



EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Tecnológico Nacional de México

Centro Nacional de Investigación
y Desarrollo Tecnológico

Tesis de Doctorado

Aprendizaje profundo para la navegación aérea
en entornos naturales desconocidos

presentada por

M.R.S.I. Alexandra Romero Lugo

como requisito para

la obtención del grado de

Doctora en Ciencias de la Computación

Directora de tesis

Dra. Andrea Magadán Salazar

Codirector de tesis

Dr. Jorge Alberto Fuentes Pacheco

Cuernavaca, Morelos, México. Febrero de 2025

 cenidet [®] Centro Nacional de Investigación y Desarrollo Tecnológico	ACEPTACIÓN DE IMPRESIÓN DEL DOCUMENTO DE TESIS DOCTORAL		Código: CENIDET-AC-006-D20
	Referencia a la Norma ISO 9001:2008 7.1, 7.2.1, 7.5.1, 7.6, 8.1, 8.2.4		Revisión: 0
			Página 1 de 1

Cuernavaca, Mor., a 16 de enero de 2025

DR. CARLOS MANUEL ASTORGA ZARAGOZA
SUBDIRECTOR ACADÉMICO
P R E S E N T E

ATn: DR. JUAN GABRIEL GONZÁLEZ SERNA
PRESIDENTE(A) DEL CLAUSTRO DOCTORAL DEL
DEPARTAMENTO DE CIENCIAS COMPUTACIONALES

Los abajo firmantes, miembros del Comité Tutorial de la estudiante **M.C. ALEXANDRA ROMERO LUGO** manifiestan que después de haber revisado el documento de tesis titulado **"APRENDIZAJE PROFUNDO PARA LA NAVEGACIÓN AÉREA EN ENTORNOS NATURALES DESCONOCIDOS"**, realizado bajo la dirección de la Dr. Andrea Magadan Salazar y codirección del Dr. Jorge Alberto Fuentes Pacheco, el trabajo se **ACEPTA** para proceder a su impresión.

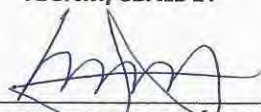
ATENTAMENTE

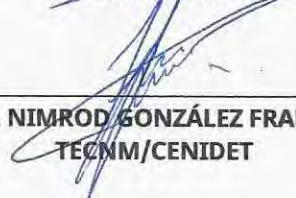
Excelencia en Educación Tecnológica®
"Conocimiento y Tecnología al Servicio de México"


DRA. ANDREA MAGADÁN SALAZAR
TECNM/CENIDET


DR. JORGE ALBERTO FUENTES PACHECO
TECNM/CENIDET


DR. RAÚL PINTO ELÍAS
TECNM/CENIDET


DR. JUAN GABRIEL GONZÁLEZ SERNA
TECNM/CENIDET


DR. NIMROD GONZÁLEZ FRANCO
TECNM/CENIDET


DR. EDDIE HELBERT CLEMENTE TORRES
TECNM/CENIDET


DRA. MIRNA CASTRO BELLO
TecNM / I.T. CHILPANCINGO

c.c.p: C María Elena Gómez Torres/ Jefa del Departamento de Servicios Escolares
 c.c.p: C Née Alejandro Castro Sánchez / Jefe del Departamento de Ciencias Computacionales
 c.c.p: Expediente



Centro Nacional de Investigación y Desarrollo tecnológico
Subdirección Académica

Cuernavaca Mor, 20/enero/2025

Oficio No. SAC/022/2025

Asunto: Autorización de impresión de tesis

ALEXANDRA ROMERO LUGO
CANDIDATA AL GRADO DE DOCTORA
EN CIENCIAS DE LA COMPUTACIÓN
P R E S E N T E

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado "APRENDIZAJE PROFUNDO PARA LA NAVEGACIÓN AÉREA EN ENTORNOS NATURALES DESCONOCIDOS", ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

A T E N T A M E N T E

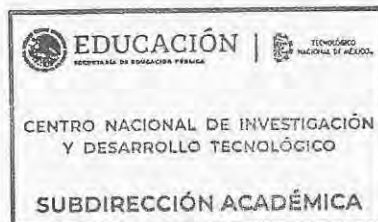
Excelencia en Educación Tecnológica®

"Conocimiento y Tecnología al Servicio de México"

CARLOS MANUEL ASTORGA ZARAGOZA
SUBDIRECTOR ACADÉMICO

c.c.p. Departamento de Ciencias Computacionales
Departamento de Servicios Escolares

CMAZ/lmz



2025
Año de
La Mujer
Indígena

Interior Internado Palmira S/N, Col. Palmira,
C. P. 62490, Cuernavaca, Morelos Tel. 01 (777) 3627770, ext. 4104,
e-mail: acad_cenidet@techn.mx techn.mx | cenidet.tecnm.mx

cenidet
Centro Nacional de Investigación
y Desarrollo Tecnológico



Agradecimiento

Agradezco al Centro Nacional de Investigación y Desarrollo Tecnológico CENIDET por la oportunidad de desarrollo, las facilidades y el apoyo proporcionado para la terminación de esta tesis.

De igual modo, agradezco al Consejo Nacional de Humanidades Ciencias y Tecnologías (CONAHCYT) por el apoyo económico que me brindó durante mis estudios de doctorado.

A mi asesora la Dra. Andrea Magadán Salazar quién además de fungir como mi directora de tesis, fue un apoyo apreciable, muchas gracias por sus enseñanzas, consejos, la paciencia, tiempo y el gran apoyo brindado.

A mi asesor la Dr. Jorge Alberto Fuentes Pacheco por compartir sus conocimientos, dedicación y ayuda invaluable.

A todos y cada uno de los miembros del comité tutorial que se me asignó para esta tesis: Dr. Raúl Pinto Elías, Dr. José Ruiz Ascencio, Dr. Gerardo Reyes Salgado, Dr. Juan Gabriel González Serna, Dr. Nimrod González Franco y Dr. Eddie Helbert Clemente Torres por sus comentarios, sugerencias y observaciones que me ayudaron a culminar este proyecto.

A mis compañeros por sus recomendaciones durante el desarrollo de mi tema doctoral.

Finalmente agradezco a todos los que de forma directa o indirecta contribuyeron a la realización de este proyecto.

Dedicatoria

A mis hijos German Antonio y Gustavo Adolfo, por ser la alegría de mi vida y la fuerza que me hace superarme y desear ser mejor cada día.

A mis padres Humberto y Alma Nelly por las enseñanzas y los valores que me dieron, que me han ayudado tanto en los momentos más difíciles y más importantes. Por su apoyo incondicional, por alentarme en cada proyecto de mi vida.

A mi hermano Humberto, mi cuñada Araceli y mi sobrino Francisco por su cariño incondicional y los buenos momentos que hemos pasado juntos.

Mi amor y agradecimiento a todos ustedes.

Resumen

El cambio climático afecta significativamente a múltiples sectores, incluida la seguridad alimentaria; por lo tanto, las industrias alimentarias, las comunidades agrícolas y los científicos vinculados al sector agrícola deben implementar nuevos procesos o adaptar adecuadamente los existentes para hacer frente a los desafíos actuales.

Aparte del cambio climático se tienen que atender retos como, las áreas de cultivo limitadas, la necesidad de agua dulce y una población en constante crecimiento. Una solución factible es implementar servicios de Tecnologías de la Información y la Comunicación, en especial con apoyo de Vehículos Aéreos No Tripulados (VANTs) combinado con el análisis de información e imágenes, lo que puede proporcionar soluciones prometedoras para la Agricultura de Precisión (AP).

La Agricultura de Precisión consiste en usar técnicas de sensores remotos y análisis de imágenes, para obtener índices de la vegetación y las condiciones de estrés, permitiendo así prevenir la pérdida de cultivos y mejorar el rendimiento de manera más eficiente.

Actualmente el uso de VANTs en el sector de la agricultura ha ido aumentando; debido a que son una solución apropiada para realizar aplicaciones de monitoreo o fumigación. Los pequeños VANTs mejor conocidos como drones se caracterizan por ser una solución económica, ya que son capaces de proveer imágenes monoculares de alta calidad. Cabe mencionar que la Asociación Internacional de Sistemas de Vehículos No Tripulados, AUVSI por sus siglas en inglés, menciona que el 80% de los VANTs se utilizarán para fines agrícolas en el futuro cercano.

No obstante, debido a que los VANTs tienen una toma de decisiones limitada estos solo pueden operar en altas altitudes. Para lograr navegar a bajas altitudes autónomamente, estos requieren la capacidad de responder a una serie de condiciones para prevenir choques. Una de las principales funcionalidades que se requiere es que sean capaces de evadir obstáculos; normalmente este problema es tratado usando sensores de distancia, escáner láser y sonar; sin embargo, estos suelen capturar limitada información, ser costoso, pesados y/o con alto consumo de energía y se pueden usar en ciertas plataformas.

En el contexto específico de la evasión autónoma de colisiones, las técnicas de Aprendizaje Profundo (DL, por sus siglas en inglés *Deep Learning*) han demostrado su eficacia. Estas técnicas resuelven una amplia variedad de tareas robóticas en áreas como percepción, planificación, posicionamiento y control, que implican el aprendizaje de comportamientos complejos.

Por lo anterior, se propone utilizar el Aprendizaje Profundo para permitir a un vehículo aéreo evadir obstáculos de forma autónoma en diferentes tipos de ambientes externos, usando como sensor de percepción una cámara monocular RGB debido a su bajo costo y su tamaño pequeño.

Abstract

Climate change significantly affects multiple sectors, including food security; therefore, food industries, agricultural communities and scientists linked to the agricultural sector must implement new processes or adequately adapt existing ones to face current challenges.

Apart from climate change, challenges such as limited crop areas, the need for fresh water and a constantly growing population must be addressed. A feasible solution is to implement Information and Communication Technology services, especially with the support of Unmanned Aerial Vehicles (UAVs) combined with information and image analysis, which can provide promising solutions for Precision Agriculture (PA).

Precision Agriculture consists of using remote sensing techniques and image analysis to obtain indices of vegetation and stress conditions, thus allowing to prevent crop loss and improve yield more efficiently.

Currently, the use of Unmanned Aerial Vehicles (UAVs) in the agricultural sector has been increasing; because they are an appropriate solution to carry out monitoring or fumigation applications. Small UAVs, better known as drones, are characterized by being an economical solution, since they are capable of providing high-quality monocular images. It is worth mentioning that the Association of Unmanned Vehicle Systems International, AUVSI, mentions that 80% of UAVs will be used for agricultural purposes in the near future.

However, because UAVs have limited intelligence, they can only operate at high altitudes; in order to navigate at low altitudes autonomously, they require the ability to respond to a series of conditions to prevent crashes. One of the main functionalities required is that they are able to avoid obstacles; normally this problem is dealt with using distance sensors, laser scanners and sonar; however, these usually capture limited information, are expensive, heavy and/or have high energy consumption and can be used on certain platforms. In the specific context of autonomous collision avoidance, Deep Learning (DL) techniques have proven effective in solving a wide variety of robotic tasks in areas such as perception, planning, positioning, and control, which involve learning complex behaviors.

Therefore, it is proposed to use Deep Learning to enable an Aerial Vehicle to autonomously avoid obstacles in different types of external environments, using a monocular RGB camera as a perception sensor due to its low cost and small size.

Contenido

Resumen	I
Abstract	II
Lista de Figuras	III
Lista de Tablas	IV
Glosario	V
CAPÍTULO 1. Introducción	11
1.1 Descripción del problema	12
1.2 Complejidad del problema.....	13
1.3 Objetivos	14
1.4 Alcances y limitaciones.....	15
1.5 Propuesta de solución	16
1.6 Organización de la tesis.....	18
CAPÍTULO 2. Estado del arte.....	19
2.1 Aprendizaje profundo en VANTs.....	19
2.2 Aprendizaje profundo para la detección de obstáculos	21
2.3 Aprendizaje profundo para la estimación de profundidad.....	24
CAPÍTULO 3. Marco teórico.....	29
3.1 Agricultura de precisión	29
3.2 Vehículos aéreos no tripulados (VANTs).....	30
3.3 Aplicaciones de los VANTS en agricultura de precisión.....	31
3.4 Detección de obstáculos para la navegación de VANTs.....	32
3.5 Técnicas de visión por computadora para navegación.....	33
3.6 Redes neuronales para la estimación de profundidad monocular	35
3.7 Métricas para la estimación de profundidad	35
CAPÍTULO 4. Método de solución.....	38
4.1 Descripción de las aportaciones realizadas a la arquitectura propuesta	39
4.1.1 Fusión de mapas	42
4.1.2 Parches piramidales	43
4.1.3 Procesamiento en paralelo.....	45
4.2 Diseño del sistema de evasión de obstáculos	46
4.2.1 Detección de obstáculos	48

4.2.2 Generación de comandos de navegación	48
CAPÍTULO 5. Pruebas y Resultados	51
5.1 Herramientas utilizadas (software y equipo).....	51
5.2 Conjuntos de datos utilizados.....	52
5.3 Experimentación con la red de estimación de profundidad	56
5.4 Experimentación con el diseño del sistema de evasión de obstáculos.....	62
5.5 Análisis de resultados.....	63
5.5.1 Comparativa de la arquitectura propuesta con la literatura	63
5.5.2 Comprobar que la arquitectura propuesta es generalizable.....	66
5.5.3 Pruebas de navegación en interiores.....	67
5.5.4 Pruebas de navegación en exteriores	70
5.6 Comentarios finales	77
CAPÍTULO 6. Conclusiones	78
6.1 Conclusiones	78
6.3 Objetivos logrados.....	79
6.4 Aportaciones	79
6.5 Trabajo futuro.....	79
6.6 Productos académicos adicionales.....	80
Referencias.....	81

Lista de Figuras

Figura 1.1 Esquema de la propuesta solución.....	16
Figura 1.2 Metodología de solución	17
Figura 3.1 Imágenes de [4] como ejemplo para la segmentación de superpíxel creada para la métrica D3R.....	37
Figura 4.1 Arquitectura de Boosting Monocular Depth [65].....	40
Figura 4.2 Arquitectura del generador "U-Net", codificador-decodificador con conexiones de salto entre capas reflejadas en las pilas de codificador y decodificador [24]	40
Figura 4.3 Arquitectura de LeReS [25].....	41
Figura 4.4 Arquitectura del módulo DPM de LeReS [25]	41
Figura 4.5 Arquitectura propuesta (Boosting Pyramidal Depth)	42
Figura 4.6 Resultados de los experimentos realizados en [23] donde se demuestra que los píxeles generados no están alineado geométricamente	43
Figura 4.7 Ejemplo de cómo se generan los parches en Boosting Monocular Depth [1]	44
Figura 4.8 Propuesta de Atrous Spatial Pyramid de DeepLab [35].....	44
Figura 4.9 Ejemplo de parches piramidales con dos niveles	45
Figura 4.10 Medidas del VANT DJI Tello utilizando protectores.....	47
Figura 4.11 Arquitectura del sistema de navegación evasiva de obstáculos	47
Figura 4.12 Rangos de colores del mapa de color JET	48
Figura 4.13 Movimientos que permite ejecutar el VANT.....	49
Figura 4.14 Áreas de posible navegación.....	49
Figura 5.1 Imágenes RGB del ambiente Gascola de TartanAir [11]	53
Figura 5.2 Imágenes RGB del ambiente Season Forest de TartanAir [11]	54
Figura 5.3 Imágenes RGB del ambiente Season Forest Winter de TartanAir [11]	54
Figura 5.4 Imágenes RGB del ambiente Amusement de TartanAir [11].....	54
Figura 5.5 Imágenes RGB del ambiente Western Desert de TartanAir [11].....	55
Figura 5.6 Imágenes del ambiente de simulación City Park [8]	55
Figura 5.7 Tiempos de los subprocesos de la red propuesta.....	61
Figura 5.8 Tiempos de generación de los comandos de navegación	62
Figura 5.9 Imágenes generadas por el sistema de navegación	62
Figura 5.10 Representación de un sendero en interiores	67
Figura 5.11 Muestra de la detección de obstáculos del sistema de navegación.....	68
Figura 5.12 Número de obstáculos identificados por el dron cada 10 segundos a lo largo de los vuelos en interiores realizados.....	70
Figura 5.13 Número de obstáculos identificados por el sistema a lo largo de los vuelos al aire libre realizados durante la mañana	72
Figura 5.14 Número de obstáculos identificados por el sistema a lo largo de los vuelos al aire libre realizados durante la tarde	74

Lista de Tablas

Tabla 2.1 Características de arquitecturas de aprendizaje profundo	20
Tabla 2.2 Resumen de los artículos relacionados con Aprendizaje Profundo para la detección de obstáculos.....	23
Tabla 2.3 Resumen de los artículos relacionados con Estimación de Profundidad	26
Tabla 4.1 Comparativa de redes de estimación de profundidad a partir de imágenes.	39
Tabla 4.2 Características del VANT DJI Tello Quadcopter	46
Tabla 5.1 Librerías para la implementación de las redes	51
Tabla 5.2 Características de las trayectorias de TartanAir [3]	53
Tabla 5.3 Características de las trayectorias de CityPark.....	56
Tabla 5.4 Resultado de las métricas de Boosting piramidal con distintos niveles utilizando la trayectoria Gascola	57
Tabla 5.5 Tiempos de ejecución de Boosting piramidal con distintos niveles utilizando la trayectoria Gascola.....	57
Tabla 5.6 Mapas de profundidad obtenidos con Boosting piramidal con distintos niveles	58
Tabla 5.7 Tiempos de inferencia obtenidos utilizando paralelización	59
Tabla 5.8 Comparativa de los resultados de las métricas de Boosting piramidal con procesamiento secuencial y en batches.....	59
Tabla 5.9 Mapas de profundidad obtenidos con Boosting piramidal secuencial y en batches.....	60
Tabla 5.10 Comparativa de los mapas de profundidad obtenidos del estado del arte y la red propuesta.....	64
Tabla 5.11 Comparativa de las redes de estimación en diferentes trayectorias de TartanAir.....	65
Tabla 5.12 Comparativa de los tiempos de ejecución de las redes de estimación en diferentes trayectorias de TartanAir	65
Tabla 5.13 Resultado de las métricas de Boosting Piramidal en diferentes trayectorias de TartanAir.....	66
Tabla 5.14 Tiempos de ejecución de Boosting Piramidal en diferentes trayectorias de TartanAir.....	66
Tabla 5.15 Tiempo de vuelo promedio de vuelos exitosos y fallidos.....	68
Tabla 5.16 Detalle de los vuelos realizados en interiores	69
Tabla 5.17 Detalles de los vuelos realizados en exteriores en la mañana (8:00am)	71
Tabla 5.18 Detalles de los vuelos realizados en exteriores en la tarde (2:00pm)	73
Tabla 5.19 Imágenes con obstáculos detectados.....	75
Tabla 5.20 Imágenes con espacio libre de navegación.....	76

Glosario

AbsRel	Absolute Relative Difference
AP	Agricultura de Precisión
Batch (Batches)	Lote(s) de imágenes
CNN	Convolutional Neural Network (Red Neuronal Convolutacional)
DBN	Deep Belief Networks
DL	Deep Learning
DNN	Deep Neural Network (Red Neuronal Profunda)
FCNN	Fully Connected Neural Network
FCRN	Fully Convolutional Residual Network
GNSS	Global Navigation Satellite Systems
GPS	Global Positioning Systems
GPU	Graphics Processing Unit (Unidad de Procesamiento Gráfico)
Ground Truth	Información que se sabe que es real o verdadera
IMU	Inertial Measurements Unit
JCR	Journal Citation Reports
LiDAR	Light Detection and Ranging
LSTM	Long Short-Term Memory
Merged	Fusionar
Mutiprocessing	Ejecución de dos o más secuencias de instrucciones simultáneamente por una computadora con más de un procesador central.
OpenCV	Librería de visión artificial
RADAR	Radio Detection and Ranging
RBM	Restricted Boltzmann Machine
Resize	Cambiar el tamaño
RF	Receptive Field (Campo receptivo)
RGB	Imagen de color con los canales Red, Green, Blue (Rojo, Verde, Azul)
RGBD	Imagen que combina información de color de los canales RGB con información de profundidad.
RMSE	Root Mean Square Error
RNN	Recurrent Neural Networks
SLAM	Simultaneous Localization And Mapping
SqRel	Square Relative Error
Upsampling	Sobremuestreo de píxeles
Umbralización	Técnica para dividir una imagen en dos (o más) clases de píxeles
VANTs	Vehículos Aéreos No Tripulados
Wi-Fi	Wireless Fidelity
2D	2 dimensiones
3D	3 dimensiones

CAPÍTULO 1

Introducción

El sector de la agricultura tiene varios retos que atender como son el cambio climático, las áreas de cultivo limitadas, la necesidad de agua dulce y una población en constante crecimiento. Una solución factible para los desafíos anteriores es implementar servicios de Tecnologías de la Información y la Comunicación, en especial el desarrollo de Vehículos Aéreos No Tripulados (VANTs) combinado con el análisis de información e imágenes, esto puede proporcionar soluciones prometedoras para la Agricultura de Precisión (AP).

La aplicación más común de la AP es evaluar la sanidad de la vegetación usando técnicas de sensores remotos y análisis de imágenes; aquí es en donde los VANTs realizan una importante labor. Los pequeños VANTs mejor conocidos como drones se caracterizan por ser una solución económica y son capaces de proveer imágenes de alta calidad. Estas imágenes se pueden utilizar para obtener índices de la vegetación y permite a los agricultores monitorear constantemente la variación del cultivo y las condiciones de estrés, permitiendo así manejar los cultivos de una manera más eficiente [1].

No obstante, debido a capacidad de toma de decisiones los VANTs solo pueden operar en altas altitudes para obtener información importante acerca de la salud de las plantas, lo que conlleva el uso de sensores costosos y pesados. Para volar a bajas altitudes, están sujetos a una serie de condiciones y una de las principales funcionalidades que se requiere es que sean capaces de evadir obstáculos para prevenir choques. Normalmente, el problema de evitar colisiones es tratado usando sensores de distancia, escáner láser y sonar; sin embargo, estos dispositivos suelen capturar información escasa, ser costosos, pesados y/o con alto consumo de energía [2].

Las técnicas de Aprendizaje Profundo (DL, por sus siglas en inglés *Deep Learning*) han surgido como una alternativa prometedora para mejorar la detección de obstáculos en tiempo real y la prevención de colisiones para los VANTs [3]. Los métodos de DL son capaces de realizar dos tareas al mismo tiempo, la representación de los datos y la clasificación con sólo alimentar la red con datos brutos. En el contexto específico de la evasión autónoma de colisiones, las técnicas de DL han demostrado su eficacia a la hora de resolver una amplia variedad de tareas robóticas en áreas como percepción, planificación, posicionamiento y control, que implican el aprendizaje de comportamientos complejos.

En resumen, el brindar a los VANTs de la capacidad de volar a bajas altitudes aumentaría el número de aplicaciones y actividades que los drones pudieran realizar en el sector de la agricultura sin depender de sensores costosos. Por lo que la presente tesis propone utilizar el Aprendizaje Profundo como una herramienta eficaz para permitir a un vehículo aéreo realizar la evasión de obstáculos de forma autónoma y en diferentes tipos de ambientes externos. Debido a que, en la Agricultura de Precisión se tienen diferentes tipos de cultivos, el VANT debe de adaptarse a los diferentes entornos, ya que el objetivo no es reentrenar el algoritmo para cada situación.

1.1 Descripción del problema

En años recientes los Vehículos Aéreos No Tripulados (VANTs) se han utilizado ampliamente en la agricultura de precisión, apoyando en la recolección de imágenes aéreas y con el uso de servicios de Tecnologías de la Información y la Comunicación (TICs) procesar la información proporcionada por múltiples fuentes para extraer conclusiones útiles sobre la comprensión del suelo, permitiendo así manejar los cultivos de una manera más eficiente.

Sin embargo, debido a la limitada inteligencia de los VANTs se han usado en altas altitudes para obtener información como la sanidad de las plantas, lo que implica el uso de sensores como IMU y GPS (por sus siglas en inglés Global Positioning Systems) para navegar autónomamente en rutas predefinidas [3].

Depender de la localización basada en GPS tiene algunas debilidades, debido a que es una señal de radio y es susceptible a interferencias en la señal. Aparte que la información obtenida mediante la recolección de imágenes aéreas permite apoyar sólo en tareas como la vigilancia de las plantaciones, definir la estructura del campo o cultivo, contar árboles, obtener índices de vegetación y monitorear constantemente la variación del cultivo y las condiciones de estrés.

Lograr que los VANTs puedan volar a bajas altitudes permite ampliar las tareas de apoyo en la Agricultura de Precisión, tales como vigilar de cerca los cultivos, rociar fertilizante con una cantidad adecuada según la condición del suelo, detectar enfermedades de los cultivos, estrés hídrico, infestaciones de plagas, deficiencias de nutrientes y otras condiciones relevantes que afectan la productividad de los cultivos [4].

Cuando se tienen robots móviles operando en el mundo real, estos dependen de múltiples condiciones, en nuestro caso se consideran: terrenos complicados, pequeños senderos, maquinaria de riego, postes de electricidad, maquinaria agrícola y personal que labora en las zonas de cultivos; por lo que una de las principales funcionalidades que los VANTs deben poseer es que sean capaces de evadir obstáculos para así prevenir colisiones. Para cumplir el objetivo, en el estado del arte, se han propuesto varios métodos utilizando sensores sofisticados como RGB-D, LiDAR (por sus siglas en inglés Light Detection and Ranging), RADAR (por sus siglas en inglés Radio Detection and Ranging), cámaras estereoscópicas, escáner sonar, sensores infrarrojos y algoritmos complejos como SLAM para construir modelos en 3D del entorno y extraer información

de profundidad para deducir el espacio navegable [5]. No obstante, estos métodos presentan un alto costo computacional, requieren sensores costosos, pesados y de alto consumo de energía; además que suelen capturar solamente información limitada y se pueden usar en ciertas plataformas.

Adicionalmente, las plataformas o sistemas VANT aún enfrentan importantes limitaciones relacionadas, principalmente, con la autonomía y el peso, lo que impacta en sus capacidades para capturar y procesar los datos requeridos para desarrollar sistemas autónomos y robustos de detección y evasión de obstáculos en tiempo real. En algunos casos como en grandes cultivos, un solo VANT no puede completar el trabajo debido a la limitada fuente de energía, la mayoría de los VANTs utilizan celdas de batería como fuente de energía y el aumentar esta capacidad implica un aumento de peso considerable limitando así la capacidad del VANT de llevar carga adicional como llevar un sistema de fumigación.

Considerando todo lo mencionado, se tiene como reto en el área de investigación de vuelo autónomo de VANTs, el permitir que estos puedan volar en entornos naturales de espacios reducidos y complejos, como cultivos, utilizando sensores que sean de bajo costo y ligeros y que representen un bajo costo computacional.

Este trabajo se basa en la hipótesis de que el aprendizaje profundo mejora las capacidades de la navegación visual automática, ya que las redes neuronales profundas permiten el desarrollo de enfoques de aprendizaje en los que la extracción de características se realiza utilizando un gran conjunto de parámetros que son aprendidos automáticamente en la fase de entrenamiento. Estas características también se pueden obtener con métodos de visión por computadora, que involucran tareas como el flujo óptico o la estimación de profundidad a partir de los datos capturados por cámaras monoculares; lo que resulta ser una opción considerablemente más barata que los sensores mencionados anteriormente.

Bajo este marco, el problema abordado en este trabajo consiste en proponer un algoritmo de Aprendizaje Profundo que detecte el espacio libre para navegar y evadir obstáculos naturales utilizando imágenes monoculares obtenidas con la cámara frontal con la que están equipados los VANTs.

1.2 Complejidad del problema

El problema abordado en este trabajo es identificar (de manera automática) en imágenes adquiridas a baja altura en un sendero o cultivo, el espacio libre para navegar y evadir obstáculos naturales; usando las imágenes monoculares obtenidas con la cámara frontal con la que están equipados los VANTs.

El desarrollo del sistema propuesto conlleva varios subproblemas complejos a resolver tales como:

- En el área de redes neuronales profundas, se pueden encontrar muchas arquitecturas diferentes y es indispensable conocer en detalle varias de ellas para determinar cuál sería la red base para modificar.

- El ambiente en los escenarios de cultivos naturales puede presentar una gran variedad de obstáculos de diferentes tamaños y formas tales como troncos, ramas, personas, maquinaria, etc.
- Al considerar bajas altitudes para la navegación del VANT se tiene como resultado que las imágenes monoculares presenten sombras de los troncos y/o ramas delgadas ocasionando complicada la detección del espacio libre.
- El sistema debe dar respuesta de forma rápida y precisa, para que el vehículo pueda realizar las maniobras suficientes para evitar el obstáculo.
- Las imágenes monoculares pueden presentar problemas de emborronamiento, cambios en la intensidad luminosa, escala y perspectiva, condiciones diferentes a las consideradas en el entrenamiento de la red neuronal.

1.3 Objetivos

General

Formular y evaluar cambios en la arquitectura de una red neuronal de aprendizaje profundo que identifique el espacio libre en imágenes monoculares y le permita a un vehículo aéreo evadir obstáculos naturales, en escenarios típicos de sembrado, de manera automática.

Específicos

- Proponer o modificar la arquitectura de una Red Neuronal Profunda para detectar el espacio libre para navegar en entornos reales de cultivos, usando como sensor de percepción una cámara monocular.
- Formular una política de planeación de movimientos que module la orientación, realizando movimientos en tres grados de libertad del vehículo aéreo, en función de las predicciones de la red neuronal profunda.
- Diseñar e implementar una estrategia de Transferencia de Aprendizaje para poder incorporar conocimiento de profundidad obtenido durante la etapa de entrenamiento, fuera de línea, de diferentes agentes reales o simulados.
- Evaluar la capacidad de generalización del sistema propuesto en diversos escenarios de cultivos reales, distintos a los utilizados en el entrenamiento; mediante las métricas clásicas utilizadas en el área.

1.4 Alcances y limitaciones

Alcances

- Se define la arquitectura de Red Neuronal Profunda del modelo.
- La red de aprendizaje profundo utiliza imágenes monoculares como su principal fuente de información para detectar y evadir obstáculos en entornos de cultivos reales.
- La red sólo detecta espacios abiertos para navegar, por lo que no se realiza un reconocimiento ni etiquetado de los obstáculos.
- Para la política de planeación de movimientos del VANT se tienen considerado realizar movimientos en 3 grados de libertad del vehículo aéreo.
- Para la etapa de entrenamiento de la red se utilizaron conjuntos de datos publicados en el estado del arte.
- En la etapa de pruebas se consideran entornos de cultivos reales diferentes a los utilizados en la etapa de entrenamiento.
- Se evaluó la navegación del VANT basándose en las métricas utilizadas en el estado del arte como son: distancia de vuelo antes de colisión, el tiempo de navegación del vuelo realizado y la tasa de éxito en vuelos realizados.
- Se evaluó la estimación de profundidad con base en las métricas clásicas utilizadas en el estado del arte, tales como: diferencia relativa absoluta (AbsRel, por su acrónimo en inglés Absolute Relative Difference), el error cuadrático medio (RMSE, por sus siglas en inglés Root Mean Square Error), el RMSE (log) y el error relativo cuadrático (SqRel, por su acrónimo en inglés Square Relative Error).

Limitaciones

La navegación aérea en entornos naturales desconocidos presenta limitaciones que son importantes considerar, como son:

- Las pruebas en entornos de cultivos reales están sujetas a los recursos facilitados por la institución.
- La capacidad de respuesta del sistema propuesto está sujeta a los recursos de cómputo facilitados por la institución.
- No se realizó ninguna aportación en la interfaz de comunicación con el robot ni en su sistema de control.
- La navegación del VANT se realizó considerando una altura mínima de 1 metro y como máximo 3 metros, dependiendo de la altura de los cultivos disponibles.
- Las pruebas que se realizaron en espacios abiertos, se llevaron a cabo cuando no haya demasiado viento, para evitar así desestabilizar el VANT.

1.5 Propuesta de solución

Las técnicas de Aprendizaje Profundo (DL, por sus siglas en inglés *Deep Learning*) han surgido como una alternativa prometedora para mejorar la detección de obstáculos en tiempo real y la prevención de colisiones para los VANTs.

Los métodos de DL son capaces de realizar dos tareas al mismo tiempo, la extracción de los atributos y la clasificación con solo alimentar la red con datos brutos. En el contexto específico de la evasión autónoma de colisiones, las técnicas de DL han demostrado su eficacia a la hora de resolver una amplia variedad de tareas robóticas en áreas como percepción, planificación, posicionamiento y control, que implican el aprendizaje de comportamientos complejos.

Partiendo de lo anterior, se propone adquirir imágenes mediante una cámara monocular en un VANT que serán la entrada a una red neuronal convolucional que realizará la descripción de las mismas para identificar la presencia del espacio libre para navegar. En caso de que enfrente del VANT no exista espacio libre suficiente para que este transite, se realizará un análisis de la escena para evitar una colisión y se predecirá el ángulo de giro de navegación el cual es considerado como un problema de regresión; para que así el VANT pueda continuar con la trayectoria planeada, tal como se puede observar en la Figura 1.1.

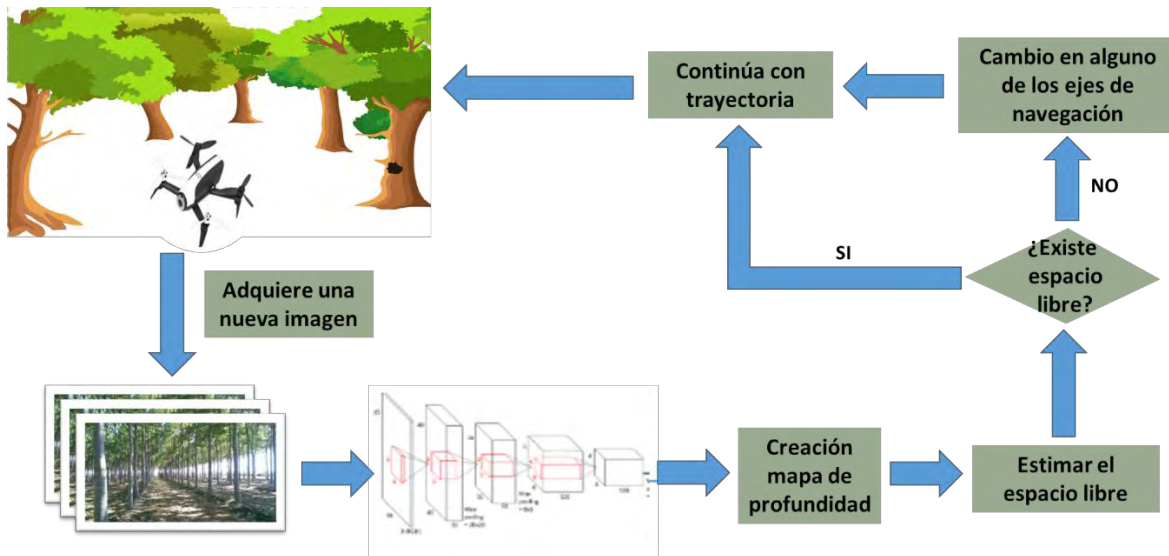


Figura 1.1 Esquema de la propuesta solución

Para la propuesta de la arquitectura se analizaron varias arquitecturas de aprendizaje profundo. El desarrollo del algoritmo de aprendizaje profundo se realiza utilizando la librería *Pytorch* [6] y *Keras* [7]; como plataforma para experimentar con las técnicas de aprendizaje profunda se consideró *AirSim* [8] para los ambientes simulados.

Como conjunto de datos de entrenamiento se consideró utilizar imágenes de varios conjuntos de datos públicos tales como *KITTI* [9], *NYU-v2* [10] y *TartanAir* [11]; los cuales cuentan con imágenes de ambientes exteriores y se han utilizado en trabajos

previos donde se realiza la tarea de evasión de obstáculos. También se generó un nuevo conjunto de datos a partir de propias imágenes recolectadas en un ambiente simulado.

En la etapa de experimentación se utiliza un VANT modelo *DJI Tello Quadcopter* debido a que su control es abierto y puede ser programado utilizando el SDK oficial de Tello. Los comandos de control de navegación para el VANT físico son producidos por la red neuronal que se ejecuta desde una estación de trabajo equipada con una tarjeta gráfica y usando una conexión Wi-Fi como medio de comunicación con el VANT físico.

La evaluación se llevó a cabo en diversos entornos reales con diferentes características, los cuales serán distintos a los utilizados en el entrenamiento. Los resultados del sistema se compararán con métodos publicados en el estado del arte, utilizando las métricas clásicas en el aprendizaje profundo, con la finalidad de demostrar la funcionalidad y la generalización del sistema propuesto.

Considerando lo antes mencionado se tiene como metodología de solución cinco etapas con diferentes actividades que se puede observar en la Figura 1.2.

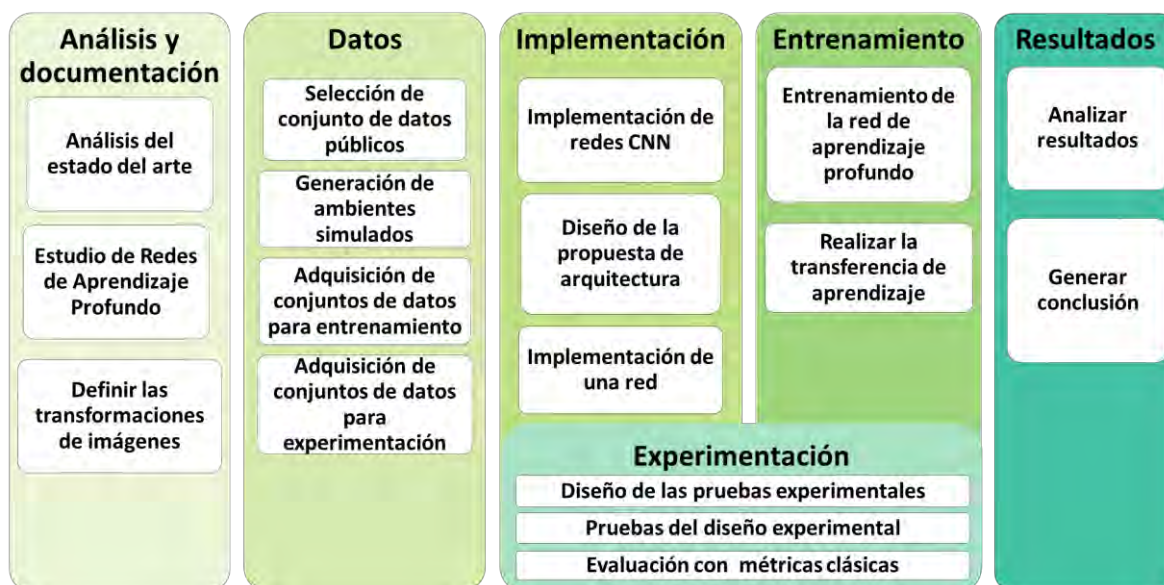


Figura 1.2 Metodología de solución

1.6 Organización de la tesis

En los próximos capítulos se presenta una descripción detallada de todos los procedimientos requeridos para abordar la problemática planteada en el presente trabajo de investigación. La estructura de la tesis se compone de la siguiente manera:

Capítulo 1: Análisis del problema de este trabajo.

Capítulo 2: Estado del arte y su discusión correspondiente.

Capítulo 3: Marco teórico.

Capítulo 4: Diseño e implementación de la solución para el problema de tesis.

Capítulo 5: Objetivo y diseño de las pruebas que se realizaron para cada módulo del desarrollo de la tesis; además del análisis de los resultados obtenidos.

Capítulo 6: Conclusiones generales, la discusión sobre el tema de tesis, trabajos futuros y aportaciones.

Anexos: Experimentación adicional realizada en el transcurso del doctorado.

CAPÍTULO 2

Estado del arte

La investigación sobre evasión de obstáculos ha sido un tema de estudio en diversas disciplinas, como la robótica, la informática y la inteligencia artificial. En las últimas décadas el interés en este tema se ha intensificado y ha demostrado avances al considerar técnicas de visión computacional en conjunto con técnicas de aprendizaje profundo.

En el presente capítulo se describen los conceptos base del tema, así como los hallazgos reportados en la literatura para dar solución a la evasión de obstáculos. El capítulo se encuentra dividido en cuatro secciones que son: aprendizaje profundo de VANTs, aprendizaje profundo para la detección de obstáculos, aprendizaje profundo para la estimación de profundidad y por último el estado de la práctica.

2.1 Aprendizaje profundo en VANTs

Durante la última década, el Aprendizaje Profundo (DL) ha demostrado ser un excelente enfoque en el área de la Inteligencia Artificial, resolviendo diferentes problemas, y ha producido buenos resultados en diversas áreas como el reconocimiento de imágenes, la obtención de imágenes médicas o el reconocimiento de voz.

Los métodos de DL son métodos de aprendizaje de representación con múltiples capas de procesamiento, obtenidas al componer módulos simples, pero no lineales, que transforman la representación en un nivel (comenzando con la entrada sin procesar) en una representación superior, más abstracta. El aspecto clave del DL es que estas capas no están diseñadas por ingenieros humanos: aprenden de los datos mediante un procedimiento de aprendizaje de propósito general [12]. El aprendizaje profundo descubre una estructura intrincada en grandes conjuntos de datos mediante el uso del algoritmo de retropropagación para indicar cómo una máquina debe cambiar sus parámetros internos que se utilizan para calcular la representación en cada capa a partir de la representación en la capa anterior.

Cuando se escucha el término Aprendizaje Profundo, sólo se piensa en una red neuronal grande y profunda. Profunda se refiere a la cantidad de capas típicamente y, por lo tanto, este es el término popular que se ha adoptado [13]. Esto permite conceptualizar el DL como grandes redes neuronales donde las capas se construyen una encima de la otra, aumentando gradualmente en profundidad. Considerando este concepto resulta la pregunta ¿Cuántas capas necesita una red neuronal para considerarse profunda? hasta el momento no se tiene un consenso entre los expertos sobre la profundidad de una red para ser considerada profunda; por lo que se prefiere enfocarse en qué tipo de redes son consideradas de DL.

En el contexto de interpretación de señales fisiológicas para aplicaciones del cuidado de la salud [14] identifican varios modelos de DL que han demostrado potencial para mejorar el rendimiento en comparación con técnicas tradicionales, estos son: redes neuronales convolucionales (CNN,

Convolutional Neural Networks); autoencoder; redes de creencia profunda (DBN, *Deep Belief Networks*), relacionadas con las máquinas Boltzmann restringidas (RBM, *Restricted Boltzmann Machine*); y las redes neuronales recurrentes (RNN, *Recurrent Neural Networks*), en particular las de tipo LSTM (*Long Short-Term Memory*). En la Tabla 2.1 se muestran las aplicaciones y características de algunos modelos de DL.

Tabla 2.1 Características de arquitecturas de aprendizaje profundo

Modelo	Aplicación	Características
CNN	Procesamiento de imágenes, reconocimiento de video y tareas dentro del procesamiento del lenguaje natural	Contiene capas de convolución (extraer características de la entrada), agrupamiento o pooling (reducir dimensionalidad) y totalmente conectadas (clasificación o identificación de características de la imagen)
RNN	Reconocimiento de voz y de escritura a mano	Tiene conexiones que retroalimentan las capas previas o la misma capa, permite mantener memoria de entradas pasadas
LSTM	Sistemas de subtítulos de imágenes y videos	Introduce el concepto de célula de memoria (con compuertas de entrada, salida y olvido) que retienen su valor por un periodo de tiempo
DBN	Reconocimiento de imágenes, recuperación de información, comprensión del lenguaje natural y predicción de fallas	Cada par de capas conectadas es una máquina Boltzmann restringida (una DBN es representada como una pila de RBM); cada RBM es entrenada para reconstruir su entrada, al final se realiza un proceso de fine-tuning, etiquetando los nodos de salida acorde a su significado o lo que representan en el contexto de la red
DSN	Recuperación de información y reconocimiento continuo de voz	La red se conforma de un conjunto de redes individuales; permite el entrenamiento de módulos individuales de forma aislada y en paralelo a otros

Fuente: Adaptada de [15]

De las diferentes arquitecturas de DL, la más usada es la Red Neuronal Convolutiva (CNN). La cual, es un tipo de arquitectura jerárquica para la extracción de características que permite generar un conjunto de mapas de características de alta dimensión a partir de los datos del sensor sin procesar. Por lo tanto, los modelos basados en CNN se utilizan ampliamente en el contexto de la prevención de colisiones [16], especialmente en las etapas relacionadas con la percepción del entorno o con la estimación de la profundidad [17].

El nombre Red Neuronal Convolutiva indica que la red emplea una operación matemática llamada convolución (multiplicación por elementos de dos matrices seguida de una suma) en, al menos una de sus capas; esta operación actúa como un extractor de características al aprender de las diversas señales de entrada. En la red se pueden realizar varias convoluciones, creando diferentes representaciones del conjunto de datos de aprendizaje, comenzando con descripciones más generales en las primeras capas y volviéndose más específicas en los niveles más profundos. Las capas convolucionales codifican múltiples características de nivel inferior en atributos más discriminatorios, de una manera que es espacialmente contextual. Las capas completamente conectadas, colocadas en muchos casos cerca de la salida del modelo, actúan como clasificadores

explotando las características de alto nivel aprendidas para clasificar imágenes de entrada en clases predefinidas o para hacer predicciones numéricas [18].

Considerando que la información de cada observación está relacionada con una representación parcial del entorno que rodea al VANT, la posibilidad de almacenar y rastrear toda la información relevante obtenida en el pasado permite tener una representación más completa del entorno. Para este propósito se usan las Redes Neuronales Recurrentes (RNN, por sus siglas en inglés *Recurrent Neural Network*) [19] que brindan la capacidad de aprender las dependencias temporales a partir de la información de una secuencia de observaciones, arbitrariamente largas.

Por lo tanto, existen enfoques basados en RNN que permiten realizar un seguimiento de observaciones pasadas, como los mapas de profundidad obtenidos a partir de imágenes RGB, posiciones u obstáculos recientemente evitados. Además, la incorporación de Atención Temporal [20] permite evaluar la información proporcionada por cada observación de una secuencia y, se puede utilizar ponderando cada una de las observaciones recientes en función de su importancia en la toma de decisiones. Esta técnica aumenta la velocidad de entrenamiento y proporciona una mejor generalización sobre el conjunto de datos de entrenamiento. Las redes convolucionales profundas han producido avances en el procesamiento de imágenes, video, voz y audio, mientras que las redes recurrentes han arrojado luz sobre datos secuenciales como texto y voz [21].

También hay una investigación de un tipo de RNN denominada como Long Short-Term Memory (LSTM) [22], que es capaz de aprender dependencias a largo plazo con la capacidad de eliminar información antigua o agregar nueva en cualquier momento, permitiendo así el uso de todos los datos relevantes de sucesos pasados.

2.2 Aprendizaje profundo para la detección de obstáculos

En la mayoría de las investigaciones encontradas utilizan modelos de aprendizaje profundo basados en redes CNN para la tarea de extracción de características y clasificación de regiones.

Debido a que la finalidad de estas investigaciones es la navegación autónoma de los VANTs, la red CNN suele estar acoplada a otra red en donde se pueden encontrar diversas propuestas. En [5] la salida de la red CNN pasa por una red FCNN (siglas en inglés, *Fully Connected Neural Network*) la cual es la encargada de generar el ángulo de orientación de giro del dron en caso de que se haya detectado un obstáculo. En el trabajo [23] se utiliza otra red CNN para predecir la trayectoria del VANT; a diferencia de lo mencionado en [24] acoplan la salida de la red CNN a una máquina de estados para obtener los comandos de vuelo del VANT.

En las investigaciones encontradas se plantearon modelos de redes para usarse en VANTs, por lo que algunos trabajos se enfocaron en reducir el tamaño de la red y el costo computacional. En la investigación de [25] se busca reducir el tiempo de procesamiento de las imágenes utilizando una red CNN de tamaño pequeño, esto lo logran compartiendo la red para realizar dos tareas simultáneamente (predicción de colisión y predecir la dirección de ángulo). En [26] realizan una convolución en puntos de 1x1 y utilizan solo 2.23 millones de parámetros logrando una precisión del 80%. En el trabajo [27] utilizan un patrón de conectividad entre todas las capas de la red y un concatenado de características logrando así tener un modelo compacto. En [28] realizan mejoras a la arquitectura *Inception* teniendo convoluciones factorizadas de 3x3 y el uso de un clasificador auxiliar para propagar la información a la parte inferior de la red, con la versión *Inceptionv3* se tiene

una red con menos costo computacional utilizando menos de 25 millones de parámetros y estableciendo nuevos resultados de precisión en el estado del arte.

En el contexto de la navegación del VANT, otras investigaciones, se enfocan en lograr que el VANT tenga trayectorias seguras y sin desviaciones. En las pruebas realizadas en el trabajo [4] lograron que el VANT navegara autónomamente 113 metros en un ambiente real y se observó que el VANT fue capaz de detenerse enfrente de un obstáculo a una distancia de 0.7 metros; en [29] proponen una red con un algoritmo capaz de recuperar la trayectoria en caso de que alguna falla en el sistema de percepción haya ocurrido logrando en las pruebas recorrer 1 km autónomamente en un bosque desordenado. En las pruebas realizadas en [30] la red puede tomar decisiones más perspicaces en casos de alta ambigüedad utilizando la representación espacio-temporal, teniendo así una reducción considerable de las oscilaciones en el eje de giro. En [5], proponen la técnica de “aumento de la trayectoria de vuelo” en la que se generan muchas trayectorias auxiliares que son ligeramente diferentes de la trayectoria óptima, formando así una envolvente de navegación para así evitar que el VANT sufra grandes desviaciones de trayectoria; en [31] proponen un sistema donde utilizan dos redes, una para la detección del sendero y otra para la detección de obstáculos, siendo este uno de los sistemas pioneros en la navegación en senderos estrechos boscosos y en las pruebas se logró navegar 1 km autónomamente.

También se utilizaron diferentes técnicas de visión artificial para la detección de obstáculos en las investigaciones encontradas. En el trabajo de [4], los árboles son delimitados por cuadros, donde la altura de las ventanas es usada para definir la distancia del VANT al obstáculo detectado y se determina la dirección de vuelo con el espacio libre más amplio; en [31], se utiliza una DNN para la detección de objetos y un componente de odometría visual para estimar la profundidad con el propósito de localizar obstáculos de bajo nivel. En el trabajo [23], se utiliza una CNN para predecir la profundidad y superficie normal formando un modelo 3D, los resultados mostraron que la percepción intermedia aumenta la precisión un 20%, a diferencia de la predicción a partir de imágenes RGB.

La Tabla 2.2 describe a detalle las arquitecturas, los escenarios y los conjuntos de datos utilizados durante las pruebas de las investigaciones mencionadas en esta sección.

Tabla 2.2 Resumen de los artículos relacionados con Aprendizaje Profundo para la detección de obstáculos

Artículo	Arquitectura	Escenario	Síntesis	Conjunto de datos
[4] - 2021	Faster R-CNN	Abierto	El modelo fusiona dos módulos: una CNN que propone regiones y un detector Faster R-CNN con Inceptionv2 que utiliza las regiones propuestas para reconocer los árboles y la profundidad de estos	COCO (Common Object in Context)
[19] - 2021	cGAN + LSTM + RL	Cerrado	El primer cGAN estima un mapa de profundidad a partir de la imagen RGB. Luego, estos mapas alimentan un RL con LSTM para devolver el comando de vuelo	Ambientes interiores simulados
[5] - 2019	CNN	Abierto	Se utiliza una red VGG-16 como clasificador y extractor de características y éstas pasan a una red FCNN o una red GRU para generar el ángulo de orientación de giro del dron	Escenario sintético de AirSim
[29] - 2018	CNN	Abierto	La red calcula la extracción de características para detectar fallas y aprende a identificar trayectorias seguras. En el caso de que ocurra alguna falla en el sistema de percepción visual se ejecuta una secuencia de comandos para restablecer el VANT a un estado seguro.	ImageNet
[24] - 2018	CNN	Abierto	Se utilizó la red AlexNet para clasificar los árboles en tres clases. Se tiene acoplado una máquina de estados para obtener los comandos de vuelo.	Imágenes simuladas y reales
[25] - 2018	CNN	Abierto	Se busca reducir el tiempo de procesamiento de las imágenes por lo que se utiliza una CNN de tamaño pequeño, la cual está parcialmente compartida para dos tareas, y en las últimas capas se separa para predecir la dirección del ángulo y la probabilidad de colisión	Udacity e imágenes recolectadas por una bicicleta
[26] - 2018	CNN	Abierto	Combina una convolución en profundidad, seguida de una convolución en puntos de 1x1, y después de cada bloque hay una activación de ReLU	ImageNet-2012 y NCTU
[30] - 2018	CNN de dos flujos con regresión	Cerrado	La red hace uso de dos imágenes RGB consecutivas y devuelve la distancia a los obstáculos en interiores, en tres direcciones	Imágenes recolectadas de vuelo en interiores
[23] - 2017	CNN	Cerrado	Se utiliza una CNN para predecir la profundidad y superficie normal para formar un modelo 3D y después otra CNN para predecir la trayectoria.	NYUv2 RGB-D
[27] - 2017	DenseNet	Abierto	La arquitectura utiliza un patrón de conectividad para garantizar el máximo flujo de información entre las capas de la red, concatenando las características	CIFAR, SVHN e ImageNet
[31] - 2017	CNN	Abierto	Usa dos redes: una para detección del sendero y otra para la detección de obstáculos	Senderos de los Alpes suizos IDSIA
[28] - 2016	Inceptionv3	Utilizan Imagenet	Inception-v3 es una arquitectura de la familia Inception con varias mejoras, incluido el uso de suavizado de etiquetas, convoluciones factorizadas de 3x3 y el uso de un clasificador auxiliar para propagar información de etiquetas en la parte inferior de la red.	ImageNet ILSVRC 2012

2.3 Aprendizaje profundo para la estimación de profundidad

De las investigaciones encontradas relacionadas con la detección de obstáculos, se revisó el desempeño de estas y se tiene que en [23] se obtuvieron los mejores resultados de precisión con una mejora del 20% al crear un modelo 3D para predecir la profundidad.

Por lo antes mencionado, se realizó una búsqueda de las investigaciones enfocadas en la estimación de profundidad a partir de imágenes monoculares. La mayoría de los artículos analizados fueron publicados entre 2019 y la actualidad, lo que pone de manifiesto el gran interés que suscita este tema.

Las investigaciones encontradas se pueden dividir por el tipo de aprendizaje utilizado en: supervisado, semi-supervisado y auto-supervisado. Los métodos supervisados aceptan una sola imagen y la información de profundidad correspondiente para el entrenamiento; sin embargo, se requiere una gran cantidad de datos de alta calidad y es difícil generalizar para todos los casos. Para no depender de grandes conjuntos de datos se han propuesto los métodos semi-supervisados, estos requieren una menor cantidad de datos etiquetados y una gran cantidad de datos no etiquetados para el entrenamiento; por otra parte, estos no pueden corregir su propio sesgo y requieren información adicional. Los métodos auto-supervisados sólo requieren una pequeña cantidad de imágenes sin etiquetar para el entrenamiento, ya que obtienen la información de profundidad automáticamente relacionando las diferentes modalidades de entrada; estos modelos funcionan en escenarios similares al conjunto de entrenamiento.

De los métodos supervisados se tiene que [32] propone estimar mapas de profundidad continuos, utiliza mapas de Hilbert para generar un mapa denso de profundidad a partir de puntos LiDAR y aparte aprovechan la red FCRN (siglas en inglés *Fully Convolutional Residual Network*) para la estimación de profundidad; aunque solo puede producir mapas en imágenes con resolución de 128x160 píxeles y la red sufre de sesgo debido a la salida del proceso de densificación de los mapas de Hilbert. En [33] utilizan un enfoque de discretización que aumenta el espaciado (SID, *Spacing-Increasing Discretization*) adoptando una estructura de red multi-escala que evita la agrupación espacial innecesaria y captura la información multi-escala en paralelo. El trabajo de [34] se enfoca en la estimación de profundidad monocular en videos, la propuesta se basa en una arquitectura CNN que utilizan dos módulos uno de estimación de profundidad y otro para el movimiento de cámara.

Continuando con los métodos supervisados, se tienen investigaciones con arquitecturas codificador y decodificador. En [35] utilizan una arquitectura residual profunda como codificador y un módulo de auto atención en el decodificador para controlar la escala espacial donde se crea una relación entre cada pixel que da como resultado la información contextual, teniendo así la estimación continua de la profundidad densa a nivel de píxeles. En el trabajo [36] proponen un método supervisado para estimar la profundidad con la ayuda de la transferencia de aprendizaje; utiliza una arquitectura basada en *DenseNet* previamente entrenada para la extracción de características y la información obtenida pasa a un decodificador para calcular los mapas de profundidad. En [37] proponen aprovechar las restricciones geométricas 3D, consideran un concepto simple denominado “virtual normal” y mediante una pérdida definida en “virtual normal” se obtiene una nube de puntos 3D de la escena para realizar la estimación de profundidad. En [38] utilizan nuevas capas de guía planas locales insertadas en la fase de decodificación, además de un extractor de características densas y de información contextual para la estimación de profundidad. En [39] se enfocan en un

diseño de baja latencia en donde el codificador extrae características de baja resolución y de alto nivel de la imagen de entrada y el decodificador que consta de 6 capas de convolución separable con conexiones de salto se encarga de ampliarlas gradualmente.

Sólo se encontró un trabajo donde proponen un método semi-supervisado; en [40] emplean transformaciones de escenarios sintéticos a realistas y de realistas a sintéticos junto con una pérdida de consistencia geométrica basada en la geometría epipolar de las imágenes estéreo reales. Este enfoque aborda el problema de la generalización de los métodos de estimación de profundidad al entrenar el modelo con datos sintéticos para estimar la profundidad en imágenes naturales.

De los métodos auto-supervisados se consideraron ocho trabajos en donde se tiene que en [41] propone una red siamesa para predecir los mapas de disparidad entre la imagen izquierda y la derecha invertida, en donde se comparten los pesos para reducir las operaciones y el tamaño de la red. En el trabajo [42] emplean un método de odometría visual para obtener puntos 3D y mapas de escasa profundidad que se envían a un codificador para obtener un mapa de profundidad más denso. En [43] utilizan un auto-codificador que se encarga de codificar la entrada en un código comprimido que captura suficiente información para después ser reconstruida con un error mínimo para obtener así la imagen de profundidad correcta. En [44] utilizan una red convolucional multiescala en donde se van refinando las predicciones progresivamente usando una secuencia de escalas, en el modelo se realizan predicciones de profundidad y de superficie; se realiza un etiquetado semántico que junto con la información anterior proporciona un relato completo de la escena.

Otros trabajos con métodos auto-supervisados utilizan redes CNN; en [45] utilizan un emparejamiento residual monocular de extremo a extremo, en donde la red considera características multi-escala para encontrar un mapa de profundidad y en la etapa final realiza un refinamiento de disparidad. En [46] utilizan bloques de empaquetado y desempaquetado simétricos para combinar la información codificada y decodificada usando convoluciones 3D, preservando información espacial importante. En [47] proponen una pérdida de reproyección mínima para hacer frente a las oclusiones entre fotogramas, una pérdida de auto-enmascaramiento para ignorar píxeles confusos y un método de muestreo de resolución múltiple; demostrando que juntos crean un modelo simple y eficiente. En [48] diseñan una red neuronal profunda que toma como entrada la vista del ojo izquierdo, estima un mapa de disparidad suave y luego genera una imagen nueva para el ojo derecho; con esto se predice una disparidad probabilística que es utilizada para modelar la representación basada en imágenes de profundidad.

Para obtener los resultados de las métricas de rendimiento en el 68% de las investigaciones utilizan el conjunto de datos KITTI Depth y en los restantes el conjunto de datos NYU Depth; observando que en [34] se tienen los mejores resultados utilizando KITTI Depth donde AbsRel=0.037, SqRel=0.174, RMSE=2.005 y RMSElog=0.074; para NYU Depth en el trabajo [39] se obtuvo el mejor resultado con RMSE=0.604.

En la Tabla 2.3 se detallan las arquitecturas, los conjuntos de datos y el hardware utilizado para el procesamiento de las arquitecturas consideradas en esta sección.

Tabla 2.3 Resumen de los artículos relacionados con Estimación de Profundidad

Artículo	Método	Síntesis	Resultados	Implementado	Conjunto de datos
[35] 2021	ACAN	Utiliza la arquitectura residual profunda y un modelo de auto atención para controlar la escala espacial y la estimación continua de la profundidad densa a nivel de píxeles	Utilizando el dataset KITTI se tienen resultados de AbsRel=0.083, SqRel=0.437, RMSE=3.599 y RMSElog=0.127	Nvidia GTX1080Ti	KITTI y NYU Depth v2
[34] 2020	DeepV2D	Se compone de algoritmos geométricos clásicos convertidos en módulos de red diferenciables; la cual contiene dos etapas: estimación del movimiento de la cámara (Flow-SE3) y estimación de profundidad	Utilizando KITTI se tienen resultados de AbsRel=0.037, SqRel=0.174, RMSE=2.005 y RMSElog=0.074	No indica	NYU, ScanNet, SUN3D y KITTI
[32] 2019	EMDEOM	Se utilizó una Red Residual Totalmente Convolutiva basada en la topología ResNet-50, donde las capas completamente conectadas han sido reemplazadas por un conjunto de proyecciones ascendentes	Utilizando KITTI se tienen resultados de AbsRel=0.119, SqRel=0.444, RMSE=3.097 y RMSElog=0.180	GPU NVIDIA Titan X con 12 GB de memoria	KITTI Depth y KITTI Continuous
[47] 2019	monodepth2	Modelo auto-supervisado con pérdida mínima de reproyección, calculada para cada píxel, para hacer frente a las oclusiones entre fotogramas en un video monocular, una pérdida de enmascaramiento automático para ignorar píxeles confusos, estacionarios, y un método de muestreo de múltiples escalas de resolución completa	Utilizando KITTI 2015 se tienen resultados de AbsRel=0.116, sqRel=0.918, RMSE=4.872 y RMSElog=0.193	NVIDIA TITAN Xp	KITTI
[42] 2019	VOMonodepth	Arquitectura codificador-decodificador auto-supervisada que utiliza un algoritmo tradicional de VO (visual odometry) para obtener el conjunto de puntos 3D para usarlos como entrada adicional y obtener estimaciones más precisas.	Utilizando KITTI se tienen resultados de AbsRel=0.091, SqRel=0.548, RMSE=3.690 y RMSElog=0.181	GPUs, NVIDIA Jetson TX2	KITTI
[40] 2019	GASDA	La arquitectura consiste de 6 sub-redes, que se pueden dividir en 3 grupos: Gs2t y Gt2s (generadores) para traducción de estilo de imagen, Dt y Ds para discriminación, Ft y Fs para estimación de profundidad monocular (codificador-decodificador con saltos de conexión)	Utilizando KITTI se tienen resultados de AbsRel=0.149, SqRel=1.003, RMSE=4.995 y RMSElog=0.227	No indica	KITTI y Make3D
[45] 2019	monoResMatch	Arquitectura con un extractor de características multiescala, las cuales se ocupan para estimar la disparidad inicial con un codificador-decodificador, y emulando una configuración binocular	Utilizando KITTI se tienen resultados de AbsRel=0.096, SqRel=0.673, RMSE=4.351 y RMSElog=0.184	GPU Titan XP	KITTI y CityScapes
[41] 2019	Siamese Network	Arquitectura codificador-decodificador donde toma una imagen de entrada es volteada sobre su eje vertical (vertical mirroring). Se utiliza una red siamesa profunda para predecir los mapas de disparidad de la imagen izquierda y la imagen de la derecha invertida	Utilizando KITTI se tienen resultados de AbsRel=0.1069, SqRel=0.653, RMSE=3.790 y RMSElog=0.195	Titan X GPU	KITTI, Make3D y CityScapes

Artículo	Método	Síntesis	Resultados	Implementado	Conjunto de datos
[46] 2019	PackNet-SfM	Arquitectura codificador-decodificador donde sustituyen el submuestreo y muestreo por bloques de empaquetado y desempaquetado para preservar y recuperar importante información espacial para la estimación de profundidad	Utilizando KITTI se tienen resultados de AbsRel=0.071, SqRel=0.359, RMSE=3.153 y RMSElog=0.109	8 Titan V100 GPUs	KITTI, NuScenes CityScapes y DDAD
[38] 2019	BTS	Arquitectura codificador-decodificador supervisada, la red utiliza capas de guías locales planas, lo que proporciona una relación explícita de los mapas de características internos a la predicción deseada para un mejor entrenamiento de la red	Utilizando KITTI se tienen resultados de AbsRel=0.059, SqRel=0.241, RMSE=2.756 y RMSElog=0.096	NVIDIA 1080ti GPUs	KITTI y NYU Depth v2
[39] 2019	FastDepth	Arquitectura de red de codificador-decodificador eficiente con un enfoque en el diseño de baja latencia. El codificador extrae características de baja resolución de alto nivel de la imagen de entrada y estas son ampliadas en el decodificador gradualmente	Utilizando NYU Depth se tienen resultados de RMSE=0.604	Jetson TX2 en modo max-N	NYU Depth v2
[37] 2019	VNL	Arquitectura codificador-decodificador en donde se utilizan las restricciones geométricas 3D para obtener las nubes de puntos y realizar una estimación de profundidad de alta calidad	Utilizando el dataset KITTI se tienen resultados de AbsRel=0.072, RMSE=3.258 y RMSElog=0.117	No indica	KITTI y NYU Depth v2
[36] 2018	DenseDepth	Arquitectura codificador-decodificador sencilla con conexiones de salto	Utilizando KITTI se tienen resultados de AbsRel=0.093, SqRel=0.589, RMSE=4.170 y RMSElog=0.171	4 NVIDIA TITAN Xp GPUs con 12GB memoria	NYU Depth v2, Unreal-1K
[33] 2018	DORN	Red de estimación de profundidad simple que aprovecha la técnica de convolución dilatada y un nuevo codificador de imagen para obtener directamente un mapa de profundidad de alta resolución. El modelo es entrenado con pérdida de regresión ordinal	Utilizando KITTI se tienen resultados de AbsRel=0.071, SqRel=0.268, RMSE=2.271 y RMSElog=0.116	GPU Titan XP	KITTI, NYUDepth, Make3D y ScanNet
[43] 2016	L12	Arquitectura de auto-codificador en donde el decodificador es sustituido por una deformación de imagen geométrica; se entrena la red usando el error de reconstrucción fotométrica	Utilizando KITTI se tienen resultados de AbsRel=0.177, SqRel=0.1.169, RMSE=5.285 y RMSElog=0.282	No indica	KITTI
[48] 2016	Deep3D	Sintetiza una vista de la otra, y usa el error de reconstrucción para producir una distribución de probabilidad de posibles disparidades para cada píxel. Utiliza la estrategia de salto de conexión para fusionar el mapa de profundidad de baja resolución con el mapa de profundidad de alta resolución espacial en capas inferiores.	Comparan el error de reconstrucción por píxeles utilizando el error absoluto medio con resultados de 5.47	NVidia GTX Titan X GPU	KITTI y NYU Depth v2
[44] 2015	Multi-scale Convolutional Architecture	Utiliza redes multi-escala que refinan el mapa de profundidad estimado desde una resolución espacial baja hasta una resolución espacial alta a través de redes independientes	Utilizando NYU Depth se tienen resultados de RMSE=0.641	No indica	NYUDepth, RGB-D, Sift Flow

2.4 Estado de la práctica

Las investigaciones relacionadas con la navegación autónoma de VANTs actualmente se han centrado en aportaciones con entornos interiores, urbanos y zonas boscosas, o que sobrevuelan en altas altitudes un área cultivada de terreno para realizar las tareas de censo de árboles, deficiencias de nutrientes, estrés hídrico, plagas o identificación de hierba.

A pesar de que la mayoría de los estudios realizados en la navegación autónoma de VANTs no consideran el vuelo en cultivos a bajas altitudes y en espacios reducidos y complejos, se han obtenido resultados relevantes:

- Se ha logrado la navegación autónoma en senderos boscosos, implementando dos redes, una para la detección del sendero y otra para la detección de obstáculos usando odometría visual monocular [31].
- Se plantearon redes donde se reduce el costo computacional y el tamaño de la red: se tienen redes que obtienen simultáneamente la probabilidad de predicción y el ángulo de giro permitiendo que el VANT pueda reaccionar rápido a los obstáculos [25]; también se han reducido los números de parámetros a 2.23 millones logrando una precisión del 80% consiguiendo así implementar el sistema en el VANT [26].
- La precisión en la predicción de obstáculos se mejoró en un 20% creando un modelo 3D a partir de imágenes RGB, demostrando que sin reentrenamiento el VANT es capaz de evitar obstáculos en tiempo real [23].
- En el entrenamiento de la red se han utilizado imágenes simuladas y reales; teniendo en el trabajo de [24] que en 10 vuelos realizado en ambientes reales se obtuvo una tasa de éxito del 100% en la evasión de obstáculos, demostrando que con la base de datos simulados se tiene aportaciones en el aprendizaje de la red.

De lo anterior se tiene que los escasos trabajos realizados en ambientes con árboles estos se enfocan principalmente en la detección de los troncos gruesos sin considerar otros factores que pudieran afectar la navegación del VANT en pequeños senderos tales como distintos tipos de ramas y hojas, postes de electricidad, personal que labora en las zonas de cultivos o maquinaria agrícola.

Al mismo tiempo se detectó que, actualmente, los conjuntos de imágenes utilizados en el estado del arte no corresponden a cultivos en bajas altitudes, siendo esto el conjunto de datos que se requiere para esta investigación. Los conjuntos de imágenes de trabajos previos que se podrían considerar son NYU-v2 [10], KITTI [49] o IDSIA [31] para las pruebas iniciales, en específico IDSIA al contar con imágenes de senderos forestales.

Además de los distintos conjuntos de datos utilizados en el entrenamiento en trabajos anteriores, en esta investigación también pueden ser consideradas imágenes simuladas en Gazebo [24] o escenarios sintéticos de zonas boscosas en AirSim [5].

CAPÍTULO 3

Marco teórico

En el presente capítulo se presentan los temas que dieron base a este tema de tesis y se describen las técnicas utilizadas en esta investigación, así como las métricas para evaluar los resultados de los algoritmos utilizados para el desarrollo de este trabajo.

3.1 Agricultura de precisión

Aunque los avances científicos en genética, química y robótica han contribuido significativamente a la evolución de la tecnología agrícola, todavía se tienen algunos retos que atender en el sector de la agricultura como el cambio climático, la limitada disponibilidad de áreas de cultivo y la creciente necesidad de agua dulce [1].

Una solución factible es el uso de servicios de Tecnologías de la Información y la Comunicación (TICs) para agregar y procesar información proporcionada por múltiples fuentes que puedan extraer conclusiones útiles sobre la comprensión del suelo, permitiendo así manejar los cultivos de una manera más eficiente. El uso extendido de las tecnologías del Internet de las Cosas, como las redes de sensores inalámbricos y los VANTs, hace posible el despliegue de aplicaciones de precisión agrícola que pueden calcular índices de vegetación eficientes, optimizando así el rendimiento de los cultivos [50].

La Agricultura de Precisión (AP) constituye un nuevo método de gestión de cultivos mediante el cual productos agroquímicos como plaguicidas, fertilizantes y agua de riego se aplican proporcionalmente en función de las necesidades específicas de los cultivos, ya que pueden variar espacial y temporalmente. Los objetivos principales de la AP son [1]:

- incrementar el rendimiento de los cultivos,
- mejorar la calidad de los productos,
- hacer un uso más eficiente de los productos agroquímicos,
- ahorrar energía,
- proteger el medio ambiente físico contra la contaminación.

Un requisito fundamental para la utilización de tecnologías de AP es el conocimiento de la variabilidad espacial y temporal [50]. El papel de la variabilidad espacial es identificar las variables de medición específicas para supervisar las variaciones de las características del cultivo, como la vegetación, el estado del agua, la humedad, la composición del suelo, la topografía, así como el estado de las enfermedades y plagas de las plantas. Por otro lado, la variabilidad temporal apunta a identificar características particulares que afectan el rendimiento del cultivo. Esta información permite al productor dividir un cultivo en subáreas de manejo específicas, en las que se pueden aplicar cantidades variables de insumos, optimizando así el desempeño general mediante el aumento de la producción y la reducción de insumos agroquímicos.

3.2 Vehículos aéreos no tripulados (VANTs)

VANT es un tipo de aeronave que tiene la capacidad de volar sin la presencia de tripulación. Comúnmente, la misión de vuelo del VANT está predefinida, o una persona puede controlar su movimiento y dirección a través de comandos de teleoperación remota desde una estación terrestre [51].

Esta tecnología ha evolucionado rápidamente en el siglo XXI, los primeros intentos de desplegar VANTs se iniciaron con fines militares a partir de la Primera Guerra Mundial. Aunque los primeros vehículos aéreos no tripulados se referían a operaciones militares, la rápida evolución de nuevas tecnologías como los sensores de imágenes, la unidad de medición inercial (IMU, siglas en inglés *Inertial Measurements Unit*), el radar de apertura sintética y los sistemas globales de navegación por satélite (GNSS, siglas en inglés *Global Navigation Satellite Systems*) dieron lugar al desarrollo de VANTs civiles capaces de ayudar a la evolución de múltiples campos, como AP, geomática, logística y monitoreo de infraestructura.

Conociendo los VANTs [1]

Por sus características aerodinámicas se tienen tres categorías de vehículos aéreos no tripulados civiles: a) de ala fija, b) de ala giratoria o multi-rotor y c) híbrido. El tipo ala fija posee un perfil aerodinámico predefinido de alas fijas y estáticas que permiten la elevación basada en la velocidad aerodinámica de avance del VANT. Para el tipo de ala giratoria el flujo de aire está compuesto por varios rotores que generan la potencia adecuada necesaria para la elevación, dependiendo del número de rotores se pueden clasificar en las siguientes categorías: a) tricópteros, b) cuadricópteros, c) hexacópteros y d) octocópteros.

El tipo de ala giratoria posee un control mejor y más fácil y, es capaz de transportar una carga útil más pesada en comparación con el tipo de ala fija. Por otro lado, el tipo de ala fija presenta una arquitectura eficiente y más simple que facilita los procesos de mantenimiento y también se caracteriza por una mayor duración de vuelo y mayor cobertura. Sin embargo, necesita de un área grande libre de obstáculos para poder realizar el despegue de vuelo o el aterrizaje.

Dependiendo del nivel de autonomía se tienen cuatro tipos, según el Departamento de Defensa de los Estados Unidos [52]. El primer tipo, denominado sistema operado por humanos, define que el operador del sistema es responsable de controlar todas las operaciones del sistema no tripulado. El segundo tipo llamado sistema humano delegado se caracteriza por un mayor nivel de autonomía en comparación con el primero, al mantener la capacidad de tomar de forma autónoma algunas decisiones restringidas. El tercer nivel se denomina sistema supervisado por humanos y puede tomar varias decisiones según las instrucciones del operador del sistema. Finalmente, el último nivel se denomina sistema totalmente autónomo y se encarga de todas sus operaciones.

Basándose en el tamaño y peso se tiene que si un VANT excede el peso de 150 kg entonces su categoría es pesado, aquellos VANTs cuyo peso es entre 20 kg y 150 kg su categoría es grande. Los VANTs de ala fija que no excedan los 20 kg y los VANTs de ala giratoria que no excedan los 25 kg son considerados pequeños y por último los VANTs cuyo peso va desde gramos hasta algunos kilos su categoría es mini.

También se clasifican los tipos de VANTs dependiendo de su fuente de energía en cuatro diferentes: a) queroseno, b) celdas de batería, c) celdas de combustible y d) celdas solares [53].

Los tipos de cargas útiles que se pueden integrar en un VANT dependen de su tamaño y peso. El sensor integrado más utilizado en los VANTs es una cámara. Hay tres tecnologías principales de cámaras: a) multiespectral, b) hiperespectral y c) térmica. Otros tipos de sensores que se pueden integrar en un VANT son químicos, sensores biológicos y meteorológicos. Cabe mencionar que hay muchas cargas útiles que no pertenecen a la categoría de sensores, por ejemplo, un VANT puede llevar un sistema de rociado.

3.3 Aplicaciones de los VANTS en agricultura de precisión

El cambio climático ha afectado significativamente a múltiples sectores, incluida la seguridad alimentaria, ya que actualmente se tienen más de 815 millones de personas que padecen hambre crónica [54]. Por lo tanto, las industrias alimentarias, las comunidades agrícolas y los científicos vinculados al sector agrícola deben implementar nuevos procesos o adaptar adecuadamente los existentes para hacer frente a los desafíos introducidos por el cambio climático.

Debido a esto, la agricultura tiene que adoptar servicios emergentes basados en Tecnologías de la Información y la Comunicación (TIC) que proporcionan datos fiables, precisos y oportunos; un ejemplo de este avance es el desarrollo de los VANTs agrícolas, que pueden proporcionar datos valiosos sobre la vegetación y los atributos químicos, lo que influye en las decisiones y políticas relevantes.

Cabe mencionar que la Asociación Internacional de Sistemas de Vehículos No Tripulados (AUVSI por sus siglas en inglés), menciona que el 80% de los VANTs se utilizarán para fines agrícolas en el futuro cercano [1].

En el artículo “A compilation of UAV applications for precision agriculture”[1] realizan el análisis de 20 diferentes aplicaciones de los VANTs en la agricultura y éstas se pueden dividir en tres categorías:

a) Aplicaciones de monitoreo

Los VANTs son usados para monitorear los cultivos y proveer imágenes que luego son procesadas para extraer alguna propiedad en particular e índices de vegetación y así identificar problemas en los cultivos que sufren de enfermedades y plagas. La información recibida por los sensores puede ser espectral, espacial y temporal; la selección del tipo de sensor apropiado depende de la aplicación.

De las diferentes aplicaciones analizadas en el artículo se tiene que 16 son en el monitoreo de cultivos, teniendo diferentes objetivos como son: la detección de tuberías de drenaje, monitoreo del estado de la vegetación (trigo, uva, yuca, camote, granada, espárrago, cítricos y caña de azúcar), evaluar el estrés hídrico, estimar la densidad de clorofila, determinar la concentración de nitrógeno e identificar la variación de arroz durante sus diferentes etapas fenológicas.

Para el monitoreo en la mayoría de los trabajos se utilizan cámaras multiespectrales para estimar el estado de la vegetación, mientras que sólo en tres se utiliza la cámara térmica para evaluar el estrés hídrico y en dos se utiliza la cámara hiperespectral para calcular los atributos químicos, como el cálculo del estado del nitrógeno y la densidad de la clorofila.

b) Aplicaciones de fumigación

En esta categoría los VANTs son utilizados para realizar la labor de fumigación sin intervención humana, previendo así problemas como desastres ambientales y enfermedades humanas como el cáncer, trastornos neurológicos y complicaciones en el sistema respiratorio. En la mayoría de los casos se trabaja en conjunto con una Red de Sensores (WSN por sus siglas en inglés) localizada en el área objetivo, la WSN transmite información al VANT relacionada al clima y la dirección y velocidad del aire; el VANT empleando un algoritmo toma las decisiones adecuadas acerca de la posición y la velocidad a utilizar.

También se tienen aplicaciones de fumigación basadas en sistemas de visión, especialmente diseñados para árboles de frutas. Esto incluye un sistema de navegación basado en el Filtro de Kalman Extendido (EKF por sus siglas en inglés), un sistema de control y un sistema de visión. El sistema de visión se integra de tres fases: pre-procesamiento, localización e identificación de objetivos. Durante la identificación de objetivos se utiliza el algoritmo de clasificación K-Vecinos más Cercanos (KNN por sus siglas en inglés) para localizar el área objetivo con más precisión y así optimizar el proceso de fumigación para rociar áreas específicas con alta precisión.

c) Aplicaciones multi-VANT

La mayoría de los trabajos existentes se enfocan en el uso de un VANT, sin embargo, en algunos casos como en grandes cultivos, un solo VANT no puede completar el trabajo debido a la limitada fuente de energía. Por el contrario, una aplicación multi-VANT es capaz de abordar este desafío, al separar el área en múltiples subáreas correspondientes al número de VANTs.

En el trabajo [55] cada VANT se encarga de monitorear un número de cuadrantes y toma en consideración dos criterios: los cuadrantes sin monitorear por largo tiempo y la complejidad de la ruta basada en la distancia y el número de turnos. Cada VANT tiene la habilidad de comunicarse con los otros para así poder intercambiar cuadrantes, minimizando así el tiempo de finalización de las rutas.

3.4 Detección de obstáculos para la navegación de VANTs

El uso de VANTs en la agricultura de precisión tiene numerosas ventajas, como el monitoreo y fumigación; sin embargo, también existen algunos desafíos importantes que deben abordarse. Entre esos desafíos, se consideran de gran importancia la gestión de interferencias y la planificación de rutas.

En cuanto a la gestión de interferencias, los hallazgos revelaron que la altitud de los drones debe ajustarse de acuerdo con los requisitos de cobertura de sus antenas direccionales; por lo que lograr que los VANTs puedan volar a bajas altitudes ayudaría a reducir las interferencias, compensando así el retraso y la cobertura [56].

En lo que respecta a la planificación de rutas, encontrar la ruta de vuelo óptima para un VANT se considera un objetivo desafiante, ya que se ve afectado por múltiples factores, como las limitaciones de energía, el tiempo de vuelo y la evitación de obstáculos.

Para atender los desafíos antes mencionados y lograr cierto nivel de autonomía en los VANTs se deben considerar múltiples condiciones variables, para nuestro caso se consideran: terrenos irregulares, caminos estrechos con diferentes obstáculos como: maquinaria de riego, postes de

electricidad, maquinaria agrícola y personal que trabaja en las áreas de cultivo; por lo tanto, una de las principales funcionalidades que deben tener los VANTs es que sean capaces de esquivar obstáculos para evitar colisiones. Para lograr este objetivo se han propuesto varios métodos utilizando sensores sofisticados como RGB-D, LiDAR, RADAR, cámaras estereoscópicas, escáneres sonares y sensores infrarrojos; sin embargo, estos métodos tienen un alto costo computacional y los sensores son costosos, pesados y con alto consumo de energía.

Considerando todo lo anterior, el reto es lograr que los VANTs puedan volar de forma autónoma utilizando sensores de bajo costo, livianos y de bajo consumo energético. Una alternativa es utilizar técnicas de Aprendizaje Profundo con métodos de visión artificial, que involucran tareas como el flujo óptico o la estimación de profundidad a partir de datos adquiridos por cámaras monoculares; lo que resulta una opción considerablemente más económica que los sensores mencionados anteriormente.

En los últimos años el Aprendizaje Profundo (DL) ha demostrado ser una excelente técnica en el área de la Inteligencia Artificial, ha logrado resolver diferentes problemas produciendo buenos resultados en áreas como reconocimiento de imágenes, imágenes médicas y reconocimiento de voz. Revisando el estado del arte se tiene que la arquitectura más ampliamente utilizada de DL son las Redes Neuronales Convolucionales (CNN) en el contexto de prevención de colisiones [16], especialmente en las etapas relacionadas con la percepción del entorno y estimación de profundidad [17].

Para la navegación del VANT se tiene que considerar que cada observación está relacionada con una representación parcial del entorno que rodea al VANT; por lo que tener la posibilidad de almacenar y rastrear toda la información relevante obtenida en el pasado permite tener una representación más completa del entorno. Para este propósito se utilizan las Redes Neuronales Recurrentes (RNN) [19] que brindan la capacidad de aprender dependencias temporales a partir de la información de una secuencia de observaciones arbitrariamente largas.

Analizando algunos trabajos del estado del arte se encontraron diferentes técnicas de visión artificial para detectar obstáculos, en los cuales resaltan los que se mencionan a continuación:

- En [4] los árboles están delimitados por cuadrados, donde se utiliza la altura de las ventanas para definir la distancia del UAV al obstáculo detectado y la dirección de vuelo donde se determina el espacio libre más amplio.
- En [31] utilizan una Red Neuronal Profunda (DNN) para la detección de objetos y un componente de odometría visual para estimar la profundidad con el fin de localizar obstáculos de bajo nivel.
- En el trabajo [23] utilizan una CNN para predecir la profundidad y la superficie normal formando un modelo 3D, en los resultados se demostró que la percepción intermedia aumenta la precisión en un 20% a diferencia de la predicción a partir de imágenes RGB.

3.5 Técnicas de visión por computadora para navegación

Con el rápido desarrollo de la visión por computadora se tiene que la navegación basada en la visión demuestra ser una dirección de investigación prometedora para la navegación autónoma. Primero, los sensores visuales pueden proporcionar abundante información del entorno; en segundo lugar, son muy apropiados para la percepción del entorno dinámico debido a su notable capacidad

antiinterferente; en tercer lugar, la mayoría son pasivos, lo que previene que este tipo de sistemas sean detectados [57].

El principio básico para evitar obstáculos es detectarlos y calcular las distancias entre el VANT y ese impedimento. Una de las soluciones es medir la distancia utilizando telémetros, como radar, ultrasónico e infrarrojo, etc.; pero estos no pueden obtener suficiente información en ambientes complejos debido a su campo de visión limitado y rango de medición. Existen principalmente dos tipos de métodos para evitar obstáculos [58]:

- Los métodos basados en flujo óptico son capaces de generar un flujo de información local y obtener la profundidad de la imagen.
- Los métodos basados en SLAM pueden proporcionar mapas métricos precisos con un algoritmo SLAM sofisticado, por lo que los VANTs pueden navegar y evitar obstáculos con más información del entorno

El concepto de estimación de profundidad se refiere al proceso de preservar la información 3D de la escena utilizando información 2D capturada por cámaras. Hay una variedad de dispositivos disponibles comercialmente para proporcionar información de profundidad; sin embargo, su potencia de procesamiento, tiempo de cálculo, limitación de rango y costo los hacen poco prácticos para dispositivos de consumo [59].

La mayoría de los métodos binoculares o de múltiples vistas pueden estimar información de profundidad bastante precisa. Sin embargo, sus requisitos de memoria y tiempo computacional son desafíos importantes para muchas aplicaciones. La idea de utilizar la imagen monocular para capturar información de profundidad podría resolver potencialmente el problema de los requisitos de memoria, pero es computacionalmente difícil capturar las propiedades globales de una escena, como la variación de textura o la información de desenfoque.

Se ha demostrado en el estado del arte que los métodos de estimación de profundidad monocular podrían ser una solución potencial para abordar muchos de estos desafíos. Estos métodos funcionan con un número relativamente pequeño de operaciones y en menos tiempo de cálculo. No requieren alineación ni calibración, lo cual es importante para los sistemas de medición de profundidad multicámara o multisensor.

La mayoría de los métodos tradicionales de estimación de profundidad se basan en la suposición de observaciones de la escena, ya sea en el espacio o en el tiempo y estos se pueden clasificar en dos conjuntos, métodos activos y pasivos [60].

Los métodos activos implican calcular la profundidad de la escena interactuando con los objetos y el entorno; existen diferentes métodos activos como la estimación de profundidad basada en la luz y métodos que utilizan la velocidad de la onda para medir el tiempo que tarda un pulso emitido en llegar a un sensor de imagen.

Los métodos pasivos aprovechan las características ópticas de las imágenes capturadas, esto implica extraer la información de profundidad mediante el procesamiento de imágenes computacional. Existen dos enfoques principales: (a) estimación de profundidad de múltiples vistas y (b) estimación de profundidad monocular.

3.6 Redes neuronales para la estimación de profundidad monocular

La estimación de profundidad monocular es un desafío fundamental en la visión artificial y tiene aplicaciones potenciales en robótica, comprensión de escenas, reconstrucción 3D e imágenes médicas. Este problema sigue siendo un desafío, ya que no existen pistas confiables para percibir la profundidad a partir de una sola imagen. Por ejemplo, en dichas imágenes faltan información temporal y correspondencias estéreo.

Representando el problema, se tiene $I \in \mathbb{R}^{w \times h}$ es una imagen de tamaño $w \times h$. El objetivo es estimar la información de profundidad correspondiente $D \in \mathbb{R}^{w \times h}$. Este es un problema mal planteado ya que existe una ambigüedad en la escala de la profundidad. Los métodos basados en el aprendizaje supervisado intentan abordar esta cuestión aprendiendo aproximadamente la escala a partir de un conjunto de imágenes de entrenamiento.

Estos métodos definen matemáticamente el problema de la siguiente manera: dado un gran conjunto de datos de imágenes de profundidad y de imágenes RGB (rojo, verde y azul), la estimación de la profundidad de una sola imagen se puede considerar como un problema de regresión que utiliza una función de pérdida estándar, como el error cuadrático medio (MSE).

Recientemente, el avance de las redes neuronales convolucionales (CNN) y los conjuntos de datos disponibles públicamente han mejorado significativamente el rendimiento de los métodos de estimación de profundidad monocular [33], [61]–[63].

Los métodos de aprendizaje profundo mejoraron el rendimiento de los métodos de estimación de profundidad monocular al explorar la información a nivel de imagen y las características jerárquicas en la red. Sin embargo, estos métodos emplean operaciones repetidas de agrupación espacial y para obtener mapas de profundidad de alta resolución, se requieren redes basadas en conexiones de salto; no obstante, estos métodos tienden a complicar el proceso de entrenamiento y requieren más tiempo computacional.

Para abordar los problemas de entrenamiento y tiempo computacional, se han empleado métodos de aprendizaje por transferencia basados en CNN, lo que dio como resultado una estimación de profundidad de alta calidad. La introducción de diferentes metodologías y arquitecturas, como las capas de guía plana local, las redes deconvolucionales multicapa y la pirámide espacial atrous, han llevado el rendimiento de estos modelos al siguiente nivel [59].

3.7 Métricas para la estimación de profundidad

Las métricas cuantitativas más utilizadas para evaluar el rendimiento de los métodos de estimación de profundidad monocular son la diferencia relativa absoluta (AbsRel, por su acrónimo en inglés Absolute Relative Difference), el error cuadrático medio (RMSE, por sus siglas en inglés Root Mean Square Error), el RMSE (log) y el error relativo cuadrático (SqRel, por su acrónimo en inglés Square Relative Error). De acuerdo con [59], las métricas están definidas como:

$$AbsRel = \frac{1}{N} \sum \frac{|d_i - d_i^*|}{d_i} \quad (1)$$

$$RMSE = \sqrt{\frac{1}{N} \sum |d_i - d_i^*|^2} \quad (2)$$

$$RMSE(\log) = \sqrt{\frac{1}{N} \sum |\log d_i - \log d_i^*|^2} \quad (3)$$

$$SqRel = \frac{1}{N} \sum \frac{|d_i - d_i^*|^2}{d_i} \quad (4)$$

$$\text{Precisión con umbral } (\delta < thr): \% \text{ de } d_i \text{ tal que } \max\left(\frac{d_i}{d_i^*}, \frac{d_i^*}{d_i}\right) < thr, \quad (5)$$

$$\text{donde } thr = 1.25, 1.25^2, 1.25^3$$

donde d_i y d_i^* son la verdad del terreno y la profundidad predicha en el píxel i y N es el número total de píxeles.

En [64] proponen la métrica ORD (Error de Relación Ordinal en el espacio de profundidad) para la evaluación de conjuntos de datos cruzados de tiro cero; este error ordinal es una métrica general para evaluar la precisión ordinal de un mapa de profundidad, y se puede usar directamente con diferentes fuentes de ground truth. ORD está definida en la Ecuación 6 en donde ω_i se establece en 1 y las relaciones ordinales ℓ_i y $\ell_{i,\tau}^*$ se calculan utilizando la Ecuación 7; en donde τ es un umbral de tolerancia y p_i^* denota el valor de pseudoprofundidad real.

$$Ord = \frac{\sum_i \omega_i \mathbb{I}(\ell_i \neq \ell_{i,\tau}^*(p))}{\sum_i \omega_i} \quad (6)$$

$$\ell = \begin{cases} +1, & \frac{p_0^*}{p_1^*} \geq 1 + \tau \\ -1, & \frac{p_0^*}{p_1^*} \leq \frac{1}{1 + \tau} \\ 0, & \text{de lo contrario} \end{cases} \quad (7)$$

Además, en [65] proponen una variación del error de relación ordinal el cual llamaron Razón de Desacuerdo de Discontinuidad de Profundidad (D^3R) para medir la calidad de las estimaciones de profundidad de frecuencias altas. En lugar de usar puntos aleatorios como en [64] para la comparación ordinal, se usan los centros de los superpíxeles calculados usando la profundidad del ground truth y se comparan los centroides de los superpíxeles vecinos en las discontinuidades de profundidad. Por lo tanto, esta métrica se centra en la precisión de los límites.

En la Figura 3.1 se muestra un ejemplo de los pares de puntos seleccionados para la métrica D^3R con una imagen de entrada RGB a la izquierda como referencia. Se superponen los superpíxeles creados en azul oscuro y todos los pares de puntos seleccionados en azul claro sobre la profundidad real del suelo en a). En b), se muestran acercamientos de dos regiones diferentes para su visualización. A continuación, se analizan los puntos de error en azul claro para la estimación original en c) y se comparan con los puntos de error (nuevamente en azul claro) después de aplicar los resultados en d) se observa cómo los puntos de error, especialmente alrededor de las hojas de la planta, se reducen drásticamente.

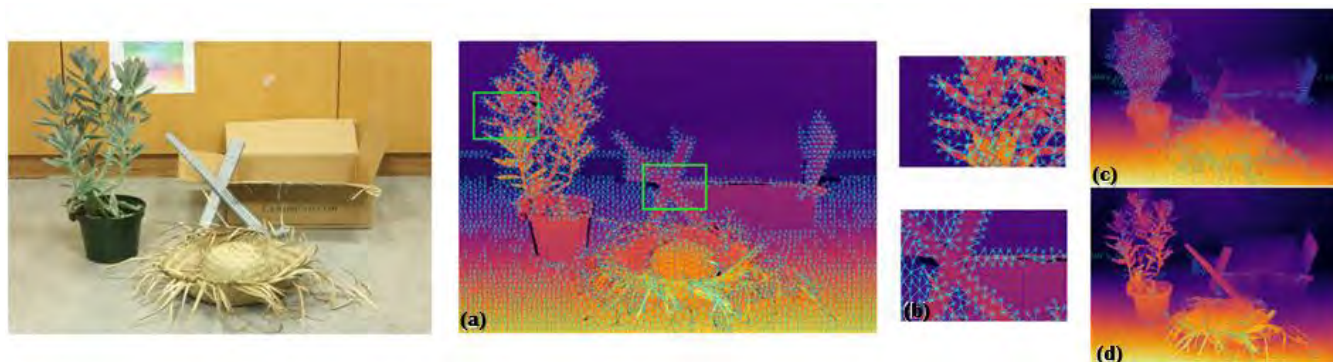


Figura 3.1 Imágenes de [4] como ejemplo para la segmentación de superpíxel creada para la métrica D3R

CAPÍTULO 4

Método de solución

En el presente capítulo se describe el método planteado que se obtuvo a partir del análisis del estado del arte y del marco teórico. En este trabajo de tesis se busca proveer a los VANTs la capacidad de navegar a bajas altitudes en entornos naturales, después de revisar la literatura acerca de la navegación de drones se encontró que la mejor opción de detectar obstáculos en el entorno es implementado aprendizaje profundo, específicamente utilizando redes neuronales para estimar la profundidad.

Por lo mencionado, en los capítulos anteriores se realizó una revisión más enfocada de los últimos artículos publicados relacionados con la estimación de profundidad monocular.

En el trabajo de [66] con la red MiDaS proponen entrenar un robusto módulo de estimación de profundidad monocular con diversos ambientes, para esto desarrollaron nuevas funciones de pérdida que son invariantes a las principales fuentes de incompatibilidad entre conjuntos de datos. En el entrenamiento utilizaron 5 diferentes conjuntos de datos y en los experimentos utilizaron *zero-shot cross-dataset transfer*, demostrando que, con un modelo entrenado en un conjunto rico y diverso de imágenes de diferentes fuentes con un procedimiento de entrenamiento adecuado, ofrecen resultados de última generación en una variedad de entornos.

La red MiDaS ha tenido actualizaciones obteniendo la versión MiDaS v3.0 DPT, en esta utilizan el modelo DPT (siglas en inglés *Dense Prediction Transform*) del trabajo de [67] donde el transformador puede utilizar claves, consultas o valores para confiar completamente en la relación de atención entre unidades para encontrar la relación entre cada unidad en la secuencia.

En el trabajo de [68] proponen una red (R-MSFM) multi-escala recurrente con modulación de funciones auto-supervisadas, en donde se extraen características por píxel y se realiza una modulación de características de escala múltiple y, actualiza iterativamente, una profundidad inversa a través de un decodificador de parámetros compartidos en la resolución fija. Al aprovechar los mapas de características de múltiples escalas se evita la propagación de errores de baja a alta resolución y con la modulación se mantienen representaciones semánticamente más ricas y espacialmente más precisas.

En [65] proponen fusionar estimaciones en diferentes resoluciones con contexto cambiante, para poder generar mapas de profundidad de varios megapíxeles con un alto nivel de detalle utilizando un modelo previamente entrenado, proponiendo así la red Boosting Monocular Depth. Para la experimentación toman dos opciones de modelos MiDaS [66] y SRG [64] para la estimación de profundidad a alta y baja resolución, al obtener ambos mapas estos son combinados utilizando una red de arquitectura Pix2Pix [69] con 10 capas de la red U-net [70] como generador. Después realizaron la implementación del modelo LeReS [71] para la estimación de profundidad, el cual en su momento logró el mejor rendimiento en el estado del arte con 0.09 AbsRel evaluado con el conjunto de datos NYU.

Recientemente en [72] implementan un codificador de transformador jerárquico para capturar y transmitir el contexto global, y diseñaron un decodificador liviano pero poderoso para generar un mapa de profundidad estimado considerando la conectividad local. Mediante la construcción de rutas conectadas entre las características locales multi-escala y el flujo de decodificación global con un módulo de fusión de características selectivas propuesto, la red puede integrar ambas representaciones y recuperar detalles finos.

Considerando los artículos mencionados se realizó una tabla comparativa, ver Tabla 4.1, de las redes de estimación de profundidad que han demostrado tener los mejores resultados, de acuerdo a las evaluaciones encontradas en el estado del arte.

Tabla 4.1 Comparativa de redes de estimación de profundidad a partir de imágenes

Ref.	Red	Tipo de red	Conjunto de datos	Resultados con métricas
[66]	MiDaS	ResNeXt-101 WSL	DIML Indoor, MegaDepth, ReDWeb, WSVD, 3D Movies	Utilizando ETH3D se tiene 0.129 AbsRel
[68]	R-MSFM	ResNet-18 con 3.8M parámetros	KITTI Eigen split	Utilizando KITTI Eigen split se tiene 0.108 AbsRel, 0.748 SqRel y 4.470 RMSE
[65]	Boosting Monocular Depth	Pix2Pix 4 Depth Model	Middlebury 2014 y Ibims-1	Utilizando Middlebury 2014 se tiene 0.1557 RMSE, con Ibims-1 se tiene 0.1598 RMSE
[72]	GLPDepth	Codificador-Decodificador con un módulo SFF (<i>Selective Feature Fusión</i>) utilizando 62M parámetros	NYU Depth V2, SUN RGB-D, KITTI y Ibims-1	Utilizando NYU Depth V2 se tiene 0.098 AbsRel y 0.344 RMSE, con KITTI se tiene 0.057 AbsRel y 2.297 RMSE; por último, con Ibims-1 0.20 AbsRel y 1.01 RMSE

4.1 Descripción de las aportaciones realizadas a la arquitectura propuesta

Para llegar al diseño final de la arquitectura se tomó como base la arquitectura de la red Boosting Monocular Depth. Se propusieron y evaluaron diversas propuestas de mejora, las cuales se citan a continuación:

1. Se utiliza un modelo de fusión en lugar del modelo Pix2Pix, sección 4.1.1.
2. Se modifica la forma en la que se generan los parches, sección 4.1.2.
3. Se modifica la forma en la que se procesan los parches de secuencial a paralelo, sección 4.1.3.

En el trabajo de [65] proponen un algoritmo que llaman doble estimación, en donde alimentan la imagen a la red en dos resoluciones diferentes y fusionan las estimaciones para obtener un resultado consistente con detalles de alta frecuencia, la arquitectura de la red Boosting Monocular Depth se muestra en la Figura 4.1. En resoluciones bajas cercanas a la resolución de entrenamiento, las estimaciones tienen una estructura consistente y carecen de detalles de alta frecuencia. Cuando la misma imagen se alimenta a la red en resoluciones más altas, los detalles de alta frecuencia se capturan mucho mejor mientras que la consistencia estructural de la estimación se degrada gradualmente.

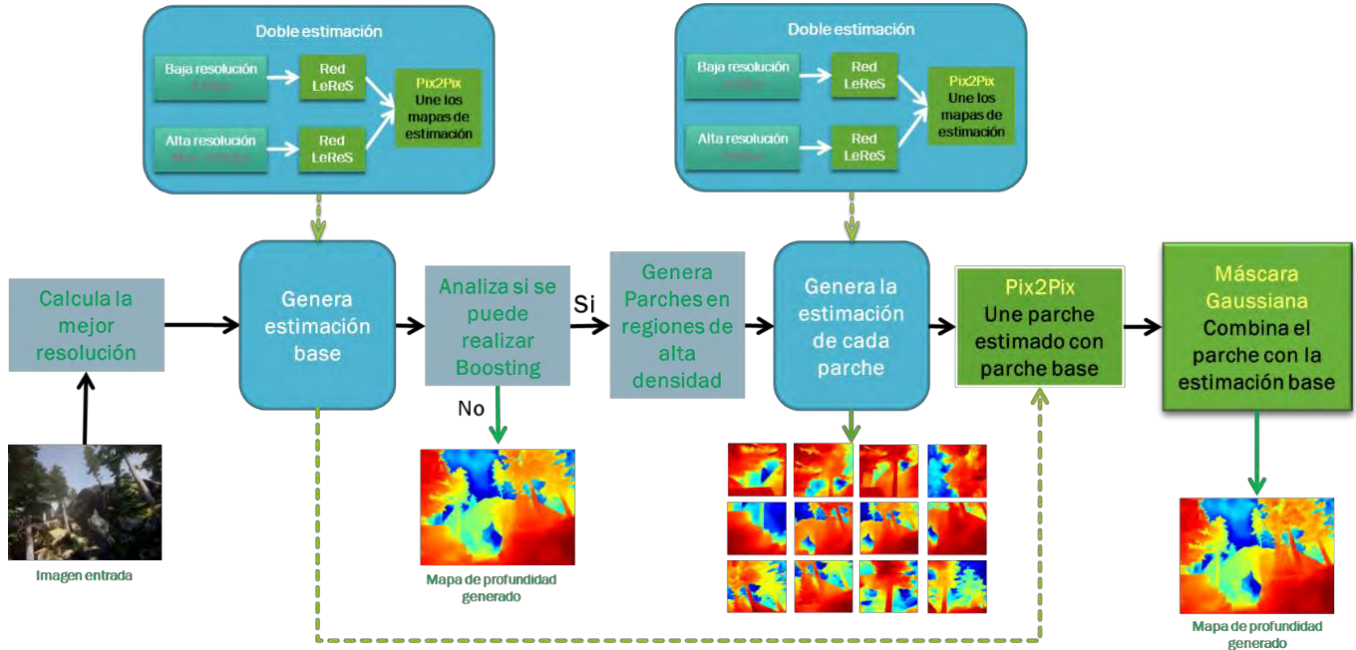


Figura 4.1 Arquitectura de Boosting Monocular Depth [65]

Como la estimación de profundidad monocular se basa en señales contextuales, cuando las señales en la imagen se separan más que el campo receptivo, la red no puede generar una estimación de profundidad coherente alrededor de los píxeles que no reciben suficiente información.

Para combinar las características de ambas imágenes utilizan el modelo Pix2Pix4DepthModel que tiene una arquitectura Pix2Pix [69] con capas U-net [70] como generador y un clasificador convolucional "PatchGAN"(solo penaliza la estructura a escala de parches de imagen) como discriminador. Tanto el generador como el discriminador usan módulos de la forma Convolución-BatchNorm-ReLu, ver Figura 4.2.

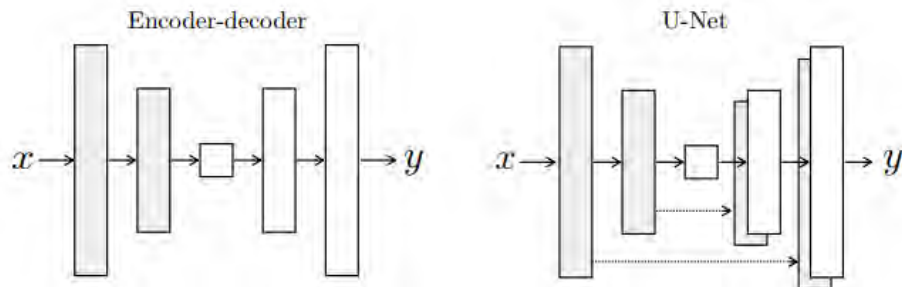


Figura 4.2 Arquitectura del generador "U-Net", codificador-decodificador con conexiones de salto entre capas reflejadas en las pilas de codificador y decodificador [24]

Con la red Boosting Monocular Depth se tienen tres opciones de modelos para realizar la estimación de profundidad: MiDaS [66], SGR [64] y LeReS [71]. De las tres opciones se optó por utilizar LeReS debido a que en pruebas preliminares fue el modelo que mostró mejor definición de detalles finos al obtener el mapa de profundidad.

Analizando la arquitectura de LeReS, la cual se puede observar en la Figura 4.3, se tiene que la red se compone de un módulo de predicción de profundidad (DPM). El cual se compone de una red convolucional que predice los mapas hasta una escala y desplazamiento; y de un módulo de nube de

puntos (PCM) para este después de varias pruebas tomaron el reciente PVCNN. Los módulos son entrenados por separado y luego son combinados para el momento de inferencia, permitiendo así una reconstrucción precisa de la forma de escena.

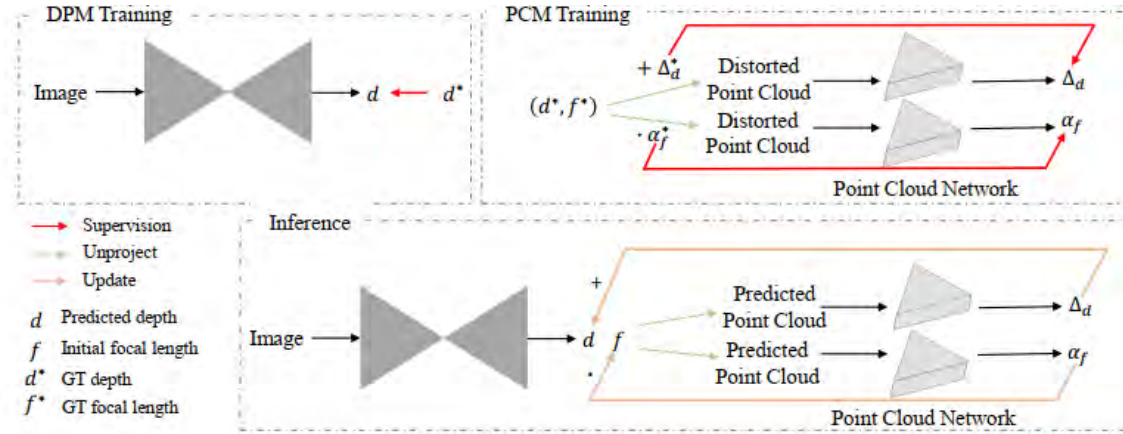


Figura 4.3 Arquitectura de LeReS [25]

Como modelo de predicción de profundidad se utiliza el propuesto en [64] el cual consiste de una red troncal con una arquitectura ResNet50 como extractor de características seguido de un decodificador. En los experimentos utilizaron como red troncal ResNet50 y ResNeXt-101, obteniendo con esta última una mejora en los resultados al ser más grande, ver Figura 4.4.

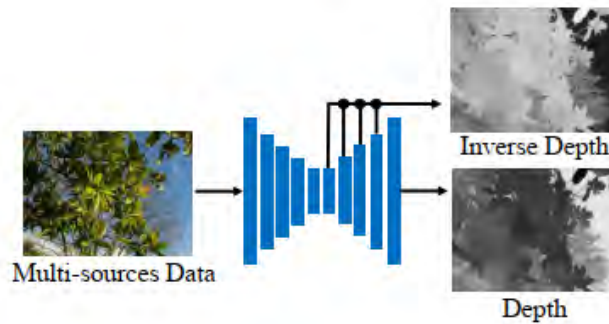


Figura 4.4 Arquitectura del módulo DPM de LeReS [25]

Por lo que la red Boosting Monocular Depth al utilizar LeReS al momento de su ejecución, primero realiza la carga de los modelos Pix2Pixmodel y LeReSmodel, luego se realiza la búsqueda de la mejor resolución de procesamiento de las imágenes. Después, se genera la doble estimación indicando el tamaño del campo receptivo (por default 448) y la óptima resolución de la imagen, aquí se realiza la estimación de profundidad en baja y alta resolución utilizando el modelo seleccionado. Por último, ambas estimaciones creadas son combinadas utilizando Pix2Pix.

La red Boosting Monocular Depth fue entrenada para transferir los detalles de granularidad fina de la entrada de alta resolución a la entrada de baja resolución, para esto se utilizaron los conjuntos de datos Middlebury2014 [73] (pares de imágenes de alta resolución de escenarios interiores) e Ibims-1 [74] (imágenes RGB-D de alta calidad de escenarios interiores).

La red LeReS también utilizó varios conjuntos de datos en el entrenamiento, solo que estos en su mayoría fueron de imágenes RGBD, listándolos son:

- Taskonomy [75] es un conjunto de datos grande y de alta calidad de escenas interiores variadas
- DIML [76] consta de más de 200 escenas interiores y exteriores
- Holopix50K [77] contiene alrededor de 21 000 imágenes de humanos, seguidos de una gran cantidad de animales, vehículos y muebles
- HRWSI [64] tiene 20 mil imágenes variadas web estéreo
- 3D Ken Burns [78] 46 secuencias variadas

Después de realizar diversas experimentaciones se logró obtener el modelo de la arquitectura propuesta, nombrada como *Boosting Pyramidal Depth*, que se puede observar en la Figura 4.5. En donde se tiene como entrada una imagen de 640 x 480 píxeles. A partir de esta se generan los parches, véase la sección 4.1.2 y la imagen base. Después se realiza el proceso de Doble estimación, para ello se crea un batch de baja resolución y un batch alta resolución. Inmediatamente cada *batch* se ingresa en una red preentrenada para obtener los mapas de profundidad. Teniendo los mapas de profundidad de baja y alta resolución se realiza el proceso de Fusión de Mapas, el cual se explica a detalle en la sección 4.1.1. Por último, cada uno de los mapas obtenidos de los parches es combinado en el mapa de profundidad de la imagen base utilizando una máscara gaussiana; para así obtener el mapa de profundidad final de 640 x 480 píxeles de tamaño.

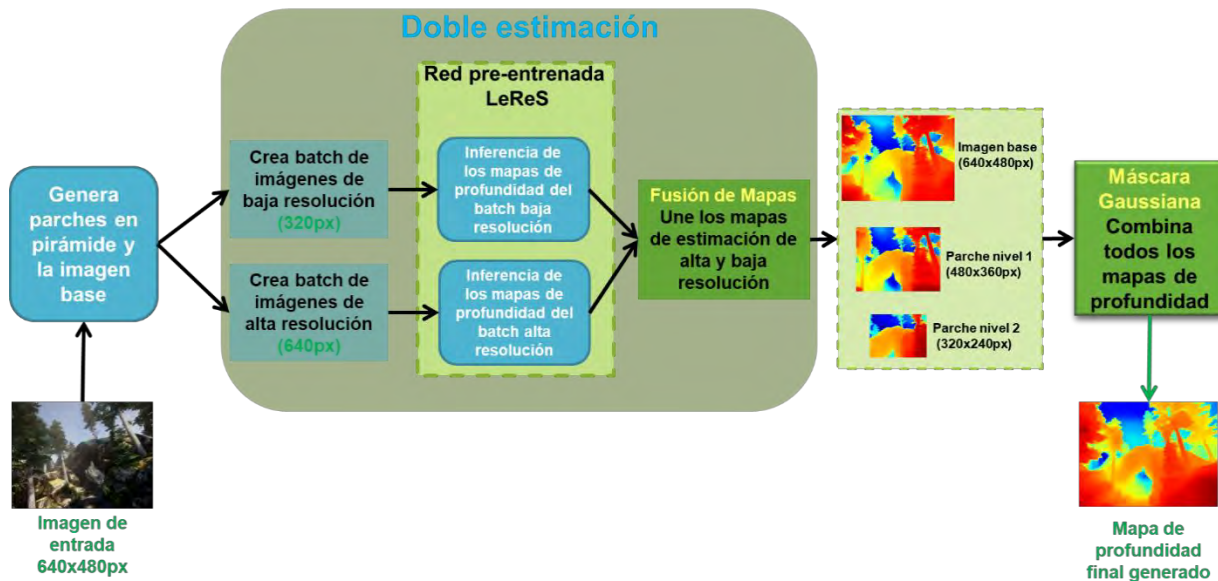


Figura 4.5 Arquitectura propuesta (Boosting Pyramidal Depth)

4.1.1 Fusión de mapas

En la red Boosting Monocular Depth utilizan la red Pix2Pix [69] basada en una cGAN (por sus siglas en inglés *Conditional Generative Adversarial Network*), para realizar la unión de los mapas de baja y alta resolución. El modelo Pix2Pix está compuesto de un generador G que está entrenado para producir resultados que no pueden ser distinguidos de las imágenes "reales" por un discriminador D entrenado adversariamente, que está entrenado para hacer lo mejor posible en la detección de las "falsificaciones" del generador.

Tratando de mejorar la red Boosting Monocular Depth se investigaron alternativas para sustituir el modelo Pix2Pix para unir los mapas de estimación de baja y alta resolución; debido a que este proceso consumía la mayor parte del tiempo de inferencia. Además analizando la experimentación realizada en [69] se logró detectar que el modelo Pix2Pix es capaz de generar resultados parecidos a los reales, pero los píxeles generados no están alineado geoméricamente y en ocasiones puede estar agregando más objetos de los existentes; véase la Figura 4.6.

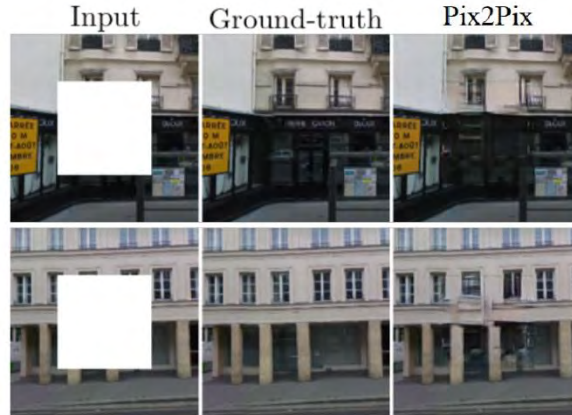


Figura 4.6 Resultados de los experimentos realizados en [23] donde se demuestra que los píxeles generados no están alineado geoméricamente

Se revisó el estado del arte en busca de una mejor solución para llevar a cabo la combinación de los mapas de baja y alta resolución. Se encontró que, en la investigación realizada en [79] para la segmentación de imágenes proponen una combinación de características obtenidas de una rama de red profunda a baja resolución que captura información de contexto global de manera eficiente con una rama poco profunda que se enfoca en detalles de segmentación de alta resolución. Para realizar la combinación de ambas ramas utilizan una unidad de fusión y dado que el tiempo de ejecución es importante, usan la adición de funciones en lugar de la concatenación.

Considerando lo realizado en ContextNet [79], para la arquitectura propuesta se decide solo realizar un upsampling en el mapa de baja resolución, para obtener ambos mapas con la misma resolución y finalmente realizar una concatenación de las estimaciones de profundidad. En el trabajo de [65] mencionan que en resoluciones bajas cercanas a la resolución de entrenamiento, las estimaciones tienen una estructura consistente y carecen de detalles de alta frecuencia; por el contrario cuando la misma imagen se alimenta a la red en resoluciones más altas, los detalles de alta frecuencia se capturan mucho mejor mientras que la consistencia estructural de la estimación se degrada gradualmente.

4.1.2 Parches piramidales

En la red Boosting Monocular Depth se generan parches en las zonas de interés, véase la Figura 4.7, por lo que la cantidad de parches puede incrementarse y por lo tanto los tiempos de inferencia son altos y variables dependiendo del contenido de la imagen. Considerando que los mapas de profundidad van a ser utilizados para la navegación evasiva de obstáculos de VANTs, se tiene que la principal zona de interés son los píxeles centrales de la imagen; por lo que se consideró modificar la forma en que se generan los parches.

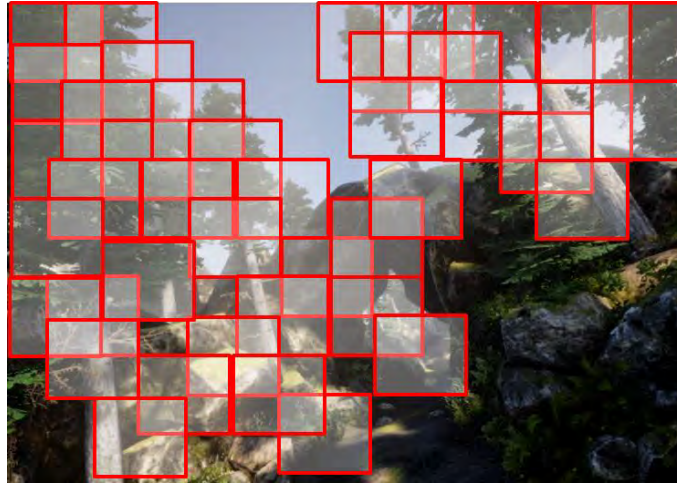


Figura 4.7 Ejemplo de cómo se generan los parches en Boosting Monocular Depth [1]

Revisando el estado del arte se encontraron diversos trabajos [6–9] en donde utilizan el método de *Spatial Pyramid*, véase la Figura 4.8; el cual consiste en utilizar la agrupación piramidal espacial para generar un amplio grupo de características piramidales con los mapas de diferentes tamaños y luego aplican una abstracción de las mismas, adicional a los mapas de características del grupo en paralelo para que todos tengan niveles similares de abstracción semántica.

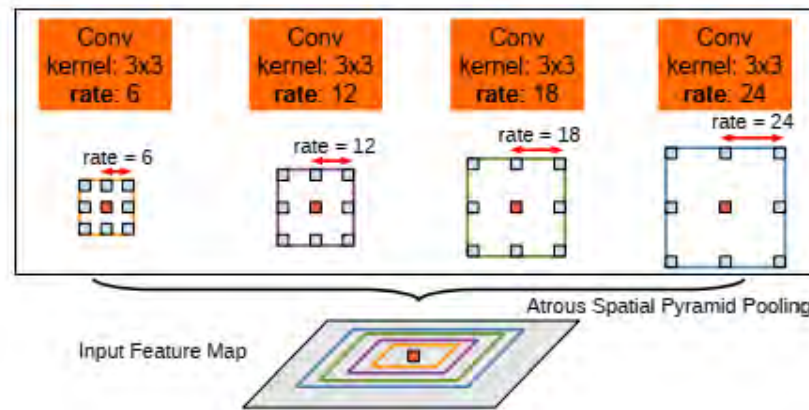


Figura 4.8 Propuesta de Atrous Spatial Pyramid de DeepLab [35]

Tomando la idea principal de tener características piramidales se propone una nueva forma de obtener los parches para así poder detectar los detalles finos en los mapas de profundidad principalmente en el área central de la imagen.

Para modificar la forma en que se generan los parches, se investigó hasta qué punto las señales de entrada pueden afectar las características de salida y mapear las características en cualquier parte de la red a la región de la entrada que las produce. El parámetro clave para asociar una característica de salida a una región de entrada es el campo receptivo de la red convolucional, que se define como el tamaño de la región en la entrada que produce la característica [84].

El tamaño del campo receptivo, RF por sus siglas en inglés Receptive Field, de una red depende principalmente de la arquitectura, así como de la resolución de entrenamiento. Se puede definir como la región alrededor de un píxel que contribuye a la salida en ese píxel [85].

Considerando el campo receptivo (RF) las medidas de los parches son calculadas como se muestra en las siguientes ecuaciones. Teniendo una imagen con tamaño $w \times h$ e indicando que se quiere una pirámide de N niveles se tiene que los bordes E_w y E_h están definidos en las Ecuaciones 8 y 9:

$$E_w = \frac{w - RF}{N} \quad (8)$$

$$E_h = \frac{h - \left(RF \left(\frac{h}{w} \right) \right)}{N} \quad (9)$$

por lo que cada parche $P_1, \dots, P_N$ quedaría definido como se muestra en la Ecuación 10:

$$\begin{aligned} P_N &= w_N \times h_N \quad \therefore \quad w_N = RF \quad y \quad h_N = RF \left(\frac{h}{w} \right) \\ &\vdots \\ P_{N-1} &= w_{N-1} \times h_{N-1} \quad \therefore \quad w_{N-1} = w_N + E_w \quad y \quad h_{N-1} = h_N + E_h \end{aligned} \quad (10)$$

En la Figura 4.9 se muestra un ejemplo con $N = 2$ niveles y un $RF=320$. El valor de RF se obtuvo por experimentación, valores menores a 320 generan un mal desempeño de la red. El parche del nivel 2 que se genera es de 320×240 píxeles, por lo que se tendría $E_w = 160$ píxeles y un $E_h = 120$ píxeles de decremento para cada parche.

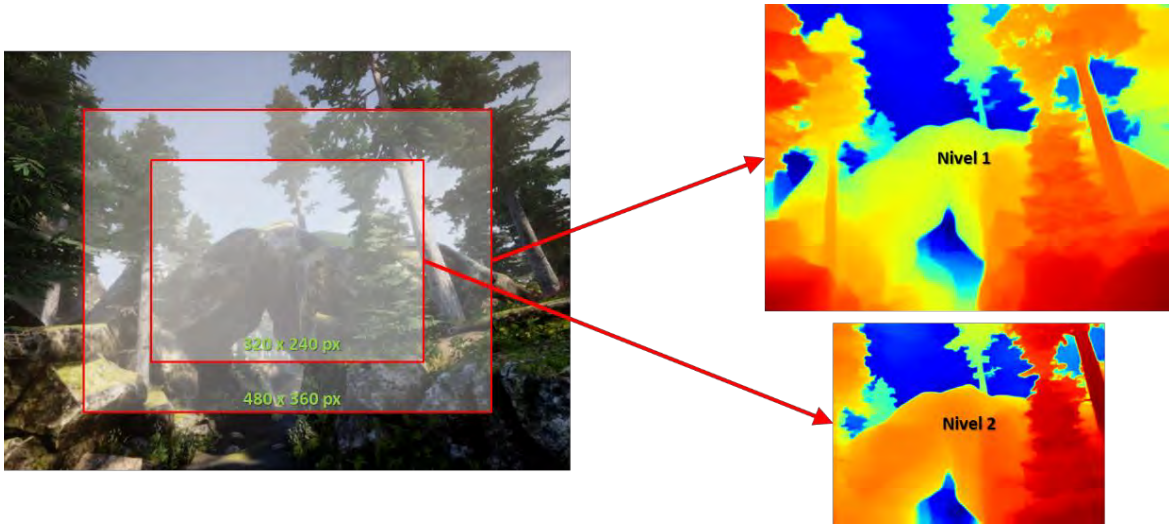


Figura 4.9 Ejemplo de parches piramidales con dos niveles

4.1.3 Procesamiento en paralelo

En el trabajo de [81] se realiza la transformación de características adicionales en paralelo, lo que genera mapas de características con niveles similares de abstracción semántica en todas las escalas. Además, en la experimentación que realizaron utilizando el conjunto de datos COCO, demostró que se ejecuta más del doble de rápido en comparación con los modelos RetinaNet y RefineDet.

Por lo anterior se propuso ejecutar en paralelo la generación de los mapas de profundidad de todos los parches de la pirámide; para poder realizarlo se utilizó la librería de multiprocessing de Python [86], con la librería se creó una función para crear y configurar el grupo de procesos que se ejecutan en paralelo.

Sin embargo, realizar *multiprocessing* utilizando una GPU no es una opción óptima; debido a que se tiene que tomar en cuenta el tiempo de transmisión de las imágenes del CPU al GPU y la entrega de los resultados del GPU al CPU. En la investigación de [87] mencionan que el tiempo de transmisión en ambas direcciones consta de dos componentes. El primer componente es la latencia, que incluye la sobrecarga de preparación. Esta sobrecarga puede ocurrir debido a la decodificación de instrucciones, la latencia de la memoria, la espera de acceso al bus y otras causas. El segundo componente es el tiempo de propagación que depende del ancho de banda (el número de bits propagados por segundo). Esta propiedad tiene un gran impacto en el rendimiento de un procesador gráfico, ya que todos los datos que se utilizarán en el cálculo deben copiarse en el procesador gráfico.

Considerando lo anterior se buscó otra alternativa para realizar el procesamiento de todos los parches en paralelo. Para realizar la inferencia de la red LeReS la imagen se transforma en un tensor utilizando Pytorch [88], a pesar de que las GPU en particular están altamente paralelizadas, una sola entrada en un modelo pequeño dejará inactivas a la mayoría de las unidades informáticas.

Revisando las funcionalidades de la librería Pytorch se encontró que se pueden generar *batches* (lotes de imágenes); al proporcionar lotes de entradas, el cálculo se puede distribuir entre las unidades que de otro modo estarían inactivas, lo que significa que los resultados por lotes regresan tan rápido como lo haría un solo resultado.

Aspirando a conseguir disminuir los tiempos de inferencia se propuso generar *batches* con los parches piramidales obtenidos y utilizar el *batch* como entrada a la red LeReS. Para poder utilizar *batches* todas las imágenes que lo componen tienen que ser del mismo tamaño, por lo anterior se están generando dos *batches*, uno con los parches en alta resolución (640 píxeles) y otro con los parches de baja resolución (320 píxeles).

4.2 Diseño del sistema de evasión de obstáculos

El modelo de VANT disponible es un DJI Tello Quadcopter, que es una plataforma popular para evaluar el aprendizaje profundo en VANTs debido a la conveniente API. Además de que permite la transmisión de video en tiempo real y el control remoto mediante el protocolo UDP; en la Tabla 4.2 se muestran las características del mismo.

Tabla 4.2 Características del VANT DJI Tello Quadcopter

Peso	Aprox. 80 g (hélices y batería incluidas)
Dimensiones	98 × 92.5 × 41 mm (sin protectores)
Funciones integradas	Wi-Fi 2.4 GHz 802.11n, Sensor telemétrico, barómetro, LED y sistema de visión
Foto	5 MP (2,592x1,936) JPG
Vídeo	HD 720p 30 fps MP4
Tiempo de vuelo	13 minutos (distancia máxima de 100 metros)

Para calcular el espacio libre disponible en los mapas de profundidad se tomaron en cuenta las medidas del dron utilizando los protectores, las cuales se muestran en la Figura 4.10.

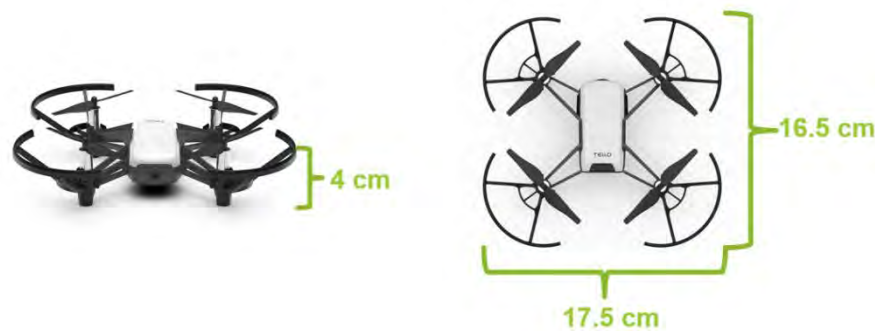


Figura 4.10 Medidas del VANT DJI Tello utilizando protectores

Revisando el estado del arte se encontró el trabajo de [89] en donde proponen un algoritmo que permite detectar obstáculos utilizando mapas de profundidad a partir de imágenes RGB proporcionadas por el VANT. Dicho algoritmo consta de varios pasos, que incluyen la estimación de distancia usando una red de tipo codificador-decodificador inspirado en U-net con una columna vertebral DenseNet169 [27], agrupamiento de mapas de profundidad usando el algoritmo DBSCAN [90] y luego usan una serie de umbrales con parámetros adaptativos para extraer el objeto que está en trayectoria de colisión con el VANT, si llegase a estar un objeto presente. Para la comunicación con el VANT utilizan una librería publica llamada DJITelloPy [91], con esta se establece la conexión y flujo de datos de video desde el dron.

Debido a que en dicho trabajo utilizan el mismo modelo de VANT, se tomó como base el código de su algoritmo para poder realizar la comunicación con el dron e implementar la red de estimación propuesta Boosting Pyramidal Depth; y así poder realizar el análisis del área libre de navegación y generar los comandos de navegación.

Por lo tanto, la arquitectura del sistema de navegación se muestra en la Figura 4.11. La cual consiste en un VANT que se comunica mediante WiFi con una computadora remota que ejecuta cuatro subprocesos: procesamiento RGB, procesamiento de profundidad, detección de obstáculos y generación de comando de navegación.

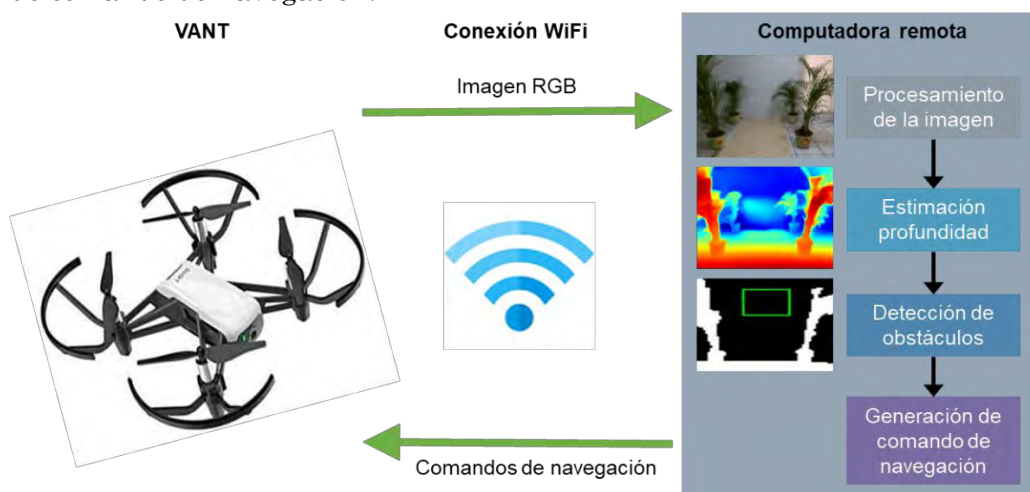


Figura 4.11 Arquitectura del sistema de navegación evasiva de obstáculos

4.2.1 Detección de obstáculos

Teniendo la estimación de profundidad a partir de la imagen RGB proporcionada por el VANT, se prosigue a la detección de obstáculos. El mapa de profundidad resultante de la red es de 8 bits y tiene un rango de 0 a 255; el mapa, ver Figura 4.12, se muestra coloreado utilizando la función `applyColorMAP` con el mapa de color JET, para a mejor apreciación.

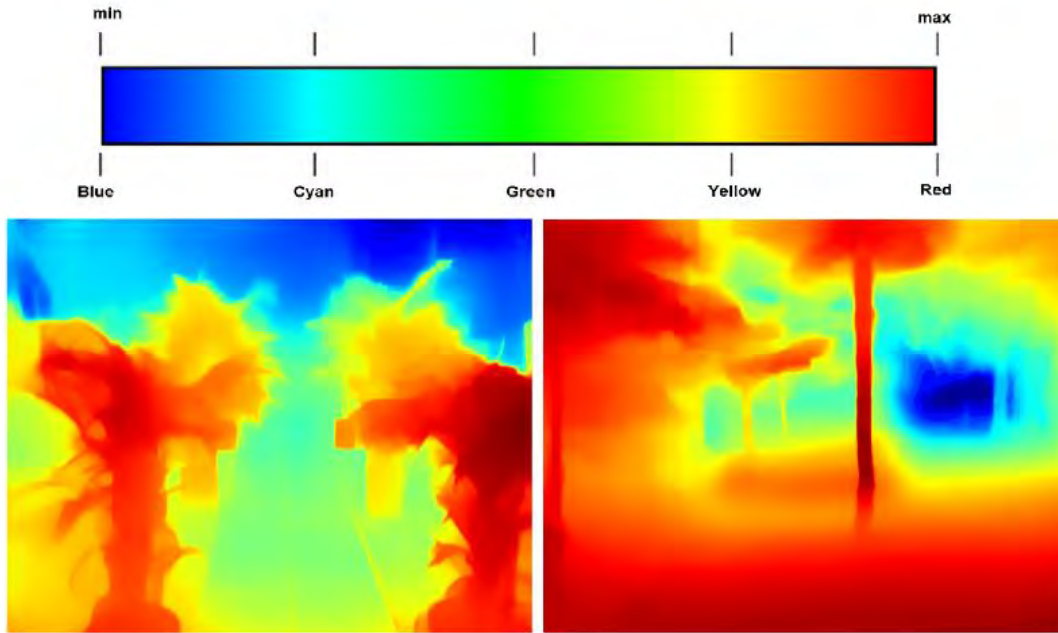


Figura 4.12 Rangos de colores del mapa de color JET

Observando los mapas de profundidad obtenidos coloreados, se tiene que los pixeles que se encuentran dentro del rango Amarillo (Yellow) y Rojo (Red) son de nuestro interés. Por lo que se realizó una umbralización del mapa de profundidad, para así solo mostrar como obstáculos los pixeles dentro del rango 191 a 255.

4.2.2 Generación de comandos de navegación

Para poder generar los comandos de navegación primero se tienen que analizar los posibles movimientos que puede realizar el VANT. En la Figura 4.13 se muestran los 6 posibles ejes de movimiento para el VANT utilizado y gracias a estos se pueden hacer giros completos y moverse en cualquier dirección.



Figura 4.13 Movimientos que permite ejecutar el VANT

Para generar los comandos de navegación se tienen que analizar diferentes áreas de la imagen obtenida, las cuales se citan a continuación en jerarquía y se muestran en la Figura 4.14:

- Central: rectángulo en color verde
- Lateral derecha: rectángulo en color azul
- Lateral izquierda: rectángulo en color morado
- Central arriba: rectángulo en color naranja
- Central abajo: rectángulo en color amarillo

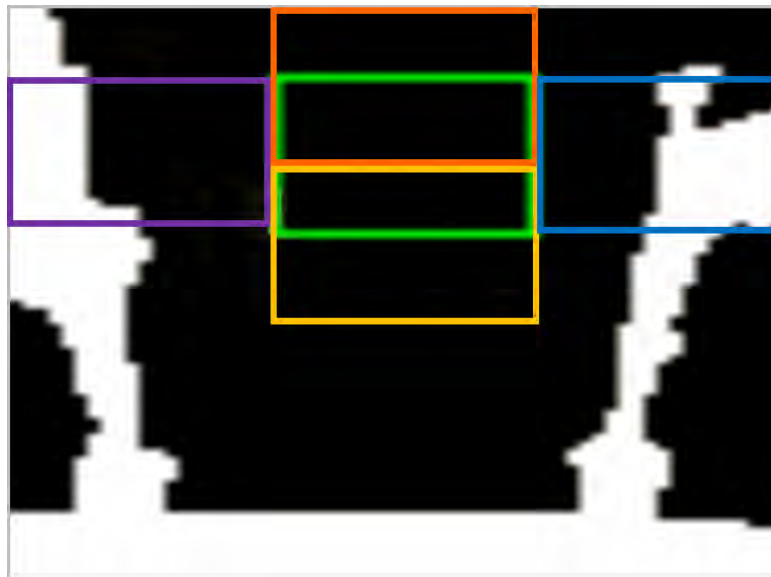


Figura 4.14 Áreas de posible navegación

Debido a que la aplicación del VANT es para el monitoreo de cultivos, por el momento no se tiene necesidad de realizar los giros de guiñada, cabeceo y alabeo. El dron solo requiere mantenerse en el sendero del cultivo por lo que los posibles movimientos a realizar son: adelante, derecha, izquierda, arriba y abajo.

Continuando con la utilización de la librería DJITellopy, se tienen los siguientes códigos para el envío de comando de navegación al VANT:

- a. Central → `move_foward(x)` - Adelante
- b. Lateral derecha → `move_right(x)` - Derecha
- c. Lateral izquierda → `move_left(x)` - Izquierda
- d. Central arriba → `move_up(x)` - Arriba
- e. Central abajo → `move_down(x)` - Abajo

CAPÍTULO 5

Pruebas y Resultados

El presente capítulo describe las pruebas realizadas para evaluar el método descrito en el capítulo 4, teniendo como objetivo de realizar las pruebas es evaluar el rendimiento y la efectividad del método propuesto.

5.1 Herramientas utilizadas (software y equipo)

Los experimentos se llevaron a cabo mediante la configuración de las librerías que se listan en la Tabla 5.1, necesarias en un equipo con Intel Core i7-11800H de 2.30GHz, 32 GB de RAM y una tarjeta gráfica Nvidia GeForce RTX3070 de 8 GB, la cual utiliza como sistema operativo Windows 11.

Tabla 5.1 Librerías para la implementación de las redes

Librerías de python
pytorch >=1.6.0
torchvision
matplotlib
h5py >=3.6.0
scipy >=1.7.3
scikit-image
opencv-python >=4.5.5
mmcv >=1.4.3
timm >=0.5.4
albumentations >=1.1.0
tensorboardX >=2.4.1

La experimentación se dividió en dos fases: primero se realizaron diversas pruebas utilizando la red propuesta para así realizar mejoras en la misma y en la segunda fase se realizaron pruebas de navegación utilizando el diseño del sistema de navegación evasiva de obstáculos con la red propuesta. Los detalles de la experimentación de cada una de las fases se mencionan a continuación.

5.2 Conjuntos de datos utilizados

Existen varios conjuntos de datos disponibles públicamente para la estimación de profundidad; estos proporcionan imágenes y mapas de profundidad correspondientes desde diferentes puntos de vista, interiores y exteriores. La siguiente sección presenta algunos conjuntos de datos populares utilizados para analizar las escenas.

- KITTI. El conjunto de datos KITTI [49] es uno de los conjuntos de datos tradicionales utilizados para la estimación de profundidad, detección de objetos y conducción autónoma. Contiene imágenes *ground truth* de alta calidad capturadas desde un vehículo utilizando el escáner láser Velodyne, dos cámaras en color de alta resolución, dos cámaras en escala de grises y un sistema de posicionamiento global (GPS). Desarrollados por el Instituto de Tecnología de Karlsruhe en Alemania y el Instituto de Tecnología de Toyota en Estados Unidos, los conjuntos de datos se capturaron conduciendo por la ciudad mediana de Karlsruhe, en áreas rurales y en carreteras.
- NYU DEPTH V2. El conjunto de datos de profundidad v2 de NYU es un conjunto de datos de interiores para estimación de profundidad, se propuso en [17] y es uno de los conjuntos de datos más utilizados. Incluye 407.000 fotogramas de 464 escenas tomadas en tres ciudades diferentes. Las escenas fueron grabadas por las cámaras RGB y de profundidad de Microsoft Kinect. El tamaño de imagen original de NYU Depth V2 es 640×480 y los rangos de profundidad del conjunto de datos son de 0.5 m a 10 m.
- MAKE 3D. El conjunto de datos de Make3D contiene escenas exteriores con imágenes RGB y su correspondiente *ground truth*. Fue introducido en el trabajo de [21] e incluye paisajes urbanos y naturales durante el día, con mapas de profundidad recopilados por un escáner láser. Cubre una amplia gama de entornos interiores y exteriores y escenas sintéticas, con un total de 534 pares de imágenes RGB-D, 400 imágenes se utilizan para entrenamiento y 134 para pruebas.
- CITYSCAPES. El conjunto de datos Cityscapes se propuso en [22] para la segmentación semántica y la comprensión de la escena urbana. Contiene un conjunto diverso de secuencias de video estéreo grabadas en escenas callejeras de 50 ciudades diferentes, con anotaciones a nivel de píxeles de alta calidad de fotogramas de 5K y un conjunto más extenso de imágenes de 20K con anotaciones débiles que se pueden usar para tareas de estimación de profundidad.
- MIDDLEBURY. El conjunto de datos de Middlebury se centró principalmente en imágenes estéreo [23] que contienen diversos objetos y plantas en interiores. También proporciona mapas de profundidad para la investigación de estimación de profundidad en sus últimos conjuntos de datos [24-27]. Incluye varias escenas y mapas de disparidad *ground truth*, lo que lo convierte en un recurso valioso para evaluar la precisión de los modelos de estimación de profundidad.
- TARTANAIR. El conjunto de datos de TartanAir es relativamente nuevo y se introdujo en [28]. Consiste en datos recopilados a partir de simulaciones fotorrealistas de escenas interiores y exteriores con objetos 3D detallados utilizando Unreal Engine y AirSim. Un objetivo distintivo de este conjunto de datos es que se centra en entornos desafiantes con diferentes condiciones de luz, clima adverso y objetos dinámicos. Las simulaciones al aire

libre incluyen escenas rurales y urbanas con varios tipos de edificios, árboles, terrenos y paisajes.

Para la evaluación también se utilizó el conjunto de datos público TartanAir [11], este se compone de diferentes recorridos de ejemplo, por lo que se decidió utilizar las trayectorias que se componen de los elementos de nuestro interés como son: diversa vegetación (árboles, arbustos, pasto) y utilería de agricultura (vallas, luminarios, postes, paredes, senderos).

Las trayectorias que se escogieron fueron: Gascola, Season Forest, Season Forest Winter, Amusement y Western Desert; para cada trayectoria se tiene las imágenes de profundidad, RGB y segmentación de la cámara izquierda y derecha; en la Tabla 5.2 se puede observar los detalles de cada conjunto de datos por trayectoria.

Tabla 5.2 Características de las trayectorias de TartanAir [3]

	Gascola	Season Forest	Season Forest Winter	Amusement	Western Desert
Tamaño	1.38 GB	1.24 GB	1.22 GB	2.72 GB	1.14 GB
Píxeles imagen	640 x 480	640 x 480	640 x 480	640 x 480	640 x 480
Tamaño imágenes	901 KB	901 KB	901 KB	901 KB	901 KB
Cantidad de imágenes	382	301	330	644	389

Las características de los cinco ambientes se mencionan a continuación:

Gascola: Ambiente de zonas boscosas, con diversas áreas rocosas, musgo y diferentes especies de pinos; con iluminación de día en la mañana. Algunas imágenes de esta simulación se pueden observar en la Figura 5.1.



Figura 5.1 Imágenes RGB del ambiente Gascola de TartanAir [11]

Season Forest: Ambiente de zonas boscosas en la estación de otoño, contiene diferentes especies de árboles con las tonalidades de otoño en los troncos y hojas, con efectos de hojas cayendo. La iluminación es al mediodía, por lo que se tienen sombras debajo de los árboles; este se puede observar en la Figura 5.2.



Figura 5.2 Imágenes RGB del ambiente Season Forest de TartanAir [11]

Season Forest Winter: Ambiente de zonas boscosas en la estación de invierno, contiene diferentes especies de árboles mostrando solo los troncos y ramas en tono gris y cubiertos de nieve; estos detalles se pueden observar en la Figura 5.3. La iluminación es en el atardecer, teniendo así sombras a un costado de los árboles.



Figura 5.3 Imágenes RGB del ambiente Season Forest Winter de TartanAir [11]

Amusement: Ambiente de un parque de diversiones abandonado, contiene vegetación como musgos, matorrales y hierba baja, además de mostrar distintos tipos de tuberías y materiales de acero que también se pueden llegar a encontrar en cultivos; estos detalles se pueden observar en la Figura 5.4. La iluminación es en el atardecer, por lo que se tienen sombras reflejadas en el suelo.



Figura 5.4 Imágenes RGB del ambiente Amusement de TartanAir [11]

Western Desert: Ambiente de zonas desérticas con iluminación en el anochecer, contiene escasa vegetación y construcciones de casas en madera; las cuales se podrían considerar que simulan graneros; estos detalles se pueden observar en la Figura 5.5.



Figura 5.5 Imágenes RGB del ambiente Western Desert de TartanAir [11]

Dataset propio

Luego de revisar los conjuntos de datos utilizados en el estado del arte, se observó que la mayoría contienen imágenes de escenas interiores, y algunos de los conjuntos de datos del exterior se obtuvieron para utilizarlos para la navegación autónoma en escenarios urbanos.

Sólo el conjunto de datos de TartanAir contiene imágenes de árboles, paisajes y terrenos. Sin embargo, aún faltan fotografías con vegetación variada como pasto, arbustos, árboles de diferentes tipos, senderos para caminar y cercas para que se considere natural. Por lo tanto, se trabajó en la creación de un nuevo conjunto de datos especializado principalmente en entornos naturales, para ello se empleó la misma iniciativa de [28] para utilizar simulaciones usando Unreal Engine [29] y AirSim [30].

Para el diseño de este conjunto de datos, se descargó y configuró AirSim para utilizar la simulación con nombre City Park, disponible de forma gratuita en Unreal Engine, en la Figura 5.6. se muestran algunas imágenes del ambiente mencionado. Esta herramienta de simulación proporciona escenas fotorrealistas, tiene efectos climáticos y es fácil de aprender.



Figura 5.6 Imágenes del ambiente de simulación City Park [8]

El entorno de simulación de City Park es un escenario de parque que contiene vegetación diversa, postes de madera, metal e iluminación, vallas y senderos para caminar. Se realizaron tres vuelos en distintos momentos del día para tener imágenes con iluminación variada, recolectando 9,288 imágenes RGB, con su correspondiente imagen de profundidad. La Tabla 5.3 muestra los detalles de cada trayectoria y nuestro conjunto de datos está disponible en: https://drive.google.com/file/d/1iYCtomNaSHsr2i3rMDkOdJOJ5FufD76W/view?usp=drive_link

Tabla 5.3 Características de las trayectorias de CityPark

Horario	8:30 AM	01:30 PM	08:30 PM
Píxeles imagen	640×480	640×480	640×480
Tamaño imágenes	866 KB – 402 KB	872 KB – 409 KB	899 KB – 406 KB
Duración del vuelo	10 min	15 min	10 min
Cantidad de imágenes	1,479	5,773	2,036

5.3 Experimentación con la red de estimación de profundidad

Se realizaron dos casos de prueba, los cuales se mencionan a continuación:

- El objetivo de esta prueba fue evaluar cuántos niveles de la pirámide son necesarios para obtener los mapas de profundidad óptimos con el conjunto de datos TartanAir.
- El objetivo fue evaluar la modificación de la red propuesta para procesar los parches de la pirámide en paralelo (*multiprocessing* o *batches*) para generar los mapas de profundidad de alta y baja calidad.

Caso A: Evaluar la cantidad de niveles de la pirámide óptimos

Se consideró la trayectoria Gascola de TartanAir, ya que cuenta con la mejor variedad de vegetación y de zonas rocosas. Se tomaron las 764 imágenes RGB y se realizó la estimación de profundidad; luego estas imágenes fueron comparadas con las imágenes de profundidad de referencia, las cuales para la evaluación serán el *Ground Truth*.

Los resultados obtenidos usando la trayectoria de Gascola se muestran en las Tablas 5.4 y 5.5. Tomando como base de comparación los resultados obtenidos de la red Boosting Monocular Depth (Boosting (LeReS)), se pudo observar que utilizando 6 niveles de parches se logran resultados muy cercanos a los de Boosting Monocular Depth en las métricas clásicas de error, así como en la diferencia relativa absoluta (AbsRel), el error cuadrático medio (RMSE), el RMSE (log) y el error relativo cuadrático (SqRel). Sin embargo, utilizando solo tres niveles de parches se logran obtener mejores resultados en las métricas de precisión del umbral. Las métricas de error son mejores cuando tienden a 0 y las métricas de precisión son mejores cuando están más cerca de 1.

Tabla 5.4 Resultado de las métricas de Boosting piramidal con distintos niveles utilizando la trayectoria Gascola

	Métricas	Boosting (LeReS)	Boosting piramidal 3 niveles	Boosting piramidal 6 niveles
minimizar	RMSE	0.023	0.062	0.031
	ORD	0.295	0.462	0.349
	D3R	0.489	0.526	0.489
	RMSE log	0.205	0.388	0.277
	Abs. Rel.	0.487	0.668	0.566
	Sq. Rel.	0.073	0.149	0.096
maximizar	$\delta > 1.25$	0.658	0.841	0.788
	$\delta > 1.25^2$	0.392	0.676	0.573
	$\delta > 1.25^3$	0.224	0.518	0.388

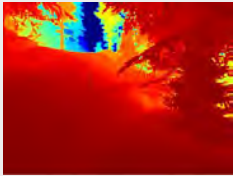
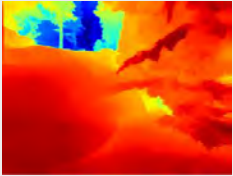
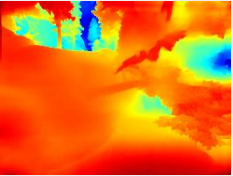
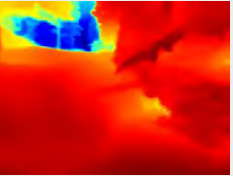
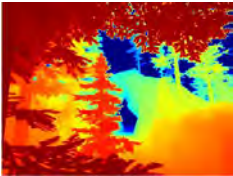
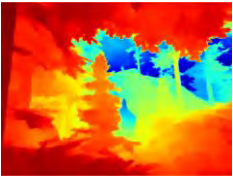
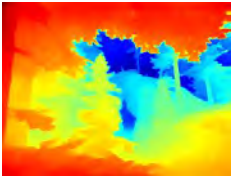
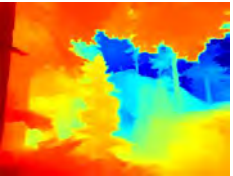
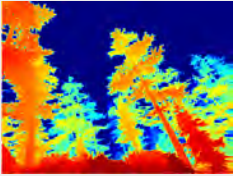
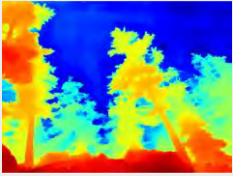
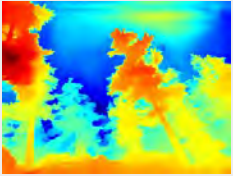
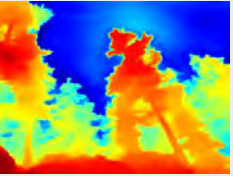
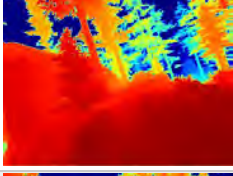
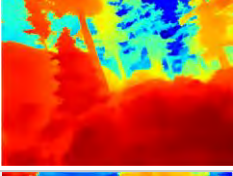
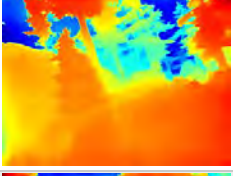
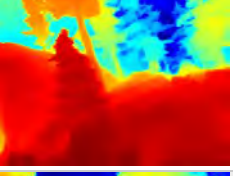
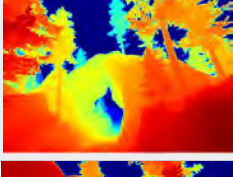
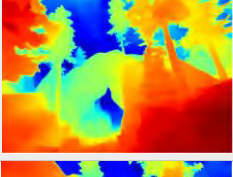
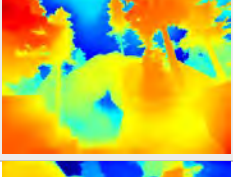
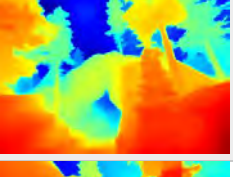
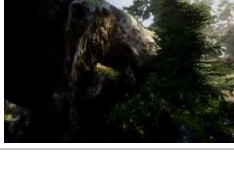
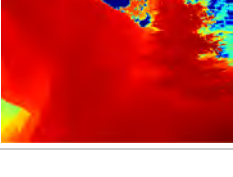
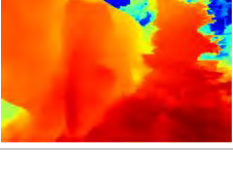
En la Tabla 5.5 se presentan los tiempos de inferencia obtenidos con las 3 propuestas. Los tiempos de predicción aumentan conforme se van agregando niveles en la pirámide; pero, aun así, lejos del tiempo de la red Boosting Monocular Depth; en el caso de 3 niveles teniendo un promedio 774 milisegundos y con 6 niveles un promedio de 1293 milisegundos.

Tabla 5.5 Tiempos de ejecución de Boosting piramidal con distintos niveles utilizando la trayectoria Gascola

Tiempo inferencia (ms)	Boosting (LeReS)	Boosting piramidal 3 niveles	Boosting piramidal 6 niveles
máximo	3,424	2,441	1,804
mínimo	1,626	415	1,161
σ	167	246	83
promedio	1,796	774	1,293

En la Tabla 5.6 se muestran los mapas de profundidad obtenidos con Boosting Piramidal utilizando distinta cantidad de niveles en la pirámide. Se observa que los mapas en las imágenes resultantes, donde se utilizan tres niveles, en algunos casos no se determina bien el espacio de las zonas libres para navegar, teniendo distintos niveles de profundidad para indicar el cielo. En las zonas donde existen obstáculos son encontrados por el algoritmo, pero la profundidad detectada no es la correcta. En los mapas obtenidos utilizando 6 niveles, se observa que los problemas mencionados, se disminuyen en su mayoría, teniendo los objetos con la profundidad bien definida y los espacios libres se muestran con sus colores intensos.

Tabla 5.6 Mapas de profundidad obtenidos con Boosting piramidal con distintos niveles

RGB	Ground Truth	Boosting (LeReS)	Boosting piramidal 3 niveles	Boosting piramidal 6 niveles
				
				
				
				
				
				

Caso B: Paralelizar el proceso de generación de mapas de los parches

Teniendo definidos la cantidad de niveles (seis) de la pirámide para obtener mapas de profundidad con resultados óptimos, se prosiguió a paralelizar el proceso de generación de los parches.

Se ejecutó la arquitectura propuesta utilizando el *multiprocessing* de los parches en el GPU y se obtuvieron los tiempos de inferencia, los cuales se muestran en la Figura 5.7. Donde se observa que no se obtuvo el resultado esperado en la reducción en los tiempos de inferencia, al contrario, se incrementa. Los resultados se deben a que el GPU trabaja de manera asíncrona y no cuenta con un modelo de memoria totalmente coherente que permita sincronizar los procesos.

Por lo anterior, se descartó utilizar el *mutiprocessing* y se optó por mandar en *batches* todos los parches y la imagen base. Para poder realizar esto se necesitan todas las imágenes del *batch* en el mismo tamaño; así que se crearon dos *batches*, uno con las imágenes en baja resolución y otro con las imágenes en alta resolución. Teniendo ambos *batches*, son procesados secuencialmente para obtener los mapas de profundidad. Los tiempos de inferencia obtenidos con esta propuesta también se pueden observar en la Tabla 5.7, en donde sí disminuyen los tiempos considerablemente.

Tabla 5.7 Tiempos de inferencia obtenidos utilizando paralelización

Tiempo inferencia (ms)	Boosting (LeReS)	Boosting piramidal secuencial	Boosting piramidal multiprocessing	Boosting piramidal batches
máximo	3,424	811	19,044	1,085
mínimo	1,626	298	18,669	1,042
σ	167	191	136	14
promedio	1,796	680	18,804	1,059

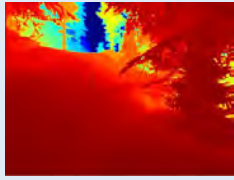
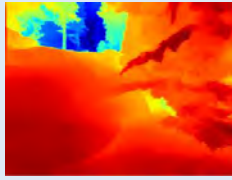
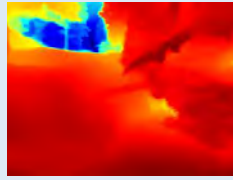
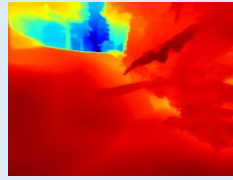
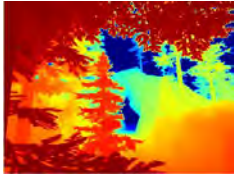
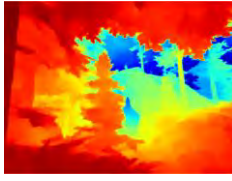
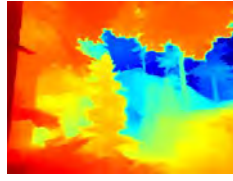
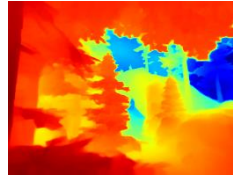
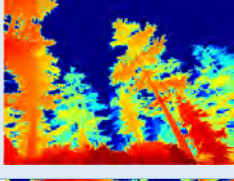
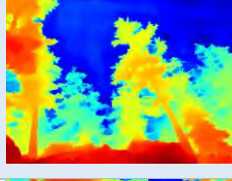
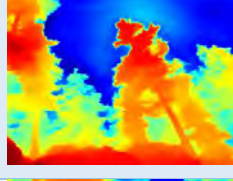
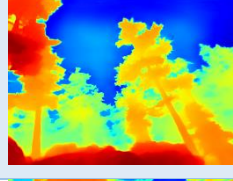
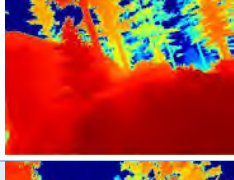
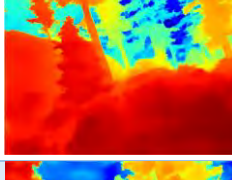
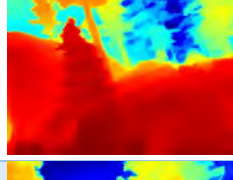
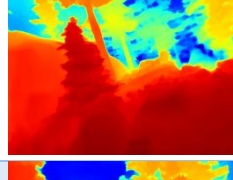
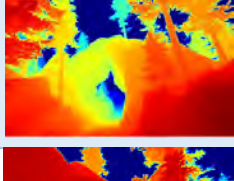
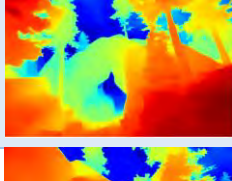
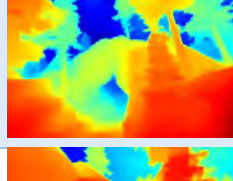
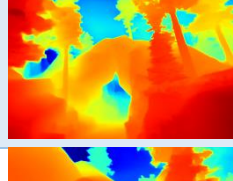
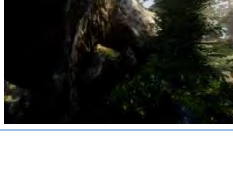
Al utilizar el procesamiento en *batches* de los parches, se realiza un cambio de tamaño de las imágenes y los mapas de profundidad resultantes pueden cambiar debido al cambio de resolución. Por ello, se realizó una comparativa de los resultados de las métricas para verificar que se siguen obteniendo mapas de profundidad óptimos, dichos resultados se pueden observar en la Tabla 5.8.

Tabla 5.8 Comparativa de los resultados de las métricas de Boosting piramidal con procesamiento secuencial y en batches

	Métricas	Boosting (LeReS)	Boosting piramidal secuencial	Boosting piramidal batches
minimizar	RMSE	0.023	0.031	0.038
	ORD	0.295	0.349	0.375
	D3R	0.489	0.489	0.505
	RMSE log	0.205	0.277	0.249
	Abs. Rel.	0.487	0.566	0.577
	Sq. Rel.	0.073	0.096	0.095
maximiza	$\delta > 1.25$	0.658	0.788	0.725
	$\delta > 1.25^2$	0.392	0.573	0.507
	$\delta > 1.25^3$	0.224	0.388	0.336

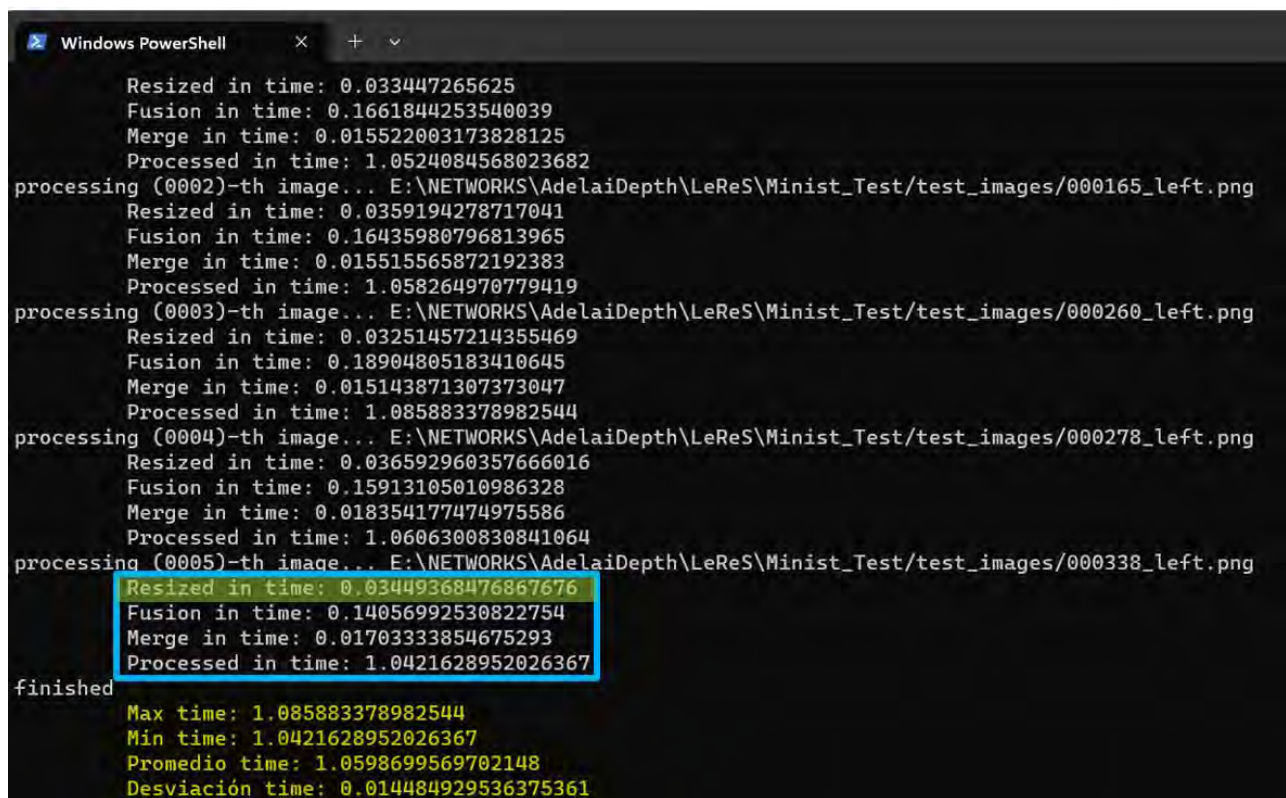
En la Tabla 5.9 se muestran los mapas de profundidad obtenidos utilizando Boosting piramidal secuencialmente y en batches. Prestando atención a los mapas de profundidad obtenidos utilizando batches, se puede observar que estos son más parecidos a los obtenidos con Boosting Monocular Depth y se logran respetar los rangos de profundidad en los objetos, teniendo los más cercanos en tonos rojo intenso y los espacios libres en azul intenso. Demostrando que el procesar todos los parches con el mismo tamaño para la baja y alta resolución ayuda a tener consistencia en la profundidad de los objetos.

Tabla 5.9 Mapas de profundidad obtenidos con Boosting piramidal secuencial y en batches

RGB	Ground Truth	Boosting (LeReS)	Boosting piramidal secuencial	Boosting piramidal batch
				
				
				
				
				
				

A pesar de que los tiempos de inferencia se disminuyeron, siguen teniendo valores por arriba de un segundo. Con la intención de seguir disminuyendo los tiempos se obtuvieron los tiempos en los siguientes subprocesos: cambio de tamaño de imágenes (*resize*), fusión de imágenes y combinación de parches (*merged*).

Analizando los tiempos de cada uno de los subprocesos, los cuales se muestran en la Figura 5.7, se observó que el realizar el procesamiento de los parches en batches, se llevan a cabo múltiples operaciones de *resize* y debido a ello se tiene un incremento en los tiempos de inferencia. Por lo anterior, se buscó una forma de reducir los tiempos del cambio de tamaño de las imágenes, entonces se cambió la operación de *resize* que se realizaba con *arrays* utilizando openCV, a realizar las operaciones utilizando tensores y haciendo el *resize* con pytorch; con esto se pudo observar una reducción en los tiempos de inferencia teniendo un promedio de 695 milisegundos.



```
Windows PowerShell
Resized in time: 0.033447265625
Fusion in time: 0.1661844253540039
Merge in time: 0.015522003173828125
Processed in time: 1.0524084568023682
processing (0002)-th image... E:\NETWORKS\AdelaiDepth\LeReS\Minist_Test/test_images/000165_left.png
Resized in time: 0.0359194278717041
Fusion in time: 0.16435980796813965
Merge in time: 0.015515565872192383
Processed in time: 1.058264970779419
processing (0003)-th image... E:\NETWORKS\AdelaiDepth\LeReS\Minist_Test/test_images/000260_left.png
Resized in time: 0.03251457214355469
Fusion in time: 0.18904805183410645
Merge in time: 0.015143871307373047
Processed in time: 1.085883378982544
processing (0004)-th image... E:\NETWORKS\AdelaiDepth\LeReS\Minist_Test/test_images/000278_left.png
Resized in time: 0.036592960357666016
Fusion in time: 0.15913105010986328
Merge in time: 0.018354177474975586
Processed in time: 1.0606300830841064
processing (0005)-th image... E:\NETWORKS\AdelaiDepth\LeReS\Minist_Test/test_images/000338_left.png
Resized in time: 0.03449368476867676
Fusion in time: 0.14056992530822754
Merge in time: 0.01703333854675293
Processed in time: 1.0421628952026367
finished
Max time: 1.085883378982544
Min time: 1.0421628952026367
Promedio time: 1.0598699569702148
Desviación time: 0.014484929536375361
```

Figura 5.7 Tiempos de los subprocesos de la red propuesta

5.4 Experimentación con el diseño del sistema de evasión de obstáculos

Teniendo bien definida la red de estimación de profundidad se prosiguió a diseñar el sistema de evasión de obstáculos. Anteriormente se había mencionado que se encontró en el estado del arte el trabajo de Tomasz Hachaj [89] en donde utilizan el mismo modelo de VANT y tiene implementada la librería DJITellopy para la comunicación con el dron.

Primero se realizaron pruebas utilizando el código de [89], pero en lugar de utilizar la red DenseDepth se implementó la red propuesta en este trabajo, Boosting Piramidal Depth. Se realizaron algunos vuelos en interiores para obtener el tiempo de procesamiento entre cada comando de navegación, véase la Figura 5.8.

```
2023-10-20 09:34:34.451237: I tensorflow/stream_executor/cuda/cuda_dnn.cc:384] Loaded cuDNN version 8100
2023-10-20 09:34:36.371475: I tensorflow/stream_executor/cuda/cuda_blas.cc:1614] TensorFlow-32 will be u
ogged once.
1/1 [=====] - 5s 5s/step
==> foward
1/1 [=====] - 0s 63ms/step
==> foward
1/1 [=====] - 0s 51ms/step
==> foward
1/1 [=====] - 0s 47ms/step
==> foward
1/1 [=====] - 0s 45ms/step
==> foward
1/1 [=====] - 0s 46ms/step
==> foward
1/1 [=====] - 0s 45ms/step
==> right
1/1 [=====] - 0s 43ms/step
==> foward
1/1 [=====] - 0s 47ms/step
==> foward
1/1 [=====] - 0s 58ms/step
==> right
```

Figura 5.8 Tiempos de generación de los comandos de navegación

Con estos primeros vuelos se detectó que el algoritmo DBSCAN incrementa los tiempos de generación de comandos, ya que su procesamiento es más lento que otras partes del sistema. Entonces, se decidió cambiar el algoritmo y realizar una umbralización en el mapa de profundidad generado para así detectar los obstáculos, en la Figura 5.9 se muestran las imágenes obtenidas por el sistema.

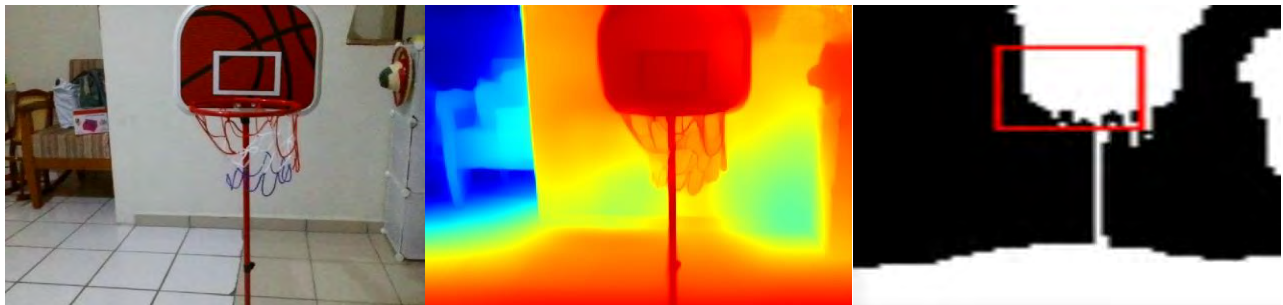


Figura 5.9 Imágenes generadas por el sistema de navegación

También se encontró que el VANT tiene un tiempo de espera para poder recibir el siguiente comando de navegación, así que la comunicación se tiene que realizar de manera asíncrona, esperando a que el dron esté disponible de nuevo.

Al finalizar el sistema de navegación completo, se decidió realizar dos etapas de experimentación: a) en interiores y b) en exteriores. Las imágenes y resultados de dichos vuelos se analizarán las adelante.

5.5 Análisis de resultados

En el presente trabajo se realizaron diversas experimentaciones y en la presente sección se presentan los resultados más relevantes obtenidos en cada una de las etapas. Primero se experimentó con la red propuesta utilizando el conjunto de datos TartanAir para obtener las métricas clásicas y así poder comparar con la literatura. En la segunda etapa se implementó la red propuesta en el sistema de navegación y se realizaron vuelos en interiores y exteriores; a continuación, se mencionan los detalles de los resultados obtenidos.

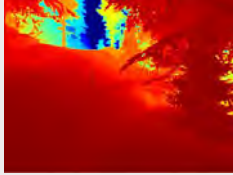
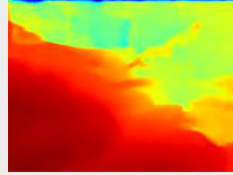
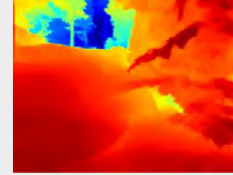
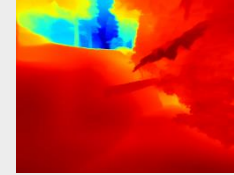
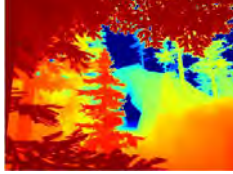
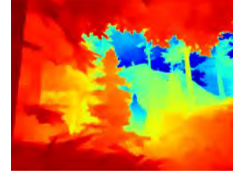
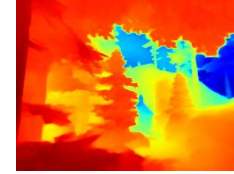
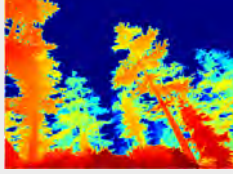
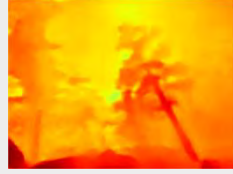
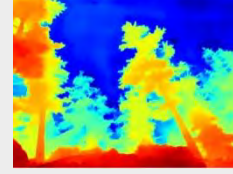
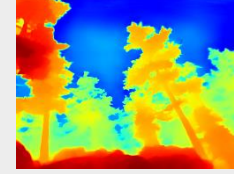
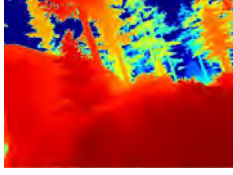
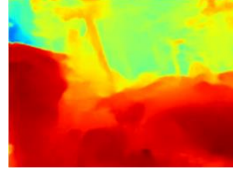
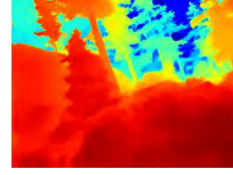
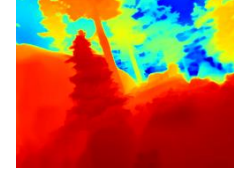
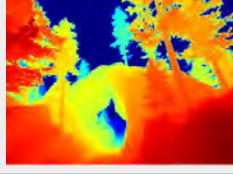
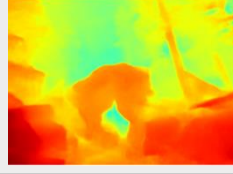
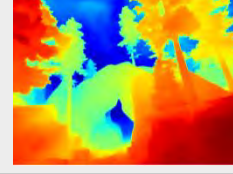
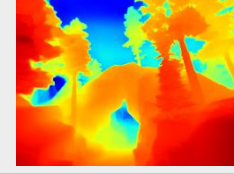
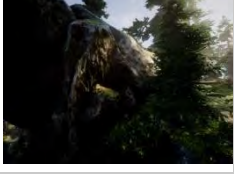
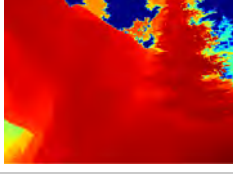
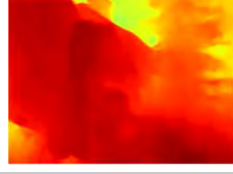
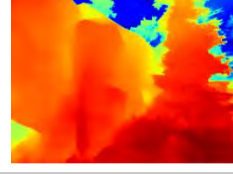
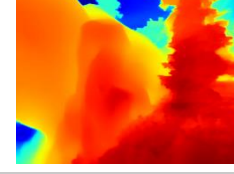
5.5.1 Comparativa de la arquitectura propuesta con la literatura

En la primera fase de experimentación se realizaron pruebas de desempeño de la red propuesta para comprobar que esta es capaz de detectar los detalles finos teniendo tiempos de inferencia menores a los que se reportan en el estado del arte. Como parte de las pruebas de desempeño también se verificó que la red propuesta tiene la capacidad de ser generalizable, para así poder ser usada en diferentes ambientes naturales, a continuación, se menciona el detalle de los dos casos mencionados de experimentación.

Para la experimentación se realizó la comparativa utilizando las trayectorias Gascola, Season Forest y Season Forest Winter con las redes del estado del arte que mostraron mejores resultados, que son GLPDepth y Boosting Monocular Depth.

En la Tabla 5.10 se muestra de manera cualitativa la comparación de los mapas de profundidad obtenidos de las redes del estado del arte que mejores resultados mostraban en su momento y los mapas de profundidad obtenidos utilizando Boosting Pyramidal Depth. En las imágenes se puede observar que los mapas de la red propuesta son parecidos a los obtenidos con Boosting Monocular Depth y que se logran respetar los rangos de profundidad en los objetos, teniendo los objetos más cercanos en tonos rojo intenso y los espacios libres en azul intenso. Mostrando que los parches ayudan a tener consistencia en la profundidad de los objetos.

Tabla 5.10 Comparativa de los mapas de profundidad obtenidos del estado del arte y la red propuesta

RGB	Ground Truth	GLPDepth	Boosting (LeReS)	Boosting piramidal
				
				
				
				
				
				

En la Tabla 5.11 se puede observar que la red propuesta:

- Con la trayectoria de Gascola logra acercarse a los resultados de Boosting Monocular.
- Con la trayectoria de Season Forest obtiene mejores resultados en las métricas de profundidad (**ORD** y **D³R**).
- Con la trayectoria Season Forest Winter logra igualarla.
- Con las métricas de diferencia relativa absoluta (Abs.Rel.), error relativo cuadrático (Sq.Rel.) y las métricas de precisión ($\delta > 1.25$) se obtienen mejores resultados.

Tabla 5.11 Comparativa de las redes de estimación en diferentes trayectorias de TartanAir

Métricas		Gascola			Season Forest			Season Forest Winter		
		GLPDepth	Boosting (LeReS)	Boosting Piramidal	GLPDepth	Boosting (LeReS)	Boosting Piramidal	GLPDepth	Boosting (LeReS)	Boosting Piramidal
minimizar	RMSE	0.086	0.023	0.027	0.133	0.089	0.089	0.201	0.092	0.092
	ORD	0.633	0.295	0.319	0.771	0.611	0.578	0.576	0.624	0.624
	D^3R	0.761	0.489	0.502	0.839	0.742	0.700	0.695	0.540	0.554
	RMSElog	0.313	0.205	0.222	0.504	0.576	0.499	0.443	0.507	0.572
	Abs.Rel.	0.925	0.487	0.528	2.009	1.477	1.742	2.165	1.460	1.260
	Sq.Rel.	0.165	0.073	0.079	0.217	0.193	0.225	0.255	0.167	0.150
maximizar	$\delta > 1.25$	0.771	0.658	0.675	0.893	0.908	0.902	0.868	0.909	0.933
	$\delta > 1.25^2$	0.572	0.392	0.434	0.785	0.819	0.794	0.736	0.807	0.859
	$\delta > 1.25^3$	0.416	0.224	0.271	0.673	0.726	0.673	0.608	0.696	0.773

En cuanto a los tiempos de inferencia, se tiene que la red propuesta muestra los menores tiempos de inferencia, ver Tabla 5.12, logrando tener los menores resultados en la desviación estándar. Con ello se asegura no tener imágenes procesadas en una mayor cantidad de tiempo, ya que se tiene en promedio 681 milisegundos con las trayectorias de Season Forest y Season Forest Winter y un promedio de 695 milisegundos con la trayectoria Gascola.

Tabla 5.12 Comparativa de los tiempos de ejecución de las redes de estimación en diferentes trayectorias de TartanAir

Tiempo inferencia (ms)	Gascola			Season Forest			Season Forest Winter		
	GLPDepth	Boosting (LeReS)	Boosting Piramidal	GLPDepth	Boosting (LeReS)	Boosting Piramidal	GLPDepth	Boosting (LeReS)	Boosting Piramidal
máximo	1,571	2,849	730	1,930	2,949	700	2,122	2,827	702
mínimo	1,317	751	659	1,220	876	659	1,229	892	658
σ	41	756	15	97	782	5	108	810	6
promedio	1,414	1,473	695	1,338	1,959	681	1,352	1,601	681

5.5.2 Comprobar que la arquitectura propuesta es generalizable

Se llevaron a cabo pruebas con todas las trayectorias de TartanAir con el objetivo de comprobar que tan generalizable es la red, considerando distintos ambientes, horarios del día y con diversos cambios climáticos. Para ello se evaluó la red con dos nuevas trayectorias: Amusement y Western Desert (sección 3.5).

En la Tabla 5.13 se muestran los resultados obtenidos en las métricas, en donde se puede observar que se alcanzan buenos valores en las métricas de error y profundidad al obtener valores cercanos a 0 y que en las métricas de precisión al tener valores cercanos a 1. Con los resultados obtenidos en los casos anteriores y este se puede afirmar y comprobar que la red es generalizable.

Tabla 5.13 Resultado de las métricas de Boosting Piramidal en diferentes trayectorias de TartanAir

	Métricas	Gascola	Season Forest	Season Forest Winter	Amusement	Western Desert
minimizar	RMSE	0.027	0.089	0.092	0.050	0.056
	ORD	0.319	0.578	0.624	0.227	0.246
	D^3R	0.502	0.700	0.554	0.531	0.505
	RMSElog	0.222	0.499	0.572	0.422	0.494
	Abs.Rel.	0.528	1.742	1.260	0.908	0.709
	Sq.Rel.	0.079	0.225	0.150	0.130	0.162
maximizar	$\delta > 1.25$	0.675	0.902	0.933	0.901	0.889
	$\delta > 1.25^2$	0.434	0.794	0.859	0.779	0.767
	$\delta > 1.25^3$	0.271	0.673	0.773	0.644	0.648

En cuanto a los tiempos de inferencia, ver la Tabla 5.14, se puede observar que la red es capaz de mantener los valores de la desviación estándar bajos y que el promedio se mantiene en un rango de 670 a 690 milisegundos. De igual forma se comprueba que la red puede mantener bajos tiempos de inferencia al procesar imágenes de distintos ambientes, por lo que sí es posible su uso para determinar el espacio libre donde navegará el VANT.

Tabla 5.14 Tiempos de ejecución de Boosting Piramidal en diferentes trayectorias de TartanAir

Tiempo inferencia (ms)	Gascola	Season Forest	Season Forest Winter	Amusement	Western Desert
máximo	747	700	702	721	695
mínimo	656	659	658	643	654
σ	13	5	6	7	6
promedio	686	681	681	683	678

5.5.3 Pruebas de navegación en interiores

Para este caso de prueba se adaptó un espacio interior para representar un sendero de ambiente natural. El sendero tiene un espacio de 3.5 metros de largo y 1 metro de ancho, teniendo dos tipos de vegetación:

- **Palmera areca:** su nombre científico es *Dypsis lutescens*, es una especie tropical de palmera y cuenta con tallos altos y voluminosas hojas arqueadas verdes. Regularmente mide de 1.5 metros a 3 metros de alto, en nuestro caso las palmeras utilizadas miden 1.2 metros aproximadamente.
- **Aralia elegante:** su nombre científico es *Dizygotheca elegantissima*, es un arbusto elegante que posee tallos con hojas amarronadas, dentadas, finas y estrechas. Puede alcanzar entre 2 metros a 4 metros de alto, en nuestro caso las aralias utilizadas miden 1.5 metros aproximadamente.

En las pruebas previas se detectaron ciertas limitaciones, entre ellas que el VANT tiene un tiempo de respuesta de 1.4 a 3 segundos entre cada envío de comando de navegación; también que teniendo implementado el sistema completo con el análisis de las cinco áreas posibles de navegación y la generación de comandos se necesitan más recursos de cómputo para una rápida respuesta, y por último, el VANT es afectado por las superficies sobre las que vuela afectando la identificación de su posición ocasionando que el VANT tenga movimientos erráticos.

Debido a las limitaciones del equipo de cómputo y el modelo del VANT utilizado, la configuración del sistema de navegación fue el siguiente: la red piramidal utilizó un modelo LeReS con una red troncal ResNet50, estimando solo 1 nivel de la pirámide.

En la Figura 5.10 se muestra como quedo la representación del sendero en interiores y se puede observar como la configuración de la red piramidal tiene la capacidad suficiente para detectar la vegetación que se encuentra en la escena.

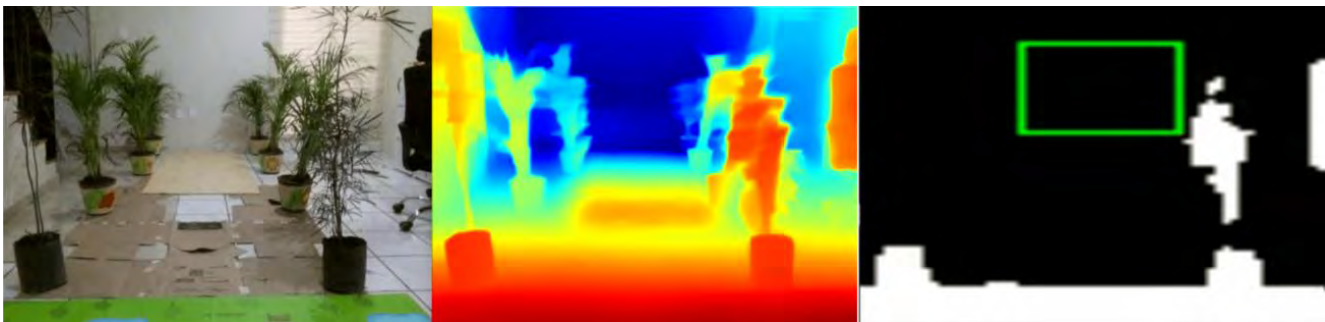


Figura 5.10 Representación de un sendero en interiores: imagen RGB monocular (izquierda), estimación del mapa de profundidad (centro) y la imagen que delimita el espacio de navegación libre (derecha).

Para realizar la navegación evasiva de obstáculos la principal área de navegación es la central, por lo que en la imagen normalizada que se muestra en la Figura 5.11 del lado derecha arriba se está dibujando el área de posible navegación del VANT. Si el área central tiene suficiente espacio libre, el borde del área se muestra en color verde; en caso de encontrarse algún obstáculo que obstruyera el espacio libre, el borde del área se muestra en color rojo. La decisión de un cambio de navegación (derecha, izquierda, arriba o abajo) se toma considerando el análisis de las cinco posibles áreas de navegación.

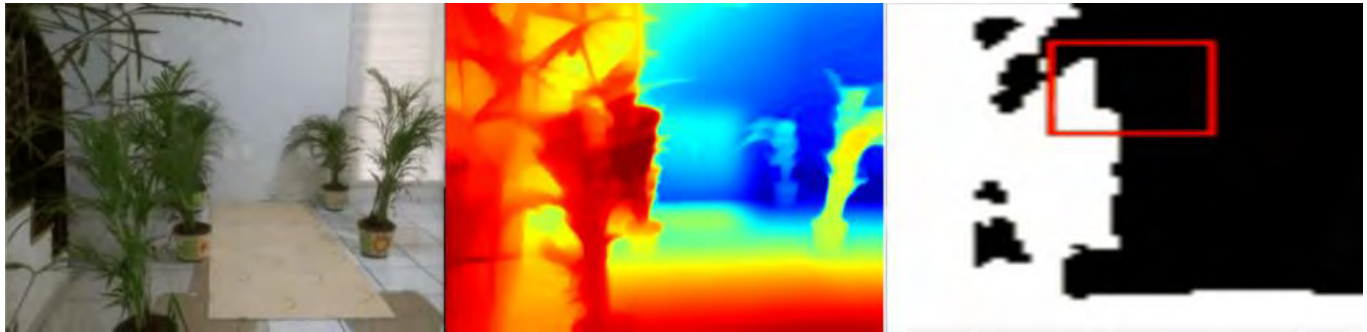


Figura 5.11 Muestra de la detección de obstáculos del sistema de navegación

En total se realizaron 10 vuelos en interiores teniendo 6 vuelos exitosos del VANT en medio del sendero y 4 vuelos fallidos donde el VANT termino colapsando con la vegetación; el tiempo promedio de duración de los vuelos exitosos y fallidos se tiene en la Tabla 5.15.

Tabla 5.15 Tiempo de vuelo promedio de vuelos exitosos y fallidos

	Vuelos exitosos	Vuelos fallidos
Porcentaje	60%	40%
Tiempo promedio de vuelo	64.2 segundos	48.6 segundos

Analizando los videos se observa que la vegetación utilizada afectaba el vuelo del VANT, debido a que el mismo aire que producen las hélices ocasiona que las hojas de las plantas se muevan y provoquen colisiones con el VANT. Además, en dos ocasiones, el sistema detectó de manera correcta el obstáculo; sin embargo, debido al campo de visión de la cámara, el obstáculo se encontraba muy cerca del vehículo, por lo que no se pudo realizar el cambio en la navegación y dio la colisión. El detalle de los vuelos se puede observar en la Tabla 5.16.

Tabla 5.16 Detalle de los vuelos realizados en interiores

Prueba	¿Colisión?	Tiempo de vuelo (segundos)	Observaciones
Vuelo 1	Si	43.5	Hojas fuera del campo de visión, las cuales colisionaron con el VANT.
Vuelo 2	Si	49	El aire de las hélices provocó el movimiento de las hojas y estas chocaron con el VANT.
Vuelo 3	No	53	Logra mantener una trayectoria recta.
Vuelo 4	No	85	Se evitan los dos primeros obstáculos volando alto y completa el vuelo.
Vuelo 5	Si	32	El VANT se encuentra muy cerca del obstáculo cuando se detectó.
Vuelo 6	No	39	En pleno vuelo, detecta obstáculos y los evita realizando cambios de navegación hacia la derecha y hacia arriba.
Vuelo 7	Si	70	El aire de las hélices provocó el movimiento de las hojas, las cuales golpean el VANT.
Vuelo 8	No	64	Evita el obstáculo y luego puede mantener una trayectoria recta.
Vuelo 9	No	84	El sistema detecta tres obstáculos en el camino, que se evitan.
Vuelo 10	No	60	El sistema detecta varios obstáculos a la vez, por lo que realiza dos cambios de navegación continuos para evitarlos.

La Figura 5.12 muestra el número de obstáculos detectados por el sistema en un intervalo de 10 segundos. En promedio, la mayoría de los vuelos alcanzan entre tres y cuatro obstáculos evitados en su punto máximo, y algunos vuelos alcanzan hasta cinco. La capacidad de evitar múltiples obstáculos de manera constante durante los primeros 30 segundos demuestra que el sistema puede ajustar sus movimientos según las condiciones ambientales. La disminución gradual de los obstáculos evitados en la segunda mitad de cada vuelo sugiere que el dron sigue una trayectoria segura, lo que reduce el número de obstáculos encontrados y, por lo tanto, evitados.

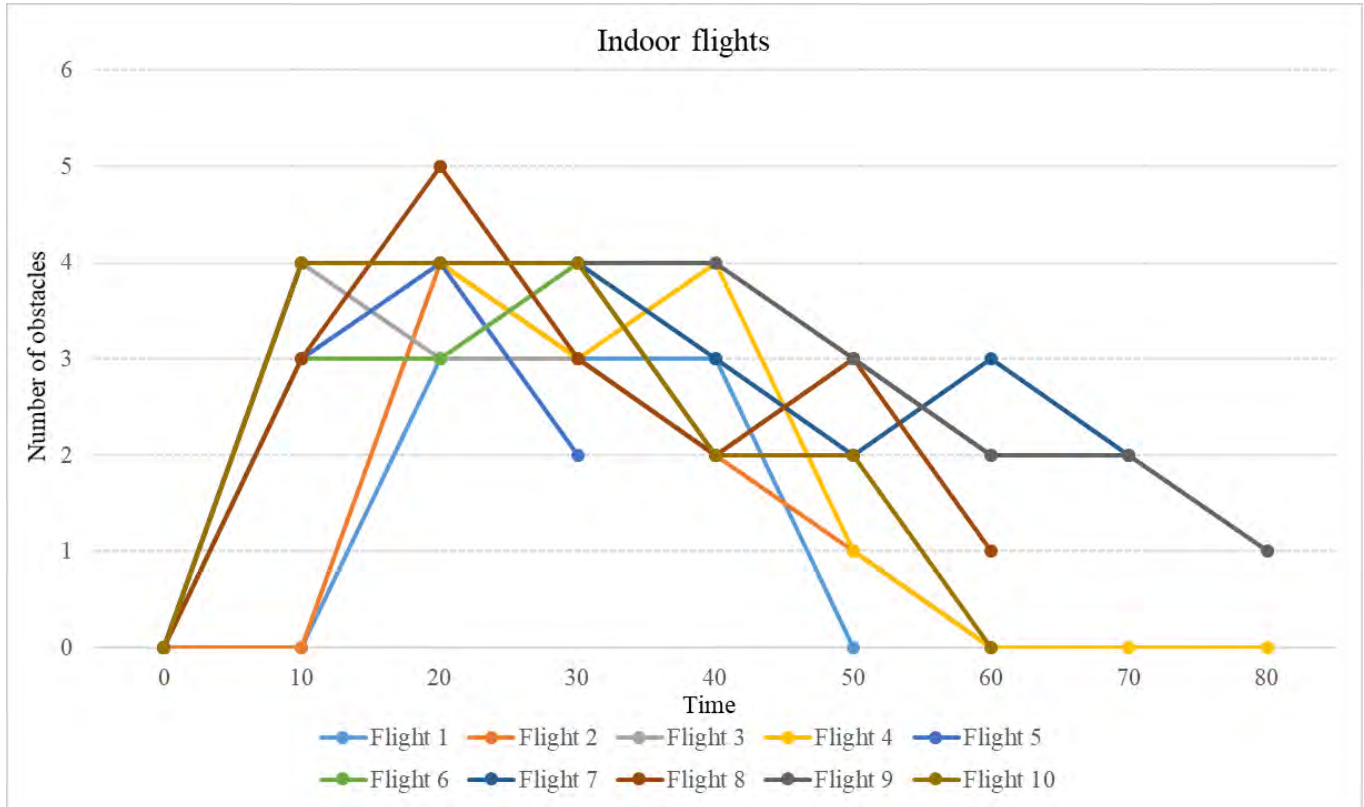


Figura 5.12 Número de obstáculos identificados por el dron cada 10 segundos a lo largo de los vuelos en interiores realizados.

5.5.4 Pruebas de navegación en exteriores

Tras la finalización de las pruebas en escenarios interiores, se realizó la experimentación en escenarios exteriores. Para realizar las pruebas de vuelo se solicitó permiso en una zona con terreno irregular y vegetación variada, en la que predominaban árboles del género pinos.

Se consideraron dos momentos distintos del día para el vuelo con el VANT: los vuelos se realizaron a las 8:00 horas de la mañana y a las 14:00 horas de la tarde. En las Tablas 5.17 y 5.18 se proporciona el detalle de cada vuelo, acompañada de las observaciones pertinentes.

De los vuelos realizados en el horario de la mañana se pudo observar que la Red Boosting Pyramidal Depth es capaz de detectar toda la superficie del suelo como obstáculo y así mantener al VANT con una altura estable para realizar el vuelo.

Los vuelos realizados tienen diferentes posiciones de despegue, esto se debe a que en varias ocasiones el VANT despegaba, pero el sistema inmediatamente detectaba muchos obstáculos sin contar con un espacio libre de navegación, por lo que el sistema indicaba al VANT aterrizar. Por lo anterior se tienen vuelos con reducidos tiempos de vuelo, los cuales se pueden observar en la Tabla 5.17.

Debido a que el escenario que se consiguió para realizar los vuelos era un terreno irregular y que los árboles no se encontraban alineados, no fue posible determinar los metros de vuelo navegados. Esto también se ve reflejado en la cantidad de obstáculos evadidos, ya que había zonas con varios árboles sin espacio suficiente entre ellos y zonas con demasiado espacio libre.

Por lo anterior, se puede determinar que al realizar vuelos en cultivos reales en donde los senderos están bien definidos, el VANT será capaz de realizar el vuelo con éxito, evitando las posibles ramas que llegase a encontrar en su trayectoria.

Tabla 5.17 Detalles de los vuelos realizados en exteriores en la mañana (8:00am)

Prueba	¿Colisión?	Tiempo de vuelo (s)	Observaciones
Vuelo 1	No	39	Iniciando el vuelo detecta varios árboles, realiza movimientos hacia la izquierda y arriba, pero no encuentra una zona libre. Finaliza el vuelo aterrizando.
Vuelo 2	No	15	Encuentra muy cerca un árbol de tronco grueso, finaliza el vuelo aterrizando.
Vuelo 3	No	42	Detecta árboles en todo el ancho de la imagen y termina aterrizando.
Vuelo 4	No	240	Al iniciar el vuelo detecta dos árboles, realiza movimientos para esquivarlos, pero se encuentra con otro árbol y termina aterrizando.
Vuelo 5	No	15	El VANT realiza movimientos a la izquierda y derecha, intentando esquivar varios árboles, no encuentra zona libre por lo que aterriza.
Vuelo 6	No	125	Detecta muchos árboles delgados en todo el ancho de la imagen y finaliza el vuelo aterrizando.
Vuelo 7	No	720	Se detectan varios árboles, arbustos y una valla, el sistema ordena al dron que aterrice

La Figura 5.13 muestra el número de obstáculos que el sistema detecta en intervalos de 10 segundos durante cada vuelo al aire libre por la mañana. Las condiciones de iluminación favorables hacen que el sistema sea más confiable a esta hora del día. En ninguno de los casos se produjo una colisión. Las variaciones en la duración de la prueba se deben a diferentes puntos de partida de los vuelos del dron, que se realizaron en otras situaciones de complejidad. El vuelo 7 muestra que el VANT puede navegar durante períodos prolongados sin incidentes, evitando varios obstáculos continuamente.

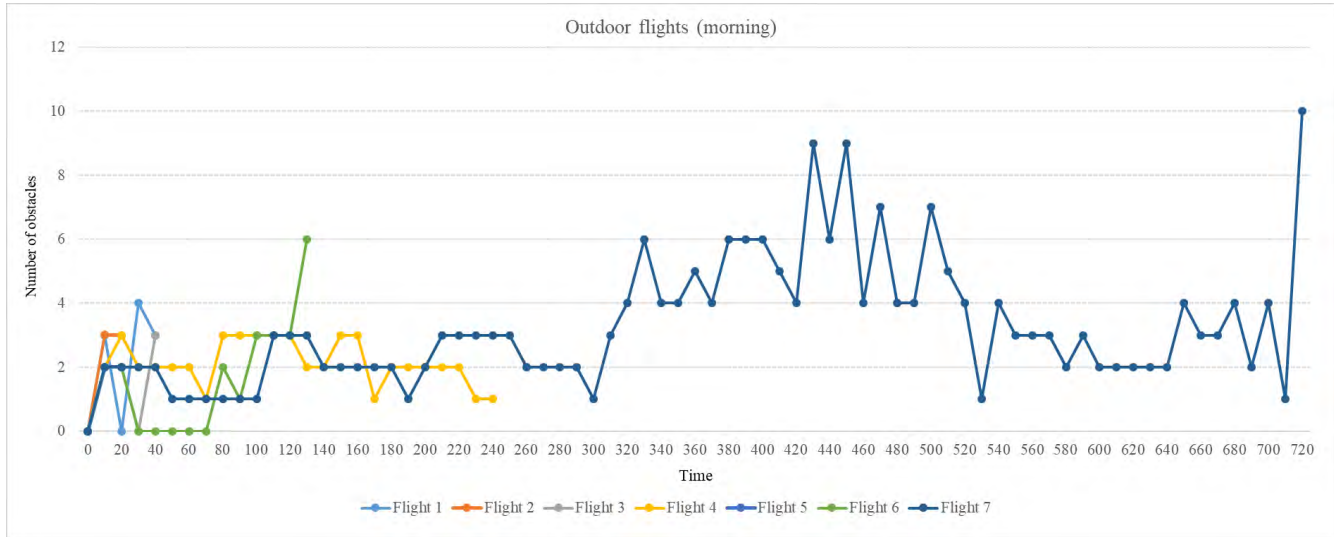


Figura 5.13 Número de obstáculos identificados por el sistema a lo largo de los vuelos al aire libre realizados durante la mañana.

En el horario de la tarde se tiene el inconveniente de que, en la superficie de vuelo, se tienen sombras de los árboles, debido a que la iluminación del sol cambia de posición. A pesar de que en el horario de la mañana el sistema detecta el suelo como obstáculo, en las imágenes obtenidas en el horario de la tarde se ven afectadas por las sombras, ya que las sombras se muestran en el mapa de profundidad en los píxeles más oscuros. Esto ocasiona que el VANT realizara movimientos de elevación iniciando el vuelo, afectando así la altura de vuelo.

Al verse afectada la altura de vuelo, el VANT llega a tener algunas colisiones en los vuelos realizados, véase la Tabla 5.18, esto se debe a que solo se están analizando imágenes de una cámara frontal y por consiguiente el sistema no tiene la capacidad de detectar obstáculos que se encuentran arriba del VANT. Cabe mencionar que en la zona donde se realizaron los vuelos, había ocasiones que se tenían ráfagas de viento y estas afectaban al VANT, ya que por su tamaño pequeño no era capaz de mantenerse estable con la misma altura de vuelo.

Tabla 5.18 Detalles de los vuelos realizados en exteriores en la tarde (2:00pm)

Prueba	¿Colisión?	Tiempo de vuelo (s)	Observaciones
Vuelo 1	Si	105	Detecta sombras en el suelo y el VANT realiza muchos movimientos ascendentes. El sistema detecta el obstáculo cuando el VANT se encuentra muy cerca y termina colisionando.
Vuelo 2	Si	45	Fallo la comunicación en el envío del comando
Vuelo 3	No	110	Detecta sombras en el suelo y el VANT hace muchos movimientos ascendentes, debido a la altura detecta muchas ramas delgadas y finaliza aterrizando.
Vuelo 4	Si	30	Detecta sombras en el suelo y realiza movimientos ascendentes. Colisiona con una rama que se encuentra encima del VANT.
Vuelo 5	No	28	Detecta sombras en el suelo y árboles en todo el ancho de la imagen, termina aterrizando.
Vuelo 6	No	15	Al iniciar el vuelo detecta dos árboles muy cerca, hace un movimiento hacia la derecha, pero se detecta otro árbol y finaliza el vuelo aterrizando.
Vuelo 7	No	30	El dron no se estabiliza debido a una corriente de aire, detecta varios árboles y finaliza el aterrizando.

La Figura 5.14 ilustra los obstáculos identificados en cada vuelo de la tarde, que difieren de los observados en la mañana. Tres vuelos concluyeron poco después del despegue, ya que el sistema identificó las sombras de los árboles como obstáculos potenciales. El dron realizó movimientos ascendentes para sortearlos, lo que provocó colisiones con ramas fuera de su rango visual. Sin embargo, en el resto de los vuelos, el sistema logró evadir los diversos objetos que se encontraban en escena.

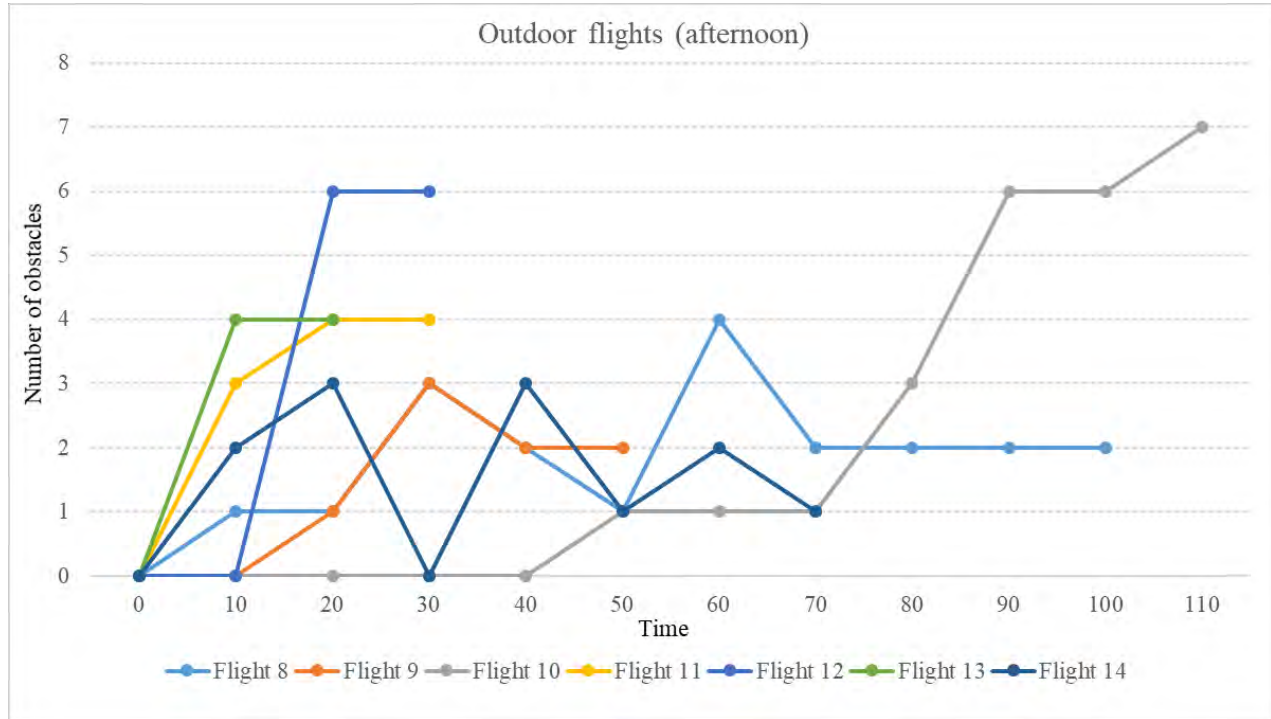
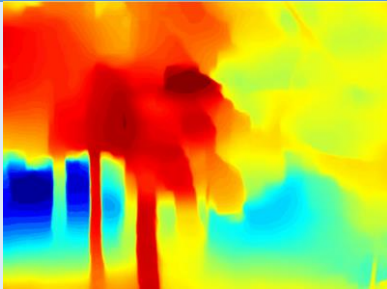
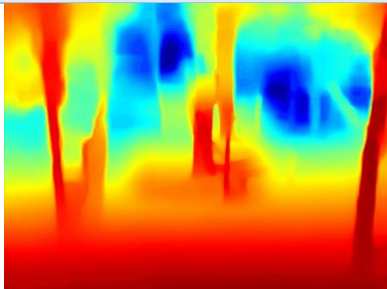
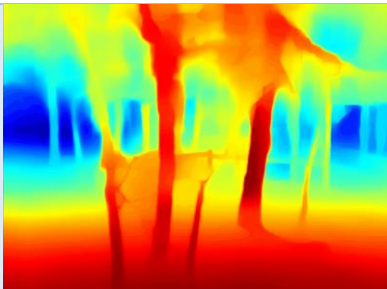
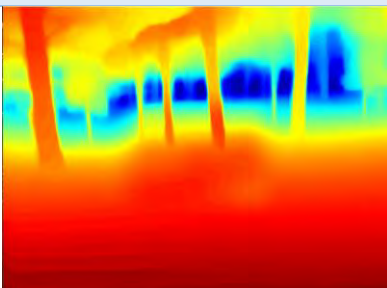
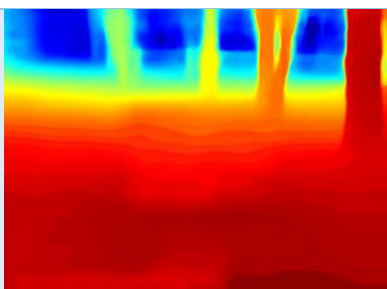


Figura 5.14 Número de obstáculos identificados por el sistema a lo largo de los vuelos al aire libre realizados durante la tarde.

Los resultados de los vuelos realizados al aire libre demuestran que la premisa de aterrizaje en caso de detectar algún obstáculo en las cinco posibles zonas de navegación de análisis es eficaz. En los 14 vuelos realizados, el VANT logró aterrizar con éxito en 11 ocasiones y, en ningún vuelo sufrió ninguna caída ni daño. Esto se logró proporcionando al VANT tiempo suficiente para evadir o aterrizar en caso de una futura colisión. Esto se determinó aplicando un umbral en el que los píxeles dentro del rango de 191 a 255 se consideran obstáculos.

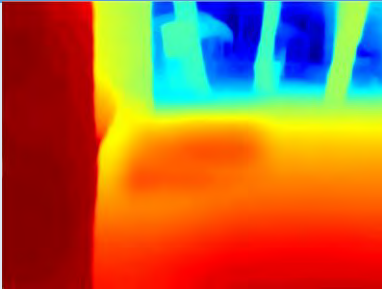
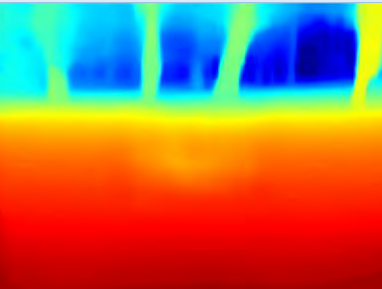
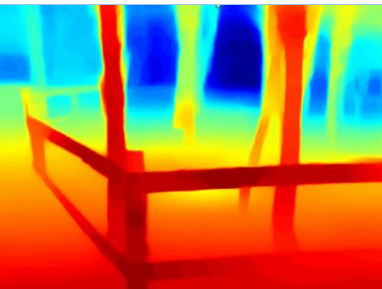
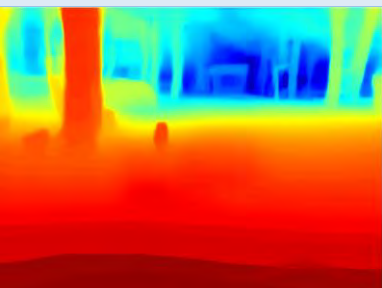
La Tabla 5.19 ilustra la eficacia del umbral definido en la detección de obstáculos. En los mapas de profundidad se definen con precisión las ramas delgadas y también se detectan los árboles muy lejanos. Aunque se observó que, en los vuelos realizados en el horario de la tarde, los mapas de profundidad se ven afectados por las sombras, se puede ver en las filas 3, 4 y 5 se puede ver que el mapa de profundidad asigna los píxeles en la zona de sombras más oscuros, teniendo como consecuencia movimientos del dron hacia arriba.

Tabla 5.19 Imágenes con obstáculos detectados

RGB	Mapa de profundidad	Obstáculo detectado
		
		
		
		
		

La Tabla 5.20 presenta ejemplos de vuelos, en los cuales el piso está claramente delineado y los árboles se pueden observar a una distancia considerable, pero no se identifican como posibles obstrucciones debido al umbral. A pesar de que el sistema detecta obstáculos en el área circundante, es capaz de identificar y utilizar el espacio libre para continuar su vuelo.

Tabla 5.20 Imágenes con espacio libre de navegación

RGB	Mapa de profundidad	Espacio libre
		
		
		
		

5.6 Comentarios finales

En la primera etapa de experimentación del sistema de navegación se realizaron vuelos en escenarios de interiores, los cuales ayudaron a identificar las limitaciones del VANT con los recursos computacionales disponibles, estas se listan a continuación:

- El VANT tiene un tiempo de respuesta de 1.4 a 3 segundos entre cada envío de comando de navegación.
- El sistema completo realiza el análisis de las cinco áreas posibles de navegación y la generación de comandos, por lo que se requieren más recursos de cómputo para una rápida respuesta.
- El modelo del VANT puede ser afectado por el tipo de superficie que sobrevuela, esto perturba la identificación de su posición, ocasionando movimientos erráticos.

Debido a las limitaciones del equipo de cómputo y el modelo del VANT utilizado, la configuración del sistema de navegación fue el siguiente: la red piramidal utilizó un modelo LeReS con una red troncal ResNet50, estimando solo 1 nivel de la pirámide.

Tomando en consideración que el VANT no cuenta con un tiempo de respuesta definido, el sistema espera a que el drone vuelva a estar disponible, para analizar la imagen que se recibe en ese momento y así generar el mapa de profundidad y el comando de navegación.

CAPÍTULO 6

Conclusiones

6.1 Conclusiones

La arquitectura propuesta para la estimación de profundidad ha mostrado un decremento en los tiempos de inferencia y en la obtención de los mapas de profundidad, manteniendo la precisión de la estimación. Esto se muestra en los resultados en donde el desempeño de la red propuesta es comparable con el desempeño de la red Boosting Monocular Depth original, pero disminuyendo considerablemente los tiempos de inferencia.

Utilizando el conjunto de datos TartanAir con sus diferentes trayectorias se pudo validar que la red es recomendable para ser utilizada en diversos ambientes de cultivos a una altura baja, ya que en los resultados obtenidos se observan buenos valores de precisión y profundidad.

Durante la experimentación del sistema de navegación evasiva de obstáculos se detectaron diversas limitaciones como resultado del modelo del VANT y los recursos de cómputo disponibles. Primero se realizaron pruebas experimentales en interiores debido a que el modelo del VANT es ligero y cualquier corriente de viento puede afectar la estabilidad del vuelo; además que el equipo de cómputo disponible requiere conexión a la corriente eléctrica para poder utilizar el procesamiento de la GPU al 100%.

En los vuelos experimentales en interiores, se tuvieron inconvenientes ya que el VANT es afectado por las superficies sobre las que vuela, la iluminación de la escena y la altura de vuelo; perturbando la identificación de su posición y ocasionando movimientos erráticos durante el vuelo.

Debido al tiempo de respuesta del VANT entre cada envío de comandos, para el algoritmo de detección de obstáculos en las cinco áreas posibles de navegación y la generación de comandos, se necesitó utilizar la red propuesta estimando solo un nivel de la pirámide con una red troncal ResNet50.

En la experimentación realizada en exteriores ya no existieron los inconvenientes ocasionados por las superficies sobre las que vuela el VANT. El drone mantenía una altura de vuelo constante y el sistema podía realizar los movimientos necesarios para evadir obstáculos y encontrar un espacio libre de navegación. Algunos mapas de profundidad mostraban las sombras de los árboles como obstáculos, el VANT solo realizaba movimientos hasta arriba o el sistema le comunicaba al drone el comando de aterrizaje, evitando de esta forma prevenir colisiones.

Por lo mencionado anteriormente se manifiesta que se cumplió el objetivo general de la propuesta de tesis; del análisis del estado del arte se consiguió formular un cambio en la arquitectura de una red neuronal para la estimación de profundidad, y con los mapas de profundidad obtenidos se identificó el espacio libre para la navegación automática de un vehículo aéreo.

En cuanto a los objetivos específicos se obtuvo la detección del espacio libre de navegación en escenarios naturales utilizando solo una cámara monocular al proponer una arquitectura de

estimación de profundidad capaz de definir los detalles finos con tiempos de inferencia menores a los reportados en el estado del arte. Con el diseño del sistema de navegación evasiva de obstáculos se formula la planeación de movimientos a partir de los mapas obtenidos con la red propuesta, permitiendo así orientar el vehículo aéreo dentro del espacio libre. Por último, se corroboró la capacidad de generalización de la red propuesta al evaluarla con las diferentes trayectorias de TartanAir, demostrando que la propuesta puede ser utilizada en diferentes tipos de cultivos.

6.3 Objetivos logrados

Los objetivos logrados se listan a continuación:

- Se propuso una arquitectura de una Red Neuronal Profunda para detectar el espacio libre para navegar, utilizando imágenes RGB.
- Se formuló una política de planeación de movimientos de navegación, en función de las predicciones de la red neuronal profunda.
- Se evaluó la capacidad de generalización del sistema propuesto en diversos escenarios de cultivos reales, distintos a los utilizados en el entrenamiento; mediante las métricas clásicas utilizadas en el área.

6.4 Aportaciones

La aportación del trabajo se lleva a cabo en dos aspectos:

- a) Al considerar la navegación del VANT a baja altura por los senderos de los cultivos reales, las imágenes a analizar pueden presentar escenarios más complejos que los encontrados en el estado del arte, ya que los objetos que se consideran en esta tesis son distintos tipos de ramas, personas, y equipo utilizado por los agricultores. Asimismo, se trabaja en condiciones de iluminación variante debido a que el dron navegará entre las plantas, lo cual puede ocasionar que el dron pase de una zona con sombra a una con bastante iluminación y viceversa, de un momento a otro.
- b) En la solución al problema se modificó la arquitectura de red neuronal profunda para crear mapas de profundidad usando como entrada imágenes monoculares que permiten detectar de forma rápida y precisa el espacio libre para navegar en ambientes de cultivos. Se analiza la manera de usar el conocimiento extraído de los mapas de profundidad obtenidos en entornos de interiores y urbanos, los cuales de acuerdo al estado del arte son bastos, para poderlo transferir al entorno de interés.

6.5 Trabajo futuro

Como trabajos futuros se tienen los siguientes puntos:

- Implementar un método de cálculo de la distancia del objeto detectado y el VANT con más precisión.
- Creación de un banco de imágenes de cultivos a baja altitud y con variaciones de iluminación.
- Realizar vuelos en diversos cultivos a baja altitud para verificar con datos reales la generalización de la red de estimación de profundidad y la eficacia del sistema para evadir obstáculos.

6.6 Productos académicos adicionales

Se listan todos los productos académicos derivados del presente proyecto, que se han logrado en el transcurso de estos tres años:

- Presentación y publicación del artículo “A review on deep learning UAV systems for visual obstacle detection in crop environments” en el Congreso Internacional de Ingeniería Mecatrónica, Electrónica y Automotriz (ICMEAE), realizado del 22 al 26 de noviembre del 2021 en la ciudad de Emiliano Zapata, Morelos, México. Las memorias las edita el IEEEExplore y tiene índice SCOPUS. DOI: <https://doi.org/10.1109/ICMEAE55138.2021.00024>
- Participación en la exposición de posters con el trabajo titulado “Aprendizaje Profundo para la Navegación Aérea en Entornos Naturales” en el Capítulo de Inteligencia Computacional y Robótica del IEEE de la Sección Morelos realizado del 16 al 20 de agosto de 2022.
- Publicación de un artículo de revista JCR, titulado “A comparison of deep neural networks for monocular depth map estimation in natural environments flying at low altitude” en la revista Sensors de la editorial MDPI (siglas en inglés, Multidisciplinary Digital Publishing Institute) para el número especial “Deep Learning Methods for Aerial Imagery. La revista Sensors es JCR – Q2 (Instruments & Instrumentation) teniendo un factor de impacto de 3.847 y un factor de impacto de 5 años de 4.050. DOI: <https://doi.org/10.3390/s22249830>
- Participación en el Simposio Internacional de Ingeniería en Sistemas Computacionales 2023 del Tecnológico de Cuautla, en la Jornada de difusión y divulgación científica, realizado del 27 al 31 de marzo.
- Publicación el artículo de revista JCR titulado “Redes neuronales residuales para la estimación de profundidad monocular en ambientes naturales” para la revista DYNA New Technologies de la editorial DYNA España. DOI: <https://doi.org/10.6036/NT11086>
- Se redactó el artículo de revista JCR titulado “Boosting Pyramidal Depth Network for UAV Obstacle Avoidance in Unknown Environments with Natural Objects” para la revista Smart Agricultural Technology de la editorial ELSEVIER. La revista Smart Agricultural Technology es JCR teniendo un factor de impacto de 6.3. Dicho artículo fue enviado el 12 de diciembre del presente año y actualmente se encuentra en proceso de evaluación.

Referencias

- [1] P. Radoglou-Grammatikis, P. Sarigiannidis, T. Lagkas, and I. Moscholios, "A compilation of UAV applications for precision agriculture," *Comput. Networks*, vol. 172, 2020, doi: 10.1016/j.comnet.2020.107148.
- [2] C. H. R. Everett, "Survey of collision avoidance and ranging sensors for mobile robots," *Rob. Auton. Syst.*, vol. 5, no. 1, 1989, doi: 10.1016/0921-8890(89)90041-9.
- [3] P. Fraga-Lamas, L. Ramos, V. Mondéjar-Guerra, and T. M. Fernández-Caramés, "A review on IoT deep learning UAV systems for autonomous obstacle detection and collision avoidance," *Remote Sensing*. 2019. doi: 10.3390/rs11182144.
- [4] H. Y. Lee, H. W. Ho, and Y. Zhou, "Deep Learning-based Monocular Obstacle Avoidance for Unmanned Aerial Vehicle Navigation in Tree Plantations," *J. Intell. Robot. Syst.*, vol. 101, no. 1, 2021, doi: 10.1007/s10846-020-01284-z.
- [5] K. Amer, M. Samy, M. Shaker, and M. ElHelw, "Deep convolutional neural network-based autonomous drone navigation," *arXiv*. 2019. doi: 10.1117/12.2587105.
- [6] PyTorch, "PyTorch documentation - PyTorch master documentation," *PyTorch*, 2019.
- [7] F. Chollet and & O., "Keras: the Python deep learning API," *Keras: the Python deep learning API*, 2020.
- [8] Microsoft Research, "AirSim," *Available Online*, 2021. <https://microsoft.github.io/AirSim/> (accessed Jun. 05, 2021).
- [9] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Rob. Res.*, vol. 32, no. 11, 2013, doi: 10.1177/0278364913491297.
- [10] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012. doi: 10.1007/978-3-642-33715-4_54.
- [11] W. Wang *et al.*, "TartanAir: A Dataset to Push the Limits of Visual SLAM," in *IEEE International Conference on Intelligent Robots and Systems*, 2020. doi: 10.1109/IROS45743.2020.9341801.
- [12] A. Rosebrock, "Deep Learning for Computer Vision with Python - Starter Bundle," *Pyimagesearch*, 2017.
- [13] J. Dean, "Large-Scale Deep Learning For Building Intelligent Computer Systems," 2016. doi: 10.1145/2835776.2835844.
- [14] O. Faust, Y. Hagiwara, T. J. Hong, O. S. Lih, and U. R. Acharya, "Deep learning for healthcare applications based on physiological signals: A review," *Computer Methods and Programs in Biomedicine*, vol. 161. 2018. doi: 10.1016/j.cmpb.2018.04.005.
- [15] M. Tim Jones, "Deep learning architectures: The rise of artificial intelligence," *IBM Corp.*, 2017.
- [16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, 2017, doi: 10.1145/3065386.
- [17] L. Tai, S. Li, and M. Liu, "A deep-network solution towards model-less obstacle avoidance," in *IEEE International Conference on Intelligent Robots and Systems*, 2016. doi: 10.1109/IROS.2016.7759428.
- [18] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Computers and Electronics in Agriculture*, vol. 147. 2018. doi: 10.1016/j.compag.2018.02.016.
- [19] A. Singla, S. Padakandla, and S. Bhatnagar, "Memory-Based Deep Reinforcement Learning

- for Obstacle Avoidance in UAV with Limited Environment Knowledge,” *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 1, 2021, doi: 10.1109/TITS.2019.2954952.
- [20] W. Pei, T. Baltrušaitis, D. M. J. Tax, and L. P. Morency, “Temporal attention-Gated model for robust sequence classification,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017. doi: 10.1109/CVPR.2017.94.
- [21] Y. Lecun, Y. Bengio, and G. Hinton, “nature-Deep learning,” *Nature*, vol. 521, no. 7553, 2015.
- [22] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, 1997, doi: 10.1162/neco.1997.9.8.1735.
- [23] S. Yang, S. Konam, C. Ma, S. Rosenthal, M. Veloso, and S. Scherer, “Obstacle avoidance through deep networks based intermediate perception,” *arXiv*. 2017.
- [24] S. Dionisio-Ortega, L. O. Rojas-Perez, J. Martinez-Carranza, and I. Cruz-Vega, “A deep learning approach towards autonomous flight in forest environments,” in *2018 28th International Conference on Electronics, Communications and Computers, CONIELECOMP 2018*, 2018. doi: 10.1109/CONIELECOMP.2018.8327189.
- [25] A. Loquercio, A. I. Maqueda, C. R. Del-Blanco, and D. Scaramuzza, “DroNet: Learning to Fly by Driving,” *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, 2018, doi: 10.1109/LRA.2018.2795643.
- [26] P. H. Chen and C. Y. Lee, “UAVNet: An Efficient Obstacle Detection Model for UAV with Autonomous Flight,” in *2018 International Conference on Intelligent Autonomous Systems, ICoIAS 2018*, 2018. doi: 10.1109/ICoIAS.2018.8494201.
- [27] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017. doi: 10.1109/CVPR.2017.243.
- [28] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016. doi: 10.1109/CVPR.2016.308.
- [29] A. Khan and M. Hebert, “Learning safe recovery trajectories with deep neural networks for unmanned aerial vehicles,” in *IEEE Aerospace Conference Proceedings*, 2018. doi: 10.1109/AERO.2018.8396807.
- [30] A. Kouris and C. S. Bouganis, “Learning to Fly by MySelf: A Self-Supervised CNN-Based Approach for Autonomous Navigation,” in *IEEE International Conference on Intelligent Robots and Systems*, 2018. doi: 10.1109/IROS.2018.8594204.
- [31] N. Smolyanskiy, A. Kamenev, J. Smith, and S. Birchfield, “Toward low-flying autonomous MAV trail navigation using deep neural networks for environmental awareness,” in *IEEE International Conference on Intelligent Robots and Systems*, 2017. doi: 10.1109/IROS.2017.8206285.
- [32] N. D. S. Rosa, V. Guizilini, and V. Grassi, “Sparse-to-continuous: Enhancing monocular depth estimation using occupancy maps,” in *2019 19th International Conference on Advanced Robotics, ICAR 2019*, 2019. doi: 10.1109/ICAR46387.2019.8981652.
- [33] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, “Deep Ordinal Regression Network for Monocular Depth Estimation,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018. doi: 10.1109/CVPR.2018.00214.
- [34] Z. Teed and Deng Jia, “DeepV2D: Video to depth with differentiable structure from motion,” in *The International Conference on Learning Representations*, Addis Ababa Ethiopia, 2020.
- [35] Y. Chen, H. Zhao, Z. Hu, and J. Peng, “Attention-based context aggregation network for monocular depth estimation,” *Int. J. Mach. Learn. Cybern.*, vol. 12, no. 6, 2021, doi: 10.1007/s13042-020-01251-y.
- [36] I. Alhashim and P. Wonka, “High quality monocular depth estimation via transfer learning,” *arXiv*. 2018.

- [37] W. Yin, Y. Liu, C. Shen, and Y. Yan, “Enforcing geometric constraints of virtual normal for depth prediction,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019. doi: 10.1109/ICCV.2019.00578.
- [38] J. H. Lee, M. K. Han, D. W. Ko, and I. H. Suh, “From big to small: Multi-scale local planar guidance for monocular depth estimation,” *arXiv*, 2019.
- [39] D. Wofk, F. Ma, T. J. Yang, S. Karaman, and V. Sze, “FastDepth: Fast monocular depth estimation on embedded systems,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2019. doi: 10.1109/ICRA.2019.8794182.
- [40] S. Zhao, H. Fu, M. Gong, and D. Tao, “Geometry-aware symmetric domain adaptation for monocular depth estimation,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019. doi: 10.1109/CVPR.2019.01002.
- [41] M. Goldman, T. Hassner, and S. Avidan, “Learn stereo, infer mono: Siamese networks for self-supervised, monocular, depth estimation,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2019. doi: 10.1109/CVPRW.2019.00348.
- [42] L. Andraghetti *et al.*, “Enhancing Self-Supervised Monocular Depth Estimation with Traditional Visual Odometry,” in *Proceedings - 2019 International Conference on 3D Vision, 3DV 2019*, 2019. doi: 10.1109/3DV.2019.00054.
- [43] R. Garg, B. G. Vijay Kumar, G. Carneiro, and I. Reid, “Unsupervised CNN for single view depth estimation: Geometry to the rescue,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016. doi: 10.1007/978-3-319-46484-8_45.
- [44] D. Eigen and R. Fergus, “Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-scale Convolutional Architecture,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2650–2658. doi: 10.1109/ICCV.2015.304.
- [45] F. Tosi, F. Aleotti, M. Poggi, and S. Mattoccia, “Learning monocular depth estimation infusing traditional stereo knowledge,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019. doi: 10.1109/CVPR.2019.01003.
- [46] V. Guizilini, R. Ambrus, S. Pillai, and A. Gaidon, “PackNet-SfM: 3D Packing for Self-Supervised Monocular Depth Estimation,” *arXiv*, 2019.
- [47] C. Godard, O. Mac Aodha, M. Firman, and G. Brostow, “Digging into self-supervised monocular depth estimation,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019. doi: 10.1109/ICCV.2019.00393.
- [48] J. Xie, R. Girshick, and A. Farhadi, “Deep3D: Fully automatic 2D-to-3D video conversion with deep convolutional neural networks,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016. doi: 10.1007/978-3-319-46493-0_51.
- [49] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, “Sparsity Invariant CNNs,” in *Proceedings - 2017 International Conference on 3D Vision, 3DV 2017*, 2018. doi: 10.1109/3DV.2017.00012.
- [50] J. V. Stafford, “Implementing precision agriculture in the 21st century,” *J. Agric. Eng. Res.*, vol. 76, no. 3, 2000, doi: 10.1006/jaer.2000.0577.
- [51] S. Chiesa, M. Fioriti, and R. Fusaro, “MALE UAV and its systems as basis of future definitions,” *Aircr. Eng. Aerosp. Technol.*, vol. 88, no. 6, 2016, doi: 10.1108/AEAT-08-2014-0131.
- [52] DOD, “Unmanned Systems Integrated Roadmap,” *Dep. Def.*, no. 2038, 2013.
- [53] B. Vergouw, H. Nagel, G. Bondt, and B. Custers, “Drone Technology: Types, Payloads, Applications, Frequency Spectrum Issues and Future Developments,” 2016. doi: 10.1007/978-

- 94-6265-132-6_2.
- [54] G. Sylvester, *E-agriculture in action: Drones for agriculture*. Bangkok, Thailand: Food and Agriculture Organization of the United Nations and International Telecommunication Union, 2018.
 - [55] P. Skobelev, D. Budaev, N. Gusev, and G. Voschuk, “Designing multi-agent swarm of UAV for precise agriculture,” in *Communications in Computer and Information Science*, 2018. doi: 10.1007/978-3-319-94779-2_5.
 - [56] T. Lagkas, V. Argyriou, S. Bibi, and P. Sarigiannidis, “UAV IoT framework views and challenges: Towards protecting drones as ‘things,’” *Sensors (Switzerland)*, vol. 18, no. 11. 2018. doi: 10.3390/s18114015.
 - [57] Y. Lu, Z. Xue, G. S. Xia, and L. Zhang, “A survey on vision-based UAV navigation,” *Geo-Spatial Inf. Sci.*, vol. 21, no. 1, 2018, doi: 10.1080/10095020.2017.1420509.
 - [58] N. Gageik, P. Benz, and S. Montenegro, “Obstacle detection and collision avoidance for a UAV with complementary low-cost sensors,” *IEEE Access*, vol. 3, 2015, doi: 10.1109/ACCESS.2015.2432455.
 - [59] F. Khan, S. Salahuddin, and H. Javidnia, “Deep learning-based monocular depth estimation methods—a state-of-the-art review,” *Sensors (Switzerland)*, vol. 20, no. 8. 2020. doi: 10.3390/s20082272.
 - [60] Y. D. V. Yasuda, L. E. G. Martins, and F. A. M. Cappabianco, “Autonomous visual navigation for mobile robots: A systematic literature review,” *ACM Computing Surveys*, vol. 53, no. 1. 2020. doi: 10.1145/3368961.
 - [61] D. Xu, W. Wang, H. Tang, H. Liu, N. Sebe, and E. Ricci, “Structured Attention Guided Convolutional Neural Fields for Monocular Depth Estimation,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018. doi: 10.1109/CVPR.2018.00412.
 - [62] Y. Kuznetsov, J. Stückler, and B. Leibe, “Semi-supervised deep learning for monocular depth map prediction,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017. doi: 10.1109/CVPR.2017.238.
 - [63] S. Bazrafkan, H. Javidnia, and J. Lemley, “Semiparallel deep neural network hybrid architecture: first application on depth from monocular camera,” *J. Electron. Imaging*, vol. 27, no. 04, 2018, doi: 10.1117/1.jei.27.4.043041.
 - [64] K. Xian, J. Zhang, O. Wang, L. Mai, Z. Lin, and Z. Cao, “Structure-Guided Ranking Loss for Single Image Depth Prediction,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020. doi: 10.1109/CVPR42600.2020.00069.
 - [65] S. H. M. Miangoleh, S. Dille, L. Mai, S. Paris, and Y. Aksoy, “Boosting Monocular Depth Estimation Models to High-resolution via Content-adaptive Multi-Resolution Merging,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2021. doi: 10.1109/CVPR46437.2021.00956.
 - [66] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, “Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, 2020, doi: 10.1109/TPAMI.2020.3019967.
 - [67] R. Ranftl, A. Bochkovskiy, and V. Koltun, “Vision Transformers for Dense Prediction,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2021. doi: 10.1109/ICCV48922.2021.01196.
 - [68] Z. Zhou, X. Fan, P. Shi, and Y. Xin, “R-MSFM: Recurrent Multi-Scale Feature Modulation for Monocular Depth Estimating,” *Iccv*, 2021.
 - [69] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proceedings - 30th IEEE Conference on Computer Vision and*

- Pattern Recognition, CVPR 2017*, 2017. doi: 10.1109/CVPR.2017.632.
- [70] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015. doi: 10.1007/978-3-319-24574-4_28.
- [71] W. Yin *et al.*, “Learning to Recover 3D Scene Shape from a Single Image,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2021. doi: 10.1109/CVPR46437.2021.00027.
- [72] D. Kim, W. Ga, P. Ahn, D. Joo, S. Chun, and J. Kim, “Global-Local Path Networks for Monocular Depth Estimation with Vertical CutDepth,” Jan. 2022.
- [73] D. Scharstein and R. Szeliski, “High-accuracy stereo depth maps using structured light,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003. doi: 10.1109/cvpr.2003.1211354.
- [74] T. Koch, L. Liebel, F. Fraundorfer, and M. Körner, “Evaluation of CNN-based single-image depth estimation methods,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019. doi: 10.1007/978-3-030-11015-4_25.
- [75] A. Zamir, A. Sax, W. Shen, L. Guibas, J. Malik, and S. Savarese, “Taskonomy: Disentangling task transfer learning,” in *IJCAI International Joint Conference on Artificial Intelligence*, 2019. doi: 10.24963/ijcai.2019/871.
- [76] S. Kim, D. Min, B. Ham, S. Kim, and K. Sohn, “Deep stereo confidence prediction for depth estimation,” in *Proceedings - International Conference on Image Processing, ICIP*, 2017. doi: 10.1109/ICIP.2017.8296430.
- [77] Y. Hua *et al.*, “Holopix50k: A Large-Scale In-the-wild Stereo Image Dataset,” Mar. 2020.
- [78] S. Niklaus, L. Mai, J. Yang, and F. Liu, “3D ken burns effect from a single image,” *ACM Trans. Graph.*, vol. 38, no. 6, 2019, doi: 10.1145/3355089.3356528.
- [79] R. P. K. Poudel, U. Bonde, S. Liwicki, and C. Zach, “ContextNet: Exploring context and detail for semantic segmentation in real-time,” in *British Machine Vision Conference 2018, BMVC 2018*, 2019.
- [80] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” *CoRR*, vol. abs/1804.0, 2018, [Online]. Available: <http://arxiv.org/abs/1804.02767>
- [81] S. W. Kim, H. K. Kook, J. Y. Sun, M. C. Kang, and S. J. Ko, “Parallel Feature Pyramid Network for Object Detection,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018. doi: 10.1007/978-3-030-01228-1_15.
- [82] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, 2015, doi: 10.1109/TPAMI.2015.2389824.
- [83] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, 2018, doi: 10.1109/TPAMI.2017.2699184.
- [84] A. Araujo, W. Norris, and J. Sim, “Computing Receptive Fields of Convolutional Neural Networks,” *Distill*, vol. 4, no. 11, 2019, doi: 10.23915/distill.00021.
- [85] H. Le and A. Borji, “What are the Receptive, Effective Receptive, and Projective Fields of Neurons in Convolutional Neural Networks?” 2018.
- [86] Z. A. Aziz, D. Naseradeen Abdulqader, A. B. Sallow, and H. Khalid Omer, “Python Parallel Processing and Multiprocessing: A Review,” *Acad. J. Nawroz Univ.*, vol. 10, no. 3, pp. 345–

-
- 354, Aug. 2021, doi: 10.25007/ajnu.v10n3a1145.
- [87] S. Aydin, R. Samet, and O. F. Bay, “Real-time parallel image processing applications on multicore CPUs with OpenMP and GPGPU with CUDA,” *J. Supercomput.*, vol. 74, no. 6, pp. 2255–2275, Jun. 2018, doi: 10.1007/s11227-017-2168-6.
- [88] A. Chaudhary, K. S. Chouhan, J. Gajrani, and B. Sharma, “Deep learning with PyTorch,” in *Machine Learning and Deep Learning in Real-Time Applications*, 2020. doi: 10.4018/978-1-7998-3095-5.ch003.
- [89] T. Hachaj, “Potential Obstacle Detection Using RGB to Depth Image Encoder–Decoder Network: Application to Unmanned Aerial Vehicles,” *Sensors*, vol. 22, no. 17, 2022, doi: 10.3390/s22176703.
- [90] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise,” in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, 1996.
- [91] J. L. Damià Fuentes Escoté, “DJITelloPy package,” <https://github.com/damiafuentes/DJITelloPy/>, Nov. 18, 2018.