EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

CONACYT

TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Ciudad Guzmán

INSTITUTO TECNOLÓGICO DE CD. GUZMÁN

TESIS

MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

TEMA:

CONTEO DE PLANTAS DE AGAVE USANDO REDES NEURONALES CONVOLUCIONALES.

QUE PARA OBTENER EL TÍTULO DE:
MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:
ING. OMAR HERNÁNDEZ CALVARIO

DIRECTORES:
DRA. MARIA GUADALUPE SÁNCHEZ CERVANTES
DR. HIMER ÁVILA GEORGE

CD. GUZMÁN JALISCO, MÉXICO, AGOSTO DE 2022

Ciudad Guzmán, 15/agosto/2022
DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN
Asunto: *Autorización de impresión de Tesis*

ING. OMAR HERNÁNDEZ CALVARIO
CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS DE LA COMPUTACIÓN
PRESENTE

De acuerdo con los Lineamientos para la Operación de los Estudios de Posgrado en el Tecnológico Nacional de México y las disposiciones en este Instituto, habiendo cumplido con todas las indicaciones que la Comisión Revisora realizó con respecto a su Trabajo de Tesis titulado **"Conteo de plantas de agave usando redes neuronales convolucionales"**, esta subdirección académica de este Instituto, concede la Autorización para que proceda a la impresión del mismo.

Sin otro particular, quedo de Usted.

A T E N T A M E N T E
Excelencia en Educación Tecnológica®
"2022, AÑO DEL CINCUENTA ANIVERSARIO DEL INSTITUTO TECNOLÓGICO DE CIUDAD GUZMÁN"

CARLOS RUVALCABA MÁRQUEZ
SUBDIRECTOR ACADÉMICO

EDUCACION
TECNOLOGICO NACIONAL DE MEXICO
INSTITUTO TECNOLÓGICO
DE CIUDAD GUZMÁN
SUBDIRECCIÓN ACADÉMICA

ccp. Archivo
CRM/MGSS/megg

Resumen

Para la industria tequilera, es muy importante conocer la cantidad de materia prima (agaves) con la que cuentan para poder estimar su producción y tomar decisiones en cuanto a los compromisos de venta para los próximos años. Cada planta de agave es muy valiosa, para que se puedan cosechar pasan de 4 a 7 años. Por lo tanto, el monitoreo e inventario de los cultivos de agave son actividades muy importantes. Dadas las grandes extensiones de terreno, es muy tedioso y tardado llevar un inventario de forma manual de los agaves. Los agricultores suelen saber cuántas plantas sembraron, porque pagaron por cada una de las plantas, pero desconocen cuantas siguen vivas. Cuando el agave se acerca a su etapa de madurez, se presenta el problema del robo de plantas. Se propone un algoritmo basado en redes neuronales convolucionales que recibe como entrada una imagen de un cultivo de plantas de agave adquiridas desde un Vehículo Aéreo no Tripulado y el resultado esperado es la detección y conteo de cada una de las plantas de agave presentes.

Para propósitos de comparación se realizó la detección y conteo de plantas de agave con imágenes utilizadas en otras investigaciones, donde se presentaron metodologías diferentes a la propuesta en esta tesis. El algoritmo propuesto tiene un mejor rendimiento, alcanzando mejores resultados en la detección y conteo de plantas superando adversidades encontradas en investigaciones previas como las sombras y el traslape de plantas. La metodología propuesta soluciona la problemática encontrada en el conteo de plantas de agave.

Agradecimientos

A mis directores de tesis la Dra. María Guadalupe Sánchez Cervantes y el Dr. Himer Avila George, sus consejos fueron siempre útiles, su conocimientos y experiencia permitieron que esta tesis se desarrollara de manera satisfactoria. Aprecio el interés mostrado y el tiempo dedicado para mi formación académica.

A los docentes que forman parte del equipo de trabajo de la Maestría en Ciencias de la Computación del Instituto Tecnológico de Ciudad Guzmán en especial a los miembros de mi comité revisor, el Dr. Daniel Fajardo Delgado y la Mtra. María Eugenia Puga Nathal por el tiempo que se tomaron en revisar mi trabajo y sus comentarios tan importantes que ayudaron a mejorarlo.

Al Dr. Miguel Ángel de la Torre Gómora de la Universidad de Guadalajara, por su tiempo y apoyo en la sintonización de los algoritmos de aprendizaje profundo.

Al Mtro. Jesús Enrique Ponce Corona por compartir su trabajo e imágenes que me permitieron realizar experimentos y pruebas preliminares, su trabajo de investigación fue una referencia para el desarrollo de esta tesis.

A los doctores Victor Daniel Arechiga Cabrera y Abraham Jair Lopez Villalvazo de la Universidad de Guadalajara por brindar todas las facilidades para mi desarrollo profesional.

Al Ing. Ricardo Velasco Vazquez por su tiempo y apoyo en el preprocesamiento de imágenes y sus aportes en el etiquetados de regiones de interés.

Al Consejo Nacional de Ciencia y Tecnología por otorgarme la beca 690754 para el desarrollo de mis estudios de maestría.

A mis compañeros de maestría Frida Mayela Florián Pinto, Juan José Solórzano Carrillo y Rosario de la Luz Cantero Ramírez por su solidaridad ante la pandemia del COVID-19, les agradezco a cada uno de ustedes el apoyo, conocimiento y tiempo compartido, fue todo un placer trabajar con ustedes.

A Gabriela Sánchez Pérez por acompañarme en esta etapa y brindarme su apoyo que fue motivo de inspiración y motivación para finalizar esta tesis.

A mis padres por brindarme su apoyo incondicional. Les dedico a ustedes este logro en mi vida profesional amados padres, también es un logro de ustedes.

Índice general

Índice de figuras

Índice de tablas

Lista de siglas y acrónimos

AP Precisión promedio por sus siglas en inglés..

CNN Convolutional Neural Network por sus siglas en inglés.

CRT Consejo Regulador del Tequila.

CSP Red parcial de etapas cruzada por sus siglas en inglés..

FPN Características de la red piramidal por sus siglas en inglés..

GPS Global Position System por sus siglas en inglés..

GPU Unidad de procesamiento gráfico.

IOU Intersección sobre la unión por sus siglas en inglés..

mAP Precisión promedio media por sus siglas en inglés..

MAPE Error de porcentaje absoluto medio.

ML Machine learning por sus siglas en inglés.

NIR Infrarrojo cercano.

PANet Red de agregación de rutas por sus siglas en inglés..

PCA Análisis de componentes principales por sus siglas en inglés..

RGB Red, Green, Blue por sus siglas en inglés.

RPAS Remotely Piloted Aircraft System por sus siglas en inglés.

SAT Entrenamiento autoadversario por sus siglas en inglés..

SPP Agrupación de pirámide espacial por sus siglas en inglés..

SVM Máquinas de vectores de soporte.

UAV Unmanned Aerial Vehicle por sus siglas en inglés.

VANT Vehículo aéreo no tripulado por sus siglas en inglés.

YOLO You Only Look Once por sus siglas en inglés.

Capítulo 1

Introducción

El aprendizaje máquina (de las siglas en inglés, ML) es una disciplina de la Inteligencia Artificial que permite a los sistemas realizar tareas específicas de forma autónoma por medio de las redes neuronales convolucionales extrayendo las características de una imagen y luego usar dichas características para detectar o clasificar los objetos en una imagen.

La detección de objetos es un área de la visión por computadora que consta de dos grandes grupos de arquitecturas: detectores de una etapa (R-CNN, Fast R-CNN, Faster R-CNN, Mask R-CNN) detectores de dos etapas (YOLO, SSD). Actualmente la detección de objetos es muy utilizada en seguridad, vigilancia, vehículos autónomos, agricultura. El uso de herramientas tecnológicas como los vehículos aéreos no tripulados de acuerdo con Hernández (2021) permite resolver problemáticas en campos de cultivos de gran extensión, ya que con cámaras de alta definición e información geográfica pueden recorrer más de mil hectáreas en menos de una hora. Según el reporte del Servicio de Información Agroalimentaria (SIAP) de la Secretaría de Agricultura y Desarrollo Rural (Sader) del gobierno federal en el año 2021 el estado de Jalisco fue líder nacional en la producción agrícola de agave tequilero, maíz , chía, lima, arándano, frambuesa, tamarindo, caña de azúcar. El tequila es una de las bebidas más famosas a nivel nacional e internacional. Se obtiene de la planta de Agave tequilana Weber, mejor conocido como Agave azul. El Consejo Regulador del Tequila, A.C. (CRT) es la organización dedicada a inspeccionar y certificar el cumplimiento con la Norma Oficial del Tequila, así como a promover la calidad, la cultura y el prestigio de la bebida nacional.

Según las estadísticas del CRT (2022) en el año 2021 se produjeron 527 millones de litros de tequila, con una producción mayor respecto al año 2020 de 30 %. La demanda de materia prima para la industria del tequila va en aumento, en consecuencia la superficie

de cultivo del agave también ha crecido.

El CRT (2022) en su reglamento interno Capitulo II articulo 11 especifica que tiene la facultad de solicitar a sus asociados datos generales del agricultor, del predio, cantidad de plantas y estado fitosanitario. Esta información es muy importante para la industria tequilera porque les permite estimar un aproximado del rendimiento de sus plantaciones de agave para planificar y predecir la producción de la bebida. La problemática que presenta el esquema del cultivo de agave tequilana Weber variedad azul se refiere específicamente al control de las plantaciones, robo de plantas e inventariado de plantas, ya que se cosecha o se jima los 365 días del año.

La presente tesis tiene como objetivo general desarrollar un algoritmo basado en redes neuronales convolucionales para el conteo de plantas de agave utilizando imágenes aéreas capturadas desde un vehículo aéreo no tripulado.

1.1. Antecedentes

El uso de Vehículos Aéreos no Tripulados (por sus siglas en inglés, VANT) en la agricultura ha crecido exponencialmente con mayor aplicación en la adquisición de imágenes para conteo de plantas y análisis de diferentes cultivos. En la búsqueda sistemática se encontraron investigaciones recientes en el área de conteo de plantas, árboles, frutos y la detección de enfermedades y plagas en plantas.

En el área de investigación para la detección de frutos, Rahnemoonfar and Sheppard (2017) identificaron plantas de tomate y frutos maduros apoyados en aprendizaje profundo, usaron imágenes sintéticas para el entrenamiento de su algoritmo, esto beneficio la detección de frutos con oclusión y detección de tomates maduros. En el trabajo realizado por Wong et al. (2020) desarrollaron un método para la detección de la madurez del fruto de la palma, el algoritmo desarrollado se concentra en detectar la madurez de la fruta consta de dos funciones principales, segmentar la sección que consiste en el árbol de la imagen y clasificar la madurez de la fruta, el algoritmo propuesto tiene una precisión del 85 %. Sarabia et al. (2020) realizó el conteo de árboles procesando imágenes adquiridas de VANT apoyándose en la morfología matemática. Su conjunto de datos constaba de 50 imágenes, de las cuales 25 se utilizaron para entrenamiento, cuatro para validación y 21 para pruebas, la exactitud de su algoritmo fue de 99 %.

Las investigaciones desarrolladas en la detección de palmeras son muy similares al caso de estudio de esta tesis desde la perspectiva de la morfología de la planta de agave y las

palmeras. El algoritmo desarrollado por Zortea et al. (2018) se basa en el aprendizaje profundo y es capaz de construir un inventario de palmeras de aceite usando imágenes adquiridas por un VANT, combinando la salida de dos redes neuronales convolucionales independientes obteniendo una precisión de 91.2 % - 98.8 %.

Li et al. (2016) utilizaron imágenes satelitales para la detección y conteo de árboles de palma aceitera, usaron la red neuronal convolucional LeNEt basada en Tensorflow. Los principales parámetros de la red neuronal convolucional (por sus siglas en inglés, CNN) los ajustaron continuamente hasta que encontraron la mejor combinación de parámetros cuya precisión general fue la más alta en sus muestras de prueba. Para evaluar cuantitativamente el rendimiento del método propuesto calculan la precisión y sensibilidad. En la investigación desarrollada por Mubin et al. (2019) aplican dos CNN diferentes para detectar palma aceitera joven y madura por separado, utilizaron un enfoque de aprendizaje profundo para predecir y contar palmas aceiteras en imágenes satelitales, la arquitectura inicial desarrollada se basa en la CNN LeNet. La detección de plantas de tabaco fue implementada por medio de un algoritmo compuesto de tres etapas, utilizando el valor RGB en la imagen para reducir las falsas detecciones separando las zonas tabacaleras de las no tabacaleras en el trabajo desarrollado por Fan et al. (2018).

El aprendizaje profundo apoyado de las redes CNN de acuerdo con Lecun et al. (1998) es una alternativa eficiente en la detección de objetos, debido al desarrollo de algoritmos de código abierto para aprendizaje automático, aprendizaje por transferencia, equipos de cómputo con mayor potencia en procesamiento. Wu et al. (2019a) diseñaron un algoritmo capaz de detectar porciones enfermas de hojas de plantas de maíz a partir de imágenes adquiridas por un VANT, el modelo de CNN se utilizó para crear mapas de calor interpretables de las imágenes originales, indicando la ubicación de las supuestas lesiones. Su conjunto de datos contenía imágenes de hojas infectadas y no infectadas y se dividió aleatoriamente en conjuntos de entrenamiento, validación y prueba en una proporción de 70:15:15. Su modelo obtuvo una exactitud del 97 %.

Mathew and Mahesh (2021) desarrollaron un algoritmo para la detección temprana de manchas bacterianas en la planta de pimiento. El sistema propuesto se desarrolló con el algoritmo YOLO (por sus siglas en inglés, *You Only Look Once*) versión 5 para la detección de las manchas bacterianas en las hojas de la planta de pimiento a partir de imágenes adquiridas. Con el algoritmo YOLO versión 5 pudieron detectar pequeñas manchas en la planta con una precisión final de 99,75 %.

El conteo y reconocimiento de insectos es importante para el control de plagas. Zhong

et al. (2018) diseñaron un modelo capaz de contar y clasificar insectos. Usando el algoritmo YOLO para la detección y conteo el método de clasificación y el conteo fino lo realizaron basandose en el algoritmo de Máquinas de vectores de soporte (por sus siglas en inglés, SVM) utilizando características globales. La precisión de conteo promedio fue del 92,50 % y la precisión de clasificación promedio fue del 90,18 %.

En el conteo plantas de sorgo Ribera et al. (2017) utilizaron una CNN con un VANT para la adquisición de imágenes, la imagen del campo se estima mediante regresión en lugar de clasificación. Con esto evitan conocer el número máximo esperado de plantas. También describen un método para extraer imágenes de secciones o "parcelas"de una imagen ortorrectificada de todo el campo de cultivo. Estas imágenes son usadas para la formación y evaluación de la CNN, demuestran que pueden obtener un error de porcentaje absoluto medio (por sus siglas en inglés, MAPE) tan bajo como 6.7 % con la arquitectura de CNN Inception-v3.

En la detección y conteo de plántulas de arroz Wu et al. (2019b) diseñaron un algoritmo basado en el método de red combinada que estaba compuesta de dos redes neuronales totalmente convolucionales y una operación de combinación. La precisión promedio de la red combinada fue de un 11 % mayor que aplicando solamente una red básica, logrando una precisión de 93.35 %. Zhou et al. (2019) desarrollaron un algoritmo capaz de realizar la detección y conteo de panículas de arroz basado en redes mejoradas totalmente convolucionales, utilizaron un VANT equipado con una cámara RGB de alta definición para recolectar imágenes. El aprendizaje por transferencia fue utilizado en el entrenamiento con diferentes arquitecturas (R-FCN, AlexNet, Inception-V3, VggNet) identificando R-FCN como la mejor, con una precisión de 88 %.

Específicamente en el área de detección y conteo de plantas de agave mediante imágenes adquiridas por un VANT se encontró los trabajos que a continuación se presentan:

En el área de conteo de plantas de agave Calvario et al. (2020) desarrollaron un algoritmo supervisado basado en morfología matemática y cómputo paralelo para el análisis de las imágenes. Este algoritmo se aplicó en algunas imágenes adquiridas desde un VANT Phantom 4, DJI equipado con una cámara digital RGB, la altitud de vuelo fue de 60 metros. La morfología matemática es un método de procesamiento que no requirió una etapa de entrenamiento. El algoritmo diseñado es capaz de separar las plantas de agave y preservar los patrones principales. La precisión de este algoritmo osciló entre 83 % y 98 %. Recomiendan aplicar técnicas de aprendizaje supervisado como una red neuronal artificial y en particular, técnicas de aprendizaje profundo.

Corona (2019) en su trabajo de investigación desarrolló un algoritmo de aprendizaje no supervisado (Birch) para la identificación y el conteo de plantas de agave. Las imágenes fuerón adquiridas a través de un VANT, Parrot BlueGrass equipado con seis sensores, cámara RGB y cámara multiespectral, la altura de vuelo fue programada con 30 metros. El conjunto de datos usado tenía un total de 584 imágenes. La métrica utilizada para medir la eficiencia del algoritmo fue el coeficiente de Silhouette logrando un coeficiente Silhouette >0.5. En trabajos futuros sugiere incursionar en técnicas de aprendizaje profundo.

La investigación más reciente es realizada por Flores et al. (2021) donde presenta un método para el conteo de plantas de agave basado en imágenes adquiridas por un VANT. Utilizaron un VANT DJI Phantom 3 con una cámara FC300S, obtuvieron un ortomosaico de las plantaciones de agave, que después utilizaron para crear una base de datos, las imágenes fueron usadas para entrenar una CNN. La arquitectura de su CNN estaba compuesta por cuatro capas convolucionales y cuatro poolings. Para la detección de plantas de agave usaron un clasificador en cascada Haar basado en el concepto de características llamadas características similares a Haar. El conjunto de datos utilizado estaba compuesto de dos clases: (1) contenía imágenes de una sola planta de agave, (2) contenía extractos del área circundante donde no había plantas de agave como el suelo y otras plantas. La primera clase la nombraron positivos y la segunda negativos. Cada clase tenia 1000 imágenes. Utilizando el 70 % de las imágenes para el entrenamiento, 15 % para validación y 15 % para pruebas. La métrica utilizada para medir la eficiencia del algoritmo fue la precisión, obteniendo un 96 %.

1.2. Planteamiento del problema

En el año 2021 fueron necesarias 2,018.7 toneladas de plantas de agave para obtener una producción de 527 millones de litros de tequila, exportando 339.4 millones de litros de tequila según datos del Consejo Regulador del Tequila (por sus siglas, CRT).

Para la industria tequilera, es de vital importancia conocer la cantidad exacta de plantas de agave, el CRT tiene la facultad de solicitar a sus asociados información actualizada de los cultivos de agave para estimar la producción y tomar decisiones en cuanto a compromisos de venta para los próximos años. Cada planta de agave es muy valiosa, para que se puedan cosechar pasan de 4 a 7 años. Actualmente el inventario de plantas de agave se realiza de forma manual en la mayoría de los cultivos de agave.

Por lo tanto, el monitoreo e inventario de los cultivos de agave son actividades muy

importantes para la economía y la generación de empleos. Dadas las grandes extensiones de terreno, es muy tedioso y tardado llevar un inventario de forma manual de las plantas de agave. Los agricultores suelen saber cuántas plantas sembraron, porque pagaron por cada una de las plantas, pero desconocen la cantidad de plantas actuales. Cuando el agave se acerca a su etapa de madurez, se presenta el problema de robo de plantas. Por lo tanto, en esta tesis se propone el desarrollo de un algoritmo basado en redes neuronales convolucionales que recibe como entrada una imagen del cultivo de terreno adquirida desde un VANT y pueda identificar cada una de las plantas de agave ahí presentes y generar un inventario.

1.3. Justificación

La agricultura de precisión tiene como característica principal el uso de tecnologías aplicando un conjunto de técnicas que permiten la gestión localizada proporcionando información detallada sobre las características del cultivo, permitiendo optimizar su gestión (Villasenor, 2018).

Actualmente el monitoreo de cultivos es costoso porque se realiza de forma manual, esto genera que esta actividad se prolongue y aumente la probabilidad de errores. Algunos cultivos son analizados con imágenes satelitales, sin embargo, los satélites tienen la limitante de suministrar información temporal discontinua e información espectral de baja resolución. Una alternativa emergente la proporcionan los Vehículos Aéreos No Tripulados, estos vehículos son de gran utilidad para el monitoreo y supervisión de la superficie terrestre a través de imágenes georreferenciadas de alta resolución espacial, temporal y espectral de baja altura. Sin embargo, existen restricciones para su adopción debido a su costo inicial, entrenamiento y software requerido y regulaciones cada vez restrictivas para su uso (Bustamante, 2017).

En esta tesis se usan técnicas de aprendizaje profundo en imágenes adquiridas por un VANT de un cultivo de agave, se tomaron en cuenta las adversidades encontradas en investigaciones previas aplicadas a la detección y conteo de plantas de agave (detección de plantas de agave de diferentes tamaños, detección de plantas de agave con adversidades como sombras u objetos que no son plantas de agave). Así mismo, se desarrolló un algoritmo de aprendizaje profundo basado en redes neuronales convolucionales que automatice el conteo de las plantas de agave de un cultivo a partir de imágenes tomadas desde un VANT de tal manera que el conteo de las plantas de agave será obtenido de manera más eficiente,

económico y en un lapso de tiempo más breve que la forma tradicional. La investigación contribuye a la solución del tiempo de preprocesamiento de las imágenes, mejora la detección de plantas de agave con adversidades como sombras en cultivos de agave, maleza abundante, objetos que nos son plantas de agave, mejora los tiempos de entrenamiento del algoritmo, reduce costos computacionales porque no es necesario aplicar operaciones morfológicas ni tratamientos especiales a las imágenes en el preprocesamiento.

1.4. Objetivos

1.4.1. General

Desarrollar un algoritmo basado en redes neuronales convolucionales para el conteo de plantas de agave utilizando imágenes aéreas capturadas desde un vehículo aéreo no tripulado.

1.4.2. Objetivos Específicos

- Establecer un protocolo de vuelo que favorezca la captura de imágenes de agave.
- Crear una base de datos con imágenes de cultivos de agave.
- Crear un algoritmo basado en redes neuronales que permita la identificación y conteo de plantas de agave.

1.5. Hipótesis

Es posible realizar el conteo de plantas de agave usando imágenes capturadas desde un vehículo aéreo no tripulado y procesando esas imágenes utilizando redes neuronales convolucionales.

1.6. Estructura del trabajo

El presente documento de tesis está estructurado en cinco capítulos los cuales se describen a continuación. En el capítulo 2 se establece el marco teórico necesario para poder

entender el trabajo propuesto, se describen los conceptos que permiten proponer la solución al problema expuesto. En el capítulo 3 se expone la metodología propuesta que permite realizar el conteo de plantas de agave. Los resultados obtenidos se describen en el capítulo 4. La conclusión y recomendaciones a futuras investigaciones se presentan en el capítulo 5.

Capítulo 2

Marco Teórico

2.1. Agricultura de precisión

La agricultura de precisión de acuerdo con Villaseñor (2018) tiene como característica principal el uso de tecnologías aplicando un conjunto de técnicas que permiten la gestión localizada (satélites, sensores, imágenes y datos geográficos). Su éxito depende de tres elementos: información, tecnología y gestión. La agricultura de precisión proporciona información detallada sobre las características del cultivo, permitiendo optimizar la gestión de un cultivo.

Una de las herramientas utilizadas con mayor frecuencia en años recientes en la agricultura de precisión, es el VANT (dron). Mayor accesibilidad a estas herramientas y un aumento en la digitalización en la agricultura, han hecho que los VANT's se conviertan en aliados para la producción agrícola.

2.2. Los VANT en la agricultura

Los primeros usos de los VANT fueron con fines militares, pero en últimos años comenzaron a utilizarse en la agricultura para monitoreo de cultivos. Hernández (2021) indica que el 80 % a 90 % del mercado de aparatos no tripulados en la próxima década se utilizará en la agricultura.

Según Keller (2022) durante la última década, el uso de VANT se ha incrementado significativamente. Hoy en día, los VANT multirrotor (con capacidades de despegue y aterrizaje vertical) se están volviendo populares en diferentes sectores para varios usos,

como cartografía, topografía, teledetección, inspección, búsqueda y rescate, filmación, recreativas y deportes.

Los VANT tienen múltiples usos en la actualidad, como su capacidad para obtener información de manera remota. Las empresas aprovechan los algoritmos de reconocimiento de patrones mediante el aprendizaje profundo para procesar los datos capturados por los VANT. Muchas de estas tareas ya se realizan en la actualidad con vehículos aéreos piloteados por personas pero el futuro de la inteligencia artificial llevará a que las mismas tareas se puedan realizar de forma autónoma por los VANT.

La implementación de nuevas tecnologías para el manejo, monitoreo y control de los cultivos agrícolas en diferentes etapas de su desarrollo, mejoran la producción y disminuye los costos.

2.3. Teledetección

El vocablo teledetección según Sobrino (2000) deriva del francés "teledetection" traducción dada en 1967 al termino anglosajón "remote sensing" o percepción remota. Definir el concepto de teledetección no es sencillo ya que no existe una definición única, universalmente aceptada. En su más amplio sentido se entiende por teledetección o percepción remota "la adquisición de información sobre un objeto a distancia, esto es, sin que exista contacto material entre el objeto o sistema observado y el observador" (Sobrino, 2000). La teledetección parte del principio de la existencia de una perturbación (energía electromagnética, campos gravitacionales, ondas sísmicas. . .) que el sistema observado produce en el medio, la cual es registrada por el sistema receptor para, posteriormente, ser interpretada. La energía electromagnética se propaga en el vacío con una velocidad de $3x10^8$ m/s, ella se constituye en el campo de fuerza más útil para las actividades de teledetección, constituyendo un medio de traspaso de información de alta velocidad, entre las sustancias y objetos de interés y el equipo sensor.

El espectro electromagnético se compone de longitudes de onda de la energía electromagnética. El espectro se subdivide en algunos tipos de energía electromagnética como los rayos X, rayos ultravioletas (UV), visibles, infrarrojo (IR), microondas, y ondas de radio. Los tipos de energía electromagnética son categorizados por sus longitudes de onda, en el espectro electromagnético. Usualmente solo una pequeña porción o banda de todo el espectro es de interés en la percepción remota. Ya que la luz del sol es la fuente más común de energía usada en la percepción remota, las longitudes de onda predominantes

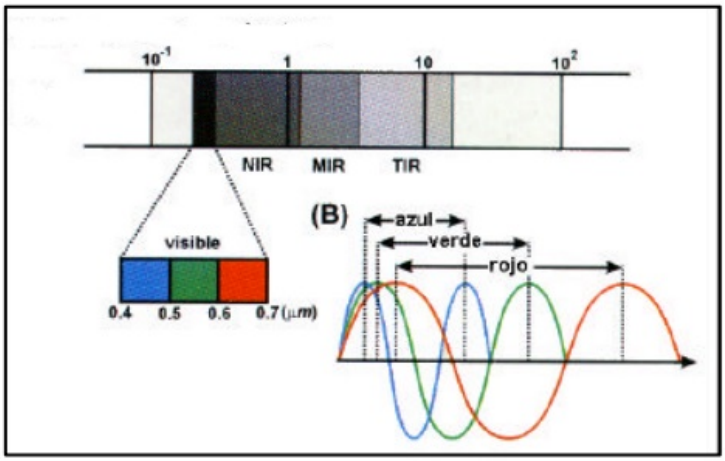


Figura 2.1: Espectro electromagnético. Fuente: Sobrino (2000).

en la luz solar son las más importantes. Para las aplicaciones agronómicas, la porción de interés es la que va desde el ultravioleta (UV) hasta el infrarrojo (IR).

Solo una parte del espectro electromagnético puede ser detectado por el ojo humano, se puede apreciar en la Figura 2.1. La banda del espectro visible se extiende de 0.4 a 0.7 micrómetros. La luz azul se encuentra cerca de la región de 0.4 µm y la luz roja cerca de los 0.7 µm. Sobre la región del rojo se encuentra la banda del infrarrojo cercano (NIR). No existe una distinción clara entre el NIR y el IR. Esta región del infrarrojo cercano, aunque es indetectable por el ojo humano, puede ser detectada por sensores artificiales y es muy importante en la percepción remota.

2.4. Espacios de color

Isaac Newton realizó experimentos donde a través de un prisma observó que la luz se descompone en una serie de colores (dispersión de la luz) explicó con su hipótesis que la luz del sol contenía rayos diversos con distinta refractividad y que se percibían como colores si se les observaba por separado. Cuando estos rayos se mezclaban, el aparato visual percibía colores distintos de los percibidos cuando estaban separados.

De acuerdo con Pérez (2009) la suma de todas las radiaciones del espectro visible en proporciones casi iguales da lugar a una sensación luminosa cuyo color es nulo (luz blanca), también llamada luz acromática. No sólo con muchas radiaciones electromagnéticas de distinta longitud de onda se puede producir esta luz acromática, sino también con la suma de tres radiaciones convenientemente elegidas. Esta luz blanca es la que permite ver

todos los colores que provienen de los objetos, ya que todos los cuerpos están constituidos por sustancias que absorben y reflejan las ondas electromagnéticas, es decir, absorben y reflejan colores.

Según Serrano (2014) el color es percepción por las longitudes de onda que llegan a la retina humana. La luz blanca se descompone en los colores del espectro cromático dependiendo de la longitud de onda, al ser percibida por el ojo este manda la información al cerebro. El espacio de color se refiere a la gama cromática que es capaz de mostrar un modelo o modo de color. El modo o modelo es la forma de representar el espacio de color que es lo que percibe el ojo humano. Percepción y representación, es la diferencia entre modo y espacio de color, cada espacio de color absoluto representa un espacio de color visible al ojo humano con diferentes valores, basados en los modos de color. La diferencia es que el ojo humano percibe mediante luz (como los dispositivos de captura digital, cámaras, escáneres, monitores). Según Serrano (2014) existen espacios de color absoluto:

- **Espacio de color CIE L*a*b*:** es el espacio de color que contiene una gama más amplia y la luminosidad es destacable.

- **Espacio sGRB:** es un espacio de color basado en RGB creado en cooperación por Hewlett-Packard y Microsoft Corporation. Fue aprobado por el W3C, Exif, Intel, Pantone, Corel.

Los espacios de color que a continuación se describen según Serrano (2014) son modos de color aunque también se les llama espacios no absolutos:

- **RGB:** comprendiendo el color rojo, verde y el azul. Usado para formatos gráficos e Internet. Diseñado para monitores CRT (tubos catódicos) de 8 bits de búfer como la mayoría de las imágenes. Los monitores LCD y dispositivos compensan esta diferencia por defecto.

- **HSB o HSV:** basado en el modo de color RGB ya que almacena la información del archivo en 3 canales, aunque con mayor tono, luminosidad y brillo. Es utilizado en compresores de video con perdida de calida ya que dicha perdida de color es casi inapresiable.

- **CMYK:** es el modo de color más restringido pero usado en impresión.

En las pruebas realizadas en esta tesis nos concentramos en los espacios de color RGB, HSV, CIE L*a*b*, por los beneficios para trabajar con diferentes bandas y resaltar el color característico azul de las plantas de agave.

2.4.1. Espacio de color RGB

El espacio RGB es el espacio de color más popular y con mayor aplicación en dispositivos para construir una imagen de color, por este motivo tiene relevancia en visión por computadora ya que trabajar con el mismo espacio de color con el que trabaja una cámara con la que se capturan las imágenes permite evitar la alteración de las propiedades del color durante el proceso de segmentación, propia de los errores de conversión y transformación, y por otro lado conseguir una mayor velocidad de segmentación por ahorro de esas operaciones de conversión.

El espacio RGB como se muestra en la Figura 2.2 se representa como un cubo dónde un color viene definido por la mezcla de valores de intensidad de tres colores primarios, rojo, verde y azul. Un color viene descrito por una tupla de 3 coordenadas en el cubo. El color negro se representa por (r=0, g=0, b=0) y el color blanco por (r=255, g=255, b=255). La gama acromática de escala de grises está representada por la diagonal del cubo (Gil, 2004).

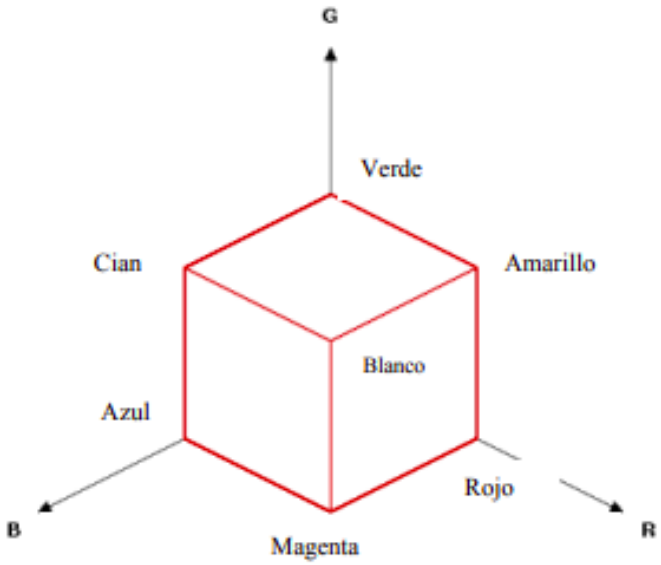


Figura 2.2: Representación del espacio RGB. Fuente: Gil et al. (2004).

2.4.2. Espacio de color HSV

El espacio HSV de acuerdo con Gil et al. (2004) representa uno de los espacios de coordenadas clásico e intuitivo. Su interpretación geométrica viene determinada por un cono de base quasi-hexagonal como se muestra en la Figura 2.3. Con esta representación del espacio de color, cada color trabaja con 3 componentes básicos: matiz, saturación y brillo. El matiz, h_{HSV} hace referencia al valor de cromaticidad o clase de color. La saturación, s_{HSV} se refiere a las longitudes de onda que se suman a la frecuencia del color y determina la cantidad de blanco que contiene un color. Contra menos saturado esté un color más cantidad de blanco y contra más saturado esté un color menor cantidad de blanco. En definitiva, la saturación representa la pureza e intensidad de un color. Así, la falta de saturación viene dada por la generatriz en la representación del cono HSV. Esa falta de saturación representa la gama de grises desde el blanco hasta el negro. La luminancia, v_{HSV} se corresponde con la apreciación subjetiva de claridad y oscuridad.

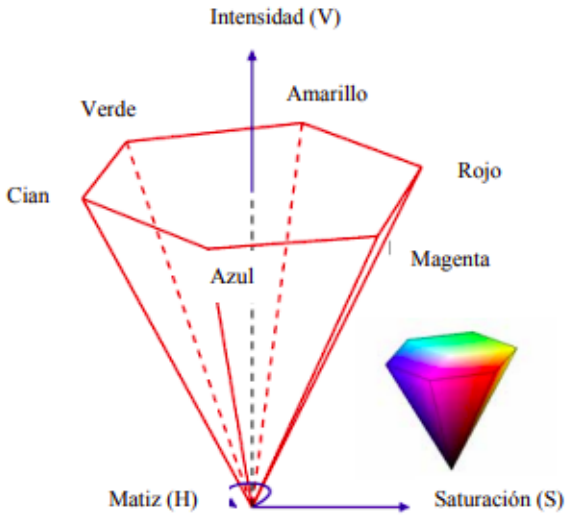


Figura 2.3: Representación del espacio HSV. Fuente: Gil et al. (2004).

2.4.3. Espacio de color CIE L*a*b*

El espacio de color L*a*b* según Minolta. (2022), también referido como CIE L*a*b*, es actualmente uno de los espacios de color más populares y uniformes usado para evaluar el color de un objeto. Este espacio de color es ampliamente usado porque correlaciona los

valores numéricos de color consistentemente con la percepción visual humana podemos apreciar el espacio de color de manera gráfica en la Figura 2.4 . El espacio de color CIE L*a*b* fue modelado en base a una teoría de color oponente que establece que dos colores no pueden ser rojo y verde al mismo tiempo o amarillo y azul al mismo tiempo. Como se muestra a continuación, L*indica la luminosidad y a* y b* son las coordenadas cromáticas. Los instrumentos de medición de color, incluyendo espectrofotómetros y colorímetros, pueden cuantificar éstos atributos de color fácilmente. Ellos determinan el color de un objeto dentro del espacio de color y muestran los valores para cada coordenada L*, a*, y b*.

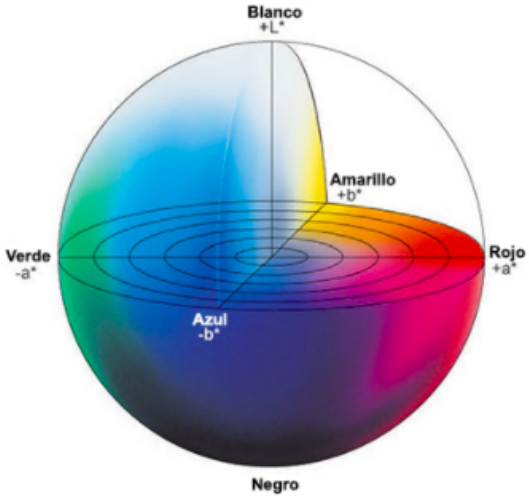


Figura 2.4: Representación del espacio CIE L*a*b*. Fuente: Minolta. (2022).

2.5. Operaciones Morfológicas

El procesamiento digital de imágenes es un área muy amplia, la morfologıa matemática es un enfoque con una teoría matemática que incluye conceptos geométricos, algebraicos y teoría de conjuntos. La importancia de sus propiedades algebraicas radica en el hecho de que éstas sirven para evaluar la validez de las transformaciones involucradas en esta teoría. Según Serra and Soille (2012) la morfología matemática es útil cuando se quiere describir y representar objetos de una imagen ya que considera principalmente el aspecto geométrico en su análisis. Con un sistema de operadores morfológicos puede formarse una composición que al actuar sobre objetos de formas complejas las descomponen en las

partes de interés.

Para de la Vega and de Teresa de Oteyza (2015) los métodos matemáticos usados en el procesamiento de imágenes dependen en general de las características propias de cada imagen. La morfología matemática es un enfoque geométrico-algebraico al procesamiento de imágenes digitales basado en los retículos completos de conjuntos y funciones. La idea fundamental del enfoque morfológico es transformar una imagen mediante diversos elementos estructurales en otra imagen que preserve las formas esenciales de los objetos contenidos en la imagen original y que facilite su análisis e interpretación.

De acuerdo con Zamora (2002) la morfologıa matemática es una técnica no lineal basada en teoría de conjuntos, teoría de retículos, topología y geometría integral. La idea principal del enfoque morfológico es trasformar una imagen en otra que sea mas apropiada para su análisis conservando las características esenciales de las formas contenidas en ella. Una imagen está compuesta de objetos o estructuras que tienen diferentes formas y niveles de intensidad las cuales son estudiadas por comparación con estructuras de referencia que tienen su propia forma y niveles de intensidad. Por lo tanto los objetos contenidos en una imagen se consideran como conjuntos cuyo análisis se lleva a cabo con simples operaciones entre ellos como intersección, unión y complemento. Actualmente, el ámbito y alcance de los procesamientos morfológicos es tan amplio como el propio procesamiento de imágenes. Se pueden encontrar aplicaciones tales como la segmentación, restauración, detección de bordes, aumento de contraste, análisis de texturas, compresión.

Los operadores morfológicos más conocidos y utilizados en el procesamiento digital de imágenes, destacan: la dilatación, la erosión, la cerradura y la apertura:

- **Dilatación:** consiste en el aumento de píxeles blancos en una imagen binaria, tomando como base la morfología de un elemento estructurante. Dicho elemento estructurante se representa a través de una matriz cuadrada conformada por unos y ceros.

- **Erosión:** la erosión consiste en la disminución de píxeles blancos en una imagen binaria, tomando como base la morfología de un elemento estructurante. Dicho elemento estructurante se representa a través de una matriz cuadrada conformada por unos y ceros.

- **Apertura:** una operación de apertura se encarga, por lo general, de suavizar el contorno de objetos, eliminando protuberancias y abriendo canales. De esta manera,

su realización se obtiene al aplicar una operación morfológica de erosión seguida de una dilatación, y utilizando siempre el mismo elemento estructurante.

- **Cerradura:** en cuanto a la cerradura, es otro operador morfológico que tiene la función de eliminar pequeños orificios, fusionar brechas y alargar pequeñas entradas. Asimismo, la obtención de este operador se realiza aplicando una operación morfológica de dilatación seguida por una erosión, utilizando siempre el mismo elemento estructurante.

Es importante mencionar que con la combinación y transformaciones de la dilatación, apertura, cerradura es posible generar nuevos operadores.

2.6. Inteligencia artificial

La Inteligencia artificial según Rouhiainen (2018) es la capacidad que tienen las máquinas para usar algoritmos, aprender de los datos y utilizar el aprendizaje en la toma de decisiones tal y como lo haría un ser humano. Los dispositivos basados en inteligencia artificial pueden analizar grandes volúmenes de información a la vez. De acuerdo con Pino et al. (2001) la inteligencia artificial como ciencia y tecnología ha ido acumulando conocimiento sobre como emular algunas capacidades del ser humano para exhibir comportamientos inteligentes se han desarrollado sistemas cada vez más perfeccionados que reproducen parcialmente dichas capacidades. En ciencias de la computación, una máquina inteligente es un agente flexible que percibe su entorno y lleva a cabo acciones que maximizan sus posibilidades de éxito en algún objetivo.

Galipienso (2003) afirma que existen diversas áreas donde la inteligencia artificial se presenta en mayor o menor medida. A continuación se comentan brevemente algunas de estas áreas:

- **Tratamiento de lenguajes naturales:** en esta área se pueden categorizar aplicaciones que realicen traducciones entre idiomas, interfaces humano/máquina que permitan interrogar una base de datos o dar ordenes a un sistema operativo.

- **Sistemas expertos:** en esta área están englobados aquellos sistemas donde la experiencia de personal cualificado se incorpora a dichos sistemas para conseguir deducciones mas cercanas a la realidad.

- **Robótica:** navegación de robots móviles, control de brazos de robot, ensamblaje de piezas, etc.

- **Aprendizaje:** modelización de conductas para su posterior implantación en computadoras.

- **Problemas de percepción: visión y habla:** reconocimiento y clasificación de objetos y del habla, detección de defectos en piezas por medio de visión artificial, apoyo en diagnósticos médicos.

2.6.1. Visión por computadora

La visión por computadora es una parte de la inteligencia artificial es definida por Mínguez (2021) como la disciplina científica formada por un conjunto de técnicas que permiten la captura, procesamiento y análisis de imágenes, el objetivo de la visión por computadora es que una computadora sea capaz de extraer información útil para responder a preguntas sobre su contenido. Las técnicas necesarias para conseguir dicho objetivo proceden de diversas áreas como la ingeniería, informática y matemáticas. El estudio de los mecanismos de la visión humana es el objetivo de la visión por computadora. La visión por computadora busca crear sistemas que sean capaces de reconocer un objeto determinado en una imagen. La visión por computadora, para los humanos como para una computadora, consta principalmente de dos fases: captar una imagen e interpretarla.

Para Marcos et al. (2006) a pesar de la complejidad que presenta el ojo humano, la fase de captación de imágenes hace mucho tiempo que está resuelta en el área computacional. El ojo de la computadora es la cámara de vídeo y su retina un sensor que es sensible a la intensidad luminosa. En la visión por computadora lo que resta es interpretar las imágenes, detectar objetos, extraer información.

Según Meré (2008) podemos clasificar el proceso de visión por computadora en dos grupos. En un primer grupo se encuentran las etapas que ejecutan métodos de bajo nivel y en un segundo grupo, las que realizan un procesamiento de la imagen de alto nivel o un análisis a nivel de escena. El objetivo de las etapas de bajo nivel es obtener las características más básicas de la imagen, como bordes, regiones y otros atributos simples. En el caso del procesamiento de alto nivel, se recogen las características extraídas en el nivel inferior y se construye una descripción de la escena. A continuación, se describen de forma breve las etapas involucradas en el proceso de visión por computadora:

- **Adquisición de la imagen:** en esta etapa se captura una proyección en dos dimensiones de la luz reflejada por los objetos de la escena.

- **Preprocesamiento:** se realizan tareas de eliminación de ruido y/o realce de la imagen.

- **Segmentación detección de bordes y regiones:** permite separar los diferentes elementos de la escena.

- **Extracción de características:** Se obtiene una representación formal de los elementos segmentados en la etapa anterior.

- **Reconocimiento y localización:** mediante técnicas, como pueda ser la triangulación, se localiza al objeto en el espacio 3D.

- **Interpretación:** a partir de la información obtenida en las etapas previas y del conocimiento acerca del entorno se interpreta la escena.

De acuerdo con Titano et al. (2018) las técnicas de aprendizaje máquina y redes neuronales proporcionan que la visión por computadora le dé sentido a lo que ve y la visión por computadora se acerca al sistema cognitivo visual humano. La visión por computadora supera la visión humana en muchas aplicaciones, como el reconocimiento de patrones.

2.6.2. Aprendizaje máquina

El aprendizaje máquina para Berzal (2019) es una disciplina de la inteligencia artificial, es una tecnología que permite hacer automáticas una serie de operaciones con el fin de reducir la necesidad de que intervengan los seres humanos. El aprendizaje consiste en la capacidad del sistema para identificar determinados patrones. El sistema de aprendizaje máquina necesita contar con un volumen de datos de relevancia para poder suministrar respuestas realmente válidas. Una de las ventajas que se obtienen del aprendizaje automático es que realiza la distinción de forma automática de patrones haciendo uso de algoritmos matemáticos. Este tipo de técnicas se usan para la clasificación de las imágenes o de la toma de decisiones.

Tipos de Aprendizaje Máquina

De acuerdo con Simeone (2018) dependiendo de los datos disponibles y la tarea que queramos abordar, podemos elegir entre distintos tipos de aprendizaje máquina. Estos son: aprendizaje supervisado, aprendizaje no supervisado y aprendizaje por refuerzo.

- **Aprendizaje supervisado:** se basa en lo que se conoce como información de entrenamiento. Se entrena al sistema proporcionándole cierta cantidad de datos etiquetados. Teniendo los suficientes datos, se le podrán ingresar nuevos datos al sistema sin necesidad de etiquetas, en base a patrones distintos que ha venido registrando durante el entrenamiento. Este sistema se conoce como clasificación. El aprendizaje supervisado se suele usar en problemas de clasificación y problemas de regresión. Estos dos tipos principales de aprendizaje supervisado, clasificación y regresión, se distinguen por el tipo de variable objetivo. En los casos de clasificación, es de tipo categórico, mientras que, en los casos de regresión, la variable objetivo es de tipo numérico.

- **Aprendizaje no supervisado:** tienen como finalidad la comprensión de patrones de información. Es un modelo de problema que se conoce como clustering. Es un método de entrenamiento más parecido al modo en que los humanos procesan la información.

- **Aprendizaje por refuerzo:** los sistemas aprenden a partir de la experiencia. Cuando el sistema toma una decisión errónea, es penalizado, dentro de un sistema de registro de valores. Mediante dicho sistema de premios y castigos, se desarrolla una forma más efectiva de realizar tareas.

Descripción del proceso de Aprendizaje máquina

Cortés (2017) describe que el proceso de aprendizaje máquina se puede analizar en 4 pasos: preparación de los datos, representación, aprendizaje y evaluación del modelo utilizado.

- **Prepara datos:** Existen distintos tipos de datos útiles se tienen que identificar. En esta etapa se deben depurar los datos.

- **Representación de los datos:** es importante que las características sean relevantes, útiles y que permitan entrenar el modelo.

- **Aprendizaje:** seleccionar un algoritmo que alimentaremos con datos de entrenamiento. Así se obtiene un modelo que puede generar predicciones basadas en datos.

- **Evaluación del modelo:** analizando los datos de salida del algoritmo podemos obtener una métrica de rendimiento para decidir si el modelo es satisfactorio o si se debe iterar el proceso.

2.6.3. Aprendizaje profundo

Según Zhang et al. (2016) el aprendizaje profundo ha logrado avances importantes en la última década. El aprendizaje profundo contiene una variedad de métodos que incluyen redes neuronales, modelos probabilísticos jerárquicos y muchos algoritmos específicos de aprendizaje de funciones supervisadas y no supervisadas. La mayor diferencia entre el aprendizaje profundo y los métodos clásicos de reconocimiento visual es que los métodos de aprendizaje profundo aprenden automáticamente las funciones de una gran cantidad de datos, en lugar de requerir funciones de ingeniería a mano.

Aunque las primeras teorías sobre el aprendizaje profundo se desarrollaron en la década de los ochenta, existen dos razones principales por las que solo ha empezado a resultar útil recientemente:

- Requiere grandes cantidades de datos etiquetados. Por ejemplo, para el desarrollo de un vehículo autónomo se necesitan millones de imágenes y miles de horas de vídeo.

- Depende de una potencia de cálculo significativa. Una Unidad de Procesamiento Gráfico (por sus siglas en inglés, GPU) es un coprocesador dedicado al procesamiento de gráficos u operaciones de coma flotante. Las GPU de alto rendimiento tienen una arquitectura paralela que resulta eficiente para el aprendizaje profundo. En combinación con clusters o con el cálculo en la nube, esto permite a los equipos de desarrollo reducir el tiempo necesario para el entrenamiento de una red de aprendizaje profundo de semanas a horas o incluso menos.

La mayor parte de los métodos de aprendizaje emplean arquitecturas de redes neuronales, por lo que a menudo, los modelos de aprendizaje profundo se denominan redes neuronales profundas. El término “profundo” suele hacer referencia al número de capas ocultas en la red neuronal.

2.6.4. Redes neuronales

Una red neuronal se puede describir como un modelo matemático inspirado en las neuronas de un ser humano y en cómo se organizan formando la estructura del cerebro. La unidad básica de la red neuronal es el perceptrón. Las entradas son las dos notas, n_1 y n_2, cada una con su correspondiente peso w_n (las caracteristicas del objeto de interes que se desea encontrar en una imagen o vídeo). La salida, n_f, será 1 si está aprobado y 0 desaprobado, como se muestra en la Figura 2.5.

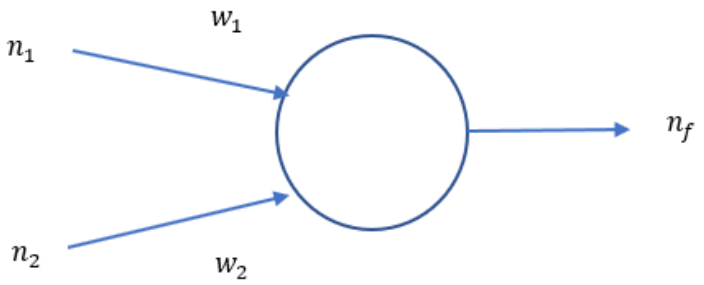


Figura 2.5: El perceptrón.

El principal objetivo de las redes neuronales de acuerdo con Becerra (2012) es la construcción de sistemas capaces de presentar un cierto comportamiento inteligente. Esto implica la capacidad de aprender a realizar una determinada tarea. Una red neuronal está compuesta de tres partes: entrada, núcleo y salidas. La Figura 2.6, muestra un esquema de una red neuronal artificial.

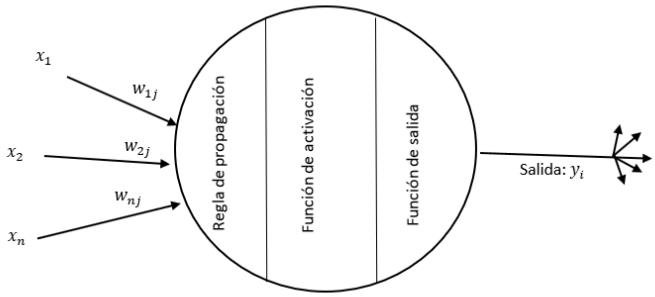


Figura 2.6: Red neuronal artificial.

Las entradas reciben los datos o parámetros que le permiten decidir a la neurona si estará activa o no, normalmente se presentan como x_1, x_2, ..., x_n. Entre la entrada y el

núcleo se tienen los pesos (w_1, w_2, ..., w_n), que representan la memoria de la red. En el núcleo se realizan todas las operaciones necesarias para determinar la salida de la neurona; el proceso que se realiza en el núcleo varía dependiendo de la red neuronal que se esté trabajando. Las salidas devuelven la respuesta de la neurona, es decir se está activa o no, representadas comúnmente como y_1, y_2, ..., y_n. En el núcleo se realizan tres tipos de operaciones para determinar la salida de la neurona que son: regla de propagación, función de activación y función de salida.

Descripción de operaciones:

- **Entrada y salida:** las entradas y salidas de una neurona pueden ser clasificadas en dos grandes grupos, binarias o continuas. Las neuronas binarias, sólo admiten dos valores posibles. En general en este tipo de neurona se utilizan los siguientes dos alfabetos 0,1 o -1,1. Por su parte, las neuronas continuas admiten valores dentro de un determinado rango, que en general suele definirse como [-1, 1]. La selección del tipo de neurona a utilizar depende de la aplicación y del modelo a construir.

- **Pesos:** el peso sináptico w_{ij} define la fuerza de una conexión sináptica entre dos neuronas, la neurona presináptica i y la neurona postsináptica j. Los pesos sinápticos pueden tomar valores positivos, negativos o cero. En caso de una entrada positiva, un peso positivo actúa como excitador, mientras que un peso negativo actúa como inhibidor. En caso de que el peso sea cero, no existe comunicación entre el par de neuronas. Mediante el ajuste de los pesos sinápticos la red es capaz de adaptarse cualquier entorno y realizar una determinada tarea.

- **Regla de propagación:** la regla de propagación se aprecia en la ecuación 2.1 determina el potencial resultante de la interacción de la neurona i con las N neuronas vecinas. La regla de propagación más simple y utilizada consiste en realizar una suma de las entradas ponderadas con sus pesos correspondientes:

$$net_i(t) \sum_{j=1}^{N} w_{ij} * x_j(t) \tag{2.1}$$

- **Función de activación:** la función de activación determina el estado de activación actual de la neurona en base al potencial net_i y al estado de activación anterior de la neurona $a_i(t-1)$. El estado de activación de la neurona para un determinado

instante de tiempo t puede ser expresado con la siguiente ecuación 2.2:

$$a_i(t) = f(a_i(t-1), net_i(t)) \tag{2.2}$$

Sin embargo, en la mayoría de los modelos se suele ignorar el estado anterior de la neurona, definiéndose el estado de activación en función del potencial resultante h_i como se muestra en la ecuación 2.3:

$$a_i(t) = f(net_i(t)) \tag{2.3}$$

- **Función de salida:** la función de salida proporciona el valor de salida de la neurona, en base al estado de activación de la neurona. como se muestra en la ecuación 2.4:

$$y_i(t) = f(net_i(t)) \tag{2.4}$$

2.6.5. Redes neuronales convolucionales

En 1998 las redes neuronales convolucionales tomaron relevancia cuando Lecun et al. (1998), con el algoritmo de Backpropagation pudo entrenar un modelo para el reconocimiento de dígitos en un documento. Dada una arquitectura de red adecuada, los algoritmos de aprendizaje basados en gradientes se pueden utilizar para sintetizar una superficie de decisión compleja que puede clasificar patrones de alta dimensión, como caracteres escritos a mano, con un preprocesamiento mínimo.

De acuerdo con Massiris et al. (2020) el aprendizaje profundo permite que modelos computacionales compuestos por varias capas de procesamiento puedan aprender representaciones sobre datos con múltiples niveles de abstracción y mediante ese concepto, descubrir representaciones precisas de forma autónoma en grandes volúmenes de datos. Las redes neuronales convolucionales es un algoritmo de aprendizaje profundo que está diseñado para trabajar con imágenes, tomando éstas como entrada, asignándole importancias (pesos) a ciertos elementos en la imagen para así poder diferenciar unos de otros. Éste es uno de los principales algoritmos que ha contribuido en el desarrollo y perfeccionamiento del campo de visión por computadora.

Las redes neuronales convolucionales trabajan dividiendo y modelando la información en partes más pequeñas y combinando esta información como se representa en la Figura

2.7. En el caso del tratamiento de una imagen, las primeras capas tratarían de detectar los bordes de las figuras. Las siguientes capas buscarían combinar los patrones de detección de bordes para conseguir formas más simples y aplicar patrones de posición de objetos, iluminación. Por último, en las últimas capas se intentaría hacer coincidir la imagen con todos los patrones descubiertos, para conseguir una predicción final de la suma de todos ellos. Así es como las redes neuronales convolucionales consiguen modelar una gran cantidad de datos, dividiendo previamente el problema en partes para conseguir predicciones más sencillas y precisas.

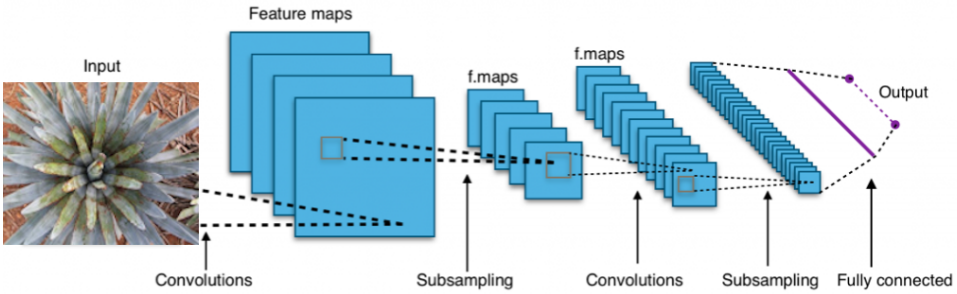


Figura 2.7: Modelo de una red neuronal convolucional. Fuente: Massiris et al. (2020).

Estructura de una red neuronal convolucional

En general, todas las redes neuronales convoluciones están formadas por una estructura compuesta por 3 capas:

- **Capa convolucional:** López et al. (2016) menciona que la capa convolucional recibe como entrada la imagen y luego aplica sobre ella un filtro o kernel que devuelve un mapa de las características de la imagen original, de esta forma se logra reducir el tamaño de los parámetros. La convolución es una operación de productos y sumas entre la imagen de entrada y un filtro que genera un mapa de características. Este mismo filtro sirve para extraer el mismo rasgo en cualquier parte de la imagen, esto permite reducir el número de conexiones y el número de parámetros a entrenar en comparación con una red multicapa full-connected.

- **Capa de reducción (pooling):** la capa de reducción o pooling se coloca generalmente después de la capa convolucional. Su utilidad principal es la reducción de las dimensiones (ancho x alto) del volumen de entrada para la siguiente capa convolucional. La operación realizada por esta capa también se llama reducción de

muestreo, ya que la reducción de tamaño conduce también a la pérdida de información. Sin embargo, una pérdida de este tipo puede ser beneficioso para la red por dos razones:

- La reducción del tamaño provoca una menor sobrecarga de cálculo en las próximas capas de la red.
- Reduce habitualmente el sobreajuste.

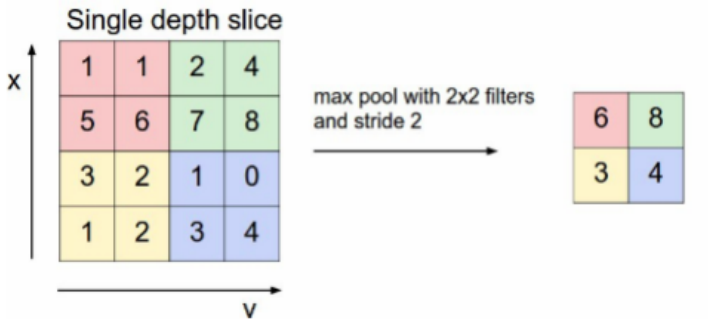


Figura 2.8: Reducción o pooling. Fuente: Arista-Jalife et al. (2017).

La operación que se suele aplicar en esta capa es "max-pooling", que divide la imagen de entrada en un conjunto de rectángulos y respecto a cada uno de ellos, se queda con el valor máximo. Como se muestra en la Figura 2.8.

- **Capa clasificadora** Después de pasar por las 2 capas anteriores se han extraído sus características más destacadas, los datos llegan a la fase de clasificación. Para ello, las redes convolucionales utilizan capas completamente conectadas en las que cada píxel se trata como una neurona independiente. Esta última capa clasificadora tendrá tantas neuronas como el número de clases que se debe predecir.

2.6.6. Algoritmo para la deteccion de objetos YOLO

Para Serrano (2017) YOLO es un algoritmo de código abierto utilizado en la detección de objetos en imágenes y vídeo, este algoritmo usa una red neuronal convolucional para dividir una imagen en regiones, prediciendo cuadros delimitadores de identificación y probabilidades por cada región. Las cajas son ponderadas a partir de las probabilidades

predichas. El algoritmo aprende representaciones generalizables de los objetos, permitiendo un bajo error de detección para entradas nuevas, diferentes al conjunto de datos de entrenamiento.

De acuerdo con Massiris et al. (2020) YOLO toma la detección de objetos como un problema único de regresión, una red convolucional única predice simultáneamente múltiples cuadros delimitadores que enmarcan los objetos en la imagen y predice probabilidades condicionales por cada clase p(Clase | Objeto) para cada uno de estos cuadros delimitadores. YOLO trabaja globalmente sobre la imagen cuando hace predicciones, a diferencia de la técnica de ventana deslizante y las técnicas basadas en el análisis de las regiones en una imagen. Por esto, codifica implícitamente la información contextual, modela el tamaño y la forma de los objetos, así como su apariencia.

La primera versión del algoritmo YOLO fue presentada por Redmon et al. (2015) con su investigación *Solo mira una vez: detección unificada de objetos en tiempo real*. En el año 2016 la arquitectura de la red neuronal convolucional estaba compuesta por 24 capas convolucionales seguidas por dos capas completamente conectadas, en lugar de los módulos iniciales propuestos por GoogLeNet, la arquitectura de YOLOv1 utilizaba capas de reducción de 1×1 seguidas de capas convolucionales de 3×3 como se puede apreciar en la Figura 2.9.

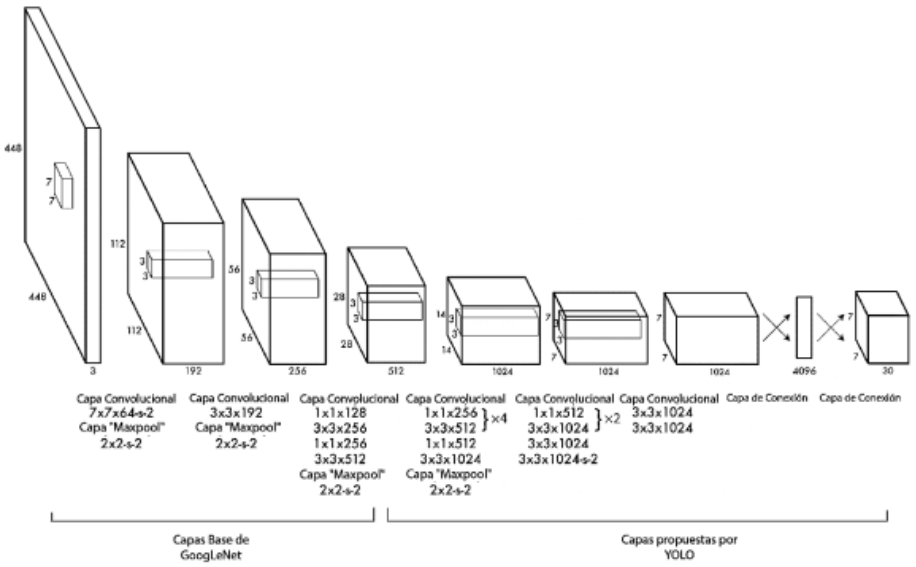


Figura 2.9: Arquitectura YOLO versión 1. Fuente: Redmon et al. (2015).

En la investigación realizada por Redmon and Farhadi (2016) *YOLO9000: Mejor, Más*

Rápido, Más Fuerte se presenta la segunda versión del algoritmo YOLO. El objetivo de esta versión es mejorar la precisión y hacer una detección aún más rápida. En cuanto a las mejoras de precisión, se añade la normalización por lotes en las capas de convolución. En esta segunda versión el clasificador entrenaba con imágenes de tamaño 448x448, mejorando la precisión promedio media (por sus siglas en inglés, mAP). Las formas de las cajas de anclaje no son seleccionadas por el usuario, sino que las selecciona YOLO de forma que facilite a la red el que aprenda a detectar objetos. El tamaño lo selecciona YOLO mediante el algoritmo de aprendizaje no supervisado K-Means. La red genera hasta 5 cuadros delimitadores para cada celda y predice 5 coordenadas para cada cuadro delimitador: t_x, t_y, t_w, t_h y t_0. Si la celda está desplazada desde la esquina superior izquierda de la imagen por c_x y c_y y el cuadro de anclaje tiene un ancho y un alto dado por p_w y p_h respectivamente podemos ver las predicciones en la Figura 2.10.

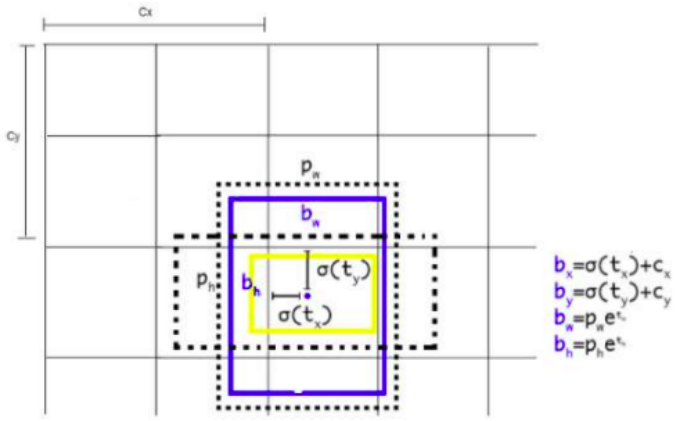


Figura 2.10: Predicciones de los cuadros delimitadores. Fuente: Redmon et al. (2015).

Esta versión de YOLO agrega cajas de anclaje, esto permite detectar más de un objeto por celda. YOLO versión 2 usa dos funciones con las cuales descarta áreas vacías y selecciona las mejores propuestas. Las funciones son:

- **Intersección sobre la unión (por sus siglas en inglés, IOU):** genera un porcentaje de acierto del área de predicción contra el cuadro delimitador real que se desea predecir.
- **Supresión no máxima:** de entre las cinco anclas generadas, selecciona la que mejor se ajusta al resultado y elimina el resto. En una imagen se pueden tener muchas áreas diferentes propuestas que se superponen.

La arquitectura de YOLO versión 2 es un sistema más preciso y complejo debido a que se propone un modelo de clasificación nuevo, denominado Darknet19. Esta red consta de 19 capas convolucionales y cinco capas de máxima agrupación. El resultado es que alcanza una precisión muy notable, mejorando a la primera versión de YOLO.

En el año 2018 surge YOLO versión 3 con la investigación *Una mejora incremental* publicada por Redmon and Farhadi (2018). La arquitectura de YOLO versión 3 es una variante de la red neuronal convolucional Darknet19, esta compuesta de 53 capas entrenada en Imagenet. Para la tarea de detección, se apilan 53 capas más, esto da como resultado una arquitectura subyacente totalmente convolucional de 106 capas.YOLO versión 3 es rápido y preciso en términos de mAP y valores de IOU. Funciona significativamente más rápido que otros métodos de detección con un rendimiento comparable como se aprecia en la Figura 2.11, donde se entrenaron diferentes modelos con el mismo conjunto de datos y equipos de cómputo con hardware similar.

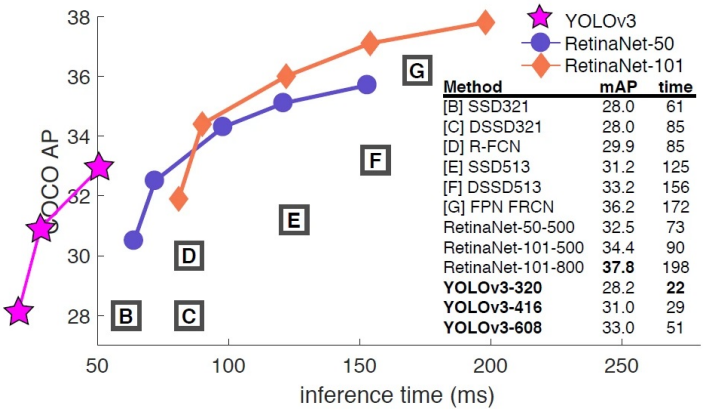


Figura 2.11: Comparación de tiempos de inferencia. Fuente: Bochkovskiy et al. (2020).

La detección se realiza mediante la aplicación de núcleos de detección $1x1$ en mapas de características de tres tamaños diferentes en tres lugares diferentes de la red. La forma del núcleo de detección es $1x1x(Bx(5+C))$. Donde B representa el número de cuadros delimitadores que puede predecir una celda en el mapa de características, 5 es para los cuatro atributos del cuadro delimitador y la confianza de un objeto y C es el número de clases.

YOLO versión 3 hace predicciones en tres escalas, que se dan al reducir la dimensión de la imagen de entrada en 32, 16 y 8 píxeles respectivamente, en total utiliza nueve cajas de anclaje. Tres para cada escala. Como nota cronológica, Glenn Jocher, que no estaba

afiliado a Joseph Redmon, creó una popular implementación de YOLOv3 en PyTorch.

En la investigación presentada por Bochkovskiy et al. (2020) agregaron dos categorías de métodos para mejorar la precisión del detector de objetos en YOLO versión 4:

- **Bolsa de regalos (BoF):** es un conjunto de métodos que cambian la estrategia de entrenamiento aumentando el costo de entrenamiento. Bolsa de regalos es el aumento de datos. El objetivo del aumento de datos es tener mayor variabilidad de las imágenes de entrada, para que el modelo de detección de objetos tenga mayor robustez.

- **Bolsa de especiales (BoS):** en términos generales, estos módulos de complemento son para mejorar ciertos atributos en un modelo, como ampliar el campo receptivo, introducir un mecanismo de atención o fortalecer la capacidad de integración de funciones y el canal en mapas de características concatenadas de múltiples escalas. En posprocesamiento es un método para evaluar los resultados de predicción.

YOLO versión 4 usa la CNN CSPDarknet53 como modelo extractor de funciones para la versión GPU. El método de aumento de datos aplicado es nombrado Mosaico, es la combinación de cuatro imágenes del conjunto de datos de entrenamiento en una imagen. Esto favorece a la normalización por lotes ya que calcula las estadísticas de activación de cuatro imágenes diferentes en cada capa permitiendo, reducir la selección de un gran tamaño de mini-lote para el entrenamiento. El entrenamiento autoadversario (por sus siglas en inglés, SAT) es una técnica de aumento de datos. La CNN calcula la pérdida y luego cambia la información de la imagen a través de la retropropagación para formar la ilusión de que no hay un objetivo en la imagen y luego realiza una detección normal del objetivo en la imagen modificada. Cabe señalar que en el proceso de retropropagación de SAT, no es necesario cambiar los pesos de la red.

En el año 2020 Glenn Jocher presento la versión 5 de YOLO basado de forma nativa en PyTorch. El autor no ha publicado un artículo pero hay una gran cantidad de análisis de la estructura de red de YOLO. Glenn Jocher es el creador del aumento de datos de mosaico aplicado en YOLOv3. Como YOLOv5 es un detector de objetos de una sola etapa, tiene tres partes importantes como cualquier otro detector de objetos de una sola etapa como podemos apreciar en la Figura 2.12.

- **Columna vertebral del modelo (Backbone):** red neuronal convolucional que agrega y forma características de imagen con granularidad fina de diferentes imágenes.

- **Modelo de cuello (Neck):** una serie de capas de red que mezclan y combinan características de la imagen y pasan las características de la imagen a la capa de predicción.
- **Modelo Cabeza (Head):** predice las características de la imagen, genera cuadros delimitadores y predice categorías.

YOLOv5 y YOLOv4 tienen estructuras de red similares, ambos usan la CNN CSP-Darknet53 como Backbone y usan la red de agregación de rutas (por sus siglas en inglés, PANet) y agrupación de pirámide espacial (por sus siglas en inglés, SPP) como cuello, y ambos usan YOLOv3 Head.

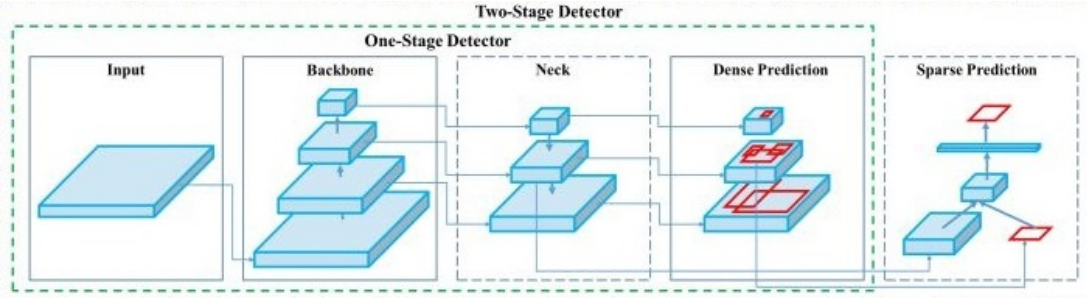


Figura 2.12: Arquitectura general de la red de detección de objetos. Fuente: Jocher et al. (2022).

En base a la documentación presentada por Jocher et al. (2022) en YOLOv5, el proceso de red parcial de etapas cruzadas (por sus siglas en inglés, CSP) se utiliza como columna vertebral para extraer características de las imágenes de entrada. El modelo Neck genera pirámides de características que ayudan a los modelos a generalizarse bien en la escala de objetos además ayuda a identificar un mismo objeto con diferentes tamaños y escalas. En YOLOv5 la red de agregación de rutas para la segmentación de instancias (PANet) se usa como cuello para obtener pirámides de funciones.

Según Liu et al. (2018) PANet se basa en el marco Mask R-CNN y las características de la red piramidal (por sus siglas en inglés, FPN) al tiempo que mejora la difusión de información. El extractor de características de esta red adopta una nueva estructura FPN que mejora la ruta ascendente, lo que mejora la propagación de características de bajo nivel. Cada etapa del tercer paso toma como entrada los mapas de características de la etapa anterior y los procesa con capas convolucionales de 3x3. Las salidas se agregan a los mapas de características de la misma etapa de la ruta de arriba hacia abajo a través de conexiones laterales y estos mapas de características forman la siguiente etapa.

El modelo Head se utiliza en la detección final, aplicando cuadros de anclaje en mapas de características y generando vectores de salida final con probabilidades de clase, puntajes de objetos y cuadros delimitadores. En el modelo YOLOv5, el modelo Head es el mismo que utilizan las versiones 3 y 4. El modelo Head puede ser aplicado en diferentes escalas como se aprecia en la Figura 2.13, su función principal es detectar objetos de diferentes tamaños. Cada Head tiene un total de (80 clases + 1 probabilidad + 4 coordenadas) * 3 cajas ancla, un total de 255 canales.

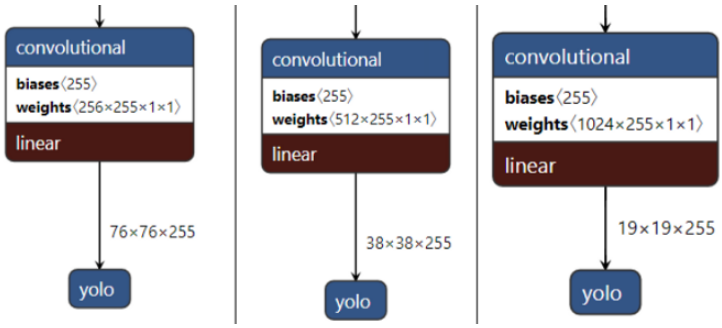


Figura 2.13: El modelo Heads aplicado a diferentes escalas. Fuente: Liu et al. (2018).

La función de activación Leaky ReLU es usada en las capas intermedias/ocultas y la función de activación sigmoidea se usa en la capa de detección final.

2.7. Métricas para la validación de modelos

En la actualidad un buen modelo para la detección de objetos no se limita a la simple detección, además es importante conocer el rendimiento de este modelo. Para lograr este objetivo es necesario evaluar el modelo con métricas que permitan un análisis claro de su desempeño.

Cuando se mide la habilidad de un algoritmo para enfrentar cierto problema, se usan métricas que pueden devolver como salida un conjunto de localizaciones de error, un intervalo de tiempo entre la aparición del objeto en la trayectoria de referencia y su detección por el sistema, en estos casos las métricas son calculadas estadísticamente (media, desviación, valores mınimos y máximos, etc.) (Vazquez, 2008).

Durante la etapa de entrenamiento, las ventanas generadas por la red tendrán una dimensión y posición azarosas. A medida que la red entrene estas ventanas generadas coincidirán en posición y tamaño con las que el banco de imágenes ofrece como verdad subyacente (etiquetas señaladas).

2.7.1. Intersección sobre la unión (IOU)

De acuerdo con Oh et al. (2021) la IOU es la medida que evalúa el porcentaje de superposición de dos cuadros delimitadores. En los modelos para la detección de objetos se tienen dos tipos de cuadros delimitadores, las etiquetas señaladas (Groundbox) son las anotaciones realizadas manualmente con ayuda de alguna aplicación en las imágenes, son etiquetas que permiten el entrenamiento y validación del algoritmo. También se tienen cuadros que se obtienen de la detección que ha realizado el algoritmo. La representación gráfica de la IOU la podemos analizar en las Figuras 2.14, 2.15, será 1 si coinciden tanto la detección como los cuadros delimitadores de las anotaciones, mientras que cuanto menor sea el valor, más alejado estará de una detección aceptable.

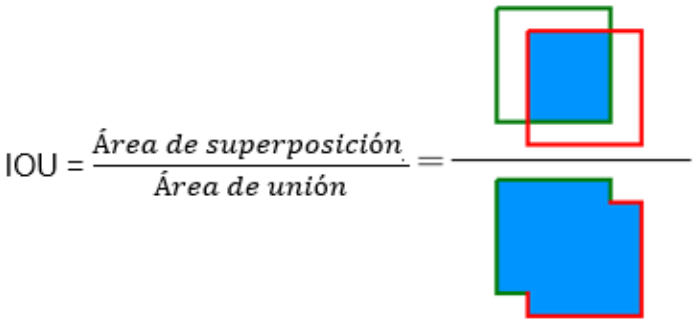


Figura 2.14: Intersección sobre la unión

Se puede obtener un numero de éxitos o fallas del algoritmo frente a diferentes situaciones según menciona Castro et al. (2019). En este caso se usan definiciones estándar:

- **Verdaderos Positivos (TP):** una detección correcta, detección con IOU mayor al umbral.
- **Falsos Positivos (FP):** una detección incorrecta, detección con IOU menor que el umbral.
- **Falsos Negativos (FN):** un objeto de interés no detectado.
- **Verdaderos Negativos (TN):** no aplica, representaría un error de detección corregido. En la tarea de detección de objetos, existen muchos cuadros delimitadores posibles que no deberían detectarse dentro de una imagen. Así, TN serían todos los

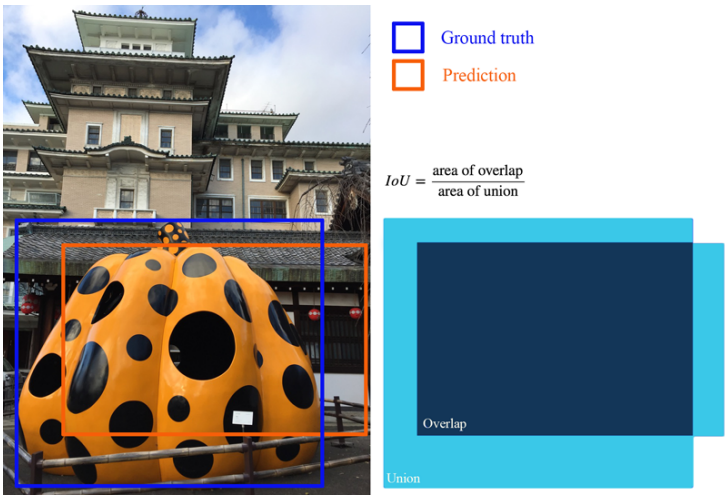


Figura 2.15: Intersección sobre la unión Ground truth/Prediction. Fuente: Rodríguez and Gómez (2020).

posibles cuadros delimitadores que no se detectaron correctamente (tantos posibles cuadros dentro de una imagen). Es por eso que no es utilizado por las métricas.

- **umbral:** dependiendo de la métrica, generalmente se establece en 50 %, 75 % o 95 %.

2.7.2. Matriz de confusión

Una matriz de confusión de acuerdo con Castro et al. (2019) es una técnica, que permite visualizar el desempeño de un algoritmo con información sobre los tipos de aciertos y errores del modelo. Una matriz de confusión se usa para evaluar el rendimiento de un modelo, normalmente de clasificación. En ella se compara los valores predichos por el sistema con los datos reales. Se puede interpretar una matriz de confusión analizado la Figura 2.16.

		Valores reales	
		Positivos	**Negativos**
Valores predichos	Positivos	TP = Verdaderos Positivos	FP = Falsos Positivos
	Negativos	FN = Falsos Negaivos	TN = Verdaderos Negativos

Figura 2.16: Matriz de confusión

Algunas medidas de desempeño se pueden definir a partir de la información contenida en una matriz de confusión. Estas medidas están determinadas por el número de errores

de clasificación y aciertos realizados por el clasificador. Estos datos pueden proporcionar mayor información en cuanto al desempeño del algoritmo, como la precisión, exactitud, sensibilidad, medida F.

2.7.3. Precisión

Precisión es la dispersión del conjunto de valores obtenidos a partir de mediciones repetidas de una magnitud. Cuanto menor es la dispersión mayor la precisión. La ecuación 2.5 representa la precisión que es la cantidad de muestras que se clasificaron correctamente con respecto a las muestras que fueron predichas como positivas por el clasificador.

$$Precision = \frac{TP}{TP + FP} \tag{2.5}$$

La precisión en el área de detección de objetos según Padilla et al. (2020) es la capacidad de un modelo para identificar solo los objetos relevantes. Es el porcentaje de predicciones positivas correctas y se representa con la siguiente ecuación:

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{\text{Todas las detecciones}} \tag{2.6}$$

2.7.4. Exactitud

De acuerdo con Moreno Diaz (2022) la exactitud es el grado de relación entre el resultado de las medidas y el valor verdadero de la magnitud que se pretende medir. No puede determinarse cuando el valor verdadero es desconocido, pero se aprecia en la Figura 2.17 sí tiene sentido cuando éste se conoce o cuando se comparan las medidas con la que proporciona un método del que se sabe de antemano que tiene un alto grado de exactitud y se emplea como referencia para la calibración. La exactitud está relacionada con la incertidumbre sistemática, que introduce desviaciones siempre en el mismo sentido que alejan el valor medido del verdadero (particularmente el error de cero).

Castro et al. (2019) se refiere a la exactitud a lo cerca que está el resultado de una medición del valor verdadero. En términos estadísticos, la exactitud está relacionada con el sesgo de una estimación esta definida por la ecuación 2.7.

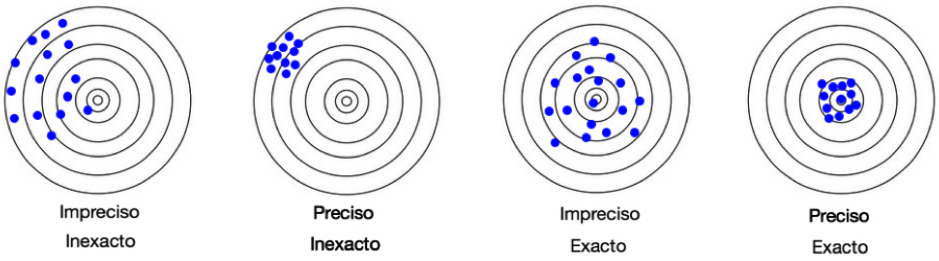


Figura 2.17: Exactitud y precisión. Fuente: Moreno Diaz (2022).

$$Exactitud = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} \sum_{j=1}^{n} N_{ij}} \tag{2.7}$$

2.7.5. Sensibilidad

Sensibilidad o tasa de verdaderos positivos definida por la ecuación 2.8 es una medida de la capacidad de un clasificador para seleccionar correctamente instancias de la clase objetivo relacionados con las muestras positivas.

$$Sensibilidad = \frac{TP}{TP + FN} \tag{2.8}$$

Según Padilla et al. (2020) la sensibilidad es la capacidad de un modelo para encontrar todos los casos relevantes (todos los cuadros delimitadores de verdad básica). Es el porcentaje de verdaderos positivos detectados entre todas las verdades de campo relevantes se representa por la ecuación:

$$Sensibilidad = \frac{TP}{TP + FN} = \frac{TP}{\text{Todos los cuadros delimitadores de verdad básica}} \tag{2.9}$$

2.7.6. Medida F

Se usa para evaluar los sistemas de recuperación de información, como los motores de búsqueda y de modelos de aprendizaje automático.

Medida F es la media armónica de la precisión y sensibilidad está definida por la ecuación 2.10. Permite evaluar un modelo teniendo en cuenta tanto la precisión como la sensibilidad utilizando una puntuación única, lo que resulta útil al describir el rendimiento

del modelo y al comparar modelos.

$$MedidaF = 2 \text{ x } \frac{Precision \text{ x } sensibilidad}{Precision + sensibilidad} \tag{2.10}$$

Es posible ajustar la medida F para dar más importancia a la precisión que a la recuperación, o viceversa. Las medidas F ajustadas comunes son la medida F 0.5 y la medida F2, así como la medida F1 estándar.

2.7.7. Precisión promedio

La precisión promedio (por sus siglas en inglés, AP) es el área bajo la curva, es decir, bajo el gráfico generado por las métricas precisión, sensibilidad, si se representa la precisión promedio en un gráfico tomando como ejes y = precisión y eje x = sensibilidad, se puede apreciar esta definición en la Figura 2.18, es también conocida como curva de recuperación de precisión.

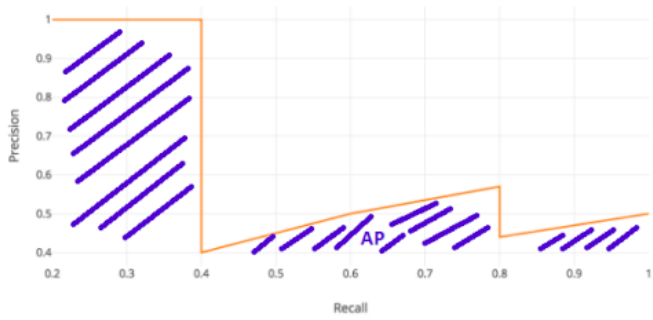


Figura 2.18: Representación gráfica de la precisión promedio. Fuente: Rodríguez and Gómez (2020).

La precisión promedio está definido por la ecuación 2.11.

$$AP = \int_0^1 p(r)dr \tag{2.11}$$

donde se puede definir p(r) como la curva formada por el gráfico precisión-sensibilidad.

El rendimiento de los detectores de objetos de acuerdo con Padilla et al. (2020) puede ser comparado calculando el área bajo la curva (por sus siglas en inglés, AUC) generada por la Precisión x Sensibilidad. Las curvas de AP suelen ser curvas en zigzag que suben y

bajan. Comparar diferentes curvas (distintos detectores) en el mismo gráfico no es tarea fácil porque las curvas tienden a cruzarse entre sí con frecuencia. Es por eso que la AP, una métrica numérica, también puede ayudar a comparar diferentes detectores. En la práctica, AP es la precisión promediada entre todos los valores de sensibilidad entre 0 y 1.

La precisión promedio interpolada clásica es utilizada por el desafío PASCAL VOC Challenge de acuerdo con Padilla et al. (2020), intenta resumir la forma de la curva Precisión x Sesibilidad promediando la precisión en un conjunto de once niveles de sesibilidad igualmente espaciados [0, 0.1, 0.2, ... , 1]:

$$AP = \frac{1}{11} \sum_{r \in 0,0,1,....,1} \rho interp(r) \tag{2.12}$$

En lugar de utilizar la precisión observada en cada punto, la AP se obtiene interpolando la precisión solo en los 11 niveles r tomando la precisión máxima cuyo valor de recuperación es mayor que r.

La AP interpolando todos los puntos en lugar de interpolar solo en los 11 puntos igualmente espaciados, podría interpolar todos los puntos de tal manera que:

$$\sum_{n=0} (r_{n+1} - r_n) \rho interp(r_{n+1}) \tag{2.13}$$

En este caso, en lugar de utilizar la precisión observada en pocos puntos, el AP ahora se obtiene interpolando la precisión en cada nivel tomando la precisión máxima cuyo valor de recuperación es mayor o igual que r + 1. De esta forma se calcula el área estimada bajo la curva.

2.7.8. Precisión promedio media

La métrica más utilizada para describir la precisión de un modelo detector de objetos de acuerdo con Rodríguez and Gómez (2020) es la Precisión promedio media (por sus siglas en inglés, mAP), esta métrica es equivalente a la medida F pero para imágenes. La métrica mAP se obtiene al final de la última época sobre el conjunto de datos de validación. Las detecciones múltiples del mismo objeto en una imagen se consideraron detecciones falsas, por ejemplo, cinco detecciones de un solo objeto se consideran como

una detección correcta y cuatro detecciones falsas. Por lo tanto, cada cuadro predicho es Verdadero-Positivo o Falso-Positivo. Cada cuadro predicho por el modelo es Verdadero-Positivo. No hay verdaderos negativos. La mAP definida por la ecuación 2.14 es la media de AP para un umbral de IOU definido. Un umbral clásico para IOU es [0.5,0.05, 0.95], de esta forma mAP será la media de los valores AP para cada valor de IOU, que va desde 0.5 a 0.95 con un paso de 0.05.

$$mAP = \frac{1}{|Classes|} \sum_{c \in classes} \frac{\#TP(c)}{\#TP(c) + \#FP(c)} \tag{2.14}$$

2.8. Algoritmo Cascada Haar

Es un algoritmo para la detección de objetos basado en el aprendizaje automático propuesto por Viola and Jones (2001). Se considera una de las primeras propuestas en la detección de objetos en una imagen, independientemente de su ubicación y escala en una imagen.

El algoritmo de Cascada Haar necesita muchas imágenes positivas y negativas para entrenar al clasificador. Es necesario extraer características del objeto de interés. Las características para el algoritmo de Cascada Haar son, kernels compuestos de un único valor, el cual se obtiene de la resta entre la suma de los píxeles en el rectángulo blanco y la suma de los píxeles en el rectángulo negro como se aprecia en la Figura 2.19.

El enfoque de ventana deslizante es ampliamente utilizado en el contexto de la detección de objetos en el algoritmo de Cascada Haar, consiste en una ventana de tamaño fijo que se desplaza a través de una imagen en varias escalas. En cada una de estas fases la ventana deslizante se detiene, calculando características y luego clasifica la región como Sí, esta región contiene un objeto de interés o No en caso de no existir objeto de interés.

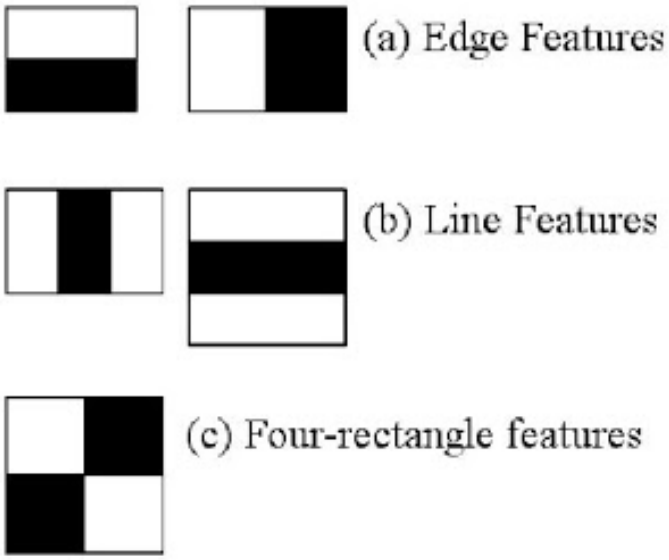


Figura 2.19: Características Haar. Fuente: Viola and Jones (2001).

Capítulo 3

Materiales y métodos

3.1. Experimentos preeliminares

Como parte del proceso para la selección de la metodología propuesta, se realizó la investigación y el desarrollo de los experimentos que a continuación se describen.

3.1.1. Segmentación por espacio de color

Se desarrolló una aplicación en el lenguaje de programación Python para extraer los valores del espacio de color RGB de un píxel seleccionado por el usuario en una imagen, internamente la aplicación realiza la conversión del espacio de color RGB a los espacios de color HSV y CIELab. Con esta aplicación se realizó la captura de 1000 registros de píxeles que pertenecían a plantas de agave y se realizó la captura de 1000 registros de píxeles que no pertenecían a plantas de agave. En la Figura 3.1 se puede apreciar la captura de datos.

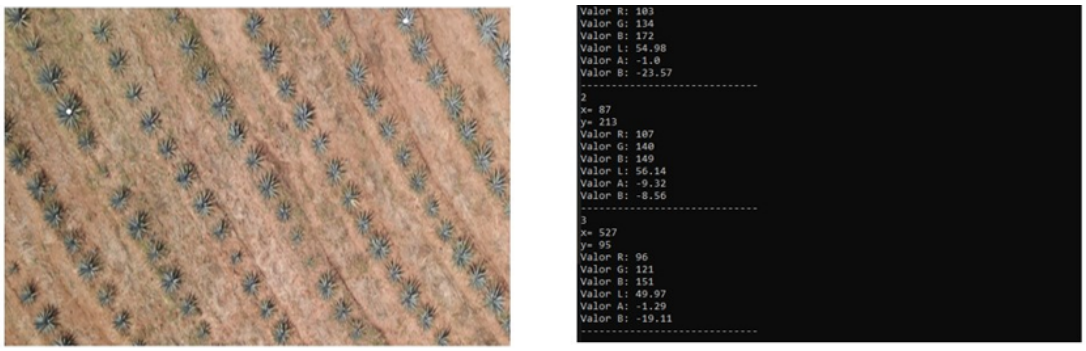


Figura 3.1: Captura de datos.

3.1.2. Operaciones morfológicas

Las operaciones morfológicas están basadas en la forma de la imagen, normalmente se aplican a imágenes binarias. Necesita dos entradas, una es nuestra imagen original, la segunda se llama elemento estructurante o núcleo (kernel) que decide la naturaleza de la operación. Dos operadores morfológicos básicos son: erosión y dilatación; Existen variantes de las operaciones básicas como Apertura, Cierre, Gradiente. La Figura 3.2 será usada para demostrar las operaciones morfológicas erosión y dilatación.

Figura 3.2: Imagen original.

La erosión es muy similar a la convolución, un kernel se desliza a través de la imagen. Un píxel de la imagen original uno o cero sólo se considerará uno si todos los píxeles que caen dentro de la ventana del kernel son uno, de lo contrario se erosiona (se convierte a cero). Por tanto, todos los píxeles cerca de los bordes de los objetos en la imagen serán descartados dependiendo del tamaño del kernel, en este ejemplo el kernel utilizado es de 3. Como consecuencia, como se aprecia en la Figura 3.3 el grosor en primer plano y la región blanca disminuyen en la imagen. Este procedimiento es útil para eliminar pequeños ruidos blancos, separar dos objetos conectados, tiene la característica de encoger la imagen una vez finalizado el proceso.

El proceso de dilatación es lo opuesto a la erosión. Un elemento de píxel es uno si al menos un píxel de la imagen de los que caen dentro de la ventana del kernel es uno. El kernel aplicado es de 3. La dilatación aumenta el tamaño de los objetos de primer plano. Normalmente, en un proceso de tratamiento de imágenes la erosión es seguida de

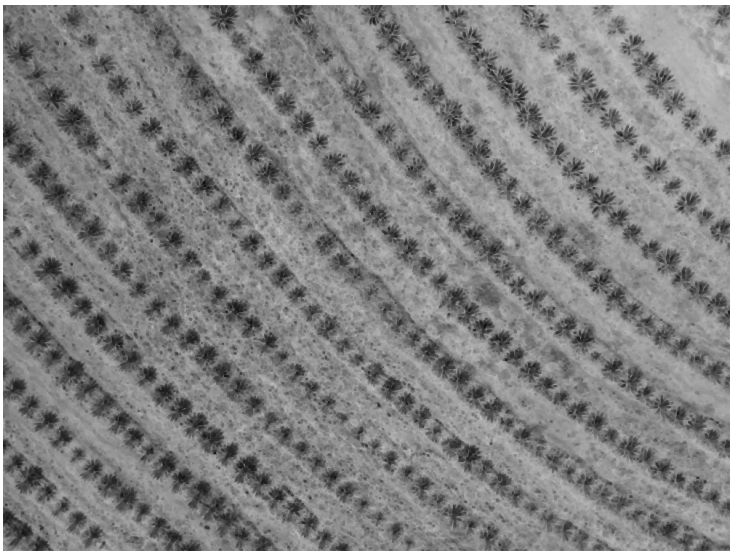


Figura 3.3: Proceso de erosión.

dilatación. La razón para esto es que, aunque la erosión elimina los ruidos blancos también encoge los objetos. Por tanto, para recuperar el tamaño inicial, este se dilata. El proceso de dilatación se puede apreciar en la Figura 3.4.

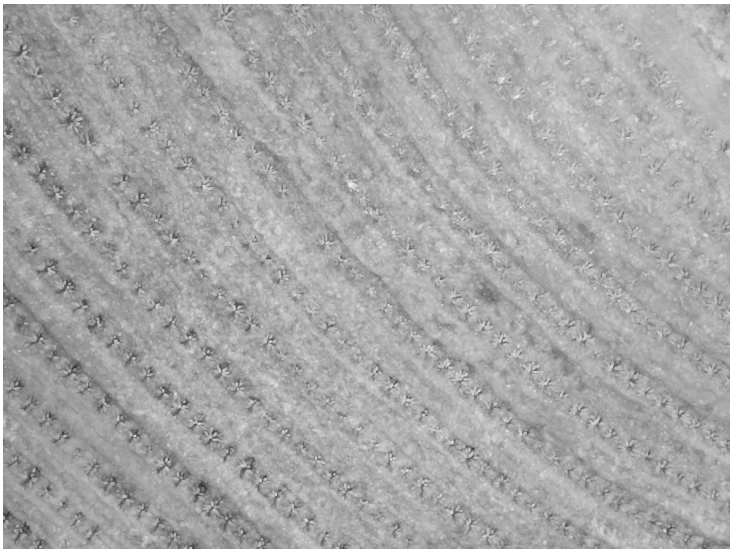


Figura 3.4: Proceso de dilatación.

3.1.3. Algoritmo de Cascada Haar

Para la detección y conteo de plantas de agave con el Algoritmo Cascada Haar se siguió la siguiente serie de pasos que serán descritos a continuación:

1. Preparar los datos de entrenamiento.
2. Entrenamiento del clasificador.
3. Aplicar el clasificador a una región de interés.
4. Optimización de parámetros.

Preparar los datos de entrenamiento

El conjunto de datos estaba compuesto de imágenes negativas (no eran plantas de agave) con 200 imágenes con un tamaño de 50 x 50 píxeles. El conjunto de datos de imágenes positivas (eran plantas de agave) estaba compuesto de 200 imágenes con un tamaño de 20 x 20 píxeles.

Entrenamiento del clasificador

OpenCV ofrece clasificadores preentrenados. Los archivos XML están disponibles en el repositorio github: https://github.com/opencv/opencv/tree/master/data/haarcascades. No se cuenta con un clasificador entrenado para la detección de plantas de agave.

Para entrenar un clasificador de plantas de agave se requieren imágenes positivas (es decir imágenes de plantas de agave) e imágenes negativas (que serían imágenes que no contengan plantas de agave). Mediante la aplicación *Cascade Trainger GUI* el clasificador fue entrenado con 20 etapas, se realizo la extracción de características de todas las imágenes, para después emplear un enfoque de aprendizaje maquina y proceder con el entrenamiento. Finalmente se podrá obtener el clasificador con la extensión XML que almacenara las características haar. Como se aprecia en la Figura 3.5

Aplicar el clasificador a una región de interés

Una vez que se entrena un clasificador, se puede aplicar a una región de interés (del mismo tamaño que se usó durante el entrenamiento) en una imagen de entrada, como se muestra en la Figura 3.6. El clasificador genera un *"1"* si es probable que la región muestre

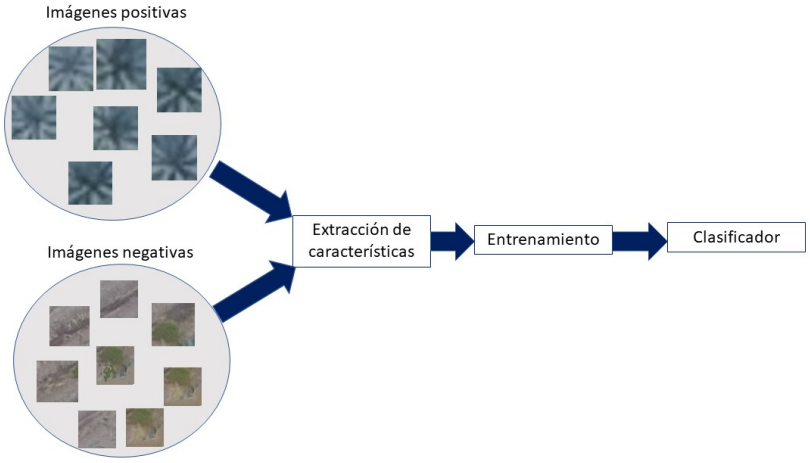


Figura 3.5: Diagrama del entrenamiento del clasificador.

el objeto (es decir, planta de agave) y *"0"* en caso contrario. Para buscar el objeto en toda la imagen, se puede mover la ventana de búsqueda por la imagen y verificar cada ubicación usando el clasificador. El clasificador está diseñado para que se pueda "redimensionar" fácilmente para poder encontrar los objetos de interés en diferentes tamaños.

Figura 3.6: Detecciones con el clasificador entrenado.

Optimizar parámetros

Algunos parámetros pueden ser modificados para adaptarlos al problema planteado en esta tesis y mejorar las detecciones. A continuación se da una breve descripción de estos

parámetros:

- **scaleFactor:** Este parámetro especifica que tanto va a ser reducida la imagen. Por ejemplo si se ingresa 1.1, quiere decir que se va a ir reduciendo la imagen en 10 %.
- **minNeighbors:** Indica la cantidad mínima de recuadros vecinos para detectar un valor como positivo.
- **minSize=(20,20):** Este parámetro indica el tamaño mínimo posible del objeto. Objetos más pequeños son ignorados.
- **maxSize=(30,30):** Este parámetro indica el tamaño máximo posible del objeto. Objetos más grandes son ignorados.

3.2. VANT Parrot Bluegrass

El VANT seleccionado para la adquisición de imágenes, es un Parrot Bluegrass™ Figura 3.7, su diseño y accesorios son recomendados para diferentes aplicaciones en la agricultura. El VANT Parrot Bluegrass™ está equipado con una cámara RGB, sensor multiespectral que captura imágenes en 4 bandas espectrales en luz visible y luz infrarroja, sensor parrot sequoia, sensor sunshine.

Figura 3.7: VANT Parrot Bluegrass™.

Las especificaciones técnicas del VANT Parrot Bluegrass™ son las siguientes:

- Sensores de vuelo

- Global Position System (por sus siglas en inglés, GPS) + GLONASS integrados.
- Inertial Navegate System (por sus siglas en inglés, INS).
- Altímetro.
- Ultrasonidos.
- Cámara vertical de flujo óptico.

- Sensor de luz solar (Ver Figura 3.8)
  - 4 sensores espectrales (filtros idénticos al cuerpo).
  - Unidad de medida inercial (por sus siglas en inglés, IMU) y magnetómetro.
  - Ranura para tarjeta tipo SD o Secure Digital (por sus siglas en inglés, SD).
  - Potencia: 1 Watt.
  - Sistema de posicionamiento global o GPS.
- Cámara RGB
  - Foto: cámara gran angular de 14MP.
  - Vídeo: 1080p Full HD.
  - Transmisión de vídeo: 360p / 720p.
  - Memoria interna de vídeo: De 32 GB.
- Cámara Multiespectral (Ver Figura 3.9)
  - Resolución: 1,2 Mpx, 1280x960 píxeles.
  - HFOV: 61.9.
  - VFOV: 48.5.
  - DFOV: 73.7.
- Bandas separadas
  - Verde: 550 NM +/- 40 NM.
  - Rojo: 660 NM +/-NM 40 NM.
  - Red Edge: 735 NM +/- 10 NM.

- Infrarrojo cercano (NIR): 790 NM +/- 40 NM.

- Segunda cámara RGB
  - Resolución: 16 MP, 4608 x 3456 píxeles.
  - HFOV: 63,9°.
  - VFOV: 50.1°.
  - DFOV 73.5°.

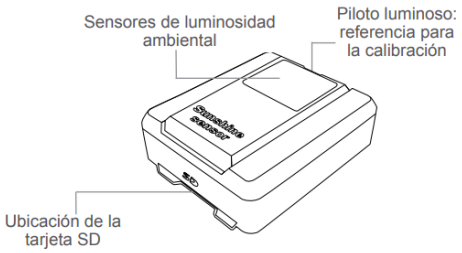


Figura 3.8: Sensor de luz solar. Fuente: support.parrot.com (2022).

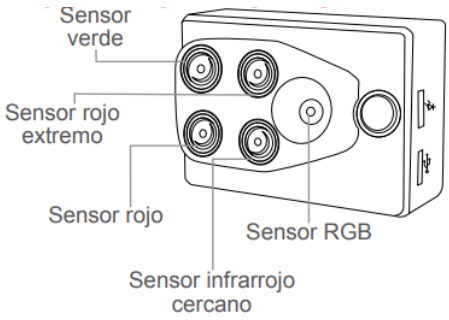


Figura 3.9: Sensor multiespectral. Fuente: support.parrot.com (2022).

3.3. Metodología propuesta

La metodología propuesta en esta investigación se divide en 4 etapas, como se muestra en la Figura 3.10. La etapa 1 comprende todo lo relacionado con el plan de vuelo y adquisición de imágenes. En la etapa 2, las imágenes son preprocesadas con el objetivo de crear un ortomosaico y delimitar la región de interés. En la etapa 3 se realiza la

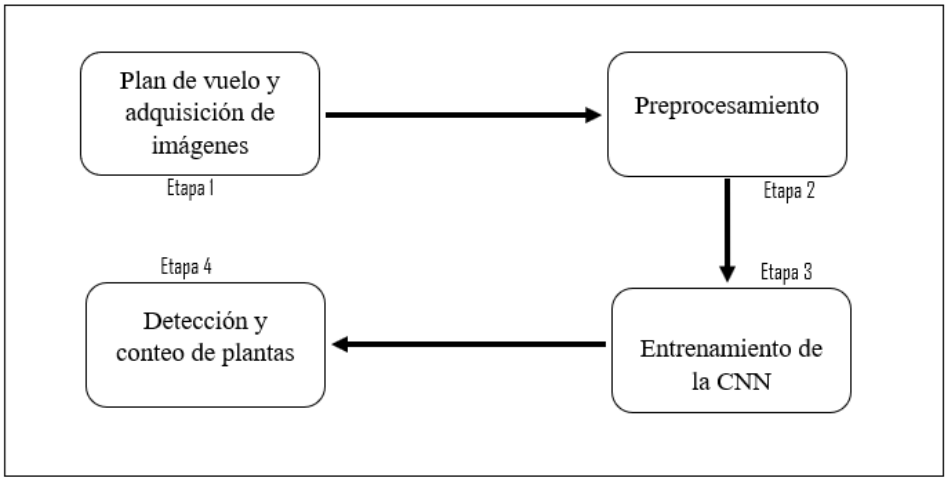


Figura 3.10: Metodología propuesta.

optimización de parámetros en la CNN para definir un modelo funcional. Finalmente en la etapa 4 se realiza la detección y conteo de plantas de agave mediante un modelo de aprendizaje profundo usando una CNN.

En las secciones posteriores se describen las actividades incluidas en cada una de las etapas de la metodología propuesta.

3.3.1. Planeación de vuelo y adquisición de imágenes

La planificación del plan de vuelo sirve de apoyo para un buen proceso fotogramétrico ya que limita las zonas de interés, la separación entre líneas de vuelo y el tiempo aproximado para el vuelo, esto nos garantiza imágenes precisas y de buena calidad. Se tienen que considerar factores externos como las condiciones geográficas y ambientales.

El plan de vuelo tiene como objetivo principal cubrir el terreno del cultivo mediante recorridos paralelos (líneas de vuelo) y equidistantes superpuestos en forma transversal, las fotografías correspondientes a cada recorrido deberán estar debidamente superpuestas en forma longitudinal.

Los factores más importantes para la planificación del vuelo de un VANT son los siguientes:

- **Altitud:** Es una medida absoluta respecto al nivel del mar, normalmente medida en pies la altitud también se expresa en metros mediante el símbolo *msnm* que significa metros sobre el nivel del mar. La altitud esta relacionada con la temperatura

ambiental, presencia de nubes, turbulencias o cantidad de oxígeno.

- **Altura:** Su expresión siempre será la diferencia entre la aeronave y el terreno, por lo que se trata de una medida relativa. Normalmente su unidad de medida serán los metros.

- **Velocidad:** Permite realizar la adquisición de imágenes en menor o mayor tiempo, este parámetro es configurable en ciertos tipos de drones.

- **Huella de registro:** Es la zona de la superficie terrestre que será capturada.

- **Solape:** Es el porcentaje de la huella de registro repetida por dos o más imágenes. El solape mínimo en el caso de fotogrametría, deberá ser mayor o igual a 70 % con drones.

El plan de vuelo se tiene que configurar considerando la norma oficial Mexicana NOM-107-SCT3-2019, donde se establecen los requerimientos para operar un sistema de aeronave pilotada a distancia (RPAS) en el espacio aéreo mexicano. En la sección 6.2 de la norma oficial Mexicana NOM-107-SCT3-2019 se encuentran las recomendaciones técnicas, preventivas y de seguridad relacionadas con los VANT.

Es importante almacenar las imágenes con algún criterio para llevar el control y tener un acceso a la información de manera eficiente. Las imágenes fueron almacenadas en base al criterio de la fecha de captura y la altura de vuelo.

3.3.2. Preprocesamiento

En la etapa 2, para procesar las imágenes y generar un ortomosaico es necesario establecer un flujo de trabajo mediante el software Pix4D como se muestra en la Figura 3.11. Mediante la herramienta Pix4Dcapture se establecen los parámetros para el plan de vuelo, las imágenes adquiridas se almacenaron en una base de datos. Pix4Dmapper permite construir un ortomosaico a partir de las imágenes adquiridas por el VANT. Una vez creado el ortomosaico, puede haber zonas que no requieren atención (objetos que no son plantas de agave) e incluso puede afectar en el análisis y estudios de los objetos de interés, se requiere realizar un recorte con el software QGIS de tal manera que solo se pueda apreciar la zona de interés.

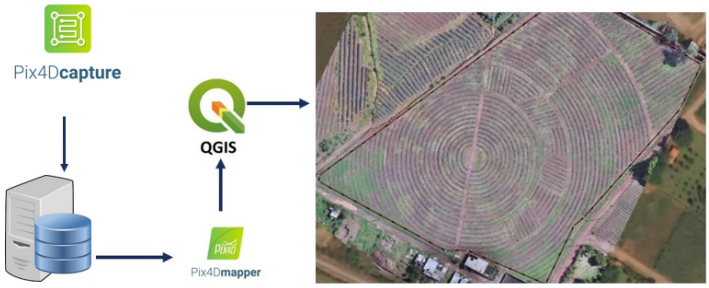


Figura 3.11: Preprocesamiento de imágenes

3.3.3. Entrenamiento de la red neuronal convolucional

Una vez que las imágenes han sido preprocesadas para la creación del ortomosaico y el recorte del área de interés, la siguiente etapa de la metodología propuesta es el entrenamiento de la CNN. El entrenamiento consta de 4 pasos, los cuales se describen a continuación.

1. **Etiquetado de imágenes:** Esta actividad consiste en etiquetar los objetos de interés en las imágenes adquiridas de los cultivos de agave. La aplicación LabelImg es una herramienta desarrollada en Python que permite seleccionar regiones de interés en una imagen y hacer anotaciones respecto a las regiones seleccionadas, como se aprecia en la Figura 3.12 (a). Las anotaciones se guardan como archivos XML, PASCAL VOC, YOLO y CreateM, donde la primera columna representa la clase a la pertenece la región seleccionada y las siguientes columnas representan las coordenadas de la región de interés en la imagen como aprecia en la Figura 3.12 (b).

2. **Ficheros con extensión .yaml:** Los ficheros con extensión *.yaml* como se muestra en la Figura 3.13 contienen las rutas donde se guarda el conjunto de datos, número de clases, etiquetas para el entrenamiento, validación y archivos xml.

3. **Definir arquitectura de la CNN:** YOLOv5 ofrece cuatro arquitecturas diferentes YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x. La diferencia principal entre estas arquitecturas radica en la cantidad de módulos de extracción de características y núcleos de convolución en ubicaciones específicas de la red. En la Figura 3.14 se muestra el contenido de la arquitectura YOLOv5s. Las arquitecturas disponibles en YOLO versión 5 fueron entrenadas previamente sobre el conjunto de datos COCO; COCO es un conjunto de datos para la detección de objetos en imágenes y

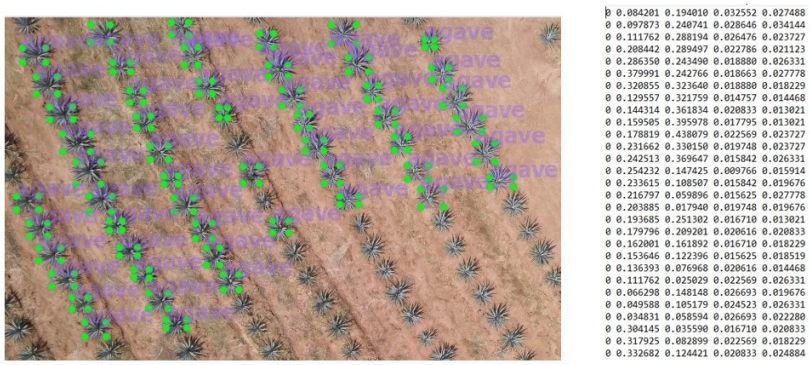


Figura 3.12: Metodología propuesta.

```yaml
path: /content/drive/MyDrive/YOLOv5/train_data
train: /content/drive/MyDrive/YOLOv5/train_data/images/train
val: /content/drive/MyDrive/YOLOv5/train_data/images/val
test:  # test images (optional)

# Classes
nc: 1  # number of classes
names: ['agave']  # class names
```

Figura 3.13: Fichero de configuración .yaml.

```yaml
nc: 1  # number of classes
depth_multiple: 1
width_multiple: 1

anchors:
  - [10,13, 16,30, 33,23]  # P3/8
  - [30,61, 62,45, 59,119]  # P4/16
  - [116,90, 156,198, 373,326]  # P5/32

# YOLOv5 v6.0 backbone
backbone:
  # [from, number, module, args]
  [[-1, 1, Conv, [64, 6, 2, 2]],  # 0-P1/2
   [-1, 1, Conv, [128, 3, 2]],  # 1-P2/4
   [-1, 3, C3, [128]],
   [-1, 1, Conv, [256, 3, 2]],  # 3-P3/8
   [-1, 6, C3, [256]],
   [-1, 1, Conv, [512, 3, 2]],  # 5-P4/16
   [-1, 9, C3, [512]],
   [-1, 1, Conv, [1024, 3, 2]],  # 7-P5/32  #%1024
   [-1, 3, C3, [1024]],
   [-1, 1, SPPF, [1024, 5]],  # 9
  ]

# YOLOv5 v6.0 head
head:
  [[-1, 1, Conv, [512, 1, 1]],
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
```

Figura 3.14: Arquitectura YOLOv5s.

vídeo. Contiene 80 clases, incluida la clase *planta en maceta* que fue lo más relacionado con nuestros objetos de interés, pero no se encontró una clase *agave*. El entrenamiento de la CNN inicializará con pesos de un modelo COCO previamente entrenado. El conjunto de datos usado es relativamente pequeño por este motivo se espera que el aprendizaje por transferencia produzca mejores resultados que el entrenamiento desde cero. La arquitectura preentrenada YOLOv5s esta compuesta de tres componentes principales: *Backbone, neck, head.*

4. **Entrenamiento de la CNN:** El entrenamiento inicia con los pesos de la arquitectura preentrenada con el comando: `!python train.py -img 800 -batch 8 -epochs 300 -data customdata.yaml -weights yolov5s.pt`

   La descripción de los parámetros utilizados en el comando anterior para el entrenamiento se pueden apreciar en la tabla 3.1.

Tabla 3.1: Parámetros para el comando de entrenamiento de la CNN.

Número	**Parámetro**	**Valor**
1	img	Tamaño de la imagen de entrada.
2	batch	Determina el tamaño de imágenes procesadas.
3	data	Archivo YAML donde se declara la ruta de almacenamiento del conjunto de datos.
4	epochs	Número de iteraciones de entrenamiento.
5	weights	Pesos previos al entrenamiento obtenidos del modelo preetrenado yolov5s.

3.3.4. Detección y conteo de plantas de agave

En la etapa 4 de la metodología propuesta se realiza la detección y conteo de plantas de agave mediante un modelo de aprendizaje profundo usando la CNN YOLOv5.

Finalizado el entrenamiento se almacenan los nuevos pesos obtenidos de la mejor y la ultima época. Los nuevos pesos pueden ser utilizados para la detección de plantas de agave con el comando:

```
!python detect.py -weights /yolov5/runs/train/exp22/weights/best.pt -img
800 -conf 0.25 -source /YOLOv5/imagen022.jpg"
```

La descripción de los parámetros utilizados en el comando anterior para la detección se pueden apreciar en la tabla 3.2.

Es importante mencionar que éste modelo solo detecta plantas de agave porque en los archivos .yaml solo se declaro la clase 0 nombrada agave y porque el entrenamiento se realizó con imágenes etiquetadas de platas de agave. Con esta configuración el modelo entrenado fue más eficiente en recursos computacionales para realizar el proceso de entrenamiento y detección.

Tabla 3.2: Parámetros para la detección.

Número	Parámetro	Valor
1	weights	Ruta de los pesos de entrenamiento.
2	img	Tamaño de la imagen de inferencia.
3	conf	Umbral de confianza del objeto.
4	source	Ruta donde se almacena la imagen de inferencia.

3.3.5. Descripción de YOLOv5 utilizado

La descripción de YOLOv5 se basa en su arquitectura que consta de tres partes (1) Backbone, (2) Neck, (3) Head, véase el Algoritmo 1.

Algoritmo 1: Algoritmo YOLO versión 5

Entrada: *Imagen de un cultivo de agave.*
Resultado: *Deteccion y conteo de plantas de agave en la imagen.*
definir ficheros .yaml;
arquitectura ← yolov5s.pt;
imagen ← agaves.jpg;
Function *backbone*
  Focus(imagen);
  Bottleneck();
  CSP();
  SPP();
end
Function *neck*
  PANet();
end
Function *head*
  Predicción de características de la imagen;
  Cuadros delimitadores;
  Categorías del objeto;
end

Backbone

El algoritmo recibe como entrada una imagen que llega al Backbone donde se extraen características importantes de la imagen y se realiza el preprocesamiento de los datos, incluido el aumento de datos de mosaico y relleno de imagen adaptable, las funciones principales del Backbone son Focus, Bottleneck, CSP, SPP. El objetivo principal de la función Focus es reducir capas, parámetros, FLOPS, reducir el uso de la memoria CUDA, aumentar la velocidad de avance y retroceso con un impacto mínimo en mAP.

La función Bottleneck es la estructura residual con una capa convolucional de 1x1 ($conv + batchnorm + leakyrelu$), luego una capa convolucional de 3x3 y finalmente se agrega la estructura residual a la entrada inicial.

En la función CSP es donde la entrada original se divide en dos ramas, la operación de convolución se realiza para reducir el número de canales a la mitad y luego la rama se realiza multiplicando Bottlenneck x N, después se concatena la rama 1 y la rama 2, para que la entrada y la salida de CSP tengan el mismo tamaño, el propósito es permitir que el modelo aprenda más funciones.

La entrada de la función SPP es de 512x20x20. Después de una capa convolucional de 1x1, la salida es de 256x20x20 y luego se muestrea a través de tres Maxpools paralelos y el resultado se agrega a sus características iniciales para generar una salida de 1024x20x20. Finalmente, se usa un kernel de convolución 512 para restaurarlo a 512x20x20.

Neck

Para la detección de objetos con diferentes tamaños y escalas YOLOv5 usa neck con la función llamada PANet que tiene como objetivo impulsar el flujo de información en un marco de segmentación de instancias basado en propuestas. La jerarquía de funciones se mejora con señales de localización precisas en las capas inferiores mediante el aumento de ruta ascendente, que acorta la ruta de información entre las capas inferiores y la función superior. Además, se emplea la agrupación de funciones adaptable, que vincula la cuadrícula de funciones y todos los niveles de funciones para hacer que la información útil en cada nivel de funciones se propague directamente a las siguientes subredes de propuestas, esto le permite al modelo identificar un mismo objeto con diferentes tamaños y escalas.

Head

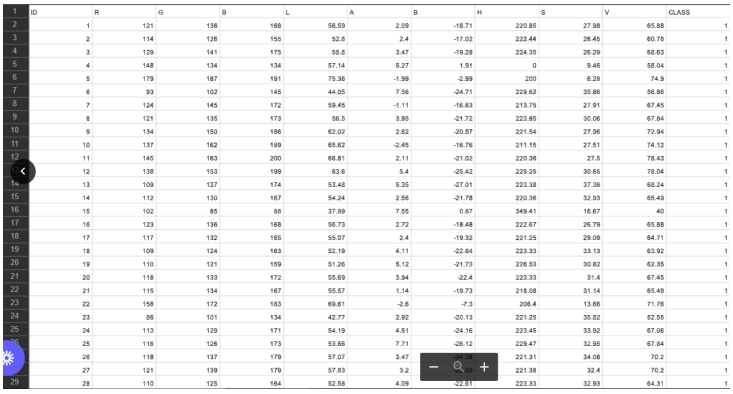
Head es la última parte de la arquitectura donde se encuentran las capas de salida o también llamadas capas de detección, es en este parte donde tiene aplicación los pesos obtenidos en el entrenamiento. La detección de un objeto de interés incluye aplicar cuadros de anclaje en características, generar probabilidades de clase, puntajes de objetividad y cuadros delimitadores.

Capítulo 4

Resultados

4.1. Resultados preliminares

Uno de los primeros resultados que se obtuvieron fue un conjunto de datos de 2000 registros clasificados en tres espacios de colores. En la Figura 4.1 se puede apreciar la organización de los datos un archivo separado por comas.



	ID	R	G	B	L	A	B	H	S	V	CLASS
2	1	121	136	168	56.59	2.09	-18.71	220.85	27.98	65.88	1
3	2	114	126	155	52.8	2.4	-17.02	222.44	26.45	60.78	1
4	3	129	141	175	58.8	3.47	-19.28	224.35	26.29	68.63	1
5	4	148	134	134	57.14	5.27	1.91	0	9.46	58.04	1
6	5	179	187	191	75.36	-1.99	-2.99	200	6.28	74.9	1
7	6	93	102	145	44.05	7.56	-24.71	229.62	35.86	56.86	1
8	7	124	145	172	59.45	-1.11	-16.63	213.75	27.91	67.45	1
9	8	121	135	173	56.5	3.95	-21.72	223.85	30.06	67.84	1
10	9	134	150	186	62.02	2.62	-20.57	221.54	27.96	72.94	1
11	10	137	162	189	65.62	-2.45	-16.76	211.15	27.51	74.12	1
12	11	145	163	200	66.81	2.11	-21.02	220.36	27.5	78.43	1
13	12	138	153	199	63.6	5.4	-25.42	225.25	30.65	78.04	1
14	13	109	127	174	53.48	5.35	-27.01	223.38	37.36	68.24	1
15	14	112	130	167	54.24	2.56	-21.78	220.36	32.93	65.49	1
16	15	102	85	88	37.99	7.55	0.87	349.41	16.67	40	1
17	16	123	136	168	56.73	2.72	-18.48	222.67	26.79	65.88	1
18	17	117	132	165	55.07	2.4	-19.32	221.25	29.09	64.71	1
19	18	109	124	163	52.19	4.11	-22.64	223.33	33.13	63.92	1
20	19	110	121	159	51.26	5.12	-21.73	226.53	30.82	62.35	1
21	20	118	133	172	55.69	3.94	-22.4	223.33	31.4	67.45	1
22	21	115	134	167	55.57	1.14	-19.73	218.08	31.14	65.49	1
23	22	158	172	183	69.61	-2.6	-7.3	206.4	13.66	71.76	1
24	23	86	101	134	42.77	2.92	-20.13	221.25	35.82	52.55	1
25	24	113	129	171	54.19	4.51	-24.16	223.45	33.92	67.06	1
26	25	116	126	173	53.66	7.71	-26.12	229.47	32.95	67.84	1
27	26	118	137	179	57.07	3.47	[illegible]	221.31	34.08	70.2	1
28	27	121	139	179	57.83	3.2	[illegible]	221.38	32.4	70.2	1
29	28	110	125	164	52.58	4.09	-22.61	223.33	32.93	64.31	1

Figura 4.1: Conjunto de datos.

El conjunto de datos fue analizado con el objetivo de aplicar una reducción de la dimensión de los datos y obtener los elementos más significativos. Se aplicó la técnica PCA. Como se aprecia en la Figura 4.2 se identificó que los cuatro elementos más significativos tomando en cuenta las 9 bandas son R, L, A, S.

Al conjunto de datos generado también se le aplicó la técnica de validación cruzada para garantizar el entrenamiento de diferentes modelos de aprendizaje máquina para la clasificación de los datos. En la Figura 4.3 se muestra los diferentes modelos entrenados con

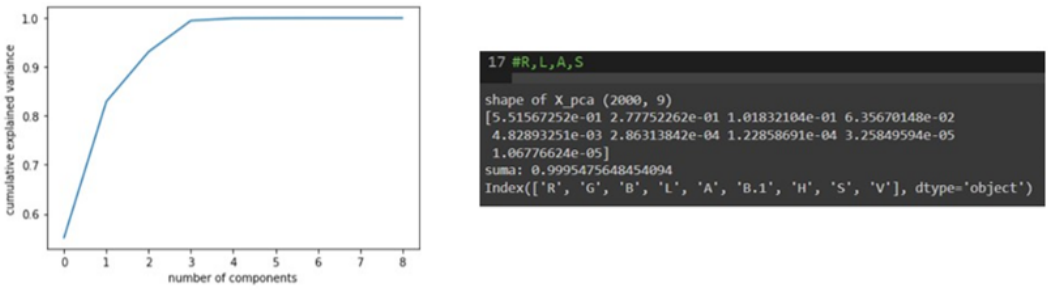


Figura 4.2: Análisis de componentes principales.

diferentes datos y la precisión obtenida. Es importante resaltar que cuando se usaron las nueve bandas de los tres espacios de color se obtienen buenos resultados en los modelos pero si los modelos son entrenados con las bandas resultantes del proceso PCA la red neuronal es la que obtiene una mejor precisión. Se realizaron experimentos con los modelos entrenados como se aprecia en la Figura 4.4, se pudo concluir que la segmentación por espacio de color tiene dificultades cuando existe maleza, sombras, diferencia en tamaños de plantas o estrés hídrico.

RGB	
Modelos	**Precisión**
Classifier tree	85
SVM	90
CNN	85
KNN	91
RF	90

LAB	
Modelos	**Precisión**
Classifier tree	82
SVM	89
CNN	76
KNN	90
RF	90

RLAS	
Modelos	**Precisión**
Classifier Tree	84
SVM	89
CNN	91
KNN	90
RF	90

HSV	
Modelos	**Precisión**
Classifier Tree	79
SVM	86
CNN	82
KNN	91
RF	92

RGB-LAB-HSV	
Modelos	**Precisión**
Classifier Tree	84
SVM	89
CNN	89
KNN	91
RF	91

Figura 4.3: Resultado del entrenamiento de modelos predictivos.

En el conjunto de datos capturado se obtuvo el valor promedio de los píxeles que pertenecen a una planta de agave en el espacio de color RGB, donde se pudo concluir que los píxeles que pertenecen a una planta de agave en el espacio de color RGB en promedio son R=129, G=138, B=163. En la Figura 4.5 se aprecia la binarización invertida obtenida con el valor promedio del color de una planta de agave.

Otro de los experimentos que se realizaron fueron con las operaciones morfológicas para eliminar todo lo que no fuera una planta de agave, como se aprecia en la Figura 4.6.

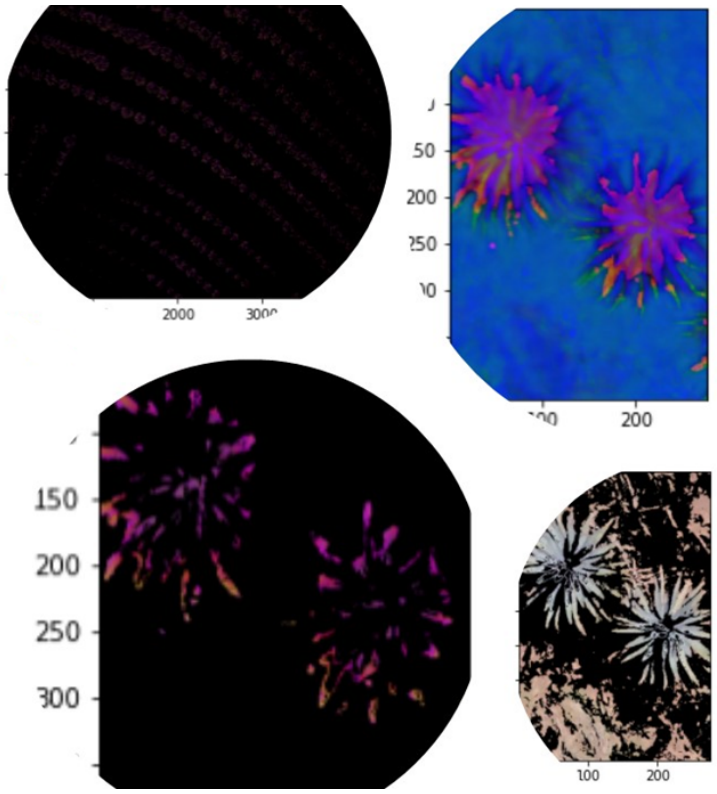


Figura 4.4: Resultados de los modelos predictivos.

Figura 4.5: Proceso de binarización invertida.

Se utilizó la operación morfologíca de apertura primero se aplicó la erosión seguida de la dilatación. El objetivo fue obtener solo las siluetas de las plantas de agave aplicando los siguientes pasos:

- Binarización invertida aplicando el rango de color de una planta de agave (R=129, G=138, B=163)
- Operación morfológica erosión con un Kernel de 7.
- Operación morfológica dilatación con un Kernel de 3.

El uso de operaciones morfológicas requiere un alto costo computacional y existen adversidades como la maleza en cultivos y el traslape de plantas que resulta complicado superar.

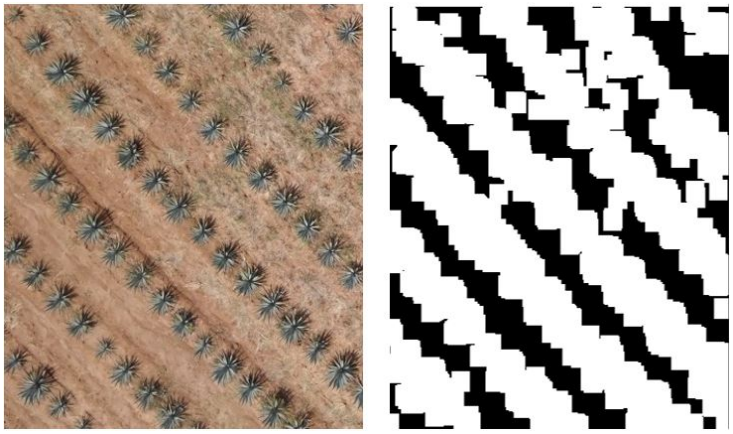


Figura 4.6: Proceso aplicando operaciones morfológicas.

Por otro lado se realizaron pruebas con el algoritmo Cascada Haar en la Figura 4.7 se puede apreciar el resultado del modelo Cascada Haar entrenado.

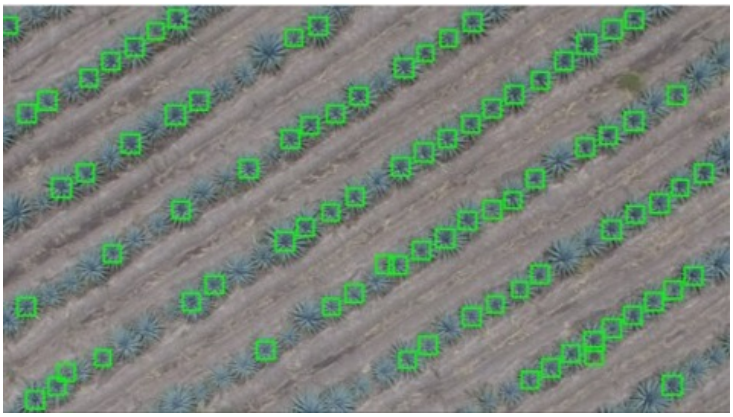


Figura 4.7: Resultado del modelo Cascada Haar.

El algoritmo Cascada Haar requiere de un conjunto grande de imágenes positivas y negativas se debe tener cuidado en respetar el tamaño de las imágenes porque es un valor importante en los parámetros de entrenamiento. El costo computacional para generar el modelo es muy alto, este algoritmo presentó dificultades cuando existía maleza en un cultivo o las plantas estaban sobrepuestas.

4.2. Plan de vuelo y adquisición de imágenes

Para el plan de vuelo y la adquisición de las imágenes se usó el software Pix4DCapture. La configuración de la altura del vuelo fue de 30 metros. El traslape entre cada fotografía fue de 70 %. Como resultado del plan de vuelo, se creó un conjunto de datos con 1204 imágenes con dimensiones de 4608 x 3456; Cada una de las imágenes pesa en promedio 3 MB. Las imágenes adquiridas fueron almacenadas en base a la fecha de la captura y configuraciones de vuelo en google drive. En la Figura 4.8 se muestra un ejemplo de las imágenes adquiridas por el VANT.

4.3. Entrenamiento de la CNN

Una vez capturadas las imagenes, se procedió a crear el ortomosaico utilizando el software Pix4D; Posteriormente, se recortó del ortomosaico la zona de interés usando el software QGIS. El ortomosaico creado se puede apreciar en la Figura 4.9.

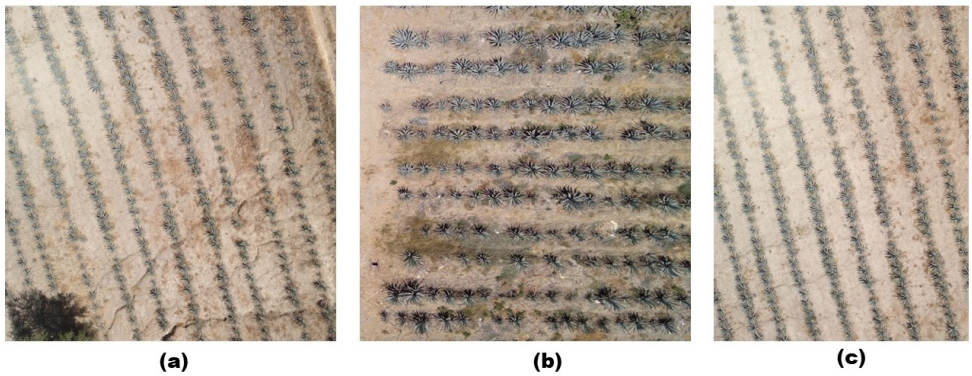


Figura 4.8: Imágenes adquiridas por el VANT.

Figura 4.9: Generación del ortomosaico.

El ortomosaico se dividió en 25 imágenes las cuales se utilizaron para el desarrollo de pruebas en Google Colab, el entorno virtual creado tiene las siguientes características: GPU Tesla T4, 12 GB memoria RAM y 78 GB de almacenamiento. Para mostrar gráficamente la detección, clasificación y conteo de las plantas de agave, se entrenaron diferentes modelos de la CNN. A continuación se realiza la descripción de los parámetros utilizados:

- **imágenes para entrenamiento:** El conjunto de imágenes para el entrenamiento de la CNN esta compuesto de 50 imágenes con dimensiones de 4608 píxeles de ancho

por 3456 píxeles de alto.

- **imágenes para validación:** El conjunto de imágenes para la validación esta compuesto de 20 imágenes con dimensiones de 4608 píxeles de ancho por 3456 píxeles de alto.
- **etiquetas:** La cantidad de etiquetas para el entrenamiento de la CNN fueron 2500 y para la validación se usaron 750.
- **img:** Define el tamaño de la imagen de entrada.
- **lote (batch):** Determina el tamaño de imágenes procesadas.
- **datos:** Se declara la ruta al archivo yaml donde almacenan las rutas absolutas del conjunto de datos.
- **épocas:** Definen el número de épocas de entrenamiento.
- **múltiplo de profundidad:** Controla la profundidad del modelo. La profundidad de YOLOv5s es 0,33 y la profundidad de V5l es 1, lo que significa que el número de cuellos de botella de YOLOv5l es 3 veces mayor que el de YOLOv5s.
- **múltiplo de amplitud:** Determina el número de núcleos de convolución, el ancho de YOLOv5s es 0,5 y el ancho de YOLOv5l es 1, lo que significa que el número de núcleos de convolución de YOLOv5s es la mitad de la configuración predeterminada.
- **modelo:** En YOLOv5 se puede seleccionar un modelo previamente entrenado para comenzar a personalizarlo y realizar pruebas; En este trabajo seleccionamos YOLOv5s, el modelo más pequeño y rápido disponible.

La Tabla 4.1, corresponde a la configuración de los parámetros del modelo que demostró mejores resultados.

4.3.1. Detección y conteo de plantas de agave.

Con el fin de analizar los resultados del método propuesto, en esta subsección se proponen tres casos: (1) imágenes con agaves de tamaño uniforme con maleza abundante, (2) imágenes con agaves de diferentes tamaños, maleza, sombras, objetos que no son plantas de agave y (3) agaves de diferentes tamaños y maleza, objetos que no son plantas

Tabla 4.1: Parámetros de la CNN.

Número	**Parámetro**	**Valor**
1	imágenes de entrenamiento	50
2	imágenes para validación	20
3	etiquetas de entrenamiento	2500
4	etiquetas para validación	750
5	img	1200
6	lote (batch)	4
7	datos	/MyDrive/YOLOv5 /customdata.yaml
8	épocas de entrenamiento	300
9	múltiplo de profundidad	1.0
10	múltiplo de amplitud	1.0
11	Modelo	yolov5s.yaml

de agave. Para medir la calidad de los resultados se utilizó la métrica mAP, que combina los conceptos de precisión, precisión promedio, sensibilidad, intersección sobre la unión.

Caso 1: agaves de tamaño uniforme, separados y con maleza abundante. La Figura 4.10 muestra los resultados del algoritmo propuesto para el caso donde los agaves son de tamaño uniforme, no existe una separación clara entre cada agave, y hay presencia de maleza abundante. Se realizó el conteo de 35 plantas de agave, 2 plantas de agaves no fueron detectadas porque solo se aprecian las hojas, la barda no deja tener una vista completa de las plantas.

Caso 2: agaves de tamaño uniforme, separados con maleza, sombras y objetos que no son plantas de agave. La Figura 4.11 muestra los resultados del algoritmo propuesto para el caso donde los agaves son de tamaño uniforme, existe una separación clara entre cada agave hay presencia de maleza sombras y objetos que no son plantas de agave. El modelo entrenado detectó 52 plantas de agave de 49 existentes, las tres plantas de agave detectadas son hijuelos plántulas que la misma planta madre produce a lo largo de su vida para su reproducción.

Caso 3: agaves de tamaño uniforme, separados con poca maleza, objetos que no son

Figura 4.10: Cultivo con agaves de tamaño uniforme con maleza abundante.

plantas de agave. La Figura 4.12 muestra los resultados del algoritmo propuesto para el caso donde los agaves son de tamaño uniforme, existe una separación clara entre cada agave hay poca presencia de maleza y tiene objetos que no son plantas de agave. El modelo entrenado detecto las 245 plantas de agave.

La metodología propuesta en esta investigación muestra una mejora significativa respecto a trabajos previos en el conteo de plantas de agave. En la investigación realizada por Corona (2019) presentaron un algoritmo de aprendizaje no supervisado para la identificación y el conteo de plantas de agave evaluaron su algoritmo con el coeficiente Silhouette logrando >0.5. Se identificaron áreas de oportunidad para mejorar su algoritmo como el problema con las sombras de árboles y maleza abundante presentes en los cultivos de agave como se puede apreciar en la Figura 4.13 .

El algoritmo de YOLOv5 es capaz de superar adversidades como las sombras de árboles y maleza presentes en cultivos de agave, usando la imagen 4.13 se obtuvo la detección y conteo de 51 plantas de agave, como se puede apreciar en la Figura 4.14.

En el método presentado por Flores et al. (2021) para el conteo de plantas de agave

Figura 4.11: Cultivo con agaves de tamaño uniforme con sombras.

Figura 4.12: El modelo es funcional en condiciones adversas.

combinan el algoritmo de Cascada Haar y una CNN obtuvieron una precisión 96 %. Algunos problemas reportados fueron el alto costo computación, la diferencia de tamaños en plantas, el solapamiento de plantas, en la Figura 4.15 obtienen la detección y conteo de 128 plantas de agave.

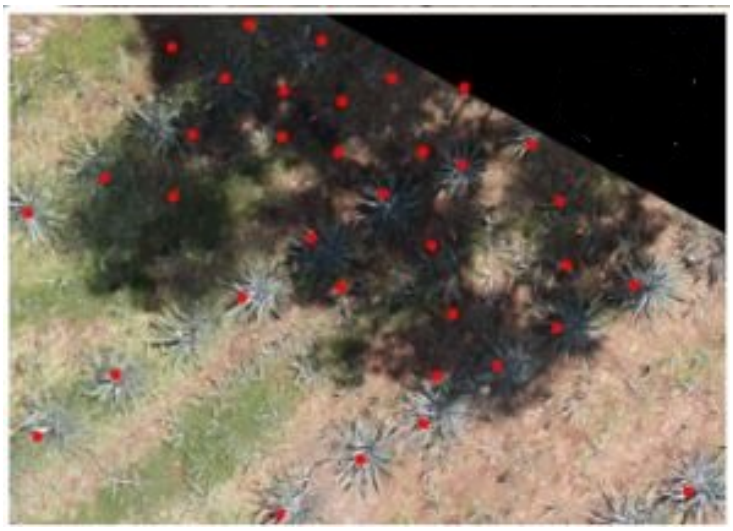


Figura 4.13: Cultivo de agave con sombras y maleza. Fuente: Corona (2019)

Figura 4.14: Detección y conteo de plantas de agave con sombras y maleza usando YOLOv5.

Con el algoritmo YOLOv5 entrenado usando la Figura 4.15 se realizó la detección y el conteo de 129 plantas de agave, como se puede apreciar en la Figura 4.16.

Para comparar el rendimiento del algoritmo con la CNN YOLOv5 con otros métodos, se aplicó la detección y conteo usando la Figura 4.15, aplicando el modelo entrenado con Cascade Haar en los resultados preliminares y los resultados obtenidos por Flores et al. (2021), aplicando las métricas de Precisión, Sensibilidad y Medida F en la Tabla 4.2 se aprecia el resultado del rendimiento del método propuesto con la CNN YOLOv5.

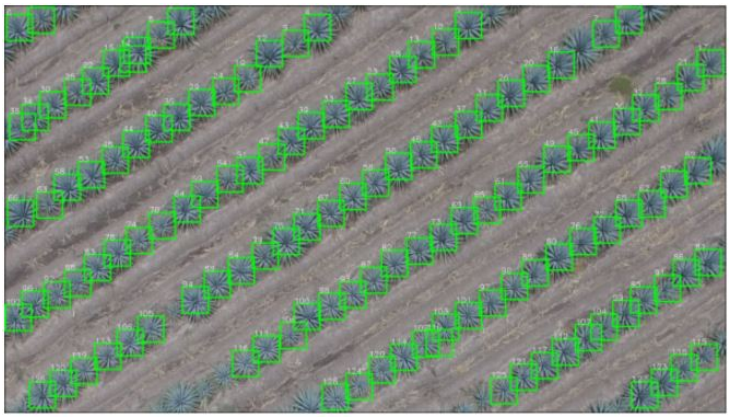


Figura 4.15: Conteo de plantas usando el algoritmo Cascada Haar y una CNN. Fuente: Flores et al. (2021)

Figura 4.16: Conteo de plantas usando YOLOv5.

4.3.2. Validación de los datos

Para la validación de los resultados, se utilizó la métrica mAP usando un umbral determinado de IoU de 0.50 y múltiples umbrales iniciando con un IoU de 0.50 con incrementos de 0.05 hasta 0.95. En la Figura 4.17 se puede observar que la curva Precisión/Sensibilidad es una buena ya que la precisión se mantiene estable en proporción a la sensibilidad con un IoU de 0.5, en la Figura 4.18 podemos observar que la confianza del modelo en proporción a su precisión. Un detector de objetos se considera bueno si su precisión se mantiene alta a medida que aumenta la sensibilidad, lo que significa que, si varía el umbral de confianza, la precisión y la recuperación seguirán siendo altas.

Tabla 4.2: Rendimiento del algoritmo propuesto.

	Sensibilidad	**Precisión**	**Medida F**
Cascada Haar	0.63	0.96	0.76
Cascada Haar y CNN Flores et al. (2021)	0.98	0.95	0.96
Método propuesto	0.99	1.0	0.99

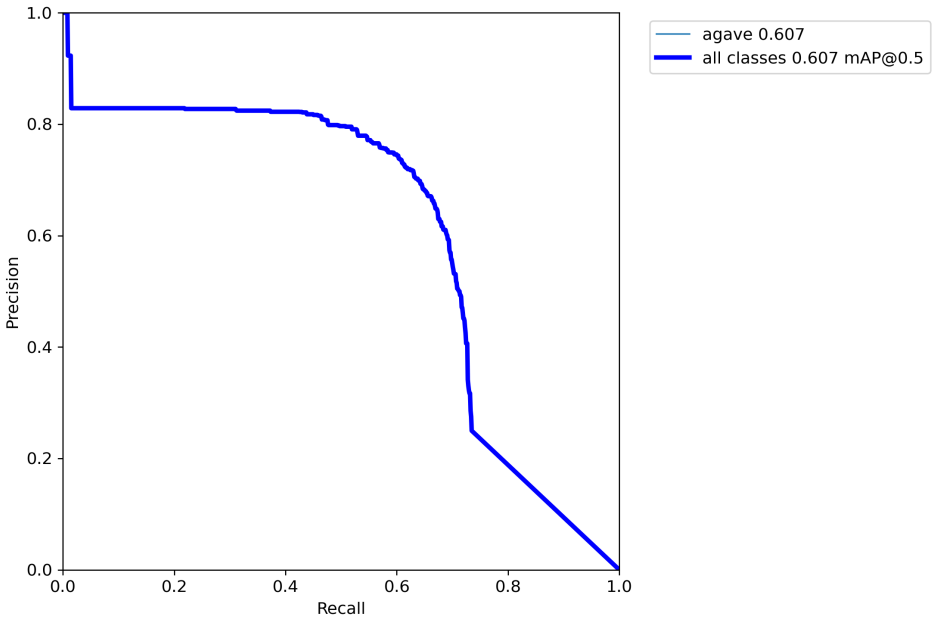


Figura 4.17: Curva Precisión/Sensibilidad.

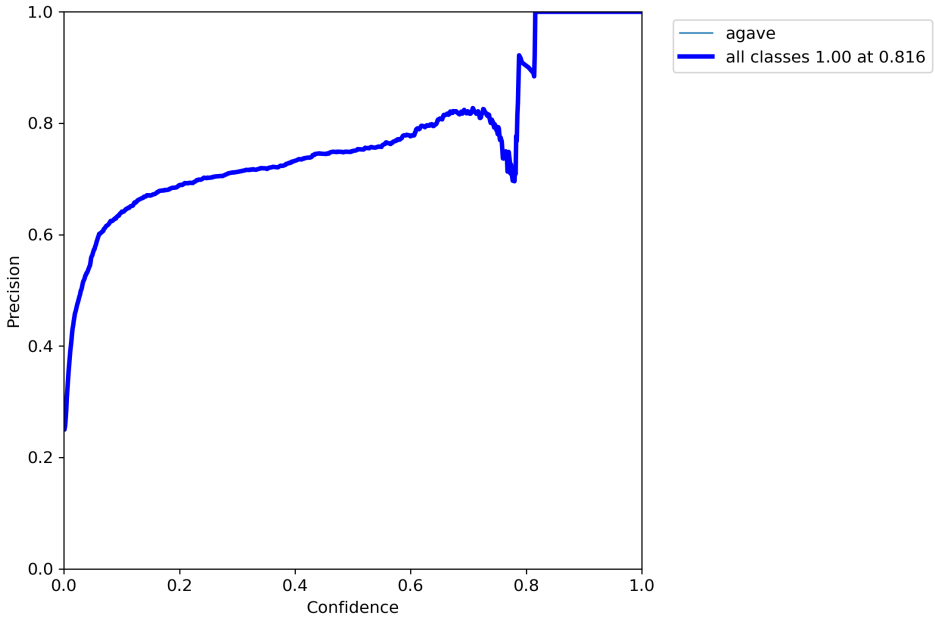


Figura 4.18: Confianza del modelo.

Capítulo 5

Conclusiones y trabajos futuros

5.1. Conclusiones

Se propuso un algoritmo basado en la CNN YOLOv5 para la detección y conteo de plantas de agave usando imágenes aéreas tomadas desde un vehículo aéreo no tripulado. Se realizaron experimentos diferentes a la metodología propuesta como la segmentación por espacios de color, morfología matemática, algoritmo de cascada Haar, fueron descartados por encontrar diferentes adversidades descritas en los resultados.

Se diseñó una metodología compuesta por cuatro etapas, las cuales son: plan de vuelo y adquisición de imágenes, preprocesamiento de las imágenes, entrenamiento de la CNN, detección y conteo de plantas de agave. Con el fin de probar el algoritmo propuesto, se propusieron tres escenarios con diferentes grados de dificultad obteniendo buenos resultados.

La detección y conteo de plantas de agave usando la CNN YOLOv5 fue evaluada con la métrica mAP, la CNN entrenada alcanzó una mejor mAP@0.5 de 0.602. La medida F1 que permite una evaluación del modelo sobre las puntuaciones de confianza alcanzó el valor más alto de 0,67 con una confianza de 0,31.

Se identificaron hiperparametros en la arquitectura de la CNN YOLOv5 que permiten optimizar diferentes procesos como el entrenamiento, validación y la detección de plantas de agave.

Se realizó la comparación del rendimiento de la metodología propuesta en esta tesis, con otros métodos de la literatura usando la misma imagen. El algoritmo desarrollado Flores et al. (2021) logra el conteo de 128 plantas de agave de 130 existentes. Como resultado en esta tesis usando la CNN YOLOv5 se logro contar 129 plantas de agave de las 130

existentes. El algoritmo cascada Haar entrenado en nuestros experimentos preliminares logra el conteo de 83 de las 130 plantas de agave.

Usando la métrica F1 el algoritmo propuesto logra obtener un 99 %. La CNN YOLOv5 entrenada fue capaz de superar adversidades como la maleza, sombras, objetos que no son plantas de agave y la sobreposición en plantas.

En esta investigación se ha presentado un método para la detección de plantas de agave usando técnicas de procesamiento de imágenes y aprendizaje profundo, donde se utiliza la CNN YOLOv5, para la detección y conteo de plantas de agave y se realiza la configuración de parámetros que se adaptan mejor a la resolución del problema específico.

Finalmente, es posible concluir que tanto la hipótesis como los objetivos establecidos en el presente trabajo se cumplieron.

5.2. Trabajo futuro

Para mejorar el algoritmo para la detección y conteo de plantas de agave se hacen las siguientes recomendaciones:

- Profundizar en la arquitectura de la CNN YOLOv5 para personalizar parte del código y optimizar resultados.
- Incrementar el conjunto de datos con imágenes adquiridas en diferentes horas del día.
- Realizar la instalación de YOLOv5 en un entorno local para tener un mejor control en los experimentos y superar adversidades con la compatibilidad de las librerías.
- Usar un algoritmo de arreglos de cobertura amplia para encontrar la mejor combinación de variables en la arquitectura.

Referencias

Arista-Jalife, A., Calderón-Auza, G., Fierro-Radilla, A., and Nakano, M. (2017). Clasificación de imágenes urbanas aéreas: Comparación entre descriptores de bajo nivel y aprendizaje profundo. *Información tecnológica*.

Becerra, J. T. (2012). Redes neuronales. *CUCEI*.

Berzal, F. (2019). *Redes neuronales & deep learning: Volumen II*. Independently published.

Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection.

Calvario, G., Alarcón, T. E., Dalmau, O., Sierra, B., and Hernandez, C. (2020). An agave counting methodology based on mathematical morphology and images acquired through unmanned aerial vehicles. *Sensors*, 20(21):6247.

Castro, W., Oblitas, J., De-La-Torre, M., Cotrina, C., Bazan, K., and Avila-George, H. (2019). Classification of cape gooseberry fruit according to its level of ripeness using machine learning techniques and different color spaces. *IEEE Access*.

Corona, J. E. P. (2019). Algoritmo para el conteo de agaves usando imágenes aéreas tomadas desde un vehículo aéreo no tripulado. Master's thesis, Instituto Tecnológico Ciudad Guzmán.

Cortés, C. A. (2017). Herramientas modernas en redes neuronales: la librería keras. Master's thesis, Universidad Autonoma de Madrid.

CRT (2022). `https://www.crt.org.mx/EstadisticasCRTweb/`, 25 de Abril de 2022.

de la Vega, M. A. G. and de Teresa de Oteyza, L. (2015). Aportaciones matemáticas. In *Memorias de la sociedad matemáticas mexicana*.

Fan, Z., Lu, J., Gong, M., Xie, H., and Goodman, E. D. (2018). 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):876–887.

Flores, D., González-Hernández, I., Lozano, R., Vazquez-Nicolas, J. M., and Toral, J.

L. H. (2021). Automated agave detection and counting using a convolutional neural network and unmanned aerial systems. *Drones*, 5(1):4.

Galipienso, M. I. A. (2003). *Inteligencia artificial: modelos, técnicas y áreas de aplicación*. Thomson Ediciones España.

Gil, P., Torres, F., and Ortiz Zamora, F. G. (2004). Detección de objetos por segmentación multinivel combinada de espacios de color. *Federación Internacional de Automatización. Comité Español de Automática*.

Hernández, R. R. (2021). Uso de los drones o vehículos aéreos no tripulados en la agricultura de precisión. *Revista Ingeniería Agrícola*.

Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., NanoCode012, Kwon, Y., TaoXie, and Fang, J. (2022). ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference.

Keller, A. (2022). Una metodología de aterrizaje robusta y precisa para drones sobre objetivos en movimiento. *Sensors*.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*.

Li, W., Fu, H., Yu, L., and Cracknell, A. (2016). Deep learning based oil palm tree detection and counting for high-resolution remote sensing images. *Remote Sensing*, 9(1):22.

Liu, S., Qi, L., Qin, H., Shi, J., and Jia, J. (2018). Path aggregation network for instance segmentation.

López, R. F., Fernandez, J. M., and Fernández, J. M. F. (2016). *Las redes neuronales artificiales*. Netbiblo.

Marcos, A. G., de Pisón Ascacíbar, F. J. M., and Espinoza;, A. V. P. (2006). *Técnicas y Algoritmos Básicos de Visión Artificial*. Universidad de La Rioja.

Massiris, M., Delrieux, C., and Fernández, J. Á. (2020). Detección de equipos de protección personal mediante red neuronal convolucional YOLO. In *Actas de las XXXIX Jornadas de Automática, Badajoz, 5-7 de septiembre de 2018*. Universidade da Coruña. Servizo de Publicacións.

Mathew, M. P. and Mahesh, T. Y. (2021). Leaf-based disease detection in bell pepper plant using YOLO v5. *Signal, Image and Video Processing*, 16(3):841–847.

Meré, J. M. (2008). *Técnicas de visión por computador para la reconstrucción en tiempo real de la forma 3d de productos laminados*. PhD thesis, Universidad de Oviedo.

Minolta., K. (2022). `tinyurl.com/4tk8nt7b`, 25 de Abril de 2022.

Moreno Diaz, O. (2022). Exactitud y precisión.

Mubin, N. A., Nadarajoo, E., Shafri, H. Z. M., and Hamedianfar, A. (2019). Young and mature oil palm tree detection and counting using convolutional neural network deep learning method. *International Journal of Remote Sensing*, 40(19):7500–7515.

Mínguez, T. D. (2021). *Visión artificial*. Book Publishing Inc.

Oh, S., Kim, Y.-J., Park, Y.-T., and Kim, K.-G. (2021). Automatic pancreatic cyst lesion segmentation on EUS images using a deep-learning approach. *Sensors*.

Padilla, R., Netto, S. L., and da Silva, E. A. B. (2020). A survey on performance metrics for object-detection algorithms. In *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 237–242.

Pino, R., Gomez, A., and de Abajo, N. (2001). *Introducción a la ingeniería Artificial: Sistemas Expertos, Redes Neuronales Artificiales y Computación Evolutiva*. Book Publishing Inc.

Pérez, M. A. A. (2009). *Espacios de Color RGB, HSI y sus Generalizaciones a n-Dimensiones*. PhD thesis, Instituto Nacional de Astrofísica, Óptica y Electrónica.

Rahnemoonfar, M. and Sheppard, C. (2017). Deep count: Fruit counting based on deep simulated learning. *Sensors*.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2015). You only look once: Unified, real-time object detection.

Redmon, J. and Farhadi, A. (2016). Yolo9000: Better, faster, stronger.

Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement.

Ribera, J., Chen, Y., Boomsma, C., and Delp, E. J. (2017). Counting plants using deep learning. In *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE.

Rodríguez, A. S. and Gómez, R. R. (2020). Evaluación de algoritmos de machine learning para conducción. *Sensors*.

Rouhiainen, L. P. (2018). *Inteligenciaartificial*. 2018.

Sarabia, R., Aquino, A., Ponce, J. M., López, G., and Andújar, J. M. (2020). Automated identification of crop tree crowns from UAV multispectral imagery by means of morphological image analysis. *Remote Sensing*, 12(5):748.

Serra, J. and Soille, P. (2012). *Mathematical morphology and its applications to image processing*. Springer Science & Business Media.

Serrano, A. S. (2017). Yolo object detector for onboard drivingimages. Technical report, UNIVERSITAT AUTONOMA DE BARCELONA.

Serrano, Y. E. L. (2014). *UF1246 - Tratamiento y edición de fuentes para productos audiovisuales multimedia*. Elearning S.L.

Simeone, O. (2018). A very brief introduction to machine learning with applications to communication systems. *IEEE Transactions on Cognitive Communications and Networking*.

Sobrino, J. A. (2000). *Teledetección*. Universidad de Valencia.

support.parrot.com (2022). *Parrot bluegrass. En línea, accesado el 06/may/2022 en https://support.parrot.com/us/support/products/parrot-bluegrass.*

Titano, J. J., Badgeley, M., Schefflein, J., Pain, M., Su, A., Cai, M., Swinburne, N., Zech, J., Kim, J., Bederson, J., Mocco, J., Drayer, B., Lehar, J., Cho, S., Costa, A., and Oermann, E. K. (2018). Automated deep-neural-network surveillance of cranial images for acute neurologic events. *Nature Medicine*.

Villaseñor, E. B. (2018). Eficiencia de las redes neuronales para la toma de decisiones en el sector agrícola análisis exploratorio. In *Eficiencia de las redes neuronales*.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001.*

Wong, Z. Y., Chew, W. J., and Phang, S. K. (2020). Computer vision algorithm development for classification of palm fruit ripeness. In *13TH INTERNATIONAL ENGINEERING RESEARCH CONFERENCE (13TH EURECA 2019)*. AIP Publishing.

Wu, H., Wiesner-Hanks, T., Stewart, E., DeChant, C., Kaczmar, N., Gore, M., Nelson, R., and Lipson, H. (2019a). Autonomous detection of plant disease symptoms directly from aerial imagery. *tppj*, 2.

Wu, J., Yang, G., Yang, X., Xu, B., Han, L., and Zhu, Y. (2019b). Automatic counting of in situ rice seedlings from UAV images based on a deep fully convolutional neural network. *Remote Sensing*, 11(6):691.

Zamora, F. G. O. (2002). *Procesamiento morfológico de imágenes en color: aplicación a la reconstrucción geodésica*. PhD thesis, Universidad de Alicante. Departamento de Física, Ingeniería de Sistemas y Teoría de la Señal.

Zhang, L., Xia, G.-S., Wu, T., Lin, L., and Tai, X. C. (2016). Deep learning for remote sensing image understanding. *Journal of Sensors*.

Zhong, Y., Gao, J., Lei, Q., and Zhou, Y. (2018). A vision-based counting and recognition system for flying insects in intelligent agriculture. *Sensors*, 18(5):1489.

Zhou, C., Ye, H., Hu, J., Shi, X., Hua, S., Yue, J., Xu, Z., and Yang, G. (2019). Automated counting of rice panicle by applying deep learning model to images from unmanned aerial vehicle platform. *Sensors*, 19(14):3106.

Zortea, M., Nery, M., Ruga, B., Carvalho, L. B., and Bastos, A. C. (2018). Oil-palm tree detection in aerial images combining deep learning classifiers. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE.

Anexos

Apéndice A

Publicación derivada de este trabajo de tesis

La Revista lbérica de Sistemas y Tecnologías de Información esta indexada en las principales bases de datos internacionales, entre las cuales destacan Web of Science y Scopus.

risti
Revista Ibérica de Sistemas e Tecnologias de Informação
Revista Ibérica de Sistemas y Tecnologías de Información

Recebido/Submission: xx/12/2021
Aceitação/Acceptance: xx/02/2021

Conteo de plantas de agave usando redes neuronales convolucionales e imágenes adquiridas desde un vehículo aéreo no tripulado

Omar Hernández-Calvario[1], Frida Florián[1], María Guadalupe Sánchez[1], Himer Ávila-George[2]

omar.calvario@cusur.udg.mx; m20291046@cdguzman.tecnm.mx; himer.avila@academicos.udg.mx; msanchez@itcg.edu.mx

[1] Departamento de Sistemas y Computación, TecNM - Instituto Tecnológico de Ciudad Guzmán, Ciudad Guzmán 49100, Jalisco, México.

[2] Departamento de Ciencias Computacionales e Ingenierías, Universidad de Guadalajara, Ameca 46600, Jalisco, México.

DOI: 10.17013/risti.45.64–76

Figura A.1: Publicación de articulo.

Apéndice B

Estancia académica

Durante la estancia en el Centro Universitario de los Valles con apoyo del Dr. Miguel Ángel de la Torre Gómora, se desarrollaron actividades relacionadas con la investigación.

UNIVERSIDAD DE GUADALAJARA
Centro Universitario de los Valles
Departamento de Ciencias Computacionales e Ingenierías

Asunto: CARTA DE FINALIZACIÓN

M.C. ADA MABEL VÁZQUEZ PAZ
COORDINADORA DE LA MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN
INSTITUTO TECNOLÓGICO DE CIUDAD GUZMÁN
P R E S E N T E:

Por este conducto me permito informar a usted que el alumno **Omar Hernández Calvario** con número de control **M20291047**, que actualmente cursa el **cuarto Semestre** de **"la Maestría en Ciencias de la Computación"**, cumplió satisfactoriamente su **Estancia en Linea, del 1 de Marzo al 1 de abril del 2022.**

Durante su estancia en esta institución el alumno continuó con el desarrollo de las actividades pertinentes a su tema de tesis **"Conteo de plantas de agave usando redes neuronales convolucionales con imágenes adquiridas de un vehículo aéreo no tripulado"**, dirigido y supervisado por el Dr. Miguel Ángel de la Torre Gómora.

Sin otro particular, me despido de usted con un cordial saludo.

A T E N T A M E N T E
"Piensa y Trabaja"
"2022, Guadalajara, Hogar de la Feria Internacional del Libro y Capital Mundial del Libro"
Ameca, Jal., 9 de junio de 2022

Dr. Miguel Ángel de la Torre Gómora
miguel.dgomora@academicos.udg.mx
Profesor investigador Asociado "C"
Departamento de Ciencias Computacionales e Ingenierías.
Centro Universitario de Los Valles, Universidad de Guadalajara

Carretera Guadalajara-Ameca Km.45.5. C.P. 46600. Apartado Postal. N° 200
Ameca, Jalisco, Mexico. Tels. 01 [375] 75 80 500/148 Ext. 47208
www.cuvalles.udg.mx

Figura B.1: Carta de finalización de estancia.