



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS
AVANZADOS DEL INSTITUTO POLITÉCNICO NACIONAL

Unidad Zacatenco

Programa de
Sistemas Autónomos de Navegación Aérea y Submarina

**Monitoreo a partir del seguimiento de un camino
con un cuadri-rotor**

TESIS

Que presenta

DONOVAN FLORES MEZA

Para obtener el grado de

DOCTOR EN CIENCIAS

En

Sistemas Autónomos de Navegación Aérea y Submarina

Directores de la Tesis:

Dr. Rogelio Lozano Leal

Dr. Iván González Hernández

Ciudad de México

Marzo de 2021

AGRADECIMIENTOS

Quisiera agradecer en primer lugar a toda mi familia por todo el apoyo que me han dado siempre para seguirme superando.

También me gustaría agradecer a mis asesores de tesis el Dr. Rogelio Lozano y el Dr. Iván Gonzalez, y al coordinador el Dr. Sergio Salazar por su guía a lo largo de mi estancia dentro del laboratorio. Sus consejos me han ayudado a la realización de todo el trabajo.

Agradezco a mis compañeros y amigos del laboratorio UMI-LAFMIA por sus consejos en lo personal y académico. A todos mis demás amigos del CINVESTAV que han hecho más amena mi estancia en este gran centro de investigación.

A mis colegas, socios y amigos de Desarrollo Tecnológico DOI, Orlando e Irving les agradezco la confianza y todo el apoyo que he recibido de ustedes desde que éramos compañeros en la maestría hasta ahora.

Al CINVESTAV y al laboratorio UMI-LAFMIA gracias por la oportunidad de formarme dentro de sus instalaciones y de este programa doctoral.

Y finalmente quisiera agradecer al Consejo Nacional de Ciencia y Tecnología, por el apoyo económico recibido a lo largo del doctorado.

RESUMEN

Existe un amplio rango de aplicación para los vehículos aéreos no tripulados, dentro de los campos más explorados están recreación y fotogrametría. Sin embargo, el nivel de autonomía de estos depende de muchos factores como señal de GPS, y si bien es cierto ya existe tecnología que empieza a atacar este problema, no llega a resolver del todo problemas específicos. La navegación de estos dispositivos se está volviendo algo crucial ya que al poderse realizar de forma autónoma deja la opción a que la persona pueda enfocar la atención en atender el problema en específico.

Para este trabajo se resuelve la tarea de navegación por medio del seguimiento de un camino geográficamente bien definido, detectando los bordes laterales por medio de un sensor laser rotatorio. La propuesta contempla que el vehículo realice la tarea de monitoreo mientras realiza la navegación, para esto se hace uso de algoritmos de visión por computadora que se están ejecutando de forma embarcada en la plataforma para con esto darle un mayor impacto en cuestión tecnológica y de autonomía, ya que al hacer el procesamiento de forma embebida ofrece mayor versatilidad y permitiría la realización de tareas más complejas.

La parte de visión por computadora envuelve varias secciones que trabajan en conjunto para lograr realizar la tarea de monitoreo de forma correcta. Para ello primero se fija un problema actual en el área de agricultura. El agave presenta un problema de seguridad en el cual el uso de VANT's para monitoreo se proyecta como una solución coherente y congruente ante este problema.

La solución que se propone es un vehículo tipo multi-rotor provisto de un sensor laser rotatorio que sirve para detectar el área de los surcos para seguirlos de forma autónoma. Al mismo tiempo que se hace la navegación, un algoritmo de visión se encarga de hacer detección y conteo de los agaves presentes en la imagen capturada por una cámara a bordo de la plataforma. Para la parte de detección de las plantas de agave se hace uso de aprendizaje profundo que forma parte de aprendizaje para máquinas, con esta técnica se logra hacer el conteo de plantas de forma automática mientras se realiza la navegación.

ABSTRACT

There is a wide range of application for unmanned aerial vehicles, among the most explored is for recreation and photogrammetry. However, the level of autonomy of these depends on many factors such as GPS signal, and if it is true there is already technology that begins to attack this problem, it does not quite solve specific problems.

The navigation of these devices is becoming something crucial since being able to perform autonomously leaves the option for the person to focus attention on addressing the specific problem. For this work, it is solved the navigation task by following a geographically well-defined path, detecting the lateral edges by means of a rotating laser sensor.

The proposal contemplates that the vehicle carries out the monitoring task while navigating, for this, computer vision algorithms are being used that are being shipped on board the platform to give it a greater impact in terms of technology and autonomy. , since by doing the processing in an embedded way it offers greater versatility and would allow the performance of more complex tasks. The computer vision part involves several sections that work together to accomplish the monitoring task correctly. For this, a current problem is first fixed in the area of agriculture.

The agave presents a security problem in which the use of VANT's for monitoring is projected as a coherent and consistent solution to this problem. The proposed solution is a multicopter type vehicle equipped with a rotating laser sensor that serves to detect the area of the grooves to follow them autonomously, at the same time that navigation is done a vision algorithm is responsible for detecting and counting of the agaves present in the image captured by a camera on board the platform.

For the detection part of agave plants, deep learning is used, which is part of machine learning, with this technique it is possible to automatically count the plants while browsing.

ÍNDICE GENERAL

AGRADECIMIENTOS	iii
RESUMEN	v
ABSTRACT	vii
ÍNDICE GENERAL	ix
1 INTRODUCCIÓN	1
1.1 Estado del arte	3
1.2 Motivación	7
1.3 Planteamiento del problema	8
1.4 Propuesta de solución al problema planteado	8
1.5 Justificación	9
1.6 Objetivo general	9
1.6.1 Objetivos particulares	9
1.7 Organización del escrito	10
2 MONITOREO	13
2.1 Reconocimiento de objetos	13
2.1.1 Aprendizaje de máquinas	14
2.2 Métricas de evaluación de desempeño de detección de objetos	22
2.3 Método de detección propuesto	23

2.3.1	Segmentación por color	25
2.3.2	Detección del centro del agave	26
2.3.3	Detección por CNN y conteo	28
2.4	Base de datos de agave	29
3	NAVEGACIÓN A TRAVÉS DEL SEGUIMIENTO DE UN CAMINO	33
3.1	Detección de los bordes del camino	34
4	PLATAFORMA EXPERIMENTAL	41
4.1	Diseño y construcción	42
4.2	Instrumentación	47
4.2.1	Autopiloto Pixhawk	47
4.3	GPS	49
4.3.1	RPLidar	50
4.3.2	Odroid XU4	52
4.4	Modelado del Cuadri-rotor	53
4.4.1	Implementación de un controlador PD	57
5	RESULTADOS EXPERIMENTALES	65
5.1	Detección de Agaves	65
5.2	Discusión	74
6	CONCLUSIONES Y TRABAJO FUTURO	77
6.1	Conclusiones	77
6.2	Trabajo futuro	79
6.3	Artículos	81
A	APÉNDICE A	83
A.1	PVTOL global stabilization using a nested saturation control	83
A.1.1	PVTOL Model	83
A.1.2	Upperbound on the orientation rate $\dot{\theta}$	84

A.1.3	Upperbound on the orientation angle θ .	85
A.1.4	Thrust control input u	86
A.1.5	Upperbound on the velocity \dot{x}	87
A.1.6	Convergence of the position x	89
A.1.7	Numerical simulation results	93
A.1.8	Conclusion	95
A.1.9	Bound for $x_4 = \dot{\theta}$	98
A.1.10	Bound for $\nu_1 = \theta + \dot{\theta}$	99
A.1.11	Bound for θ	100
A.1.12	Bound for e_1 in (A.25)	102
A.1.13	Bound for ν_2	104
A.1.14	Choice of k and n	104
BIBLIOGRAFÍA		107

CAPÍTULO 1

INTRODUCCIÓN

Los vehículos aéreos no tripulados ofrecen una gran versatilidad a la hora de recorrer grandes distancias en un corto tiempo, de aquí las múltiples aplicaciones y el auge que han tenido en las últimas décadas. Además, se han popularizado por el uso en conjunto de cámaras a bordo y poder conseguir tomas aéreas de forma fácil. Esta característica ofrece un amplio abanico de posibilidades para usar estas imágenes, desde fotografía artística, topografía, vigilancia, etc. La vigilancia o monitoreo de grandes extensiones es una aplicación que en particular resulta muy interesante ya que abarca de forma rápida un área y ofrece seguridad para el vigilante ya que no es expuesto a los peligros que podría representar hacer la tarea de forma física.

Dentro de la agricultura de precisión estos vehículos se han empleado principalmente para riego de fertilizantes y pesticidas. Sin embargo, aplicaciones como el monitoreo pueden ser cruciales para preservar la integridad de la cosecha, en este trabajo se propone un monitoreo contando las plantas en un área específica, con esto obtenemos una estimación del volumen de la cosecha y agregamos seguridad para las personas que cuidan los plantíos.

En México, el agave atraviesa una grave situación de seguridad, el robo de esta planta ha aumentado en los últimos años. Esta especie es nativa del continente americano, y tiene una amplia variedad de usos, pero se utiliza principalmente para la producción de tequila y mezcal. El agave tiene la propiedad de ser una planta de lenta maduración, el tiempo mínimo

de maduración para esta especie es de 8 años y puede demorar hasta 15 años. Otro de los inconvenientes que presenta es que su cosecha requiere cortar prácticamente toda la planta, lo que significa que solo se cosecha una vez por cada vez que se planta. En ocasiones los ladrones suplantan las plantas maduras por plantas jóvenes, lo que hace más difícil localizar en específico donde ocurren los robos.



Fig. 1.1: Jimador de agave

En los últimos años, ha habido un aumento en la tasa de robo en las plantaciones de agave. Solo en 2016, se registraron robos por 193 toneladas de agave azul equivalentes a pérdidas de casi 3 millones de pesos mexicanos, según el Consejo Regulador del Tequila (CRT) para el año 2017, esta cantidad aumentó más del triple (Fig. 1.2). Este problema se ha incrementado por muchas razones, pero una de ellas se debe a la dificultad de mantener las parcelas supervisadas con regularidad.

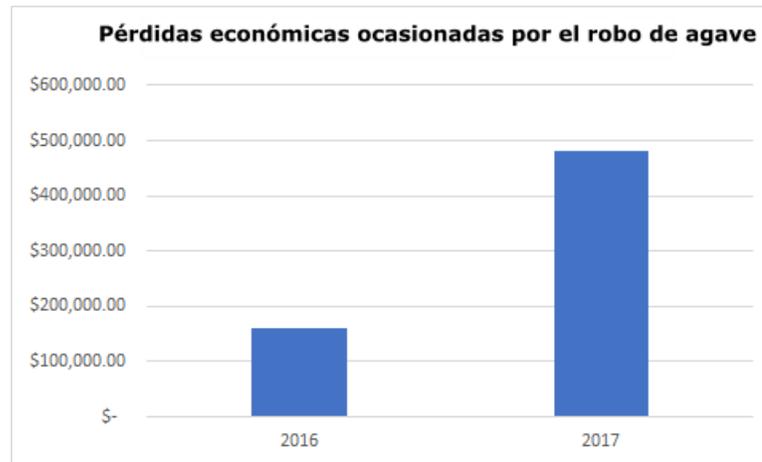


Fig. 1.2: Comparación de pérdidas monetarias entre los años 2016 y 2017

1.1 Estado del arte

La detección y el seguimiento de objetos por medio de visión artificial se han convertido en un recurso bien conocido en todo el mundo, principalmente porque tienen una gran variedad de aplicaciones [45]. Por ejemplo, se pueden utilizar para el reconocimiento de personas o automóviles que cruzan un área determinada. Estos aspectos sin duda serán necesarios para lograr la conducción de vehículos autónomos en zonas urbanas. La detección y el seguimiento de objetos también pueden ser útiles en otras aplicaciones como la agricultura de precisión, que se destacarán en el presente documento. La vigilancia de una plantación antes de la cosecha es una tarea bastante complicada que a menudo se realiza de forma empírica. Sin embargo, el rendimiento dependerá en gran medida del número de plantas maduras durante un largo período de tiempo. El robo de plantas representa un problema enorme al que se enfrentan los propietarios de plantaciones. Por tanto, es necesario desarrollar técnicas de seguimiento de las plantas en un campo determinado. Sería útil poder controlar continuamente el crecimiento de las plantas, así como verificar si algunas plantas han sido robadas. Esto ocurre generalmente en las grandes plantaciones, que son más difíciles de monitorear.

Existen varios puntos de vista sobre cómo implementar mejor la tecnología en la agricul-

tura como [11], [31], [38]. Es difícil elegir solo uno, ya que todas las técnicas ofrecen algo nuevo en su aplicación específica. Sin embargo, todos estos trabajos recientes que vamos a mencionar coinciden en el uso de la visión por computadora y las imágenes tomadas por un UAV. El hardware específico de cada uno depende de la especie de planta que se estudia o la misión a realizar. En cuanto a la agricultura de precisión en Mexico podemos encontrar trabajos como los de [22] donde el autor precisa la necesidad en el país de este tipo de tecnologías, trabajos donde de forma similar se trata de resaltar la necesidad de nuevas tecnologías en el campo mexicano se pueden encontrar en [34].

Hay diferentes configuraciones para los VANT y cada una ofrece diferentes características. Dentro de la literatura reciente, encontramos el trabajo de [23], que describe las ventajas y desventajas de las dos configuraciones principales, que son ala fija y multirrotor. Una de las conclusiones de este trabajo es que los vehículos de ala fija ofrecen un mayor tiempo de vuelo, pero requieren una mayor altitud debido a la velocidad a la que viajan. Por otro lado, los multirrotos tienen mayores ventajas para vuelos de baja altitud, pero la duración del vuelo disminuye con respecto a los aviones de ala fija. Dado que el objetivo principal es poder contar el agave con precisión, se requerirá de una aeronave que vuele a baja altura, lo que también abriría la posibilidad de rastrear los surcos que se forman y podría ser utilizado como referencia para la navegación. La aeronave debe tener sensores a bordo para realizar esta función. Por tanto, la mejor opción es una plataforma multirrotor. Por lo tanto, realizaremos vuelos utilizando múltiples rotores para recopilar los datos necesarios. Los vuelos deben planificarse con anticipación siguiendo algunas reglas y consideraciones como se describe en [5]. Los caminos que se forman entre las hileras de agave se consideran surcos, y estos son generalmente paralelos entre sí y paralelos a las hileras de agave. En primer lugar, debemos nombrar las obras donde se implementa la visión por computadora [6], [4], [35]. [6] propone el uso de un algoritmo de etapas múltiples para detectar plantas de tabaco, donde hay etapas que extraen regiones candidatas mediante operaciones morfológicas, segmentación y aprendizaje profundo.

El trabajo reportado en [6] trata de la detección de plantas de tabaco utilizando un algoritmo

de tres fases. Para reducir las falsas detecciones, primero separaron las zonas de tabaco de las zonas de no tabaco, lo que se llevó a cabo separando el suelo de las plantas verdes utilizando el valor RGB en la imagen. Sin embargo, en ocasiones, hay zonas que aparecen con la clasificación incorrecta, y para solucionar este problema hay que tener en cuenta situaciones reales que aparecen con frecuencia. De hecho, esta técnica se utilizará como primer paso para detectar plantas de agave. El valor de color también se utilizó para separar la planta del suelo y se utilizó la silueta de la planta obtenida de esta segmentación. Otro algoritmo de inspección basado en la segmentación se desarrolló en [4]. La segmentación se realizó para palmas y descompuso las imágenes aéreas en dos clases teniendo en cuenta que algunas imágenes pueden incluir nubes y vegetación desconocida. Esto refuerza la idea de que es útil tener una fase de validación. En nuestro caso, no esperamos tener imágenes con nubes porque proponemos un vuelo a baja altura. Sin embargo, hay que afrontar el problema del cruce de las hojas de agave.

Otro algoritmo de detección para palmeras fue propuesto por [44]. En este artículo, se combinaron dos redes neuronales para mejorar la detección. Una red neuronal se dedicó a detectar grandes detalles de palmeras, mientras que la otra red neuronal se dedicó a detectar detalles más gruesos. En nuestro caso, no estamos usando dos redes neuronales, sino dos algoritmos para detectar el centro de la planta de agave para mejorar la detección. El término cultivo profundo(crop deep) fue introducido por [42], que es una red neuronal capaz de detectar y diferenciar 31 tipos de frutas y verduras para aplicaciones agrícolas. Esto se utilizó para tener una estimación de cuántas imágenes se requieren para realizar una buena detección. En promedio, se requieren mil imágenes para entrenar un algoritmo de detección. En [17], se propuso un algoritmo que se ocupa de la detección de palmeras mediante aprendizaje profundo. Los autores estudiaron el caso de tener una ventana deslizante, que también se utilizará en nuestro caso, el problema de las detecciones múltiples estaba presente, es decir, la misma planta podía detectarse dos o tres veces. Para solucionar este problema, se detectaron los centros de las ventanas candidatas y se definió una distancia mínima entre los

centros. Esto aseguró tener una sola detección. En nuestro caso utilizamos la misma técnica ya que las plantas de agave pueden estar demasiado cerca unas de otras.

También se propuso un algoritmo de detección de palmeras en [38] y [39] mediante aprendizaje profundo. En [39] los autores analizaron ortomosaicos de alta resolución. Inicialmente definieron zonas para realizar la detección. Recomendaron que las zonas se superpongan un poco entre sí para evitar perder información importante. En el presente trabajo seguimos las principales métricas propuestas en [13] para profundizar en el campo de la agricultura de precisión. Otra área de investigación interesante es la detección de frutos en el interior del árbol, ya que pueden existir casos en los que dos objetos se superponen, y puede ocurrir que solo se tenga en cuenta uno. Situaciones similares ocurren en otras obras que se mencionan a continuación, por lo que es importante analizar cómo afrontar el problema de la superposición, pero con una planta diferente. La detección de frutas se discutió recientemente en [14] para el caso de detección de manzanas mediante aprendizaje profundo. En general, el conteo de frutas es una tarea complicada, como se puede ver en [32] para la detección de duraznos y [26] para contar mango en el arboles. Estos artículos recomiendan usar la primero segmentación de textura antes de aplicar la detección de fruta.

En los últimos años, el aprendizaje profundo ha demostrado ser una de las mejores opciones para detectar objetos. Dado que nos interesa la detección de plantas, es relevante revisar trabajos en el área de agricultura de precisión ocupando estos algoritmos, entre los más relevantes podemos encontrar los siguientes trabajos [6], [15], [42], [27], [44], [16], [17], [39], [13]. El trabajo de Kitano en [15] es interesante ya que las plantas de maíz se detectaron en diferentes etapas durante su crecimiento mediante el aprendizaje profundo. Este trabajo realizó una segmentación de las imágenes, separando el suelo de la zona verde. Dado que Kitano requería mediciones cuando el tamaño de la planta era pequeño, esto requirió prototipos que pudieran volar a altitudes bajas. La detección de plantas de tomate se informó en [27] mediante aprendizaje profundo. Curiosamente, las imágenes de entrenamiento contenían imágenes virtuales. Esto significa que una computadora generó aleatoriamente imágenes virtuales para abordar el problema de la oclusión y la detección de tomates que no estaban tan

maduros. YOLO es una herramienta que se ha utilizado últimamente, ofreciendo buenos resultados en conjunto con otros algoritmos, como se puede ver en [12,43]. El primero presentó una reconstrucción 3D de árboles para poder utilizar los datos en procesos de forestación. El último consideró las tareas de conteo mediante la detección de insectos y podría considerarse como un trabajo futuro. Usando hardware más específico, se adquieren y tratan imágenes multiespectrales en obras de [40], [10], [29], [41]. En [1], [37] se implementa un sensor lidar además de la visión artificial. Para la tarea de monitoreo, tenemos trabajos en [36], [1], [37], [15] y [27] que incluyen procesos de conteo de elementos. Las imágenes tomadas por los UAV se pueden usar para formar ortomosaicos u ortoimágenes y en obras de [9] y [28] se mencionan las propiedades y usos en la agricultura. En relación con las especies específicas estudiadas en este trabajo, hay una especie similar en forma gráfica. De hecho, la vista superior de la palmera de aceite es similar a la planta de agave.

Esta similitud es significativa para nuestro trabajo, es vista dentro de los trabajos de [44], [4], [17], [16] y [39]. En [3] se analizan diferentes técnicas y sensores para adquirir propiedades de las plantas y en [25] también se estudian las propiedades de la imagen con diferentes sensores.

1.2 Motivación

La motivación de este trabajo es la supervisión de espacios donde es necesario contabilizar el número de elementos en una escena, estos pueden variar. Para ello el primer paso es distinguir la escena que se propone, sea un camino o un surco y pueda la cámara desplazarse a través de él con ayuda de un sensor a través de la detección de los bordes del camino.

Una aplicación específica que se contempla para este proyecto es el monitoreo de plantíos de maguey, sin embargo, este tipo de proyectos pueden ser aplicados para otros tipos de plantas o también en eventos masivos como pueden ser mítines, eventos de entretenimiento, manifestaciones, etc.

1.3 Planteamiento del problema

Un problema común que tienen las personas que trabajan dentro de la agricultura es el robo de su cosecha dentro de sus plantíos, muchas veces debido a que el área es bastante grande que les es imposible vigilar su terreno en todo momento. Un caso en específico en la agricultura de magueyes para producción de tequila es que gente se roba los magueyes más maduros y en su lugar deja magueyes jóvenes por lo que en muchos de los casos no se sabe cuánto se han robado, ya que tampoco se lleva un control exacto de su producción.

Además de esto, está el hecho de que para el caso del maguey se tiene un tiempo de maduración de aproximadamente 10 años, por lo que el robo tiene un impacto mucho mayor ya que no solo se trata de dinero invertido, sino que también de tiempo, por lo que es necesario encontrar una manera de monitorear sus cultivos y contabilizar lo que se tiene en cada uno de forma rápida y eficiente.

1.4 Propuesta de solución al problema planteado

El monitoreo podría ayudar a los agricultores a localizar más fácilmente las plantas que están listas para la cosecha y evitar el robo. En este trabajo se propone una solución tecnológica para reducir el problema del robo de agave mediante una plataforma VANT de tipo multirrotor que realiza un monitoreo de un plantío mediante el seguimiento de los surcos. El vehículo se equipa con un gran número de sensores que le permiten conseguir un nivel de autonomía tanto para el monitoreo como para la navegación. Esto también es posible gracias a los algoritmos de visión artificial, inteligencia artificial y de control embebidos dentro de las computadoras con las que contaría la plataforma y que serán descritas a detalle a lo largo de este escrito. Este monitoreo que se propone llevaría un algoritmo de visión artificial que cuente las plantas de forma automática.

1.5 Justificación

El monitoreo de plantíos es una de las muchas aplicaciones de los vehículos aéreos no tripulados en la agricultura, su empleo se hace congruente ya que las personas que se dedican a esto tienen dinero y tiempo invertido en la tierra que trabajan y al tener bien monitoreados sus cultivos hace que tengan una mejor noción de sus cosechas. Para ello es necesario como primer paso hacer el conteo de las plantas de forma individual y para el caso del maguey también sería útil saber el nivel de maduración de cada uno ya que su tiempo de maduración se mide por años, todo esto para saber cuáles plantas ya son cosechables y así poder evitar que sean robados o suplantados por otros más jóvenes. Desde una toma aérea del plantío o parte de él se podría hacer una estimación, ya que también se pueden empatar varias tomas y hacer una imagen completa del área que se requiera.

El conteo se puede realizar fuera de línea o a bordo.

1.6 Objetivo general

Desarrollar un vehículo aéreo no tripulado que haga tareas de monitoreo de plantíos, realizando conteo de plantas de forma automática por medio de algoritmos de aprendizaje de máquinas y visión artificial.

1.6.1 Objetivos particulares

- Desarrollar un método para la detección y seguimiento de caminos.
- Hacer el programa del método de detección y conteo.
- Evaluar la exactitud del conteo por detección.
- Hacer una comparación con respecto a otro algoritmo de visión artificial.

1.7 Organización del escrito

En el Capítulo 2 se verá lo correspondiente a la parte de monitoreo que es a grandes rasgos la parte fundamental de la solución que se propone al problema de robo de agave. Se empieza con el reconocimiento de objetos por medio de visión artificial haciendo una breve reseña de los algoritmos y métodos más aplicados en esta área. Se hará especial énfasis en lo referente al aprendizaje profundo debido a que este enfoque es en el que se basa el método de solución propuesto. Se verán las principales métricas para medir el desempeño de un algoritmo de detección de objetos. A continuación, se explica a fondo el funcionamiento del método de detección propuesto y sus correspondientes submódulos. Por último, en este capítulo se verá el proceso y clasificación de la base de datos creada para este trabajo.

El Capítulo 3 se compone por la parte de navegación por seguimiento de un camino, se explica cómo es que se propone la detección de un camino y como se aplicaría al dron para poder detectar los surcos y tomarlos como un camino a seguir.

En el Capítulo 4 se describe la plataforma experimental construida para el proyecto. Todo el proceso de diseño y construcción, la elección de la aviónica y los sensores necesarios para cumplir con la tarea. Al final del capítulo se muestra el modelo matemático de un cuadrirotor.

En el Capítulo 5 se presentan los resultados experimentales del método de detección y conteo de agaves aplicado a un fragmento de un ortomosaico de plantaciones de agave para poder apreciar el nivel de precisión del método. Se presentan los resultados de todas las fases del método para culminar con el conteo final y establecer a partir de las métricas cual es el porcentaje de precisión que presenta el método.

En el último Capítulo se presentan las conclusiones del trabajo, para poner un poco mejor en contexto la viabilidad del método de detección y conteo se compara con otro algoritmo de detección conocido. Para terminar, se proponen algunas tareas como trabajo futuro y que

podrían mejorarse si se le da continuidad a este trabajo, así como los artículos científicos derivados de este trabajo de tesis.

CAPÍTULO 2

MONITOREO

La parte de monitoreo es crucial para la aplicación que se es pretendida en este trabajo. Es por eso por lo que gran parte del aporte está en cómo se lleva a cabo este procedimiento. Como ya se mencionó anteriormente su base está en el uso de visión por computadora, a su vez ésta se apoya del uso de técnicas de inteligencia artificial para el reconocimiento de objetos dentro de una imagen, esta rama de la inteligencia artificial es conocida como aprendizaje profundo. Haciendo uso de este enfoque logramos detectar nuestro objeto de interés en la imagen que para nuestro caso son plantas de agave y hacer un conteo del número de elementos contenidos en ella. A continuación, se aunará en los tópicos que estos involucran.

2.1 Reconocimiento de objetos

El reconocimiento de objetos se puede lograr de varias formas y la opción correcta va a depender de la complejidad del objeto a detectar y del entorno de donde se busca diferenciarlo, pero primero se debe fijar hasta qué punto se desea llegar. Ya que no es lo mismo detectar por ejemplo formas en una imagen binaria a detectar a una especie de animal dentro de una foto a color con varios tipos de animales.

2.1.1 Aprendizaje de máquinas

El aprendizaje de máquinas es solo una rama de las diferentes ramas que tiene la inteligencia artificial y se enfoca en el aprendizaje de patrones o datos. Estos datos pueden ser acerca de cualquier cosa, ya que con el aprendizaje de máquinas lo único que hace es “aprender” el comportamiento de estos datos, pueden ser imágenes, datos en texto plano, bases de datos de números, etc.

Clasificadores en cascada basados en características de Haar

La detección de objetos por medio de este método fue propuesta por Paul Viola y Michel Jones en 2001 en el artículo “Rapid Object Detection using a Boosted Cascade of Simple Features”. Este es un enfoque de aprendizaje de máquinas donde funciones en cascada son entrenadas a partir de imágenes positivas y negativas, a partir de este entrenamiento pueden ser detectadas en otras imágenes.

Para esto primero hay que aclarar que son estos dos tipos de imágenes:

- **Imágenes positivas:** son imágenes donde se encuentra el objeto que se desea detectar. Para estas imágenes se recomienda tener imágenes de varios ángulos del objeto o bien de las posibles variedades que este pudiera tener.
- **Imágenes negativas:** son imágenes donde no debe encontrarse el objeto a detectar, se recomienda utilizar imágenes del entorno donde se podría localizar el objeto, pero no esté en él, además de imágenes de objetos con los que pudiera confundirse.

La detección de objetos involucra dos etapas que son la de entrenamiento y la de detección como tal. Primero se lleva a cabo la etapa de entrenamiento que es donde se hace uso de las imágenes positivas y negativas. Básicamente este proceso extrae características de estas imágenes al partir del uso de máscaras o kernels conocidos como descriptores de Haar como las que se muestran en la siguiente figura.

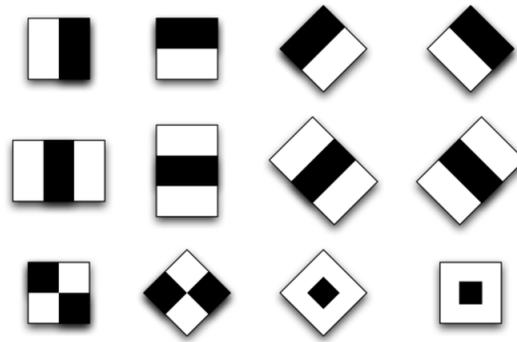


Fig. 2.1: Patrones de Haar

Estos descriptores son utilizados para extraer las características de las imágenes, se prueban diferentes kernel con diferentes tamaños para sacar la mayor cantidad de características de las imágenes. Pero para el cálculo de todas estas características se necesitarían muchas operaciones y por ende el costo computacional se vuelve más alto. Es por esto que se emplea algo que se conoce como imagen integral o tabla de área sumada (Summed-area table) que es una tabla de las mismas dimensiones de la imagen con los valores de la integral de los valores de los pixeles de esa coordenada en la imagen. Esto se hace para hacer más ligero el computo de los valores de los pixeles de un área determinada ya que estos datos los ocupamos a la hora que buscamos las características de Haar.

1	2	2	4	1	0	0	0	0	0	0
3	4	1	5	2	0	1	3	5	9	10
2	3	3	2	4	0	4	10	13	22	25
4	1	5	4	6	0	6	15	21	32	39
6	3	2	1	3	0	10	20	31	46	59
					0	16	29	42	58	74

Imagen de entrada Imagen integral

Fig. 2.2: Ejemplo de una imagen integral

Con este formato reducimos el cálculo de los valores a una suma de cuatro números sin importar que tan grande sea el área ya que solo necesitaríamos los valores de las esquinas. La

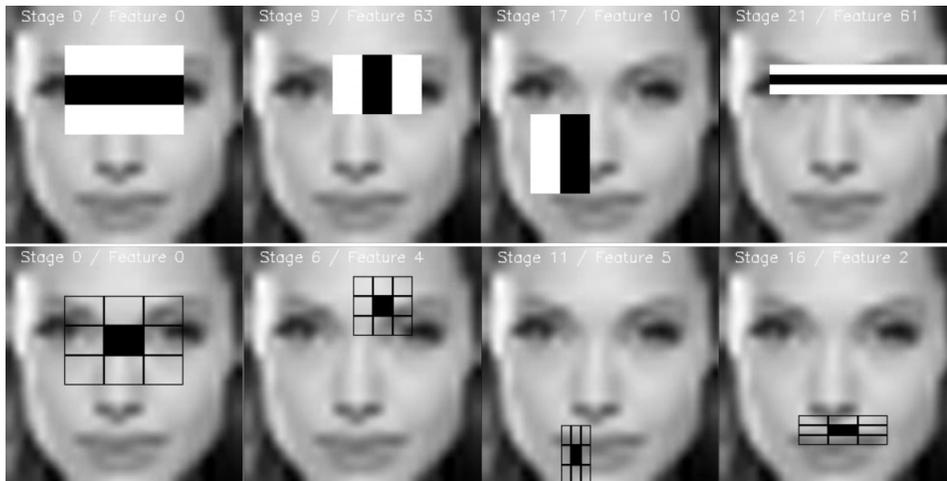


Fig. 2.3: Patrones de Haar aplicados a una imagen

utilidad de esto se ve muy marcada al momento de que se observa que los kernel que buscan características en la imagen pueden tener dimensiones muy variadas.

Lo siguiente es elegir que kernel es el mejor para la detección de cada área y para ello se utiliza Adaboost.

Aprendizaje profundo

El Aprendizaje profundo es una rama del aprendizaje de máquinas que a su vez forma parte de la inteligencia artificial. Por su parte el aprendizaje de máquinas ofrece soluciones de aprendizaje basándose en el análisis de datos y que partir de estos y de las salidas asociadas a ellos crean reglas para así poder reconocer patrones ante datos nunca vistos. Para comenzar a comprender lo que se puede hacer con el enfoque de aprendizaje profundo se tienen ciertos requerimientos.

- Datos de entrada.
- Ejemplos de la salida esperada.
- Una forma de medir si el algoritmo está haciendo un buen trabajo.

El aprendizaje de máquinas busca que los datos sean significativamente útiles mediante la búsqueda de diferentes representaciones ya que representar los datos de diferente forma puede mejorar la forma de visualizar la clasificación. El learning o aprendizaje en el contexto del aprendizaje de máquinas describe el proceso de búsqueda automática de mejores representaciones.

Una vez dicho esto, se tiene el contexto suficiente para decir que en específico el aprendizaje profundo se enfoca en el aprendizaje de capas sucesivas de representaciones cada vez más significativas. Y que es precisamente lo que le da el nombre de Deep queriendo decir que cuenta con una profundidad en el número de capas.

Esta representación en modo de capas suele estructurarse de forma que funcionen como lo que se conoce como redes neuronales. Uno de los ejemplos más usados por verse de forma muy didáctica es el reconocimiento de texto y en específico de números.

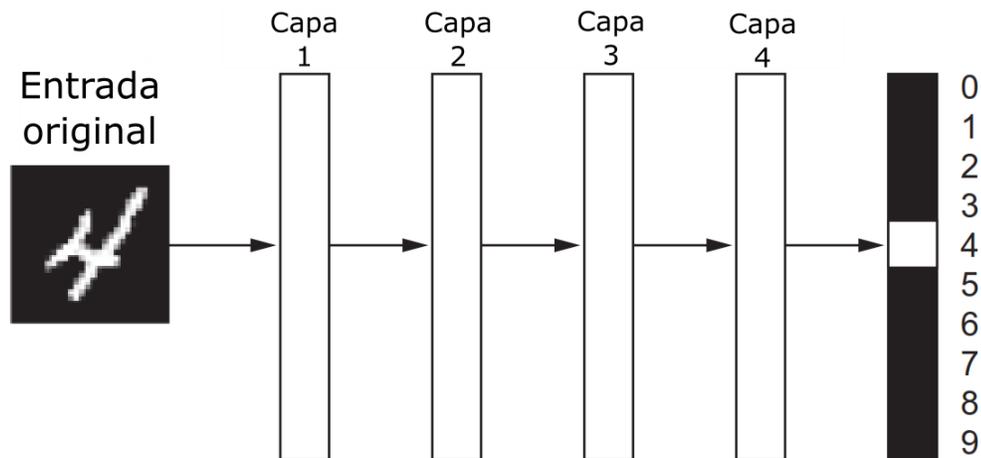


Fig. 2.4: Diagrama de reconocimiento de patrones(Imagen: [2])

Aquí se puede ver un ejemplo de una imagen de un número en el que la red reconoce de que número se trata. Y la forma en que analiza la imagen es la siguiente.

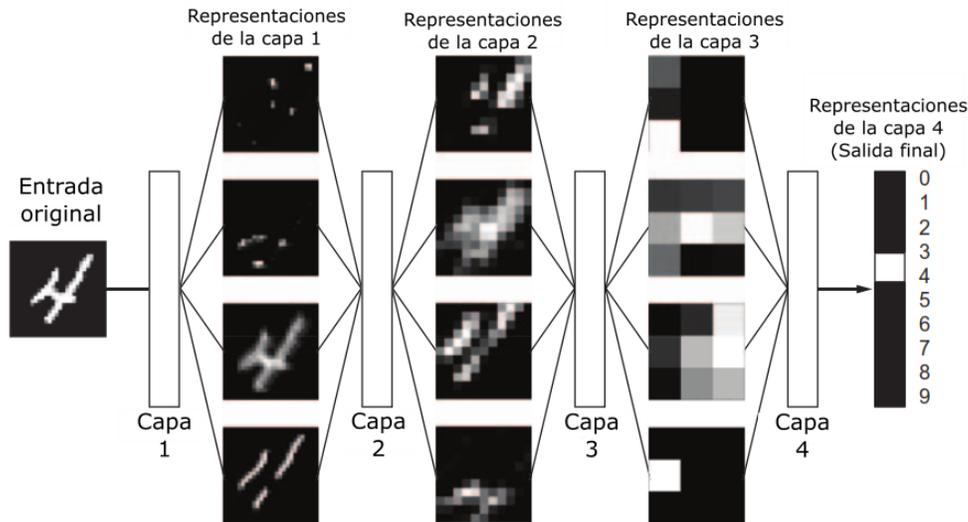


Fig. 2.5: Representación gráfica de reconocimiento de patrones(Imagen: [2])

Como se puede ver cada capa detecta diferentes patrones (o representaciones) de la imagen. El funcionamiento del aprendizaje de máquinas básicamente se trata de mapear entradas (inputs) con objetivos(targets) por medio de la observación de varios ejemplos de estos. Las redes neuronales profundas hacen esta relación input-target a través de una secuencia de transformaciones de los datos y estas transformaciones son aprendidas al exponer a la red a varios ejemplos.

La especificación de lo que cada capa hace con los datos de entrada es almacenada en los pesos de la capa, lo que esencia son un montón de números. Se puede decir que las implementadas por cada capa es parametrizada por los pesos. Es por eso por lo que los pesos son comúnmente llamados los parámetros de la capa. En este contexto el aprendizaje o learning significa encontrar un grupo de valores para los pesos de todas las capas de la red de forma que la red mapee correctamente las entradas con sus targets asociados. Sin embargo, una red neuronal puede tener miles o millones de parámetros, por lo que encontrar lo mejores valores resulta una tarea difícil.

Para controlar la salida de una red neuronal es necesario medir que tan lejos está la sal-

ida de lo que se espera. Esto se logra con la función de pérdida (loss function) de la red o también conocida como función objetiva (objective function). La función de pérdida toma las predicciones de la red y los targets correctos y mide la distancia o error, captando que tan bien la red lo ha hecho para ese ejemplo.

Una parte importante dentro del aprendizaje profundo es usar este error como retroalimentación y ajustar el valor de los pesos de los parámetros. Intentando minimizar ese error para ese ejemplo. Este ajuste es el trabajo de optimizar, lo cual implementa algo conocido como algoritmo de Backpropagation que es el algoritmo central del aprendizaje profundo.

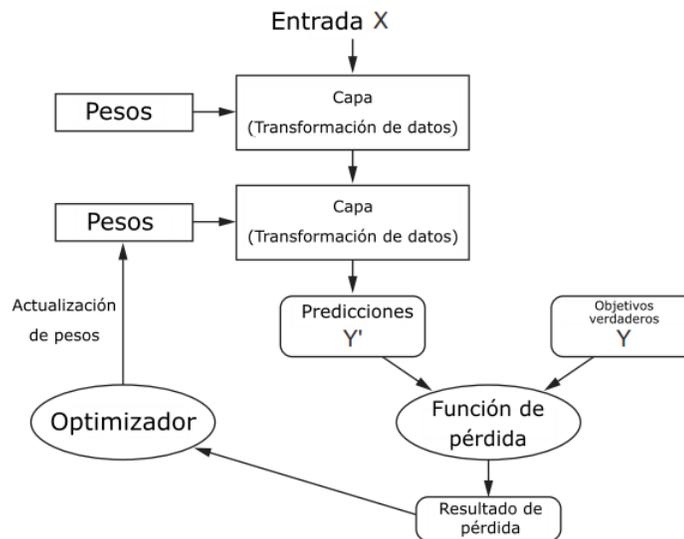


Fig. 2.6: Diagrama de funcionamiento del proceso de optimización. (Imagen: [2])

Inicialmente los pesos son asignados de forma aleatoria, y como es de esperarse en ese momento la salida está lejos de lo esperado y la tasa de pérdida es alta. Pero los pesos se irán ajustando conforme se vayan procesando más ejemplos. Esto corresponde al ciclo de entrenamiento, el cual se repite las veces que sea suficientes hasta que la función de pérdida sea minimizada.

Entrenamiento de la red neuronal convolucional

El procedimiento para entrenar una red de tipo convolucional es un poco diferente, para esto se hace uso de las librerías de Keras y se ocupa el lenguaje de programación Python, además de que la base de datos se clasifica y secciona dependiendo para que se ocupar. Para el proceso de entrenamiento de este tipo de redes se suele separar la base de datos completa en tres partes:

- 70% Imágenes para entrenamiento
- 15% Imágenes para validación
- 15% Imágenes para prueba

Haciendo esta separación se pueden obtener valores de exactitud de entrenamiento y validación al momento de estar entrenando la red. Se entreno por 100 épocas(iteraciones). También se tiene que definir una estructura para el entrenamiento de la red, para nuestro caso se empleó la siguiente arquitectura:

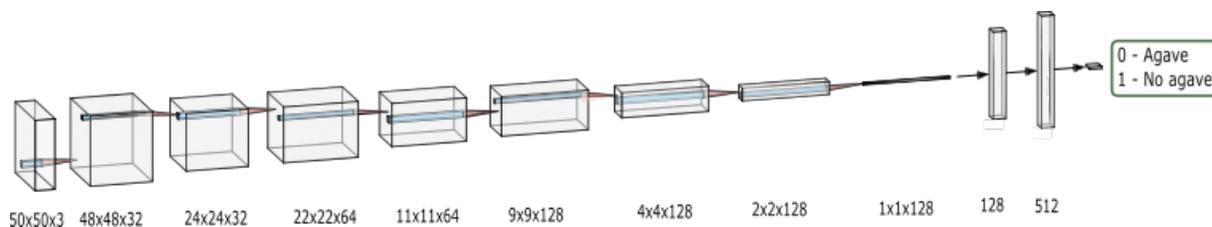


Fig. 2.7: Estructura de la red aplicada para reconocimiento de agaves

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 48, 48, 32)	896
max_pooling2d_1 (MaxPooling2)	(None, 24, 24, 32)	0
conv2d_2 (Conv2D)	(None, 22, 22, 64)	18496
max_pooling2d_2 (MaxPooling2)	(None, 11, 11, 64)	0
conv2d_3 (Conv2D)	(None, 9, 9, 128)	73856
max_pooling2d_3 (MaxPooling2)	(None, 4, 4, 128)	0
conv2d_4 (Conv2D)	(None, 2, 2, 128)	147584
max_pooling2d_4 (MaxPooling2)	(None, 1, 1, 128)	0
flatten_1 (Flatten)	(None, 128)	0
dense_1 (Dense)	(None, 512)	66048
dense_2 (Dense)	(None, 1)	513
Total params: 307,393		
Trainable params: 307,393		
Non-trainable params: 0		

Fig. 2.8: Composición de número de parametros y entradas/salidas de cada capa

Lo que se muestra en la imagen (Fig. 2.8) es una estructura por capas en la que la primera capa o capa de entrada se meten las imágenes que en este caso son imágenes de 50x50 y x3 al ser de tres canales de acuerdo con los mapas de color.

Posteriormente se aplican capas de convolución y pooling cuatro veces seguidas esta combinación es para no saturar tanto el proceso de entrenamiento y las otras tres capas son para adecuar la salida de la red y al final tener la clasificación como maguey o no maguey.

Para nuestro entrenamiento se obtuvieron los siguientes resultados:

Monitoreo a partir del seguimiento de un camino con un cuadri-rotor

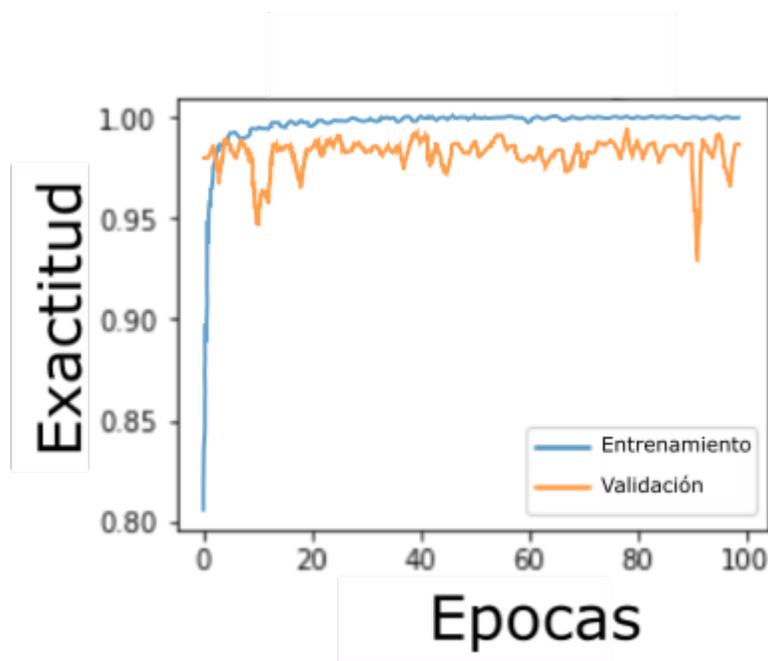


Fig. 2.9: Valores de exactitud a lo largo de todo el entrenamiento

El proceso de entrenamiento de la red se realizó correctamente con entrenamiento y validación buenas y congruentes. Dentro de las redes convolucionales se busca que estos dos valores sean cercanos.

2.2 Métricas de evaluación de desempeño de detección de objetos

Para medir el rendimiento de detección y comparar varios métodos, es necesario calcular las siguientes métricas:

$$\text{Precisión}(PPV) = \frac{TP}{TP + FP}$$

$$\text{Sensibilidad}(\text{Recall} \text{ ó } TPR) = \frac{TP}{TP + FN}$$

$$F_1 = \frac{2}{\frac{1}{pres} + \frac{1}{recall}} = \frac{TP}{TP + \frac{FN+FP}{2}}$$

donde TP es el número de detecciones positivas verdaderas, es decir, detecciones donde se detectó una planta y en efecto corresponde a una planta real; FP es el número de falsos positivos, es decir, detecciones donde se detectó una planta, pero no corresponde a una planta y FN corresponde al número de plantas reales que se omitieron como detecciones. *Precisión*, *recall* y F_1 son las métricas más comunes para medir la efectividad de los algoritmos de detección visual. La *precisión* o *PPV* (valor predicho positivo) mide de todas las detecciones el porcentaje de los que son correctos. La *recall* o *TPR* (tasa positiva verdadera) mide la sensibilidad de toda la clase positiva, incluidas la verdadera y la falsa, la *recall* muestra qué porcentaje es correcto. F_1 es una métrica comúnmente utilizada para medir el rendimiento del clasificador de manera general, ya que combina *precisión* y *recall*, y solo se obtendrá una puntuación alta si tanto la *recall* como *precisión* son altas. Estas métricas son las más comunes en el campo del aprendizaje de máquinas.

2.3 Método de detección propuesto

El método de detección que se propone para este trabajo consta de tres etapas principales que funcionan de manera secuencial y que la salida de una etapa es la entrada de la siguiente para que al final se tenga un conteo de los agaves detectados dentro una imagen. Cada etapa a su vez cuenta con subprocesos que ayudan a que la etapa se pueda llevar a cabo de forma correcta. Un esquema general de este método se puede observar en la imagen siguiente.

A continuación, se verá a más detalle cómo funciona cada etapa del método propuesto, cabe mencionar que este método está enfocado en la detección de agaves únicamente ya que se hace uso de las propiedades gráficas y forma de esta planta vista desde una toma aérea para lograr una detección precisa.

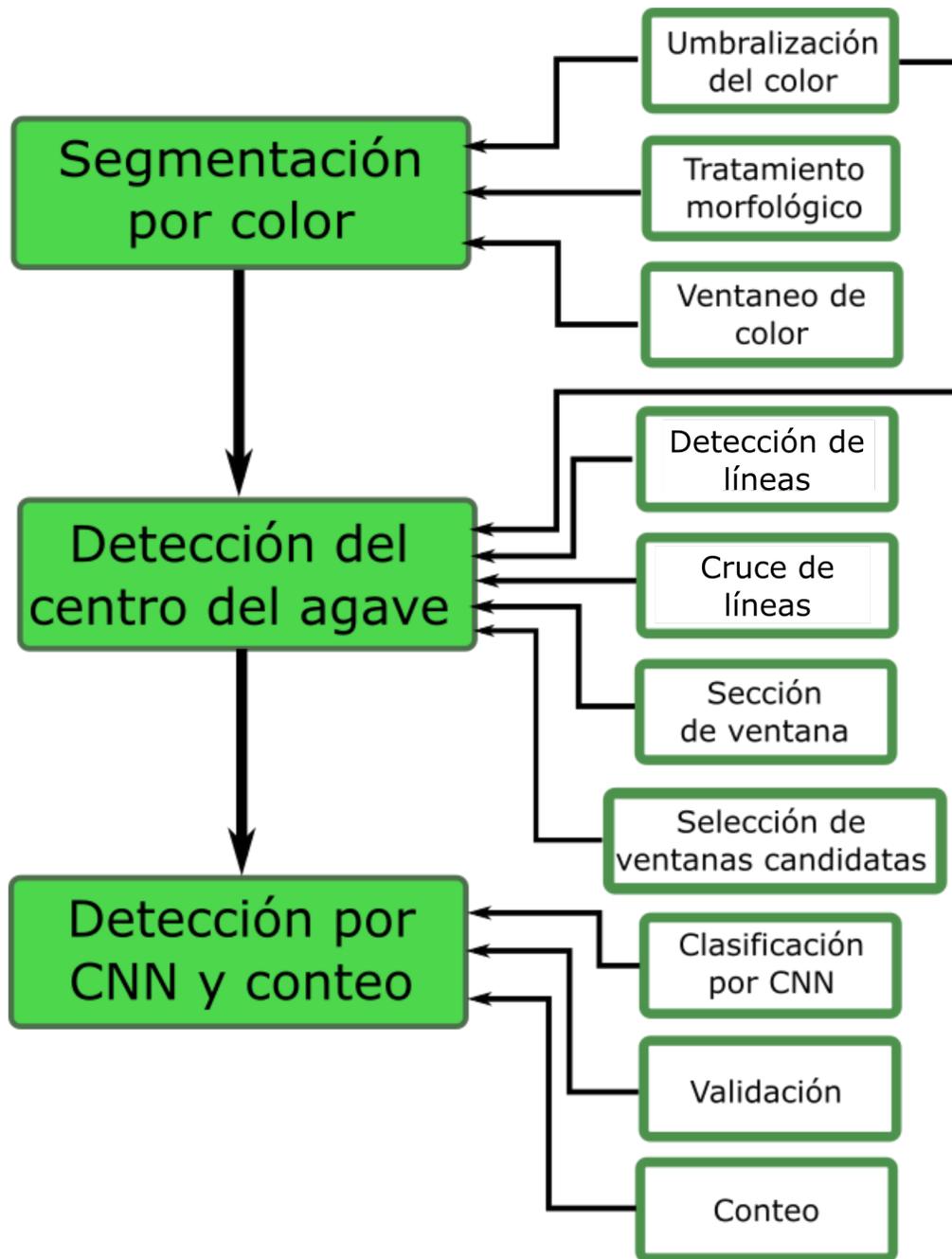


Fig. 2.10: Diagrama general del método propuesto

Canal de color	Mín	Máx
R (Rojo)	65	85
G (Verde)	82	255
B (Azul)	90	255

Tabla 2.1: Umbral de valores RGB correspondientes al color del agave.

2.3.1 Segmentación por color

La primera fase de este método se encarga de segmentar dentro la imagen las zonas donde se podría localizar plantas de agave mediante la búsqueda de su valor de colores en formato RGB. Para ello lo primero que se hace es realizar un proceso de umbralización para poder abarcar las diferentes tonalidades de color de las pencas que presenta esta planta en específico. Por medio de un proceso de análisis de los rangos de valores se encontraron que los valores en los canales RGB son los siguientes (Tabla 2.1):

Al buscar en la imagen los pixeles que cumplen con estos valores se obtendrán los pixeles que corresponden al color de las pencas de los agaves, el siguiente paso es llevarlo a un formato binario para obtener una mejor visualización, por lo que se espera tener como imagen de salida una imagen en blanco y negro, donde veamos de color blanco las pencas de los agaves y de color negro todo lo demás. Además de la visualización esto nos ayuda para el procesamiento ya que ahora en lugar de trabajar con una imagen de tres canales se trabaja con una imagen de un solo canal.

No se puede abarcar los colores de la planta completa debido a que hay partes de la planta como el centro que parecen ser más oscuras y la combinación de valores se asemeja a colores como el del suelo es por esto por lo que solo se optó por resaltar las partes de las pencas. Dicho lo anterior, lo que prosigue es generalizar más las zonas resaltadas y tener una noción más uniforme del área de lo agaves, para lograr esto se hace un tratamiento morfológico a la imagen binaria por medio del dilatado de la imagen. Con este proceso se resalta mejor las

zonas donde hay agaves completos y ya no solo la pencas. A partir de esto se puede localizar donde hay agaves, pero no es posible diferenciar uno de otro y esto es una dificultad para realizar un conteo.

Las siguientes fases del método se enfocan en resolver el problema de diferenciación de las plantas de agave, pero se requiere definir que el tamaño en formato de pixeles que podría medir un agave y este es de 50 x 50 pixeles. Es por esto por lo que se va a tomar como base esta medida para hacer la búsqueda de elementos individuales, para esto se hace uso del concepto de ventana deslizante que es una ventana de una parte de la imagen que se va recorriendo a lo largo y ancho de toda la imagen en busca de propiedades de interés, esta ventana se traslapa con la ventana anterior para una mejor búsqueda y si la ventana cumple con la propiedad establecida es guardada.

Para esta primera fase se aplica la ventana deslizante para encontrar las ventanas en las cuales hay plantas de agave por medio de la imagen binaria. Con esto simplificamos el proceso para las fases siguientes ya que no se hace búsqueda donde no existen plantas.

2.3.2 Detección del centro del agave

Al haber definido al tamaño en pixeles de la planta el problema se reduce a diferenciar plantas de forma individual por lo que en esta segunda fase se encarga de aprovechar el patrón visual que presenta la vista superior de los agaves para poder separarlos. Esta vista provee una forma semejante a una estrella de múltiples picos por lo que se busca encontrar las características graficas particulares de esta figura dentro de la imagen, tal como se hizo con la parte de los valores RGB en el paso anterior.

Estas señas particulares que se buscan son las líneas que forman las pencas del agave, y como estas tiende a cruzarse en el centro de la planta. Por lo que se hace uso de la detección de líneas de Hough para cada una de las ventanas del paso anterior pero aplicadas sobre la

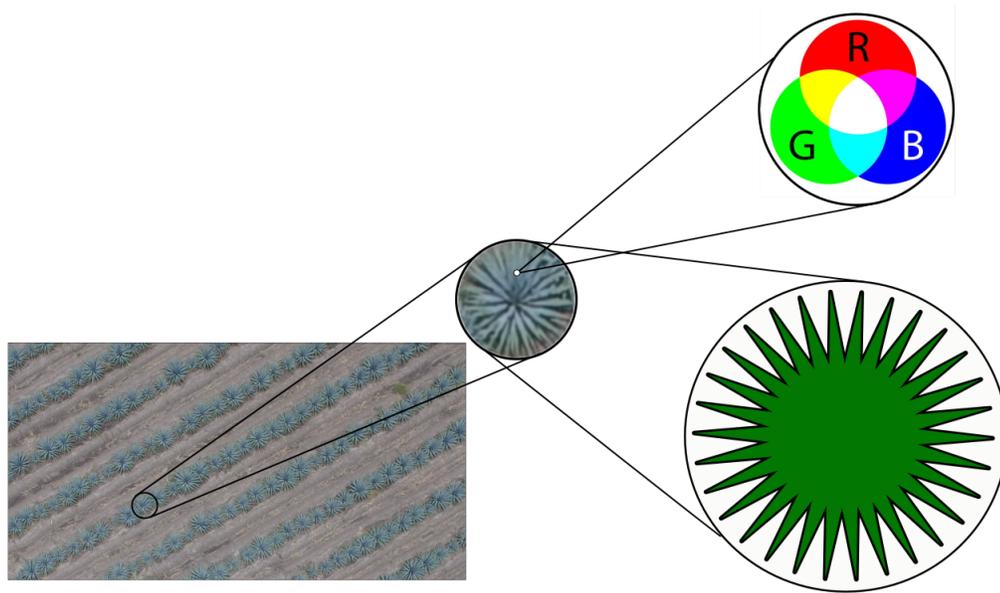


Fig. 2.11: Características principales de las imagenes de agaves

imagen binaria sin dilatar. Esto es porque en esta imagen se aprecian de mejor manera las líneas que forman las pencas, ya que en la dilatada se pierden completamente debido a que se resalta la zona completa de forma homogénea.

El siguiente paso encontrar el punto donde estas líneas en caso de que se detecten las suficientes, en que punto de la ventana se interceptan que no más que encontrar lo que se conoce como el punto de fuga. Esto para tomar en cuenta las ventanas que contienen una sola planta de forma completa y descartar las ventanas que se encuentran entre una planta y otra. Para lograr esto se va a separar la ventana en nueve celdas iguales y se va a determinar en qué celda de cada ventana se encuentra su punto de fuga, si el punto de fuga se encuentra en la celda central será considerada como una celda candidata para una planta de agave. Al obtener las ventanas candidatas nos enfrentamos a que en varias ventanas pueden caber la misma planta obteniendo detecciones múltiples de la misma planta, esto se debe al tamaño de la ventana que se tomó en cuenta para las detecciones y la justificación del porque se tomó la medida de 50 x 50 pixeles es que todas las plantas son de diferente tamaño y para

no omitir ninguna se decidió tomar la mayor medida posible que pudiera llegar a tener.

El último paso de esta etapa consta de aplicar algo conocido como mapa de calor que se trata de crear un mapa del mismo tamaño de la imagen original y vamos a ponderar las zonas de las ventanas de modo se vayan resaltando las zonas que más se traslapen para ir unificando las múltiples detecciones de la misma planta y poder considera una sola por cada una. Este mapa se crea con una imagen en blanco y negro, y los valores se van sumando de forma que los puntos más blancos corresponden a la zona del centro de la planta. A partir de este punto se creó una sola ventana con centro en el centro del punto resaltado. Se repite esto para todas las ventanas y se obtienen nuevas ventanas candidatas que son la salida de esta etapa y la entrada de la etapa siguiente.

2.3.3 Detección por CNN y conteo

La última etapa del método que se propone en este trabajo de tesis consta de la aplicación de una red neuronal para poder detectar si las ventanas que resultaron candidatas de la etapa anterior corresponden realmente a una planta de agave, para ello se realizó previamente un proceso de entrenamiento de un clasificador biclase “agave” y “no agave” utilizando aprendizaje profundo.

Entonces, partiendo de que se tienen las ventanas de la etapa anterior se mete la imagen de la ventana correspondiente al clasificador y lo que se obtiene es una probabilidad de que pertenezca a la clase denominada “agave”. Se define que si la probabilidad de que pertenezca a esta clase es mayor a 0.8 se es considerada como tal y la ventana pasa a convertirse formalmente como una ventana de agave. Al realizar este mismo procedimiento a través de todas las ventanas candidatas se obtiene el número estimado de plantas que corresponden al número de ventanas.

Esta parte se debe llevar a cabo debido a que la fase de detección de centros de maguey es

susceptible a sufrir falsas detecciones debido a que se basa en la detección de cruce de líneas y existen cruces de líneas entre las pencas de las plantas cuando estas están demasiado grandes y largas. Por lo que esta fase corrobora que en realidad sea una sola planta por ventana.

2.4 Base de datos de agave

Para conseguir realizar el entrenamiento de la red neuronal convolucional es necesario contar con datos de lo que se pretende detectar, para el caso de este trabajo la planta de agave es lo que se busca detectar dentro de una imagen donde además muy probablemente habrá más de una.

Entonces, se necesitan datos de estas plantas, debido a que la detección será de forma visual es necesario contar con una gran cantidad de imágenes de estas plantas. Otro aspecto importante es que al ser una detección aérea el perfil superior es el que va a ser visto por la cámara, por lo tanto, se necesitaran una mayor cantidad de imágenes desde una perspectiva superior.

Al buscar en las principales páginas de bases de datos de imágenes para aprendizaje profundo se notó que no existe una base de datos de estas plantas por lo que se concluyó que se debía construir desde cero. Es por esta razón que se decidió acercarse a la empresa Drones México, ellos ya cuentan con imágenes aéreas en alta resolución de plantíos de agave de la zona de Jalisco y otras partes donde se cultiva esta planta.

Nos fueron proporcionadas también las condiciones en las que fueron tomadas estas imágenes y en general del ecosistema en el que crece esta planta. Los suelos de esta zona suelen ser arcillosos, ricos en hierro y tienen un pH que varía de 6,0 a 8,5. La temperatura oscila entre los 15 C y los 32 C durante el día. La humedad media es del 60%. Por la noche, la temperatura puede descender hasta los 10 C. Un dato importante es que esta planta se ve muy afectada por las bajas temperaturas (menos de 0 C).

Estos ortomosaicos corresponden a tres hectáreas (7.5 acres) de área de plantación de agave cerca de la ciudad de Guadalajara, México ($20^{\circ}26'30.0026''N102^{\circ}51'20.3962''W$). Las imágenes fueron tomadas desde una altitud de 600 m sobre el nivel del mar y 50 m con respecto al suelo. Se utilizó un DJI Phantom 3 para tomar las imágenes con una cámara FC300S. Las imágenes fueron tomadas con una cámara que tiene una distancia focal f de 4 mm, por lo que la escala f / H es igual a 1: 12,500. Las imágenes fueron tomadas el 15 de mayo de 2018 entre las 13:38 y las 13:55 hora local. La temperatura era de $27^{\circ}C$ y el cielo estaba 65% nublado. El mediodía solar ocurrió en ese período de tiempo a las 12:48, y la velocidad promedio del viento fue de 9.2 km / h (4.9676 kt) en la dirección oeste-suroeste (WSW), entre 270 a 0.

Nos compartieron algunas de sus mejores imágenes y partir de ellas se empezó a trabajar en el armado de la base de datos. Se decidió que utilizaríamos algunas de esas imágenes para hacer la base de datos y guardaríamos otras para probar el método de detección. Para el entrenamiento de la red se requieren imágenes individuales del objeto que se quiere detectar por lo que el primer paso fue recortar imágenes individuales de las imágenes de los plantíos que se nos habían proporcionado. Para hacer que la red fuera mas fuerte se eligieron agaves de todos lo tamaños y con diferentes ángulos de iluminación. A partir de estas imágenes se crearon varias versiones de esta base de datos. Esto debido a que se fue probando la eficiencia del método de detección hasta el punto donde se observó un buen porcentaje de detección con cierto número de imágenes y llegando a la versión 3 que se consideró la versión final (Tabla 2.2).

Ahora para la estructura de la base de datos se organiza de una forma que usualmente se suele usar en aprendizaje profundo del 100% de las imágenes contando positivas y negativas que se considera como la base de datos ahora se reorganiza en un nuevo formato, donde, el 70% se van a destinar para hacer el entrenamiento, 15% para realizar el proceso de validación y 15% para el proceso de prueba. A su vez esta separación contempla que estas carpetas

Versión	Imágenes Positivas	Imágenes Negativas
1	157	198
2	782	500
3	1000	1000

Tabla 2.2: Cantidad de imágenes en cada versión de la base de datos

tengan sus propias carpetas de imágenes positivas y negativas. Las imágenes creadas para esta base de datos miden 50 x 50 pixeles cada una, esto es importante para el proceso de entrenamiento.

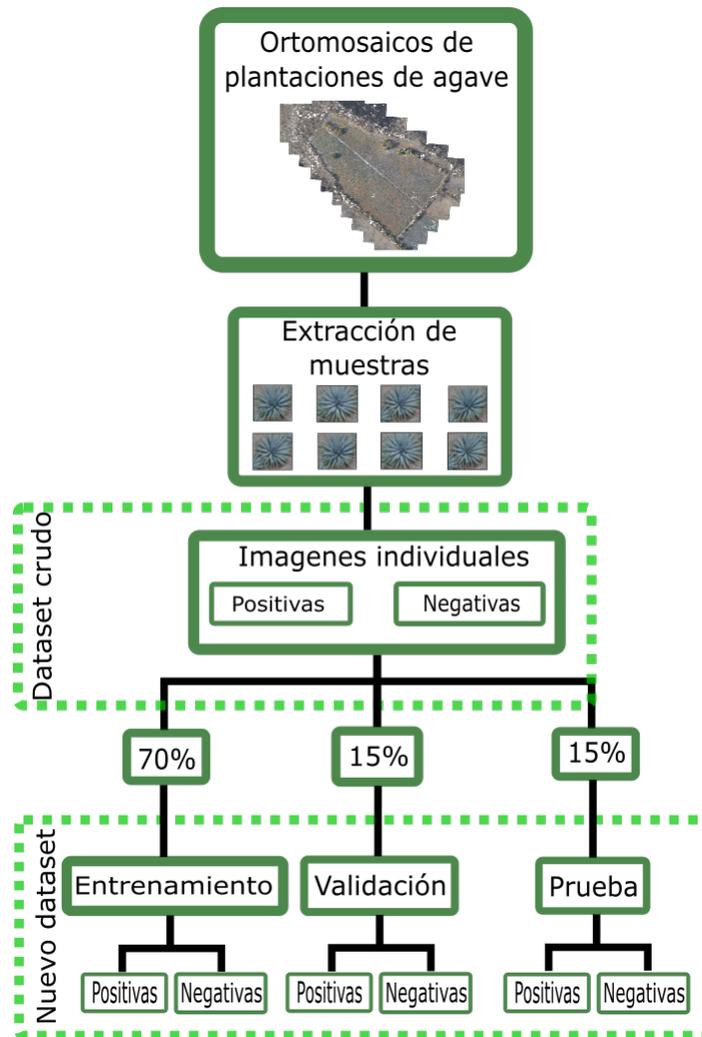


Fig. 2.12: Composición de la base de datos cruda y de la nueva para el entrenamiento

CAPÍTULO 3

NAVEGACIÓN A TRAVÉS DEL SEGUIMIENTO DE UN CAMINO

Para poder realizar la tarea de monitoreo se tiene que contemplar que el área a inspeccionar es grande y no basta con elevar la altura del dron ya que se requeriría subirlo a una altura en la que se vería comprometida la eficiencia del algoritmo de detección y conteo. Es por esta razón que se plantea realizar un recorrido a lo largo del terreno a inspeccionar a través del seguimiento de los surcos.

Para conseguir este seguimiento se implementa el uso de un sensor laser rotatorio el cual gracias a que está orientado con un ángulo hacia abajo permite detectar el surco y genera un camino seguro por el cual puede seguir el dron. El hecho de no hacerlo por medio de visión artificial es que por ahora la visión artificial es capaz de distinguir solamente agaves del entorno; sin embargo, no distingue otra cosa que pudiera ser un obstáculo para seguir adelante por lo que presenta un riesgo para el vehículo o para alguna persona que pudiera estar cerca y no notar su presencia.

3.1 Detección de los bordes del camino

El proceso para seguir los surcos empieza por la detección de los mismo por lo que primera tarea es detectar donde se encuentran para así poder centrar el vehículo con respecto a él y poder avanzar a través de él. Para la detección del surco no se hace uso de un algoritmo de visión artificial debido a que el principal objetivo de su uso es para la detección y conteo de planta, y para lograr realizar estas tareas la cámara necesita estar orientada hacia abajo. La intención de detección del surco es que sea detectado por delante, es decir, para detectar anticipadamente donde termina y poder seguir con el siguiente es por ello por lo que se hace uso del sensor laser rotatorio. Este nos entrega una trama de datos en grupos de dos correspondientes a un ángulo y a la lectura de distancia en ese ángulo, todo esto en un plano lineal; es decir que se tiene una lectura de 360° de lo que rodea al sensor como se puede ver Fig. 3.1.

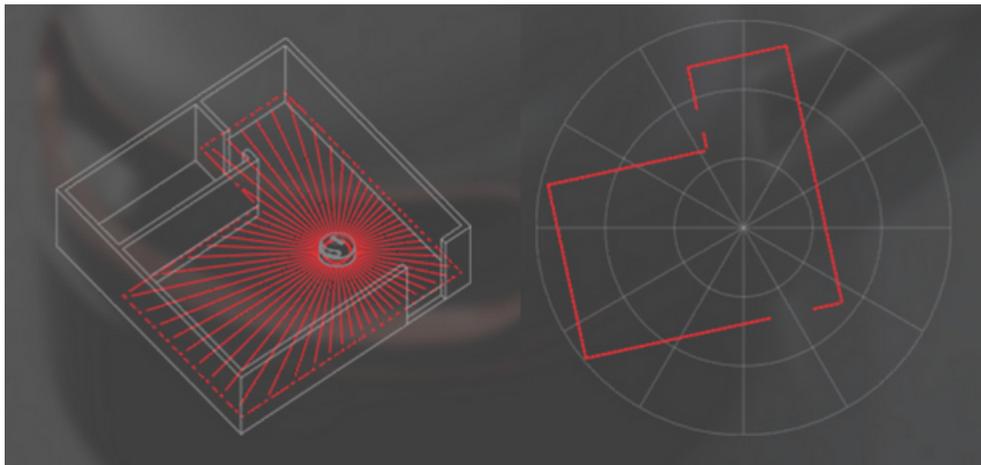


Fig. 3.1: Funcionamiento del sensor RPLiDAR

Sin embargo, como ya se dijo, esta lectura es en un plano, es decir que no tiene apertura por lo que solo se puede generar un mapa en dos dimensiones a partir de la nube de puntos que se puede generar a partir de la trama de mediciones. Por esta razón para la detección de los surcos no es de utilidad manejarlo con la orientación convencional que es de modo completamente paralelo al suelo ya que los surco se encontrarían por debajo del vehículo

en el momento del vuelo. Al cambiar el ángulo de cabeceo del sensor se puede obtener información más útil como se puede observar en la Fig. 3.2.

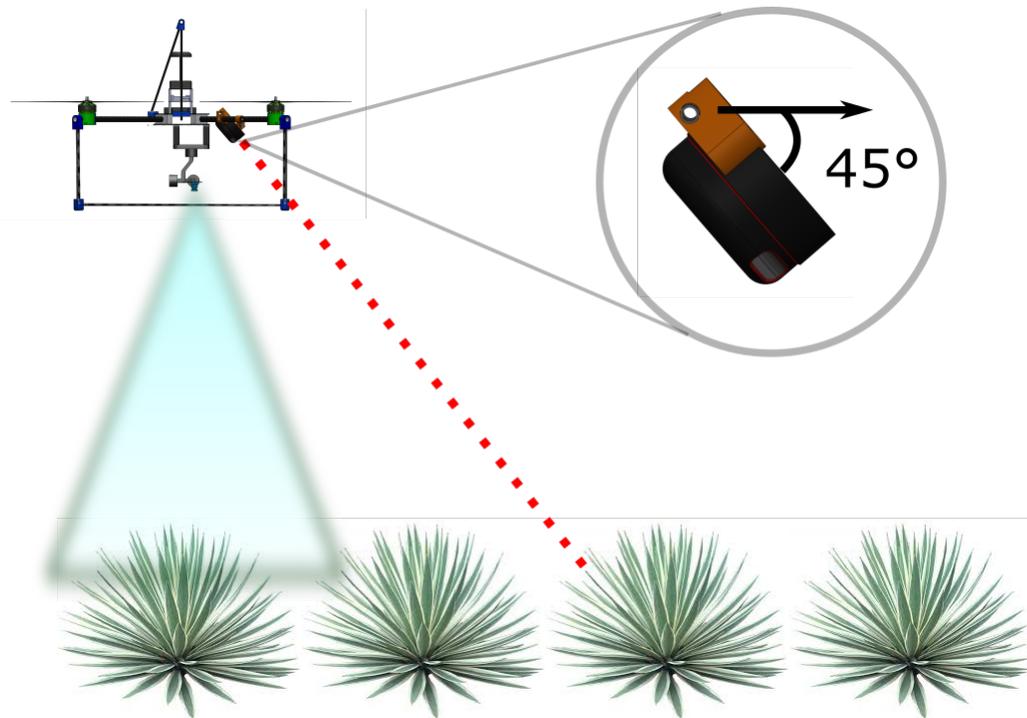


Fig. 3.2: Orientación de los sensores durante el vuelo

De esta forma se pueden detectar los espacios que producen los surcos y seguirlos, esto sirve para la navegación sea más exacta apoyándose de la señal de GPS, la altura nos permite también detectar los surcos laterales para así poder centrarnos con se observa en la Fig. 3.3.

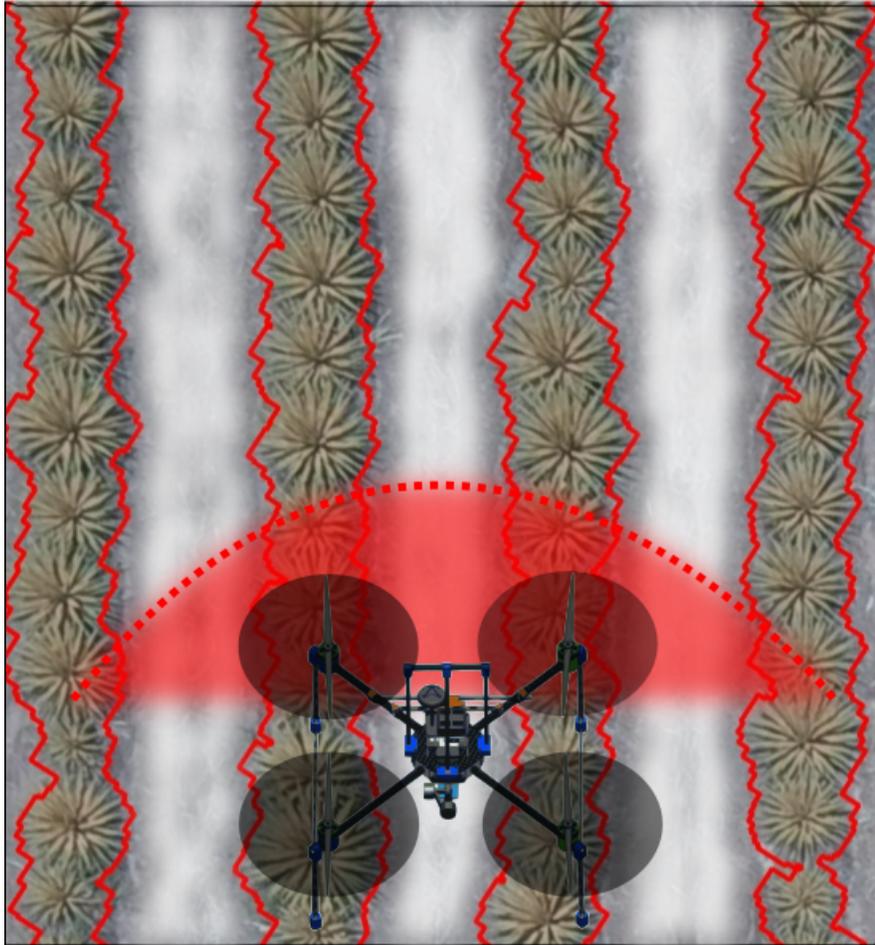


Fig. 3.3: Detección de los surcos en un plantío de agaves

Es necesario saber interpretar los datos proporcionados por el sensor para identificar donde se localizan los espacios correspondientes a los surcos. A continuación, se muestra las lecturas de del sensor en un formato de nube de puntos durante un vuelo sobre una pista de atletismo (Fig. 3.4).

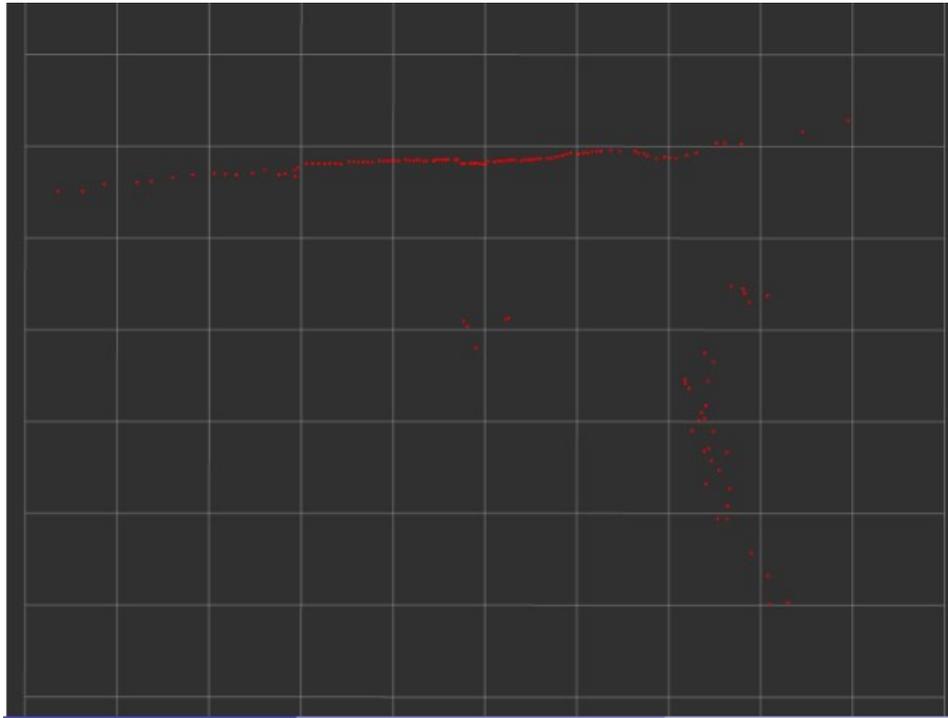


Fig. 3.4: Lecturas de RPLiDAR en pruebas de detección de carril de pista de atletismo

Los puntos rojos en el mapa corresponden a las lecturas de distancia del sensor. En la parte superior central del mapa se puede notar una cuenca en la continuidad de los puntos, esta cuenca corresponde a la zona que delimita el arco del carril de la pista de atletismo, en la Fig. 3.5 marcamos la zona correspondiente al carril, esta lectura se hizo con el dron a una altura de dos metros. Con esto probamos que con la pura curvatura del sendero es posible detectar el camino.

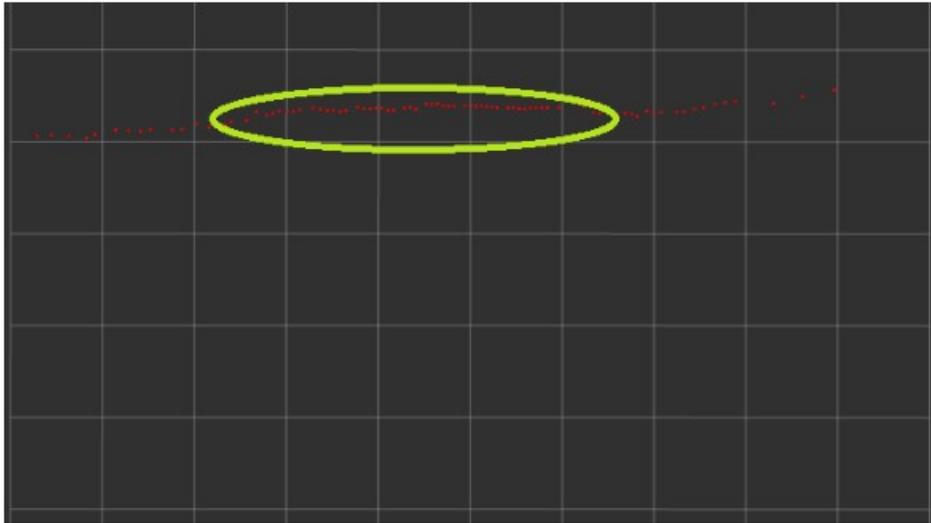


Fig. 3.5: Zona de carril detectado

Más adelante en esta prueba se pasó por una zona donde a los costados del camino hay arbustos, en esta zona las lecturas del arrojaron los datos mostrados en la Fig. 3.6 donde se remarcan los puntos detectados que corresponden a los arbustos, con esto validamos la detección de elementos a los costados del sendero que en el caso ideal serían las plantas de agave en lugar de los arbustos. Esto es caso de que el suelo del sendero fuera irregular y no se detecte en el suelo un patrón para los surcos y se recurra a las lecturas de las plantas que lo siguen.



Fig. 3.6: Detección de árboles laterales a la pista

La detección de la planta se puede extender de modo que logre detectar varios surcos y así poder hacer más fácil el cambio de uno a otro. Para lograrlo se debe de incrementar la altura del vuelo de modo que se alcancen a detectar tres surcos completamente y utilizar el criterio de la detección de los bordes de las plantas como se observa en la Fig. 3.3.

CAPÍTULO 4

PLATAFORMA EXPERIMENTAL

En este capítulo se aborda la validación de la plataforma experimental y para ello se requiere de la construcción de la plataforma, así como de la implementación de la aviónica correspondiente sobre la estructura del tipo Cuadri-rotor para tal efecto.



Fig. 4.1: Plataforma experimental completa

4.1 Diseño y construcción

El diseño de la plataforma experimental tuvo determinados cambios y ajustes a lo largo de los cuatro años de la duración del presente proyecto de tesis con base en a las necesidades que iban surgiendo en pro de las necesidades de la tarea a realizar. Cabe resaltar que la plataforma original con la que se empezó este trabajo era únicamente una estructura de fibra de carbono tipo Talon de cuatro brazos en los cuales los motores no estaban fijos correctamente a la estructura debido a que los orificios de la pieza que une el brazo con el motor no coincidían, por lo que solo era fijado con un solo tornillo y esto presentaba un riesgo ya que podía causar que el motor vibrara y por tanto el tornillo terminara por soltarse y provocar que la plataforma cayera durante el vuelo.

Por esta razón se diseñaron nuevas bases para los motores donde se pudieran fijar correc-

tamente, aprovechando que el diseño podría adaptarse al diseño de los motores, se incluyó una forma ergonómica de colocar los controladores en la parte inferior de la base. Con esto ahorramos espacio parte central del vehículo, además de que se distribuye mejor el peso en toda la estructura.

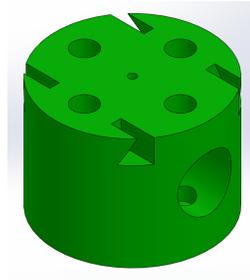


Fig. 4.2: Diseño CAD de base de motor



Fig. 4.3: Brazo de multi-rotor con la base impresa en PLA

Otra de las partes más importantes del diseño es la de la parte inferior y esta involucra varios elementos. En primer lugar, se tiene que mencionar que desde una etapa temprana del proyecto se planteó el uso de visión artificial por lo que era necesario definir como iría colocada la cámara a utilizar y qué se requeriría para fijarla a la estructura. Se eligió que se utilizaría un gimbal de tres ejes, debido a que se necesitaría tener una imagen hasta cierto punto estable para que se pudiera hacer la detección, el empuje de los motores permitía agregar aún más carga por lo que no se veía afectado significativamente agregar este elemento, sin embargo el único espacio disponible donde sería útil el uso de la cámara era la parte inferior, al modificar las bases de los motores el diseño se había quedado sin patas que le ofrecieran

la estabilidad para que se mantuviera estático en el suelo. La única forma de colocarlo en el suelo era sobre el centro de la parte baja que es donde se localizaba la batería, por lo que para agregar la cámara con el gimbal se tuvo que diseñar un tren de aterrizaje ligero y lo suficientemente alto para que el gimbal no tocara el suelo y que fuera rígido para mantuviera estática la plataforma antes del despeje.

Se realizaron varias versiones para este tren de aterrizaje con la suficiente rigidez, pero el punto de apoyo del tren se encontraba aún muy cercano a la línea del centro de gravedad lo que ocasionaba que si no se despegaba desde una superficie completamente vertical se volteara fácilmente antes de que el dron se despegara del suelo. Por lo que surgieron nuevas versiones del tren, que perdieron rigidez para compensar el problema del punto de apoyo, pero que la pérdida de rigidez afectó en algunas pruebas de vuelo ya que en medio del vuelo en ocasiones la vibración ocasionó que el tren de desensamblara en el aire. Finalmente se llegó a la versión final de este, con una rigidez aceptable que, fijada con tornillos, impedían que se desensamblara en el aire y lograba que el despegue se realizara sin gran problema.

También se diseñaron los soportes correspondientes para la fijación del gimbal y el sensor laser rotatorio, en ambos casos también se hicieron varias versiones de estos soportes, sin embargo, el que más complicaciones presentaba era el caso del sensor ya que la ubicación de este sensor era determinante para que los datos obtenidos fueran útiles. Por lo que la mejor zona en donde se pueden recopilar mejor los datos es en la parte frontal del vehículo, sin embargo, un constante problema es de como fijar elementos al vehículo sin afectar el funcionamiento ya que idealmente el peso agregado debería de distribuirse de forma uniforme en todo el cuerpo, por lo que al colocar el sensor en la parte frontal modificaba considerablemente el centro de gravedad. Además de que se tenía que tomar en cuenta que este sensor girar para poder funcionar por lo que el giro provoca cierta vibración si no está bien sujetado a la estructura.

Primero se probó un diseño donde el sensor iba en borde de la parte central del vehículo,

sin embargo, al hacer las primeras pruebas se notó tener la sujeción suficiente para que no le afectara el giro del motor por lo que provocaba una vibración considerable a toda la estructura. Se optó por sujetarlo por medio de dos postes de fibra de carbono para reducir al máximo el peso y estos se fijaron dos de los brazos del vehículo, la posición final del sensor se logró recorrer unos centímetros más hacia el exterior gracias a los postes manteniendo la vibración controlada. Al recorrer el centro de gravedad hacia el eje x del vehículo se tuvo que modificar la posición de los demás elementos para compensarlo y distribuir de mejor manera el peso. Se recorrió tres centímetros hacia atrás el soporte del gimbal y detrás de este se colocó la batería, con estos ajustes se volvió a centrar el centro de gravedad.

El hardware necesario para la estabilización del vehículo y el procesamiento de los datos van montados en el parte central del cuerpo, el autopiloto Pixhawk que es el elemento que debe de estar colocado en el centro de inercia fue el primero en fijarse, sobre él se coloca la computadora principal, la ODROID XU4 que es la que se encarga de recibir los datos de todos los sensores, el procesamiento de imagen y de la red neuronal para la detección de las plantas. En esta misma parte central está fijado el GPS y además de un hub para expandir el número de puertos USB ya que se requieren por lo menos 4 puertos y la computadora solo cuenta con tres puertos. El colocar todos estos elementos uno encima de otro en la parte central los expone en el caso de algún percance además de que para hacer pruebas de calibración y vuelo básicas se requiere sujetar con una mano el vehículo y ahora resulta muy complicado. Por esta razón se diseñó un elemento con el cual se pueda proteger hasta cierto punto estos dispositivos y además ofreciera la facilidad para poderlo sujetar para las pruebas de calibración.

Esta guarda que a su vez funciona como elemento de sujeción está formada por 4 postes de fibra de carbono unidas a través de piezas de PLA con las cuales se fija al cuerpo principal del vehículo, estas están aseguradas por unos tornillos para evitar que se afloje y se desarme por la vibración de la estructura. La forma es realmente básica y no cubre completamente

los dispositivos debido a que dentro de esta zona también está el GPS y por funcionalidad no debe de ser cubierto para que las mediciones sean buenas además de que permitimos la circulación de aire para evitar sobrecalentamiento.

La versión final de la plataforma tiene unas dimensiones de 57cm x 50cm sin contar las propelas y 47cm de altura y un peso de 3.28 Kg con toda la instrumentación.

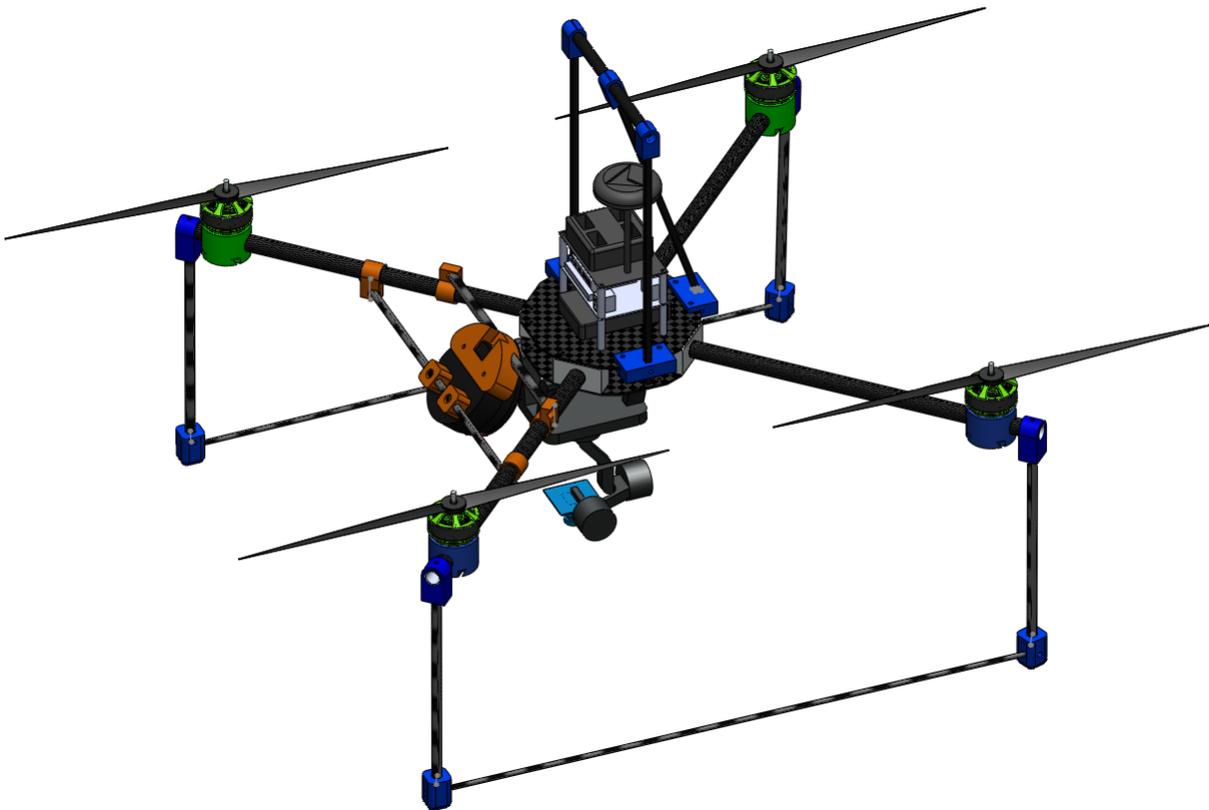


Fig. 4.4: Diseño CAD de la plataforma completa

4.2 Instrumentación

4.2.1 Autopiloto Pixhawk

El Pixhawk [24] es un autopiloto de código abierto derivado del proyecto independiente FMUv2, su diseño es de hardware abierto y combina la funcionalidad del px4FMU al poder incluir todas las librerías ya existentes de px4 . Todo con el objetivo de conseguir una plataforma estándar, con código igualmente estándar, de alta calidad y de precio accesible. La principal idea de este proyecto fue para propósitos académicos, aficionados y desarrolladores para poder explotar toda la versatilidad que ofrece. Al volverse una plataforma de hardware abierto surgieron varias plataformas derivadas de este proyecto como Pixfalcon, Pixhawk 2, Pixracer, Pixhawk 3 Pro, Pixhack v3, Pixhawk 4, Pixhack v5, etc. Originalmente, las primeras versiones de este autopiloto eran manufacturadas por 3DR que al trabajar en conjunto con los desarrolladores de px4 lograron que esta plataforma se convirtiera en la opción numero 1 de desarrollo e investigación para vehículos autónomos. Para la realización de esta tesis se empleó la versión 1 del Pixhawk [20].



Fig. 4.5: Autopiloto Pixhawk version 1 fabricado por 3DRobotics.

Las principales características de este autopiloto son las siguientes:

- Microcontrolador: STM32F427.
- CPU: ARM Cortex M4 corriendo a 180 MHz con una FPU de precisión simple.

- RAM: 256 KB SRAM. Co-procesador para fallos: STM32F100 con CPU: ARM Cortex M3 a 24 MHz, RAM: 8 KB SRAM.
- Giroscopio de 16 bits ST Micro L3GD20H.
- Acelerómetro y magnetómetro de 14 bits ST Micro LSM303D
- Acelerómetro y magnetómetro de 3 ejes Invensense MPU 6000.
- Barómetro MEAS MS5611
- GPS: Compatible con U-Blox 7/8 y U-Blox 6.
- Flujo óptico: Compatible con *PX4 Flow*.
- Botón de seguridad externo.
- Indicador Led multicolor.
- Indicador auditivo multitono.
- Memoria Micro SD de alta tasa de escritura.
- Puerto I2C con extensión de 3 puertos.
- Puerto CAN.
- Puerto ADC.
- 4 puertos UART.
- 8 salidas PWM.
- 6 salidas auxiliares PWM, GPIO, o entrada PWM.
- Puerto S.BUS, PPM y entrada para spektrum.
- Puerto S.BUS de salida.

4.3 GPS

La aplicación que busca este proyecto requiere de navegación en exteriores por lo que se requiere un dispositivo que provea mediciones de posicionamiento con respecto de la tierra. Este dispositivo es el GPS ya que nos ofrece mediciones de latitud y longitud en la tierra, sin embargo, este sensor está limitado en precisión por lo que nos servirá para posicionarnos hasta cierto punto, para una posición mas precisa se hace uso de otro sensor. El GPS utilizado en esta plataforma es el Ublox Neo-M8N el cual cuenta con un conector compatible con el Pixhawk mediante uno de los puertos UART.



Fig. 4.6: GPS Ublox Neo-M8N.

Al ser diseñado para usarlo junto el Pixhawk, las prestaciones de diseño y funcionaidad de este sensor lo hacen ideal para aplicaciones de navegación autónoma en vehículos ligeros. Ofrece una rápida búsqueda de satélites encontrando en condiciones típicas de espacio abierto fácilmente 6 satélites en aproximadamente 10 segundos. Este modelo en específico viene incluye un magnetómetro el cual junto con el interno del pixhawk proveen la señal de guiñada al vehículo, su velocidad de actualización es de 18Hz. A diferencia de versiones anteriores del NEO-7N que ofrecían una precisión de 1.4 a 1.7 metros, éste tiene una precisión de 0.6 metros. Soporta GPS+BD+SBAS y GPS+GLONASS+SBAS. Sus características técnicas más importantes son:

- Tipo de receptor: 72-canales, u-blox M8.
- Soporte de: GPS/QZSS L1 C/A, GLONASS L10F, BeiDou B1, SBAS L1 C/A: WAAS, EGNOS, MSAS Galileo-ready E1B/C (NEO-M8N).
- Velocidad de actualización GNSS: mas de 18 HZ.
- Concurrencia GNSS: arriba de 10 Hz.
- Inicio en frío: 26s.
- Re-adquisición de señal: 1.5 s.
- Temperatura de operación: -40° C a 85° C.
- Voltaje de entrada: 1.65 V a 3.6 V (NEO-M8M).

4.3.1 RPLidar

Este escáner laser es un sensor tipo LiDAR [30] el cual por medio de un sensor laser infrarrojo y un mecanismo de rotación interno ofrece lecturas de detección de 360 grados. Es un sensor ideal para detectar y evitar obstáculos en robots móviles siendo muy común verlos en vehículos terrestres para localización y mapeo simultaneo.



Fig. 4.7: Sensor láser rotatorio utilizado para detectar el camino durante el vuelo

Las características principales desde el punto de vista del hardware son las siguientes:

- Rango de detección: 18 m de radio (36 metros de cobertura total).
- Campo de detección de 360 grados.
- 4cm de grosor (diseño compacto).
- Hasta 8000 muestras de puntos de detección por segundo.
- Frecuencia de rotación: 10 Hz
- Voltaje de alimentación: 5V

Las mediciones de este sensor proveen el ángulo y la lectura de distancia en ese ángulo. Este sensor tiene soporte para poder trabajarlo mediante el sistema operativo de software libre ROS. Tiene una apertura vertical de menos de un grado. A diferencia de otros escáneres infrarrojos este modelo es silencioso y ligero.

Se eligió esta computadora debido a que ofrece el poder de procesamiento suficiente para hacer visión artificial y todo en una tarjeta bastante compacta y ligera. Además de que al ser compatible con Linux ofrece una compatibilidad importante para lograr comunicar todos los sensores. Las características técnicas más importantes de esta computadora son:

- Procesadores: Samsung Exynos5422 Cortex™-A15 2Ghz and Cortex™-A7 Octa core CPUs
- Memoria RAM: 2Gbyte LPDDR3 RAM PoP stacked
- Almacenamiento compatible con eMMC5.0 HS400 Flash Storage
- Puertos USB: 2 x USB 3.0 Host, 1 x USB 2.0 Host
- Puerto Gigabit Ethernet
- Puerto HDMI 1.4a
- Alimentación 5V/4A

4.3.2 Odroid XU4

Esta computadora [21] fue elegida la computadora maestra, es decir a ella llegan la señal de datos provenientes de todos los sensores y es la computadora que hace el procesamiento de imágenes y donde está cargada la red neuronal convolucional. Para funcionar se le carga un sistema operativo de software libre ligero basado en la versión de escritorio de Ubuntu llamada Ubuntu Mate. Sobre el sistema operativo se cargan las librerías de OpenCV y en el entorno de comunicación de sensores ROS en su distribución melodic.



Fig. 4.8: Computadora XU4

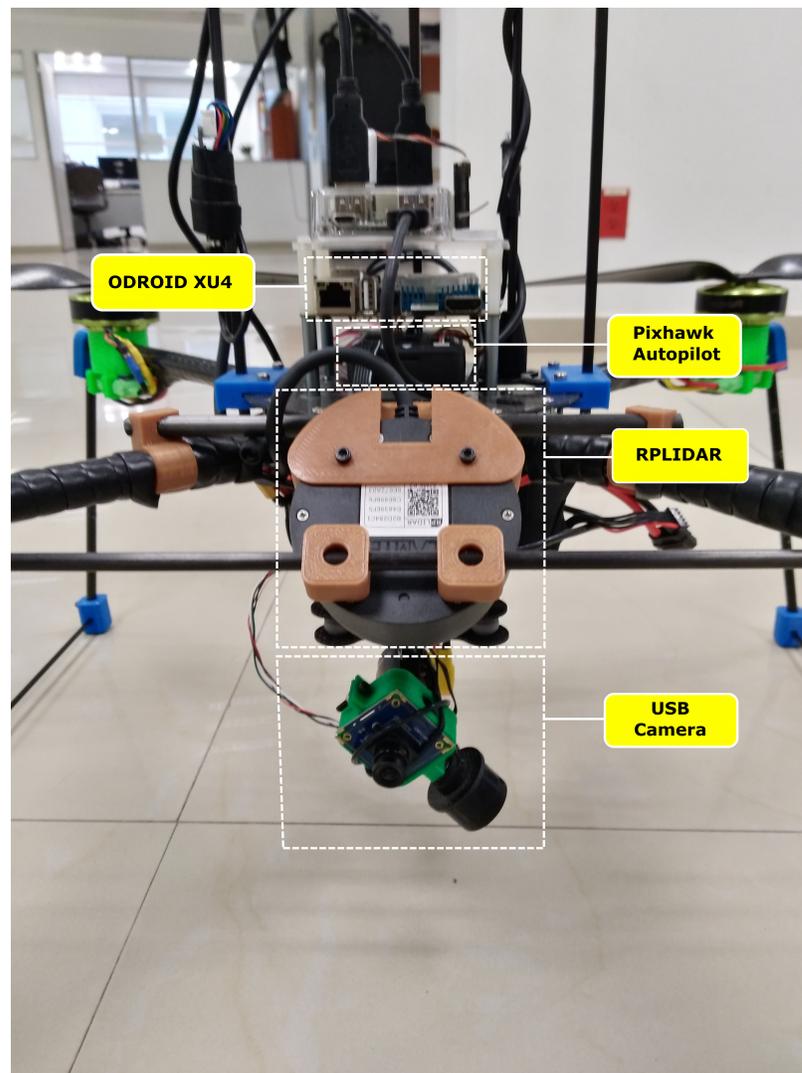


Fig. 4.9: Distribución de los sensores utilizados

4.4 Modelado del Cuadri-rotor

Para un helicóptero de cuatro rotores se emplean las ecuaciones de Euler-Lagrange como en [19].

Las coordenadas generalizadas del vehículo pueden escribirse como:

Monitoreo a partir del seguimiento de un camino con un cuadri-rotor

$$q = (\xi, \eta)^T = (x, y, z, \psi, \theta, \phi)^T \in \mathbb{R}^6, \quad (4.1)$$

donde:

$\xi = (x, y, z) \in \mathbb{R}^3$: denota la posición del centro de masa del helicóptero con respecto al marco inercial.

$\eta = (\psi, \theta, \phi) \in \mathbb{R}^3$: son los ángulos de Euler.

ψ es el ángulo de yaw, θ el ángulo de pitch y ϕ el ángulo de roll que representan la orientación del vehículo.

Se define el Lagrangiano como:

$$L(q, \dot{q}) = T_{trans} + T_{rot} - U, \quad (4.2)$$

donde:

$T_{trans} = \frac{m}{2} \dot{\xi}^T \dot{\xi}$: es la energía cinética translacional.

$T_{rot} = \frac{1}{2} \dot{\omega}^T \mathbf{I} \dot{\omega}$: es la energía cinética rotacional.

$U = mgz$: es la energía potencial del sistema.

z : representa la altura de vehículo.

m : es la masa.

ω : es la velocidad angular.

\mathbf{I} : es la matriz de inercia.

g : es la aceleración gravitacional.

El vector ω respecto a los ejes de coordenadas del cuerpo se relaciona con las velocidades generalizadas, en donde los ángulos de Euler son válidos, utiliza la siguiente relación cinemática.

$$\dot{\eta} = W_{\eta}^{-1} \omega \quad (4.3)$$

donde:

$$W_\eta^{-1} = \begin{bmatrix} -\sin\theta & 0 & 1 \\ \cos\theta\sin\psi & \cos\psi & 0 \\ \cos\theta\sin\psi & -\sin\psi & 0 \end{bmatrix}. \quad (4.4)$$

$$\omega = \begin{bmatrix} \dot{\phi} - \dot{\psi}\sin\theta \\ \dot{\psi}\cos\phi + \dot{\psi}\cos\theta\sin\phi \\ \dot{\psi}\cos\theta\cos\phi - \dot{\theta}\sin\phi \end{bmatrix} \quad (4.5)$$

Se define $\mathbb{J}(\eta) = W_\eta^T \mathbf{I} W_\eta$ tal que

$$T_{rot} = \frac{1}{2} \dot{\eta}^T \mathbb{J} \dot{\eta}. \quad (4.6)$$

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (4.7)$$

La matriz \mathbb{J} actúa como la matriz de inercia para la energía cinética rotacional del helicóptero, expresada en términos de coordenadas generalizadas η . El modelo dinámico completo del helicóptero se obtiene de las ecuaciones de Euler-Lagrange con fuerzas externas generalizadas.

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \begin{bmatrix} F_\xi \\ \tau \end{bmatrix} \quad (4.8)$$

donde F_ξ es la fuerza traslacional aplicada al vehículo debido a la entrada de control principal, entonces se puede expresar lo siguiente: $\hat{F} = [0 \ 0 \ u]$ y $u = f_1 + f_2 + f_3 + f_4$, donde

$$f_i = k_i \omega_i^2 \forall i = 1, \dots, 4 \ k_i > 0, \quad (4.9)$$

k_i es una constante y ω_i es la velocidad angular del motor i -ésimo. Por lo tanto $F_\xi = R\hat{F}$ donde R denota la matriz rotacional $R(\psi, \theta, \phi)$ que representa la orientación del cuadrirotor relacionada al eje de referencia fijo.

Los momentos o pares generalizados son:

$$\tau = \begin{bmatrix} \tau_\psi \\ \tau_\theta \\ \tau_\phi \end{bmatrix} \triangleq \begin{bmatrix} \sum_{i=1}^4 \tau M_i \\ (f_2 - f_4)l \\ (f_3 - f_1)l \end{bmatrix}, \quad (4.10)$$

donde l es la distancia entre los motores y el centro de gravedad, y τM_i es el momento producido por el motor M_i , $i = 1, \dots, 4$, alrededor del centro de gravedad del vehículo.

Debido a que el Lagrangiano no contiene términos en la energía cinética combinando $\dot{\xi}$ con $\dot{\eta}$, las ecuaciones de Euler-Lagrange pueden ser divididas en las dinámicas para las coordenadas de ξ y las coordenadas de η .

$$m\ddot{\xi} + \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} = F_\xi, \quad (4.11)$$

$$\mathbb{J}\ddot{\eta} + \dot{\mathbb{J}}\dot{\eta} - \frac{1}{2} \frac{\partial}{\partial \eta} (\dot{\eta}^T \mathbb{J} \dot{\eta}) = \tau. \quad (4.12)$$

Definiendo los términos de coriolis que contiene los efectos giróscopos y centrífugos asociados a η como:

$$C(\eta, \dot{\eta})\dot{\eta} = \dot{\mathbb{J}}\dot{\eta} - \frac{1}{2} \frac{\partial}{\partial \eta} (\dot{\eta}^T \mathbb{J} \dot{\eta}), \quad (4.13)$$

Finalmente se obtiene:

$$m\ddot{\xi} + \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} = F_\xi. \quad (4.14)$$

$$\mathbb{J}\ddot{\eta} = -C(\eta, \dot{\eta})\dot{\eta} + \tau.$$

Con el objetivo de simplificar el modelo se propone el siguiente cambio de variable:

$$\tau = C(\eta, \dot{\eta})\dot{\eta} + \mathbb{J}\tilde{\tau}, \quad (4.15)$$

donde $\tilde{\tau} = [\tilde{\tau}_\psi, \tilde{\tau}_\theta, \tilde{\tau}_\psi]$ son las nuevas entradas y por lo tanto,

$$\ddot{\eta} = \tilde{\tau}. \quad (4.16)$$

El sistema puede reescribirse como:

$$\begin{aligned} m\ddot{\xi} + mgE_z &= F_\xi \\ \ddot{\eta} &= \tilde{\tau} \end{aligned} \quad (4.17)$$

donde F_ξ está definida como $F_\xi = \begin{bmatrix} -f \sin \theta \\ f \sin \phi \cos \theta \\ f \cos \phi \cos \theta \end{bmatrix}$

Finalmente al multiplicar se obtiene:

$$\begin{aligned} m\ddot{x} &= -u \sin \theta \\ m\ddot{y} &= u \cos \theta \sin \phi \\ m\ddot{z} &= u \cos \theta \cos \phi - mg \\ \ddot{\phi} &= \tilde{\tau}_\phi \\ \ddot{\theta} &= \tilde{\tau}_\theta \\ \ddot{\psi} &= \tilde{\tau}_\psi. \end{aligned} \quad (4.18)$$

donde x y y son las coordenadas en el plano horizontal, z es la posición vertical, y $\tilde{\tau}_\psi$, $\tilde{\tau}_\theta$, y $\tilde{\tau}_\phi$, son los momentos de yaw, pitch y roll respectivamente, los cuales están relacionados con los momentos generalizados τ_ψ , τ_θ y τ_ϕ de la ecuación (4.15).

4.4.1 Implementación de un controlador PD

Para implementar este controlador en la plataforma, se utilizó el modelo reducido (4.18), y se dividió en subsistemas, posteriormente se aplicó una cancelación de no-linealidades como en [19].

Subsistema z :

Considere el subsistema correspondiente a la posición vertical:

$$m\ddot{z} = u \cos \theta \cos \phi - mg.$$

Aplicamos el siguiente control con el objetivo de cancelar las no-linealidades:

$$u = m(u_1 + g)(\cos \theta \cos \phi)^{-1} \quad (4.19)$$

y sustituyendo nos queda lo siguiente:

$$m\ddot{z} = \frac{m(u_1 + g) \cos \theta \cos \phi}{\cos \theta \cos \phi} - mg \quad \ddot{z} = u_1$$

Y se obtiene su representación en espacio de estados:

$$x_1 = z \quad x_2 = \dot{z} \quad \dot{x}_1 = x_2 \quad \dot{x}_2 = u_1$$

con $\cos \theta \cos \phi \neq 0$, $\theta, \phi \in (-\frac{\pi}{2}, \frac{\pi}{2})$

y obtenemos un nuevo sistema lineal de la forma:

$$\dot{x}_z = A_z x_z + B_z u_1, \quad (4.20)$$

donde las matrices $A_z = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, $B_z = \begin{bmatrix} 0 \\ 1/m \end{bmatrix}$.

Para verificar que las matrices son controlables suponemos $m = 1$ para armar la matriz de controlabilidad:

$$C = [B \ AB] = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

como se puede observar la matriz es de rango 2 y es rango pleno por lo que se puede afirmar que las matrices A y B son controlables.

Entonces podemos implementar un controlador PD y para ello fijamos la referencia en cero . Por lo que podemos proponer u_1 como

$$u_1 = -kp_z x_z - kd_z \dot{x}_z. \quad (4.21)$$

donde:

kp_z : ganancia proporcional del subsistema z

kd_z : ganancia derivativa del subsistema z

Se obtiene la determinante de $[sI - (A + BK)]$, donde los valores de $k_{p,z} = 30$, $kd_z = 120$ fueron elegidas experimentalmente para la plataforma utilizada:

$$P(s) = s^2 + 120s + 30 \quad (4.22)$$

Las raíces obtenidas son:

$$s_1 = -119.7495,$$

$$s_2 = -0.2505,$$

Las raíces se encuentran en la parte real negativa por lo que podemos concluir que el sistema se comporta en forma estable.

Subsistema ψ :

$$\ddot{\psi} = \tau_\psi, \quad (4.23)$$

Cuya representación en espacio estado es:

$$\dot{x}_\psi = A_\psi x_\psi + B_\psi \tau_\psi \quad (4.24)$$

donde

$$A_\psi = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad x_\psi = \begin{bmatrix} x_{1,\psi} \\ x_{2,\psi} \end{bmatrix} = \begin{bmatrix} \psi \\ \dot{\psi} \end{bmatrix}, \quad B_\psi = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (4.25)$$

y se puede elegir un controlador Proporcional Derivativo, suponiendo que la referencia es cero,

$$\tau_\psi = -kp_\psi x_{1,\psi} - kd_\psi x_{2,\psi}, \quad (4.26)$$

donde:

$$kp_\psi = 72$$

$$kd_\psi = 92$$

Para verificar la colocación de las raíces, se obtiene la determinante de $[sI - (A + BK)]$,

$$P(s) = s^2 + 92s + 72 \quad (4.27)$$

cuyas raíces son:

$$\begin{aligned} s_1 &= -119.2106, \\ s_2 &= -0.7894, \end{aligned} \quad (4.28)$$

Las raíces están en la parte real negativa por lo que se puede decir que el subsistema es estable.

Subsistema $y - \phi$:

Considere el subsistema:

$$\begin{aligned} m\ddot{y} &= u \cos \theta \sin \phi \\ \ddot{\phi} &= \tau_\phi \end{aligned} \quad (4.29)$$

sustituyendo el control (4.19) se obtiene lo siguiente:

$$m\ddot{y} = \frac{m(u_1 + g) \cos \theta \sin \phi}{\cos \theta \cos \phi} \quad (4.30)$$

$$\ddot{\phi} = \tau_\phi \quad (4.31)$$

simplificando

$$\begin{aligned} \ddot{y} &= (u_1 + g) \tan \phi \\ \ddot{\phi} &= \tau_\phi \end{aligned} \quad (4.32)$$

En esta parte se realizan algunas suposiciones la primera es que:

$$u_1 \longrightarrow 0 \text{ cuando } t \longrightarrow \infty.$$

Por lo que el sistema queda simplificado como:

$$\begin{aligned} \ddot{y} &= g \tan \phi \\ \ddot{\phi} &= \tau_\phi \end{aligned} \quad (4.33)$$

Enseguida realizamos una segunda suposición:

que τ_ϕ es capaz de estabilizar al sistema cerca de la referencia por lo que se puede asumir que $\tan \phi \cong \phi$ y entonces obtenemos una nueva aproximación lineal del sistema.

$$\dot{x}_{y,\phi} = A_{y,\phi}x_{y,\phi} + B_{y,\phi}\tau_\phi \quad (4.34)$$

$$\text{donde } A_{y,\phi} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & g & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad x_{y,\phi} = \begin{bmatrix} y \\ \dot{y} \\ \phi \\ \dot{\phi} \end{bmatrix}, \quad B_{y,\phi} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

La referencia está fijada en cero y se considera $g = 9.8$, se puede definir el control τ_ϕ como:

$$\tau_\phi = -kp_1y - kd_1\dot{y} - kp_2\phi - kd_2\dot{\phi} \quad (4.35)$$

donde:

$$kp_1 = 0.1$$

$$kd_1 = 0.98$$

$$kp_2 = 30$$

$$kd_2 = 60$$

son las ganancias para el controlador diseñado, éstas fueron elegidas experimentalmente.

Utilizando las ganancias anteriormente planteadas podemos hallar el siguiente polinomio característico:

$$P(s) = s^4 + 60s^3 + 30s^2 + 9.604s + 0.98, \quad (4.36)$$

y las raíces obtenidas son:

$$\begin{aligned} s_1 &= -59.4985 + 0.0000i \\ s_2 &= -0.1744 + 0.2783i, \\ s_3 &= -0.1744 - 0.2783i, \\ s_4 &= -0.1527 + 0.0000i, \end{aligned} \quad (4.37)$$

De acuerdo al lugar de las raíces se puede concluir que el sistema se comportará de manera estable.

Subsistema $x - \theta$:

Considere la siguiente ecuación:

$$\begin{aligned} m\ddot{x} &= -u \sin \theta \\ \ddot{\theta} &= \tau_\theta \end{aligned} \quad (4.38)$$

Se sustituye el control (4.19) y se obtiene:

$$\begin{aligned} m\ddot{x} &= -\frac{m(u_1+g) \sin \theta}{\cos \theta \cos \phi} \\ \ddot{\theta} &= \tau_\theta \end{aligned} \quad (4.39)$$

Simplificando:

$$\begin{aligned} \ddot{x} &= -\frac{(u_1+g) \tan \theta}{\cos \phi} \\ \ddot{\theta} &= \tau_\theta \end{aligned} \quad (4.40)$$

Al igual que en el subsistema anterior se hace la suposición de que:

$$u_1 \longrightarrow 0 \text{ cuando } t \longrightarrow \infty.$$

Se realiza además la suposición de que τ_θ es capaz de estabilizar al sistema cerca de la referencia por lo que se puede asumir que $\cos \theta \cong 1$ y $\tan \theta \cong \theta$ entonces obtenemos una nueva aproximación lineal del sistema.

$$\dot{x}_{x,\theta} = A_{x,\theta}x_{x,\theta} + B_{x,\theta}\tau_\theta \quad (4.41)$$

donde $A_{x,\theta} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -g & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$, $x_{x,\theta} = \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix}$, $B_{x,\theta} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$

Se puede definir el control τ_θ cuando la referencia se fija en 0, con $g = 9.8$ de manera siguiente:

$$\tau_\theta = -kp_1x - kd_1\dot{x} - kp_2\theta - kd_2\dot{\theta} \quad (4.42)$$

donde:

$$kp_1 = 0.1$$

$$kd_1 = 0.99$$

$$kp_2 = 30$$

$$kd_2 = 60$$

se eligieron experimentalmente.

Con las ganancias presentadas anteriormente se puede obtener el polinomio característico siguiente:

$$P(s) = s^4 + 60s^3 + 30s^2 + 9.7s + 0.98, \quad (4.43)$$

donde las raíces obtenidas son:

$$s_1 = -59.4985 + 0.0000i$$

$$s_2 = -0.1760 + 0.2414i$$

$$s_3 = -0.1760 - 0.2414i$$

$$s_4 = -0.1495 + 0.0000i$$

(4.44)

Como se observa todas las raíces son de parte real negativa por lo que se garantiza que el sistema es estable.

CAPÍTULO 5

RESULTADOS EXPERIMENTALES

A partir de los algoritmos propuestos en las secciones anteriores se realizaron experimentos, al tener un enfoque de aplicación dadas circunstancias actuales a continuación se muestran los resultados obtenidos con datos provenientes de condiciones reales actuales. Además, se incluyen algunas métricas que evidencian el desempeño real de los algoritmos propuestos.

5.1 Detección de Agaves

Para las pruebas experimentales de detección se utilizó una sección de un ortomosaico (Fig. 5.1) en las condiciones que se mencionan en la sección 2.3. La imagen corresponde a un plantío de agave visto desde una altura de 50m con respecto del suelo. Como se puede notar en la imagen existen plantas de diferentes tamaños y que en algunos casos las más grandes ocultan a algunas muy pequeñas.



Fig. 5.1: Imagen original

Como primer paso y de acuerdo con el método de detección propuesto (sección 2.2.1) está la parte de segmentación por color que comprende un subproceso de umbralización de valores para los tres canales de color RGB (rojo, verde, azul) que corresponden a los colores que presenta la planta. Aplicando dicha umbralización de los colores indicados se obtiene la siguiente imagen (Fig. 5.2).

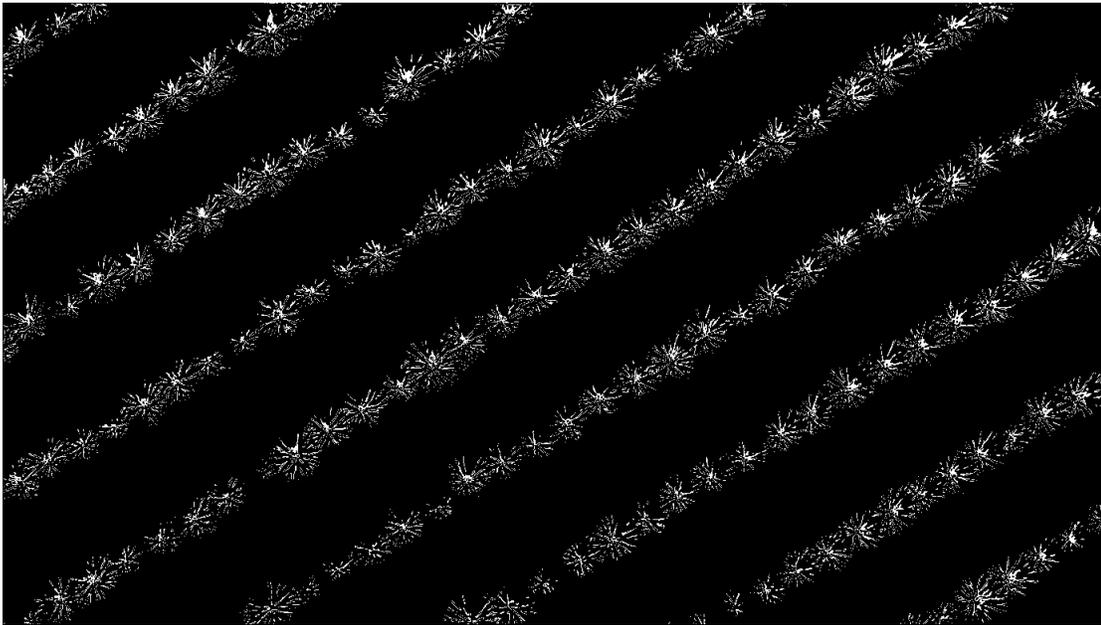


Fig. 5.2: Imagen binaria umbralizada de los valores RGB de los agaves

Como se puede ver en la imagen anterior (Fig. 5.2) se resaltan las pencas de los agaves, esto servirá para la siguiente fase, pero por ahora se desea establecer las áreas donde hay agaves por lo que siguiendo con el procesamiento de segmentación de la imagen, como se menciona en la sección 2.2.1 se hizo un proceso de tratamiento morfológico de dilatación de 10 iteraciones con el fin de diferenciar las zonas de agave de las zonas de tierra y demás, para ellos se utilizó un kernel 3 x 3 obteniendo el siguiente resultado (Fig. 5.3).



Fig. 5.3: Imagen dilatada de las zonas de agave

A partir de la imagen binaria dilatada (Fig. 5.3) se puede obtener una visualización de nuestra área de interés. Para efectos de visualización y apreciación de esta área se creó la Figura 5.4, esta es la resultante de un proceso de detección de contornos, la cual resalta los límites del área que en este caso es el área de color blanco de la Fig. 5.3 y que se imprime en su versión a color para una mejor apreciación. Esta imagen (Fig. 5.4) no forma parte del método propuesto, pero se muestra para que se pueda apreciar que efectivamente segmenta de forma correcta el área de los agaves.

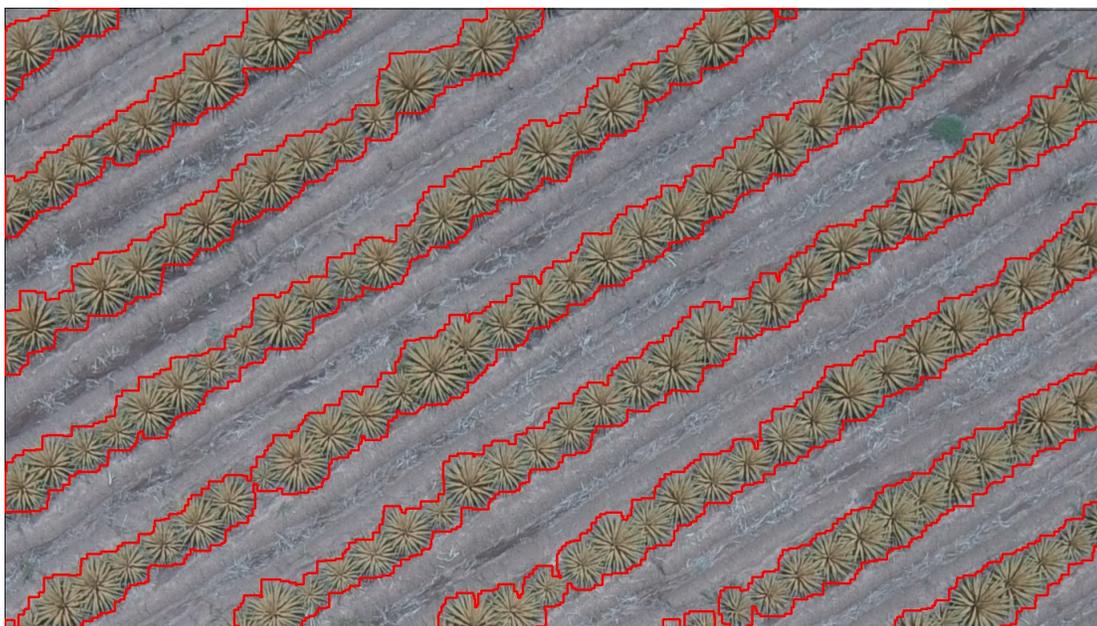


Fig. 5.4: Área de agave segmentada

Claramente se puede observar en la Fig. 5.4 que se cubre la zona donde están los magueyes sin hacer distinción entre ellos, es por ello que la siguiente fase realizará el proceso de separación, pero antes se tiene que pasar esta información en un formato que sea más útil para las fases posteriores. Como se muestra en la Figura 2.10, el siguiente subproceso en esta fase consta de un proceso de ventaneo del área de interés. Es por ello que una matriz de 50 x 50 píxeles a la que denominamos ventana es pasada a lo largo de toda la imagen. Aquellas ventanas que contengan áreas donde hay una parte de un agave (zona blanca de la Fig. 5.3) en su mayoría serán guardadas y consideradas como ventanas candidatas descartando ventanas donde no se contengan mucha área de agave (ventanas con mayor parte negra o completamente negra de Fig. 5.3), esto con el fin de empezar a diferenciar un maguay de otro. En este paso todas las ventanas que cumplen con este criterio son consideradas como un agave.

En este punto entramos a la siguiente fase de método de detección descrito en la sección 2.2.2 correspondiente a la detección del centro. Y si observamos en el esquema de la Figura

2.10 vamos a agregar un siguiente criterio para considerar un agave y para este paso se necesita ocupar la imagen segmentada sin dilatar para encontrar líneas dentro de cada ventana. Una ventana corresponde a una planta de agave si cumple con las propiedades graficas mostradas en la Figura 2.11 y se puede separar las ventanas que tengan imágenes centradas del agave por medio de la detección de líneas que, si recordamos de la sección 2.2.2, se puede notar que se intersectan en el centro de la planta. Si separamos estas ventanas reducimos el número de ventanas candidatas y se van quedando las únicas que tienen un centro de agave en el área central de la ventana como se puede observar en la Fig. 5.5.

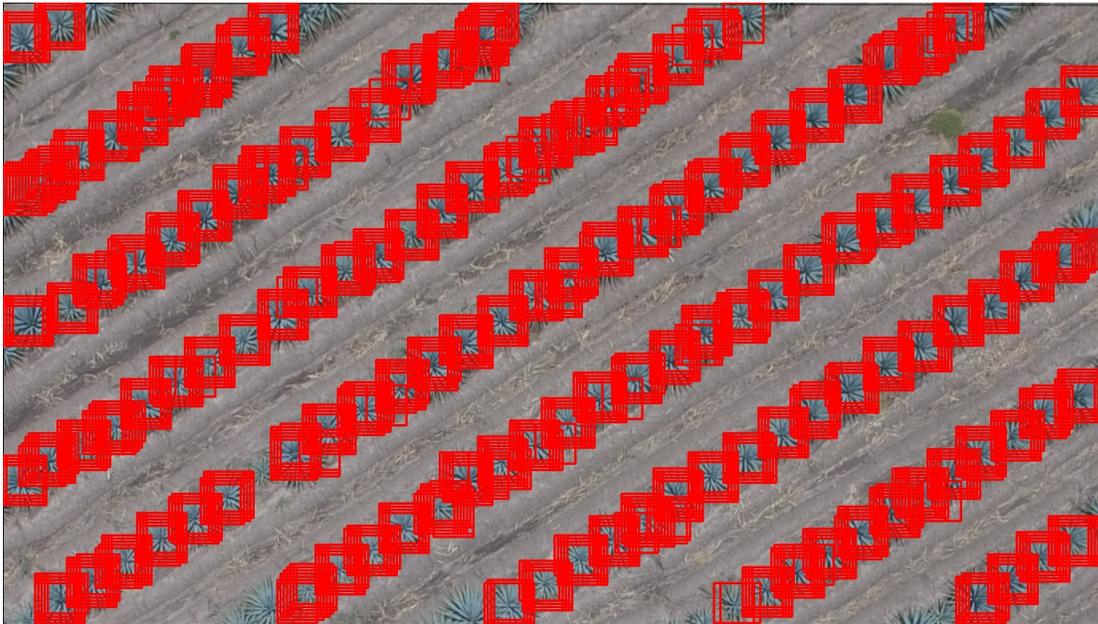


Fig. 5.5: Ventanas que se detectaron con centros de plantas de agave

De forma resumida todas las fases de este proceso de detección de los centros para la selección de ventanas candidatas se observan en la siguiente imagen (Fig. 5.6). De izquierda a derecha, primero se ve la imagen umbralizada, después las líneas detectadas en esa ventana, posteriormente se separa la ventana en celdas y se revisa donde se intersectan las líneas detectadas, si las líneas se intersectan en la celda central se considera una nueva ventana candidata.

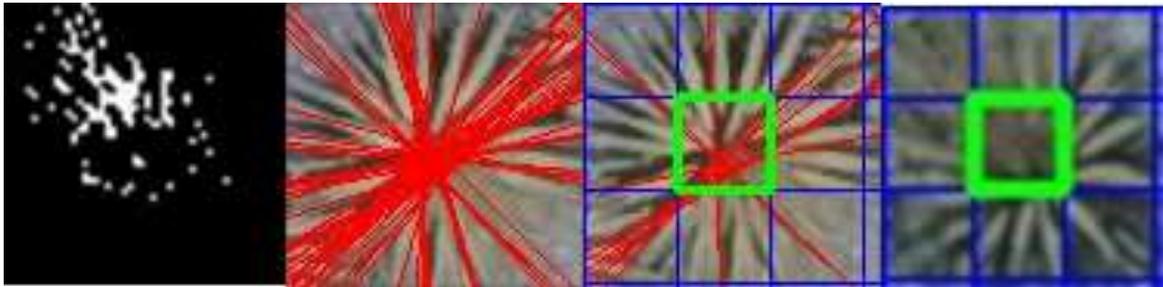


Fig. 5.6: Proceso de selección de ventanas por detección de centros de plantas de agave

Con este paso se pueden empezar a ver que se resaltan los agaves de forma individual con varias detecciones de cada uno por lo que ahora se aplica el mapa de calor descrito en la sección 2.2.2 de los centros para poder visualizar mejor cada uno. Este se crea a partir de las ventanas candidatas y como se puede ver en la Figura 5.7 se forman puntos blancos que corresponden a los centros de cada uno de los agaves por lo que se puede elegir una sola ventana para cada punto y que resulta en las ventanas que se muestran en la Figura 5.8.

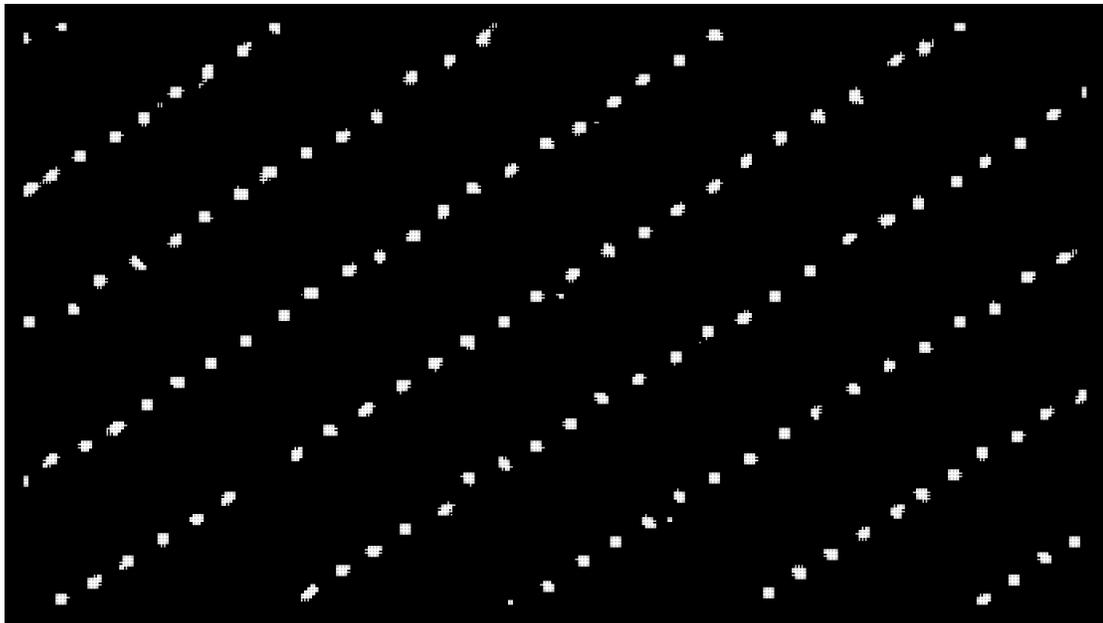


Fig. 5.7: Mapa de calor generado por las ventanas candidatas

En esta imagen(Fig. 5.7) se pueden ver individualmente los centros de cada maguey

considerando que se detectó correctamente un centro. Sin embargo, el hecho de que los agaves estén plantados tan cerca uno del otro puede ocasionar que se detecten falsos centros, esto es causado a que las pencas más largas suelen llegar a cruzarse con las de otra planta y ocasionar que se detecte como un centro como se puede observar en la imagen de las ultimas ventanas candidatas (Fig. 5.8).

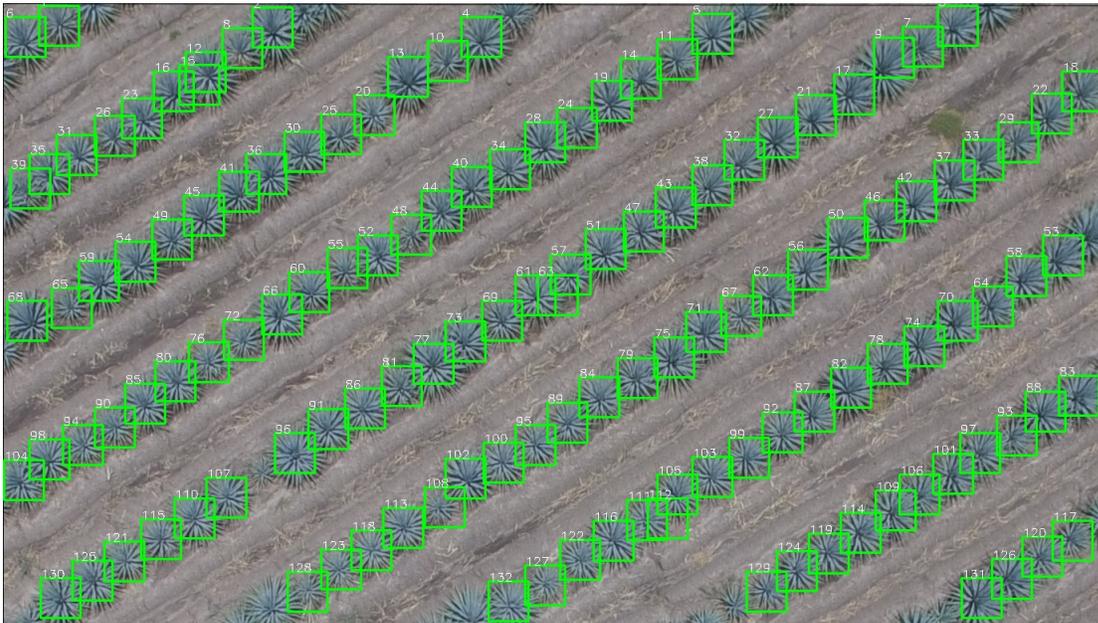


Fig. 5.8: Preconteo de agaves a partir del mapa de calor

Para solucionar el problema del cruce de pencas la última fase incluye la clasificación por medio de la red neuronal convolucional. A partir de esta detección se puede desprejar las ventanas que no corresponden completamente a un maguey, al trabajar con menos ventanas el proceso se hace más rápido que si se hubiera hecho con todas las ventanas de la imagen inicial sin manejar ventanas candidatas.

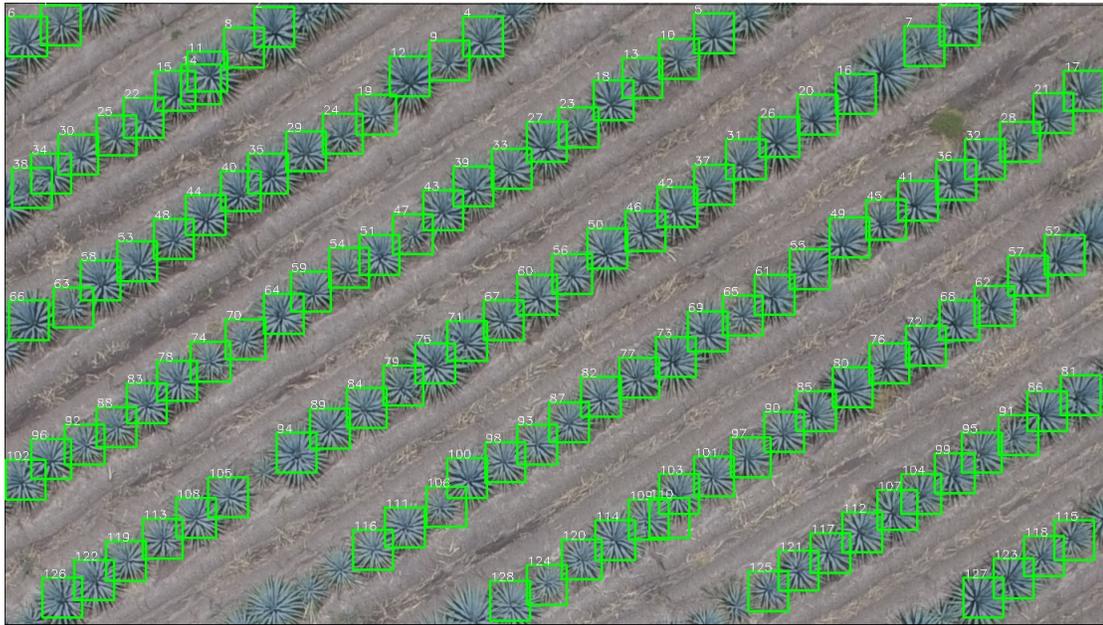


Fig. 5.9: Validación y conteo final de agaves con CNN

Con esto se procede al último paso que es el conteo de elementos que cumplieron con todo. De la imagen original manualmente se contaron el número de agaves para compararlos con el arrojado por el algoritmo. La imagen original cuenta con 130 plantas de maguey y el algoritmo arroja un total de 128 con lo que se logra un mejor desempeño que con el método anterior.

Verificando las detecciones verdaderas, se obtiene una tasa de detección verdadera del 98% para este algoritmo. Para tener una referencia de comparación se realizó un clasificador tipo Haar en cascada con las herramientas que ofrece OpenCV, se utilizó la misma base de datos para el entrenamiento del clasificador. Al aplicar este clasificador a la misma imagen de prueba arrojó un nivel de detección menor con respecto al método que se propone en este trabajo. Se calcularon las métricas mencionadas en la sección 2.1.2 para medir el desempeño para ambos métodos y hacer una comparación correcta. La comparación se muestra en la Tabla 5.1.

Metodo	Recall	Precision	F 1
Clasificador Haar	0.42	0.9	0.57
Metodo propuesto	0.98	0.95	0.96

Tabla 5.1: Desempeño de los dos métodos de detección

5.2 Discusión

De los resultados observados se puede decir que, a diferencia de la mayoría de los trabajos citados en la Introducción, este trabajo se centró en la solución de un problema social con una solución tecnológica. Los resultados de detección y recuento con la precisión mostrada pueden interpretarse como una señal de que es posible llevar este trabajo aún más lejos. Como trabajo futuro, planeamos extender nuestro estudio también para detectar la edad de la planta y detectar si está madura y lista para ser cosechada, y así informar al agricultor de la ubicación de la planta. Observe que se pueden lograr resultados muy prometedores diseñando un método que utilice herramientas que sean simples y más conocidas que los algoritmos de detección más sofisticados. Sin embargo, se pueden probar otros enfoques para detectar el agave en el futuro, como los métodos de extremo a extremo como YOLO.

La comparación del enfoque propuesto con respecto al método YOLO podría ser interesante y podría mejorar la velocidad de procesamiento con una precisión similar. El clasificador Haar es una herramienta universal que se ha incluido en OpenCV desde hace mucho tiempo, y el entrenamiento es más rápido, lo que simplifica el aumento del número de pruebas y podría ser programado en un mayor número de sistemas embebidos. A pesar de esto, el método propuesto no es perfecto; como se puede ver en la Figura 5.9, todavía tiene algunos falsos positivos. Esto significa que hay aspectos a mejorar. Se podrían haber utilizado cámaras espectrales para mejorar la tarea de detección. Sin embargo, dado que esta tecnología tiene que estar disponible para los agricultores, preferimos no utilizar cámaras espectrales, lo que habría aumentado el precio del prototipo. Además, durante los experimentos, se observó que

el preprocesamiento mejoraba el nivel de detección. El preprocesamiento elimina aquellas áreas donde no hay plantas. Por tanto, el procesamiento de detección con la red neuronal se aligera ya que la búsqueda se realiza en una ventana más pequeña.

El método propuesto es preciso, pero, por el contrario, tiene un alto costo de cómputo debido a todas las etapas que se incluyen. El método propuesto todavía falla a veces en detectar una planta. Este problema se podría intentar resolver aumentando el número de imágenes de entrenamiento positivas. El algoritmo propuesto también podría mejorarse agregando imágenes de plantas de agave más jóvenes y más pequeñas. Otra mejora podría obtenerse agregando imágenes con diferente iluminación tomadas en diferentes momentos del día y en diferentes condiciones climáticas.

CAPÍTULO 6

CONCLUSIONES Y TRABAJO

FUTURO

Basado en los resultados obtenidos se obtuvieron las siguientes conclusiones que se muestran dentro de este capítulo. También, se analizaron las posibles mejoras a desarrollar sobre la plataforma para dar continuidad a este proyecto de tesis.

6.1 Conclusiones

Se presentó un método basado en aprendizaje profundo para la detección y recuento en las plantaciones de agave. Con este método propuesto se obtuvo como resultado un buen porcentaje de detección y conteo individual de las plantas de agave del 98%. Este porcentaje de desempeño justifica que la aplicación de la técnica propuesta puede ser útil para mantener monitoreadas las plantaciones de esta especie de planta. Como se pudo observar en comparación con otros métodos, la precisión del método propuesto es mejor, pero representa un costo de computación más alto debido a todas las etapas que se incluyen. Para incrementar las detecciones correctas que se obtienen de una buena identificación de la planta (agave), se propuso aumentar el número de imágenes positivas de entrenamiento. Además, se consideró



(a) Detección con clasificador Haar de cascada

(b) Detección con método propuesto

Fig. 6.1: Comparación de detección

agregar imágenes de plantas de agave más jóvenes y más pequeñas, así como imágenes con diferentes luces, como imágenes tomadas en diferentes momentos del día y otras condiciones climáticas, teniendo en cuenta que es probable que aparezcan sombras que podrían afectar las etapas finales del procesamiento para la detección adecuada de la planta induciendo determinado tipo de ruido en la captura y adquisición de la imagen a analizar. El uso de este tipo de tecnología combinado con redes neuronales aún puede crecer en tareas más complejas para aplicaciones como esta.

Con este trabajo se demostró el importante impacto de los vehículos aéreos no tripulados en el área de la agricultura de precisión y como puede contribuir también en el aspecto de seguridad y eficacia al reducir los costos de operación en determinados sectores agrícolas. Hoy en día las estimaciones de producción en esta área todavía se suelen hacer mediante cálculos empíricos por parte de las personas que trabajan en los plantíos, con esta solución basada en la aplicación de redes neuronales se puede lograr una mayor exactitud en el cálculo de las cosechas que se tendrán. Además, se reduce el tiempo de inspección para plantaciones muy extensas en cuestión de varias hectáreas. En el aspecto de seguridad también se presentó una

solución al problema que se planteó del robo de agaves en plantíos y esto tiene impacto de dos formas:

- La primera es que al realizar el monitoreo vía aérea evita que una persona exponga su vida al cuidar estos plantíos.
- La segunda es que al poder hacer el monitoreo de forma rápida esto da pie a que se pueda realizar de forma regular y con esto conseguir un control al día de cuantas plantas hay en cierta área y poder focalizarse en áreas donde hay una mayor cantidad de plantas cosechables.

6.2 Trabajo futuro

Como trabajo futuro se plantea mejorar la parte de detección de los agaves de modo que sea capaz de diferenciar entre un agave que aún esta joven y uno que esta listo para ser cosechado. Esto impactaría directamente a otro de los problemas que fueron planteados al principio el cual, consiste en evitar los robos de las plantas de agave en dichas plantaciones.

Además, como se mencionó en la discusión se puede mejorar la detección del método propuesto agregando imágenes positivas y negativas a la base de datos creada para observar que tanto mejora la detección y tal vez poder modificar alguna de las fases que tienen una gran carga computacional. También se pueden experimentar con otros tipos de algoritmos de detección como YOLO que esta en constante crecimiento últimamente. En la parte de la plataforma experimental también hay muchos aspectos que se pueden mejorar, los sensores ocupados en este proyecto pueden ser remplazados por otro tipo de sensores que, si bien pueden ser más caros, pudieran ofrecernos otro tipo de información útil. Las cámaras multi-espectrales pudieran ofrecer información útil referente al estado de la planta y ayudarnos a la idea de detectar si la planta ya está lista para ser cosechada y poder mandar algún tipo de señal a una base de control en tierra. La parte de navegación que fue con la que más faltó

experimentar es una de las que también se puede empezar, fortalecer el método de detección de los caminos también es algo importante que se puede trabajar en un futuro cercano. Obviamente que también al cada año llegar tecnología nueva la recomendación también sería renovar la parte de sensores y computadora de procesamiento de imagen ya que es otro de los problemas que se tiene con esta versión, con alguna computadora compacta de última generación pudiéramos disminuir el retraso el proceso de análisis de las imágenes.

6.3 Artículos

Revistas internacionales

- Donovan Flores, J. M. Vazquez-Nicolas, Iván Gonzalez, Rogelio Lozano, Jeso Luis Hernandez Toral. “Automated Agave Detection and Counting Using a Convolutional Neural Network and Unmanned Aerial Systems”. *MDPI Drones*, 2021. **Publicado** [7]
- Lozano R., Salazar S., Flores D. Iván Gonzalez. “PVTOL global stabilization using a nested saturation control”. *International Journal of Control (IJC)*, 2021. **Sometido**

Congresos internacionales

- Orlando García, Donovan Flores, Omar Santos, Hugo Romero, Sergio Salazar, y Rogelio Lozano. “Autonomous take-off and landing on a colored platform”. *In Unmanned Aircraft Systems (ICUAS)*, 2017. *International Conference on* (pp. 877-884). *IEEE*. [8]

ANEXOS A

APÉNDICE A

A.1 PVTOL global stabilization using a nested saturation control

A.1.1 PVTOL Model

The model of the PVTOL is given by [18]:

$$m\ddot{x} = u \sin \theta \tag{A.1}$$

$$m\ddot{z} = u \cos \theta - mg \tag{A.2}$$

$$\ddot{\theta} = \tau \tag{A.3}$$

where m is the mass, θ is the angle of the aircraft with respect to the horizontal line, g is the gravitational acceleration, x is the horizontal displacement, z is the vertical displacement. u and τ are the total thrust and torque respectively.

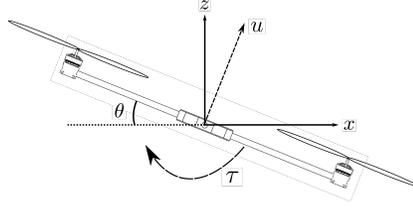


Fig. A.1: PVTOL configuration

A.1.2 Upperbound on the orientation rate $\dot{\theta}$

In this section we will present the nested saturation controller applied to the $x - \theta$ subsystem. We will show that after some finite time the upperbound on $\dot{\theta}$ is smaller than a given value.

The state of the $x - \theta$ subsystem is given as

$$x_1 = \frac{x}{g} \quad x_2 = \frac{\dot{x}}{g} \quad x_3 = \theta \quad x_4 = \dot{\theta} \quad (\text{A.4})$$

Let us define the following variables

$$\nu_1 = x_3 + x_4$$

$$\nu_2 = x_2 + 2x_3 + x_4 \quad (\text{A.5})$$

$$\nu_3 = x_1 + 3x_2 + 3x_3 + x_4$$

consider the subsystem

$$\dot{x}_4 = \tau \quad (\text{A.6})$$

$$\dot{\nu}_1 = x_4 + \tau \quad (\text{A.7})$$

and the torque input given by the following nested saturation control law

$$\tau = -\sigma_a(x_4 + \sigma_b(\nu_1 + \sigma_c(\nu_2 + \sigma_d(\nu_3)))) \quad (\text{A.8})$$

where σ_a is the saturation function:

$$\sigma_a = \begin{cases} a & x > a \\ x & -a \leq x \leq a \\ -a & x < -a \end{cases} \quad (\text{A.9})$$

Introducing (A.8) into (A.6)

$$\dot{x}_4 = -\sigma_a(x_4 + \sigma_b(w_1)) \quad (\text{A.10})$$

with

$$w_1 = \nu_1 + \sigma_c(\nu_2 + \sigma_d(\nu_3)) \quad (\text{A.11})$$

Assume that

$$a \geq 2b + \epsilon_1 \quad (\text{A.12})$$

for some $\epsilon_1 > 0$.

Appendix A gives the expression for a finite time t_1 such that for some $\epsilon_1 > 0$

$$|\dot{\theta}| = |x_4| \leq b + \epsilon_1 \quad \forall \quad t \geq t_1 \quad (\text{A.13})$$

A.1.3 Upperbound on the orientation angle θ .

In this section we will show that after some finite time θ will belong to an interval which does not include the singular point $\theta = 90^\circ$.

After $t = t_1$ and in view of (A.7), (A.8), (A.12) and (A.13), (A.7) reduces to

$$\dot{\nu}_1 = -\sigma_b(\nu_1 + \sigma_c(w_2)) \quad (\text{A.14})$$

with

$$w_2 = \nu_2 + \sigma_d(\nu_3) \quad (\text{A.15})$$

Let

$$b \geq 2c + \epsilon_2 \quad (\text{A.16})$$

for $c > 0$ and some $\epsilon_2 > 0$. Appendix B gives the expression for a finite time t_2 such that

$$|\theta + \dot{\theta}| = |\nu_1| \leq c + \epsilon_2 \quad \forall \quad t \geq t_2 \quad (\text{A.17})$$

Let us choose c such that

$$c + \epsilon_2 < \frac{\pi}{n+1} \quad (\text{A.18})$$

for some integer $n = 3, 4, \dots$ to be determined. From (A.4) and (A.5) it follows

$$\dot{\theta} = -\theta + \nu_1 \quad (\text{A.19})$$

Appendix C gives the expressions for a finite time t_3 such that

$$|\theta| \leq \frac{\pi}{n} \quad \forall \quad t \geq t_3 \quad (\text{A.20})$$

The integer $n = 3, 4, 5, \dots$ will be chosen later so that $(\tan \theta - \theta)$ is small enough to guarantee the stability of the overall system.

A.1.4 Thrust control input u

In this section we present the thrust control input u which is based on nonlinear compensation.

Notice that after time t_3 the value of θ in (A.20) is far away from the singular points $\frac{\pi}{2}$ and $-\frac{\pi}{2}$. We can then propose the following thrust control input u :

$$u = \begin{cases} \frac{[-2\dot{z} - (z - z^d)]m + mg}{\cos \theta} & |\theta| < \frac{\pi}{n} \\ mg & \frac{\pi}{n} \leq |\theta| \leq \frac{\pi}{3} \\ 0 & |\theta| > \frac{\pi}{3} \end{cases} \quad (\text{A.21})$$

where z^d is the constant desired altitude. The constant values for u above when $|\theta| > \frac{\pi}{n}$ do not play a roll in the convergence analysis. However they can be used to improve the performance of the algorithm in the presence of disturbances as will be shown in the numerical simulations.

Introducing (A.58) into (A.55) gives

$$\ddot{z} + 2\dot{z} + (z - z^d) = 0 \quad (\text{A.22})$$

or

$$\begin{bmatrix} \dot{z} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix} \begin{bmatrix} z - z^d \\ \dot{z} \end{bmatrix} \quad (\text{A.23})$$

A.1.5 Upperbound on the velocity \dot{x}

In this section it will be shown that the velocity \dot{x} will converge to a bounded interval.

Introducing (A.58) into (A.54) and (A.4) gives

$$\ddot{x}_1 = \frac{\ddot{x}}{g} = \tan \theta + e_1 \quad (\text{A.24})$$

where

$$e_1 = \frac{\tan \theta [-2\dot{z} - (z - z^d)]}{g} \quad (\text{A.25})$$

which can be rewritten as

$$\ddot{x}_1 = \theta + e_1 + e_2 \quad (\text{A.26})$$

where

$$e_2 = \tan \theta - \theta \quad (\text{A.27})$$

Thus, from (A.54) and (A.57), the subsystem for x , \dot{x} , θ and $\dot{\theta}$ is given by (see (A.4))

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 + e \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= \tau \end{aligned} \quad (\text{A.28})$$

with

$$e = e_1 + e_2 \quad (\text{A.29})$$

From (A.5)

$$\begin{aligned} \dot{\nu}_2 &= \dot{x}_2 + 2\dot{x}_3 + \dot{x}_4 \\ &= x_3 + e + 2x_4 + \tau \\ &= x_4 + \nu_1 + e + \tau \end{aligned} \quad (\text{A.30})$$

Taking into account (A.13) and (A.17), (A.8) reduces to

$$\tau = -x_4 - \nu_1 - \sigma_c(\nu_2 + \sigma_d(\nu_3))$$

Introducing the above into (A.30) leads to

$$\dot{\nu}_2 = -\sigma_c(\nu_2 + \sigma_d(\nu_3)) + e \quad (\text{A.31})$$

Choose d so that for some $\epsilon_3 > 0$

$$c \geq 3d + \epsilon_3 \quad (\text{A.32})$$

Appendix D gives the expression for a finite time t_4 such (see (A.25))

$$|e_1| \leq \frac{d}{6} \quad \forall \quad t \geq t_4 \quad (\text{A.33})$$

The integer $n \geq 3$ will be chosen such that for some $k > 0$ the following holds (see (A.20) and (A.27))

$$\begin{aligned} |e_2| &= |\tan \theta - \theta| \\ &\leq k|\theta| \\ &\leq k \frac{\pi}{n} \leq \frac{d}{6} \end{aligned} \quad (\text{A.34})$$

Introducing (A.33) and (A.34) into (A.29)

$$\begin{aligned} |e| &\leq |e_1| + |e_2| \\ &\leq \frac{d}{3} \end{aligned} \quad (\text{A.35})$$

Using (A.31), (A.32) and the above, the expression for a finite time t_5 is given in Appendix E such that

$$|\nu_2| \leq 2d + \epsilon_3 \quad \forall \quad t \geq t_5 \quad (\text{A.36})$$

Therefore (A.31) reduces to

$$\dot{\nu}_2 = -\nu_2 - \sigma_d(\nu_3) + e \quad (\text{A.37})$$

A.1.6 Convergence of the position x

In this section we will prove that the horizontal displacement x converges to zero.

From (A.5) and using (A.28) and (A.8)

$$\begin{aligned}
\dot{\nu}_3 &= \dot{x}_1 + 3\dot{x}_2 + 3\dot{x}_3 + \dot{x}_4 \\
&= x_2 + 3x_3 + 3e + 3x_4 + \tau \\
&= x_4 + \nu_1 + \nu_2 + 3e + \tau \\
&= -\sigma_d(\nu_3) + 3e
\end{aligned} \tag{A.38}$$

From (A.35) and (A.38) a finite time t_6 is given in Appendix F such that

$$|\nu_3| \leq d \quad \forall \quad t \geq t_6$$

From (A.4), (A.5) and (A.28) we get

$$\dot{x}_3 = -x_3 + \nu_1 \tag{A.39}$$

$$\dot{\nu}_1 = -\nu_1 - \nu_2 - \nu_3 \quad (\text{see (A.14) and (A.15)}) \tag{A.40}$$

$$\dot{\nu}_2 = -\nu_2 - \nu_3 + e \quad (\text{see (A.37)}) \tag{A.41}$$

$$\dot{\nu}_3 = -\nu_3 + 3e \quad (\text{see (A.38)}) \tag{A.42}$$

Define

$$X^T = \begin{bmatrix} x_3 & \nu_1 & \nu_2 & \nu_3 \end{bmatrix} \tag{A.43}$$

$$b^T = \begin{bmatrix} 0 & 0 & 1 & 3 \end{bmatrix} \tag{A.44}$$

and

$$A = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & -1 & -1 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & -1 \end{bmatrix} \tag{A.45}$$

Therefore (A.39) through (A.42) can be rewritten as

$$\dot{X} = AX + be \quad (\text{A.46})$$

The state space representation above has a system matrix A that is stable with 4 eigenvalues located at -1 . Recall that $e = e_1 + e_2$ (see A.29) where e_1 converges to zero, (see (A.22) and (A.25)). Furthermore $|e_2| \leq k|\theta| = k|x_3|$ for a value of “ k ” that has to be determined so that it satisfies (A.34) and is simultaneously small enough such that (A.46) remains stable.

The values for k and n such that (A.46) is a stable system could be obtained by using a Lyapunov function $X^T P X$ where $P > 0$ is the solution of the Lyapunov equation $A^T P + P A = -Q$ for some $Q > 0$. Nevertheless, for simplicity we will proceed as follows.

From (A.20) and (A.22) it follows that e_1 in (A.25) converges to zero. Since A in (A.46) is a stable matrix, the contribution of e_1 in X will also converge to zero. From (A.34) $|e_2| \leq k|\theta|$. Thus, following the procedure developed in Appendix C, we can prove using (A.42) that there exists a finite time t_7 such that

$$|\nu_3| \leq 3k|\theta| + \epsilon \quad \forall \quad t \geq t_7 \quad (\text{A.47})$$

for some $\epsilon > 0$.

Similarly, using (A.41), there exists a finite time t_8 such that

$$|\nu_2| \leq 4k|\theta| + 2\epsilon \quad \forall \quad t \geq t_8 \quad (\text{A.48})$$

Similarly, from (A.40) it follows that there exists t_9 such that

$$|\nu_1| \leq 7k|\theta| + 3\epsilon \quad \forall \quad t \geq t_9 \quad (\text{A.49})$$

Let us assume that k also satisfies

$$\mu^2 = 1 - 7k > 0 \quad (\text{A.50})$$

Finally from (A.4) and (A.39)

$$\dot{\theta} = -\theta + \nu_1 \quad (\text{A.51})$$

Let $V = \frac{1}{2}\theta^2$

then, using (A.49)

$$\begin{aligned} \dot{V} &= \theta\dot{\theta} = \theta(-\theta + \nu_1) \\ &\leq -\theta^2 + 7k\theta^2 + 3|\theta|\epsilon \\ &\leq -\mu^2\theta^2 + 2\left(\frac{\mu|\theta|}{2}\right)\left(\frac{3\epsilon}{\mu}\right) \\ &\leq -\frac{3}{4}\mu^2\theta^2 + \frac{9\epsilon^2}{\mu^2} \\ &\leq -\frac{3}{2}\mu^2V + \frac{9\epsilon^2}{\mu^2} \end{aligned} \quad (\text{A.52})$$

Since ϵ can be chosen arbitrarily small it follows from the above that $V \rightarrow 0$ and therefore $\theta \rightarrow 0$. From (A.51) and (A.4) we conclude that $x_3 \rightarrow 0$ and $x_4 \rightarrow 0$. From (A.47)-(A.49) it follows that $\nu_1 \rightarrow 0$, $\nu_2 \rightarrow 0$ and $\nu_3 \rightarrow 0$. From (A.5), $x_2 \rightarrow 0$ and $x_1 \rightarrow 0$. From (A.4) it finally follows $x \rightarrow 0$, $\dot{x} \rightarrow 0$, $\theta \rightarrow 0$, $\dot{\theta} \rightarrow 0$.

The constraints required for n and k are given in (A.34) and (A.50). Appendix F proves that for $n = 9$ and $k = 0.0424$ both constraints are verified. Notice that $\frac{\pi}{9} = 20^\circ$.

Finally let us summarize the constraints on the saturation upperbounds a , b , c and d . See (A.12), (A.16), (A.18), (A.32) and (A.34)

$$\begin{aligned} a &\geq 2b + \epsilon_1 \\ b &\geq 2c + \epsilon_2 \\ 0.1\pi &\geq c + \epsilon_2 \\ &\geq 3d + \epsilon_2 + \epsilon_3 \\ &\geq 0.085\pi + \epsilon_2 + \epsilon_3 \end{aligned} \quad (\text{A.53})$$

The choice of the parameter a depends on the upperbound of the torque control input τ . In practice the higher the value of a the faster the angular rate $\dot{\theta}$ will decrease after a disturbance

and therefore the aerial vehicle attitude will be more robust to disturbances. Nevertheless, when the parameter a is chosen too high, it is common that mechanical vibrations appear. The parameters b , c and d are determined by the choice of the parameter a . Concerning the parameters ϵ_i , they have been introduced so that the corresponding state variables are smaller than a given bound in finite time. The higher the value of ϵ_i the shorter the corresponding time period t_i will be.

A.1.7 Numerical simulation results

In this section we will present a comparison between the proposed control algorithm and the control strategy introduced in [18]. We will also show the robustness properties of the proposed algorithm with respect to plant parameter changes. We will consider that the PVTOL is represented by the following equations

$$m\ddot{x} = u \sin \theta \quad (\text{A.54})$$

$$m\ddot{z} = u \cos \theta - mg \quad (\text{A.55})$$

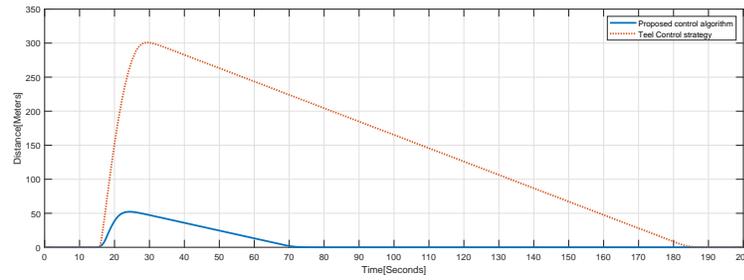
$$\dot{\theta} = \theta_1 + w_p \quad (\text{A.56})$$

$$\dot{\theta}_1 = k\tau \quad (\text{A.57})$$

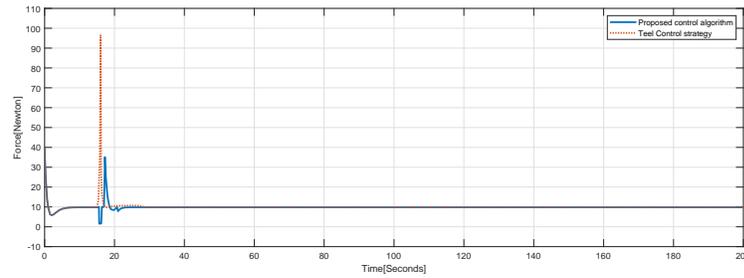
where w_p is a perturbation given as follows:

$$w_p = \begin{cases} 2.3 & 15s \leq t \leq 16s \\ 0 & otherwise \end{cases} \quad (\text{A.58})$$

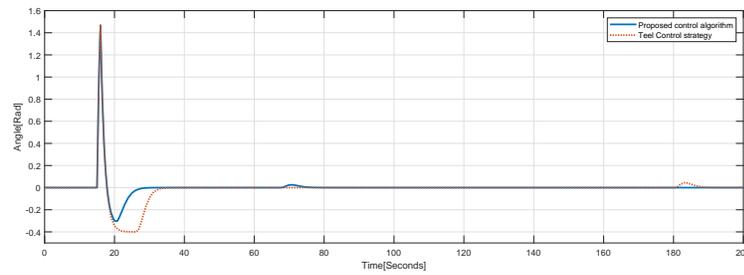
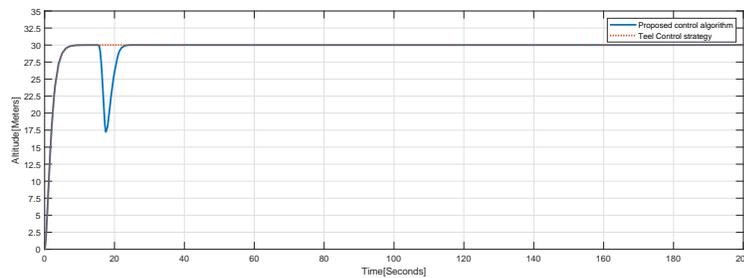
Figure A.2 presents the comparison between the control algorithm proposed in the present paper and the algorithm in [18]. This figure presents a) the displacement x , b) the control input u , c) the orientation angle θ , and d) the altitude z . The initial conditions have been chosen as: $z(0) = 0, \theta(0) = 0, x(0) = 0$.



(a) x-position



(b) u control input

(c) θ orientation angle

(d) z-position

Fig. A.2: Comparison between the proposed control algorithm and the original Teel's strategy when a disturbance is introduced at time 15 sec.

We have introduced a disturbance at time 15s in the orientation rate $\dot{\theta}$. As can be seen in figure A.2 a) the disturbance produces a much smaller displacement x in the case of the proposed algorithm and the displacement x takes much less time to come back to the origin. Figure A.2 b) shows that the thrust input is smaller for the proposed algorithm. The angular displacement in figure A.2 c) is just a little smaller for our algorithm. Figure A.2 d) shows that the price to be paid for obtaining the above advantages with the proposed algorithm is to reduce the altitude of the PVTOL during a few seconds.

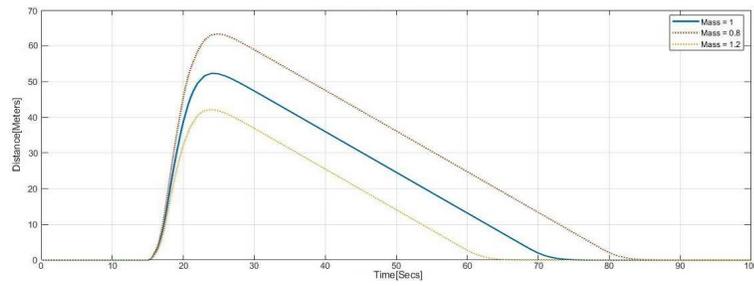
Concerning the robustness of the proposed algorithm with respect to changes in the plant parameter we have considered two different plant parameters: the mass m and the torque control gain k .

We have used the same initial conditions as before as well as the same disturbance in the angular orientation rate $\dot{\theta}$. Figure A.3 shows the performance of the proposed controller when the plant mass m increases or decreases 20%. As can be seen a change in the plant m produces an error in between the altitude and the desired altitude. The rest of the closed-loop system behaves with no significant changes.

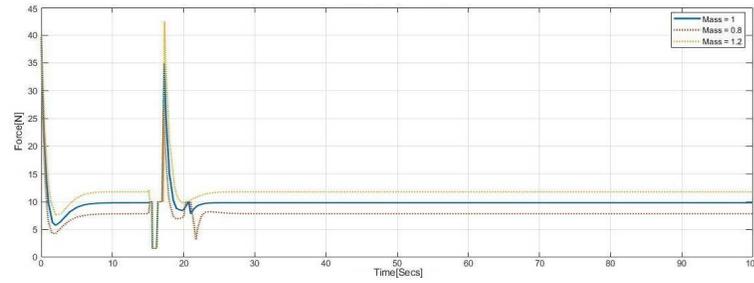
Figure A.4 shows the performance of the proposed controller when the torque gain k increases or decreases 20%. As can be seen when k increases the closed loop system behaves well. However, when k decreases the system requires a larger control input and more time to converge.

A.1.8 Conclusion

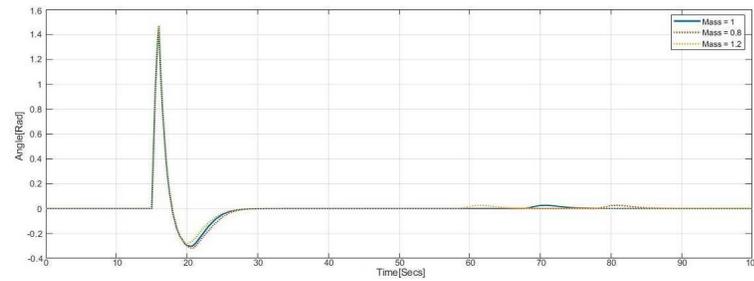
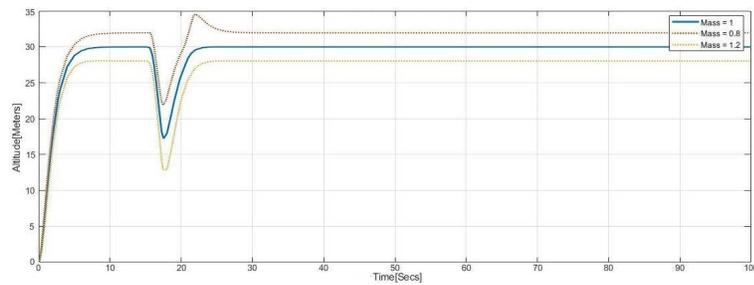
This paper addressed the problem of obtaining a simple control algorithm for the PVTOL. The nonlinearity of the PVTOL has been an obstacle for obtaining stable controllers. Since the PVTOL can be approximated by four integrators in cascade, the nested saturations controller proposed in [33] has been currently used to control the PVTOL. However no proof of stability of the closed loop has been obtained in the past. This paper has shown that a particular choice of the amplitudes of the saturations involved in the technique in [33] allows



(a) x-position

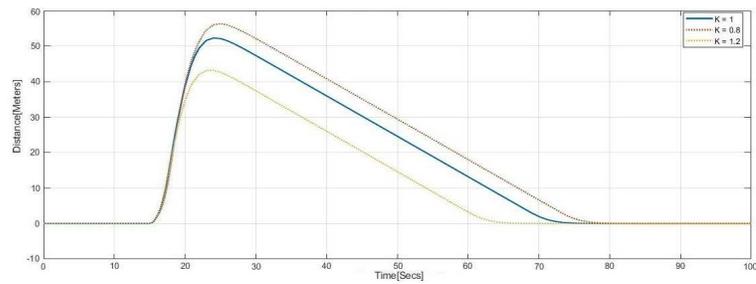


(b) u control input

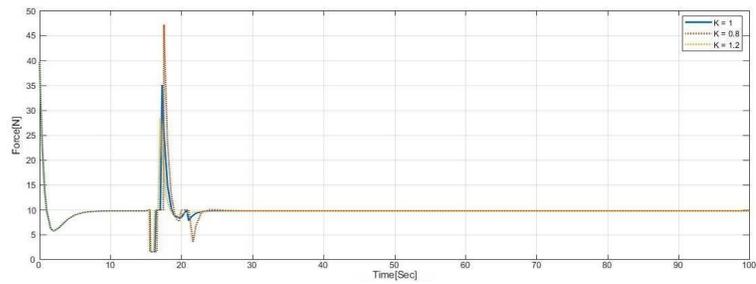
(c) θ orientation angle

(d) z-position

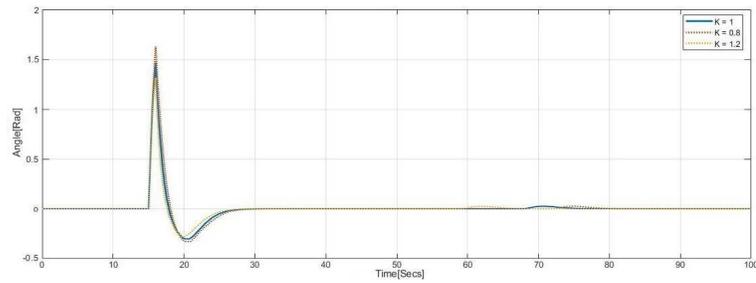
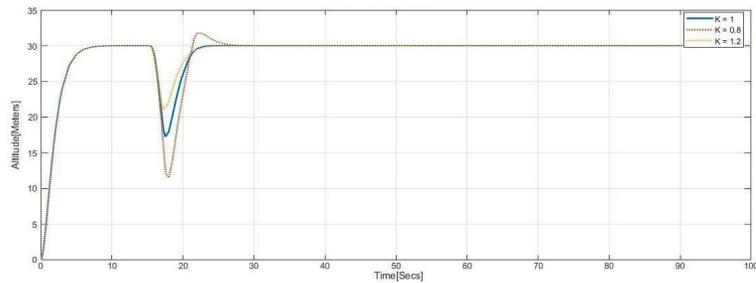
Fig. A.3: Robustness of the proposed algorithm with respect to variations of the mass value m



(a) x-position



(b) u control input

(c) θ orientation angle

(d) z-position

Fig. A.4: Robustness of the proposed algorithm with respect to variations of the gain value k .

for the proof of global stability of the closed loop. The performance of the proposed controller is shown in numerical simulations.

A.1.9 Bound for $x_4 = \dot{\theta}$

Let

$$V_1 = \frac{1}{2}x_4^2 \quad (\text{A.1})$$

Introducing (A.10) into above

$$\dot{V}_1 = x_4\dot{x}_4 = -x_4\sigma_a(x_4 + \sigma_b(w_1)) \quad (\text{A.2})$$

If

$$|x_4| > b + \epsilon_1 \quad (\text{A.3})$$

it follows that

$$\dot{V}_1 \leq -(b + \epsilon_1)\epsilon_1 \quad (\text{A.4})$$

or

$$\dot{V}_1 + (b + \epsilon_1)\epsilon_1 \leq 0 \quad (\text{A.5})$$

Integrating

$$V_1 - V_1(0) + (b + \epsilon_1)\epsilon_1 t \leq 0 \quad (\text{A.6})$$

or

$$V_1 \leq V_1(0) - (b + \epsilon_1)\epsilon_1 t \quad (\text{A.7})$$

Define t_1 such that the RHS of the above satisfies

$$V_1(0) - (b + \epsilon_1)\epsilon_1 t_1 = \frac{1}{2}(b + \epsilon_1)^2 \quad (\text{A.8})$$

thus

$$t_1 = \frac{V_1(0)}{(b + \epsilon_1)\epsilon_1} - \frac{(b + \epsilon_1)}{2\epsilon_1} \quad (\text{A.9})$$

Therefore

$$|x_4| \leq b + \epsilon_1 \quad \forall \quad t \geq t_1 \quad (\text{A.10})$$

then, for $t \geq t_1$, (A.10) reduces to

$$\dot{x}_4 = -x_4 - \sigma_b(w_1) \quad (\text{A.11})$$

thus

$$\dot{V}_1 = x_4 \dot{x}_4 = -x_4(x_4 + \sigma_b(w_1)) \quad (\text{A.12})$$

Notice that when $|x_4| > b + \epsilon_1$ it follows that $\dot{V}_1 < 0$, and V_1 will decrease. So if $|x_4| \leq b + \epsilon_1$ at some time, it will remain so from then on.

A.1.10 Bound for $\nu_1 = \theta + \dot{\theta}$

Let

$$V_2 = \frac{1}{2}\nu_1^2 \quad (\text{B.1})$$

$$\dot{V}_2 = \nu_1 \dot{\nu}_1 = -\nu_1 \sigma_b(\nu_1 + \sigma_c(w_2))$$

If $|\nu_1(t_1)| > c + \epsilon_2$ it follows from (A.14) that for $t \geq t_1$

$$\dot{V}_2 \leq -(c + \epsilon_2)\epsilon_2 \quad (\text{B.2})$$

or

$$\dot{V}_2 + (c + \epsilon_2)\epsilon_2 \leq 0 \quad (\text{B.3})$$

Integrating from $t = t_1$

$$V_2 - V_2(t_1) + (c + \epsilon_2)\epsilon_2(t - t_1) \leq 0 \quad (\text{B.4})$$

or

$$V_2 \leq V_2(t_1) - (c + \epsilon_2)\epsilon_2(t - t_1) \quad (\text{B.5})$$

Define t_2 such that the RHS of the above satisfies

$$V_2(t_1) - (c + \epsilon_2)\epsilon_2(t_2 - t_1) = \frac{(c + \epsilon_2)^2}{2} \quad (\text{B.6})$$

thus

$$t_2 = t_1 + \frac{V_2(t_1)}{(c + \epsilon_2)\epsilon_2} - \frac{(c + \epsilon_2)}{2\epsilon_2} \quad (\text{B.7})$$

Therefore

$$|\nu_1| \leq c + \epsilon_2 \quad \forall \quad t \geq t_2 \quad (\text{B.8})$$

A.1.11 Bound for θ

From (A.19)

$$\dot{\theta} = -\theta + \nu_1 \quad (\text{C.1})$$

Let

$$V_3 = \frac{1}{2}\theta^2 \quad (\text{C.2})$$

Introducing (B.8)

$$\begin{aligned}\dot{V}_3 &= \theta\dot{\theta} = \theta[-\theta + \nu_1] \\ &= -\theta^2 + \theta\nu_1 \leq -\theta^2 + |\theta|(c + \epsilon_2)\end{aligned}\tag{C.3}$$

From (A.18)

$$(c + \epsilon_2) \leq \frac{\pi}{n+1}\tag{C.4}$$

Assume that $|\theta| \geq \frac{\pi}{n}$, then

$$\begin{aligned}\dot{V}_3 &\leq |\theta|\left[\frac{\pi}{n+1} - |\theta|\right] \leq |\theta|\left[\frac{\pi}{n+1} - \frac{\pi}{n}\right] = |\theta|\left[-\frac{\pi}{n(n+1)}\right] \\ &\leq -|\theta|\frac{\pi}{n(n+1)} \leq -\frac{\pi^2}{n^2(n+1)}\end{aligned}\tag{C.5}$$

or

$$\dot{V}_3 + \frac{\pi^2}{n^2(n+1)} \leq 0\tag{C.6}$$

Integrating from $t = t_2$

$$V_3 - V_3(t_2) + \frac{\pi^2}{n^2(n+1)}(t - t_2) \leq 0\tag{C.7}$$

or

$$V_3 \leq V_3(t_2) - \frac{\pi^2}{n^2(n+1)}(t - t_2)\tag{C.8}$$

Let t_3 be such that the RHS of the above satisfies

$$V_3(t_2) - \frac{\pi^2}{n^2(n+1)}(t_3 - t_2) = \frac{1}{2}\left(\frac{\pi}{n}\right)^2\tag{C.9}$$

Then

$$t_3 = t_2 + \frac{V_3(t_2)n^2(n+1)}{\pi^2} - \frac{(n+1)}{2}\tag{C.10}$$

Thus

$$|\theta| \leq \frac{\pi}{n} \quad \forall \quad t \geq t_3 \quad (\text{C.11})$$

A.1.12 Bound for e_1 in (A.25)

Consider (A.23) and define

$$Z^T = \begin{bmatrix} z - z^d & \dot{z} \end{bmatrix} \quad \text{and} \quad \bar{A} = \begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix} \quad (\text{D.1})$$

then the Lyapunov equation

$$\bar{A}^T P + P \bar{A} = -Q \quad (\text{D.2})$$

holds for

$$P = \begin{bmatrix} 3 & 1 \\ 1 & 1 \end{bmatrix} > 0 \quad \text{and} \quad Q = 2I_2 \quad (\text{D.3})$$

Let

$$V = Z^T P Z \quad (\text{D.4})$$

then using (A.23) and (D.1) - (D.4)

$$\begin{aligned} \dot{V} &= Z^T \bar{A}^T P Z + Z^T P \bar{A} Z \\ &= -Z^T Q Z = -2\|Z\|^2 \\ &\leq -\alpha V \end{aligned} \quad (\text{D.5})$$

with

$$\alpha = \frac{2}{\lambda_{\max} P} \quad (\text{D.6})$$

thus

$$\dot{V} \leq -\alpha V \quad (\text{D.7})$$

Let

$$\dot{W} = -\alpha W \quad (\text{D.8})$$

with

$$W(0) = V(0) \quad (\text{D.9})$$

Since $\dot{V} \leq \dot{W}$

$$V \leq W = V(0)e^{-\alpha t} \quad (\text{D.10})$$

From (A.10) and (A.25) it follows that

$$|e_1|^2 \leq k_1^2(|\dot{z}|^2 + |z - z_d|^2) \quad (\text{D.11})$$

with

$$k_1 = \frac{6 \tan(\frac{\pi}{3})}{g} \quad (\text{D.12})$$

From (D.3) and (D.4)

$$\begin{aligned} |\dot{z}|^2 + |z - z_d|^2 &\leq \frac{Z^T P Z}{\lambda_{\min} P} \\ &= \frac{V}{\lambda_{\min} P} \\ &\leq \frac{W}{\lambda_{\min} P} \\ &\leq \frac{V(0)e^{-\alpha t}}{\lambda_{\min} P} \end{aligned} \quad (\text{D.13})$$

Let $t = t_4$ be such that

$$e^{-\alpha t_4} = \frac{\lambda_{\min} P}{V(0)k_1^2} \left(\frac{d}{6}\right)^2 \quad (\text{D.14})$$

Introducing the above into (D.13) and (D.11) leads to (A.33), i.e.

$$|e_1| \leq \frac{d}{6} \quad (\text{D.15})$$

A.1.13 Bound for ν_2

Define

$$V = \frac{1}{2}\nu_2^2 \quad (\text{E.1})$$

then using (A.31)

$$\begin{aligned} \dot{V} &= \nu_2 \dot{\nu}_2 \\ &= -\nu_2[\sigma_c(\nu_2 + \sigma_d(\nu_3)) + e] \end{aligned} \quad (\text{E.2})$$

If

$$|\nu_2| \geq 2d + \epsilon_3 \quad (\text{E.3})$$

then from (A.32) and (A.35)

$$\dot{V} \leq -(2d + \epsilon_3)\left(\frac{2d}{3} + \epsilon_3\right) \quad (\text{E.4})$$

Integrating and proceeding as in appendices A and B we conclude that there exists a finite time t_5 such that (A.36) is satisfied.

A.1.14 Choice of k and n

Consider $n = 9$. From (A.34) we have that for $\bar{\theta} = \frac{\pi}{9} = 0.3491 = 20^\circ$

$$\begin{aligned} |e_2| &= \tan \frac{\pi}{9} - \frac{\pi}{9} \\ &= 0.3639 - 0.3491 \\ &= 0.0148 \leq 0.3491k \end{aligned} \tag{F.1}$$

which implies $k \geq 0.0424$

On the other hand (A.50) requires

$$k < \frac{1}{7} = 0.1428 \tag{F.2}$$

Therefore $k = 0.0424$ verifies both constraints.

Notice that $\tan \theta - \theta$ is a strictly increasing function and the result will hold for values of $|\theta|$ smaller than $\frac{\pi}{9}$.

BIBLIOGRAFÍA

- [1] ALMEIDA, D., BROADBENT, E. N., ZAMBRANO, A. M. A., WILKINSON, B. E., FERREIRA, M. E., CHAZDON, R., MELI, P., GORGENS, E., SILVA, C. A., STARK, S. C., ET AL. Monitoring the structure of forest restoration plantations with a drone-lidar system. *International Journal of Applied Earth Observation and Geoinformation* 79 (2019), 192–198.
- [2] CHOLLET, F., ET AL. *Deep learning with Python*, vol. 361. Manning New York, 2018.
- [3] DAS, J., CROSS, G., QU, C., MAKINENI, A., TOKEKAR, P., MULGAONKAR, Y., AND KUMAR, V. Devices, systems, and methods for automated monitoring enabling precision agriculture. In *2015 IEEE International Conference on Automation Science and Engineering (CASE)* (2015), IEEE, pp. 462–469.
- [4] DONG, R., LI, W., FU, H., XIA, M., ZHENG, J., AND YU, L. Semantic segmentation based large-scale oil palm plantation detection using high-resolution satellite images. In *Automatic Target Recognition XXIX* (2019), vol. 10988, International Society for Optics and Photonics, p. 109880D.
- [5] DUFFY, J. P., CUNLIFFE, A. M., DEBELL, L., SANDBROOK, C., WICH, S. A., SHUTLER, J. D., MYERS-SMITH, I. H., VARELA, M. R., AND ANDERSON, K. Location, location, location: considerations when using lightweight drones in challenging environments. *Remote Sensing in Ecology and Conservation* 4, 1 (2018), 7–19.

-
- [6] FAN, Z., LU, J., GONG, M., XIE, H., AND GOODMAN, E. D. Automatic tobacco plant detection in uav images via deep neural networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 11, 3 (2018), 876–887.
- [7] FLORES, D., GONZÁLEZ-HERNÁNDEZ, I., LOZANO, R., VAZQUEZ-NICOLAS, J. M., AND HERNANDEZ TORAL, J. L. Automated agave detection and counting using a convolutional neural network and unmanned aerial systems. *Drones* 5, 1 (2021), 4.
- [8] GARCIA, O., FLORES, D., SANTOS, O., ROMERO, H., SALAZAR, S., AND LOZANO, R. Autonomous take-off and landing on a colored platform. In *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on* (2017), IEEE, pp. 877–884.
- [9] GÓMEZ-CANDÓN, D., DE CASTRO, A., AND LÓPEZ-GRANADOS, F. Assessing the accuracy of mosaics from unmanned aerial vehicle (uav) imagery for precision agriculture purposes in wheat. *Precision Agriculture* 15, 1 (2014), 44–56.
- [10] HASSAN, M. A., YANG, M., RASHEED, A., YANG, G., REYNOLDS, M., XIA, X., XIAO, Y., AND HE, Z. A rapid monitoring of ndvi across the wheat growth cycle for grain yield prediction using a multi-spectral uav platform. *Plant science* 282 (2019), 95–103.
- [11] HERWITZ, S., JOHNSON, L., DUNAGAN, S., HIGGINS, R., SULLIVAN, D., ZHENG, J., LOBITZ, B., LEUNG, J., GALLMEYER, B., AOYAGI, M., ET AL. Imaging from an unmanned aerial vehicle: agricultural surveillance and decision support. *Computers and electronics in agriculture* 44, 1 (2004), 49–61.
- [12] ITAKURA, K., AND HOSOI, F. Automatic tree detection from three-dimensional images reconstructed from 360 spherical camera using yolo v2. *Remote Sensing* 12, 6 (2020), 988.
- [13] KAMILARIS, A., AND PRENAFETA-BOLDÚ, F. X. Deep learning in agriculture: A survey. *Computers and electronics in agriculture* 147 (2018), 70–90.
-

-
- [14] KANG, H., AND CHEN, C. Fast implementation of real-time fruit detection in apple orchards using deep learning. *Computers and Electronics in Agriculture* 168 (2020), 105108.
- [15] KITANO, B. T., MENDES, C. C., GEUS, A. R., OLIVEIRA, H. C., AND SOUZA, J. R. Corn plant counting using deep learning and uav images. *IEEE Geoscience and Remote Sensing Letters* (2019).
- [16] LI, W., DONG, R., FU, H., ET AL. Large-scale oil palm tree detection from high-resolution satellite images using two-stage convolutional neural networks. *Remote Sensing* 11, 1 (2019), 11.
- [17] LI, W., FU, H., YU, L., AND CRACKNELL, A. Deep learning based oil palm tree detection and counting for high-resolution remote sensing images. *Remote Sensing* 9, 1 (2016), 22.
- [18] LOZANO, R., C. P. . D. A. Global stabilization of the pvtol: real-time application to a mini-aircraft. *IFAC Proceedings Volumes* ((2004).), 37(21), 235–240.
- [19] LOZANO, R. *Unmanned aerial vehicles: Embedded control*. John Wiley & Sons, 2013.
- [20] MEIER, L., TANSKANEN, P., FRAUNDORFER, F., AND POLLEFEYS, M. Pixhawk: A system for autonomous flight using onboard computer vision. In *Robotics and automation (ICRA), 2011 IEEE international conference on* (2011), IEEE, pp. 2992–2997.
- [21] MINI COMPUTADORA ODROID XU4. <https://magazine.odroid.com/odroid-xu4/>, 2017.
- [22] NEGRETE, J. C., ET AL. Precision agriculture in mexico; current status and perspectives. *International Journal of Horticulture* 7 (2017).
- [23] NORASMA, C., FADZILAH, M., ROSLIN, N., ZANARIAH, Z., TARMIDI, Z., AND CANDRA, F. Unmanned aerial vehicle applications in agriculture. In *IOP Conference Series: Materials Science and Engineering* (2019), vol. 506, IOP Publishing, p. 012063.

- [24] PIXHAWK. <https://pixhawk.org>, Last modification, 2016/12/30 10:35.
- [25] PUTRA, B. T. W., SONI, P., MARHAENANTO, B., HARSONO, S. S., FOUNTAS, S., ET AL. Using information from images for plantation monitoring: A review of solutions for smallholders. *Information Processing in Agriculture* (2019).
- [26] QURESHI, W., PAYNE, A., WALSH, K., LINKER, R., COHEN, O., AND DAILEY, M. Machine vision for counting fruit on mango tree canopies. *Precision Agriculture* 18, 2 (2017), 224–244.
- [27] RAHNEMOONFAR, M., AND SHEPPARD, C. Deep count: fruit counting based on deep simulated learning. *Sensors* 17, 4 (2017), 905.
- [28] ROKHMANA, C. A. The potential of uav-based remote sensing for supporting precision agriculture in indonesia. *Procedia Environmental Sciences* 24 (2015), 245–253.
- [29] ROMERO-TRIGUEROS, C., NORTES, P. A., ALARCÓN, J. J., HUNINK, J. E., PARRA, M., CONTRERAS, S., DROOGERS, P., AND NICOLÁS, E. Effects of saline reclaimed waters and deficit irrigation on citrus physiology assessed by uav remote sensing. *Agricultural water management* 183 (2017), 60–69.
- [30] RPLIDAR A2 LASER RANGE SCANNER. <https://www.slamtec.com/en/Lidar/A2>, 2016.
- [31] SAARI, H., PELLIKKA, I., PESONEN, L., TUOMINEN, S., HEIKKILÄ, J., HOLMLUND, C., MÄKYNEN, J., OJALA, K., AND ANTILA, T. Unmanned aerial vehicle (uav) operated spectral camera system for forest and agriculture applications. In *Remote Sensing for Agriculture, Ecosystems, and Hydrology XIII* (2011), vol. 8174, International Society for Optics and Photonics, p. 81740H.
- [32] SEVERTSON, D., CALLOW, N., FLOWER, K., NEUHAUS, A., OLEJNIK, M., AND NANSEN, C. Unmanned aerial vehicle canopy reflectance data detects potassium defi-

- ciency and green peach aphid susceptibility in canola. *Precision Agriculture* 17, 6 (2016), 659–677.
- [33] TEEL, A. R. Global stabilization and restricted tracking for multiple integrators with bounded controls. *Systems & control letters* 18, 3 (1992), 165–171.
- [34] TORRES, C. L. F., OLVERA-VARGAS, L. A., GÓMEZ, J. S., AND CONTRERAS-MEDINA, D. I. Discovering innovation opportunities based on seci model: reconfiguring knowledge dynamics of the agricultural artisan production of agave-mezcal, using emerging technologies. *Journal of Knowledge Management* (2020).
- [35] TORRES-SÁNCHEZ, J., LÓPEZ-GRANADOS, F., SERRANO, N., ARQUERO, O., AND PEÑA, J. M. High-throughput 3-d monitoring of agricultural-tree plantations with unmanned aerial vehicle (uav) technology. *PloS one* 10, 6 (2015), e0130479.
- [36] VEGA, F. A., RAMIREZ, F. C., SAIZ, M. P., AND ROSUA, F. O. Multi-temporal imaging using an unmanned aerial vehicle for monitoring a sunflower crop. *Biosystems Engineering* 132 (2015), 19–27.
- [37] WALLACE, L., LUCIEER, A., WATSON, C., AND TURNER, D. Development of a uav-lidar system with application to forest inventory. *Remote Sensing* 4, 6 (2012), 1519–1543.
- [38] WANG, Y., ZHU, X., AND WU, B. Automatic detection of individual oil palm trees from uav images using hog features and an svm classifier. *International Journal of Remote Sensing* 40, 19 (2019), 7356–7370.
- [39] XIA, M., LI, W., FU, H., YU, L., DONG, R., AND ZHENG, J. Fast and robust detection of oil palm trees using high-resolution remote sensing images. In *Automatic Target Recognition XXIX* (2019), vol. 10988, International Society for Optics and Photonics, p. 109880C.

-
- [40] XIANG, H., AND TIAN, L. Development of a low-cost agricultural remote sensing system based on an autonomous unmanned aerial vehicle (uav). *Biosystems engineering* 108, 2 (2011), 174–190.
- [41] YANG, Q., SHI, L., HAN, J., ZHA, Y., AND ZHU, P. Deep convolutional neural networks for rice grain yield estimation at the ripening stage using uav-based remotely sensed images. *Field Crops Research* 235 (2019), 142–153.
- [42] ZHENG, Y.-Y., KONG, J.-L., JIN, X.-B., WANG, X.-Y., AND ZUO, M. Cropdeep: The crop vision dataset for deep-learning-based classification and detection in precision agriculture. *Sensors* 19, 5 (2019), 1058.
- [43] ZHONG, Y., GAO, J., LEI, Q., AND ZHOU, Y. A vision-based counting and recognition system for flying insects in intelligent agriculture. *Sensors* 18, 5 (2018), 1489.
- [44] ZORTEA, M., NERY, M., RUGA, B., CARVALHO, L. B., AND BASTOS, A. C. Oil-palm tree detection in aerial images combining deep learning classifiers. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium* (2018), IEEE, pp. 657–660.
- [45] ZOU, Z., SHI, Z., GUO, Y., AND YE, J. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055* (2019).