2.

Para el tercer modelo se utilizó una red convolucional de 5 capas ocultas, 2 capas de convolución Conv1D con un *kernel* de 3 en donde se varía la cantidad de filtros con los valores de N anteriormente mencionados, 1 capa *Dropout* que reduce los datos a la mitad, 1 capa *Flatten*, 1 capas *dense* con un numero de neuronas de la mitad del valor de N con una función de activación tipo *Relu* y en la capa de salida

se tiene 3 neuronas con una función de activación tipo *Softmax*. Como se ve en la Figura 17.

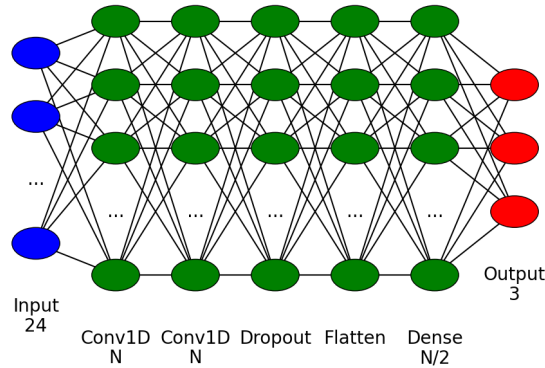


Figura 17. Arquitectura de clasificación 1D-CNN.

Para los 15 modelos propuestos se realizó el entrenamiento siguiendo los parámetros descritos en la (Sección 4.2.1), considerando que este proceso es aleatorio, se realizaron 50 entrenamientos para cada modelo realizando un análisis estadístico descriptivo que elimino la aleatoriedad del proceso y obteniendo una métrica más exacta en el proceso de selección del modelo.

La evaluación de estas arquitecturas tuvo en cuenta su precisión, tiempo de ejecución, variabilidad al entrenar un modelo y numero de parámetros. El criterio de selección se establece como un modelo que tuviera una precisión promedio en la validación superior al 96%, y sea el que menor número de parámetros y tiempo de ejecución tuviera.

4.3. Medición de la inclinación y la altura

La medición de la altura del UAV con respecto al suelo, no es trivial, debido a que el dron utilizado se encuentra en movimiento y tiene una desviación medida en las pruebas de 7 a 10 cm del punto objetivo. Teniendo en cuenta esto se propone un método de medición basado en el modelo de una cámara estéreo, usando solo la cámara principal del UAV.

Para poder minimizar el efecto que tiene la desviación en la posición del dron, se realizó un análisis de la propagación del error descrito en (9), al modelo de cámara estéreo descrito en (7). Este análisis se hizo de la siguiente manera:

$$\delta Z = \sqrt{\left(\frac{\partial Z}{\partial b} \delta b\right)^2 + \left(\frac{\partial Z}{\partial d} \delta d\right)^2 + \left(\frac{\partial Z}{\partial f} \delta f\right)^2} \quad (13)$$

Teniendo en cuenta que la longitud focal de la cámara no cambia durante el vuelo y haciendo un enfoque en el error en la línea base, se llegó a que el error en la altura estaba dado por:

$$\delta Z = \frac{f}{d} \delta b \quad (14)$$

Para poder analizar de mejor forma el impacto de este error en la medida, se analizó el error porcentual, dando como resultado la siguiente relación:

$$\frac{\delta Z}{Z} = \frac{\delta b}{b} \quad (15)$$

A partir de esta relación se determinó que aumentando la distancia base entre la toma de las fotos, se puede disminuir el impacto de la desviación de posición del UAV en la medición de la altura y la inclinación mediante el modelo de cámara estéreo.

Para términos del proyecto se propuso usar una distancia de base de dos metros el cual genera una propagación del error a la medición de la altura entre el 3.5% y el 5% utilizando la relación encontrada en (15).

El modelo se implementó utilizando exclusivamente la cámara principal del UAV en dirección Nadir y modificando la posición del dron para obtener dos fotografías con una separación de dos metros para emular la funcionalidad estereoscópica.

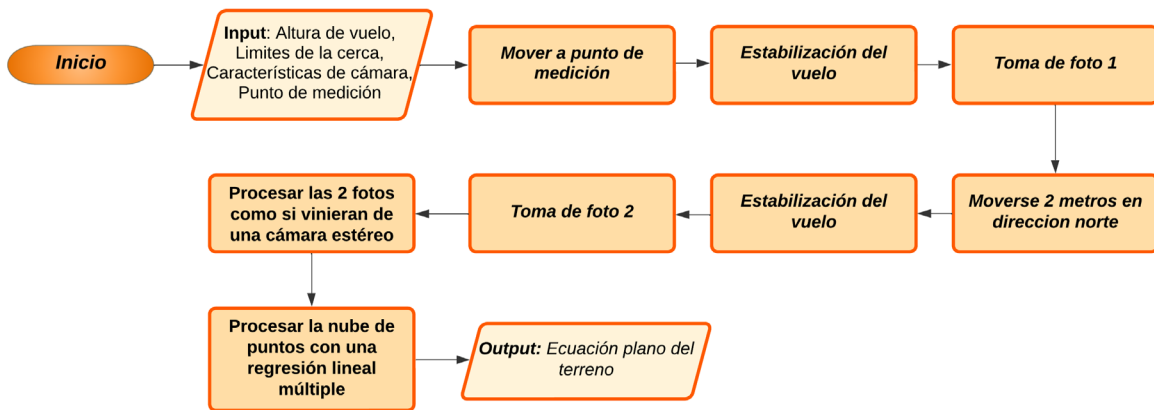


Figura 18. Algoritmo para medición de altura e inclinación del terreno.

En la Figura 18 se describe el algoritmo con el que se realizaron las mediciones, iniciando por determinar la posición adecuada que brinde más información de la altura e inclinación del terreno a partir de la curva paramétrica encontrada en la medición del porcentaje de fondo. Utilizando la altura preestablecida para el vuelo, se traza una línea desde el punto de despegue con el ángulo dado por la orientación de la inclinación hasta el límite de la cerca externa establecida, y se fija el punto de medición a una distancia correspondiente al 70% de la longitud total de la recta. Como se observa en la Figura 19, en donde el punto rojo es el punto de corte entre la recta y la cerca externa, y el punto verde es el punto de medición calculado.

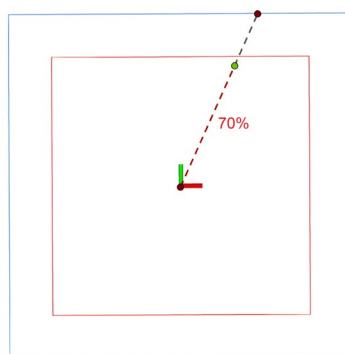


Figura 19. Cálculo para el punto de medición de altura e inclinación del terreno.

Para una ladera, se realizan dos mediciones: una en la posición de despegue y otra dada por la orientación encontrada en la curva paramétrica como se ve en la

Figura 13. En un valle, se efectúan 2 mediciones en las direcciones identificadas en la curva como se ve en la Figura 14.

Una vez calculadas las posiciones de medición, se procede a mover el UAV a esta posición. Ya en el lugar, se estabilizará la posición del UAV durante 10 segundos antes de tomar una fotografía Nadir del terreno. Posteriormente, se desplaza dos metros hacia el norte, se estabiliza nuevamente y se captura una segunda fotografía.

Utilizando estas dos imágenes, se genera una nube de puntos empleando las librerías de OpenCV (2021), adaptadas para su integración en ROS mediante un nodo que rectifica las imágenes y publica la nube de puntos en un mensaje.

Para la correcta medición de las distancias utilizando una cámara, se requiere que haya sido previamente calibrada utilizando por ejemplo el método del tablero de ajedrez. Teniendo en cuenta que esta cámara es simulada, se calibró también en un entorno simulado, empleando un tablero de ajedrez de 22 metros por 14 metros con una disposición de 9x7 casillas mostrado en la Figura 20, situándolo en diversas posiciones y orientaciones.

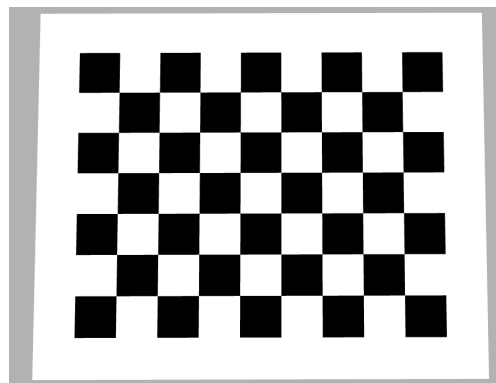


Figura 20. Tablero de ajedrez de 22x14 metros usado en el proceso de calibración.

Para obtener la información de la altura del terreno y de la inclinación de este, se propuso hacer una regresión lineal múltiple para hallar una ecuación de plano que representa la distribución promedio de la nube de puntos dentro del *sistema de referencia de la cámara*.

Para hacer la regresión lineal se establece que la ecuación del plano va a tener la siguiente forma.

$$Z = \beta_0 + \beta_1 \times X + \beta_2 \times Y \quad (16)$$

Esta ecuación representa a un plano que pasa por el punto $(0, 0, \beta_0)$ y con un vector normal de $(-\beta_0, -\beta_1, 1)$.

Este plano debe de moverse al sistema de referencia local, para esto ROS tiene una librería llamada TF el cual permite hacer estas transformaciones mediante un árbol de coordenadas, el cual contiene la información actualizada de las relaciones entre los diferentes sistemas coordenados.

4.4. Planeación de ruta de vuelo.

La segunda etapa del algoritmo es la planeación de la ruta de vuelo, para la cual se subdivide la tarea en tres actividades, el cálculo del ángulo de entrada, la planeación de la ruta de vuelo con altura constante y la corrección de alturas.

Teniendo como premisa que el algoritmo de planificación de trayectorias para fotogrametría debe garantizar que los parámetros de vuelo permitan generar un foto mosaico de calidad, se utilizan los parámetros definidos por el usuario, las características intrínsecas de la cámara, la clasificación del terreno, y las mediciones de altura e inclinación de los planos que describen el terreno.

4.4.1. Cálculo del ángulo de entrada en ruta de vuelo

Se va a definir el ángulo de entrada, como, el ángulo de las líneas de vuelo con relación al norte. Como se puede ver en Figura 21, se buscó en la investigación que el ángulo de vuelo siguiera la dirección de las curvas de nivel.

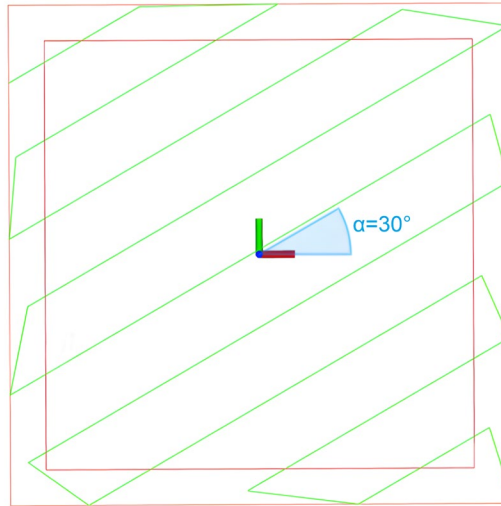


Figura 21. Angulo de entrada de 30° en vuelo fotogramétrico.

Dependiendo del tipo de terreno se calcula de una forma diferente el ángulo de entrada. Si es una pradera se toma un ángulo de 90° para tener la mejor ruta en un terreno que no presenta curvas de nivel pronunciadas como para seguirlas.

Para el cálculo del ángulo en una ladera, primero se tiene que unir los datos de las dos medidas realizadas, cada medida da como resultado una ecuación de un plano, Para unir estas dos ecuaciones que se encuentran en el sistema de referencia local, se van a calcular una cantidad de 100 puntos distribuidos en una matriz de 10 metros por 10 metros y centrada en el origen sobre cada plano. Para posteriormente hacer una regresión lineal múltiple de estos puntos y encontrar un plano promedio en el sistema de referencia local que tiene la forma descrita en (16).

A partir de la ecuación del plano promedio, se sacó el vector normal al plano, posteriormente se proyectó este vector al plano XY , se calculó el ángulo de la proyección con el eje X en el plano XY y finalmente se sumaron 90° a este ángulo.

Como se muestra en la Figura 22, en donde el vector normal al plano U se proyecta al plano XY en el vector V , posteriormente se saca un vector ortogonal al vector V sobre el plano XY , resultando en el vector W , este último vector se encontrará sobre el plano promedio y sobre el plano XY .

Este procedimiento permite encontrar el ángulo de entrada que siga la orientación del plano promedio.

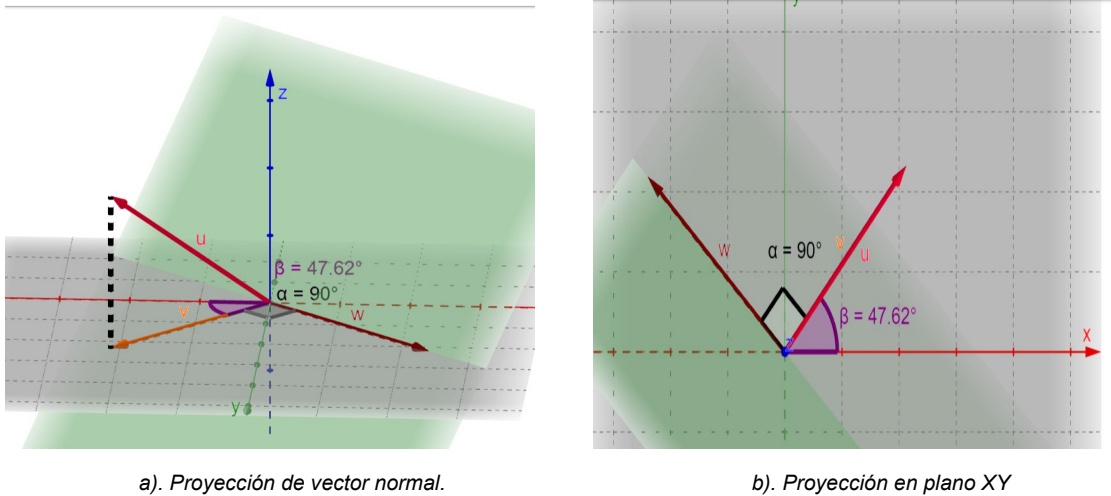


Figura 22. Cálculo de ángulo de entrada en laderas.

Si es un valle el ángulo se encuentra calculando la recta de intersección entre los dos planos medidos, y se toma el vector director de la recta y se proyecta al plano XY, para posteriormente encontrar el ángulo de la proyección con el eje X en el plano XY, como se visualiza en la Figura 23.

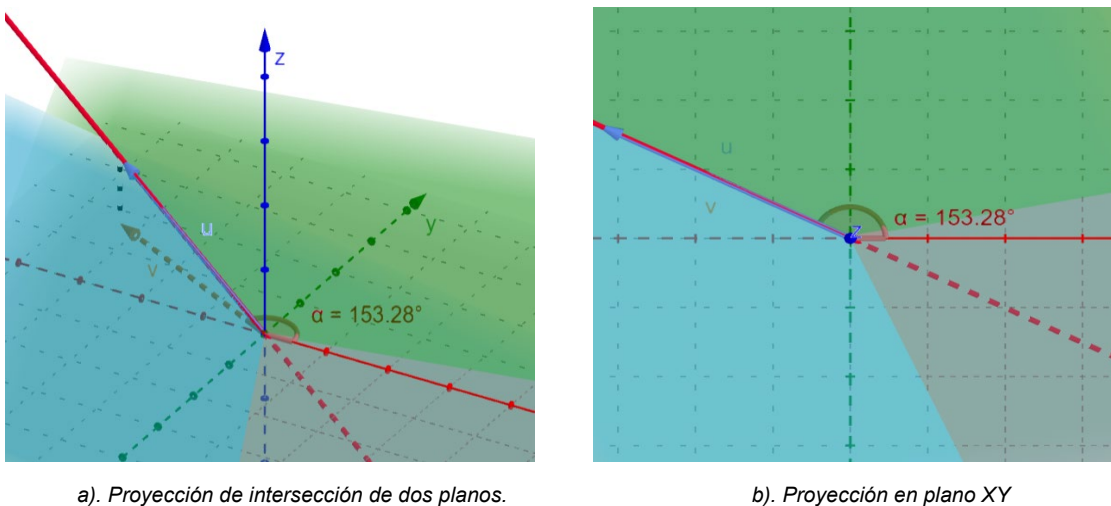


Figura 23. Cálculo de ángulo de entrada en valles.

4.4.2. Cálculo de ruta vuelo a altura constante.

Lo primero que se realiza para calcular la ruta de vuelo es calcular el GSD, el ancho de huella, el alto de huella, la separación de líneas de vuelo y la base en aire a partir de los parámetros definidos por el usuario y las características intrínsecas de la cámara.

Luego se realiza el cálculo de la cerca externa a partir de la cerca interna y la distancia de giro definidos por el usuario.

Con todos los parámetros definidos para el vuelo fotogramétrico se procede a definir el punto de arranque sobre uno de los vértices definidos en la Figura 24 en la cerca externa, usando el ángulo de entrada para definir cuál vértice es usado. Se uso el vértice 1, si el ángulo está entre 180° y 90° , el vértice 2, si está entre 90° y 0° , el vértice 3 si está entre 0° y -90° , y el vértice 4 si está entre -90° y -180° .

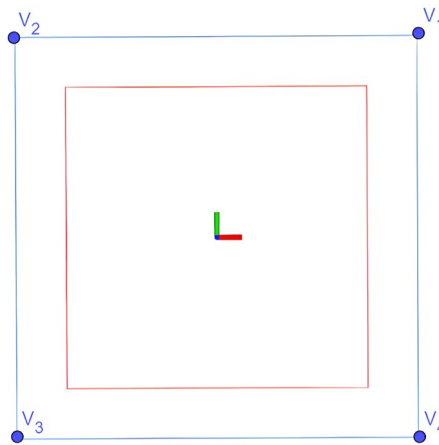


Figura 24. Punto de entrada en trayectoria de vuelo.

Después de calcular el punto de inicio, se define una recta con un vector director de magnitud igual a la base en aire (BA) con una dirección dada por el ángulo de entrada α , y que pasa por el punto de inicio (X_i, Y_i) , como se ve en la Figura 25.

$$(X, Y) = (X_i, Y_i) + k \cdot (BA \cdot \cos(\alpha), BA \cdot \sin(\alpha)) \quad (17)$$

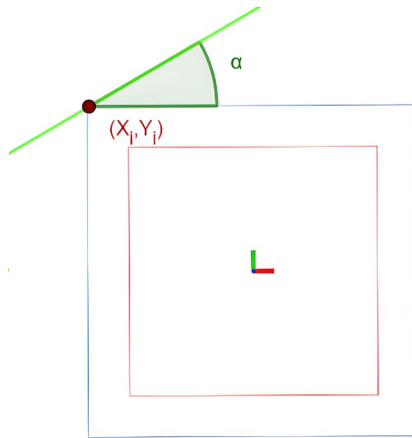


Figura 25. Primera recta definida para ruta de vuelo.

Esta primera recta no contiene puntos en la zona de medición, por lo que se procede a trasladar la recta a una distancia que corresponde a la separación de líneas de vuelo (*SLV*), y se busca el punto de intersección entre la cerca que permita avanzar dentro de la zona de vuelo, Este punto se denomina (X_1, Y_1) , como se puede ver en la Figura 26.

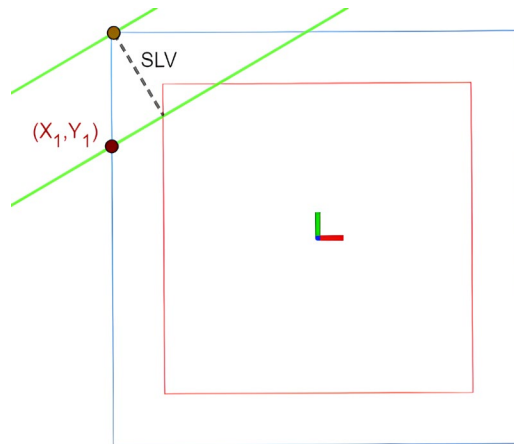


Figura 26. Recta trasladada y primer punto de ruta de vuelo.

Esta nueva recta estaría definida de la siguiente forma, la cual permite encontrar puntos con una separación de BA a partir del punto inicial al ir variando el parámetro k con valores enteros mayores a 0. Como se ve en la Figura 27.

$$(X, Y) = (X_1, Y_1) + k \cdot (BA \cdot \cos(\alpha), BA \cdot \sin(\alpha)) \quad (18)$$

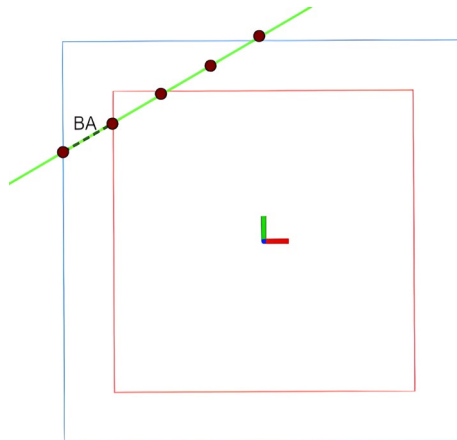


Figura 27. Puntos en ruta de vuelo sobre recta con separación entre ellos de BA.

Durante el proceso de variación del parámetro k , se verifica la pertenencia de cada punto dentro de la cerca. Si se detecta que un punto no se encuentra dentro de los límites de la cerca, este punto es descartado de la ruta de vuelo y se procede a hacer un desplazamiento de la recta a una distancia de SLV . Posteriormente, se recalcula la posición del siguiente punto como el punto de intersección más cercano entre la nueva recta y la cerca externa con el punto anterior de la ruta de vuelo, como se ve en la Figura 28.

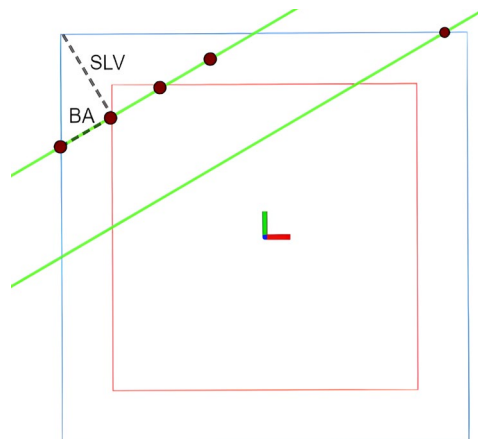


Figura 28. Punto en nueva recta trasladada de ruta de vuelo.

Este proceso se repite hasta que no se puede encontrar un punto en la recta trasladada que este dentro de los límites de la cerca, como se ve en la Figura 29, dando por terminado la planeación de la ruta de vuelo con altura constante.

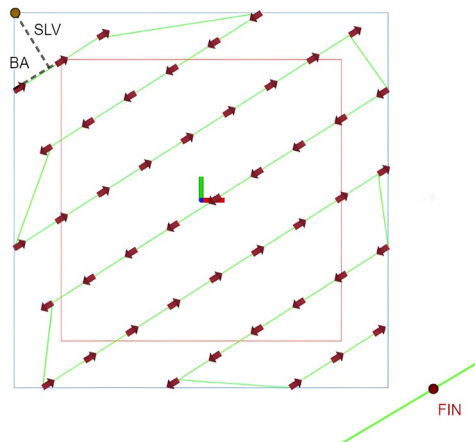


Figura 29. Fin de planeación de ruta de vuelo.

4.4.3. Corrección de altura en ruta de vuelo.

La siguiente etapa es hacer una corrección de alturas dependiendo del tipo de terreno clasificado.

Para una pradera no se va a realizar corrección de alturas, dado que este tipo de terreno no presenta inclinaciones considerables para tener en cuenta.

En el caso de una ladera se utiliza el plano promedio que se encontró durante el cálculo del ángulo de entrada, y que está en el sistema de referencia local, la corrección en la altura se calcula al revisar la altura de la proyección del punto a altura constante sobre el plano promedio, y posteriormente sumarle la altura deseada de vuelo.

La Figura 30 muestra como es esta corrección de alturas, teniendo en color azul el plano sobre el que se hizo la ruta de vuelo con altura constante, en color verde se muestra el plano promedio encontrado en el cálculo del ángulo de entrada, y en color rojo se muestra la proyección sobre la que estará la ruta de vuelo para hacer la corrección de la altura.

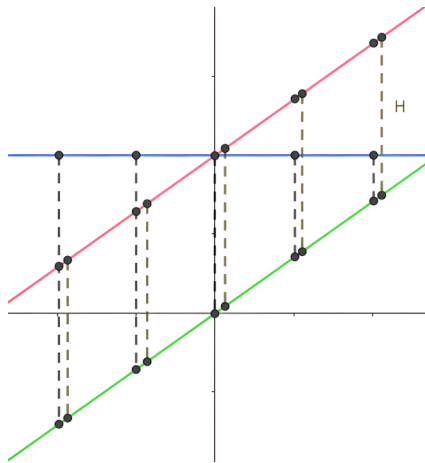


Figura 30. Corrección de alturas en laderas.

Para el caso del valle la corrección en la altura se calculo al revisar la altura de la proyección del punto a altura constante sobre los dos planos medidos y el plano XY , y posteriormente sumarle a cada una la altura deseada de vuelo, la altura seleccionada para cada punto será la mayor de las 3 calculadas.

En la Figura 31 se muestra como se hace la corrección de alturas para un valle, donde en color azul está el plano de la ruta de vuelo a altura constante, en verde están los 2 planos que se midieron, y en rojo se encuentran los puntos que resultan de la corrección de altura al seleccionar la altura mayor de los 3.

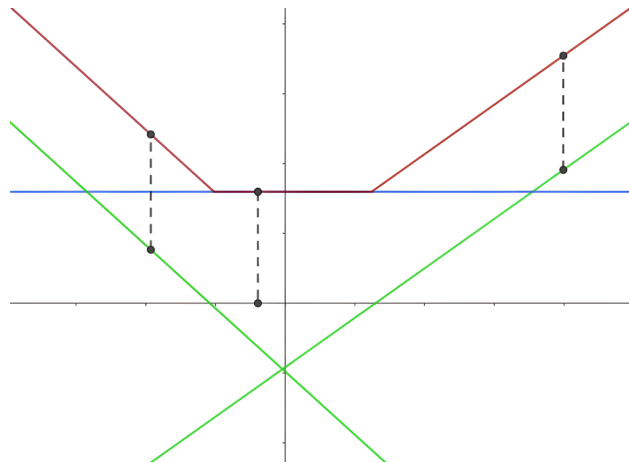


Figura 31. Corrección de alturas en valle.

5. Resultados y Análisis

En este apartado se presentan los principales resultados relacionados con el diseño de un vuelo fotogramétrico con UAV para Agricultura de Precisión en terrenos inclinados y el análisis correspondiente para validar el proceso de la metodología propuesta para esta investigación.

5.1. Resultado en la medición de características del terreno.

Como resultado para el algoritmo de medición de las características del terreno se muestra de ejemplo el proceso efectuado en el terreno Ladera 1. En la Figura 32 se observan las 24 imágenes tomadas durante el proceso de medición, en el que se observó como varia la cantidad de fondo visible conforme se rota en dirección a la ladera. En la Figura 33 las imágenes de disparidad calculadas con MiDaS, en las que se muestra la forma del terreno, en donde en blanco aparecen los puntos más lejanos y en negro los más cercanos. En la Figura 34 se muestra en color naranja los puntos que representan el porcentaje de fondo en la imagen de disparidad, en donde se observó que el porcentaje no es afectado por el terreno que se encuentra en la lejanía.

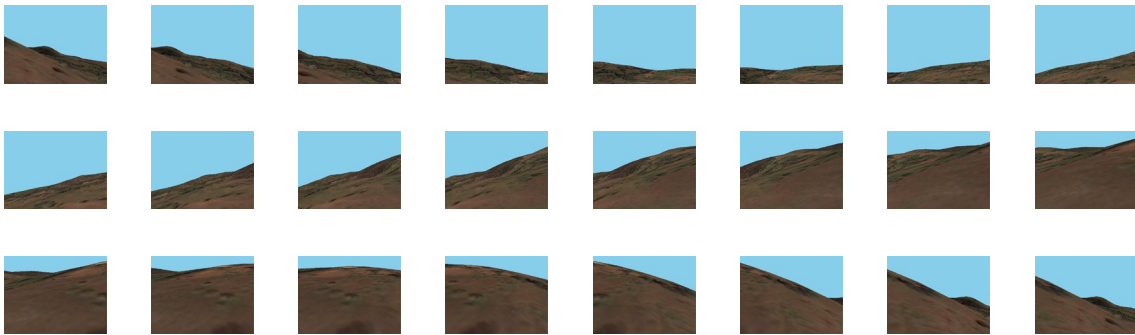


Figura 32. Imágenes tomadas durante el proceso de medición de características del terreno.

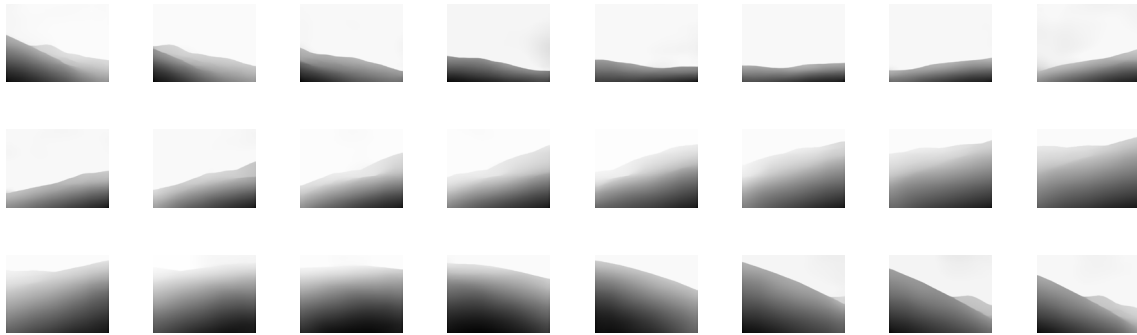


Figura 33. Imágenes de disparidad calculadas durante el proceso de medición de características del terreno.

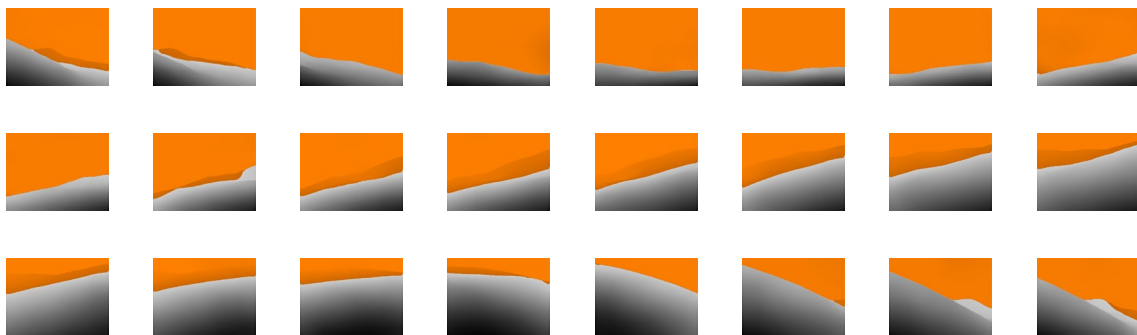


Figura 34. Porcentaje de fondo representado en naranja sobre las imágenes de disparidad.

Por último, en la Figura 35 y Figura 36 se muestra en color rojo la curva paramétrica graficada sobre el terreno en rviz, en color verde un modelo del terreno Ladera 1 y en color negro, las curvas de nivel del terreno. En estas figuras se observó que la curva paramétrica sigue la forma del terreno cercano, permitiendo usar esta medida para hacer la clasificación del terreno.

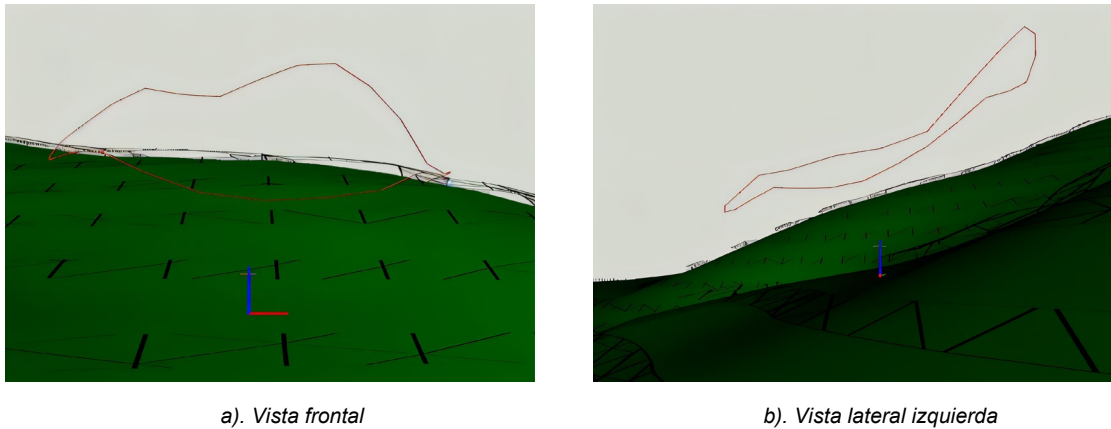


Figura 35, Vista frontal y vista lateral izquierda de curva paramétrica medida sobre el terreno en rviz.

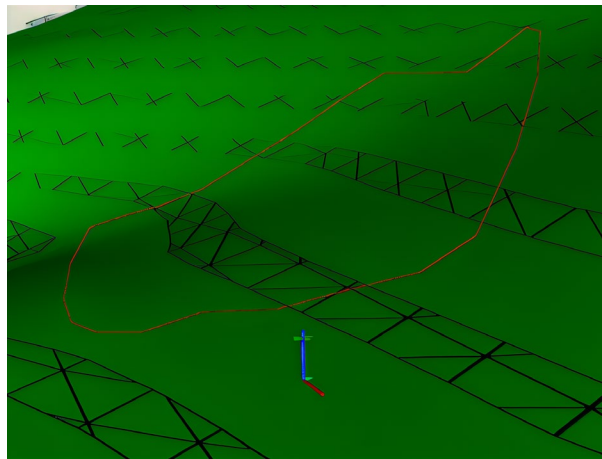


Figura 36. Vista isométrica de curva paramétrica medida sobre el terreno en rviz.

5.2. Entrenamiento y selección de red neuronal para la clasificación del terreno.

A continuación, se presentan los resultados del proceso de entrenamiento, selección del modelo de clasificación y las métricas de validación del modelo seleccionado.

5.2.1. Resultado del entrenamiento de cada modelo de clasificación propuesto.

En la Tabla 1, se presentan los estadísticos descriptivos que se encontraron para los 15 modelos propuestos, cada uno con 50 entrenamientos. Estos son la precisión

promedio en la validación y la desviación estándar en la precisión de cada entrenamiento.

Modelo	Precisión Promedio	Desviación Estándar
ANN-8	66.53%	7.04%
ANN-16	82.66%	6.25%
ANN-32	90.40%	3.09%
ANN-64	93.60%	2.78%
ANN-128	95.86%	1.51%
ANN2-8	80.26%	5.33%
ANN2-16	87.33%	4.82%
ANN2-32	93.73%	2.74%
ANN2-64	96.53%	1.49%
ANN2-128	99.20%	1.07%
1D-CNN-8	94.40%	2.59%
1D-CNN-16	97.20%	2.11%
1D-CNN-32	98.00%	1.61%
1D-CNN-64	98.93%	0.80%
1D-CNN-128	99.20%	0.65%

Tabla 1. Resultados del entrenamiento de las arquitecturas evaluadas.

En la Figura 37 se observa los resultados en la variabilidad del proceso de entrenamiento, mediante un análisis de cajas y vigotes de cada modelo propuesto.

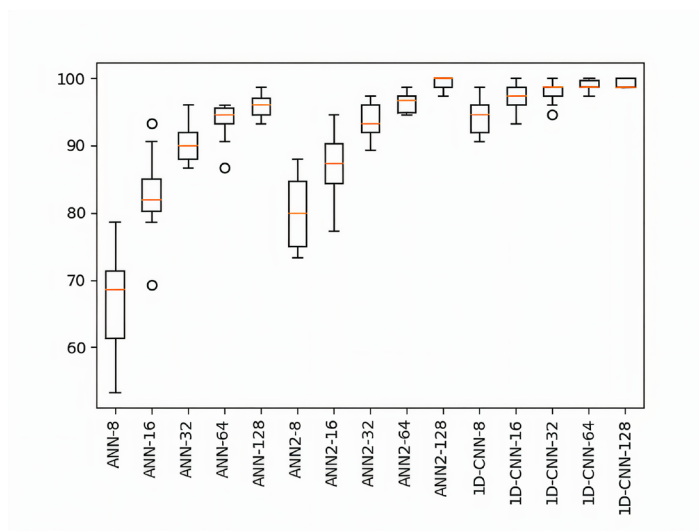


Figura 37. Diagrama de cajas y bigotes del entrenamiento de los modelos de clasificación propuestos.

5.2.2. Selección del modelo de clasificación.

Con el análisis descriptivo anterior, la selección del modelo de clasificación se realizó teniendo en cuenta la precisión promedio en la validación, el tiempo de ejecución y número de parámetros de cada modelo.

El criterio de selección se estableció como un modelo que tuviera una precisión promedio en la validación superior al 96%, y fuera el que menor número de parámetros en modelo y tiempo de ejecución tuviera. En la Tabla 2 se muestran los resultados obtenidos de tiempo de ejecución, número de parámetros, y la precisión promedio en la validación de cada modelo de clasificación propuesto.

Modelo	Tiempo de ejecución	Precisión en la validación	Numero de parámetros
ANN_8	496.63 us	66.53%	235
ANN_16	509.50 us	82.66%	467
ANN_32	527.86 us	90.40%	931
ANN_64	510.22 us	93.60%	1859
ANN_128	514.27 us	95.86%	3715
ANN2_8	905.99 us	80.26%	1411
ANN2_16	950.34 us	87.33%	2419
ANN2_32	925.06 us	93.73%	4435
ANN2_64	912.19 us	96.53%	8467
ANN2_128	943.42 us	99.20%	16531
1D_CNN_8	1783.13 us	94.40%	17435
1D_CNN_16	1736.64 us	97.20%	34851
1D_CNN_32	1686.57 us	98.00%	70835
1D_CNN_64	1702.07 us	98.93%	147411
1D_CNN_128	1781.94 us	99.20%	318995

Tabla 2. Parámetros de selección para los 15 modelos de clasificación.

En la Figura 38 se observan los datos de Tabla 2 graficados en escala logarítmica, representando el número de parámetros de cada modelo contra tiempo de ejecución que tiene el modelo para clasificar una medición y se muestra la precisión promedio de cada uno.

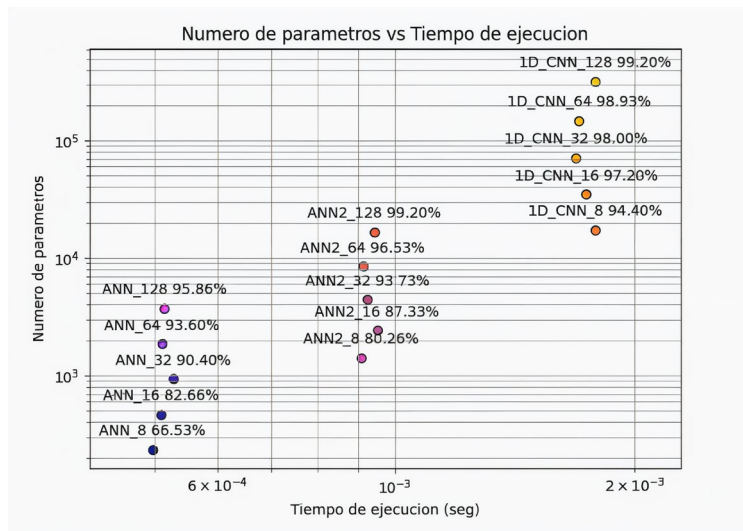


Figura 38. Numero de parámetros Vs Tiempo de ejecución de modelos de clasificación.

Teniendo en cuenta la Figura 38 y el criterio de establecido, se seleccionó el modelo **ANN2_64** para la solucionar el problema de clasificación.

5.2.3. Métricas de validación del modelo seleccionado.

A continuación, se muestran las métricas de validación para el modelo ANN2_64, el cual tiene una precisión del 98% en la validación, en estas métricas se muestra la matriz de confusión, la sensibilidad, la especificad la precisión y el *f1-score* para cada una de las clases.

En la Figura 39 se observa que este modelo tiene pequeñas dificultades al clasificar correctamente un valle, confundiéndola con una pradera, y clasificando una pradera en un valle. Este modelo logra clasificar correctamente la clase ladera.

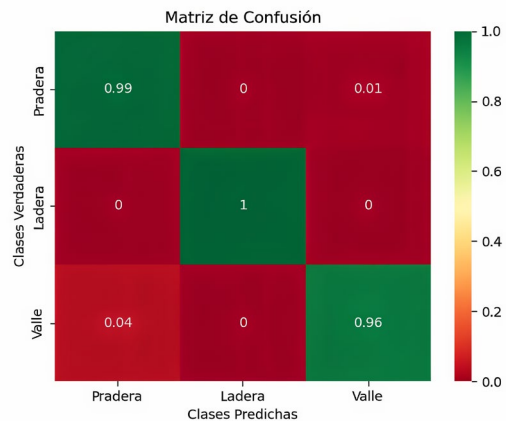


Figura 39. Matriz de confusión modelo ANN2_64.

En la Tabla 3 se agrupan los datos de validación del modelo, demostrando que este logró resolver adecuadamente la tarea de clasificación del terreno al tener una validación general superior al 97% destacando que logró un 100 % en la clasificación del terreno en ladera.

Clase	Precisión	Sensibilidad	Especificidad	f1-score
Pradera	94.00%	99.00%	97.82%	97.00%
Ladera	100.00%	100.00%	100.00%	100.00%
Valle	99.00%	96.00%	99.56%	98.00%

Tabla 3. Métricas de validación modelo ANN2_64.

5.3. Precisión de la medición de altura del terreno

La medición de la precisión de la altura usando la cámara estéreo simulada, se realizó tomando 1000 mediciones al mismo punto con el UAV a una altura de 50 metros con respecto al suelo determinando la exactitud y precisión del modelo propuesto, Para esto se calculó la raíz del error cuadrático medio o *RMSE*, la raíz del error cuadrático medio porcentual o *RMSPE*, la desviación estándar y el promedio en las medidas. Esto se compara con el error teórico de la cámara estéreo el cual es del 3.5% al 5%, como se observa en la Tabla 4 el *RMSPE* es de 2.54%, validando el uso del modelo matemático de una cámara estéreo simulada, para la medición de la altura del terreno en un UAV.

Desviación Estándar	Promedio	RMSE	RMSPE
1.18 m	50.02 m	1.26 m	2.54 %

Tabla 4. Exactitud y precisión de medidas con cámara estéreo simulada.

Esto se ejemplifica en la Figura 40 en la que se observa una nube de puntos obtenida con la cámara estéreo simulada sobre el modelo del terreno en escala de grises.

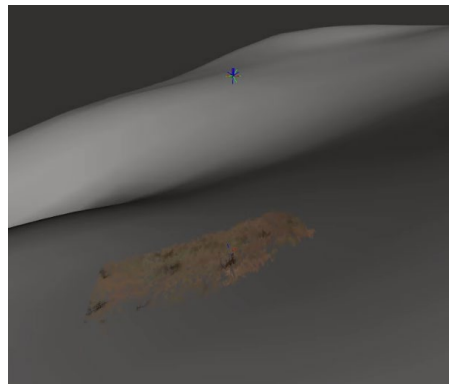


Figura 40. Nube de puntos recreada por cámara estéreo simulada.

5.4. Análisis de los resultados de la planeación de rutas de vuelo.

En esta sección se muestran los resultados de la planificación de las rutas de vuelo para los 15 terrenos estudiados, para esto se muestran en planos verticales las rutas de vuelo sobre las curvas de nivel de cada terreno, y se muestra el análisis de regresión que se encontró para determinar las desviaciones presentadas en el GSD para cada terreno.

5.4.1. Resultado de la ruta de vuelo.

A continuación, se observan las rutas de vuelo generadas por el algoritmo propuesto para cada terreno, que se generaron con los siguientes parámetros de vuelo: un GSD esperado de 2 centímetros por píxel, una cerca cuadrada de 240

metros de largo por 240 metros de ancho, un traslape longitudinal y lateral del 70 % y una distancia de giro de 20 metros.

En la Figura 41, se muestran las rutas de vuelo generadas para los 5 terrenos de praderas.

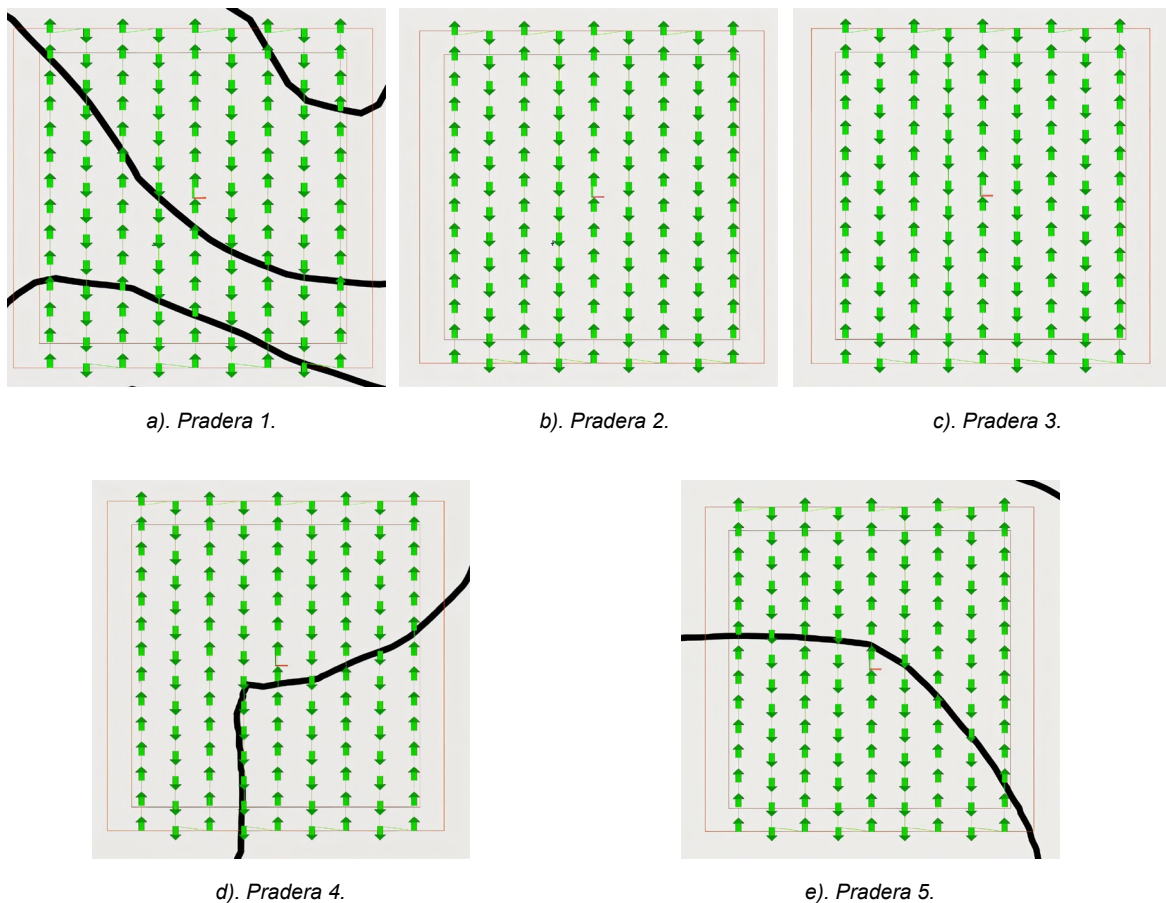


Figura 41. Rutas de vuelo generadas para los 5 terrenos de pradera.

Los 5 terrenos de pradera fueron clasificados correctamente como terrenos sin inclinación y se realizó una planeación de la ruta de vuelo manteniendo una altura constante en consecuencia, el ángulo de entrada en todos los terrenos fue de 90°.

En la Figura 42, se muestran las rutas de vuelo generadas para los 5 terrenos de laderas.

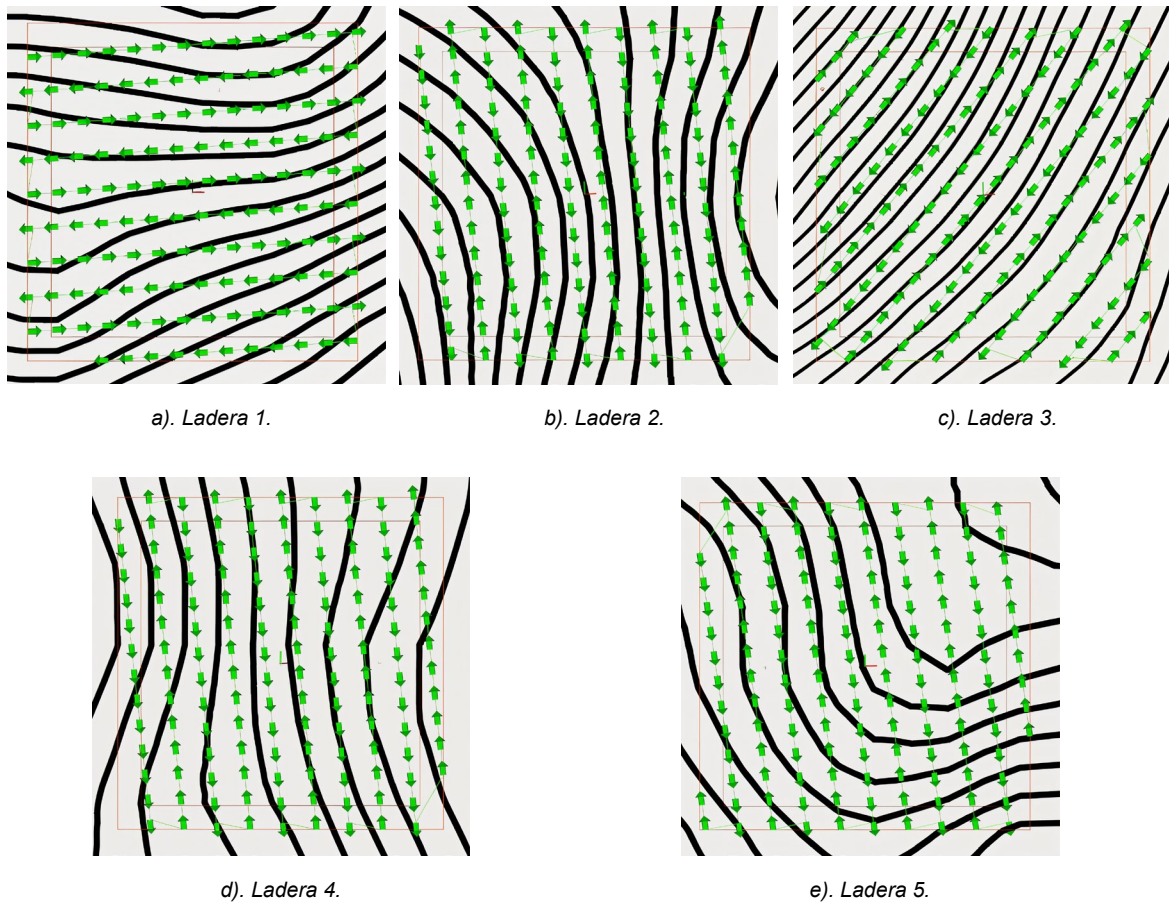


Figura 42. Rutas de vuelo generadas para los 5 terrenos de ladera.

Cuatro de los terrenos fueron clasificados correctamente como laderas, excepto el terreno Ladera 5, el cual, por su topografía, en ocasiones es clasificado como un valle, esto es de esperarse dado que el terreno cuenta con algunas características de un valle, como se observa en las curvas de nivel. Los ángulos de entrada de cada ruta de vuelo, se observa que se alinea con varias curvas de nivel en todos los terrenos, y para el terreno ladera 5, que es el más complicado, se observa que, al clasificarlo como ladera, sigue en mayor medida las curvas de nivel en la mayoría de su recorrido.

En la Figura 43, se muestran las rutas de vuelo generadas para los 5 terrenos de valles.

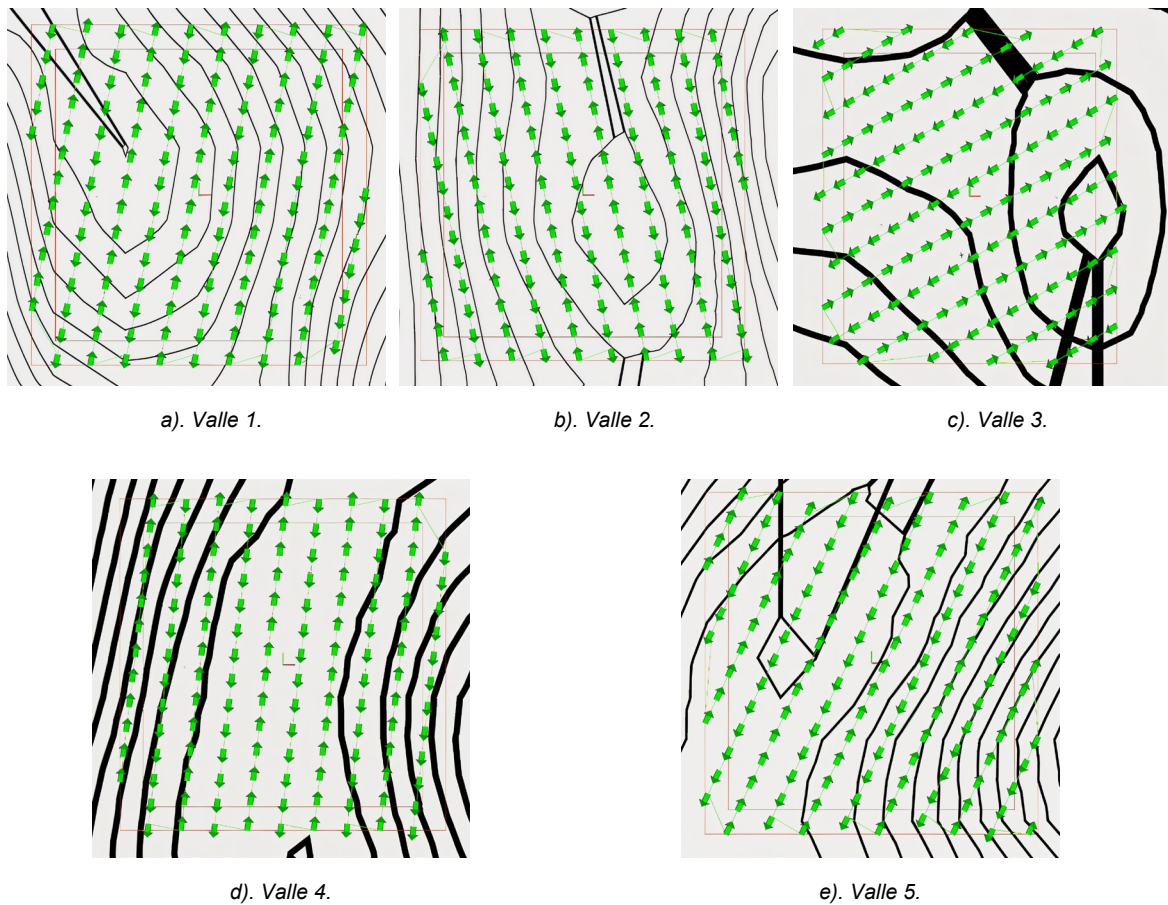


Figura 43. Rutas de vuelo generadas para los 5 terrenos de valle.

Los 5 terrenos fueron clasificados correctamente como valles, Se observa que el ángulo de entrada se ajusta con las curvas de nivel de las zonas más inclinadas del terreno, siendo en la mitad del valle el punto en el que se observan mayores diferencias al seguir las curvas de nivel, este debido a que en esta parte no hay una pendiente tan clara a seguir. Esto se percibe en mayor medida en el terreno valle 3, para el cual la parte central es lo suficientemente grande como para no visualizarse las dos inclinaciones que presenta el terreno.

5.4.2. Evaluación de la desviación del GSD en la ruta de vuelo.

Para analizar la desviación en el GSD durante el recorrido de una ruta de vuelo, se implementó en Gazebo un plugin el cual permite conocer la altura a la cual se

encuentra el UAV con respecto al terreno en todo momento, lo cual permite calcular el GSD de cada imagen capturada, cabe aclarar que este plugin es solo para la validación de los datos y no se usa para la planeación de la ruta de vuelo.

Para cada ruta de vuelo se calculó el promedio, la desviación estándar el RMSE y el RMSPE con las mediciones del GSD de las fotografías realizadas.

Para tener una comparativa del nivel de mejora que el algoritmo presenta, se va a tomar como una línea base una planeación de una ruta de vuelo a altura constante.

En la Tabla 5, se encontró que el GSD tiene un error que va hasta el 89.4%, no se consigue un GSD promedio cercano al esperado, y se tiene una alta desviación entre todos los puntos medidos para los terrenos de ladera y valle. Para el terreno de pradera se observa que el RMSPE es menor al 12%, presenta un promedio en la medida cercano al valor pedido, y tienes una dispersión en los datos menor a 0.22 cm/pix.

Terreno	Sin Método			
	Desviación Estándar (cm/pix)	Promedio (cm/pix)	RMSE (cm/pix)	RMSPE (%)
Pradera 1	0.215	2.142	0.235	11.468
Pradera 2	0.003	2.052	0.005	0.252
Pradera 3	0.003	2.053	0.005	0.280
Pradera 4	0.048	2.056	0.048	2.378
Pradera 5	0.089	2.004	0.100	4.894
Ladera 1	0.916	2.045	0.916	44.814
Ladera 2	0.757	1.901	0.771	40.580
Ladera 3	1.153	3.474	1.833	89.452
Ladera 4	0.533	2.102	0.536	26.182
Ladera 5	0.538	1.734	0.624	30.456
Valle 1	0.785	2.035	0.785	38.335
Valle 2	0.416	1.563	0.640	31.236
Valle 3	0.259	2.135	0.273	13.353
Valle 4	0.297	1.756	0.417	20.369
Valle 5	0.627	1.932	0.638	31.159

Tabla 5. Análisis de regresión del GSD efectuado en un vuelo a altura constante.

Estos datos confirman la necesidad de efectuar una planeación de la ruta de vuelo realizando cambios en la altura poder tener valores de GSD constantes.

En la Tabla 6, se consiguió un RMSPE que oscila entre 7.5% y 20% para las laderas y los valles, logrando cumplir el objetivo principal del proyecto. También se observa que los datos están menos dispersos en comparación a la línea base, y se obtiene un promedio en la medición del GSD cercano a lo esperado

Terreno	Con Método			
	Desviación Estándar (cm/pix)	Promedio (cm/pix)	RMSE (cm/pix)	RMSPE (%)
Pradera 1	0.215	2.142	0.234	11.432
Pradera 2	0.003	2.053	0.005	0.263
Pradera 3	0.002	2.054	0.005	0.280
Pradera 4	0.047	2.057	0.048	2.371
Pradera 5	0.090	2.005	0.100	4.895
Ladera 1	0.259	1.852	0.245	12.113
Ladera 2	0.338	1.984	0.325	17.552
Ladera 3	0.147	2.011	0.344	16.796
Ladera 4	0.270	2.089	0.152	7.454
Ladera 5	0.370	1.868	0.273	13.328
Valle 1	0.186	1.835	0.412	20.120
Valle 2	0.262	2.162	0.283	13.823
Valle 3	0.242	2.087	0.285	13.925
Valle 4	0.211	2.097	0.245	11.975
Valle 5	0.214	2.142	0.216	10.569

Tabla 6. Análisis de regresión del GSD efectuado con la planeación de ruta propuesta.

En la Tabla 7 se observó una mejora con el algoritmo propuesto comparando con la línea base para los terrenos de ladera y valle, Para los terrenos de pradera no se consigue una mejora debido a que en ambos casos la planeación de la ruta de vuelo asume un terreno sin inclinación y para el que no se necesita cambios en la altura. Cabe resaltar que para este tipo de terreno se cumple el objetivo establecido.

Terreno	Sin Método	Con Método
	RMSPE (%)	RMSPE (%)
Pradera 1	11.46	11.43
Pradera 2	0.25	0.26
Pradera 3	0.28	0.28
Pradera 4	2.37	2.37
Pradera 5	4.89	4.89
Ladera 1	44.81	12.11
Ladera 2	40.58	17.55
Ladera 3	89.45	16.79
Ladera 4	26.18	7.454
Ladera 5	30.45	13.32
Valle 1	38.33	20.12
Valle 2	31.23	13.82
Valle 3	13.35	13.92
Valle 4	20.36	11.97
Valle 5	31.15	10.56

Tabla 7. Comparación del RMSPE del GSD entre línea base y algoritmo propuesto para la planeación de rutas de vuelo.

5.5. Resultados de Integración en controladores Pixhawk 2 y Nvidia Jetson TX2.

En la Tabla 8 se observa el resultado de las pruebas de *hardware-in-the-loop*, en las que se evaluó el tiempo de ejecución desde el despegue hasta la planificación de la ruta de vuelo. Teniendo un tiempo de 3 minutos y medio para una pradera, 6 minutos y 21 segundos para la ladera y para el valle como se ve en la Tabla 9.

Terreno	Tiempo de vuelo de reconocimiento (seg)
Pradera 1	243.38
Pradera 2	243.38
Pradera 3	195.68
Pradera 4	185.91
Pradera 5	187.42
Ladera 1	421.44
Ladera 2	450.04
Ladera 3	328.49
Ladera 4	370.45
Ladera 5	347.68
Valle 1	353.43
Valle 2	348.27
Valle 3	375.47
Valle 4	411.15
Valle 5	421.24

Tabla 8. Tiempos de ejecución de algoritmo de planeación de rutas de vuelo en *hardware in the loop*.

TERRENO	PROMEDIO (seg)	DESVIACION ESTÁNDAR
Pradera	211.15	29.65
Ladera	379.85	37.76
Valle	381.91	33.11

Tabla 9. Tiempos de ejecución promedios por tipo de terreno en hardware in the loop.

Al hacer esta prueba se comprueba que este método es válido para funcionar en escenarios del mundo real.

6. Conclusiones y Trabajos Futuros

6.1. Conclusiones.

Los resultados obtenidos tras esta investigación permiten afirmar que tanto los objetivos específicos, como el objetivo general fueron logrados.

El algoritmo desarrollado para medir las características del terreno demostró ser efectivo, permitiendo obtener una curva paramétrica que describe la forma del terreno con un alto grado de certeza, lo que permitió crear una base de datos para el entrenamiento de la red neuronal ANN2_64 alcanzando una precisión del 98% en la clasificación del terreno.

En cuanto a la medición de la altura, la utilización del modelo de una cámara estéreo proporcionó resultados con una precisión mejor a la esperada, teniendo un RMSPE de 2.54%, en comparación al valor teórico que estaba entre el 3.5% y 5%, esto valida la efectividad del modelo matemático empleado y refuerza la viabilidad del algoritmo para medir la altura del terreno bajo condiciones reales de vuelo.

La planificación de los vuelos fotogramétricos demostró ser efectiva, adaptándose adecuadamente a las características de cada tipo de terreno. A pesar de las desviaciones observadas en terrenos como "Ladera 5", que presentaba características mixtas de ladera y valle, el sistema logró adaptarse y seguir en gran medida las curvas de nivel. Se consiguió una apreciable mejora en el RMSPE del GSD para los terrenos de ladera, reduciendo los valores que inicialmente oscilaban entre el 26% y el 89%, a un rango más ajustado del 7% al 17%. Para los terrenos de valle, también se consiguió mejorar el RMSPE, pasando de un rango entre el 13% y el 38%, a valores más reducidos que oscilan entre el 10% y el 20%. Aunque se encontraron desviaciones en el GSD, la mejora en la precisión y la menor dispersión de los datos en comparación con la línea base de vuelos a altura constante valida la efectividad del algoritmo desarrollado.

Las pruebas de *hardware-in-the-loop* realizadas con el controlador Pixhawk 2 y la Nvidia Jetson TX2, registraron tiempos de ejecución de tres minutos y medio para praderas, y seis minutos y veintiún segundos para laderas y valles. Estos resultados destacan la viabilidad del sistema para ser adaptado y escalado en futuras aplicaciones de robótica y vuelos con UAV. Además, marcan un avance hacia la automatización completa de la planificación de vuelos en diversos terrenos montañosos.

6.2. Trabajos futuros.

En la búsqueda de expandir las capacidades del algoritmo para adaptarse a una variedad más amplia de topografías, se propone incorporar el análisis de terrenos con características mixtas y con mayor número de inclinaciones. Para mejorar la precisión en la clasificación de estos terrenos mixtos, se propone desarrollar módulos que empleen técnicas avanzadas de aprendizaje profundo, tales como redes neuronales convolucionales o redes adversarias generativas, que puedan identificar las diversas pendientes que pueden aparecer en el terreno.

Además, para enfrentar el desafío de terrenos variados, una segmentación de una zona de vuelo en varios tipos de terreno, mediante un procesamiento de la nube de puntos diferente. Esta mejora permitiría hacer una planeación de la ruta de vuelo de una forma más compleja, resultando en mediciones más precisas.

También se puede hacer un análisis del funcionamiento en un entorno 100% real en un terreno de cultivo, con el fin de hacer un análisis en el uso de la batería y su durabilidad en la planificación de la ruta de vuelo.

Otra mejora es adaptar diferentes formas geométricas para la delimitación de terrenos, mejorando la forma en la que se genera la planeación de la ruta de vuelo. para esto se puede implementar un módulo que permita a los usuarios seleccionar o diseñar la forma de la cerca según las necesidades específicas del terreno, esto aumentaría la flexibilidad en la cobertura del área de interés.

Finalmente, para abordar la necesidad de mediciones de altura más precisas y en tiempo real, se sugiere explorar el uso de versiones futuras de modelos como MIDAS. Estos modelos avanzados permitirían hacer correcciones de altura en tiempo real con una mayor precisión. La integración de esta tecnología no solo mejoraría la exactitud de los datos recogidos, sino que también haría el sistema más robusto y adaptable a diferentes escenarios operativos.

Referencias Bibliográficas

- Blackmore, S. (1994). Precision Farming: An Introduction. *Outlook on Agriculture*, 23(4), 275–280. <https://doi.org/10.1177/003072709402300407> Recuperado el 25 de Julio del 2021
- Breiman, L., & Friedman, J. H. (2002). Predicting multivariate responses in multiple linear regression. *Journal of the Royal Statistical Society: Series B (Methodological)*, 59(1), 3-54. <https://doi.org/10.1111/1467-9868.00054>
- Dorta-González, P. (2023). A multiple linear regression analysis to measure the journal contribution to the social attention of research. *Axioms*, 12(4), Article 337. <https://doi.org/10.3390/axioms12040337>
- Gargari, A. M., Ebadi, H., & Esmaeili, F., et al. (2022). Dynamic 3D Network Design for UAV Based Photogrammetry in Mountainous Terrain. *Research Square*. <https://doi.org/10.21203/rs.3.rs-2336499/v1>
- Hsieh, C.-S., Hsiao, D.-H., & Lin, D.-Y. (2023). Contour Mission Flight Planning of UAV for Photogrammetric in Hillside Areas. *Applied Sciences*, 13(13), 7666. <https://doi.org/10.3390/app13137666>
- Jin, X., Li, Z., Opinat Ikiela, N. V., He, X., Wang, Z., Tao, Y., & Lv, H. (2024). An Efficient Trajectory Planning Approach for Autonomous Ground Vehicles Using Improved Artificial Potential Field. *Symmetry*, 16(1), Article 106. <https://doi.org/10.3390/sym16010106>
- Karunanayake, N., & Gnanasekera, M. (2016). An enhanced computer vision based hand movement capturing system with stereo vision. *International Journal of Computer Science and Security*, 10, 1-7.

- Kozmus Trajkovski, K., Grigillo, D., & Petrovič, D. (2020). Optimization of UAV Flight Missions in Steep Terrain. *Remote Sensing*, 12(8), 1293. <https://doi.org/10.3390/rs12081293>
- Li, H., Ye, W., Liu, J., Tan, W., Pirasteh, S., Fatholahi, S. N., & Li, J. (2021). High-Resolution Terrain Modeling Using Airborne LiDAR Data with Transfer Learning. *Remote Sensing*, 13(17), 3448. <https://doi.org/10.3390/rs13173448>
- Li, Z., Wu, X., Zhang, W., & Yu, W. (2024). Obstacle Avoidance Trajectory Planning for Autonomous Vehicles on Urban Roads Based on Gaussian Pseudo-Spectral Method. *World Electric Vehicle Journal*, 15(1), Article 7. <https://doi.org/10.3390/wevj15010007>
- Mike1024. (2010, febrero 18). ECEF ENU Longitude Latitude relationships [Imagen digital]. Basada en EarthTangentialPlane.png por Usuario:Raffyl99. Dominio público. Recuperado el 22 de abril de 2024, de https://en.wikipedia.org/wiki/Local_tangent_plane_coordinates#/media/File:ECEF_ENU_Longitude_Latitude_relationships.svg
- Mir, I., Gul, F., Mir, S., Khan, M. A., Saeed, N., Abualigah, L., Abuhaija, B., & Gandomi, A. H. (2022). A Survey of Trajectory Planning Techniques for Autonomous Systems. *Electronics*, 11(18), Article 2801. <https://doi.org/10.3390/electronics11182801>
- Nebel, S., Beege, M., Schneider, S., & Rey, G. D. (2020). A review of photogrammetry and photorealistic 3D models in education from a psychological perspective. *Frontiers in Education*, 5. <https://doi.org/10.3389/feduc.2020.00144>
- OpenCV. (2021). Camera Calibration and 3D Reconstruction. Recuperado de https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html

- Pachas, R. (2009). El Levantamiento Topográfico: Uso Del GPS Y Estación Total. *Academia*, 8(16), 29-45. Recuperado el 28 de mayo del 2021.
- Palomino, D. (2020). Supervisión Autónoma De Cultivos Agrícolas Por Medio De Vehículos Aéreos No Tripulados (UAV) Usando Técnicas De Procesamiento De Imágenes Y Fotogrametría [Tesis de Maestría, Escuela Colombiana de Ingeniería Julio Garavito]. <https://repositorio.escuelaing.edu.co/handle/001/1130>. Recuperado el 18 de mayo del 2021
- Pérez, J. (2001). Apuntes De Fotogrametría III. Universidad De Extremadura Centro Universitario de Mérida, Obtenido de <http://redgeomatica.rediris.es/cartoprofesores/Fotogrametria/ApuntesFotogrametria3.pdf>. Recuperado el 20 de mayo del 2021.
- Puce, S., Riccardi, A., & Cerrano, C. (2023). Photogrammetry, from the land to the sea and beyond: A unifying approach to study terrestrial and marine environments. *Journal of Marine Science and Engineering*, 11(4), 759. <https://doi.org/10.3390/jmse11040759>
- PX4. (2024). Hardware in the Loop Simulation (HITL). Recuperado el 16 de mayo de 2024, de <https://docs.px4.io/main/en/simulation/hitl.html#hitl-vs-sitl>
- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., ... & Ng, A. Y. (2009, May). ROS: an open-source Robot Operating System. In ICRA workshop on open source software (Vol. 3, No. 3.2, p. 5). Recuperado el 25 de julio del 2021.
- Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., & Koltun, V. (2022). Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3), 1623-1637. <https://doi.org/10.1109/TPAMI.2020.3019967>

- Riaño, E.(2018). Metodología Para El Diseño De Un Vuelo Fotogramétrico Usando UAV´ S [Bachelor's Thesis, Universidad Distrital Francisco José De Caldas]. <http://hdl.handle.net/11349/13097>. Recuperado el 10 de mayo del 2021.
- Ryan, WBF, SM Carbotte, J. Coplan, S. O'Hara, A. Melkonian, R. Arko, RA Weissel, V. Ferrini, A. Goodwillie, F. Nitsche, J. Bonczkowski y R. Zemsky (2009), Conjunto de datos de síntesis de topografía global de resolución múltiple (GMRT), *Geochem. Geofís. Geosistema.*, 10, Q03014, <https://doi.org/10.1029/2008GC002332>
- School of Geomatics, Liaoning Technical University, et al. (2022). Camera auto-calibration in GPS/INS/stereo camera integrated kinematic positioning and navigation system. *Journal of Global Positioning Systems*.
- Singh, C. H., Jain, K., & Mishra, V. (2023). UAV-Based Terrain-Following Mapping Using LiDAR in High-Relief Areas. In K. Jain, V. Mishra, & B. Pradhan (Eds.), *Proceedings of UASG 2021: Wings 4 Sustainability* (pp. 1-15). Springer, Cham. https://doi.org/10.1007/978-3-031-19309-5_3
- Srinivasu, P. N., SivaSai, J. G., Ijaz, M. F., Bhoi, A. K., Kim, W., & Kang, J. J. (2021). Classification of Skin Disease Using Deep Learning Neural Networks with MobileNet V2 and LSTM. *Sensors*, 21(8), 2852. <https://doi.org/10.3390/s21082852>
- Taye, M. M. (2023). Understanding of Machine Learning with Deep Learning: Architectures, Workflow, Applications and Future Directions. *Computers*, 12(5), 91. <https://doi.org/10.3390/computers12050091>
- Taylor, J. R. (1997). *An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements*. University Science Books. ISBN: 9780935702422.
- TensorFlow. (2024). Keras layers - API documentation. Recuperado el 14 de mayo de 2024, de https://www.tensorflow.org/api_docs/python/tf/keras/layers

- Walter, T., & Blanch, J. (2017). Improved user position monitor for WAAS. *Navigation*, 64(1), 165-175. <https://doi.org/10.1002/navi.180>
- Walter, T., Gunning, K., Phelts, R. E., & Blanch, J. (2018). Validation of the Unfaulted Error Bounds for ARAIM. *Navigation*, 65(1), 117-133. <https://doi.org/10.1002/navi.214>
- Wei, et al. (2023). A Systematic Stereo Camera Calibration Strategy: Leveraging Latin Hypercube Sampling and 2k Full-Factorial Design of Experiment Methods. *Sensors*.
- Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11), 1330-1334
- Zhao, H., & Xu, A. (2023). A Stereo Calibration Method of Multi-Camera Based on Circular Calibration Board. *Electronics*.

Anexo 1. Instalación del entorno de desarrollo.

El entorno de desarrollo para la simulación de vuelo de UAVs se basa en repositorio PX4-Autopilot. Este marco de código abierto soporta la simulación de controladores de vuelo Pixhawk, incluye modelos de diversos robots y proporciona soporte para las comunicaciones *hardware-in-the-loop*. La flexibilidad y precisión de PX4-Autopilot facilitan las pruebas de desarrollo y aseguran que las simulaciones reflejen condiciones reales lo más fielmente posible.

El proyecto está diseñado como un paquete del sistema de *Robot Operating System* (ROS). Los archivos del proyecto están disponibles en GitHub en el siguiente repositorio: <https://github.com/andres14077/offbnode.git>

Para los usuarios de Linux, la ejecución del proyecto a través de Docker es una opción recomendada por su compatibilidad y capacidad para manejar dependencias sin alterar el sistema operativo anfitrión. Después de clonar el repositorio, se debe acceder a la carpeta del proyecto y ejecutar:

```
docker-compose up --build -d
```

Luego, para configurar el entorno de simulación de Gazebo, se debe ejecutar el siguiente comando dentro de la carpeta ***offbnode/source***:

```
source gazebo_env.sh
```

Es importante destacar que, aunque Docker facilita la configuración inicial, tiene la limitación de no permitir el uso de sensores de cámara de manera interna. Esto significa que la simulación estará restringida a pruebas de movimiento y visualización de modelos, sin capacidad para interactuar con entradas de video.

Para aquellos que requieren un uso completo de todas las capacidades del proyecto, incluyendo sensores de cámara, la instalación local sobre Ubuntu 18.04 LTS es la mejor opción. Este proceso inicia con la clonación del repositorio en el directorio personal del usuario y la ejecución del script de instalación:

`./Instalacion.sh`

Este script se encuentra en **offbnode/source** y configura automáticamente todo lo necesario, como ROS, Gazebo, y diversas librerías de Python y OpenCV. Tras completar la instalación, es de verificar que todo funcione correctamente. Esto se puede hacer ejecutando:

```
roslaunch offbnode simulacion.launch gui:=true rviz:=true
```

Este comando inicia las interfaces de Gazebo y Rviz, permitiendo la visualización en tiempo real de los datos y el procesamiento efectuado durante la simulación. Como se observa en la Figura 44.

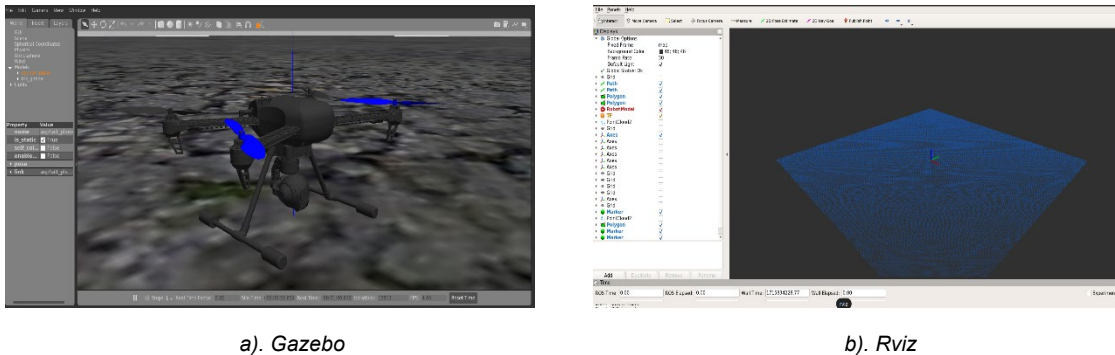


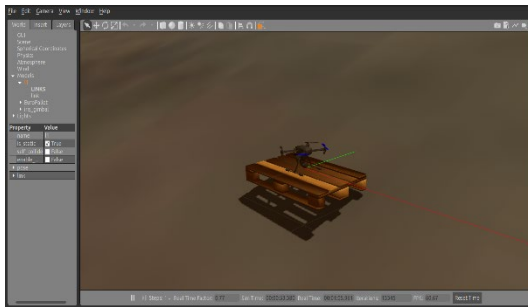
Figura 44. Entorno de simulación gazebo y rviz.

Después de configurar y verificar la instalación básica del proyecto, es importante familiarizarse con las herramientas adicionales que permiten un control más detallado y una personalización del entorno de simulación. Estas herramientas incluyen la carga de diferentes terrenos, la modificación de parámetros de vuelo y cámaras, y la ejecución de algoritmos específicos.

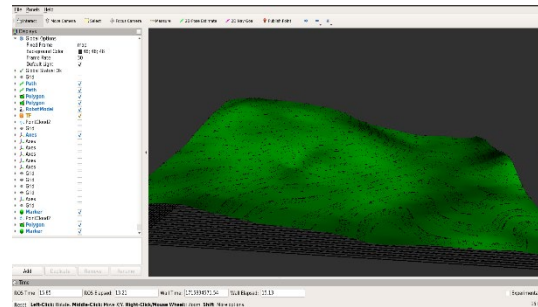
Para cargar diferentes terrenos y ajustar la configuración del entorno de simulación, se pueden usar variaciones en el comando de lanzamiento. Por ejemplo, para iniciar la simulación con un terreno específico de ladera 1, se utilizaría el comando:

```
roslaunch offbnode simulacion.launch gui:=true rviz:=true mundo:=1
```

Este comando permite al usuario especificar el tipo de terreno sobre el que operará el UAV, donde **“mundo:=1”** indica el modelo de ladera 1, como se ve en Figura 45. Los usuarios pueden personalizar este parámetro para cambiar entre praderas, laderas y valles, facilitando pruebas en diversos entornos de vuelo.



a). Gazebo



b). Rviz

Figura 45. Entorno de simulación gazebo y rviz para terreno ladera 1.

Además, se puede ajustar los parámetros de vuelo fotogramétrico y las características de la cámara. Esto se puede realizar mediante la herramienta **rqt_reconfigure**, que ofrece una interfaz gráfica para modificar estos parámetros en tiempo real:

```
roslaunch rqt_reconfigure rqt_reconfigure
```

Utilizando esta herramienta que se ve en la Figura 46, los usuarios pueden modificar ajustes como la altura de vuelo, las especificaciones de la cámara y otros parámetros importantes que afectan la planeación de la ruta de vuelo.

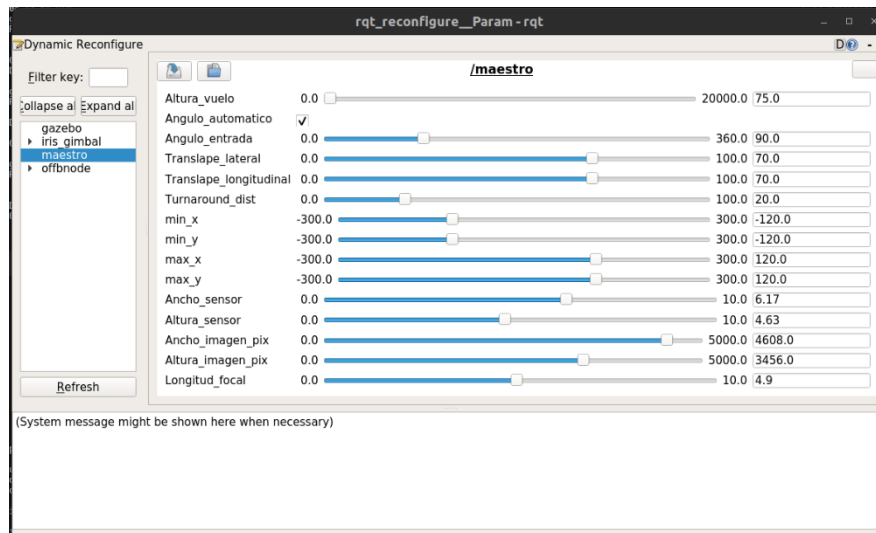


Figura 46. Interfaz para modificación de parámetros de rqt reconfigure

Finalmente, para iniciar la toma de datos o ejecutar el algoritmo diseñado, se utiliza el siguiente comando de servicio de ROS:

```
rosservice call /offbnode/iniciar_toma "{}"
```

Este comando inicia el vuelo de reconocimiento que clasifica y mide el terreno, genera una planeación de vuelo según los parámetros objetivo, y finalmente hace el seguimiento de la ruta del vuelo fotogramétrico para terrenos inclinados.

Anexo 2. Selección del vehículo y adaptaciones realizadas

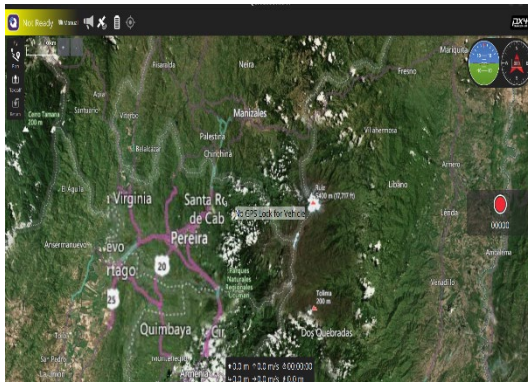
Para el desarrollo de este proyecto, se seleccionó un UAV modelo Iris, compatible con las plataformas PX4 y Ardupilot. Esta elección se fundamentó en las capacidades de adaptación y compatibilidad del drone con diversos componentes y sensores.

La principal modificación realizada al UAV fue la incorporación de una cámara gimbal, especificada para ser compatible con una cámara Parrot Sequoia RGB, como se ve en la Figura 47. Esta adaptación permitió capturar imágenes de alta calidad, necesarias para la precisión de los datos recopilados en el estudio.

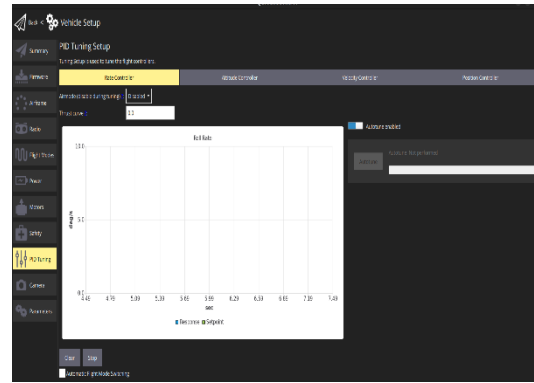


Figura 47. UAV con carga útil de cámara gimbal usado en el proyecto.

Con el cambio en la carga útil, se hizo necesario realizar una recalibración de los controladores internos del UAV para asegurar su estabilidad y precisión en el vuelo. Este proceso se llevó a cabo utilizando el programa QGroundControl Figura 48, que proporciona las herramientas necesarias para ajustar los parámetros de vuelo de acuerdo con las nuevas especificaciones y peso del UAV.



a). QGroundControl.

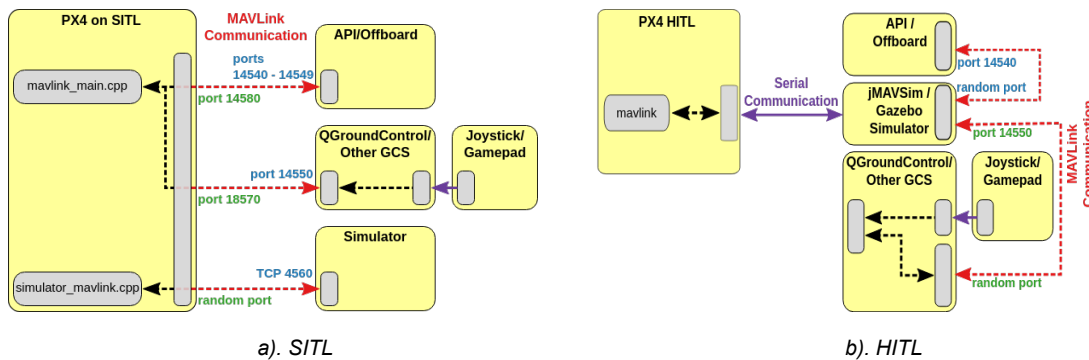


b). Interfaz de calibración controladores PID UAV

Figura 48. Interfaz en QGroundControl para calibrar controladores PID del UAV.

Una característica dinámica del proyecto fue la forma en la que simula el dron Iris, que varió según el modo de operación durante las pruebas. El diseño del proyecto permitió realizar simulaciones utilizando la Pixhawk en un entorno virtual a través del modo *Software-in-the-loop* (SITL), así como simulaciones con dispositivos reales utilizando el modo *Hardware-in-the-loop* (HITL).

En la Figura 49 se muestra una comparación de los modos operativos SITL y HITL, ilustrando las diferencias en la configuración del sistema y la interacción con el entorno de simulación o con hardware real.



a). SITL

b). HITL

Figura 49. Software in the loop (SITL) vs Hardware in the loop (HITL). Fuente: PX4. (2024).

Anexo 3. Descripción y modelado de los terrenos utilizados para las pruebas

Para la creación de entornos utilizados en Gazebo, se empleó el programa Blender en conjunto con el *addon* BlenderGIS. El proceso comenzó seleccionando áreas específicas dentro de la cordillera central de los andes en Colombia, particularmente zonas de cultivo que cumplieran con las características topográficas deseadas, clasificadas en tres tipos: praderas, laderas y valles. Estas categorías fueron elegidas para estudiar los efectos de terrenos sin inclinación, con una inclinación y con dos inclinaciones pronunciadas.

Utilizando Google Maps 3D, se buscó e identificó la región de interés. Posteriormente, en Blender y mediante el uso del *addon* BlenderGIS, se importó el terreno junto con su textura. También se obtuvieron los datos de elevación proporcionados por el US Marine Geoscience Data System. Una vez importada la topografía, se realizaron ajustes de escala para asegurar que los modelos fueran adecuados para las simulaciones. El último paso consistió en exportar estos modelos en formato DAE, lo cual facilitó su incorporación como modelos SDF en Gazebo.

En total, se modelaron 15 terrenos distintos, distribuidos equitativamente con cinco ejemplos de cada tipo de terreno. Este enfoque permite una evaluación en los diferentes tipos de topográficas evaluadas en el proyecto, proporcionando una base sólida para pruebas y desarrollos futuros.

Estos modelos de terrenos quedan a disposición en el siguiente enlace <https://drive.google.com/uc?export=download&id=1Jek71VO0AhEsg4dGM9NalkFbJyFQfCmB&confirm=t>

Ladera 1

Es un terreno de 1050X1950 metros de extensión, tiene una inclinación que va de sureste a noroeste, con una ganancia de 400 metros de altura de punta a punta, es un terreno de cultivo

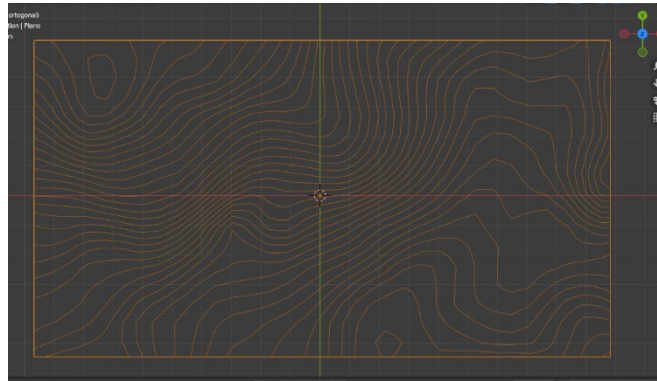
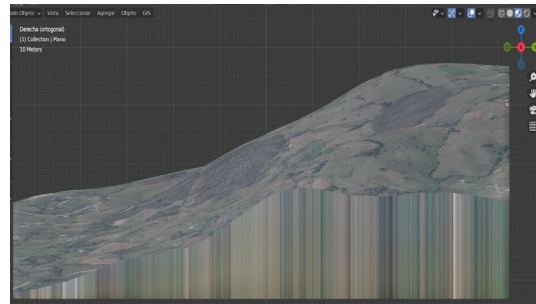


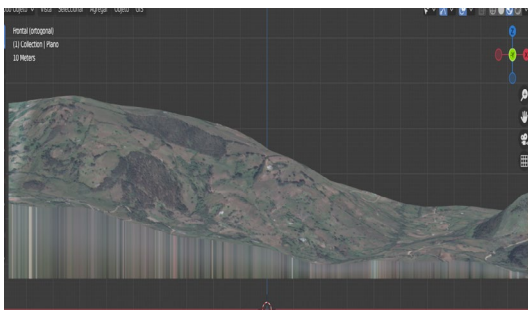
Figura 50. Curvas de nivel con resolución de 10 metros para el terreno ladera 1.



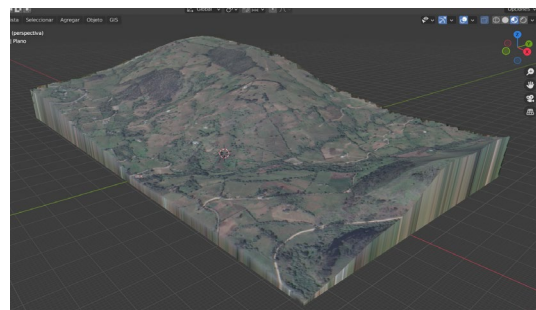
a). Vista superior



b). Vista frontal



c). Vista lateral derecha



d). Vista Isométrica

Figura 51. Vista de superior, frontal, lateral derecha e isométrica del terreno ladera 1.

Ladera 2

Es un terreno de 1065X1950 metros de extensión, tiene una inclinación que va de suroeste a noreste, con una ganancia de 600 metros de altura de punta a punta, es un terreno de cultivo.

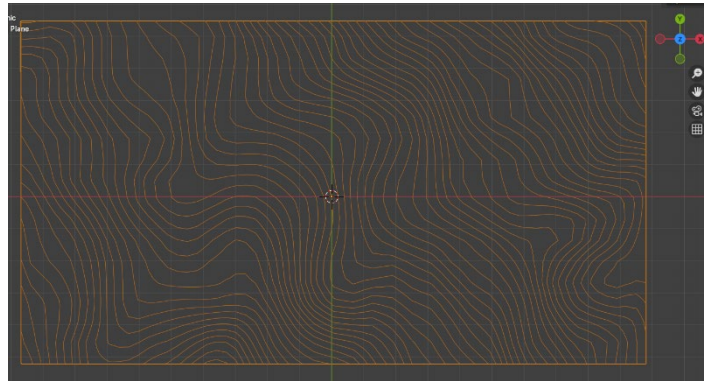
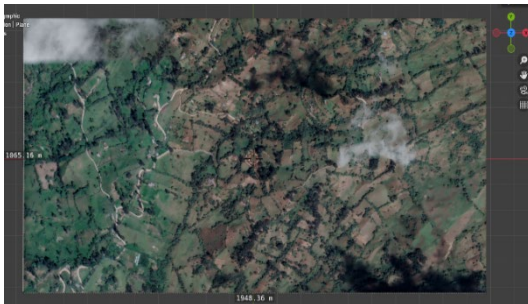
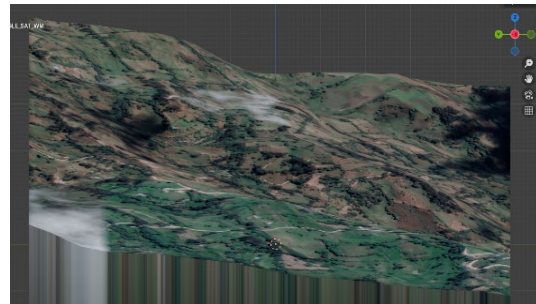


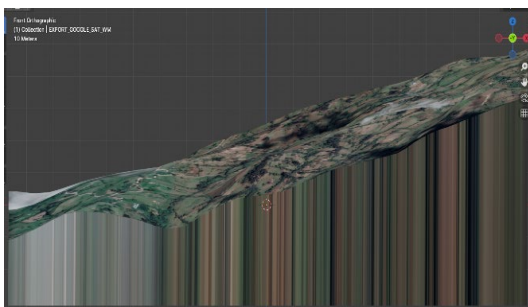
Figura 52. Curvas de nivel con resolución de 10 metros para el terreno ladera 2.



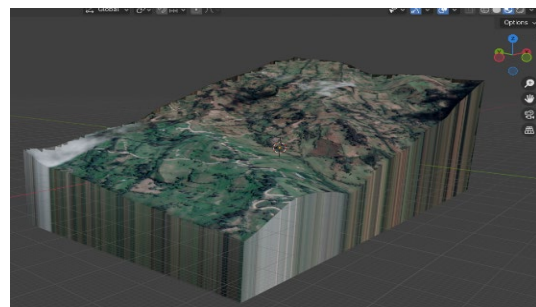
a). Vista superior



b). Vista posterior



c). Vista lateral derecha



d). Vista Isométrica

Figura 53. Vista superior, posterior, lateral derecha e isométrica del terreno ladera 2.

Ladera 3

Es un terreno de 1066X1950 metros de extensión, tiene una inclinación que va de sureste a noroeste, con una ganancia de 640 metros de altura de punta a punta, es un terreno de cultivo

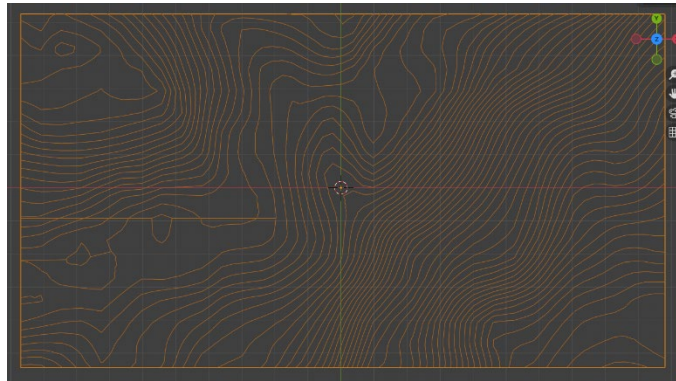
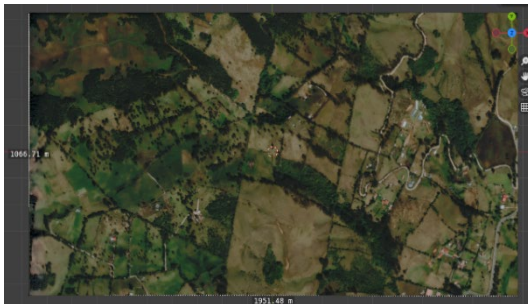
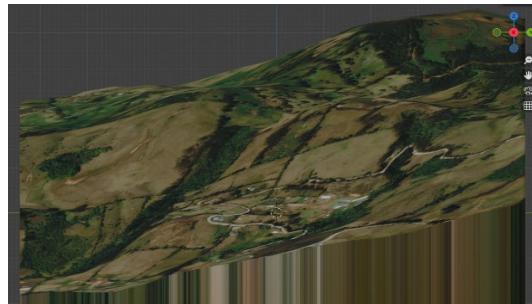


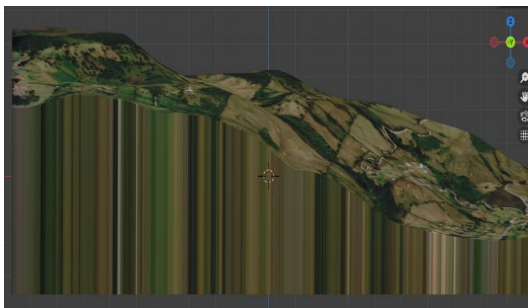
Figura 54. Curvas de nivel con resolución de 10 metros para el terreno ladera 3.



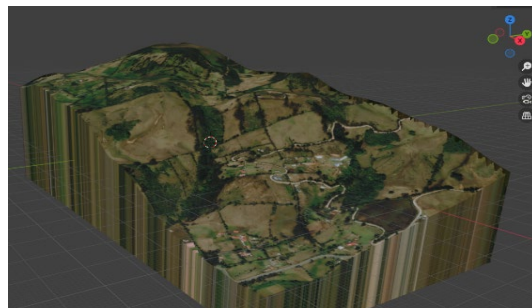
a). Vista superior



b). Vista frontal



c). Vista lateral derecho



d). Vista Isométrica

Figura 55. Vista superior, frontal, lateral derecha e isométrica del terreno ladera 3.

Ladera 4

Es un terreno de 1066X1950 metros de extensión, tiene una inclinación que va de oeste a este, con una ganancia de 525 metros de altura de punta a punta, es un terreno de cultivo con amplias zonas boscosas

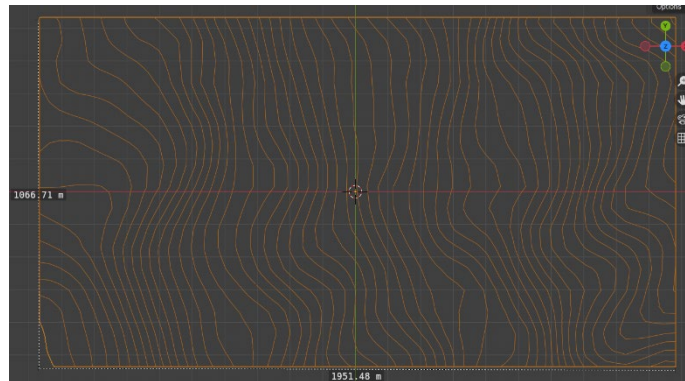
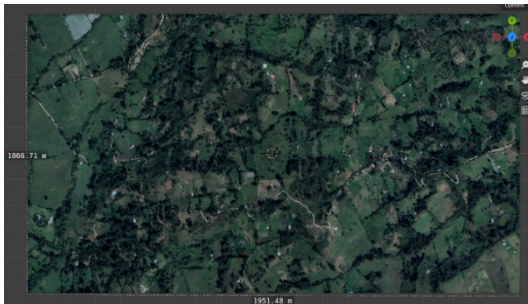
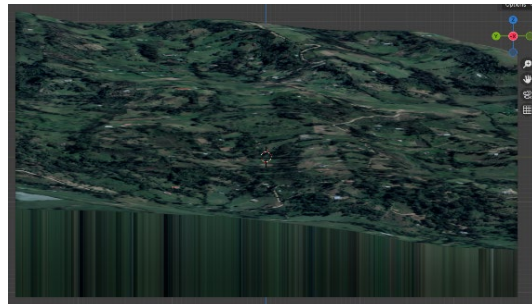


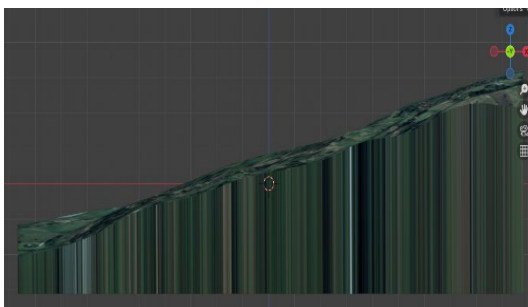
Figura 56. Curvas de nivel con resolución de 10 metros para el terreno ladera 4.



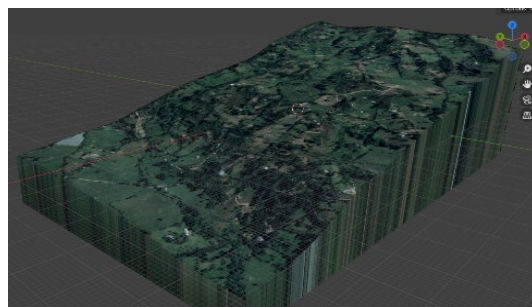
a). Vista superior



b). Vista posterior



c). Vista lateral derecha



d). Vista Isométrica

Figura 57. Vista superior, posterior, lateral derecha e isométrica del terreno ladera 4.

Ladera 5

Es un terreno de 1100X1950 metros de extensión, tiene una inclinación que va de norte a sur, con una ganancia de 290 metros de altura de punta a punta, es un terreno de cultivo con algunas zonas boscosas

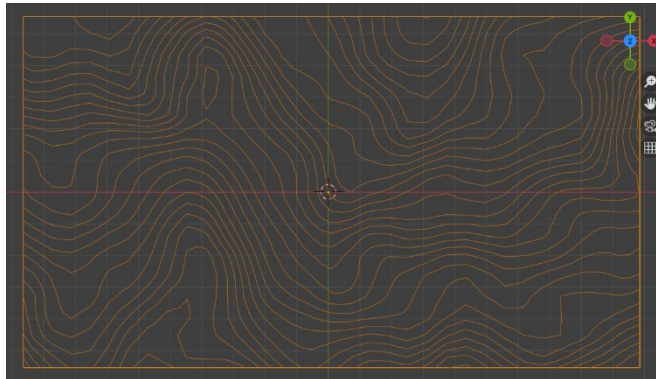
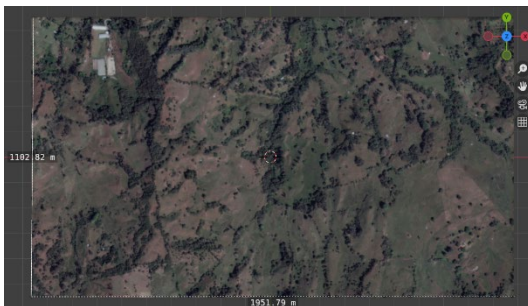
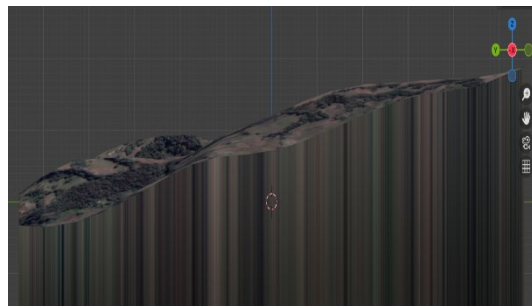


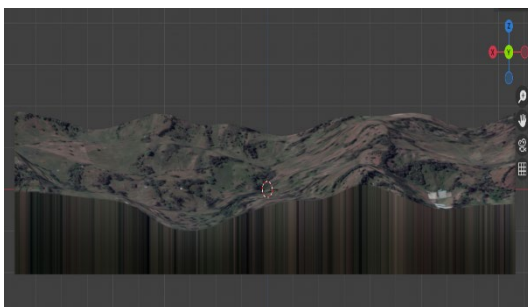
Figura 58. Curvas de nivel con resolución de 10 metros para el terreno ladera 5.



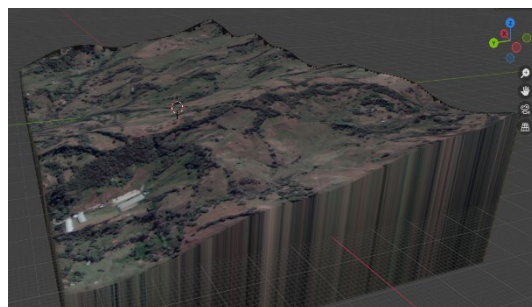
a). Vista superior



b). Vista posterior



c). Vista lateral izquierda



d). Vista Isométrica

Figura 59. Vista superior, posterior, lateral izquierda e isométrica del terreno ladera 4.

Pradera 1

Es un terreno de 550X950 metros de extensión, con una inclinación de extremo a extremo de 50 metros, es un terreno de cultivo

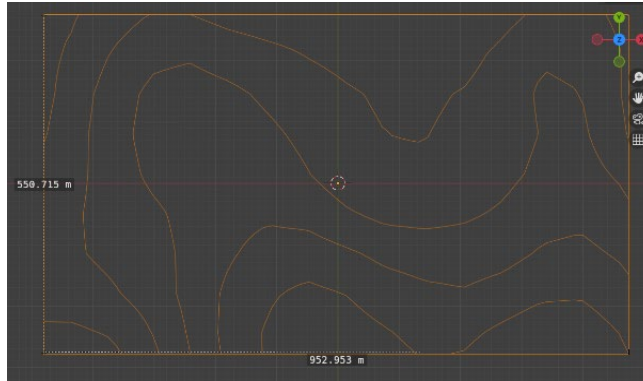
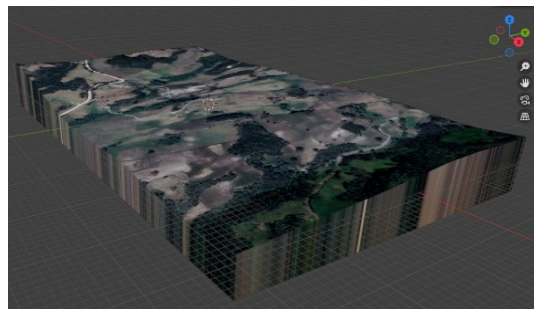


Figura 60. Curvas de nivel con resolución de 10 metros para el terreno pradera 1.



a). Vista superior



b). Vista isométrica

Figura 61. Vista superior e isométrica del terreno pradera 1.

Pradera 2

Es un terreno de 600X1066 metros de extensión sin inclinaciones , es un terreno de cultivo

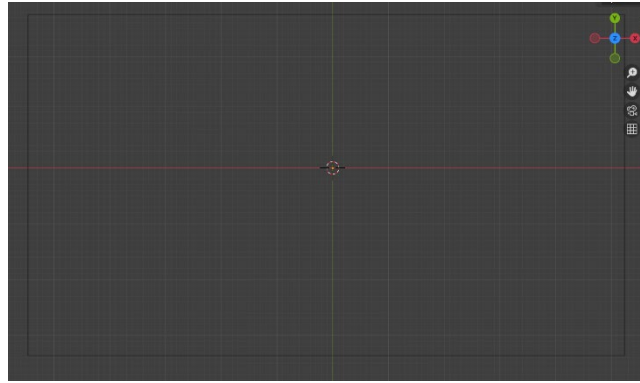
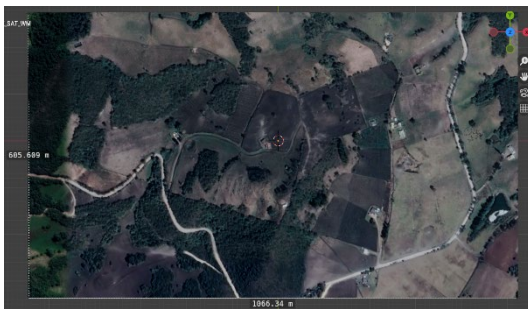
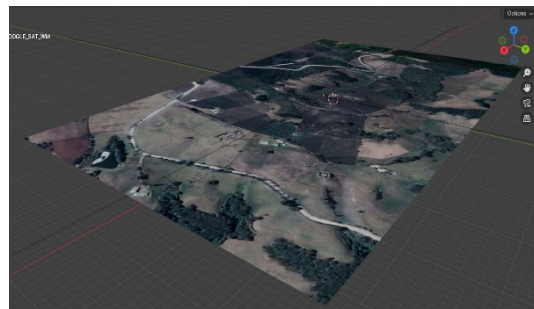


Figura 62. Curvas de nivel con resolución de 10 metros para el terreno pradera 2.



a). Vista superior



b). Vista isométrica

Figura 63. Vista superior e isométrica del terreno pradera 2.

Pradera 3

Es un terreno de 1100X1950 metros de extensión sin inclinaciones, es un terreno de cultivo

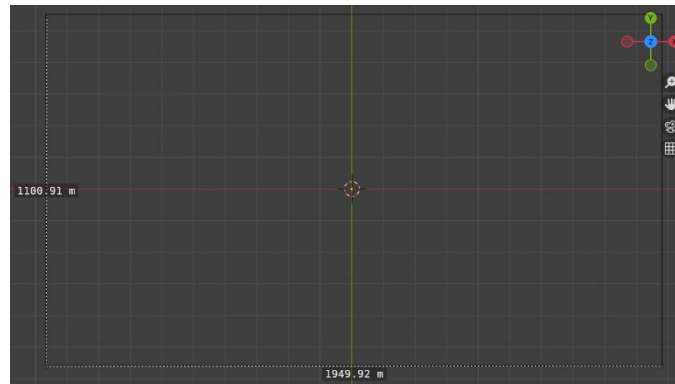
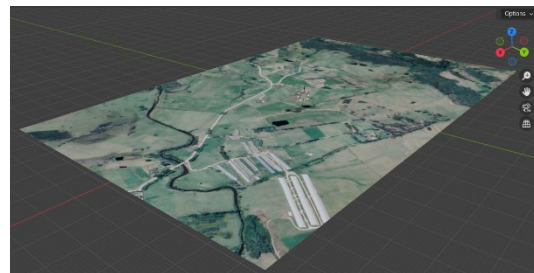


Figura 64. Curvas de nivel con resolución de 10 metros para el terreno pradera 3.



a). Vista superior



b). Vista isométrica

Figura 65. Vista superior e isométrica del terreno pradera 3.

Pradera 4

Es un terreno de 1065X1900 metros de extensión con una inclinación de extremo a extremo de 50 metros , es un terreno de cultivo

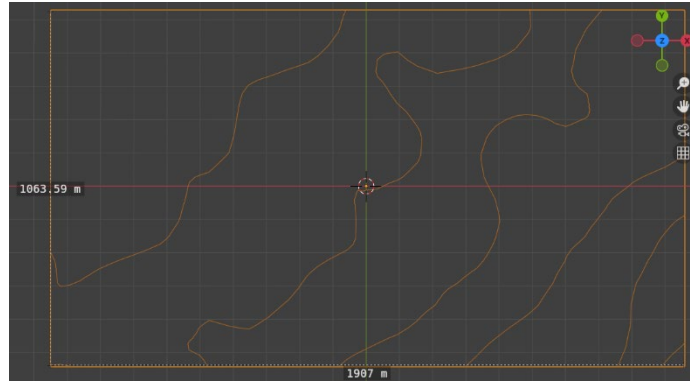
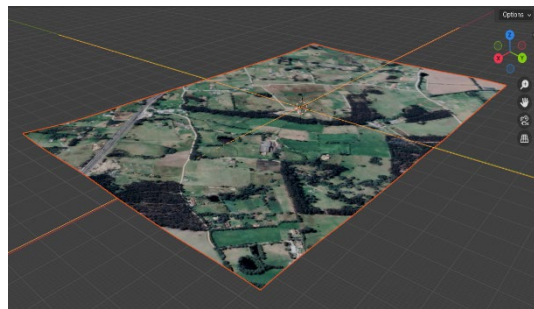


Figura 66. Curvas de nivel con resolución de 10 metros para el terreno pradera 4.



a). Vista superior



b). Vista isométrica

Figura 67. Vista superior e isométrica del terreno pradera 4.

Pradera 5

Es un terreno de 1100X1950 metros de extensión con varias irregularidades en el terreno que van hasta los 10 metros de altura, es un terreno de cultivo.

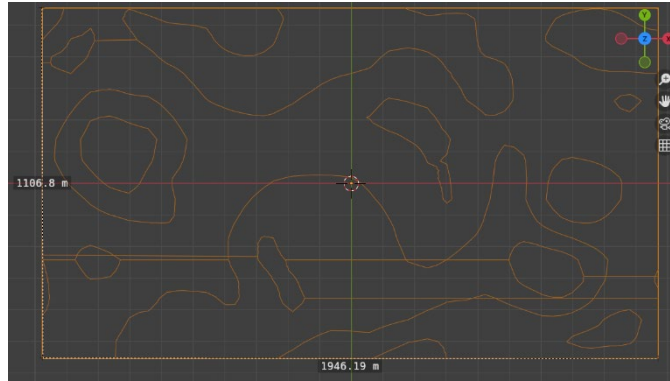
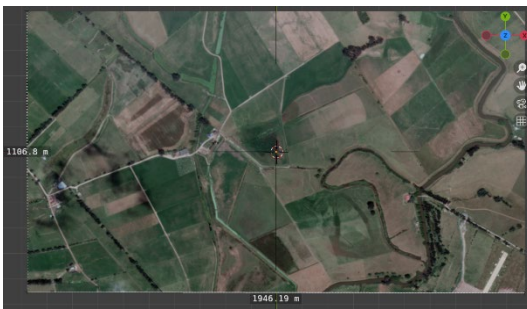
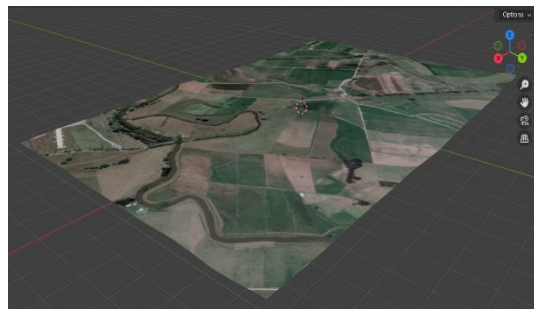


Figura 68. Curvas de nivel con resolución de 10 metros para el terreno pradera 5.



a). Vista superior



b). Vista isométrica

Figura 69. Vista superior e isométrica del terreno pradera 5.

Valle 1

Es un terreno de 540X950 metros de extensión, en medio se encuentra un río que atraviesa el terreno de extremo a extremo, Este valle tiene una profundidad de 390 metros.

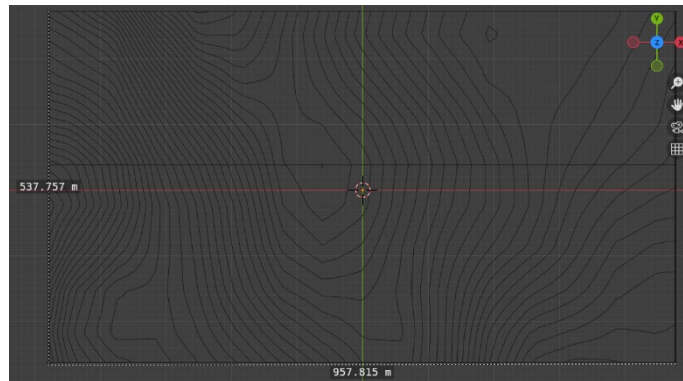
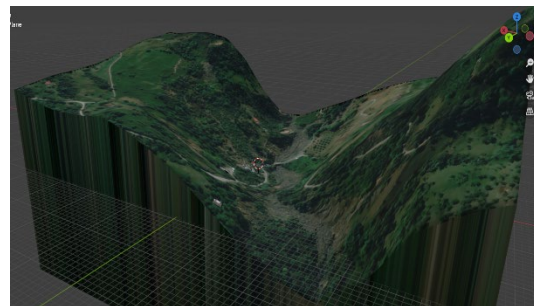


Figura 70. Curvas de nivel con resolución de 10 metros para el terreno valle 1.



a). Vista superior



b). Vista isométrica

Figura 71. Vista superior e isométrica del terreno valle 1.

Valle 2

Es un terreno de 1100X1910 metros de extensión, en medio se encuentra un río que atraviesa el terreno de extremo a extremo, Este valle tiene una profundidad de 750 metros.

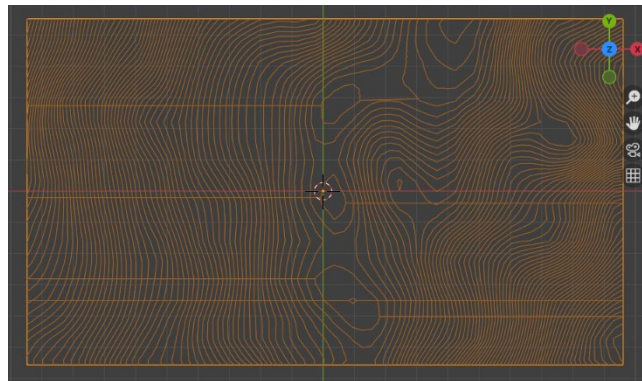
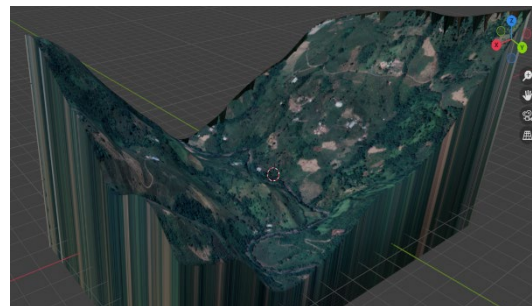


Figura 72. Curvas de nivel con resolución de 10 metros para el terreno valle 2.



a). Vista superior



b). Vista isométrica

Figura 73. Vista superior e isométrica del terreno valle 2.

Valle 3

Es un terreno de 1100X1910 metros de extensión, en medio se encuentra un río que atraviesa el terreno de extremo a extremo, Este valle tiene una profundidad de 330 metros.

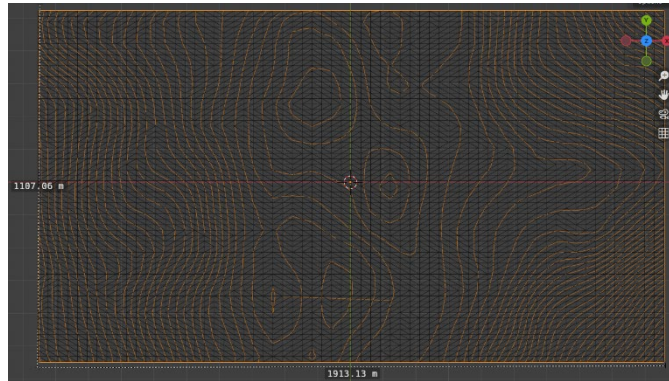
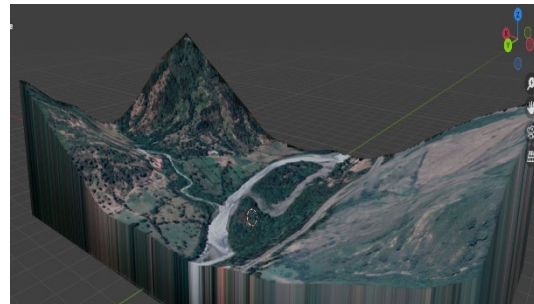


Figura 74. Curvas de nivel con resolución de 10 metros para el terreno valle 3.



a). Vista superior



b). Vista isométrica

Figura 75. Vista superior e isométrica del terreno valle .

Valle 4

Es un terreno de 2140X3920 metros de extensión, en medio se encuentra un río que atraviesa el terreno de extremo a extremo, Este valle tiene una profundidad de 940 metros.

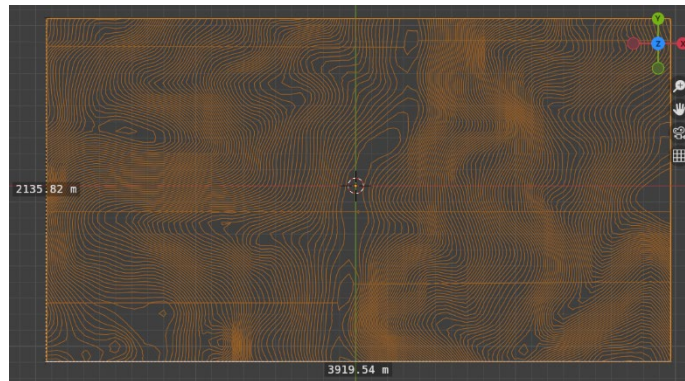
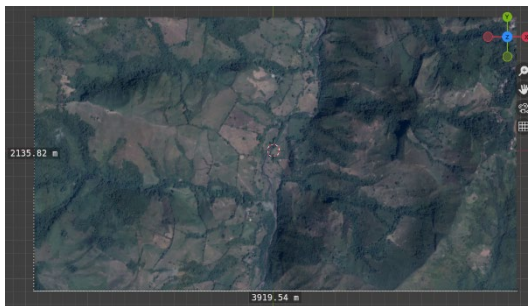
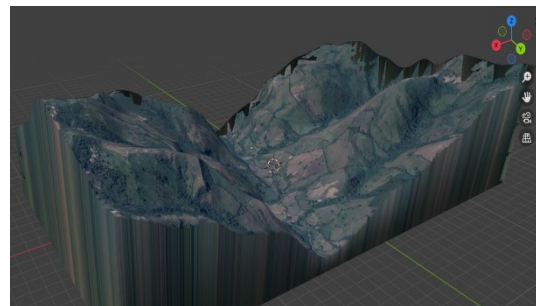


Figura 76. Curvas de nivel con resolución de 10 metros para el terreno valle 4.



a). Vista superior



b). Vista isométrica

Figura 77. Vista superior e isométrica del terreno valle 4.

Valle 5

Es un terreno de 2140X3920 metros de extensión, en medio se encuentra un río que atraviesa el terreno de extremo a extremo, Este valle tiene una profundidad de 920 metros.

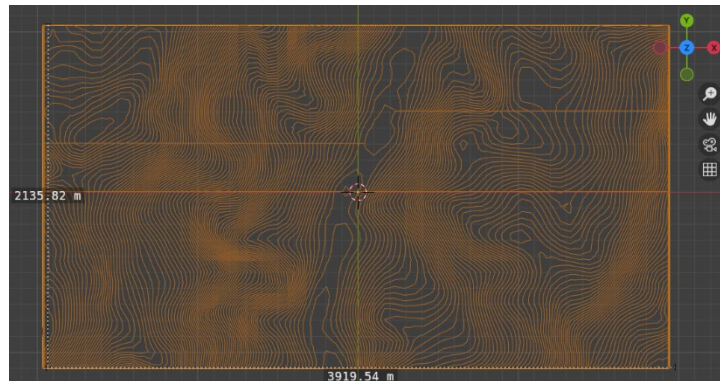
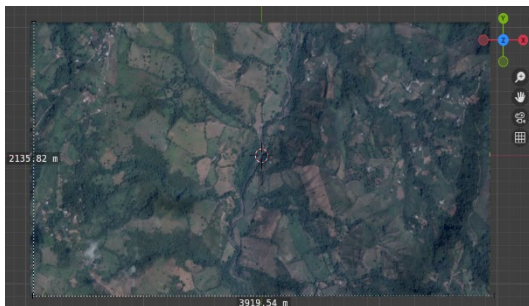
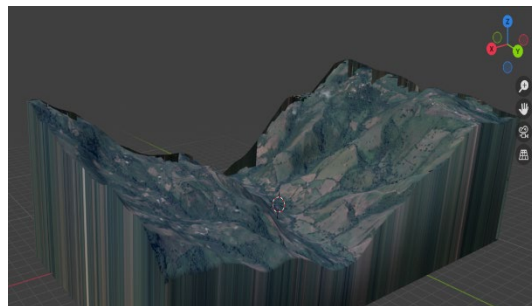


Figura 78. Curvas de nivel con resolución de 10 metros para el terreno valle 5.



a). Vista superior



b). Vista isométrica

Figura 79. Vista superior e isométrica del terreno valle 5.

Anexo 4. Evaluación de métodos de clasificación tradicionales

En este anexo se realizó un análisis comparativo de métodos de clasificación tradicionales como *Support Vector Machine* (SVM), **Random Forest**, **Naive Bayes** y el algoritmo de k vecinos más cercanos (KNN). Aunque estos métodos son comúnmente empleados en diversas aplicaciones de aprendizaje automático, se determinó que no eran adecuados para este proyecto específico debido a varias razones.

Primero, la base de datos presenta una complejidad media y características de datos que no son linealmente independientes. Específicamente, métodos como *Naive Bayes* asumen independencia entre los atributos, una suposición que el conjunto de datos no cumple. Aunque los modelos SVM con *kernel*s polinomiales y RBF demostraron una capacidad para manejar las no linealidades de los datos, alcanzando una precisión del 100% en la validación, un análisis más detallado reveló limitaciones importantes en su adaptabilidad. Durante varias sesiones de entrenamiento bajo condiciones controladas, estos modelos mostraron una falta de variabilidad en sus resultados, indicando un posible sobreajuste y una baja capacidad de generalización frente a nuevos datos.

En contraste, las redes neuronales implementadas demostraron una mayor flexibilidad y capacidad para modelar y clasificar con éxito los datos a clasificar. La capacidad de las redes neuronales para aprender representaciones jerárquicas de los datos les permite capturar interacciones complejas entre atributos, algo necesario para el proyecto. Estas redes no solo ofrecieron una mejor representación de los datos, sino que también mostraron una consistencia superior en términos de precisión y validación a través de los entrenamientos, reflejando una verdadera capacidad de aprendizaje y adaptación.

Por lo tanto, a pesar de los puntos fuertes que los métodos tradicionales puedan ofrecer en escenarios más controlados o menos complejos, para los objetivos específicos de del proyecto y las características únicas del conjunto de datos, las

redes neuronales se establecieron como la herramienta más adecuada, superando los desafíos presentados por la variabilidad y la complejidad de los datos.

En la Figura 80 se muestra un análisis de cajas y bigotes de los resultados del entrenamiento en los métodos tradicionales, usando las condiciones descritas en la sección 5.2.1. Los resultados muestran que el modelo de SVM-poly, SVM-rbf son los que mejor resultado en el entrenamiento lograron obtener, también se observó que en el modelo de knn, entre más vecinos se usaba, menor era la precisión de su clasificación.

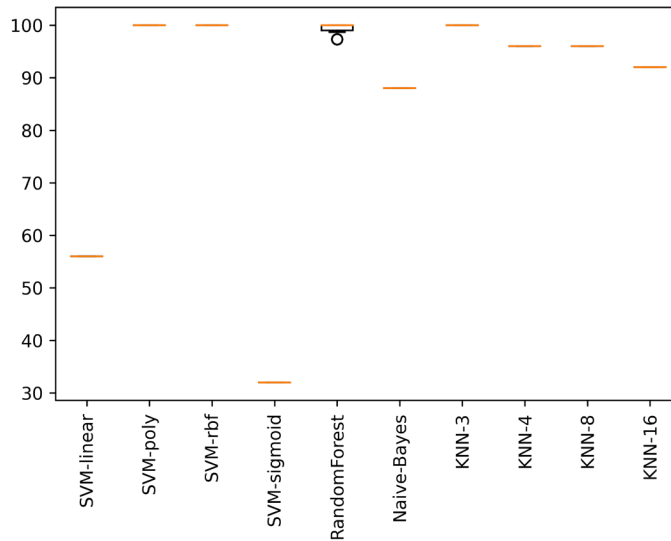


Figura 80. Diagrama de cajas y bigotes del entrenamiento de los modelos de clasificación tradicionales.

En la Figura 81, se pudo observar que entre los métodos tradicionales el mejor es el de SVM con un kernel poly, teniendo la mejor precisión de clasificación, el menor tamaño, y el menor tiempo de ejecución.

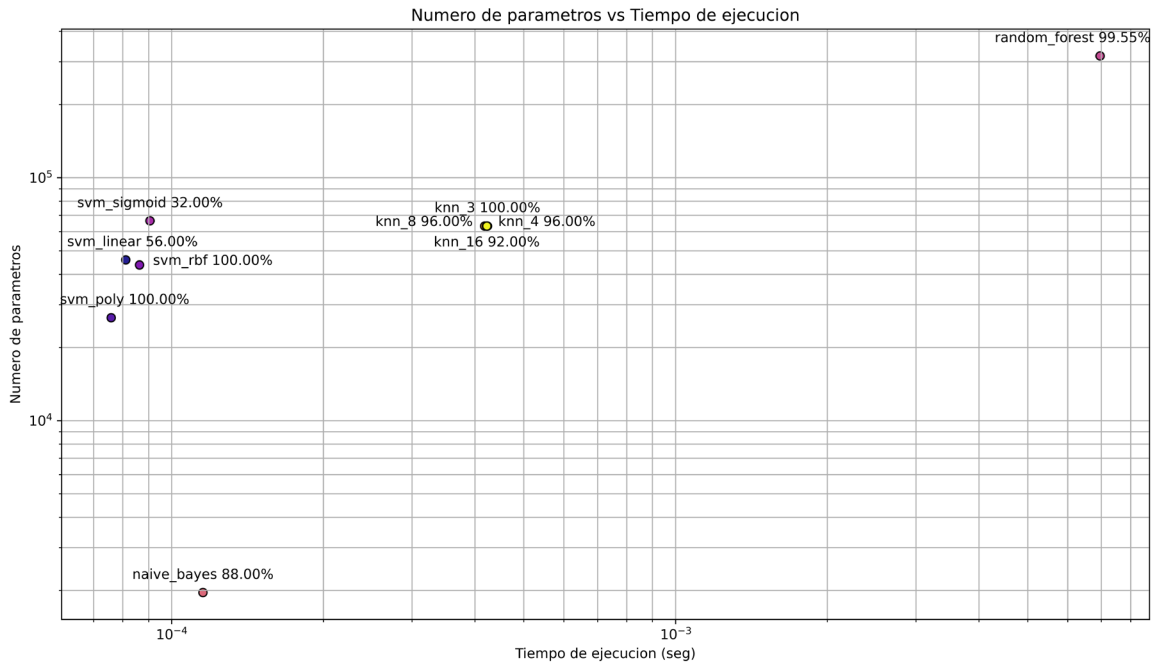


Figura 81. Numero de parámetros Vs Tiempo de ejecución de modelos de clasificación clásicos.