
Detección de FOD por medio de inteligencia artificial con inspecciones por UAV

Presentado por:

Duvan Nicolas Forero Aranzalez

Andrés Sebastián Albarracín Carranza



LOS LIBERTADORES
FUNDACIÓN UNIVERSITARIA

Fundación Universitaria Los Libertadores

Facultad de Ingeniería y Ciencias Básicas Programa de

Ingeniería Aeronáutica

Bogotá D.C, Colombia

2022

Detección de FOD por medio de inteligencia artificial con inspecciones por UAV

Presentado por:

Duvan Nicolas Forero Aranzalez

Andrés Sebastián Albarracín Carranza

En cumplimiento parcial de los requerimientos para
optar por el título de

Ingeniero Aeronáutico

Dirigida por

Cristian Lozano Tafur

Codirector

Didier Aldana Rodríguez

Luis Felipe Gutierrez Marcantoni

Presentada a

Programa de Ingeniería

Aeronáutica Fundación

Universitaria Los Libertadores

Bogotá D.C, Colombia



LOS LIBERTADORES

FUNDACIÓN UNIVERSITARIA

Notas de aceptación



Firma de presidente del jurado

LOS LIBERTADORES
FUNDACIÓN UNIVERSITARIA

Firma del jurado

Firma del jurado

Bogotá DC, Enero 2022



LOS LIBERTADORES

FUNDACIÓN UNIVERSITARIA



LOS LIBERTADORES
FUNDACIÓN UNIVERSITARIA

Las directivas de la Fundación Universitaria Los Libertadores, los jurados calificadores y el cuerpo docente no son responsables por los criterios e ideas expuestas en el presente documento. Estos corresponden únicamente a los autores y a los resultados de su trabajo.



LOS LIBERTADORES

FUNDACIÓN UNIVERSITARIA

Agradecimientos

La culminación de mi carrera la cual es un homónimo del último proyecto que presentare de mi carrera profesional es un símbolo de logro y no hay logro que pueda ser más agradecido que a mí mismo por el esfuerzo que he dedicado a sacar adelante mi carrera profesional desde el día cero, pasando por grandes adversidades y no dejarme decaer ante ellas, siempre con la frente en alto poniendo el empeño suficiente para cumplir esta meta. Siendo un camino propio en el cual no puedo dejar de lado el agradecimiento a las personas que han estado ahí apoyándome a salir adelante caminando de mi lado en este camino desde mucho antes de iniciarlo, a mis padres que día a día se esfuerzan en dar lo mejor de ellos para poder verme salir adelante, a ellos les doy mi más grande agradecimiento de culminar este camino.

Sumado a ellos no puedo ignorar a mis hermanos que junto a mis padres han estado presentes con su acompañamiento y apoyo en generarme el empuje suficiente para culminar mis proyectos de vida y así mismo mi título de grado contando con ellos en todo proyecto en que me he visto envuelto.

Agradeciendo de mismo modo a esos compañeros como Andrés que iniciaron este camino conmigo y que al recorrerlo se han convertido en amistades, ellos con quienes día a día conviví y siempre buscamos apoyarnos en dar lo mejor de nosotros y poder cumplir nuestras metas de vida, amistades que espero conservar después de culminar este camino llamado Ingeniería Aeronáutica el cual abrirá muchos más. Incluyendo así mismo a aquellos profesores que en verdad dieron de sí mismos por brindar sus conocimientos en pro de los estudiantes y generaron no solo conocimiento si no pasión y búsqueda por lo desconocido.

Siendo así de último doy gracias a las ciencias e ingeniería por mostrarme el mundo de invenciones y conocimientos que dan explicación a el funcionamiento de muchas áreas de este mundo dejándome ser parte de ellos generando ese ingenio y ganas de seguir su camino en mi proyecto de vida.

Duvan Nicolas Forero Aranzalez

El cumplimiento de un logro no viene solo, siempre alguien o algunas personas, se cruzarán en nuestras vidas, dejando una huella que se marca para siempre, por tal motivo gracias Dios, por ser mi guía y brindarme la luz para recorrer cada camino, por permitirme comprender y obtener la sabiduría necesaria para culminar este proyecto, gracias a Dios se hace realidad un sueño, que permite se abran nuevas puertas y nazcan nuevos proyectos de vida.

Doy gracias a Dios, por mis padres, seres de amor y bondad, quienes me han guiado por los caminos de la vida, gracias a mis padres, por dedicar su vida a enseñarme y educarme de la mejor manera, gracias a ellos logro un sueño bendecido por Dios, un sueño que no es solo personal, es un sueño familiar que hoy podemos decir como familia gracias, Dios por tan inmensa bendición. Gracias padres, por apoyarme cada día, por creer en mí, también a mi hermano compañero de aventuras quien ha marcado una gran parte en el cumplimiento de mis sueños, gracias hermanito por brindarme tu mano con amor y sinceridad.

Gracias a Dios, por permitirme trabajar junto a mi compañero Duván Nicolas Forero, quien se esmeró por dar cumplimiento a este proyecto, persona que logra demostrar la calidad de ser que es, gracias a Dios juntos logramos culminar este proyecto, no fue siempre o sencillo, pero gracias a Dios pudimos lograr juntos esta meta. Gracias por su fidelidad a esta amistad.

LOS LIBERTADORES
FUNDACIÓN UNIVERSITARIA

Andres Sebastián Albarracín Carranza

Resumen

Los FOD son un problema que se da en la aviación general tanto en la aviación comercial como militar, se tiene estimado que los objetos extraños que pueden afectar a las aeronaves le cuestan al sector aeronáutico anualmente más de 4.000 millones de dólares tanto en incidentes como en mantenimiento, teniendo en cuenta principalmente los gastos en mantenimiento. Viendo esta gran problemática que se genera mundialmente en el sector aeronáutico se han implementado distintas alternativas de prevención, unas poco eficientes, y otras económicamente poco viables para ciertos aeropuertos, por esto se busca implementar las tecnologías más modernas que hay actualmente, con la inteligencia artificial se puede generar detección de estos FOD entrenando una red neuronal convolucional de aprendizaje profundo para que por medio de un dron automatizado con su plan de vuelo establecido pueda realizar las inspecciones mucho más eficientes y rápidas a las inspecciones actuales que se enfocan en el uso de personal humano para detectar estos objetos. Teniendo este proyecto un resultado de detección de objetos con una CNN del 83% con un entrenamiento de 2236 imágenes con 7.838 etiquetas. Validándose en pruebas de campo, recomendando que tipo de drones que ofrece el mercado podría ser utilizado para este análisis y de igual modo indicando la altitud de 2 a 5 metros necesaria para tener una precisión adecuada de por lo menos 58%, siendo privado con su plan de vuelo en base al aeropuerto internacional El Dorado de Bogotá.

Palabras clave: FOD, CNN, Dron, Seguridad operacional

Abstract

The FOD are a problem that occurs in general aviation, both in commercial and military aviation. It is estimated that foreign objects that can affect aircraft cost the aviation sector more than 4 billion dollars annually in both incidents and accidents, mainly taking into account maintenance costs. Seeing this great problem that is generated worldwide in the aviation sector have been implemented various alternatives for prevention, some inefficient, and other economically unviable for certain airports, so it seeks to implement the most modern technologies that are currently available, with artificial intelligence can generate detection of these FOD training a deep learning convolutional neural network for through an automated drone with its established flight plan can perform inspections much more efficient and faster than the current inspections that focus on the use of human personnel to detect these objects. Having this project a result of detection of objects with a CNN of 83% with a training of 2236 images with 7,838 labels. Validating in field tests, recommending what type of drones offered by the market could be used for this analysis and likewise indicating the altitude of 2 to 5 meters needed to have an adequate accuracy of at least 58%, being private with its flight plan based on the international airport El Dorado in Bogota.

Keywords: FOD, CNN, Drone, Operational Safety

Contenido

	Pág.
Resumen	XI
Lista de figuras	XV
Lista de tablas	XVII
Lista de graficas	XVIII
Glosario de abreviaturas	XIX
Introducción	1
Antecedentes	3
1. Planteamiento de problema	9
1.1 Objetivo.....	11
1.1.1 Objetivo General.....	11
1.1.2 Objetivos Específicos	11
1.2 Justificación	11
2. Marco Referencial	13
2.1 Marco Teórico	13
2.1.1 Seguridad Operacional.....	13
2.1.2 SMS.....	13
2.1.3 FOD.....	14
2.1.4 Inspección FOD	14
2.1.5 Drones.....	17
2.1.6 Tipos de drones	18
2.1.7 Análisis de imágenes mediante IA.....	19
2.1.8 Inteligencia artificial.....	19
2.1.9 Machine Learning.....	20
2.1.10 Deep Learning	20
2.1.11 Redes neuronales.....	20
2.1.12 Red neuronal artificial	21
2.1.13 Función de activación	23
2.1.14 Red neuronal convolucional.....	23
2.1.15 Detección de objetos	29
2.1.16 Modelos de mono-etapa.....	29
2.1.17 Modelos de dos etapas	30
2.1.18 Modelos Deep CNN	30

2.1.19 Métricas de modelos de detección de objetos.....	32
2.1.20 Herramientas de procesamiento.....	33
2.2 Marco Legal.....	34
2.2.1 Normativas FOD.....	35
2.2.2 Normativas UAV.....	37
3. Diseño metodológico.....	39
3.1 Tipo de Investigación.....	39
3.1.1 Recopilación de datos e información.....	40
3.1.2 Desarrollo del modelo.....	40
3.1.3 Criterios de dron a usar.....	41
3.1.4 Caracterización del vuelo.....	41
3.1.5 Implementación.....	41
3.2 Delimitación.....	41
3.2.1 Espacial.....	41
3.2.2 Temporal.....	42
3.2.3 Académica.....	42
3.3 Valor de la investigación.....	42
4. Desarrollo del algoritmo de visión artificial.....	43
4.1 Desarrollo del análisis de imágenes.....	43
4.2 Requisitos.....	45
4.3 Procedimiento.....	45
4.3.1 Data Set.....	45
4.3.2 Imágenes.....	46
4.3.3 Etiquetado.....	47
4.3.4 Entrenamiento.....	51
5. Criterios de selección del dron.....	59
5.1 Características del UAS.....	59
5.2 Recomendación de dron.....	60
6. Caracterización de vuelo.....	61
6.1 Política de operación.....	61
6.2 Planeamiento de Vuelo.....	62
7. Implementación.....	67
7.1 Resultados de entrenamiento de modelo.....	67
7.2 Tiempo de Inspecciones.....	73
8. Conclusiones y recomendaciones.....	77
8.1 Conclusiones.....	77
8.2 Recomendaciones.....	78
A. Anexo: Código solución de entrenamiento del modelo.....	81
Bibliografía.....	113

Lista de figuras

	Pág.
Figura 1 y 2: Ilustración de FOD BOSS y FOD BUSTER	4
Figura 3: FODetect®	5
Figura 4: iferret™	5
Figura 5: Tarsier®	6
Figura 6: FOD Finder™	6
Figura 7: Resultados de detección de ganado.....	7
Figura 8: Resultado de ganadería vacuna.....	7
Figura 9: Prototipo de ala Zagi pa evasión de obstaculos.....	8
Figura 10: FOD.....	14
Figura 11: Inspección de pista.....	16
Figura 12: Causalidad de Modelo del queso suizo	17
Figura 13: Esquema de una unidad de proceso típica	21
Figura 14: Arquitectura de un perceptrón de una capa	22
Figura 15: Arquitectura del perceptrón multicapa	22
Figura 16: Funciones de activación del perceptrón multicapa	23
Figura 17: Ejemplo de red neuronal común con una imagen en escala de grises.	24
Figura 18: Filtro Kernel realizando su función de convolución	26
Figura 19: Reducción pooling de una imagen.....	26
Figura 20: Tipos de reducción Max Pooling y Average Pooling	27
Figura 21: Funcionamiento de una red convolucional.	28
Figura 22: Red neuronal convolucional con capa completamente conectada	28
Figura 23: Ecuación de métrica de Intersection Over Unión (IoU)	32
Figura 24: Predicción IoU.....	32
Figura 25: Diagrama de flujo del proyecto.	39
Figura 26: Recomendación Roboflow para data set de imágenes	46
Figura 27: Imagen de FOD para analisis.	46
Figura 28: Funcionamiento de Download All Images.....	47
Figura 29: MakeSense Carga de imágenes a la plataforma.	48
Figura 30: MakeSense Creación de clases de etiquetas.....	48
Figura 31: Etiquetado de imágenes en MakeSense.	49
Figura 32: Descarga de etiquetas de MakeSense.....	49
Figura 33: Modificación de tamaño de imágenes de entrenamiento por Roboflow.	50
Figura 34: Google Colab – Entrenamiento personalizado con YOLOv5.....	54
Figura 35: Valor máximo de precisión del modelo.....	56
Figura 36: Resultado de imágenes de entrenamiento en la CNN.....	57

Figura 37: Forma de cuadrícula para plan de vuelo	63
Figura 38: Forma de doble cuadrícula para plan de vuelo	63
Figura 39: Forma de tres trayectorias para plan de vuelo.....	64
Figura 40: Cuadrícula simple del plan de vuelo	65
Figura 41: Cuadrícula del plan de vuelo rampa	66
Figura 42: Pruebas de vuelo en la FULL.....	67
Figura 43: Imagen a 1.5 m	68
Figura 44: Imagen a 1.5m	68
Figura 45: imagen a 2m	69
Figura 46: Imagen a 2m.....	69
Figura 47: Imagen a 5m	70
Figura 48: Imagen a 5m.....	70
Figura 49: Imagen a 8m.....	71
Figura 50: Imagen a 10m.....	71

Lista de tablas

	Pág.
Tabla 1: Características de modelos Deep CNN.....	31
Tabla 2: Datos finales de imágenes en Roboflow	51
Tabla 3: Comparación de las versiones de YOLO v5.	53
Tabla 4: Recomendación de drones.....	60
Tabla 5: Plan de vuelo en pista	65
Tabla 6: Plan de vuelo en pista	66
Tabla 7; Precisión de detección de imágenes en dron con respecto a la altura.....	72
Tabla 8: Plan de vuelo en pista con valores de precisión	72
Tabla 9: Plan de vuelo en rampa con valores de precisión.....	73
Tabla 10: Comparativa de tipos de inspecciones	75

Lista de graficas

	Pág.
Gráfica 1: Comparación de versiones de YOLO.....	52
Gráfica 2: Comparación de versiones de YOLOv5.....	53
Gráfica 3: Precisión del modelo con respecto a sus épocas de entrenamiento.	56

Glosario de abreviaturas

Abreviaturas

Abreviatura	Término
<i>2D</i>	Dos dimensiones
<i>3D</i>	Tres dimensiones
<i>4k</i>	Imagen ultra alta definición de casi 4000 píxeles
<i>ACM</i>	Manual aeroportuario certificado
<i>AM</i>	Modulación de amplitud
<i>BB</i>	Caja delimitadora
<i>CNN</i>	Red neuronal convolucional
<i>DPM</i>	Modelo basado en partes deformables
<i>EASA</i>	Agencia europea de seguridad aérea
<i>FAA</i>	Administración federal de aviación
<i>FAR</i>	Reglamentos federales de aviación
<i>FM</i>	Frecuencia modulada
<i>FOD</i>	Objeto extraño que podría producir algún daño
<i>FPS</i>	Fotograma por segundo
<i>GHz</i>	Giga Hertz, mil millones de hercios
<i>HOG</i>	Histogramas de gradientes orientados
<i>IA</i>	Inteligencia artificial
<i>LAR</i>	Reglamentos aeronáuticos de Latinoamérica
<i>LIDAR</i>	Detección y localización de imágenes por láser
<i>MP</i>	Megapíxel, un millón de píxeles
<i>NIR</i>	Infrarrojo cercano
<i>OACI</i>	Organización de aviación civil internacional
<i>OPAIN</i>	Operadora aeropuerto internacional
<i>RAC</i>	Reglamentos aeronáuticos de Colombia

Abreviatura	Término
<i>RGB</i>	Rojo, verde y azul
<i>RNA</i>	Red neuronal artificial
<i>ROI</i>	Región de interés
<i>RPA</i>	Vehículo aéreo controlado de forma remota
<i>RPN</i>	Red de propuestas regionales
<i>SIFT</i>	Transformación de características invariantes de escala
<i>SMM</i>	Manual de gestión de la seguridad operacional
<i>SMS</i>	Sistema de gestión de seguridad operacional
<i>SRVSOP</i>	Sistema regional de cooperación para la vigilancia de la seguridad operacional
<i>SSD</i>	Detector de un solo disparo
<i>SVM</i>	Máquinas de vectores soportados
<i>UAEAC</i>	Unidad administrativa especial de aeronáutica civil
<i>UAS</i>	Sistema aéreo no tripulado
<i>UAV</i>	Vehículo aéreo no tripulado
<i>VANT</i>	Primera ley de la termodinámica
<i>YOLO</i>	Solo vives una vez
<i>DJI</i>	Dà-Jiāng Innovations
<i>VTOL</i>	Despegue y aterrizaje vertical

Introducción

Los objetos extraños que pueden producir un daño a la industria aeronáutica (FOD) son un problema que genera gastos enormes a las compañías, se tiene estimado que superan los 4.000 millones de dólares por año, siendo así un gasto que se da por efectos de incidentes y accidentes por estos objetos (Ukessays, 2020; Vara Castañeda, 2013).

Teniendo en cuenta que son objetos que pueden ser monitoreados por ayudas tecnológicas contemporáneas, en el auge de este inconveniente se han implementado diferentes alternativas de prevención como ha sido los FOD BOSS(Aerosweep Pty Ltd, 2020), FOD BUSTER(The FOD Control Corp., 2020), FODetect(FAA, E. Herricks, Woodworth, & Patterson, 2012), iferret (FAA, E. Herricks, et al., 2011) y FOD Finder (FAA, Herricks, et al., 2011), los cuales son alternativas eficientes, pero a su vez costosas siendo implementados alrededor del mundo enfocándose en Estados Unidos, Asia y Europa, dando resultados favorables en su disminución.

Abarcando la detección de estos objetos se puede ver que nacionalmente no se ha intentado adentrar en este tipo de problemática, esencialmente se han generado accidentes e incidentes por estos FOD como lo ha sido con el avión de Avianca (Grupo de Investigación de Accidentes – GRIAA, 2020) y el DC-3 (Grupo de Investigación de Accidentes – GRIAA, 2018), se presenta constantemente ese riesgo latente de ocasionarse un posible incidente o accidente.

Por esto mismo se plantea generar nuevas alternativas de inspección y prevención de FOD en Colombia teniendo muy presente las alternativas que se usan hoy en día en el país, por esto mismo se tiene como hipótesis generar un sistema de detección de estos objetos con ayudas modernas como lo es la inteligencia artificial mediante aprendizaje profundo implementándolo así mismo en drones de inspección que puedan detectar estos objetos mediante alternativas económicas y eficientes.

Para llevar a cabo este proyecto se ha estructurado en 4 Capítulos principales. En el “Marco referencial” se presentan algunas precisiones teóricas y conceptuales de seguridad operacional con su enfoque en FOD, integrando así la tecnología de inteligencia artificial enfocándose en las redes neuronales convolucionales que son quienes le darán vida al modelo de detección, sin dejar de lado su parte legal en la cual se puede ver qué aspectos legales entrarían en el tema y cuáles son las restricciones que se generarían para su implementación; En el capítulo del “Desarrollo del modelo” se hará el procedimiento de entrenamiento de la red neuronal que se encargara de identificar los objetos en superficies de posible aparición dando su porcentaje de precisión; pasando por el capítulo de “Criterios de selección del dron” se hace un análisis de los tipos de drones que hay en el mercado y cuales se han aplicado en proyectos similares seleccionando el tipo de dron a usar y recomendando cuales del mercado actual podrían ser usados. Finalizando por el capítulo de “Caracterización de vuelo” en donde se hace una prueba de vuelo con un dron y se realiza prueba el modelo de detección en esta prueba definiendo así la mejor opción de uso para el dron.

Antecedentes

El inicio de los problemas con los objetos extraños que pueden producir un daño llamados FOD se dio con sus accidentes, que nacieron en sí con la misma invención de los aviones. Con el pasar de los años se han producido incidentes y accidentes a costa de estos objetos extraños y constantemente se generaban sistemas de control con respecto a estos objetos, manteniendo las áreas de labor constantemente limpias, todo fuera de residuos que pudieran afectar de algún modo. Todos estos factores se controlaban a la hora de estar cerca del sector, pero no en gran magnitud.

Gracias a los grandes y fuertes accidentes que se han generado en el mundo es que los FOD se volvieron un foco principal de la aviación, como lo fue con el accidente del primer avión comercial supersónico Concorde, en el cual por un FOD de un pedazo de lámina que fue mal instalado de una aeronave DC-10 de la aerolínea Continental, se desprendió en su carrera de despegue quedando en la pista y reventando el neumático del tren izquierdo del Concorde de Air France, generando que un pedazo del neumático golpeará el tanque de combustible, dando como resultado, una tragedia que terminó con 113 víctimas (Bureau d'Enquetes et d'Analyses pour la securite de l'Aviation civile, 2002), dando así un ejemplo de los peligros latentes que se tiene en la aviación. Se busca combatir estos peligros mediante sistemas de control, tal y como lo son los generados por los mismos entes gubernamentales de seguridad aeroportuaria como la FAA y la EASA, basándose en esencia en las indicaciones de la Organización Aeronáutica Civil Internacional (OACI) en el anexo 14 (OACI, 2016) que identifica las medidas necesarias para preservar la seguridad aeroportuaria junto con múltiples manuales de seguridad operacional (OACI, 1983, 1986, 2018).

Viendo la constante necesidad de generar sistemas de prevención de accidentes por FOD, a lo largo de los años se han detectado investigaciones y emprendimientos con respecto a esto como lo son los dispositivos de barredora FOD BOSS (1994, Australia) y FOD BUSTER (Aerosweep Pty Ltd, 2020; The FOD Control Corp., 2020) que tienen como objetivo prevenir los FOD mediante la adaptación de herramientas de barredoras diseñadas para

recoger objetos del pavimento. Funcionando como dispositivos que son arrastrados y remolcados por un vehículo a una velocidad máxima de 40 km/h, pero siendo también un instrumento que no garantiza el 100% de recolección de elementos FOD. Aunque funcionando de manera adecuada para una inspección de FOD.



Figura 1 y 2: Ilustración de FOD BOSS y FOD BUSTER

Recuperado de (Aerosweep Pty Ltd, 2020; The FOD Control Corp., 2020)

Habiendo diferentes alternativas de detección de objetos, se encuentran las inspecciones de objetos extraños por medio de radares como lo es el FODetect® (Waltmath - Massachusetts, 2007) que es un sistema híbrido que usa el sistema de radar y de electróptica que cumple en detectar estos objetos funcionando por radiobalizas puestas en los bordes de pista como se puede ver en la figura 3, siendo aprobado por la FAA cumpliendo con éxito las pruebas de detección en superficies de las pistas de aterrizaje y despegue en condiciones nocturnas, diurnas, lluvia, niebla, neblina y nieve (FAA, E. Herricks, Woodworth, & Patterson, 2012). Junto con este sistema también se encuentra el iferret™ (Singapore, 2008) que es muy similar al anterior, pero con un funcionamiento más enfocado en sistemas electrópticos y un tamaño de sensores más apartado a la pista de despegue y aterrizaje para poder abarcar toda el área como se ve en la Figura 4, (FAA, E. Herricks, Woodworth, Lazar III, et al., 2012).



Figura 3: FODetect®

Recuperado de (FAA, E. Herricks, Woodworth, & Patterson, 2012)



Figura 4: iferret™

Recuperado de (FAA, E. Herricks, Woodworth, Lazar III, et al., 2012)

Junto a estos sistemas se puede hablar de los radares de ondas milimétricas que han sido adaptados a la detección de estos objetos extraños como lo es el Tarsier® (Farnborough – Hampshire, 2004) que funciona como un radar que mediante ondas electromagnéticas que funcionan en frecuencias de 24 y 100 GHz y junto a procesos patentados por ellos mismos, localizan estos objetos desde puntos fijos al igual que los sistemas anteriores como se muestra en la figura 5 (FAA, E. Herricks, et al., 2011), a diferencia del FOD Finder™ (San Diego – California, 2008) que cumple igualmente con un radar de ondas milimétricas, solo que su ubicación es móvil puesto que se encuentra en una camioneta y su funcionamiento se da al momento en que ésta se va desplazando por las zonas de inspección. Se puede observar en la figura 6 (FAA, Herricks, et al., 2011).



Figura 5: Tarsier®

Recuperado de (FAA, E. Herricks, et al., 2011)



Figura 6: FOD Finder™

Recuperado de (FAA, Herricks, et al., 2011)

Adentrándose en el contexto nacional, las investigaciones enfocadas en la detección de FOD no han iniciado. Pero sumando al contexto de detección de objetos, nacionalmente hay diferentes investigaciones de detección de objetos con un enfoque de análisis de imágenes bastante similar al deseado en este proyecto de grado, como lo es la investigación de Gómez Bedoya & Castrillón en 2019 acerca de la construcción de un sistema enfocado en la vigilancia automática de ganado mediante drones en el sector ganadero del oriente

antioqueño por la necesidad de control de robo de ganado, usando alternativas de detección de objetos de aprendizaje profundo como YOLO, con desempeños de precisión de 82% (Gómez Bedoya & Castrillón, 2019), siendo así no el único estudio generado para mantener el control del ganado. Se puede hablar del estudio realizado por (Giraldo Jimenéz et al., 2021) en el cual realizaron una detección de ganado centrado en el uso de los algoritmos de Mask-RCNN y Detectron2 usando alrededor de 1400 imágenes para lograr hacer el seguimiento de ganado de esta zona ganadera. Un ejemplo de su resultado se puede ver en la Figura 7.



Figura 7: Resultados de detección de ganado

Recuperado de (Giraldo Jimenéz et al., 2021; Gómez Bedoya & Castrillón, 2019)

Otro documento que se podría añadir es el de Cuevas Castañeda de 2019, enfocado en la aplicación de visión artificial con un dron para la aplicación específica en el ámbito de la ganadería vacuna en el territorio de Tunja, implementando herramientas algorítmicas como Python, Open CV y Yolo dando como resultado la Figura 8.



Figura 8: Resultado de ganadería vacuna

Recuperado de (Cuevas Castañeda, 2019)

Siendo el análisis de imágenes y la detección de FOD temas que a nivel nacional no están tan inmersos se podría añadir las regulaciones aeronáuticas RAC que se dan para estos temas las cuales se hablarán más a fondo en el marco legal.

Centralizando las investigaciones generadas acerca de la detección de objetos y la prevención de FOD se puede dirigir por la detección de objetos ya que a nivel local no se ha fomentado la investigación para la prevención de FOD, por esto se puede hablar de investigaciones como la de Gómez Marín y Hernández en 2019 enfocada en la detección de aves en cultivos de fresa y a su vez mediante un dron repeler las aves dentro del cultivo. Aunque unificándose más se puede hablar de investigaciones como la de Aviles Correa y Garzón Bernal en 2021 de la FULL que implementa y diseña un prototipo de sistema de detección de obstáculos instalado en un dron de ala fija tipo ala Zagi que mediante la lógica definida en tablas de verdad maneje la señal de control de los motores y la posición de los alerones con respecto al objeto detectado usando un sensor LIDAR que mediante un código referenciado de Rojas Ramos en 2020 que usa neuronas convolucionales para su detección permite el funcionamiento del prototipo de evasión como se puede ver en la Figura 9.



Figura 9: Prototipo de ala Zagi pa evasión de obstaculos
Recuperado de (Aviles Correa & Garzón Bernal, 2021)

1. Planteamiento de problema

La seguridad operacional en la aviación es el enfoque fundamental de toda operación aeroportuaria, ya sea de carga o pasajeros. Mundialmente se han presentado grandes desastres por no tener un buen sistema de gestión de seguridad aeroportuaria (SMS), siendo que el día a día es de vital importancia en cada industria relacionada con la aviación y un buen sistema de seguridad. Determinado por dos niveles, los factores humanos o factores externos al manejo humano, generan diferentes alternativas para mantener una eficiencia exponencial de la seguridad operacional, realizando labores de detección de peligros que pueden llegar a ser un riesgo a la seguridad y efectuando tomas de decisiones basadas en toda la información que se pueda juntar, como datos que ayuden a identificar las preguntas pertinentes y así después buscar una respuesta adecuada para mejorar la seguridad operacional, teniendo en cuenta qué tipo de áreas necesitan de una mayor atención. (OACI, 2018) Como se viene dando en el área de los Objetos Extraños que pueden generar un Daño a una aeronave (FOD) siendo un referente que se presenta en cualquier aeródromo, pista o rampa al momento que una aeronave realice su carrera de taxeo a pista o así mismo en su despegue o aterrizaje.

Siendo los FOD un problema que mundialmente ha estado presente en varios accidentes como lo fue el accidente del concorde mencionado anteriormente que dejó 113 víctimas (Bureau d'Enquetes et d'Analyses pour la securite de l'Aviation civile, 2002), siendo este un gran gran ejemplo del peligro de un FOD que demuestra la importancia que hay en prevenir a toda costa cualquier objeto extraño en la pista que pueda provocar un daño, así sea tan solo una tuerca, al momento de desplazarse una aeronave por la pista uno de sus motores puede succionar esa pieza y en el interior puede generar daños menores en la turbina y que estos generen una secuencia de averías que lleguen a un incidente o accidente que termine en tragedia, sumando a esto que un daño así puede generar gastos imprevistos a aerolíneas

por la reparación de un motor por el simple daño que generó una tuerca. Ejemplos como estos son muchos, así mismo la aeronave puede ser afectada en alguna de sus superficies de control, como alerones, elevadores o timón de dirección generando a su vez posibles desastres.

Como este mismo accidente se han presentado accidentes a nivel nacional, como lo fue en Julio del 2018 en el corregimiento de San Felipe, Guainía donde un DC-3 de Air Colombia al aterrizar en la pista no preparada, terminando su sentada de ruedas ya en suelo por una varilla de hierro de una construcción, reventó el neumático del tren de aterrizaje izquierdo lo cual produjo un pivote en el terreno, que generó un golpe del ala izquierda y la afectación de ambas hélices en los motores (Grupo de Investigación de Accidentes – GRIAA, 2018) Gracias a objetos pequeños en la pista se pueden generar accidentes graves, aunque a su vez también hay incidentes que se producen por equivocaciones premeditadas como el accidente de un Airbus A-320 de Avianca en octubre de 2020 que realizaba un vuelo desde el aeropuerto Antonio Nariño de Pasto, el cual recientemente había tenido una reparación en la pista hacia el aeropuerto El Dorado en Bogotá cuando iniciaba su carrera de despegue tuvo una turbulencia que se generó por el desprendimiento de asfalto de la pista por el empuje de sus motores, por una falla de adherencia entre capas que a su vez levantó el asfalto (FOD) generando un golpe en el estabilizador horizontal de la aeronave (Grupo de Investigación de Accidentes – GRIAA, 2020).

De esta forma y en pro de mejorar la seguridad aeroportuaria en pistas viendo el avance tecnológico que se ha dado en los últimos años, se plantea una nueva propuesta teniendo en cuenta específicamente el problema que generan los FOD en pista durante el despegue y aterrizaje, de esta forma se propone como grupo de estudiantes de ingeniería aeronáutica, adelantar una investigación de tipo cualitativa en la cual se pueda aplicar tecnologías modernas como el análisis de imágenes para identificar los FOD a través de un UAV, aplicando el uso de programas como Python, GoogleColab, RoboFlow, Makesence y YOLO que apoyan a la detección de objetos mediante neuronas convolucionales teniendo en cuenta el tiempo en que se genera actualmente para tareas de inspección de FOD mediante personal humano y a su vez la duración de la realización de estas tareas.

1.1 Objetivo

1.1.1 Objetivo General

Diseñar un sistema a partir de inteligencia artificial para la detección de FOD en aeropuertos por medio de inspecciones con UAS.

1.1.2 Objetivos Específicos

- Caracterizar los equipos UAS que permitan realizar inspecciones de áreas de movimiento de aeródromos.
- Desarrollar el algoritmo para la detección de objetos mediante softwares adaptativos para el objetivo de análisis de imágenes.
- Validar el modelo de detección de posibles FOD mediante pruebas de campo para identificar qué sistema cumple con la tarea de detección.

1.2 Justificación

Ante la gran problemática que día a día se va generando mediante los posibles peligros que generan los FOD y que a su vez a lo largo de la historia aeronáutica se ve que es un problema latente que se presenta y genera accidentes e incidentes al costo de humanas en el proceso para avanzar y mejorar la seguridad operacional, resulta de especial interés ver hoy en día la importancia que se tiene con respecto al tema y a su vez actualmente ver que se siguen generando incidentes que podrían haberse evitado con un mejor sistema de prevención de peligros por FOD, ampliando más el rango de potenciales peligros, aminorando su posible devenir.

El presente proyecto de grado surge por la necesidad de proporcionar alternativas nuevas con todas las ayudas tecnológicas modernas donde se puede integrar una mejor seguridad operacional previniendo accidentes generados por causa de los FOD, con el propósito de identificar de una manera más eficiente los objetos extraños en la pista mediante la modalidad de un dron que cuenta con una cámara con la cual se realizara el análisis de imágenes para identificar correctamente estos objetos y así mismo recogerlos, disminuyendo tiempo en tareas de inspección de pista por objetos de FOD.

El proyecto busca beneficiar la seguridad de las personas y a su vez la eficiencia de las empresas, primero generando una seguridad operacional evitando incidentes y accidentes

los cuales pueden llegar a provocar pérdidas humanas, y a las empresas disminuyendo costos de operación y de personal al realizar tareas de inspección de pista.

Debido a que actualmente no se cuenta con alternativas más modernas para la inspección de este tipo de objetos, se busca profundizar en este tema generando alternativas modernas, por medio de un método de inspección más alternativo al actual, buscando con este proyecto afianzar los conocimientos del tema haciendo que este sea un punto de partida en el cual se empiece a generar estudios futuros profundizando en alternativas más eficientes para la seguridad operacional, siendo así mismo este proyecto contrastado con temas relacionados que ya se han realizado para mantener una buena seguridad aeroportuaria.

2. Marco Referencial

2.1 Marco Teórico

Para la implementación de este sistema de detección de FOD para la mejora de la seguridad operacional para el transporte aeroportuario se precisa definir ciertos conceptos claves, los cuales ayudarán a esclarecer todo el tema a abordar, por esto se enfocará en tres grandes áreas, Seguridad Operacional, Drones y Sistemas de análisis de imágenes.

2.1.1 Seguridad Operacional

La OACI (2018) define la seguridad operacional como el área de la aviación que “procura mitigar en forma proactiva los riesgos de seguridad operacional antes de que resulten en accidentes e incidentes de aviación. Mediante la implementación de la gestión de la seguridad operacional SMS”, siendo así un área de suma importancia en la cual se busca ante todo pronóstico mantener la seguridad, dividiéndose este sistema en dos grandes áreas de la seguridad, como lo es la Seguridad Operacional y la Seguridad de la aviación, al hablar de ellos se podría decir que son lo mismo, pero no, la seguridad de la aviación se enfoca en actos maliciosos y voluntarios que buscan afectar el funcionamiento del sistema de la aviación, en cambio la seguridad operacional se enfoca en los actos involuntarios donde busca mejorar estas actuaciones, aunque los dos buscan la protección de las personas y los bienes aeronáuticos.

2.1.2 SMS

Teniendo claro el concepto de seguridad operacional se puede adentrar en el sistema que se ha integrado, definido por la misma OACI (2018) el Sistema de Gestión de Seguridad operacional es el estado en el que los riesgos asociados a las actividades de aviación relativas a la operación de aeronaves, o que apoyan directamente dicha operación, se reducen y

controlan a un nivel aceptable, considerando a la industria de la aviación en su totalidad como todo un sistema. Todos los proveedores de servicios y sus sistemas para la gestión de la seguridad operacional se consideran como subsistemas, teniendo en cuenta que organizaciones y personas están en el SMS y quien debe responder por la seguridad operacional, teniendo una evaluación adecuada mediante legislaciones y reglamentos que tratan todos los peligros posibles.

2.1.3 FOD

Como el enfoque principal de esta investigación se define gracias a la OPAIN (OPAIN, 2019) que un FOD es el “daño causado a una aeronave por objetos pequeños o aves aspirados por los motores o cualquier objeto o material suelto en el área de movimiento que pueda ser impulsado por el aire o arrojado hacia una aeronave, persona, edificio o vehículo.”, señalando así al FOD como un riesgo que debe ser removido con prontitud.



Figura 10: FOD

Recuperado de (Ortíz, 2020)

Siendo un peligro latente que siempre se mantendrá en latencia siendo una amenaza significativa a toda la Seguridad Operacional, siendo afectada junto con gastos suficientemente grandes por algún error, afectando a las aerolíneas como a los mismos aeródromos o aeropuertos donde opere.

2.1.4 Inspección FOD

Las inspecciones que se realizan a las zonas de funcionamiento de las aeronaves son inspecciones que vienen muy ligadamente con la prevención de FOD por esto se puede

categorizar las inspecciones y auditorías periódicas en las zonas de operación como lo hace en 2010 el Departamento de Operaciones y Seguridad de la Zona de Operaciones de Aeropuertos en tres niveles con el enfoque de garantizar altos niveles de seguridad operacional:

- **Nivel 1:** Se basa en inspecciones rutinarias que deben ser llevadas a cabo por el personal de operaciones que se basa en realizar una inspección que abarque toda el área de movimiento adyacente al aeropuerto.
- **Nivel 2:** Este tipo de inspección es mucho más rigurosa ya que debe ser llevada a cabo por el personal de operaciones integrando los departamentos técnicos, de mantenimiento, eléctrico, infraestructura y todo aquel que este estrictamente relacionado con todas las pistas, zonas de taxeo y plataformas realizando una inspección muy detallada.
- **Nivel 3:** Las inspecciones de este nivel son mucho más rigurosas ya que buscan proporcionar una perspectiva mayor de supervisión de seguridad operacional llevándose a cabo de acuerdo a las normas exigidas incluyendo el correcto cumplimiento de inspecciones nivel 1 y 2.

Como se puede ver las inspecciones se diferencian en diferentes niveles de profundidad, la (FAA & Dermody, 2022) se refiere a estas inspecciones en función a su programa como son divididas en:

- **Programa de inspección diaria**

Basada en las inspecciones diarias que se deben realizar como requisito adicional del FAR 139 (FAA, 2004) que se refiere a que no es solo una inspección diaria, si no que estas inspecciones se deben hacer una a la primera hora de la mañana, antes de cualquier operación refiriéndose a realizar una inspección a la luz del día y otra nocturna teniendo en cuenta la clase de aeropuerto que se esté explotando ya que dependerá de esto las inspecciones que se deben realizar en el día.

- **Inspección de vigilancia continua**

Estas inspecciones se refieren a las que se deben realizar en áreas que el aeropuerto indica en su Manual Aeroportuario Certificado (ACM) y deben tener vigilancia continua cuando el personal ingresa a dichas áreas ya que se pueden producir condiciones peligrosas significativas.

- **Inspección especial**

Se refiere a las inspecciones que se dan cuando se produce una condición inusual o acontecimiento inusitado, incluyendo en ocasiones cuando se tiene lugar un proyecto de

construcción garantizando que no hay alguna condición insegura, realizándolas antes de que el personal de construcción se retire en caso de que se deba tomar medidas correctivas.

- **Inspección periódica del estado**

Las inspecciones de este tipo se presentan cuando se dan condiciones que cambian lentamente y pueden ser desapercibidas y se ven constantemente relacionadas con las inspecciones diarias, un ejemplo sería la inspección de las pistas verificando el repintado de las señalizaciones, o la verificación de desagües de la pista luego de lluvias fuertes, todo este tipo de inspecciones no se deben dejar de lado y cuentan con su periodicidad.



Figura 11: Inspección de pista

Recuperado de (EGA Master S.L, 2020)

Modelo de queso suizo

A medida del estudio de las causas de varios desastres se ha propuesto el modelo del queso suizo el cual el profesor James Reason (2018) explica por qué suceden las fallas, desastres y accidentes, el cual funciona para el análisis y la prevención de riesgos. Este modelo se basa en explicar que cualquier componente de un sistema u organización se considera como un segmento siendo relacionado con una rebanada de queso, definiendo en el caso de la aviación como ejemplo el técnico que realiza la tarea de reparación de la aeronave un segmento, luego el supervisor que revisa la tarea confirma la reparación hecha adecuadamente como otro segmento, el inspector actúa como otro segmento, luego el segmento en el que actúa el piloto al hacer una inspección de la aeronave y por último segmento los mismos sistemas de la aeronave al afirmar que la aeronave funciona correctamente (dependiendo de la reparación que se hubiera hecho), como se puede ver hay varios segmentos que tienen que ser sobrepasados para que suceda un accidente, esto viene relacionado con las rebanadas de queso, cada rebanada representa un posible error si se ponen en hilera las rebanadas y hay una secuencia en los agujeros del queso ocurrirá una

catástrofe, en términos más técnicos es una falla de errores, principalmente por la combinación de fallas activas que expresan mediante el factor humano, esta suceden inmediatamente y las condiciones latentes que se dan mucho antes de que suceda un daño, pueden estar presentes durante un tiempo y así mismo estas pudieron ser creadas con buenas intenciones.

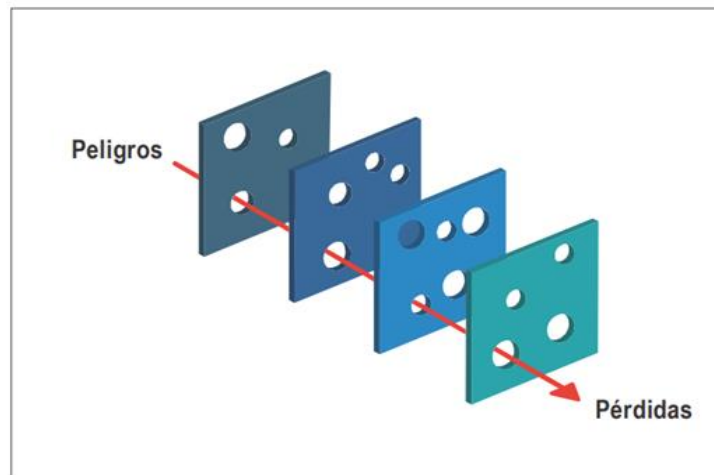


Figura 12: Causalidad de Modelo del queso suizo

Recuperado de (OACI, 2018)

2.1.5 Drones

Dron una palabra que en la cultura general la cual se puede identificar fácilmente su significado, pero más técnicamente se divide en diferentes términos como lo es en un UAV (vehículo aéreo no tripulado en inglés) que es el término técnico más adecuado o así mismo como UAS (sistema aéreo no tripulado en inglés), RPA (aeronave pilotada a distancia en inglés), multirrotor, hexacóptero y octocóptero. (Asadi, 2016), teniendo diferentes términos para su uso.

Los drones son una idea descubierta hace más de un siglo, sus inicios fueron con los globos aerostáticos usados como bombas lanzadas a distancia sin tripulación para la industria militar (Ana Velázquez Olivera, 2017), luego a medida de que llegó la aviación con los hermanos Wright llegó la Primer Guerra Mundial donde se lanzan torpedos de ataque mediante la frecuencia AM (Villalta Ayala & Guerra Jara, 2016) y llegar a la Segunda Guerra Mundial con la producción en serie de Radioplane OQ-2 que funcionaba lanzándolo mediante una catapulta y se recuperaba con un paracaídas, del cual se hicieron más de 9400

unidades para entrenar a los artilleros antiaéreos, luego de este avance empezaron en la década de los 50s a integrar cámaras cinematográficas para uso de reconocimiento (Gibeault, 2016).

Gracias a este impulso en la industria de los VANT (Vehículos Aéreos No Tripulados) al pasar de los años con la tecnología de las aeronaves tripuladas se implementó el vuelo mediante sistemas de propulsión a reacción en la Guerra Fría, para luego las fuerzas aéreas estadounidenses en la guerra de vietnam se empezará a usar más los VANT de reconocimiento terrestre (Villasenor, 2013) Generando este avance y gracias a los conflictos que se presentaban constantemente se fueron integrando armas a los VANT para dar respuesta rápida a un enemigo, en este tiempo se popularizó la palabra Drone que se traducía como Zángano (Abeja macho). Como a lo largo de los años la tecnología militar llegó a pasar a uso civil generando un uso más popular a los drones con aplicaciones de ingeniería, salud, prevención de desastres, cinematografía entre muchas más aplicaciones cotidianas (Reuter & Pedenovi, 2019).

2.1.6 Tipos de drones

Existen distintos tipos de drones, donde se ven con múltiples rotores y formas, también varía el tipo de uso que se le dará al UAS (Hassanalian & Abdelkefi, 2017), donde se reflejan ventajas y desventajas según el campo en el que se emplee, los drones que se comercializan a la fecha se pueden clasificar según su tipo de vuelo.

- **Multirrotores:** Son los drones que cuentan con una configuración de tres, cuatro, seis y ocho rotores de empuje, dependiendo de su configuración tendrá una mayor o menor estabilidad dependiendo de sus características.
- **Ala Fija:** Tienen una configuración similar a la de un avión, comúnmente su inicio de vuelo se da por catapulta sea humana o mecánicamente, teniendo la desventaja de no poder lograr un vuelo estacionario.
- **VTOL:** Hace referencia a una combinación de estos dos tipos siendo UAS que cuentan con muy buena autonomía gracias a su desplazamiento, pero poca maniobrabilidad generando problemas con su estabilidad.

2.1.7 Análisis de imágenes mediante IA

Según González, R.C., Woods, R.E (2002) hace referencia al análisis de imágenes como al conjunto de datos que a partir de imágenes se dan datos más numéricos al identificar distancias entre coordenadas (x,y) identificando el tamaño de las imagen y a su vez la distancia entre un punto y otro mediante píxeles que son los elementos que identifican una imagen mediante su tamaño y calidad, como se ve constantemente en estudios geográficos, de seguridad, astronómicos, químicos y muchas áreas de alta necesidad de análisis de datos e imágenes evitando el uso de sistemas anticuados por el factor humano dejando todos estos análisis en manos de una computadora que mediante su enseñanza podrá realizar el estudio necesario al que se tiene estipulado (López Poveda, 2002).

2.1.8 Inteligencia artificial

Hoy en día en plena segunda década del siglo XXI se ha dado una tecnología tan avanzada que es difícil de saber a diario que se va ideando con la tecnología desde la microscópica a la aeroespacial, una gran rama que sobresale de la tecnología actual es la inteligencia artificial (IA), una inteligencia que en términos simples es la inteligencia que pueden tener las maquinas, computadoras que pueden tener procesos mentales y razonamientos de estos procesos llegando a soluciones o a interactuar con múltiples variables y llegar a una conclusión, esta inteligencia se realiza para que piense como lo haría un ser humano, realizando tareas como el procesamiento de lenguaje natural, representación del conocimiento, razonamiento automático, aprendizaje autónomo de máquina, vista computacional y robótica de análisis de sonido e imágenes, entre muchas más, como se ve cotidianamente en los teléfonos inteligentes al momento en que se usa el asistente de audio, el reconocimiento facial del celular, la lectura de textos, la cámara del celular al reconocer una sonrisa (Russell et al., 2004).

Mucha de esta tecnología se encuentra en nuestras manos, pero todo esto inicio hace muchos años, exactamente en 1950 con Alan Turing y su prueba de diferenciar la inteligencia que podría dar una computadora con la de un ser humano, realizando una prueba de un interrogador que identifique mediante preguntas la inteligencia de la maquina y el ser humano, sin saber cuál es cual, hasta que el interrogador sea incapaz de reconocer alguna diferencia entre los dos.

2.1.9 Machine Learning

Al avanzar de la tecnología y de los modelos tecnológicos que esta misma requiere a inteligencia artificial parte desde un punto inicial el cual es el Aprendizaje Automático (Machine Learning) que proporciona a los ordenadores computacionales la capacidad de aprender sin que se esté presente el programador, es decir sin que se le programe alguna función o tarea, ya que este mismo lo hace automáticamente, funcionando mediante un algoritmo que aprenderá de los datos a aprender y así identificar patrones o tendencias que le ayuden a clasificar elementos (Torres, 2018).

Con el avance que se ha generado a lo largo de los años el machine learning ha tenido gran variedad de algoritmos de aprendizaje dividiéndose en supervisado, no supervisado y profundo (Deep learning):

- Aprendizaje supervisado: Se refiere a los datos que incluyen una etiqueta (label) generada por el programador para que el modelo aprenda.
- Aprendizaje no supervisado: Es lo contrario al supervisado ya que no se basa en etiquetas ya que el algoritmo clasifica la información por sí mismo.

2.1.10 Deep Learning

Al pasar del machine learning es necesario hablar más ampliamente del Aprendizaje Profundo (Deep learning) el cual consiste en facilitar que los modelos aprendan respecto a la experiencia y en especial en términos de “jerarquía” lo cual permite que el programador tenga que especificar los datos o conocimiento que el mismo algoritmo necesita y funcionando con el concepto de jerarquía el mismo modelo aprende conceptos complejos de modelamiento a partir de conceptos sencillos de modelar, de ca su nombre ya que estos conceptos se van construyendo mediante demasiadas capas que van profundizando cada vez más, lo cual permite crear modelos de IA que puedan identificar imágenes, ondas de audio para lenguaje, archivos de lectura (Centeno Franco, 2019).

2.1.11 Redes neuronales

Para que toda esta tecnología se diera principalmente la inteligencia artificial se basa en el funcionamiento principal del ser humano, un cerebro que cuenta con millones de neuronas que interactúan entre sí, las maquinas usan este mismo método con redes neuronales que al interactuar juntas cumplirán con tareas de aprendizaje y así realizar tareas. Funcionando

como una unidad proceso típica, con una entrada de datos, un procesamiento de estos datos dando una salida de datos.

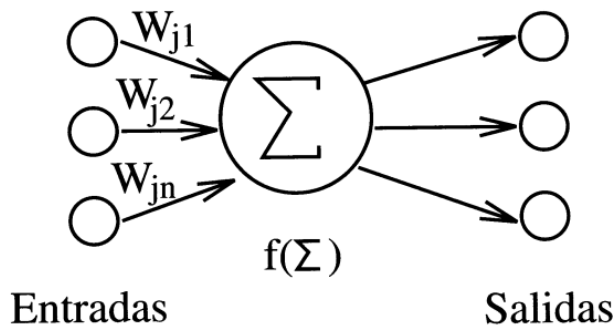


Figura 13: Esquema de una unidad de proceso típica

Recuperado de (Isasi Vinuela & Galván León, 2004)

2.1.12 Red neuronal artificial

Una red neuronal artificial (RNA) como su nombre lo dice es un conjunto de neuronas que se relacionan generando un sistema de aprendizaje automático organizado por niveles o capas, funcionando interconectadamente neurona con neurona (Isasi Vinuela & Galván León, 2004). Con cada neurona realizando cálculos matemáticos aplicando una función de activación, como se verá adelante el funcionamiento y los componentes de la red neuronal artificial.

- Perceptrón

El perceptrón es la unidad más simple de la red neuronal artificial siendo el fundamento básico de la capa considerándose como el bloque de construcción de dicha capa. Su funcionamiento es muy simple, primero consta de valores de entrada, luego estas entradas pasan a unos pesos para luego el resultado llegue a ingresarse en la función de activación que da la solución final, dividiéndose en dos modelos.

o Modelo de una capa

Su funcionamiento de red es simple basándose en un conjunto de células de entrada y así mismo de salida siendo estas células conectadas entre sí determinando la discriminación del sistema como lo describe Isasi Vinuela & Galván León y se observa en la figura 14 “x1” y “x2” son las células de entrada y la salida “y” y la conexión de estas células pasada a pesos como “w1” y “w2”

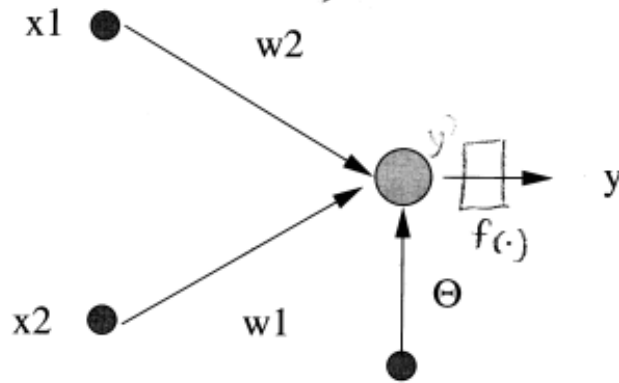


Figura 14: Arquitectura de un perceptrón de una capa
 Recuperado de (Isasi Vinuela & Galván León, 2004)

Pasando del esquema se obtiene la Ecuación 1. de activación de la célula mediante la suma ponderada por los pesos de cada entrada:

$$y' = \sum_{i=0}^n \omega_i x_i \tag{1}$$

○ **Modelo multicapa**

Este modelo es igual al de una sola capa, pero con capas ocultas que interactúan cíclicamente, es decir las de entrada, las ocultas y de salida solucionando la separación y discriminación que genera el de una capa (Isasi Vinuela & Galván León, 2004).

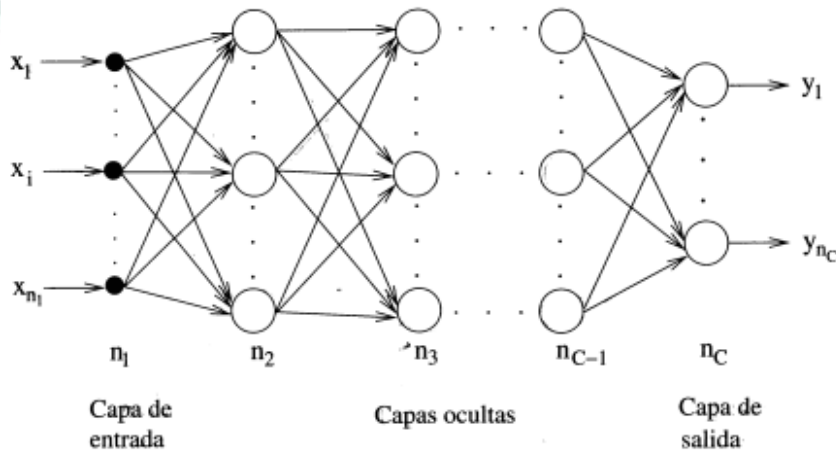


Figura 15: Arquitectura del perceptrón multicapa
 Recuperado de (Isasi Vinuela & Galván León, 2004)

Definiendo la capa de entrada como la capa visible que recibe los señales o patrones que utilizara en el entrenamiento de la red neuronal y luego la capa oculta que es el grupo de neuronas que se encuentran entre las capas de entrada y de salida, que representa la salida de respuesta del valor de los vectores de entrada.

2.1.13 Función de activación

En contexto la función de activación es un valor que retorna de la salida que ha sido generada por las capas anteriores a la salida, esta neurona depende principalmente de la función que se le ha impuesto a hacer, las funciones de activación son normalmente continuas y diferenciales. Un ejemplo son las siguientes funciones (Isasi Vinuela & Galván León, 2004):

- Función sigmoideal expresada por la Ecuación 2:

$$f_1(x) = \frac{1}{1 + e^{-z}} \quad (2)$$

- Función tangente hiperbólica expresada por la Ecuación 3:

$$f_1(x) = \frac{1 - e^{-z}}{1 + e^{-z}} \quad (3)$$

Siendo definidas como funciones crecientes limitadas por dos niveles de saturación de salida 1 y 0 para la sigmoideal, y la de tangente hiperbólica de 1 y -1 como se ve adelante.

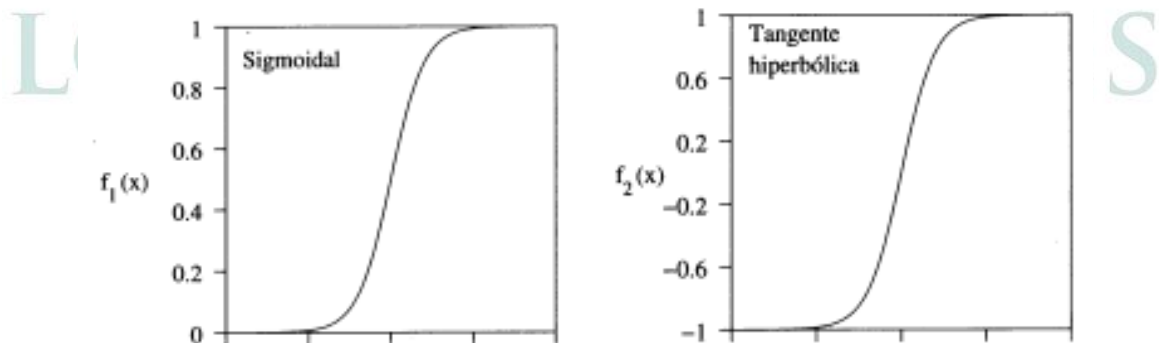


Figura 16: Funciones de activación del perceptrón multicapa

Recuperado de (Isasi Vinuela & Galván León, 2004)

2.1.14 Red neuronal convolucional

Las redes neuronales convolucionales (CNN) el enfoque fuerte de las redes neuronales definen la más poderosa técnica de las redes neuronales artificiales, naciendo su nombre de

la operación matemática convolución usada en matrices, usadas enormemente en la actualidad en aplicaciones de área de visión con cámara o video, de detección de imágenes por computadora, conectando una o más capas convolucionales generando una red neuronal estándar dependiendo del objetivo que se le desee integrar como clasificación de imágenes o detección de objetos.

Este tipo de redes neuronales iniciaron con su funcionamiento en 1998 con el primer modelo de reconocimiento de dígitos escritos, siendo en su momento poco eficiente por el poco volumen de base de datos que se tenía por la cantidad de imágenes usadas sumando el factor computacional que se tenía tecnológicamente en su época, lo cual llevo a la actualidad a iniciar con su funcionamiento en mayor volumen (Lecun et al., 1998).

a. Arquitectura de CNN

Una comparación que se tiene presente con las CNN y las RNA es que con las redes de neuronas comunes se pueden hacer modelos complejos como el análisis de imágenes, pero hay un gran inconveniente al realizar el algoritmo de estas es que al realizar el análisis el vector de entrada tendría que usar una matriz con los pixeles de una imagen, llegando a necesitar miles de neuronas por cada pixel que tendría la imagen si solo fuera en escala de grises como se puede ver la figura 17 un ejemplo menor con solo 16 pixeles, todo esto simplemente para integrar una imagen en escala de grises lo cual con una imagen a color por contar con colores se generarían tres matrices de RGB, identificando los colores rojo, verde y azul, por lo tanto se generaría una cantidad de neuronas tres veces mayor.

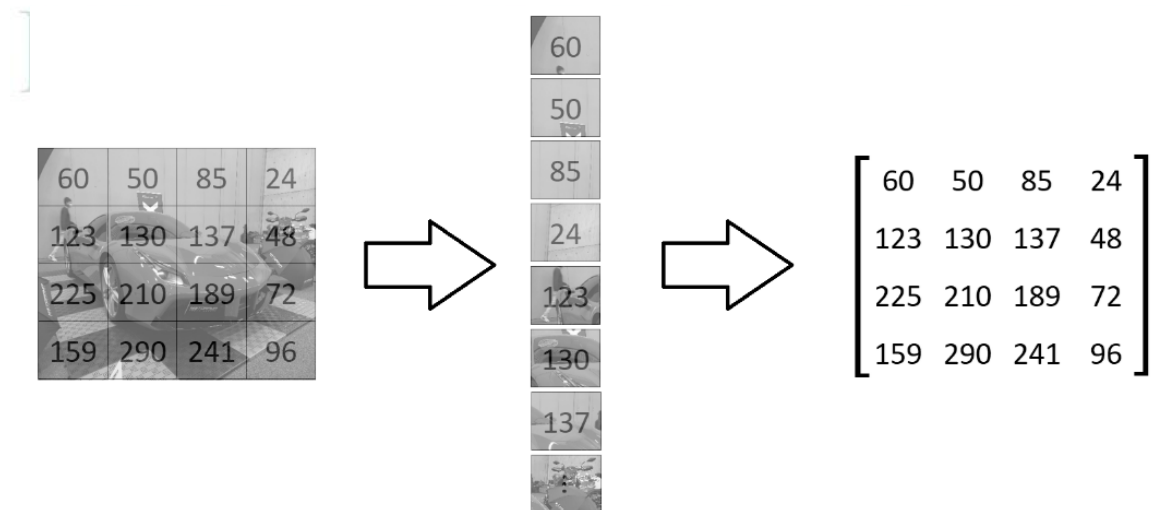


Figura 17: Ejemplo de red neuronal común con una imagen en escala de grises.

Fuente: Elaboración propia.

Al realizar este análisis con este tipo de redes neuronales se puede ver la necesidad de usar una capacidad computacional gigante con imágenes que cuenten con millones de píxeles, por esto es que las CNN son la solución para este inconveniente ya que en la capa de entrada se iniciara con una imagen en sí en vez de una matriz, de igual modo se evitara la dependencia espacial y temporal que permite realizar análisis de imágenes más eficientes independiente de la posición espacial que tenga un objeto en una imagen.

El funcionamiento de una red de neuronas convolucionales se puede resumir en que su función es disminuir las imágenes que se quieren analizar a una forma más fácil de procesar, claramente evitando perder las características principales de reconocimiento de la imagen, para que así no se baje la predicción, gracias a esto el modelo podrá aprender funciones complejas y así mismo será capaz de manejar sets de datos masivos (Saha, 2018).

b. Capa de convolución

Para el funcionamiento de esta CNN se debe realizar un ajuste del conjunto de datos que analizara ya que debido a la reducción en los parámetros involucrados usara diferentes capas para su funcionamiento, como lo es con la capa de convolución que es el pilar principal de la CNN, el cual se basa en la aplicación de filtros que aprenden patrones pequeños, los cuales se llaman núcleos o “Kernels” que tienen la función de cubrir todo el campo de la imagen siendo recorrido en todo el ancho y alto de la imagen, por lo cual calcula el producto escalar entre las entradas del filtro y la capa de entrada, generando un mapa bidimensional de la imagen. Es decir, el filtro Kernel es una matriz que asigna valores a los píxeles que desea analizar de la imagen bidimensionales teniendo como enfoque principal y secundarios los píxeles contiguos. Generalmente para esta operación los Kernels son elementos de 3×3 , operando mediante el movimiento hacia la derecha de cada píxel hasta lograr el ancho completo de dicha imagen saltando hacia abajo al principio de la imagen de igual manera hasta lograr recorrer toda la imagen completa, un ejemplo de este se puede observar en la figura 18.

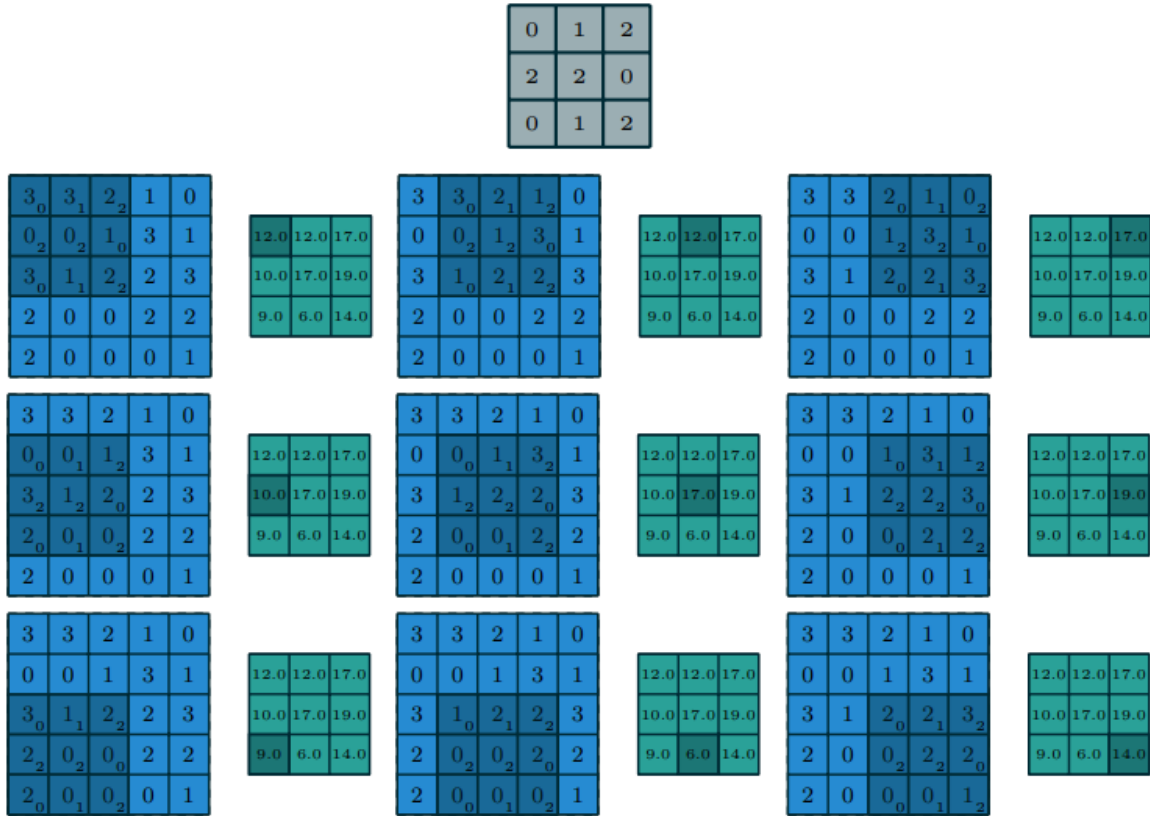


Figura 18: Filtro Kernel realizando su función de convolución

Recuperado de (Dumoulin & Visin, 2016)

c. Capa de reducción pooling

Para el correcto funcionamiento de las capas de una red neuronal convolucional las capas convolucionales trabajan conjuntamente con las capas de reducción pooling, las cuales cumplen con la función de reducir el tamaño de las imágenes cuando son grandes disminuyendo su mapa de dimensionalidad sin dejar de lado la información importante de la imagen como se puede ver en la imagen 19.

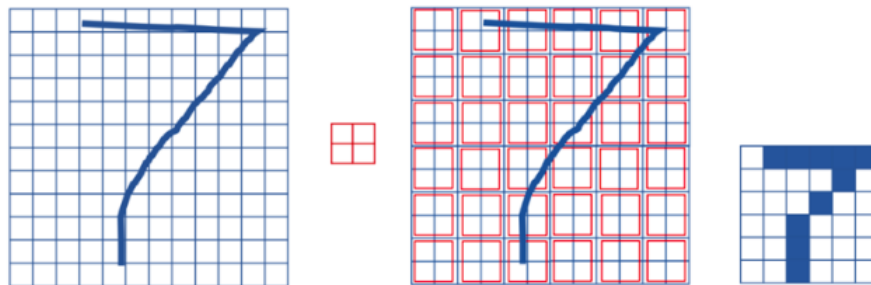


Figura 19: Reducción pooling de una imagen.

Recuperado de (Torres, 2018)

Este proceso consigue evitar los altos costes computacionales para procesar los sets de datos gigantes, siendo de utilidad para extraer las características dominantes como rotaciones y posiciones manteniendo la efectividad del modelo (Saha, 2018). Esta capa se divide en dos tipos:

- Max Pooling: Funciona devolviendo el valor máximo de la posición de la imagen que fue envuelta por el filtro Kernel.
- Average Pooling: Al igual que el Max pooling resuelve el valor promedio de la zona que fue cubierta por el filtro Kernel, como se muestra en la figura 20.

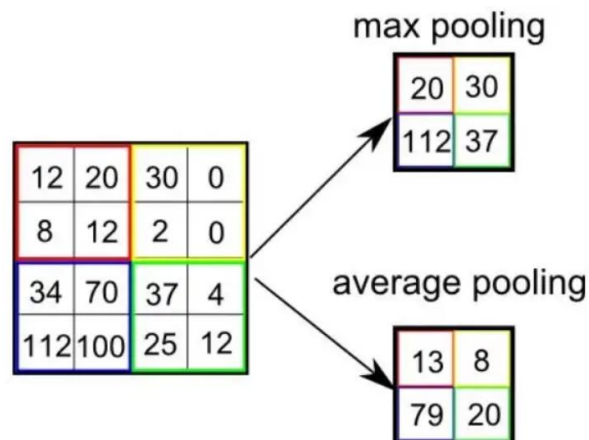


Figura 20: Tipos de reducción Max Pooling y Average Pooling.

Recuperado de (Saha, 2018).

De manera que la capa de pooling se integra con el funcionamiento de la capa convolucional se puede definir el propósito principal de esta que es poder detectar características o rasgos visuales de las imágenes, como lo son aristas, líneas verticales o horizontales, gotas de color, curvas, ángulos, entre muchos más, lo cual mediante varias capas convolucionales el modelo aprende jerarquías espaciales y temporales, como puede ser una primer capa que identifica líneas verticales, horizontales o diagonales de una imagen, la segunda capa identifica rasgos más claros como ojos, nariz, orejas y boca, pasando a la tercera capa que mediante distintos apartados espaciales puede encontrar los patrones de una cara (Torres, 2018), como se explica en la figura 21.

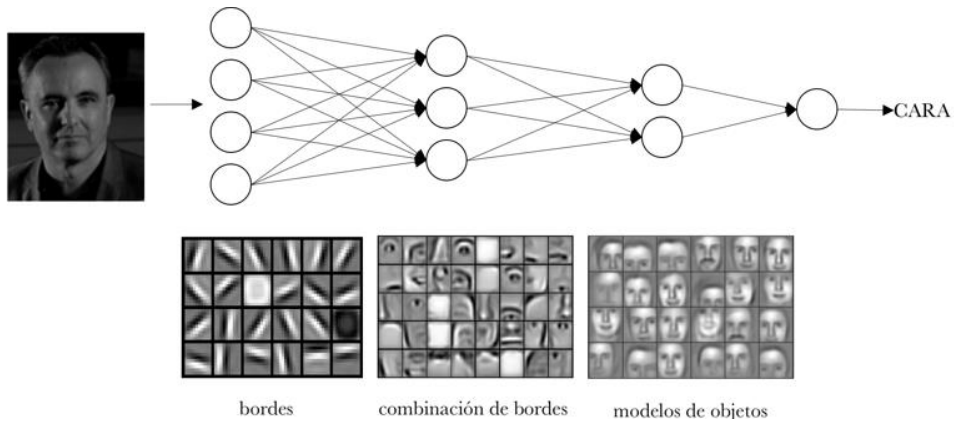


Figura 21: Funcionamiento de una red convolucional.

Recuperado de (Torres, 2020)

d. Capa completamente conectada

La capa totalmente conectada es la última capa del esquema de las CNN y su función es clasificar y determinar según su entrenamiento a que clase pertenece la imagen de entrada, esta capa se encuentra al final de la última capa de pooling, refiriéndose así esta capa a que cada nodo de la primera capa está conectada directamente a cada nodo de la segunda formando así esta capa lo más parecido a una red neuronal tradicional como un perceptrón multicapa que funciona con la activación Softmax que sirve para normalizar exponencialmente los valores que se obtuvieron de la red ante una nueva entrada y así saber la probabilidad de que pertenezca a cada categoría (Chen et al., 2018).

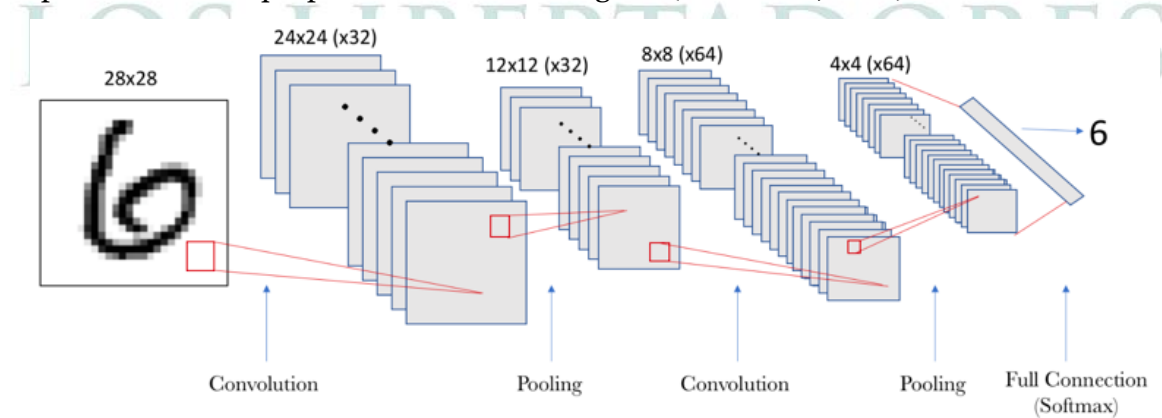


Figura 22: Red neuronal convolucional con capa completamente conectada

Recuperado de (Torres, 2018)

2.1.15 Detección de objetos

El machine learning ha traído la gran tecnología de detección de objetos con el aprendizaje profundo en el área de visión por computadora, siendo aplicado en distintas áreas actuales como lo es en la conducción autónoma, los sistemas de correo, sistemas de seguridad, entre muchos más. Siendo un sistema tan complejo tiene el problema en determinar donde se encuentran los objetos localizados en una imagen o video, sin poder determinar a la clase que pertenecen, por lo cual se genera la Localización y Clasificación de objetos que se divide en tres fases:

- Categorización de regiones: Teniendo las diferentes relaciones de aspecto y tamaño los objetos en una imagen pueden aparecer en cualquier posición, por esto una opción factible es escanear toda la imagen con una ventana multiescalar y así detectar todas las posibles posiciones de los objetos, pero se tiene el inconveniente de usar muchas ventanas por lo cual se necesita un nivel de computación muy grande.
- Extracción de características: Para identificar los objetos se deben caracterizar estos mediante una representación semántica y robusta, en estos casos se puede usar los métodos de Scale-Invariant feature transform (SIFT), Histograms of Oriented Gradients (HOG) y Haar-like, no obstante, por las diversas apariencias y condiciones de iluminación y fondos no se puede generar un sistema adecuado de detección.
- Categorización de objetos: Se basa en la detección de objetos mediante un clasificador que ayuda a distinguir una clase de objetos con respecto a otra dependiendo del set de datos que se le proporcione, generando representaciones jerárquicas, informativas y semánticas distinguiendo cada objeto, métodos como este son Supported Vector Machines (SVM), AdaBoost o Deformable Part-based Model (DPM).

Para la detección de objetos como se expuso anterior mente pueden haber diferentes métodos de análisis, pero no lo suficientemente eficientes ya que no logran tener un grado aceptable sin la supervisión humana, viendo este inconveniente se generan dos tipos de modelos, de mono etapa y de dos etapas (Perales Izquierdo, 2022).

2.1.16 Modelos de mono-etapa

En este caso, los modelos se diferencian por utilizar una única red profunda para la detección de objetos; esto quiere decir que se encarga de predecir las diferentes Bounding Box (BBs). En estos modelos su coste computacional otorga una mayor ventaja en cuanto a

lo simple y rápido. Se destacan los modelos SSD, algunas versiones de You Only Look Once (Liu et al., 2016).

2.1.17 Modelos de dos etapas

Estos modelos se caracterizan porque comparten los pesos de las capas iniciales, en primera instancia se enfoca en detectar las regiones de la imagen en donde es más probable que existan objetos; en la segunda instancia se enfoca en calcular la probabilidad que exista un objeto, como lo es la arquitectura Faster R-CNN donde se evidencian las distintas variables.

Faster R-CNN

Se caracteriza por utilizar un sistema de propuesta de regiones (RPN) para lograr que la red de detección de objetos sea entrenable desde el inicio hasta el final. Además usa los últimos mapas de características convolucionales para formular propuestas de regiones que luego se proporcionan a las capas conectadas para la detección final (Ren et al., 2015).

Mask R-CNN

En este modelo se caracteriza que a través de sus procesos añade una nueva rama a su predecesor, excluyendo el alineamiento de pixel-pixel. En su primera etapa una RPN propone múltiples localizaciones de posible sujetos dentro de la imagen; en la segunda etapa se basa en los resultados obtenidos en el primer proceso, prediciendo las clases individuales de los objetos, los bounding boxes reales y de esta forma calcula la máscara binaria para rengio de interés (ROI)(He et al., 2017).

2.1.18 Modelos Deep CNN

Los detectores de objetos anteriores sirven como columna vertebral (Backbones) para que los diferentes modelos Deep CNN puedan ser utilizados (Yan et al., 2020). La siguiente tabla demuestra los más representativos:

Modelo DCNN	Parm (mill)	Cap.DCNN (CONV+FC)	Error	Descripción
AlexNet [Krizhevsky et al., 2012]	57	5 + 2	15.3 %	Fue la primera Red Profunda DCNN presentada den ILSVRC2012.
VGGNet16 [Simonyan and Zisserman, 2014]	134	13 + 2	6.8 %	Incrementa la profundidad de la red. Usa filtros 3×3. Tiene un mejor rendimiento.
GoogLeNet [Szegedy et al., 2015]	6	22	6.7 %	Se basa en la red Inception, este concatena el mapa de características que son el resultado de diferentes tamaños de filtros.
InceptionV2 [Ioffe and Szegedy, 2015, Szegedy et al., 2016]	5	31	4.8 %	Es muy veloz al momento del entrenamiento. Introduce la normalización por Lotes.
YOLONet [Redmon et al., 2016]	64	24 + 1	-	Se basa en la red GoogleNet y el uso del detector YOLO
RestNet50 [He et al., 2016a]	23.4	49	3.6 %	Uso de conexiones residuales con pocos parámetros previos DCNNs.
DarkNet [Redmon and Farhadi, 2017]	20	19	-	Similar a VGGNet pero con menos parámetros para usar en cada capa.
InceptionV4 [Szegedy et al., 2017]	41	75	3.1 %	No tiene conexiones residuales, tiene un rendimiento similar que InceptionResNetV2.
MobileNet [Howard et al., 2017]	3.2	27 + 1	-	Red muy liviana. Usa convoluciones separables para aplicaciones móviles.

Tabla 1: Características de modelos Deep CNN.

Recuperado de (Barba Guamán, 2021)

2.1.19 Métricas de modelos de detección de objetos

De las técnicas más usadas para evaluar la precisión de un conjunto de datos de imágenes es Intersección sobre la Unificación. Este es usado por varios detectores de las CNNs. Esta métrica IoU se define como se ve en la figura 23, teniendo en cuenta que en diferentes estudios se considera un buen resultado si $IoU > 70\%$ como se observa en la figura 24:


$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


Figura 23: Ecuación de métrica de Intersection Over Unión (IoU)

Recuperado de (Rosebrock, 2017)



Figura 24: Predicción IoU

Recuperado de (Rosebrock, 2017)

Otra forma de métrica utilizada en la evaluación de algoritmos de detección de objeto se denomina mean average precision- (mAP). /en esta mAP es el producto de la precisión y recall de las cajas delimitadoras que han sido detectadas, están en un rango de 0 a 100.

Rosebrock menciona que para encontrar el valor mAP se debe tener presente los siguientes pasos: calcular la precisión basado en los verdaderos positivos. Calcular el recall que es una proporción de los verdaderos positivos de todos los positivos. Finalmente, obtener el valor promedio máximo entre la precisión y el recall (Rosebrock, 2017). La expresión matemática sería la Ecuación 4.:

$$mAP = \frac{1}{N} \sum_{n=1}^N AP_i \quad (4)$$

2.1.20 Herramientas de procesamiento

- Python

Python “es un lenguaje de programación que posee eficientes estructuras de datos de alto nivel y un enfoque sencillo pero eficaz sistema de programación orientada a objetos convirtiéndolo en un lenguaje ideal para la creación de scripts y el desarrollo rápido de aplicaciones” según su misma definición dada, es muy usado actualmente en la programación de aplicaciones web, desarrollo de softwares, ciencia de datos y el machine learning que define el constante aprendizaje de un sistema como la enseñanza que un programador le puede dar a un software (Python Software Fndn., 2022).

- Yolo

YOLO definido como Solo Se Vive Una Vez (You Only Look Once en inglés) no da una definición muy clara, pero en si el sistema YOLO es un software que funciona para la detección de objetos en tiempo real, el cual usa una red neuronal convulsionar, la cual ayuda a hacer el análisis de una imagen dividiéndola en regiones y predice en la imagen la aparición de un objeto el cual previamente se le enseñó mediante un entrenamiento, usando un análisis para diferenciar los objetos y eliminar los que no se relacionan con lo que se enseñó, luego de esto establece los objetos que detectó y el tiempo en que se ejecutó, como se puede ver en la imagen (Redmon et al., 2015).

- MakeSense.AI

La aplicación de MakeSense es una herramienta de etiquetado de fotos en línea de código abierto y de uso gratuito bajo la licencia de GPLv3 la cual no requiere de un sistema operativo complejo enfocado en proyectos de aprendizaje profundo, enfocándose en la reducción de tiempo de etiquetado de imágenes mediante IA generando archivos de salida como YOLO, VOC XML, VGG JSON, CSV (Makesense.ai, 2022).

- Roboflow

Roboflow es una plataforma online de libre acceso con diferentes funciones, como su enfoque principal para apoyar en el etiquetado de imágenes fundamentadas para usarse en el entrenamiento de un sistema de detección de objetos permitiendo al usuario usar su software y a su vez apoyarle en su proyecto mediante una librería que tiene múltiples data

set de imágenes previamente etiquetadas y usadas en otros proyectos, sirviendo de ayuda para realizar el entrenamiento de un modelo de análisis de imágenes (Roboflow, 2022).

- **Google Colaboratory**

Es un software online de código abierto proporcionado por Google más llamado GoogleColab el cual es una extensión de su amplia red de usos de Gmail el cual está basado en un entorno de notebook jupyter que se basa en la creación e intercambio de documento de programas literarios para usos de simulación numérica, modelado estadístico y de máquinas el cual permite el uso de una memoria RAM de 12 GB ampliable a 25 GB y un disco duro de 358.27 GB usando su nube de archivos Drive para acceso de archivos (Panchal, 2019).

- **Download All Images**

Es un software online que funciona mediante la opción de extensiones del navegador web Chrome que cumple con la función de recopilar copias de imágenes de una página web descargándolas como archivo .zip protegiéndolas contra pérdida de datos, usada en generar conjunto de datos basados en imágenes para el machine learning (Download All Images, 2022).

- **DroneDeploy**

DroneDeploy es un software enfocado en automatizar el vuelo de un avión teledirigido UAV facilitando la captura de imágenes o videos, generando sistemas de vuelo ya estandarizados integrando rutas de funcionamiento productivas teniendo en cuenta el tiempo de vuelo que normalmente se tiene con los UAV, usándose en industrias de la agricultura, construcción, minerías, petróleo y gas, energías renovables, cubiertas, inspección, entre muchas más (DroneDeploy, 2023).

2.2 Marco Legal

Para abordar cualquier tema aeronáutico civil es necesario tener en cuenta que leyes, reglamentos hay que saber para interiorizarse en el problema a abarcar, por esto en la aviación civil en la cual se tienen múltiples sistemas de prevención de riesgos de seguridad operacional se de profundizar, no solo nacional si no internacionalmente, por esto se abarca que reglamentos hay actualmente y rigen las áreas a integrar en este proyecto de grado, por esto este apartado en las normas que rigen el área de FOD y la de los UAV.

2.2.1 Normativas FOD

- OACI

Entrando en contexto se partirá de lo más amplio a lo más restringido, por eso se parte con la Organización de Aviación Civil Internacional, la cual se basa en el comportamiento principal de todas las normas nacionales, para el funcionamiento de un sistema adecuado de prevención de FOD se integra el Anexo 14 (OACI, 2016) que se encarga del funcionamiento de aeródromos, parte fundamental del control de pistas, aeródromos, rampas y demás áreas de rodaje en los aeródromos o aeropuertos, siempre teniendo en enfoque la seguridad operacional.

Para enfocarse en las inspecciones de prevención de accidentes por objetos extraños se dirige al Manual de Servicios de Aeropuertos (Doc. 9137) en su parte 8_(OACI, 1983)enfocándose en el funcionamiento de los servicios operacionales del aeropuerto de los cuales indica a las normas adecuadas que se deben mantener para mantener una seguridad operacional adecuada, integrando las inspecciones rutinarias que se deben realizar a las zonas del aeropuerto y que direcciones deben ser encargadas para estas tareas por el estado en el cual se desee operar un aeródromo. Junto con este documento viene enlazado el Manual de Sistemas de Guía y Control del Movimiento en la Superficie (SMGCS, Doc. 9476) (OACI, 1986)en el cual se especifica más integralmente el procedimiento de realización de las inspecciones abarcando las tareas de funciones y responsabilidades para la operación del aeródromo, junto con los procedimientos adecuados para realizar las operaciones que se dan en ocasiones de alta dificultad.

- FAA

Basándose en las normativas internacionales y viendo la constante conexión que tienen los aeródromos y aeropuertos con países de gran flujo como lo es Estados Unidos se integra la normativa norteamericana de aviación civil bajo el mando de la Administración Federal de Aviación, basándose en el objetivo de proporcionar las normativas adecuadas para la prevención de FOD, habiendo bastantes documentos relacionados con estos objetivos, pero en esencia están las “Advisory Circular” 150/5210-24 , 150/5200-18D y 150/5220-24_(FAA & Airport Engineering Division, 2010; FAA & Dermody, 2022a; FAA & O’donnell, 2009) que hablan de temas de orientación para poder gestionar y crear un programa de prevención FOD, especificaciones de equipos para operaciones de retirada de estos objetos, identificando a su vez las inspecciones necesarias para identificar los elementos y así mantener una seguridad operacional adecuada.

- **EASA**

La Agencia de la Unión Europea para la Seguridad Aérea siendo el ente administrativo de Europa también tiene su regulación de manejo de FOD que aplica a todos los aeródromos y aeropuertos que se encuentren ligados a su reglamentación, basándose en la OACI genera su propio reglamento (EU) No 139_(EASA, 2022a)_el cual indica todo el funcionamiento correcto que deben llevar los explotadores de aeródromos exigiendo la operación correcta y segura mediante su programa de control garantizando la concientización y participación de personal aeroportuario estableciendo medidas de prevención y procedimientos para la detección de los FOD junto con su retirada y eliminación.

- **SRVSOP**

Integrándose más en el funcionamiento de seguridad operacional se puede hablar del Sistema Regional de Cooperación para la Vigilancia de la Seguridad Operacional, siendo como su nombre lo dice el ente encargado de vigilar y mantener la seguridad operacional de los aeropuertos en Latinoamérica en el cual se puede integrar la circular de asesoramiento AGA-153-015 (SRVSOP, 2020) que “contiene material explicativo e informativo (MEI) y métodos aceptables de cumplimiento (MAC) relativas al cumplimiento de la obligación del operador/explotador de aeródromo de eliminar objetos extraños en el área de movimiento que puedan ser absorbidos por los motores o causar daño a las aeronaves” basándose en los reglamentos internacionales mencionados anteriormente y así mismo en los reglamentos aeronáuticos latinoamericanos (LAR) 153 y 139 (SRVSOP, 2017, 2018) enfocándose en las concesiones que tienen la tarea mantener la seguridad aeroportuaria y así mismo generar un programa de prevención de FOD.

- **OPAIN y AIRPLAN**

Con diferentes aeródromos y aeropuertos a nivel nacional en el país de Colombia se puede hablar de las concesiones de OPAIN y AIRPLAN que son las empresas operadoras de aeropuertos importantes como lo son El Dorado de Bogotá, Olaya Herrera de Medellín, Jose María Córdova de Rionegro, Los Garzones de Montería, Antonio Roldán Betancur de Carepa, El Caraño de Quibdó y Las Brujas de Corozal. Siendo los operadores de todos estos aeropuertos necesitan registrarse por las normativas que aplican nacional e internacionalmente generando sus propios planes operativos de prevención de FOD, como lo son los documentos GSO-PG-0002 y GOP-PN-003 (OPAIN, 2017, 2020) de la OPAIN y PL -03-006 de AIRPLAN (Airplan, 2018).

2.2.2 Normativas UAV

- OACI

La aviación mundial ha entrado en tecnologías constantemente avanzadas, como lo es hoy en día con la integración de los vehículos aéreos no tripulados o drones que se integran más a la aviación civil, viendo este inconveniente la OACI se ha visto forzada a empezar a implementar metodologías para mantener la seguridad aeroportuaria y operacional, como lo inicio con la Circular 328 AN/190 (OACI, 2011) en la cual hace la introducción de los drones a sus estados miembros, junto con el Manual de los Sistemas de Aeronaves Pilotadas a Distancia Doc. 10019 AN/507 (OACI, 2015) en el cual la OACI entra en contexto con las practicas estándar que se deben dar para el manejo de los drones en el espacio aéreo y así integrar la regulación para drones hasta 2025, en adición el plan que tiene dirigido para integrar por completo a los drones en las operaciones cotidianas con reglamentaciones rígidas y solventes en las cuales se explicara su correcto funcionamiento.

- FAA

Teniendo la adaptación de los drones a la sociedad civil la FAA tuvo que integrarse al funcionamiento de estos dispositivos generando la regulación de estas generando la parte 107 de las Regulaciones Federales de Aviación (FAA, 2016), en la cual se integra toda norma para el uso de estas aeronaves como lo es las licencias que se necesitan para su uso, la definición de cuales usos se les puede dar a los drones y así mismo las limitaciones que estos tendrán, así mismo la FAA adiciona información mayor en el apartado “dronezone” (FAA, 2022a) de su página web en el cual se podrá informar sobre las regulaciones que se dan en los distintos estados de Estados Unidos y así mismo dando recomendaciones de uso como el de aplicaciones como B4UFLY que indican las restricciones de espacio aéreo que pueden haber y muchos más datos de uso de drones (FAA, 2022b).

- EASA

Europa entrando en el mundo tecnológico de los drones tuvo que integrarse forzosamente por medio de la Agencia de la Unión Europea para la Seguridad Aérea a regular el funcionamiento de las aeronaves no tripuladas indicando de diferentes formas las regulaciones que regían el comportamiento de estos dispositivos en el aire, como lo son con los “acceptable means of compliance” (AMC) y “guidance material” (GM) que son documentos regulatorios e informativos que suministran la información de vuelo, creación, pilotaje, entre otros requisitos para el uso de drones, por esto la misma EASA genero un documento donde indicaba la ilustración de todas estas normas el cual se llama “Easy Access Rules for Unmanned Aircraft Systems” en el cual se indica cada regulación

establecida para el funcionamiento y vuelo de drones en el territorio Europeo (EASA, 2022b).

- **UAEAC**

A nivel nacional se fue integrando de a poco el uso de drones para uso civil, pero la Unidad Administrativa Especial Aeronáutica Civil tuvo al igual que muchos países integrarse a esta nueva tecnología, en 2015 se generó la Circular Reglamentaria 002, 2015- Aerocivil. La cual dio inicio a los requisitos generales de aeronavegabilidad y operaciones de drones, siendo reemplazada posteriormente en el 2018 por el apéndice 13 del Reglamento Aeronáutico Colombiano (RAC) 91 (UAEAC & Salazar Gómez, 2018) en el cual se convirtió en regulación el funcionamiento de los drones independientemente del propósito que se le tuviera, regulando su funcionamiento en vuelo, aptitud técnica y tipo de operaciones que podría tener, preservando constantemente la seguridad operacional al hacer su uso.



LOS LIBERTADORES
FUNDACIÓN UNIVERSITARIA

3. Diseño metodológico

3.1 Tipo de Investigación

El proyecto a realizar tiene un enfoque cualitativo experimental el cual se basa en un flujo de pasos para un análisis investigativo de un sistema y la aplicación de este donde las etapas de todo el proyecto interactúan hasta llegar a un objetivo como lo menciona Roberto Hernández Sampieri en el libro “Metodología de la investigación” (Hernández Sampieri et al., 2014), en el proyecto se usarán las técnicas y herramientas para ahondar sobre el estudio de sistemas de análisis de imágenes basándose a su vez en el modelo de seis fases: comprensión del sistema y datos, preparación de los datos, modelado, evaluación y despliegue, del modelo de “CRISP-DM” (Shearer, 2000) adaptando estas dos metodologías en el uso de identificación de FOD que se desea abarcar en este proyecto, investigando sobre que modelos se pueden usar para la prevención de estos objetos y a su vez encontrar la mejor manera de integrar el uso de drones para mejorar los procesos de seguridad operacional.

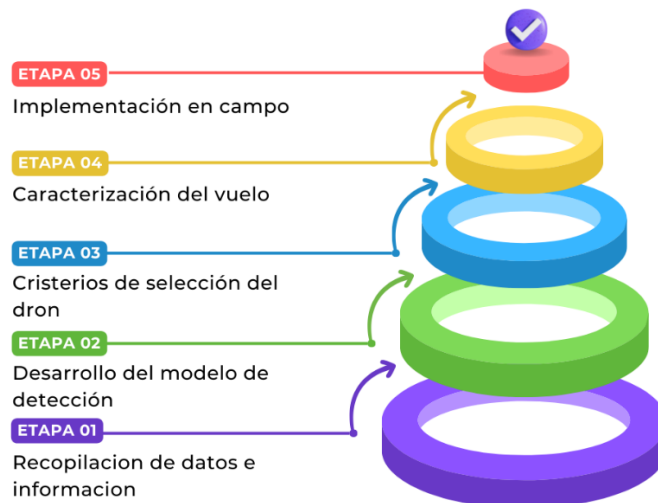


Figura 25: Diagrama de flujo del proyecto.

Autoría propia basado en el libro de Hernández Sampieri y Shearer

3.1.1 Recopilación de datos e información

Para dar inicio a este proyecto de grado se parte desde la información necesaria para entender el diseño que se desea realizar, adquiriendo los conocimientos adecuados para iniciar en el mundo de la inteligencia artificial y entender que se está entrenando, luego de esto se inicia con una base de datos para crear el modelo de detección de objetos, para esto se recopilará esta información autónomamente e investigativamente, usando alternativas en línea que permitan ayudar a generar una base de datos robusta usando imágenes de bases de repositorios como lo son en Google y Roboflow. Al basarse en fotos recopiladas de la red se tomaron fotos propias de alta resolución a una distancia adecuada a las que necesitaría el modelo para poder ser usado en la cámara de un dron y así realizar la detección de objetos en las zonas a analizar.

3.1.2 Desarrollo del modelo

Para este paso se realizara el etiquetado del data set de información que se recopiló en imágenes que estarían listas para el entrenamiento donde se utilizara el software de la plataforma de MakeSense diseñado para el desarrollo de visión por computadora para realizar el etiquetado de las imágenes manualmente para que el sistema pueda aprender mediante estas etiquetas a identificar estos objetos al haber aprendido después de su entrenamiento, pero antes de pasar a este entrenamiento se debe validar todas estas etiquetas usando la plataforma de Roboflow que al igual que MakeSense tiene la herramienta de etiquetado, pero a su vez cuenta con la herramienta de validación de imágenes y etiquetas depurando todo fallo que pudiera haber en las etiquetas mostrando el error que pudiera haber al etiquetar este volumen de imágenes para posteriormente generar un código para realizar el entrenamiento en formato Yolo.

Ya teniendo estos pasos realizados se procede a iniciar con el entrenamiento del modelo de detección de objetos usando la neurona convolucional de código abierto YOLO V5 el cual mediante herramientas como Google Colab se generará el proceso de codificación del sistema de análisis de imágenes realizando la conexión de las imágenes ya recopiladas e identificándolas con sus respectivas etiquetas culminando su entrenamiento.

Teniendo el entrenamiento del modelo listo para funcionar se pasa a hacer la validación de datos confirmando que los datos generados sean lo más acertados a la realidad mediante los datos de validación generando el entrenamiento por completo y así generar un porcentaje de funcionamiento aceptable para iniciar con el proceso de detección de los objetos FOD.

3.1.3 Criterios de dron a usar

Aca gracias a la información adquirida no solo del modelo a usar para la detección de imágenes se investiga de la actualidad, donde se usan los drones actualmente, en que industrias se aplican y cómo se aplican investigando que inferencia tendría con este proyecto identificando que características debería tener el dron para cumplir con su correcto análisis de imágenes detectando los FOD en una pista o zonas de movimiento de aeronaves.

3.1.4 Caracterización del vuelo

Al comprender ya que dron se llegaría a utilizar para la detección de objetos se procede a preparar dicho funcionamiento, teniendo en cuenta aeropuertos como lo es El Dorado de la ciudad de Bogotá, realizando el plan de vuelo que tendría el dron y así mismo ver la viabilidad de su uso solo o el uso conjunto de diferentes drones para poder barrer zonas de inspección y lograr identificar los FOD.

3.1.5 Implementación

Culminado todos estos procesos se emplea el modelo de detección con pruebas de campo entrando en funcionamiento este sistema en el dron generando pruebas en un ambiente natural similar al que tendría en un aeropuerto empezando a detectar los objetos en el suelo para así generar su ubicación y así mismo dar inicio con el proceso de recolección por parte del personal de FOD.

3.2 Delimitación

3.2.1 Espacial

El proyecto a realizar se propone ser una alternativa de seguridad operacional a nivel mundial refiriéndose a una problemática común en todas las pistas aeroportuarias como se ha visto en distintas partes del mundo accidentes producidos por este fenómeno de los FOD, aunque inicialmente se propone adaptar este sistema a nivel nacional enfocándose en el aeropuerto El Dorado de la ciudad de Bogotá de Colombia, sin embargo para la recopilación de información será necesario ampliar la visión a nivel internacional.

3.2.2 Temporal

Los datos que serán recopilados para este trabajo de grado estarán enmarcados dentro del periodo 1980 – 2022 integrando en este periodo todas las alternativas tecnológicas que se encuentran actualmente para la generación de sistemas de análisis de imágenes y su segmentación a través de fuentes secundarias en áreas de seguridad operacional abarcando temas de Objetos Extraños que generen algún Daño a aeronaves teniendo en cuenta todo el tiempo pasado en el cual se han generado múltiples incidentes y accidentes.

3.2.3 Académica

El proyecto de grado planteado obedecerá con lo exigido por la Fundación Universitaria Los Libertadores entorno a su nivel investigativo y cuadro de sustentación de proyecto de grado integrando en este la información clara y concisa apoyándose en bibliografías, textos, libros, regulaciones y estudios que proporcionaran los conceptos e información necesaria para generar un sistema de análisis de imágenes usado mediante tecnologías vanguardistas como los drones la prevención de incidentes o accidentes producidos por los FOD.

3.3 Valor de la investigación

- Conveniencia

Siendo una problemática presente constantemente en problemáticas operacionales la implementación de este proyecto de grado en la industria aeronáutica generara una alternativa adicional a la prevención de accidentes aeroportuarios.

- Relevancia social

La proyección social que se tendrá será adaptada en toda persona que use el medio de transporte aeroportuario ya que al implementar este sistema se tiene como visión generar una mejor seguridad aeroportuaria evitando accidentes.

- Implicaciones prácticas

Gracias a problemáticas actuales que se presentan día a día en busca de la prevención de incidentes y accidentes aeroportuarios se evitara practicas anticuadas de inspección que buscan evitar los FOD.

4. Desarrollo del algoritmo de visión artificial

Dando inicio al desarrollo puntual del proyecto de grado se tiene claro el gran problema que se busca abordar, viendo el peligro constante y latente que se tiene en la seguridad operacional del daño que podría ser producido por objetos extraños. Partiendo de las alternativas que actualmente se tienen para la prevención de estos FOD, como lo son las tecnologías de barredoras de objetos FOD BOSS o FOD BUSTER que cumplen con la función de recoger objetos en la pista aunque de una manera poco eficiente y otras alternativas como los radares y sensores FODetect®, iferret™, Tarsier® y FOD Finder™ son posibles detectores de FOD mucho más sofisticados y costosos a diferencia de la inspección común realizada por el personal que conlleva el uso de personal humano mediante una inspección visual del terreno y así mismo el desgaste que puede tener el personal al realizar esta actividad lo cual no lograría generar un 100% de verificación de FOD y su recolección.

Viendo la gran rama de posibilidades de detección de FOD se desea aplicar la tecnología computacional en la cual mediante cámaras usadas por un Dron se lograra identificar estos objetos mediante inteligencia artificial con el entrenamiento de una red neuronal convolucional de aprendizaje profundo análisis de imágenes por medio de alternativas contemporáneas como lo son Python y Yolo.

4.1 Desarrollo del análisis de imágenes

Para iniciar con el análisis de imágenes es de suma importancia saber los pasos que conllevaran a la realización de detección de objetos que necesitaría el modelo, se ha optado por el sistema de (Jocher et al., 2022) en el cual explica cómo se puede realizar la automatización de un detector de objetos con las alternativas que da YOLO V5 y su programación mediante Python.

El objetivo en este capítulo es generar el modelo de detección con objetos siendo entrenado para poder reconocer ciertos objetos y observar cómo procedería ante la aparición de dichos objetos en imágenes o fotogramas, apoyándose en el concepto de deep learning mediante la neurona convolucional de YOLO.

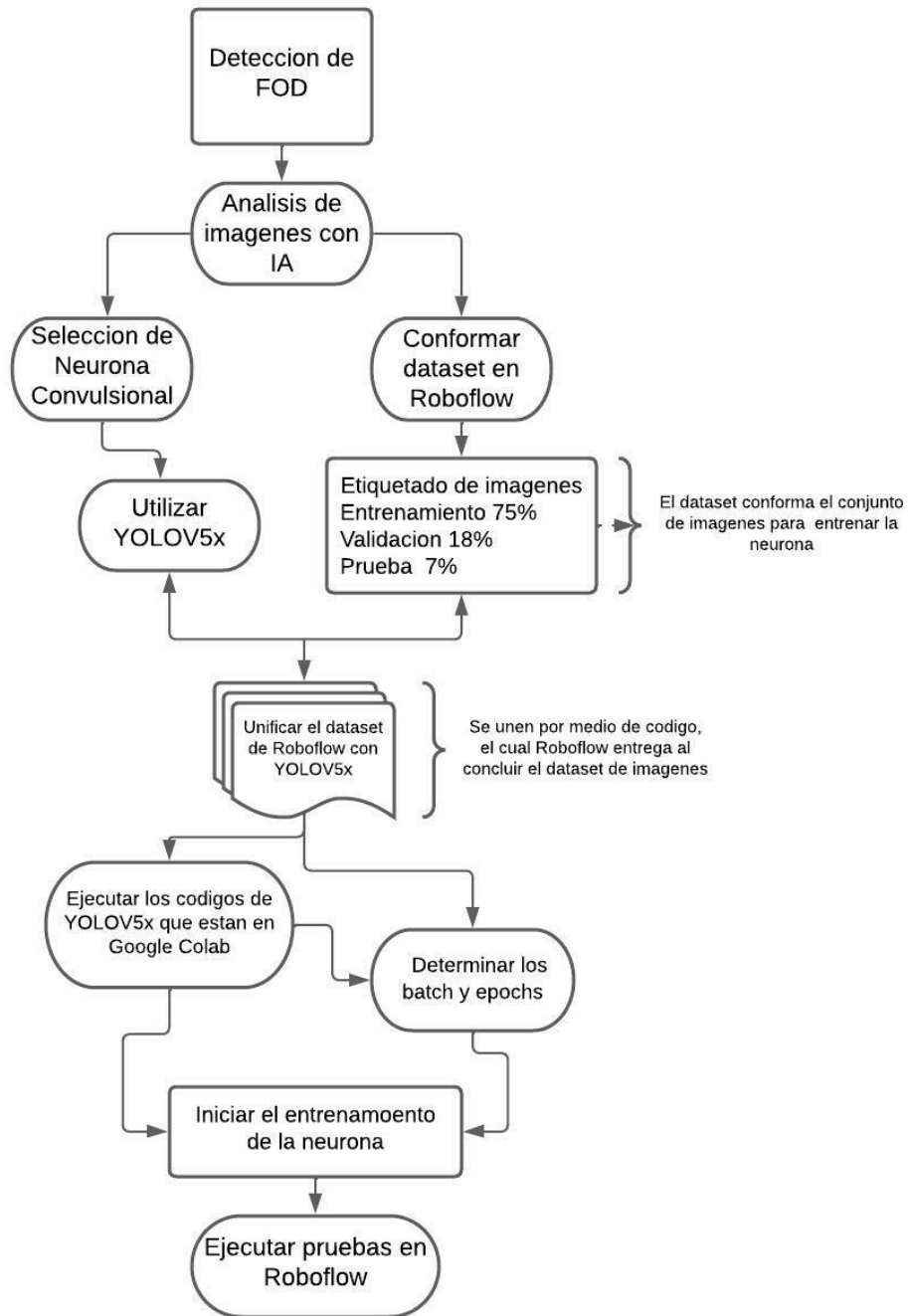


Ilustración 1: Diagrama de flujo de detector de imágenes

Autoría propia

4.2 Requisitos

- Detectar objetos

Para la detección de objetos es necesario iniciar con qué clase de características debe tener estos objetos que el programa va a identificar, identificando así si el objeto cuenta con una estructura con ciertas dimensiones y con colores comunes que caractericen a cierto objeto, identificando múltiples objetos y así identificar su clase, por esto se inicia con establecer a qué clase de objetos pertenece, en este caso se abarcan las siguientes clases:

- Arandelas
- Avión
- Tractorista
- Tuerca
- Basura

- Ubicación del objeto

Liego de tener el modelo identificando los objetos que se desean detectar este mismo debe se entregar la posición de donde se encontraran estos objetos, su ubicación en la imagen identificando su tamaño, y lugar donde se encuentra identificándolo con una región de posición mediante un cuadro de seguimiento que caracteriza el objeto con el tipo de clase a que pertenece.

4.3 Procedimiento

El procedimiento realizado para que el modelo cumpla con su detección de objetos se divide en diferentes etapas, en las cuales el sistema tendrá la capacidad de aplicar el machine learning, dependiendo de la información que se le suministre al sistema así mismo tendrá la capacidad adecuada para identificar los objetos de manera más correcta.

4.3.1 Data Set

Para iniciar en la enseñanza del modelo de detección de objetos se debe preparar un conjunto de imágenes de los objetos a analizar, de los cuales se dividirán basándose en el principio de Pareto (Dunford et al., 2014) que define una división de 80/20, empleando un 80% de las imágenes totales para uso de entrenamiento y el 20% demás para la validación del entrenamiento, aunque también se recomienda usar tres divisiones de estas imágenes en vez de dos, añadiendo el conjunto de imágenes para prueba como fue recomendado por

el software de Roboflow de 70/20/10 (Roboflow, 2022) se usó el 75% de imágenes para entrenamiento, un 18% para validación y un 7% del total de las imágenes para prueba.

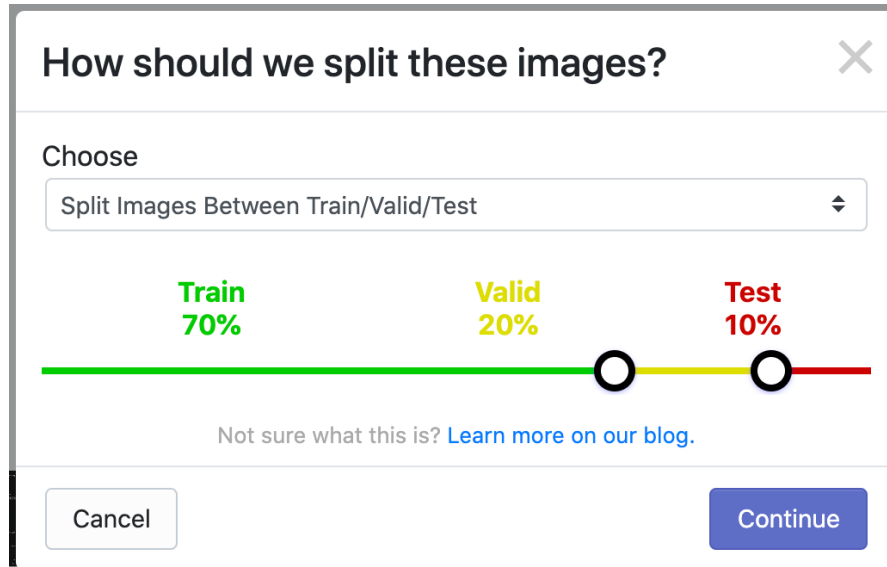


Figura 26: Recomendación Roboflow para data set de imágenes

Recuperado de (Nelson, 2020)

4.3.2 Imágenes

Al analizar la cantidad de imágenes necesarias para enseñarle al modelo de detección de objetos se recopilaron 727 de las imágenes de bases de datos en línea y las demás imágenes fueron tomadas por autoría propia en el aeropuerto y zonas similares al área que llegaría a analizar el Dron.



Figura 27: Imagen de FOD para análisis.

Autoría propia.

Para recopilar las imágenes de estos objetos en línea se utilizó la extensión de Google Chrome de Download All Images la cual al momento de entrar en un buscador de una base de datos como lo es Google se buscó imágenes de los objetos a analizar que fueran concordes con lo deseado a detectar.

Esta función de Download All Images permite descargar todas las imágenes que aparecen en la ventana que uno está utilizando, agrupándolas en un archivo “.zip”, al momento en que se realizó la búsqueda se activó esta función descargando dichas imágenes listas para ser usadas para el entrenamiento y validación del modelo.

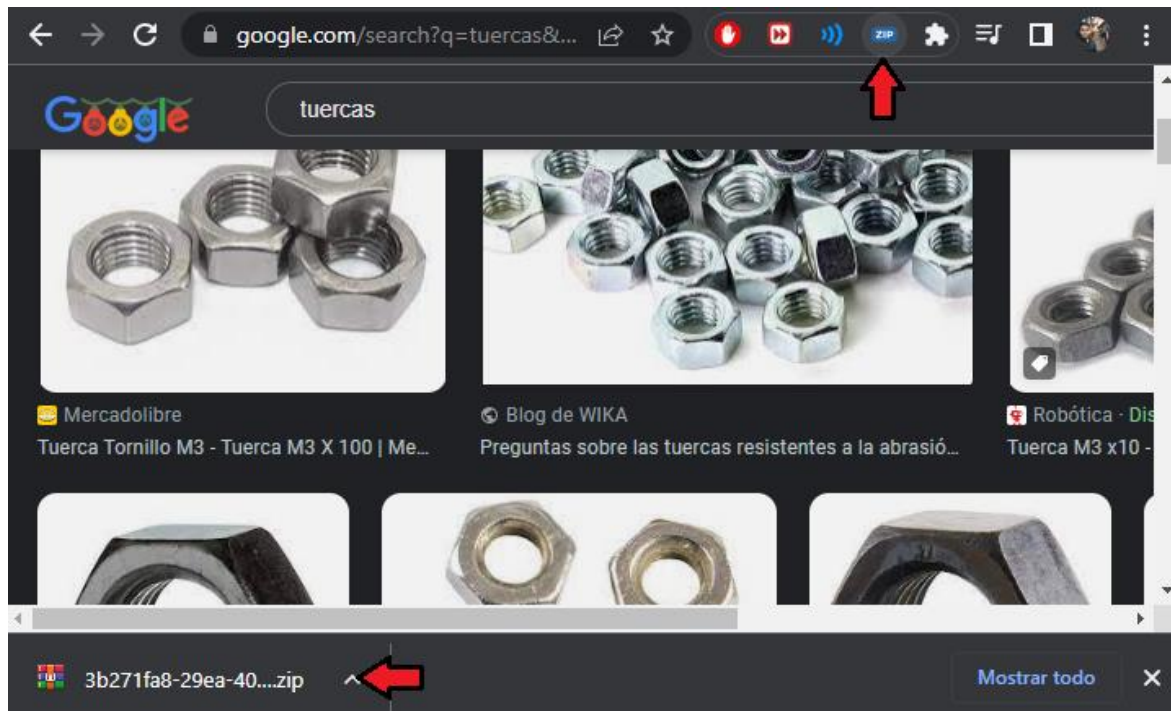


Figura 28: Funcionamiento de Download All Images.

Autoría propia.

4.3.3 Etiquetado

Luego de haber recopilado la información necesaria para entrenar el modelo, estas imágenes deberán ser etiquetadas para que el sistema identifique correctamente los objetos al momento de realizar una detección. Para este caso se usará el programa MakeSense para realizar el etiquetado de cada imagen a analizar.

Explicación de proceso de etiquetado

- Primero se debe subir las imágenes destinadas para el entrenamiento en la plataforma de MakeSense identificando que son para una detección de objetos, imagen 29.



Figura 29: MakeSense Carga de imágenes a la plataforma.

Autoría propia.

- Luego de haber cargado las imágenes a la plataforma se inició con el etiquetado de las imágenes a analizar, designando las clases de objetos que se van a analizar, imagen 30.

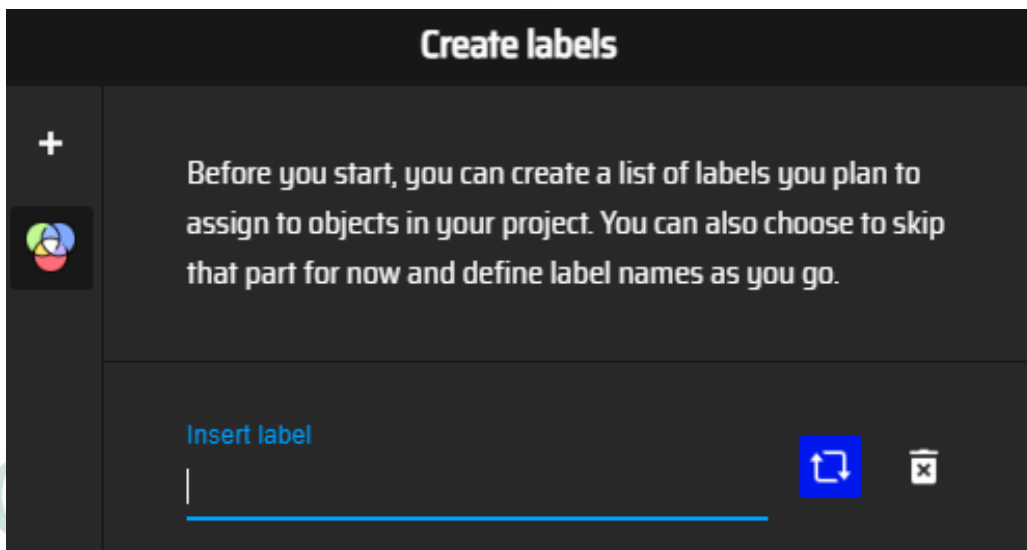


Figura 30: MakeSense Creación de clases de etiquetas.

Autoría propia.

- Al tener claro qué clase de objetos se desean analizar en el modelo se empiezan a identificar mediante recuadros de cada clase los objetos que se desean identificar en las imágenes una por una como se muestra en la siguiente imagen:

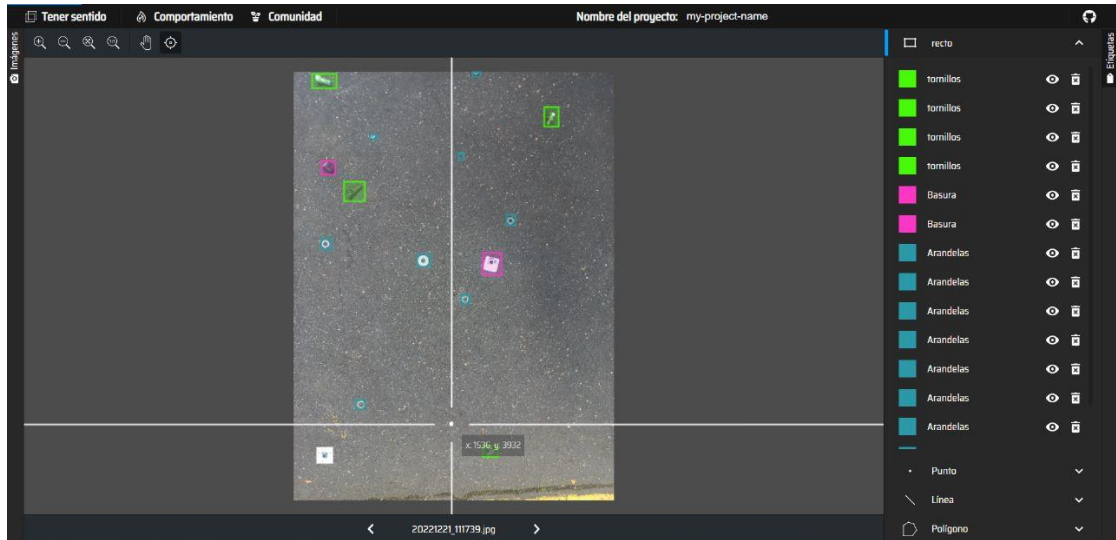


Figura 31:Etiquetado de imágenes en MakeSense.

Autoría propia.

- Luego de tener etiquetadas todas las imágenes se exportan las etiquetas de dichas imágenes en un archivo con archivos con formato de Yolo, imagen 32.

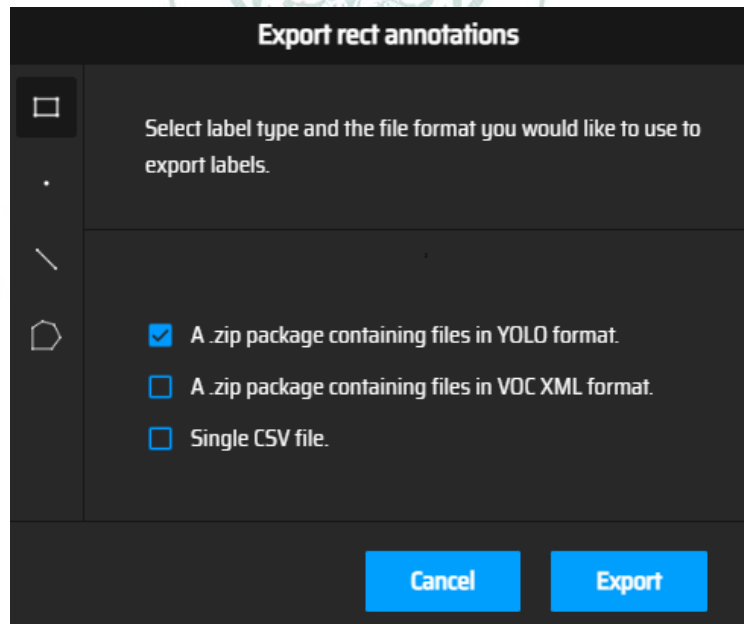


Figura 32: Descarga de etiquetas de MakeSense.

Autoría Propia.

- Teniendo exportados todas estas etiquetas, se guardan en la carpeta de “train” que se encuentra en la carpeta de “labels” siendo la misma cantidad de las imágenes anexadas en la carpeta “train” de la carpeta de “images”.

- Luego de identificar las etiquetas de las imágenes de entrenamiento se debe hacer el mismo proceso con las imágenes de validación ingresándolas en la plataforma de MakeSence y guardándolas en las carpetas “val” de “labels” e “images”.

Luego de tener completo el dataset de 1530 imágenes se procedió a cargarlas en el sistema de Roboflow en el cual se hizo una corrección y depuración de las imágenes recopiladas eliminando las imágenes y etiquetas mal hechas que no podrían funcionar en el sistema de YOLO dejando 1274 imágenes para el inicio del modelo, de las cuales se modificaron a un tamaño adecuado de uso en YOLO, modificando las imágenes a un tamaño de 416x416, como se observa en la Figura 33.

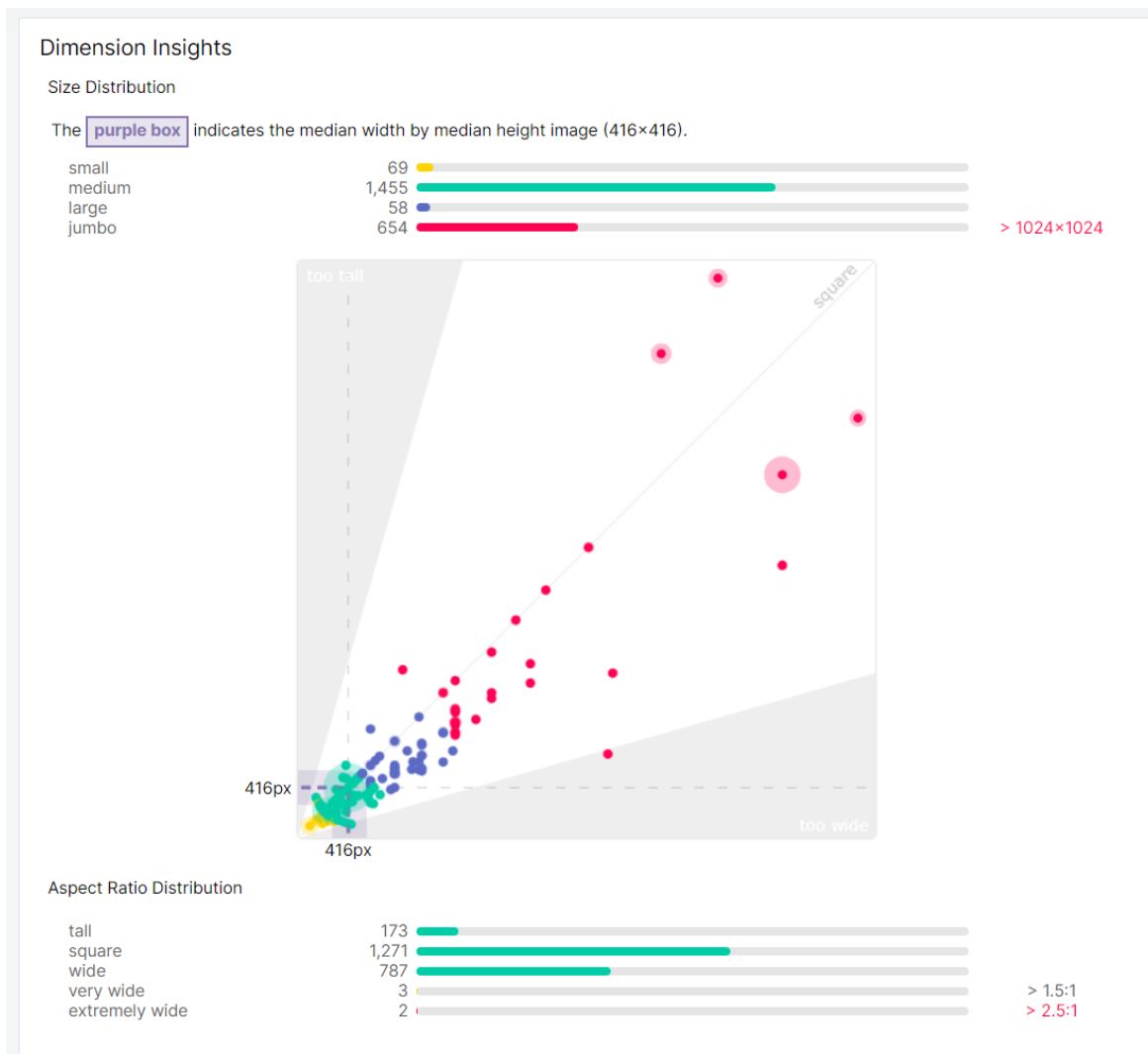
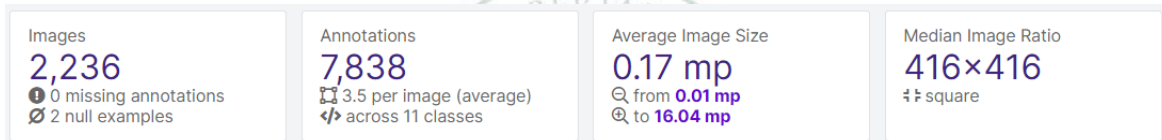


Figura 33: Modificación de tamaño de imágenes de entrenamiento por Roboflow.

Autoría propia del software de Roboflow

Para mayor eficiencia y basándose en proyectos como el de (Giraldo Jimenéz et al., 2021) en el cual utilizaron para su análisis 1400 imágenes de entrenamiento y el proyecto de (Gómez Bedoya & Castrillón, 2019) en el cual menciona diferentes data sets que contienen entre 1000 y 5000 imágenes para su desarrollo teniendo en cuenta de igual modo la recomendación de (Rosenbacher, 2022) en el cual menciona que se puede generar un modelo desde tan solo 100 imágenes hasta 10.000 imágenes, dependiendo simplemente del nivel de precisión que se llegase a requerir. Por estos diferentes aspectos es que se ha deseado aumentar la cantidad de imágenes a usar para el entrenamiento, por esto se usaron data sets ya usados en la herramienta adicional que tiene Roboflow que cuenta con diferentes datos de libre acceso de diferentes proyectos realizados, adicionando de estos proyectos 963 imágenes previamente ya etiquetadas dando un total de 2236 imágenes con 7838 etiquetas listas para el entrenamiento del modelo de detección.



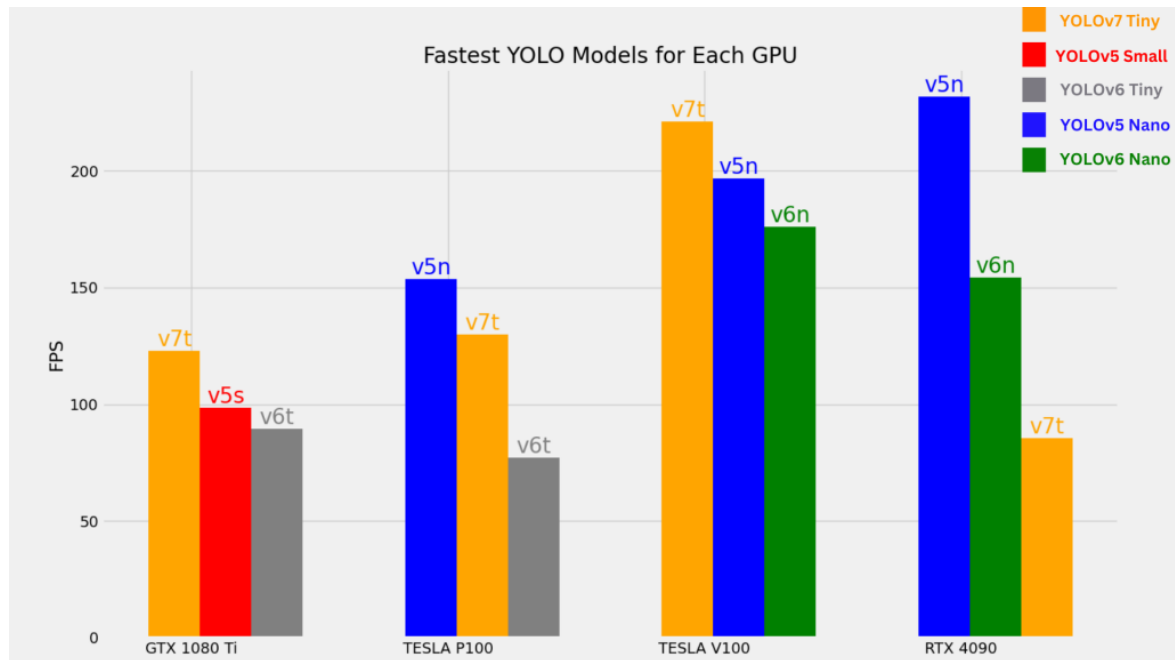
Images 2,236 0 missing annotations 2 null examples	Annotations 7,838 3.5 per image (average) across 11 classes	Average Image Size 0.17 mp from 0.01 mp to 16.04 mp	Median Image Ratio 416×416 square
---	--	--	---

Tabla 2: Datos finales de imágenes en Roboflow

Autoría propia en Roboflow

4.3.4 Entrenamiento

Contando con el Data Set necesario para el detector de imágenes se procede a iniciar con el entrenamiento del modelo, identificando la red neuronal que realizara el aprendizaje para el detector de objetos, en este proyecto se utilizó la CNN de YOLO por comodidad de uso y eficiencia de análisis por el estudio realizado por (Barba Guamán, 2021) acerca de los diferentes modelos de aprendizaje profundo de CNN, pero en si enfocándose no en la última versión de YOLO si no en su Quinta versión que en el estudio hecho por (Rath & Gupta, 2022) en el cual realizaron la comparación del comportamiento de las tres últimas versiones de Yolo como se puede ver en la siguiente gráfica:



Gráfica 1: Comparación de versiones de YOLO.

Recuperado de (Rath & Gupta, 2022)

Como se puede ver el comportamiento de la 5 versión y su 7 versión con respecto a su uso con diferentes GPU es diferente su nivel de eficiencia dejando de lado la 6 versión, aunque en su rango de cambio la diferencia no es muy grande si se llega a comparar con la capacidad computacional que necesitaría la séptima versión de YOLO con su quinta versión que no necesita tanta capacidad computacional, por eso la opción más factible para este proyecto es usar la Quinta Versión que tiene YOLO.

Identificando la versión de YOLO v5 que se usará se observa que esta versión cuenta con diferentes redes neuronales ya entrenadas y verificadas para ser usadas para el entrenamiento, como se puede ver en la siguiente tabla:

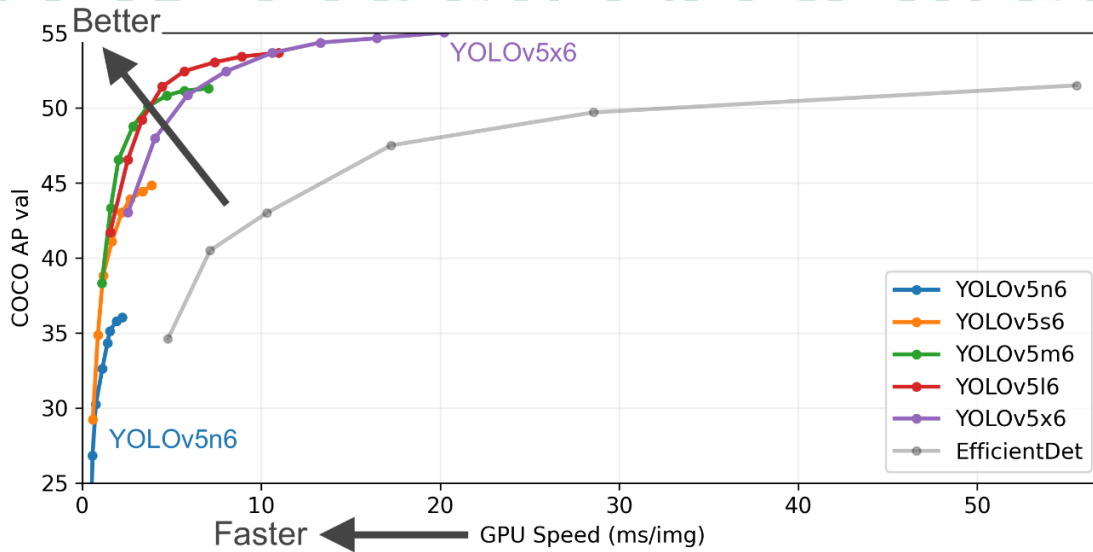
Puntos de control preentrenados

Modelo	tamaño (píxeles)	valor de mapa 50-95	valor de mapa 50	Velocidad CPU b1 (ms)	Velocidad V100 b1 (ms)	Velocidad V100 b32 (ms)	parámetros (M)	FLOP a 640 (B)
YOLOv5n	640	28,0	45,7	45	6,3	0,6	1,9	4,5
YOLOv5s	640	37,4	56,8	98	6,4	0,9	7,2	16,5
YOLOv5m	640	45,4	64,1	224	8,2	1,7	21,2	49,0
YOLOv5l	640	49,0	67,3	430	10,1	2,7	46,5	109,1
YOLOv5x	640	50,7	68,9	766	12,1	4,8	86,7	205,7
YOLOv5n6	1280	36,0	54,4	153	8,1	2,1	3,2	4,6
YOLOv5s6	1280	44,8	63,7	385	8,2	3,6	12,6	16,8
YOLOv5m6	1280	51,3	69,3	887	11,1	6,8	35,7	50,0
YOLOv5l6	1280	53,7	71,3	1784	15,8	10,5	76,8	111,4
YOLOv5x6 + TTA	1280 1536	55,0 55,8	72,7 72,7	3136 -	26,2 -	19,4 -	140,7 -	209,8 -

Tabla 3: Comparación de las versiones de YOLO v5.

Recuperado de (Jocher, 2023)

Siendo expresado esto en la siguiente gráfica comparando la velocidad de rendimiento de los modelos:



Gráfica 2: Comparación de versiones de YOLOv5

Recuperado de (Jocher, 2023)

Viendo estas comparaciones se puede ver que la versión de YOLOv5x6 es la que ofrece mayor velocidad de GPU y eficiencia de ejecución con el set de datos de COCO AP, por esto se utilizara la versión de YOLO XLarge (YOLOv5x6).

Para dar inicio a el programa se dirigió a un sitio web donde se encuentran los datos de esta red neuronal (Jocher, 2023) la cual explica todo el proceso necesario para realizar un análisis de imágenes en el cual dirige a su neurona convolucional de GoogleColab que da por defecto en el cual se podrá empezar a generar el entrenamiento mediante las líneas de código ya dispuestas y datos previamente cargados en este libro de datos:

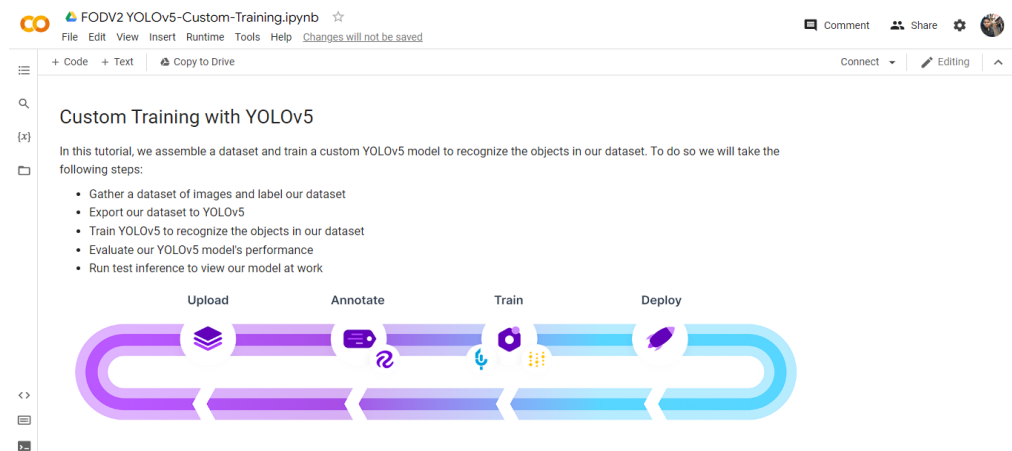


Figura 34: Google Colab – Entrenamiento personalizado con YOLOv5.

Autoria propia de (Ultralytics, 2022)

Para este proceso de entrenamiento se utilizó una capacidad computacional de una computadora de disco solido de 512 GB, una memoria RAM de 8 GB, Tarjeta Gráfica NVIDIA GTX 1650 con una RAM dedicada de 4 GB y un procesador Intel Core i5 de decima generación, dando inicio a el código predeterminado de la CNN se asigna por GoogleColab una GPU Tesla T4 de 16 GB con la cual se podrá realizar el completo entrenamiento del modelo de detección FOD, el código resultante se puede ver en el Anexo A.

```
#clone YOLOv5 and
!git clone https://github.com/ultralytics/yolov5 # clone repo
%cd yolov5
%pip install -qr requirements.txt # install dependencies
%pip install -q roboflow

import torch
import os
from IPython.display import Image, clear_output # to display images
```

```
print(f"Setup complete. Using torch {torch.__version__} ({torch.cuda.get_device_properties(
o).name if torch.cuda.is_available() else 'CPU'})")
```

Iniciando con el proceso de entrenamiento en GoogleColab después de asignársele la GPU de entrenamiento se dirige a la aplicación de Roboflow en la cual se hará una interconexión de datos entre los dos softwares, descargando la línea de código que da al finalizar el etiquetado Roboflow para integrarse en GoogleColab y así hacer la llamada de datos de imágenes y etiquetas para realizar el proceso de entrenamiento, como se puede observar:

```
from roboflow import Roboflow

rf = Roboflow(api_key="xvnat5dgt152y5h5ploz")

project = rf.workspace("deteccion-fod").project("deteccion-fod")

dataset = project.version(2).download("yolov5")
```

Activar la línea de código en GoogleColab se realiza la interacción generando las líneas de código que se pueden observar en el Anexo A.

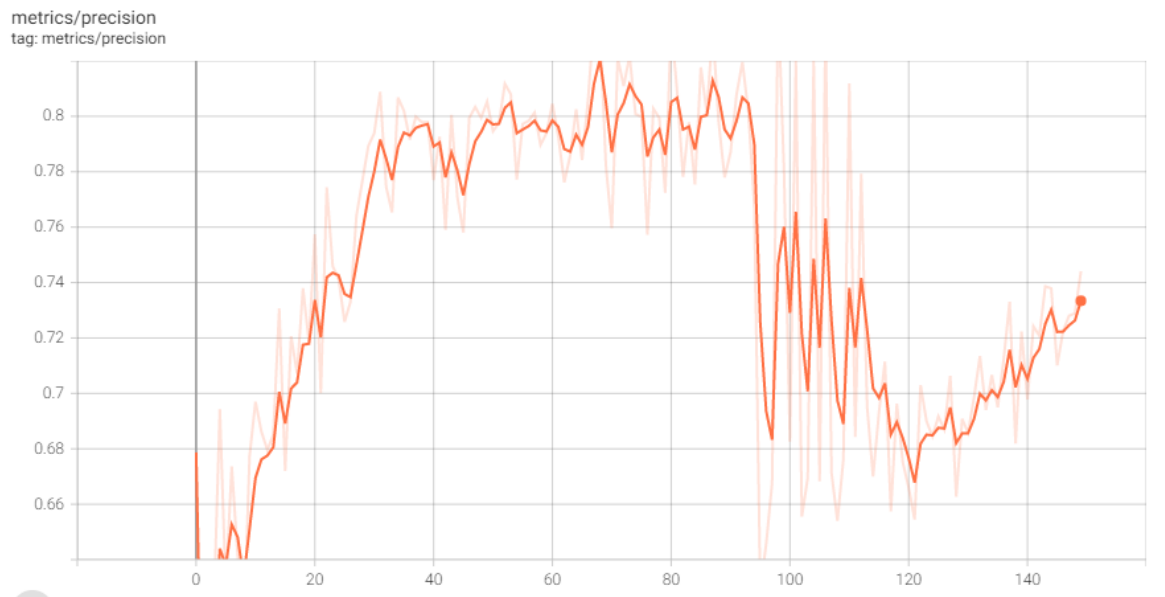
Al realizar la llamada de datos de Roboflow al código de YOLO automáticamente el sistema conecta el entrenamiento dando los datos del tamaño de imagen (img) que previamente Roboflow modifico para el correcto etiquetado las imágenes y así mismo la cantidad adecuada de parámetros de optimización (batch) que necesita el entrenamiento, junto con las épocas (epochs) o veces que debe generar el entrenamiento ejecutando los algoritmos de la red neuronal con los datos de entrenamiento, sumando a su vez la ubicación de las imágenes junto con sus etiquetas (data), sin dejar de lado la especificación del tipo de neurona que se utilizara de YOLO (weights), la cual se especifica cambiando la línea de código de “weights yolov5s.pt” por “weights yolov5x.pt ” y por ultimo los datos de las imágenes para el entrenamiento rapido (cache) se dejan como predeterminados.

```
!python train.py --img 416 --batch 16 --epochs 150 --data {dataset.location}/data.yaml --
weights yolov5x.pt --cache
```

Finalizando con el procedimiento de entrenamiento dejando que el sistema ejecute su funcionamiento con la capacidad computacional que tiene la máquina de entrenamiento y su apoyo en GoogleColab. Culminando con el ultimo código de llamada de datos de las pérdidas de entrenamiento y métricas de rendimiento del modelo de detección.

```
# Start tensorboard
# Launch after you have started training
# logs save in the folder "runs"
%load_ext tensorboard
%tensorboard --logdir runs
```

Luego de haber realizado el entrenamiento del modelo se puede hablar de los resultados de este detector de imágenes, al tener la neurona completamente entrenada se puede calificar la eficiencia del sistema gracias a las épocas de entrenamiento que este obtuvo, como se puede observar en la siguiente grafica se indica el proceso de entrenamiento dado:



Gráfica 3: Precisión del modelo con respecto a sus épocas de entrenamiento.

Autoría Propia del modelo de entrenamiento.

Al generar esta grafica de precisión se puede identificar el tiempo completo del entrenamiento el cual como resultado fue de 2 horas con 34 minutos y 28 segundos con un promedio de entrenamiento de cada época en la cual el sistema enseña al sistema mediante las imágenes de entrenamiento a la neurona, dando un promedio de entrenamiento de cada época de 55 segundos.

Teniendo estos resultados se puede observar el punto más efectivo que se dio al entrenar la neurona de detección dando así un porcentaje de precisión de 83,36% en su época 68 en la cual alrededor de esta cantidad de épocas se ve su precisión entre valores de 75%.

Name	Smoothed Value	Value	Step	Time	Relative
train/exp	0.8205	0.8336	68	Sat Dec 31, 12:57:59	1h 11m 16s

Figura 35: Valor máximo de precisión del modelo.

Autoría Propia del modelo de entrenamiento.

Una demostración generada del sistema se puede ver en las siguientes imágenes identificando los FOD en el suelo:

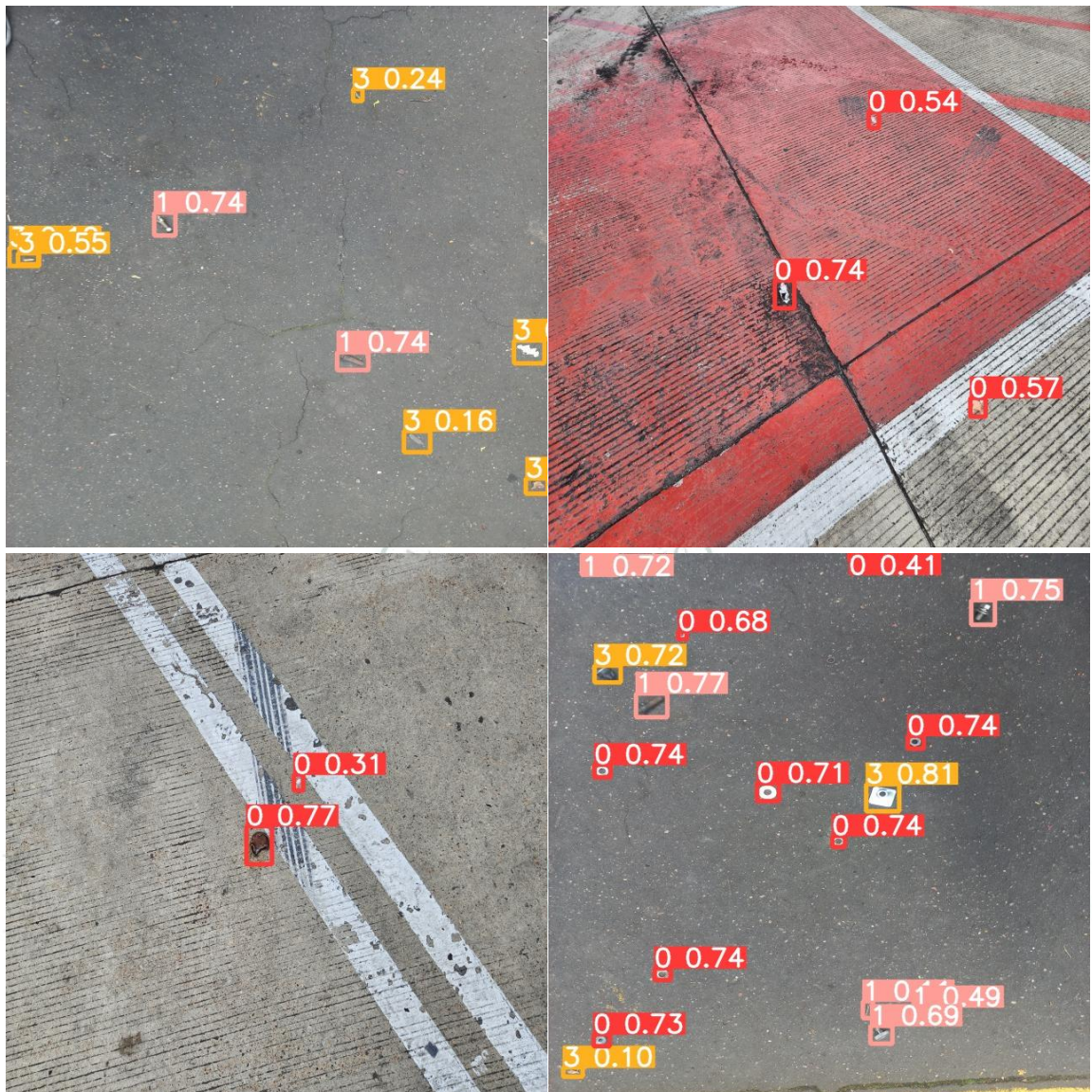


Figura 36: Resultado de imágenes de entrenamiento en la CNN.

Autoría Propia por modelo entrenado.

El porcentaje de precisión de detección de estos objetos se realizó internamente en este entrenamiento de YOLO usando la alternativa de la métrica de la ecuación de la figura 23 en donde se utilizó el método de intersección sobre unión donde se identifica que el funcionamiento de un sistema de detección para que logre grandes resultados debe tener un porcentaje superior a $>70\%$.

5. Criterios de selección del dron

5.1 Características del UAS

Al tener el modelo de inspección de FOD, listo para ser usado, con un porcentaje de precisión adecuado para iniciar su funcionamiento de detección de objetos mediante análisis de imágenes, en base de su CNN se entró en el contexto de identificar qué tipo de UAS es el adecuado para ejecutar la detección mediante el modelo. Por esto se partió en la búsqueda de que tipo de dron beneficiaria a la inspección que desea hacer el modelo de detección de FOD, siendo así se inició en la búsqueda de los tipos de UAS que hay en la actualidad. Se determina que para la detección de FOD por medio de UAS, se requiere de un dron con unas características específicas mínimas de vuelo y video como lo son:

- Tipo: Teniendo como objetivo realizar inspecciones es necesario el uso de un dron multirrotor, ya que por su tipo se mantiene una estabilidad aceptable para lograr utilizar el detector de objetos mediante cámaras y sensores.
- Autonomía: Para cumplir con una inspección, por lo mínimo cuenta con una autonomía de 18 minutos, con el fin de inspeccionar una pista de vuelo por secciones, considerando que se inspeccione un área mínima de 200.000 m².
- Tipo de vuelo: El UAS debe permitir la planificación de distintos tipos de vuelos (inspecciones), ya que las áreas de inspección en un aeropuerto varían según la operación, debe permitir vuelo combinado, automático y vuelo manual, con el fin de poder recuperar el dron en caso de emergencia.
- Estabilizadores: Debe contar con un estabilizador de 3 ejes (x, y, z), tanto para el vuelo, como para la cámara. Con el fin de disminuir la perturbación del vuelo por vibraciones, el cual permita que el dron tenga ángulos de inclinación entre -90° y +30°.

- Cámara: Se requiere de un lente mínimo de 1 pulgada, el cual capture videos en una resolución de 1080p, siendo la toma de video preferiblemente en 4k a 30 FPS, el cual permita el movimiento de cámara entre -90° y 90° , capturando video en formato RGB.
- Sensores: Contar con un sensor de geolocalización mínimo GLONASS, el cual permita la geolocalización satelital aumentando la precisión del plan de vuelo, con una conexión compatible a distintas bases de control.
- Velocidad de vuelo: Para la operación de detección de objetos debe contar con una velocidad optima de 5 m/s.
- Rango de operación: Para la correcta operación de vuelo el dron debe contar con un rango de operación de 3 km de recepción mínima.

5.2 Recomendación de dron

Teniendo en cuenta las características mínimas necesarias para el UAS, se puede recomendar unos drones que actualmente ofrece el mercado con estas especificaciones:

Dron	Resolución	Tiempo de vuelo (min)	Velocidad (m/s)	Precio (USD)
DJI Inspire 1 (DJI, 2014)	4K a 30 FPS	18	22	1.490
DJI Mavic Pro (DJI, 2016)	4K a 30 FPS	27	18	1.999
DJI Phantom 3 Profesional (DJI, 2015)	4K a 30 FPS	23	16	1.799
Yuneec Typhoon H (Yuneec Futurhobby, 2016)	4K a 30 FPS	25	19	1.499
Yuneec Q500 4K (Yuneec futurhobby, 2015)	4K a 30 FPS	25	18	1.699

Tabla 4: Recomendación de drones

6. Caracterización de vuelo

Al momento de volar un dron, se debe tener en cuenta la seguridad del mismo y de los terceros, por tal motivo se debe planificar el vuelo para cada misión a ejecutar, la cual debe cumplir las regulaciones aéreas, y se deben seguir ciertos pasos para dar cumplimiento a la misión de forma segura y correcta. El plan de vuelo previene y clarifica cada paso que se deberá ejercer en la misión.

6.1 Política de operación

- Al momento de volar, la seguridad es el primer paso de la misión, ya que cualquier incidente que ponga en riesgo la seguridad de personas y del mismo equipo, serán un motivo para la cancelación de la misión.
- No se ejecutará una misión sin el cumplimiento a un plan de vuelo estrictamente basado en la seguridad, lo que implica que se deberá de analizar el área de trabajo o donde se ejecutara la misión, con el fin de identificar los obstáculos que están presentes, para así poder programar las líneas de vuelo de forma correcta.
- En misiones de fotografía o vídeo donde se deba seguir una ruta automática y no se tenga control sobre el vuelo, se debe estar atento a cualquier contratiempo y en lo posible mantener a la vista el Dron o supervisarlos por medio de su sistema de FPV para reaccionar de inmediato según el protocolo de seguridad. El monitoreo del vuelo debe ser contante, ya que una pérdida de contacto de la telemetría superior a un minuto o un contacto intermitente, que interfiera en la correcta recepción de datos, será un motivo para la cancelación de la misión.
- Se deberán de revisar las baterías de los equipos, tales como sensores, cámaras, monitores, controles de mando, sistemas de video y equipos de apoyo, verificando que estas estén cargadas y funcionales, antes de iniciar una misión.

- Se deben considerar que las condiciones meteorológicas sean las óptimas, garantizando la seguridad del dron y de las personas que operan en la misión. Las condiciones de visibilidad deben asegurar que el dron este a la vista directa mínimo al punto más lejano y de ahí en adelante hacer uso de las herramientas FPV si se tiene. Está prohibido iniciar un vuelo con condiciones de humedad visible (llovizna, lluvia o rocío) así como también en la presencia de descargas eléctricas.
- Esta estrictamente prohibido que personal ajeno al piloto o personal sin el debido entrenamiento certificado tenga acceso al control del Dron. Solo los pilotos debidamente entrenados y certificados pueden tener el control del equipo para su operación de vuelo.
- Ningún tripulante podrá ejercer sus funciones en operación de vuelo mientras esté bajo la influencia de drogas o haya consumido alcohol al menos 24 horas antes de la asignación.

6.2 Planeamiento de Vuelo

Cuando un vuelo de fotografía es autónomo, debe ser planificado de forma detallada, garantizando la seguridad de la misión, la calidad tanto de las imágenes como de los videos a obtener, por tales razones el plan de vuelo automático se deberá programar de forma meticulosa, casi el 100% del vuelo se da de un modo automático por esto el conocimiento del programa y los procedimientos de planeación de un vuelo será de gran necesidad para mantener una buena resolución de imágenes y videos (Aerial Insights, 2017). Se deberán de tener en cuenta unos sencillos pasos.

- Evaluar el área a sobrevolar, los sensores y la programación, validando los parámetros que se deberán de tener en cuenta para la misión.
- Se deben obtener los datos de la ubicación de vuelo y de los obstáculos, para poder realizar la programación de las áreas, garantizar una distancia segura a los obstáculos presentes, Se debe tener en cuenta que los obstáculos naturales pueden haber crecido referente a las imágenes que se utilizan para la planificación.
- El tamaño del área programada deberá de ser acorde a la duración de la batería, es decir la autonomía del dron, Sin embargo, áreas más grandes pueden ser programadas y divididas en varios vuelos como corresponda para que el levantamiento sea cubierto en su totalidad.

Teniendo en cuenta los puntos anteriores, se realiza la planificación de vuelo para la inspección de FOD (Hsu et al., 2019). Se debe tener en cuenta el solape (traslape) lo que

indicara la precisión y permite que el recorrido de la aeronave sea mayor y se puedan obtener mejores datos en los videos o imágenes tomar(CIAT, 2018), Se cuentan con distintos métodos donde los principales son:

- Forma de Cuadrícula: contiene solapes mínimos entre 75% frontal y 60% lateral, la cámara estará totalmente perpendicular apuntando al suelo.

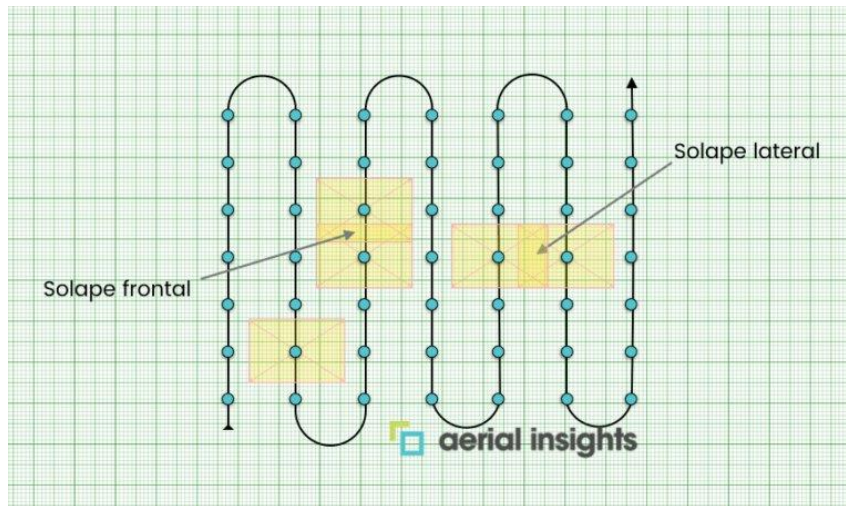


Figura 37: Forma de cuadrícula para plan de vuelo

Recuperado de (Aerial Insights, 2017)

- Doble cuadrícula: utilizado para canteras, zonas montañosas y áreas construidas, con el fin de hacer reconstrucciones en modelos 2D y 3D

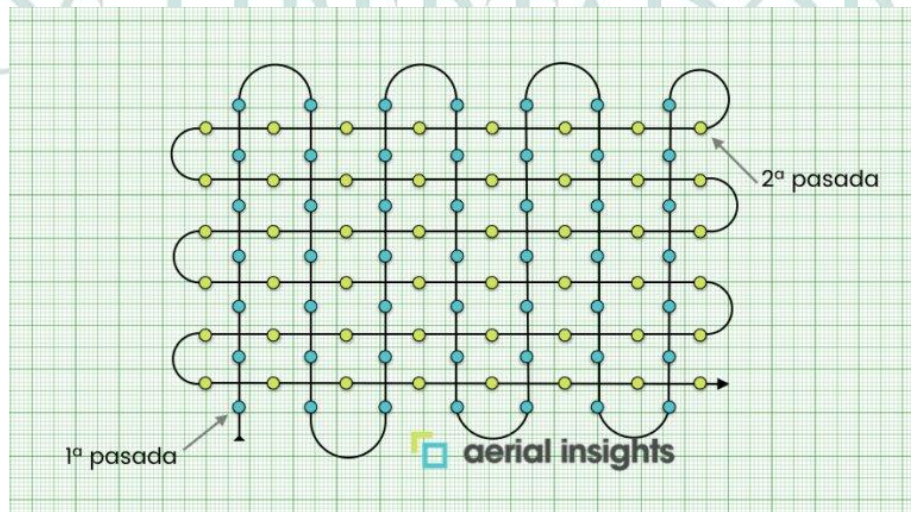


Figura 38: Forma de doble cuadrícula para plan de vuelo

Recuperado de(Aerial Insights, 2017)

- Tres trayectorias: Tiene el fin de inspeccionar estructuras levantar volúmenes y realizar modelos 2D y 3D

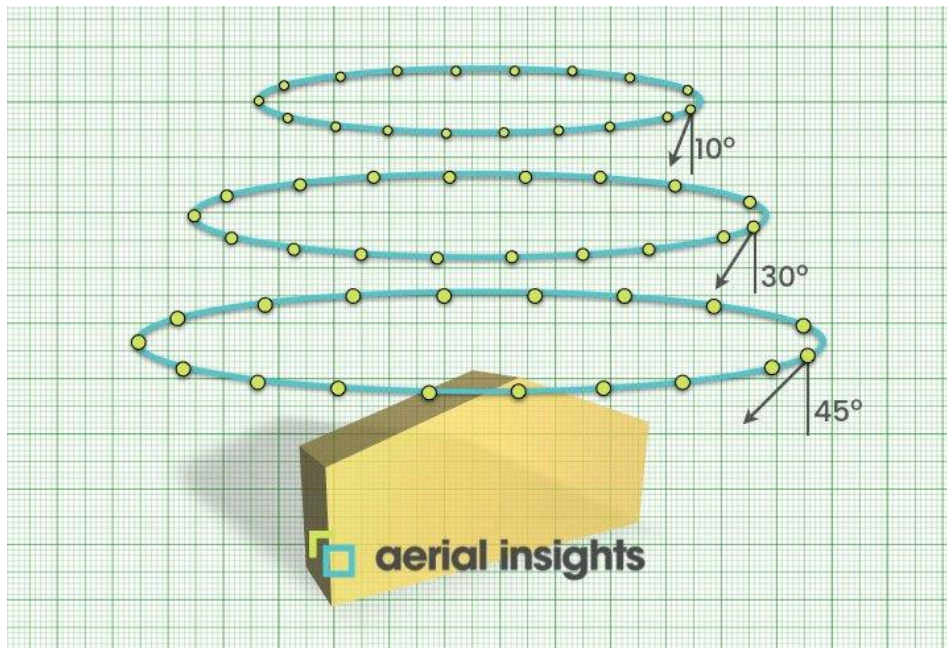


Figura 39: Forma de tres trayectorias para plan de vuelo.

Recuperado de (Aerial Insights, 2017)

Plan de vuelo

Se determina que la mejor forma de planificar el vuelo será por el método de cuadrícula, ya que permite obtener un video detallado y en dos posiciones con un solape entre los valores óptimos de 60% y 90%, la planificación no solo se limita a este método, se pueden combinar dependiendo del terreno y del objetivo de la misión.

Se propone un plan de vuelo, en el Aeropuerto Internacional El Dorado, para el cual se tiene en cuenta los obstáculos y el tipo de terreno, se planificará el vuelo con un UAV DJI Phantom 3 Pro, con una cámara 4K el plan de vuelo se planifica de la siguiente manera:

- Cuadrícula simple

Con la aplicación DroneDeploy se planifica vuelo de inspección para detectar elementos FOD, en la pista 13R-31L en la posición de la cabecera Sur, el terreno de la pista es plano con una longitud de 1000m y un ancho de 300m, el dron inspeccionara un área de 300.000m² como se observa en la figura 40, donde se muestra la aérea marcada para la inspección de la pista en busca de elementos FOD.



Figura 40: Cuadrícula simple del plan de vuelo

Autoría propia

Con la implementación de drones para la detención de FOD por medio de IA se pueden determinar los siguientes tiempos de inspección para un área de 300.000m².

Altura (mt)	Velocidad (m/s)	Tiempo de Vuelo (minutos)	Tiempo Video (minutos)	Baterías (Unidades)
1.5	5	7:15	7:20	1
2	5	7:18	7:20	1
5	5	7:22	7:20	1
8	5	7:28	7:20	1
10	5	7:31	7:20	1

Tabla 5: Plan de vuelo en pista

Autoría propia

- **Cuadrícula**

Con la aplicación DroneDeploy se planifica vuelo de inspección para detectar elementos FOD, entre Muelle Nacional Norte y Muelle Nacional Sur, con una longitud de 370m y un ancho de 270m, el dron inspeccionara un área de 99.900m² como se observa en la figura

41, donde se muestra la aérea marcada para la inspección de la zona de Muelles Nacionales en busca de elementos FOD.

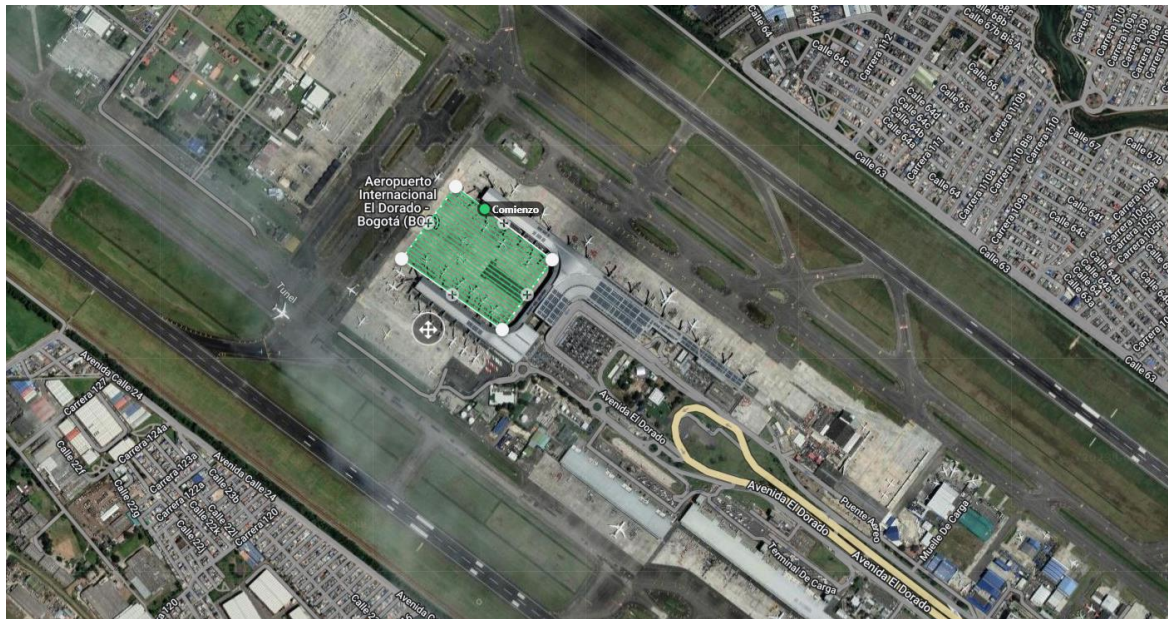


Figura 41: Cuadrícula del plan de vuelo rampa

Autoría propia

Con la implementación de drones para la detección de FOD por medio de IA se pueden determinar los siguientes tiempos de inspección para un área de 99.900m².

Altura	Velocidad (m/s)	Tiempo de Vuelo (minutos)	Tiempo Video (minutos)	Baterías (Unidades)
1.5	5	40	33:40	2
2	5	40	33:40	2
5	5	40	33:40	2
8	5	35:10	33:40	2
10	5	34:12	33:40	2

Tabla 6: Plan de vuelo en pista

Autoría propia

7. Implementación

7.1 Resultados de entrenamiento de modelo

Al complementar todo el proceso de entrenamiento del modelo de detección de objetos se procedió con las pruebas de vuelo y validación en campo, usando las instalaciones de la Fundación Universitaria Los Libertadores con el uso del dron DJI Mini pro 2 con el cual contaba la institución.



Figura 42: Pruebas de vuelo en la FULL

Autoría propia

Contando con este dron se empezaron a realizar pruebas de vuelo de funcionamiento para lograr identificar la capacidad que tendrían las imágenes en diferentes alturas con respecto al suelo, tomando capturas en alturas de:

- 1.5 m
- 2 m

- 5 m
- 8 m
- 10 m

De las cuales se obtuvieron los siguientes resultados a 1.5 metros se obtuvo porcentajes de precisión desde 18% hasta 78%.

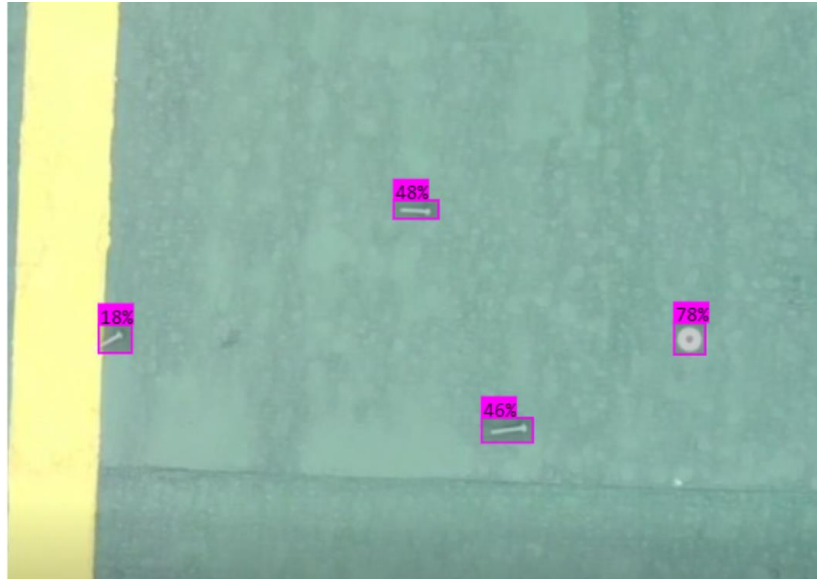


Figura 43: Imagen a 1.5 m



Figura 44: Imagen a 1.5m

Calculando el promedio de las detecciones en las imágenes se puede obtener una precisión aproximada de 47 % a la altura de 1.5 m.

Teniendo estos valores se puso a prueba la neurona de detección en las imágenes tomadas a 2 metros de altura dando valores de precisión desde 43% hasta 74% dando un promedio de 58 %.

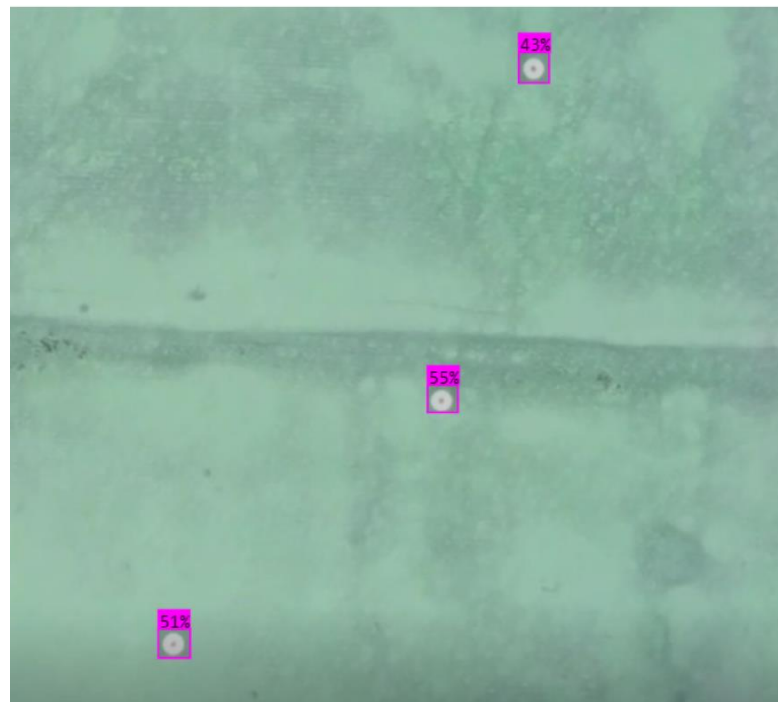


Figura 45: imagen a 2m



Figura 46: Imagen a 2m

Luego se realizó de nuevo el análisis, pero a una altura de 5 metros con valores desde 9% hasta 42%, dando un promedio de 22% de precisión contando que deja objetos dispersos sin reconocer si quiera un porcentaje.

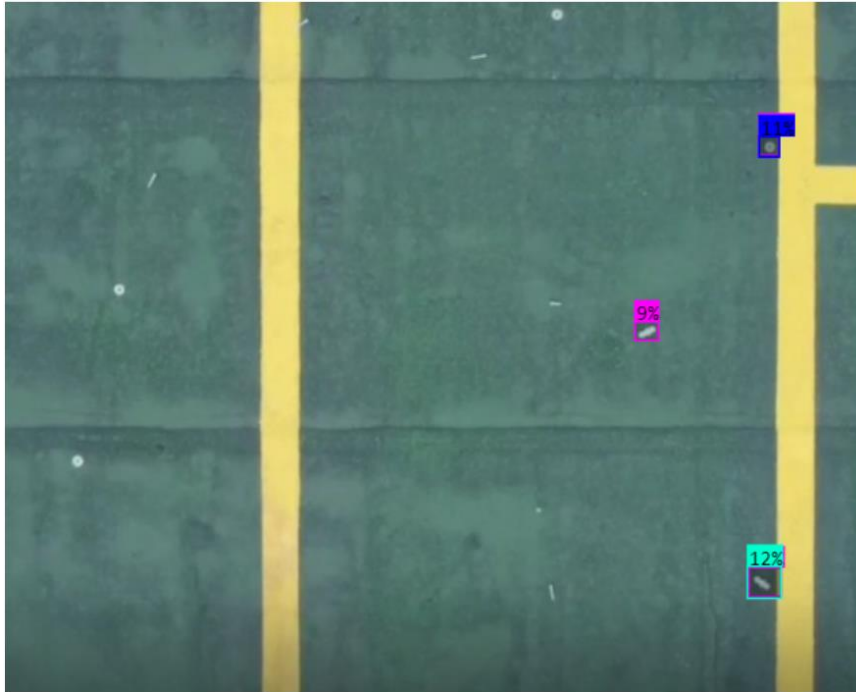


Figura 47: Imagen a 5m



Figura 48: Imagen a 5m

Procediendo a analizar la conducta de la neurona con imágenes tomadas a 8 metros obteniendo resultados de objetos de mayor tamaño, pero ignorando los de menor que son también el objetivo de esta prueba, dejando solo una imagen con análisis de precisión de 34% y dando como respuesta el ultimo análisis de la imagen a una altura de 10 metros sin ninguna identificación en el análisis.



Figura 49: Imagen a 8m



Figura 50: Imagen a 10m

En resumen, del análisis de imágenes que se le hizo a estas se puede ver en la siguiente tabla:

Altitud (mt)	Precisión (%)
1.5	47
2	58
5	22
8	34*
10	0

Tabla 7; Precisión de detección de imágenes en dron con respecto a la altura
Teniendo estos valores se puede hacer una conexión entre la tabla y la tabla 7, como se puede ver a continuación:

Altura (mt)	Velocidad (m/s)	Tiempo de Vuelo (minutos)	Tiempo Video (minutos)	Baterías (Unidades)	Precisión (%)
1.5	5	7:15	7:20	1	47
2	5	7:18	7:20	1	58
5	5	7:22	7:20	1	22
8	5	7:28	7:20	1	34*
10	5	7:31	7:20	1	0

Tabla 8: Plan de vuelo en pista con valores de precisión

Autoría propia

Se puede observar que el mejor plan de vuelo en pista con respecto a la precisión es que se da a la altura de 2 metros, manteniendo una velocidad de 5 m/s y un tiempo de vuelo que no es el más bajo quedando en 7 minutos con 18 segundos y el mayor rango de precisión para detectar objetos de 58%.

De igual modo se puede hacer la misma comparación con el plan de vuelo de rampa como se observa en la siguiente tabla:

Altura	Velocidad (m/s)	Tiempo de Vuelo (minutos)	Tiempo Video (minutos)	Baterías (Unidades)	Precisión (%)
1.5	5	40	33:40	2	47
2	5	40	33:40	2	58
5	5	40	33:40	2	22
8	5	35:10	33:40	2	34*
10	5	34:12	33:40	2	0

Tabla 9: Plan de vuelo en rampa con valores de precisión

Autoría propia

Ahora se puede interpretar que el mejor plan de vuelo en pista con respecto a la precisión es que se da a la altura de 2 metros y 1.5 metros, manteniendo una velocidad de 5 m/s y un tiempo de vuelo que siendo el mayor de 40 minutos y el mayor rango de precisión para detectar objetos de 58%. Entrando en discusión por el rango de precisión de 34% que podría ser el mejor con respecto a su altura de 8 metros y el segundo menor tiempo de vuelo de 35 minutos con 10 segundos, siendo poco fiable por su rango de detección ya que no tiene una variación de promedio a dos datos de guía, como las demás precisiones que contaban con seis datos base de promedio.

7.2 Tiempo de Inspecciones

Las inspecciones actualmente llevan un procedimiento simple, que se basa en una inspección visual del personal encargado de esta tarea, en zonas de rampa donde contantemente se presentan FOD se debe hacer una inspección y según la (EASA, 2019) estas inspecciones no deben durar más de 10-15 minutos en zonas de parqueo de aeronaves de fuselaje estrecho y de 20-25 minutos de fuselaje ancho, dejando de lado las secciones es de zonas de carga que pasarían este lapso de tiempo.

Sumando así a estas inspecciones las que se deben hacer en pista que según la, se deben hacer cuatro diarias dedicando aproximadamente 15 minutos por pista con dos recorridos, usando los vehículos de inspección del aeropuerto.

Sin dejar de lado los posibles factores que demorarían esta tarea en rampa, como lo es el embarque y desembarque de pasajeros; Servicios; Abastecimiento de combustible;

Mantenimiento; Manejo de equipaje; y actividades de los operadores que interferirían en las tareas de dicha inspección (OACI, 1983).

Integrando así con las limitaciones en inspecciones mucho más detalladas a las rutinarias que por su lapso de tiempo de inspección no pueden garantizar la mejor calidad de inspección dejando un gran porcentaje de detección de objetos inconcluso, por esto otra restricción que se pone en inspecciones más rigurosas es que se debe procurar mantener una velocidad lo más baja posible, porque a mayor velocidad menor eficacia como se mencionó anteriormente, teniendo como límite en aeropuertos como El Dorado de 30 Km/h en plataforma, 15 Km/h en horarios de 18:00 a 6:00 y en el diamante de seguridad del avión a 5 Km/h (OPAIN, 2018). Dejando las inspecciones más importantes como tarea que realiza el personal que pertenece al departamento de mantenimiento haciendo el recorrido a pie identificando más fácilmente los FOD, pero en un tiempo más prolongado.

Haciendo cálculos con los datos recogidos de podría ver la efectividad que llegaría a tener el sistema de detección de pista realizando la inspección que haría el personal encargado al momento de realizar inspecciones en vehículos designados enfocándose en la eficiencia de estas inspecciones tomando la recomendación de la OACI y así mismo si mantuviera la velocidad de inspección de 5 km/h tendría que inspeccionar toda la pista más larga que tiene el dorado de 13R/31L que cuenta con la distancia de 4400 metros (IDEAM, 2012) demoraría una hora con ocho minutos y nueve segundos, un cálculo sencillo que podría llegar a definir la viabilidad del sistema de detección de objetos mediante drones.

Factor	Inspecciones actuales	Inspecciones con IA usando UAS
Tiempo	La realización de estas inspecciones se da en lapsos de tiempo superiores a 30 minutos superando 1 hora de inspección	La aplicabilidad de estas inspecciones depende del sistema de detección pueden durar menos de 34 minutos llegando a ser lapsos mucho menores
Costos	El uso de personal exclusivo de plataforma puede llegar a costos de personal humano constantemente para estas tareas dependiendo de su inspección	El uso de drones y algoritmos reduce costos de funcionamiento siendo el personal humano disminuido evitando costes constantes de personal

Recursos humanos	La detección de objetos en pista se debe realizar cotidianamente y entran los factores humanos por la efectividad de la vista del personal al detectar objetos podría fallar	Siendo un sistema que no es afectado por fallas humanas funcionando por una maquina se gana una efectividad constante sin fallos de detección
Regulaciones	Es un sistema probado y en constante uso, a nivel mundial ya se encuentra adaptado y su funcionamiento es factible	Tiene el reto de integrarse en regulaciones nuevas para integrarse en el sistema aeronáutico actual

Tabla 10: Comparativa de tipos de inspecciones

Autoría propia



LOS LIBERTADORES
FUNDACIÓN UNIVERSITARIA

8. Conclusiones y recomendaciones

8.1 Conclusiones

Como culminación de este proyecto se puede basar en los resultados del proyecto que pueden dar una idea del aspecto práctico del proyecto expuesto.

El uso de tecnología artificial es una tecnología que a medida en que se adentra en ella se puede generar grandes soluciones, en este caso el uso de ella mediante YOLO ha sido una decisión acertada ya que se logró obtener un porcentaje adecuado de 83% para análisis de imágenes y detección de objetos.

La adaptación del sistema de análisis de imágenes con la tecnología de los drones resulta muy efectiva, siendo una idea de partida tiene una gran aplicación para detectar objetos FOD en pistas de aeropuertos.

Al momento de generar el plan de vuelo del sistema se podría pasar ya a la prueba de este observando el comportamiento que tendría y que mejoras se le podrían llegar a hacer en cuestión de prueba de detección de objetos.

Se puede concluir, al momento de planificar el vuelo en la pista 13R-31L, que el tiempo de vuelo disminuye a menor altura debido a que no se presentan obstáculos, adicional el dron tendrá mejor estabilidad al momento de realizar la inspección ya que no peleara tanto con los fuertes vientos que se pueden presentar.

Para las inspecciones de FOD se requieren cámaras de alta calidad, cámaras de 720p HD en adelante, las cámaras 4K brindan las mejores imágenes para el reconocimiento de objetos FOD, ya que estos objetos suelen camuflar en el pavimento, por tal motivo la definición de la imagen suma gran importancia, en la precisión de los objetos FOD

El tipo de terreno a inspeccionar determina el tiempo de operación, los tiempos pueden variar en una misma área según sea la cantidad de obstáculos, tipo de terreno si es montañoso o plano, se deben tener en cuenta estos factores al momento de planear una inspección.

Los FOD se logran identificar con una mejor precisión a una altura de entre 2 y 5 metros, llegando a incluir a 1.5 metros, esta distancia permite que el dron en áreas despejadas realice inspecciones en una excelente relación tiempo vs calidad de imagen, lo que permite que la detección de objetos sea más precisa y segura.

Las áreas de Muelles tardaran más tiempo en ser inspeccionadas, la cantidad de obstáculos aumenta el tiempo de vuelo, debido a que el dron debe realizar más maniobras al evadir los obstáculos que se ingresan al plan de vuelo. El sistema de DroneDeploy permite activar las funciones para evadir obstáculos, esto protegerá el dron y la seguridad de terceros ante objetos no previstos en el plan de vuelo.

Dependiendo de las distancias que se fueran a inspeccionar se podría ver la viabilidad de no solo el uso de un simple dron de inspección, si no una flota designada a la inspección de zonas para detectar de una manera más eficiente.

La comparativa del tiempo que normalmente se toma el personal del aeropuerto con respecto a la eficiencia del uso de drones es sustancial ya que una inspección eficiente con el uso de vehículos superaría la hora, siendo perjudicial para la operación de un aeropuerto, por esto la inspección mediante drones podría ser una gran alternativa de uso.

8.2 Recomendaciones

Como recomendación en base al análisis de imágenes por medio de la red neuronal convolucional se podría adicionar un banco de imágenes mucho mayor y con imagenes mucho más parecidas a lo que se llegaría a ver en una pista.

Se podría profundizar más en el proyecto llevándolo a pruebas reales en un aeropuerto viendo la viabilidad de implementar este proyecto en las normativas de seguridad operacional.

Al igual adicionar para la ampliación de este proyecto se puede implementar a un sistema más robusto adicionando un set de datos más grande y a su vez pasarlo a un funcionamiento en tiempo real.

Modificaciones a las normativas sería un reto para la implementación de este proyecto, pero asi mismo por su posible futuro se podría ver las formas de recomendar su uso y asi integrarse no solo en la seguridad operacional si no en las normativas de los reglamentos de aviación

Como foco principal de este proyecto se busca fomentar la investigación de este tema, actualmente en Colombia no se hace mucha investigación en temas de prevención de FOD, por esto el propósito de este proyecto es ser el punto de partida para futuras investigaciones de este tema en el país.

A. Anexo: Código solución de entrenamiento del modelo

Resultado de primera línea de código de YOLO:

```
Cloning into 'yolov5'...
remote: Enumerating objects: 14906, done.
remote: Total 14906 (delta 0), reused 0 (delta 0), pack-reused 14906
Receiving objects: 100% (14906/14906), 13.95 MiB | 25.70 MiB/s, done.
Resolving deltas: 100% (10237/10237), done.
/content/yolov5
| ██████████ | 184 kB 14.8 MB/s
| ██████████ | 62 kB 794 kB/s
| ██████████ | 1.6 MB 68.7 MB/s
| ██████████ | 45 kB 3.2 MB/s
| ██████████ | 54 kB 3.6 MB/s
| ██████████ | 178 kB 69.3 MB/s
| ██████████ | 67 kB 7.0 MB/s
| ██████████ | 138 kB 50.9 MB/s
| ██████████ | 62 kB 1.9 MB/s
Building wheel for wget (setup.py) ... done
Setup complete. Using torch 1.13.0+cu116 (Tesla T4)
```

Resultado de segunda línea de código de Roboflow conectada con YOLO:

```
loading Roboflow workspace...
loading Roboflow project...
Downloading Dataset Version Zip in Detección-FOD-2 to yolov5pytorch:
100% [122012194 / 122012194] bytes
Extracting Dataset Version Zip to Detección-FOD-2 in yolov5pytorch::
100%|██████████| 4484/4484 [00:01<00:00, 2443.78it/s]
```

Resultado del entrenamiento del modelo de detección:

```
train: weights=yolov5x.pt, cfg=, data=/content/yolov5/Detección-FOD-2/data.yaml, hyp=data/hyps/hyp.scratch-low.yaml, epochs=150, batch_size=16, imgsz=416, rect=False, resume=False, nosave=False, noval=False, noautoanchor=False, noplots=False, evolve=None, bucket=, cache=ram, image_weights=False, device=, multi_scale=False,
```

```
single_cls=False, optimizer=SGD, sync_bn=False, workers=8,
project=runs/train, name=exp, exist_ok=False, quad=False,
cos_lr=False, label_smoothing=0.0, patience=100, freeze=[0],
save_period=-1, seed=0, local_rank=-1, entity=None,
upload_dataset=False, bbox_interval=-1, artifact_alias=latest
```

github: up to date with <https://github.com/ultralytics/yolov5> ✓

YOLOv5 🚀 v7.0-53-g65071da Python-3.8.16 torch-1.13.0+cu116 CUDA:0
(Tesla T4, 15110MiB)

hyperparameters: lr0=0.01, lrf=0.01, momentum=0.937,
weight_decay=0.0005, warmup_epochs=3.0, warmup_momentum=0.8,
warmup_bias_lr=0.1, box=0.05, cls=0.5, cls_pw=1.0, obj=1.0,
obj_pw=1.0, iou_t=0.2, anchor_t=4.0, fl_gamma=0.0, hsv_h=0.015,
hsv_s=0.7, hsv_v=0.4, degrees=0.0, translate=0.1, scale=0.5,
shear=0.0, perspective=0.0, flipud=0.0, fliplr=0.5, mosaic=1.0,
mixup=0.0, copy_paste=0.0

ClearML: run 'pip install clearml' to automatically track, visualize
and remotely train YOLOv5 🚀 in ClearML

Comet: run 'pip install comet_ml' to automatically track and
visualize YOLOv5 🚀 runs in Comet

TensorBoard: Start with 'tensorboard --logdir runs/train', view at
<http://localhost:6006/>

Downloading <https://ultralytics.com/assets/Arial.ttf> to

/root/.config/Ultralytics/Arial.ttf...

100% 755k/755k [00:00<00:00, 53.7MB/s]

Downloading

<https://github.com/ultralytics/yolov5/releases/download/v7.0/yolov5x.pt>
to yolov5x.pt...

100% 166M/166M [00:04<00:00, 36.3MB/s]

Overriding model.yaml nc=80 with nc=11

arguments	from	n	params	module
0	-1	1	8800	models.common.Conv
[3, 80, 6, 2, 2]	-1	1	115520	models.common.Conv
1	-1	1	115520	models.common.Conv
[80, 160, 3, 2]	-1	4	309120	models.common.C3
2	-1	4	309120	models.common.C3
[160, 160, 4]	-1	1	461440	models.common.Conv
3	-1	1	461440	models.common.Conv
[160, 320, 3, 2]	-1	8	2259200	models.common.C3
4	-1	8	2259200	models.common.C3
[320, 320, 8]	-1	1	1844480	models.common.Conv
5	-1	1	1844480	models.common.Conv
[320, 640, 3, 2]	-1	12	13125120	models.common.C3
6	-1	12	13125120	models.common.C3
[640, 640, 12]	-1	1	7375360	models.common.Conv
7	-1	1	7375360	models.common.Conv
[640, 1280, 3, 2]	-1	4	19676160	models.common.C3
8	-1	4	19676160	models.common.C3
[1280, 1280, 4]	-1	4	19676160	models.common.C3

```

 9          -1  1  4099840  models.common.SPPF
[1280, 1280, 5]
10          -1  1    820480  models.common.Conv
[1280, 640, 1, 1]
11          -1  1          0
torch.nn.modules.upsampling.Upsample  [None, 2, 'nearest']
12          [-1, 6]  1          0  models.common.Concat
[1]
13          -1  4  5332480  models.common.C3
[1280, 640, 4, False]
14          -1  1    205440  models.common.Conv
[640, 320, 1, 1]
15          -1  1          0
torch.nn.modules.upsampling.Upsample  [None, 2, 'nearest']
16          [-1, 4]  1          0  models.common.Concat
[1]
17          -1  4  1335040  models.common.C3
[640, 320, 4, False]
18          -1  1    922240  models.common.Conv
[320, 320, 3, 2]
19          [-1, 14]  1          0  models.common.Concat
[1]
20          -1  4  4922880  models.common.C3
[640, 640, 4, False]
21          -1  1  3687680  models.common.Conv
[640, 640, 3, 2]
22          [-1, 10]  1          0  models.common.Concat
[1]
23          -1  4  19676160  models.common.C3
[1280, 1280, 4, False]
24          [17, 20, 23]  1    107664  models.yolo.Detect
[11, [[10, 13, 16, 30, 33, 23], [30, 61, 62, 45, 59, 119], [116, 90,
156, 198, 373, 326]], [320, 640, 1280]]
Model summary: 445 layers, 86285104 parameters, 86285104 gradients,
204.8 GFLOPs

```

Transferred 739/745 items from yolov5x.pt

AMP: checks passed

optimizer: SGD(lr=0.01) with parameter groups 123 weight(decay=0.0), 126 weight(decay=0.0005), 126 bias

augmentations: Blur(p=0.01, blur_limit=(3, 7)), MedianBlur(p=0.01, blur_limit=(3, 7)), ToGray(p=0.01), CLAHE(p=0.01, clip_limit=(1, 4.0), tile_grid_size=(8, 8))

train: Scanning /content/yolov5/Detección-FOD-2/train/labels... 1682 images, 2 backgrounds, 0 corrupt: 100% 1682/1682 [00:00<00:00, 2080.16it/s]

train: New cache created: /content/yolov5/Detección-FOD-2/train/labels.cache

train: Caching images (0.8GB ram): 100% 1682/1682 [00:08<00:00, 189.49it/s]

val: Scanning /content/yolov5/Detección-FOD-2/valid/labels... 402 images, 1 backgrounds, 0 corrupt: 100% 402/402 [00:00<00:00, 834.97it/s]

val: New cache created: /content/yolov5/Detección-FOD-2/valid/labels.cache
val: Caching images (0.2GB ram): 100% 402/402 [00:03<00:00, 110.87it/s]

AutoAnchor: 5.06 anchors/target, 0.998 Best Possible Recall (BPR).
 Current anchors are a good fit to dataset
 Plotting labels to runs/train/exp/labels.jpg...
 Image sizes 416 train, 416 val
 Using 2 dataloader workers
 Logging results to **runs/train/exp**
 Starting training for 150 epochs...

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
0/149	6.9G	0.08775	0.0399	0.05129	3
416:	100% 106/106	[00:57<00:00, 1.83it/s]			
	Class	Images	Instances	P	R
mAP50	mAP50-95: 100%	13/13	[00:07<00:00, 1.78it/s]		
0.22	all	402	1482	0.679	0.229
0.0852					

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
1/149	7.81G	0.06666	0.03395	0.02401	19
416:	100% 106/106	[00:52<00:00, 2.02it/s]			
	Class	Images	Instances	P	R
mAP50	mAP50-95: 100%	13/13	[00:06<00:00, 2.09it/s]		
0.261	all	402	1482	0.522	0.332
0.118					

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
2/149	7.81G	0.06393	0.03117	0.01619	8
416:	100% 106/106	[00:52<00:00, 2.01it/s]			
	Class	Images	Instances	P	R
mAP50	mAP50-95: 100%	13/13	[00:06<00:00, 2.12it/s]		
0.315	all	402	1482	0.598	0.347
0.137					

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
3/149	7.81G	0.05779	0.03062	0.0139	5
416:	100% 106/106	[00:52<00:00, 2.01it/s]			
	Class	Images	Instances	P	R
mAP50	mAP50-95: 100%	13/13	[00:06<00:00, 2.12it/s]		
0.367	all	402	1482	0.624	0.415
0.166					

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
4/149	7.81G	0.05205	0.03118	0.01159	27
416:	100% 106/106	[00:53<00:00, 1.99it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:06<00:00,	2.15it/s]	
		all	402	1482	0.694	0.455
0.42	0.175					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	5/149	7.81G	0.04951	0.03028	0.009989	10
416:	100%	106/106	[00:53<00:00,	1.99it/s]		
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00,	2.19it/s]	
		all	402	1482	0.63	0.414
0.377	0.16					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	6/149	7.81G	0.04778	0.03004	0.01066	7
416:	100%	106/106	[00:52<00:00,	2.00it/s]		
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00,	2.20it/s]	
		all	402	1482	0.674	0.428
0.433	0.202					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	7/149	7.81G	0.04668	0.03035	0.008784	13
416:	100%	106/106	[00:53<00:00,	1.98it/s]		
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00,	2.20it/s]	
		all	402	1482	0.641	0.431
0.426	0.183					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	8/149	7.81G	0.04524	0.03006	0.007988	2
416:	100%	106/106	[00:53<00:00,	1.99it/s]		
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00,	2.18it/s]	
		all	402	1482	0.612	0.46
0.445	0.208					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	9/149	7.81G	0.04464	0.03012	0.008669	16
416:	100%	106/106	[00:53<00:00,	1.98it/s]		
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00,	2.17it/s]	
		all	402	1482	0.677	0.438
0.465	0.225					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	10/149	7.81G	0.04403	0.02917	0.007408	6
416:	100%	106/106	[00:53<00:00,	1.99it/s]		

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.21it/s]		
		all	402	1482	0.697	0.466
0.487	0.241					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	11/149	7.81G	0.04322	0.02994	0.007194	5

416:	100%	106/106	[00:53<00:00, 1.99it/s]			
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.21it/s]		
		all	402	1482	0.686	0.447
0.474	0.219					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	12/149	7.81G	0.04297	0.02985	0.006917	10

416:	100%	106/106	[00:52<00:00, 2.00it/s]			
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.19it/s]		
		all	402	1482	0.68	0.475
0.476	0.245					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	13/149	7.81G	0.04266	0.02847	0.006359	15

416:	100%	106/106	[00:53<00:00, 1.97it/s]			
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.21it/s]		
		all	402	1482	0.685	0.469
0.497	0.243					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	14/149	7.81G	0.04193	0.02868	0.006655	14

416:	100%	106/106	[00:53<00:00, 2.00it/s]			
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.21it/s]		
		all	402	1482	0.731	0.432
0.482	0.248					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	15/149	7.81G	0.04115	0.02898	0.005979	9

416:	100%	106/106	[00:53<00:00, 1.99it/s]			
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.21it/s]		
		all	402	1482	0.672	0.475
0.487	0.255					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	16/149	7.81G	0.04065	0.0285	0.005466	5

416:	100%	106/106	[00:53<00:00, 1.99it/s]			
------	------	---------	-------------------------	--	--	--

	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.20it/s]			
	all	402	1482	0.721	0.481	
0.518	0.271					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	17/149	7.81G	0.04022	0.02829	0.0057	6
416:	100%	106/106	[00:53<00:00, 1.99it/s]			
	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.21it/s]			
	all	402	1482	0.707	0.502	
0.514	0.264					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	18/149	7.81G	0.03991	0.02779	0.00518	4
416:	100%	106/106	[00:53<00:00, 2.00it/s]			
	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.19it/s]			
	all	402	1482	0.738	0.451	
0.504	0.26					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	19/149	7.81G	0.03932	0.02718	0.004539	17
416:	100%	106/106	[00:53<00:00, 1.98it/s]			
	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.20it/s]			
	all	402	1482	0.719	0.496	
0.513	0.268					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	20/149	7.81G	0.03898	0.02702	0.004986	15
416:	100%	106/106	[00:53<00:00, 2.00it/s]			
	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.22it/s]			
	all	402	1482	0.757	0.462	
0.525	0.272					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	21/149	7.81G	0.03846	0.0273	0.004819	34
416:	100%	106/106	[00:53<00:00, 1.99it/s]			
	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.23it/s]			
	all	402	1482	0.7	0.462	
0.5	0.251					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	22/149	7.81G	0.03749	0.0262	0.004986	1
416:	100%	106/106	[00:53<00:00, 2.00it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.22it/s]		
		all	402	1482	0.774	0.494
0.554	0.296					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	23/149	7.81G	0.03846	0.02782	0.004877	11
416:	100%	106/106	[00:53<00:00, 1.98it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.21it/s]		
		all	402	1482	0.746	0.425
0.488	0.255					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	24/149	7.81G	0.03769	0.02668	0.004333	13
416:	100%	106/106	[00:53<00:00, 1.99it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.20it/s]		
		all	402	1482	0.741	0.494
0.551	0.287					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	25/149	7.81G	0.03681	0.02587	0.004506	6
416:	100%	106/106	[00:53<00:00, 1.99it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:06<00:00, 2.14it/s]		
		all	402	1482	0.726	0.449
0.502	0.258					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	26/149	7.81G	0.03733	0.02661	0.004277	3
416:	100%	106/106	[00:53<00:00, 1.98it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.22it/s]		
		all	402	1482	0.733	0.461
0.49	0.26					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	27/149	7.81G	0.03713	0.02564	0.004021	4
416:	100%	106/106	[00:53<00:00, 2.00it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:06<00:00, 2.12it/s]		
		all	402	1482	0.764	0.465
0.521	0.293					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	28/149	7.81G	0.03724	0.02569	0.004119	17
416:	100%	106/106	[00:53<00:00, 2.00it/s]			

	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.23it/s]			
	all	402	1482	0.777	0.487	
0.549	0.298					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	29/149	7.81G	0.03618	0.02527	0.00347	3
416:	100%	106/106	[00:53<00:00, 1.99it/s]			
	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:06<00:00, 2.14it/s]			
	all	402	1482	0.789	0.488	
0.545	0.287					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	30/149	7.81G	0.03593	0.02571	0.00374	22
416:	100%	106/106	[00:53<00:00, 2.00it/s]			
	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.24it/s]			
	all	402	1482	0.794	0.471	
0.539	0.284					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	31/149	7.81G	0.03572	0.02434	0.00359	4
416:	100%	106/106	[00:53<00:00, 2.00it/s]			
	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:06<00:00, 2.13it/s]			
	all	402	1482	0.809	0.452	
0.518	0.271					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	32/149	7.81G	0.03533	0.02501	0.003551	18
416:	100%	106/106	[00:53<00:00, 2.00it/s]			
	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.24it/s]			
	all	402	1482	0.775	0.498	
0.543	0.295					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	33/149	7.81G	0.03499	0.02525	0.003235	6
416:	100%	106/106	[00:53<00:00, 1.99it/s]			
	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:06<00:00, 2.10it/s]			
	all	402	1482	0.765	0.5	
0.541	0.296					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	34/149	7.81G	0.03475	0.02535	0.003493	4
416:	100%	106/106	[00:53<00:00, 1.99it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.22it/s]		
		all	402	1482	0.807	0.475
0.541	0.296					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	35/149	7.81G	0.03465	0.02487	0.003097	23

416:	100%	106/106	[00:53<00:00, 2.00it/s]			
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:06<00:00, 2.10it/s]		
		all	402	1482	0.802	0.474
0.546	0.294					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	36/149	7.81G	0.03395	0.02446	0.003526	6

416:	100%	106/106	[00:53<00:00, 2.00it/s]			
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.25it/s]		
		all	402	1482	0.792	0.459
0.544	0.301					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	37/149	7.81G	0.03393	0.02483	0.00325	14

416:	100%	106/106	[00:53<00:00, 1.98it/s]			
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.21it/s]		
		all	402	1482	0.8	0.463
0.536	0.287					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	38/149	7.81G	0.03356	0.0241	0.003336	16

416:	100%	106/106	[00:53<00:00, 2.00it/s]			
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.24it/s]		
		all	402	1482	0.798	0.477
0.545	0.303					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	39/149	7.81G	0.03326	0.0239	0.003369	7

416:	100%	106/106	[00:53<00:00, 1.99it/s]			
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.23it/s]		
		all	402	1482	0.798	0.49
0.56	0.304					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	40/149	7.81G	0.03273	0.02303	0.003248	7

416:	100%	106/106	[00:53<00:00, 2.00it/s]			
------	------	---------	-------------------------	--	--	--

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00,	2.23it/s]	
		all	402	1482	0.777	0.493
0.539	0.302					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	41/149	7.81G	0.0325	0.02319	0.002847	4
416:	100%	106/106	[00:53<00:00,	1.99it/s]		
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00,	2.22it/s]	
		all	402	1482	0.793	0.471
0.53	0.291					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	42/149	7.81G	0.03281	0.02305	0.003136	8
416:	100%	106/106	[00:53<00:00,	2.00it/s]		
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00,	2.24it/s]	
		all	402	1482	0.759	0.489
0.547	0.293					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	43/149	7.81G	0.03251	0.02318	0.002932	11
416:	100%	106/106	[00:53<00:00,	1.98it/s]		
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00,	2.25it/s]	
		all	402	1482	0.8	0.464
0.547	0.31					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	44/149	7.81G	0.03267	0.02196	0.002832	6
416:	100%	106/106	[00:53<00:00,	2.00it/s]		
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00,	2.24it/s]	
		all	402	1482	0.771	0.505
0.563	0.319					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	45/149	7.81G	0.03249	0.02249	0.002626	7
416:	100%	106/106	[00:53<00:00,	1.99it/s]		
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00,	2.24it/s]	
		all	402	1482	0.758	0.495
0.521	0.276					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	46/149	7.81G	0.03171	0.02241	0.002865	9
416:	100%	106/106	[00:53<00:00,	2.00it/s]		

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.24it/s]		
		all	402	1482	0.799	0.474
0.545	0.299					

Size	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
47/149	47/149	7.81G	0.03166	0.02218	0.002496	7
416:	100%	106/106	[00:53<00:00,	1.98it/s]		

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.26it/s]		
		all	402	1482	0.803	0.479
0.544	0.305					

Size	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
48/149	48/149	7.81G	0.03139	0.02203	0.002667	2
416:	100%	106/106	[00:53<00:00,	2.00it/s]		

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.24it/s]		
		all	402	1482	0.799	0.478
0.527	0.294					

Size	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
49/149	49/149	7.81G	0.03107	0.0221	0.002477	8
416:	100%	106/106	[00:53<00:00,	1.99it/s]		

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.24it/s]		
		all	402	1482	0.805	0.491
0.549	0.305					

Size	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
50/149	50/149	7.81G	0.03068	0.02151	0.002775	9
416:	100%	106/106	[00:53<00:00,	2.00it/s]		

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.26it/s]		
		all	402	1482	0.795	0.477
0.534	0.296					

Size	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
51/149	51/149	7.81G	0.03048	0.02241	0.002684	9
416:	100%	106/106	[00:53<00:00,	1.98it/s]		

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.24it/s]		
		all	402	1482	0.798	0.481
0.533	0.297					

Size	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
52/149	52/149	7.81G	0.03056	0.02265	0.002533	8
416:	100%	106/106	[00:53<00:00,	2.00it/s]		

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00,	2.24it/s]	
		all	402	1482	0.812	0.492
0.555	0.305					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	53/149	7.81G	0.03053	0.02153	0.002525	6
416:	100%	106/106	[00:53<00:00,	1.99it/s]		
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00,	2.26it/s]	
		all	402	1482	0.808	0.496
0.56	0.323					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	54/149	7.81G	0.02977	0.02143	0.002467	8
416:	100%	106/106	[00:53<00:00,	1.99it/s]		
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00,	2.25it/s]	
		all	402	1482	0.777	0.49
0.564	0.316					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	55/149	7.81G	0.02979	0.02114	0.002192	11
416:	100%	106/106	[00:53<00:00,	1.99it/s]		
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00,	2.25it/s]	
		all	402	1482	0.797	0.462
0.549	0.309					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	56/149	7.81G	0.02955	0.02109	0.002214	15
416:	100%	106/106	[00:53<00:00,	2.00it/s]		
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00,	2.25it/s]	
		all	402	1482	0.798	0.495
0.544	0.313					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	57/149	7.81G	0.02945	0.02065	0.002721	15
416:	100%	106/106	[00:53<00:00,	1.99it/s]		
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00,	2.25it/s]	
		all	402	1482	0.801	0.476
0.536	0.304					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	58/149	7.81G	0.02907	0.02046	0.002437	28
416:	100%	106/106	[00:53<00:00,	1.99it/s]		

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.25it/s]		
		all	402	1482	0.79	0.512
0.562	0.314					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	59/149	7.81G	0.02879	0.02062	0.002357	6
416:	100%	106/106	[00:53<00:00, 1.99it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.24it/s]		
		all	402	1482	0.794	0.488
0.556	0.306					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	60/149	7.81G	0.02896	0.02065	0.002141	7
416:	100%	106/106	[00:53<00:00, 2.00it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.23it/s]		
		all	402	1482	0.805	0.489
0.568	0.328					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	61/149	7.81G	0.02803	0.01977	0.001858	7
416:	100%	106/106	[00:53<00:00, 1.98it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.25it/s]		
		all	402	1482	0.793	0.474
0.556	0.313					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	62/149	7.81G	0.02802	0.02036	0.002071	12
416:	100%	106/106	[00:52<00:00, 2.00it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.26it/s]		
		all	402	1482	0.776	0.501
0.587	0.341					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	63/149	7.81G	0.02775	0.02053	0.002029	21
416:	100%	106/106	[00:53<00:00, 1.98it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.26it/s]		
		all	402	1482	0.786	0.489
0.569	0.329					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	64/149	7.81G	0.02826	0.01984	0.002595	10
416:	100%	106/106	[00:53<00:00, 2.00it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.17it/s]		
		all	402	1482	0.803	0.494
0.582	0.337					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	65/149	7.81G	0.02747	0.02075	0.001818	15
416:	100%	106/106	[00:53<00:00, 1.99it/s]			
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.26it/s]		
		all	402	1482	0.784	0.49
0.573	0.331					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	66/149	7.81G	0.02778	0.02054	0.002115	7
416:	100%	106/106	[00:53<00:00, 1.98it/s]			
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.26it/s]		
		all	402	1482	0.806	0.495
0.604	0.342					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	67/149	7.81G	0.02732	0.01943	0.002253	13
416:	100%	106/106	[00:53<00:00, 1.99it/s]			
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.25it/s]		
		all	402	1482	0.835	0.475
0.561	0.322					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	68/149	7.81G	0.02713	0.01974	0.002001	4
416:	100%	106/106	[00:53<00:00, 1.99it/s]			
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.26it/s]		
		all	402	1482	0.834	0.48
0.558	0.317					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	69/149	7.81G	0.02666	0.02095	0.002386	47
416:	100%	106/106	[00:53<00:00, 2.00it/s]			
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.25it/s]		
		all	402	1482	0.783	0.509
0.623	0.377					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	70/149	7.81G	0.02716	0.01922	0.001952	1
416:	100%	106/106	[00:53<00:00, 1.99it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.25it/s]		
		all	402	1482	0.76	0.506
0.614	0.359					

Size	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
	71/149	7.81G	0.02672	0.01899	0.001677	12
416:	100%	106/106	[00:52<00:00, 2.00it/s]			
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.24it/s]		
		all	402	1482	0.821	0.476
0.613	0.356					

Size	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
	72/149	7.81G	0.02667	0.019	0.002071	5
416:	100%	106/106	[00:53<00:00, 1.99it/s]			
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.23it/s]		
		all	402	1482	0.811	0.477
0.572	0.328					

Size	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
	73/149	7.81G	0.02669	0.01902	0.001926	8
416:	100%	106/106	[00:53<00:00, 1.99it/s]			
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.24it/s]		
		all	402	1482	0.822	0.483
0.596	0.344					

Size	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
	74/149	7.81G	0.02594	0.01882	0.002163	5
416:	100%	106/106	[00:53<00:00, 2.00it/s]			
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.25it/s]		
		all	402	1482	0.801	0.491
0.597	0.339					

Size	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
	75/149	7.81G	0.02589	0.01839	0.001594	4
416:	100%	106/106	[00:53<00:00, 1.99it/s]			
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.27it/s]		
		all	402	1482	0.8	0.485
0.565	0.324					

Size	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
	76/149	7.81G	0.02597	0.0184	0.001776	22
416:	100%	106/106	[00:52<00:00, 2.00it/s]			

	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.27it/s]			
	all	402	1482	0.757	0.494	
0.558	0.314					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	77/149	7.81G	0.02579	0.01855	0.001701	16
416:	100%	106/106	[00:53<00:00, 1.99it/s]			
	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.27it/s]			
	all	402	1482	0.803	0.499	
0.585	0.334					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	78/149	7.81G	0.0259	0.01887	0.001561	12
416:	100%	106/106	[00:53<00:00, 2.00it/s]			
	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.27it/s]			
	all	402	1482	0.799	0.521	
0.614	0.35					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	79/149	7.81G	0.02577	0.01916	0.0015	6
416:	100%	106/106	[00:53<00:00, 1.99it/s]			
	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.28it/s]			
	all	402	1482	0.772	0.517	
0.58	0.335					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	80/149	7.81G	0.0246	0.01758	0.001684	8
416:	100%	106/106	[00:52<00:00, 2.00it/s]			
	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.26it/s]			
	all	402	1482	0.834	0.466	
0.604	0.348					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	81/149	7.81G	0.02514	0.0183	0.001574	16
416:	100%	106/106	[00:53<00:00, 1.99it/s]			
	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.28it/s]			
	all	402	1482	0.809	0.468	
0.588	0.335					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	82/149	7.81G	0.02488	0.01811	0.001678	13
416:	100%	106/106	[00:52<00:00, 2.00it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:06<00:00, 2.16it/s]		
		all	402	1482	0.778	0.499
0.578	0.327					

Size	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
	83/149	7.81G	0.02458	0.01747	0.001486	4
416:	100%	106/106	[00:52<00:00, 2.00it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.27it/s]		
		all	402	1482	0.798	0.502
0.598	0.35					

Size	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
	84/149	7.81G	0.02495	0.01815	0.001761	10
416:	100%	106/106	[00:53<00:00, 1.99it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.27it/s]		
		all	402	1482	0.775	0.496
0.596	0.35					

Size	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
	85/149	7.81G	0.02443	0.01758	0.001515	6
416:	100%	106/106	[00:53<00:00, 2.00it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.27it/s]		
		all	402	1482	0.818	0.473
0.578	0.327					

Size	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
	86/149	7.81G	0.0241	0.01802	0.001476	18
416:	100%	106/106	[00:53<00:00, 1.99it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.27it/s]		
		all	402	1482	0.801	0.498
0.623	0.366					

Size	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
	87/149	7.81G	0.02438	0.01824	0.001519	9
416:	100%	106/106	[00:52<00:00, 2.00it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.26it/s]		
		all	402	1482	0.832	0.482
0.611	0.352					

Size	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
	88/149	7.81G	0.02386	0.01761	0.001654	9
416:	100%	106/106	[00:53<00:00, 1.99it/s]			

	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.27it/s]			
	all	402	1482	0.798	0.498	
0.579	0.335					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	89/149	7.81G	0.02361	0.01683	0.001536	18
416:	100%	106/106	[00:52<00:00, 2.00it/s]			
	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.26it/s]			
	all	402	1482	0.778	0.495	
0.577	0.328					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	90/149	7.81G	0.02394	0.0178	0.001454	16
416:	100%	106/106	[00:53<00:00, 1.99it/s]			
	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.28it/s]			
	all	402	1482	0.787	0.505	
0.593	0.349					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	91/149	7.81G	0.02351	0.01758	0.001536	26
416:	100%	106/106	[00:53<00:00, 1.98it/s]			
	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.26it/s]			
	all	402	1482	0.808	0.487	
0.57	0.328					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	92/149	7.81G	0.02319	0.01694	0.001562	5
416:	100%	106/106	[00:52<00:00, 2.00it/s]			
	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.27it/s]			
	all	402	1482	0.82	0.487	
0.619	0.353					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	93/149	7.81G	0.02329	0.01674	0.001511	2
416:	100%	106/106	[00:53<00:00, 1.99it/s]			
	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.28it/s]			
	all	402	1482	0.801	0.484	
0.59	0.347					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	94/149	7.81G	0.02325	0.01758	0.001349	9
416:	100%	106/106	[00:53<00:00, 1.99it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.28it/s]		
		all	402	1482	0.768	0.519
0.578	0.326					

Size	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
	95/149	7.81G	0.02319	0.01682	0.001338	12
416:	100%	106/106	[00:53<00:00, 1.99it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.27it/s]		
		all	402	1482	0.629	0.569
0.563	0.318					

Size	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
	96/149	7.81G	0.0228	0.01654	0.001396	18
416:	100%	106/106	[00:52<00:00, 2.00it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.28it/s]		
		all	402	1482	0.646	0.543
0.606	0.347					

Size	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
	97/149	7.81G	0.02272	0.01682	0.001481	7
416:	100%	106/106	[00:53<00:00, 1.99it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.30it/s]		
		all	402	1482	0.668	0.564
0.571	0.323					

Size	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
	98/149	7.81G	0.02244	0.01684	0.001304	6
416:	100%	106/106	[00:53<00:00, 2.00it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:06<00:00, 2.15it/s]		
		all	402	1482	0.842	0.459
0.562	0.317					

Size	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
	99/149	7.81G	0.02248	0.01702	0.001385	11
416:	100%	106/106	[00:53<00:00, 2.00it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.29it/s]		
		all	402	1482	0.78	0.497
0.572	0.334					

Size	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
	100/149	7.81G	0.02241	0.01649	0.001369	3
416:	100%	106/106	[00:53<00:00, 1.99it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.28it/s]		
		all	402	1482	0.683	0.565
0.604	0.345					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	101/149	7.81G	0.0219	0.01598	0.001114	6
416:	100%	106/106	[00:53<00:00, 1.99it/s]			
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.28it/s]		
		all	402	1482	0.82	0.471
0.593	0.339					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	102/149	7.81G	0.02207	0.01645	0.001223	3
416:	100%	106/106	[00:53<00:00, 1.99it/s]			
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.29it/s]		
		all	402	1482	0.656	0.546
0.583	0.331					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	103/149	7.81G	0.02167	0.01631	0.001287	26
416:	100%	106/106	[00:53<00:00, 2.00it/s]			
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.28it/s]		
		all	402	1482	0.67	0.526
0.598	0.341					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	104/149	7.81G	0.02172	0.01612	0.001412	11
416:	100%	106/106	[00:53<00:00, 1.99it/s]			
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.28it/s]		
		all	402	1482	0.82	0.483
0.618	0.354					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	105/149	7.81G	0.02143	0.01572	0.001131	19
416:	100%	106/106	[00:52<00:00, 2.00it/s]			
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.17it/s]		
		all	402	1482	0.668	0.517
0.551	0.316					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	106/149	7.81G	0.02128	0.01585	0.001196	7
416:	100%	106/106	[00:52<00:00, 2.00it/s]			

10: **Detección de FOD por medio de inteligencia artificial con inspecciones por UAV**

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.27it/s]		
		all	402	1482	0.833	0.483
0.6	0.347					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	107/149	7.81G	0.02108	0.01556	0.001251	9

416:	100%	106/106	[00:53<00:00, 1.99it/s]			
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.29it/s]		
		all	402	1482	0.671	0.544
0.6	0.355					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	108/149	7.81G	0.0214	0.01595	0.001142	16

416:	100%	106/106	[00:53<00:00, 2.00it/s]			
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.28it/s]		
		all	402	1482	0.654	0.559
0.593	0.339					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	109/149	7.81G	0.02082	0.01567	0.0009747	18

416:	100%	106/106	[00:53<00:00, 1.99it/s]			
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.28it/s]		
		all	402	1482	0.676	0.526
0.589	0.338					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	110/149	7.81G	0.0207	0.01548	0.001078	9

416:	100%	106/106	[00:52<00:00, 2.00it/s]			
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.28it/s]		
		all	402	1482	0.812	0.478
0.606	0.358					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	111/149	7.81G	0.02034	0.01547	0.001092	9

416:	100%	106/106	[00:53<00:00, 1.99it/s]			
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.28it/s]		
		all	402	1482	0.684	0.557
0.595	0.349					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	112/149	7.81G	0.02073	0.01581	0.00127	4

416:	100%	106/106	[00:53<00:00, 1.99it/s]			
------	------	---------	-------------------------	--	--	--

	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.29it/s]			
	all	402	1482	0.779	0.492	
0.595	0.352					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	113/149	7.81G	0.02013	0.0149	0.001214	9
416:	100%	106/106	[00:53<00:00, 2.00it/s]			
	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.29it/s]			
	all	402	1482	0.695	0.528	
0.562	0.326					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	114/149	7.81G	0.02027	0.01516	0.001087	13
416:	100%	106/106	[00:53<00:00, 1.99it/s]			
	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.27it/s]			
	all	402	1482	0.67	0.538	
0.551	0.319					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	115/149	7.81G	0.02038	0.01515	0.001151	6
416:	100%	106/106	[00:52<00:00, 2.00it/s]			
	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.28it/s]			
	all	402	1482	0.693	0.524	
0.546	0.321					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	116/149	7.81G	0.01965	0.01492	0.0008824	3
416:	100%	106/106	[00:53<00:00, 1.99it/s]			
	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.28it/s]			
	all	402	1482	0.712	0.519	
0.554	0.322					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	117/149	7.81G	0.0199	0.01456	0.0009682	7
416:	100%	106/106	[00:53<00:00, 2.00it/s]			
	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.29it/s]			
	all	402	1482	0.657	0.54	
0.548	0.321					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	118/149	7.81G	0.01937	0.01451	0.001184	18
416:	100%	106/106	[00:52<00:00, 2.00it/s]			

10. Detección de FOD por medio de inteligencia artificial con inspecciones por UAV

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.29it/s]		
		all	402	1482	0.696	0.518
0.543	0.316					

Size	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
	119/149	7.81G	0.01956	0.01496	0.001035	17

		Class	Images	Instances	P	R
416:	100%	106/106	[00:53<00:00, 1.99it/s]			
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.30it/s]		
		all	402	1482	0.675	0.514
0.563	0.33					

Size	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
	120/149	7.81G	0.01941	0.01478	0.0009348	8

		Class	Images	Instances	P	R
416:	100%	106/106	[00:52<00:00, 2.00it/s]			
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.28it/s]		
		all	402	1482	0.666	0.533
0.605	0.369					

Size	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
	121/149	7.81G	0.01921	0.01442	0.0009547	10

		Class	Images	Instances	P	R
416:	100%	106/106	[00:53<00:00, 1.99it/s]			
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.28it/s]		
		all	402	1482	0.654	0.557
0.614	0.373					

Size	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
	122/149	7.81G	0.01929	0.01434	0.0009074	4

		Class	Images	Instances	P	R
416:	100%	106/106	[00:53<00:00, 1.99it/s]			
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.28it/s]		
		all	402	1482	0.703	0.502
0.587	0.356					

Size	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
	123/149	7.81G	0.01885	0.01466	0.0009654	8

		Class	Images	Instances	P	R
416:	100%	106/106	[00:53<00:00, 1.99it/s]			
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.27it/s]		
		all	402	1482	0.69	0.52
0.55	0.321					

Size	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
	124/149	7.81G	0.01876	0.01424	0.000907	4

416:	100%	106/106	[00:53<00:00, 1.99it/s]			
------	------	---------	-------------------------	--	--	--

	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.27it/s]			
	all	402	1482	0.684	0.532	
0.58	0.346					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	125/149	7.81G	0.01865	0.01438	0.0008244	16
416:	100%	106/106	[00:53<00:00, 2.00it/s]			
	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.18it/s]			
	all	402	1482	0.692	0.532	
0.562	0.324					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	126/149	7.81G	0.01853	0.0139	0.0009038	6
416:	100%	106/106	[00:53<00:00, 2.00it/s]			
	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.27it/s]			
	all	402	1482	0.687	0.541	
0.592	0.354					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	127/149	7.81G	0.01841	0.01384	0.0009691	6
416:	100%	106/106	[00:53<00:00, 1.98it/s]			
	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.26it/s]			
	all	402	1482	0.706	0.51	
0.588	0.347					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	128/149	7.81G	0.01815	0.01381	0.0008237	9
416:	100%	106/106	[00:53<00:00, 2.00it/s]			
	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.25it/s]			
	all	402	1482	0.663	0.533	
0.598	0.356					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	129/149	7.81G	0.01849	0.01396	0.0009516	18
416:	100%	106/106	[00:53<00:00, 1.99it/s]			
	Class	Images	Instances	P	R	
mAP50	mAP50-95: 100%	13/13	[00:05<00:00, 2.28it/s]			
	all	402	1482	0.691	0.55	
0.595	0.357					
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	
Size	130/149	7.81G	0.01816	0.01351	0.0008744	12
416:	100%	106/106	[00:53<00:00, 1.99it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.26it/s]		
		all	402	1482	0.686	0.515
0.594	0.364					

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size					
131/149	7.81G	0.01809	0.01361	0.0007642	3
416:	100%	106/106	[00:53<00:00, 2.00it/s]		

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.26it/s]		
		all	402	1482	0.699	0.509
0.6	0.369					

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size					
132/149	7.81G	0.01747	0.01335	0.0008232	8
416:	100%	106/106	[00:53<00:00, 1.99it/s]		

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.27it/s]		
		all	402	1482	0.713	0.526
0.57	0.342					

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size					
133/149	7.81G	0.01775	0.01349	0.0009258	12
416:	100%	106/106	[00:53<00:00, 2.00it/s]		

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.29it/s]		
		all	402	1482	0.694	0.522
0.596	0.354					

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size					
134/149	7.81G	0.01695	0.01317	0.000825	9
416:	100%	106/106	[00:53<00:00, 1.99it/s]		

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.28it/s]		
		all	402	1482	0.707	0.522
0.586	0.355					

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size					
135/149	7.81G	0.01758	0.01338	0.0007463	14
416:	100%	106/106	[00:53<00:00, 1.99it/s]		

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.28it/s]		
		all	402	1482	0.695	0.534
0.607	0.368					

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size					
136/149	7.81G	0.01733	0.0135	0.0008187	9
416:	100%	106/106	[00:52<00:00, 2.00it/s]		

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00,	2.28it/s]	
		all	402	1482	0.712	0.509
0.601	0.354					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	137/149	7.81G	0.01748	0.01311	0.0007759	7
416:	100%	106/106	[00:53<00:00,	1.99it/s]		
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00,	2.28it/s]	
		all	402	1482	0.733	0.506
0.584	0.343					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	138/149	7.81G	0.01722	0.01354	0.0007466	8
416:	100%	106/106	[00:52<00:00,	2.00it/s]		
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:06<00:00,	2.15it/s]	
		all	402	1482	0.682	0.526
0.573	0.339					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	139/149	7.81G	0.01702	0.01278	0.0007931	17
416:	100%	106/106	[00:52<00:00,	2.00it/s]		
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00,	2.29it/s]	
		all	402	1482	0.722	0.504
0.573	0.336					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	140/149	7.81G	0.01735	0.01281	0.0007695	11
416:	100%	106/106	[00:53<00:00,	1.99it/s]		
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00,	2.29it/s]	
		all	402	1482	0.698	0.534
0.58	0.338					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	141/149	7.81G	0.01649	0.01214	0.0005526	7
416:	100%	106/106	[00:52<00:00,	2.00it/s]		
		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00,	2.30it/s]	
		all	402	1482	0.724	0.5
0.568	0.334					
	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	142/149	7.81G	0.01712	0.01309	0.0006345	7
416:	100%	106/106	[00:53<00:00,	1.99it/s]		

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.29it/s]		
		all	402	1482	0.721	0.503
0.566	0.326					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	143/149	7.81G	0.01687	0.01254	0.0006187	10
416:	100%	106/106	[00:53<00:00, 1.99it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.28it/s]		
		all	402	1482	0.739	0.497
0.596	0.353					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	144/149	7.81G	0.01667	0.01283	0.0007548	9
416:	100%	106/106	[00:52<00:00, 2.00it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.28it/s]		
		all	402	1482	0.738	0.508
0.599	0.357					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	145/149	7.81G	0.0166	0.01284	0.0006245	10
416:	100%	106/106	[00:53<00:00, 1.99it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.29it/s]		
		all	402	1482	0.71	0.52
0.601	0.356					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	146/149	7.81G	0.01637	0.01233	0.000698	9
416:	100%	106/106	[00:52<00:00, 2.00it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.20it/s]		
		all	402	1482	0.723	0.518
0.599	0.357					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	147/149	7.81G	0.01605	0.01261	0.0006965	27
416:	100%	106/106	[00:53<00:00, 2.00it/s]			

		Class	Images	Instances	P	R
mAP50	mAP50-95:	100%	13/13	[00:05<00:00, 2.29it/s]		
		all	402	1482	0.728	0.507
0.604	0.355					

	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances
Size	148/149	7.81G	0.01627	0.0129	0.001388	2
416:	100%	106/106	[00:53<00:00, 1.99it/s]			

```

          Class      Images  Instances      P          R
mAP50   mAP50-95: 100% 13/13 [00:05<00:00, 2.28it/s]
          all          402      1482      0.729      0.493
0.6      0.361

```

```

      Epoch      GPU_mem  box_loss  obj_loss  cls_loss  Instances
Size
    149/149      7.81G    0.01625  0.01297  0.0007606      19
416: 100% 106/106 [00:53<00:00, 2.00it/s]
          Class      Images  Instances      P          R
mAP50   mAP50-95: 100% 13/13 [00:05<00:00, 2.29it/s]
          all          402      1482      0.744      0.492
0.601    0.36

```

150 epochs completed in 2.593 hours.

Optimizer stripped from runs/train/exp/weights/last.pt, 173.1MB

Optimizer stripped from runs/train/exp/weights/best.pt, 173.1MB

Validating runs/train/exp/weights/best.pt...

Fusing layers...

Model summary: 322 layers, 86240704 parameters, 0 gradients, 204.0 GFLOPs

```

          Class      Images  Instances      P          R
mAP50   mAP50-95: 100% 13/13 [00:07<00:00, 1.66it/s]
          all          402      1482      0.783      0.51
0.622    0.376
          0          402      323      0.577      0.641
0.562    0.274
          1          402      542      0.739      0.746
0.746    0.44
          2          402      53      0.77      0.774
0.784    0.399
          3          402      41      0.594      0.585
0.589    0.341
          4          402      83      0.828      0.755
0.808    0.472
          5          402      2      1          0
0.745    0.608
          Arandelas      402      8      1          0
0.147    0.0647
          airplane      402      34      0.786      0.853
0.922    0.624
          garbage      402      208      0.55      0.423
0.366    0.152
          screw      402      188      0.984      0.319
0.55     0.385

```

Results saved to **runs/train/exp**

Bibliografía

- Abel Nájera Ramos, I., Vázquez Jiménez, R., Rocío Ramos Bernal, D. N., Gloria Rojas Sánchez, I., & Ana Ma Liborio Vicente, I. (2019). *PROPUESTA METODOLÓGICA PARA LA GENERACIÓN DE ORTOFOTOS Y MODELOS DIGITALES DE ELEVACIÓN DE ALTA RESOLUCIÓN A TRAVÉS DE VUELOS CON DRONES NO-FOTOGRAFÉTRICOS*. <https://www.dji.com/mx/mavic>
- ACI World Operational Safety Sub-Committee. (2010). *Airside safety handbook*. (Vol. 4). ACI World.
- Aerial Insights. (2017). *Cómo planificar el vuelo del drone*. <https://www.aerial-insights.co/blog/como-planificar-capturas-de-dron/>
- Aerosweep Pty Ltd. (2020). *THE FOD*BOSS OPERATORS MANUAL*. www.aerosweep.com
- Airplan. (2018). *PL-03-006 Plan operativo aeropuerto Antonio Roldan Betancur*.
- Ana Velázquez Olivera. (2017). Drones al ataque, ya no hay dónde esconderse. *CIENCIORAMA*. http://www.cienciorama.unam.mx/a/pdf/535_cienciorama.pdf
- Asadi, A. (2016). *Drones The Complete Manual -The essential handbook for drone enthusiasts*. www.imagine-publishing.co.uk
- Aviles Correa, C. L., & Garzón Bernal, J. D. (2021). *Implementación de un sistema de detección y evasión de obstáculos de bajo costo para drones de ala fija*. Fundación Universitaria Los Libertadores.
- Barba Guamán, L. R. (2021). *Uso de técnicas deep learning para reconocimiento de objetos en áreas rurales*. UNIVERSIDAD POLITÉCNICA DE MADRID.
- Bureau d'Enquetes et d'Analyses pour la securite de l'Aviation civile. (2002). *Accident on 25 July 2000 at La Patte d'Oie in Gonesse (95) to the Concorde registered F-BTSC operated by Air France*. https://bea.aero/fileadmin/uploads/tx_elydbrapports/f-sc000725a.pdf
- Centeno Franco, A. (2019). *TRABAJO FIN DE GRADO DEEP LEARNING* [Universidad de Sevilla].

<https://idus.us.es/bitstream/handle/11441/90004/Centeno%20Franco%20Alba%20OTFG.pdf?sequence=1&isAllowed=y>

Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2018). DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4), 834–848. <https://doi.org/10.1109/TPAMI.2017.2699184>

CIAT. (2018). *USO SEGURO DE DRONES*.

Cuevas Castañeda, C. C. (2019). *ACONDICIONAMIENTO E IMPLEMENTACIÓN DE ALGORITMO PARA LA DETECCIÓN DE GANADERÍA VACUNA*. UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA –UNAD.

DJI. (2014). *Inspire 1 - DJI*. <https://www.dji.com/inspire-1>

DJI. (2015). *Phantom 3 Professional - DJI*. <https://www.dji.com/phantom-3-pro>

DJI. (2016). *Mavic Pro - DJI*. <https://www.dji.com/mavic>

Download All Images. (2022). *Download All Images | Chrome Extension*. <https://download-all-images.mobilefirst.me/>

DroneDeploy. (2023). *Drone & Plataforma de mapeo UAV | DroneDeploy*. <https://www.dronedeploy.com/es-mx/>

Dumoulin, V., & Visin, F. (2016). *A guide to convolution arithmetic for deep learning*. <http://arxiv.org/abs/1603.07285>

Dunford, R., Su, Q., & Tamang, E. (2014). The Pareto Principle. *The Plymouth Student Scientist*, 07(1), 140–148. <http://hdl.handle.net/10026.1/14054>

EASA. (2019). *Ramp Inspection Manual (RIM)*. <https://www.easa.europa.eu/en/downloads/94197/en>

EASA. (2022a). *Easy Access Rules for Aerodromes*. <http://eur-lex.europa.eu/>,

EASA. (2022b). *Easy Access Rules for Unmanned Aircraft Systems*. <http://eur-lex.europa.eu/>,

EGA Master S.L. (2020, July 24). *Foreign Object Debris*.

FAA. (2004). *14 CFR Part 139 - Certification of Airports*. <https://www.ecfr.gov/current/title-14/chapter-I/subchapter-G/part-139>

FAA. (2016). *14 CFR PART 107 - SMALL UNMANNED AIRCRAFT SYSTEMS*. <https://www.ecfr.gov/current/title-14/chapter-I/subchapter-F/part-107>

FAA. (2022a). *FAADroneZone*. <https://faadronezone-access.faa.gov/#/>

- FAA. (2022b, May 31). *Restricciones del espacio aéreo | Federal Aviation Administration*. https://www.faa.gov/uas/espanol/restricciones_del_espacio_aereo
- FAA, & Airport Engineering Division. (2010). *AC 150/5210-24 Airport Foreign Object Debris (FOD) Management*.
- FAA, & Dermody, J. (2022a). *AC 150/5200-18D Airport Safety Self-Inspection*.
- FAA, & Dermody, J. (2022b). *AC 150/5200-18C Airport Safety Self-Inspection*.
- FAA, E. Herricks, E., Woodworth, E., Lazar III, P., & Patterson, Jr., J. (2012). *DOT/FAA/AR-11/13 Performance Assessment of an Electro-Optical-Based Foreign Object Debris Detection System*.
- FAA, E. Herricks, E., Woodworth, E., Majumdar, S., & Patterson, Jr., J. (2011). *DOT/FAA/AR-10/33 Performance Assessment of a Radar-Based Foreign Object Debris Detection System*.
- FAA, E. Herricks, E., Woodworth, E., & Patterson, Jr., J. (2012). *DOT/FAA/TC-12/22 Performance Assessment of a Hybrid Radar and Electro-Optical Foreign Object Debris Detection System*.
- FAA, Herricks, E. E., Lazar III, P., Woodworth, E., & Patterson, Jr., J. (2011). *DOT/FAA/AR-11/12 Performance Assessment of a Mobile, Radar-Based Foreign Object Debris Detection System*.
- FAA, & O'donnell, M. J. (2009). *AC 150/5220-24, Airport Foreign Object Debris (FOD) Detection Equipment*.
- Gibaja Bautista, R. S., & Rojas Ruiz, R. D. (2021). *Controles topográficos en pendientes pronunciadas para evitar deslizamientos utilizando vehículos aéreos no tripulados (UAV)*.
- Gibeault, F. (2016). *BART CUSTERS, EDITOR, THE FUTURE OF DRONE USE, THE HAGUE, T.M.C. ASSER PRESS*.
- Giraldo Jimenéz, J. J., Jimenez Gómez, A. J., & Henao Cardona, J. M. (2021). *CONSTRUCCIÓN DE UN MODELO DE DETECCIÓN DE GANADO BOVINO EN ELORIENTE ANTIOQUEÑO, MEDIANTE EL SISTEMA DE VISIÓN DE UN DRON USANDO ALGORITMOS MODERNOS DE DETECCIÓN DE OBJETOS*. Universidad Católica de Oriente.
- Gómez Bedoya, D. I., & Castrillón, R. (2019). Reconocimiento automático de ganado bovino a partir de imágenes aéreas tomadas con drones: Un enfoque exploratorio. *III Congreso Internacional En Inteligencia Ambiental, Ingeniería de Software y Salud Electrónica y Móvil*, 32–39.

- Gómez Marín, F. D., & Hernández, H. M. (2019). *DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA AUTÓNOMO DE DETECCIÓN Y REPULSIÓN DE AVES EN UN CULTIVO DE FRESAS MEDIANTE EL USO DE UN DRON COMERCIAL*. Universidad Piloto de Colombia.
- Gonzalez, R. C., & Woods, R. E. (2002). *Digital Image Processing*.
- Grupo de Investigación de Accidentes – GRIAA. (2018). *Informe Final Accidente DC-3 HK3293*. <https://www.aerocivil.gov.co/autoridad-de-la-aviacion-civil/investigacion/Accidentes%202018/1.%20Acc%20HK3293%20Aircolombia%20San%20Felipe%202015-jul-18.pdf>
- Grupo de Investigación de Accidentes – GRIAA. (2020). *Informe Final Incidente Daños en aeronave por FOD al iniciar despegue Airbus A320-214 Matrícula N740AV*. <https://www.aerocivil.gov.co/autoridad-de-la-aviacion-civil/investigacion/Incidentes%20Graves%202020/Informe%20Final%20Incid%20Gr%20N740AV%20Avianca%20Pasto%2004-oct-2020.pdf>
- Hassanalian, M., & Abdelkefi, A. (2017). Classifications, applications, and design challenges of drones: A review. In *Progress in Aerospace Sciences* (Vol. 91, pp. 99–131). Elsevier Ltd. <https://doi.org/10.1016/j.paerosci.2017.04.003>
- He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2), 386–397. <https://doi.org/10.48550/arxiv.1703.06870>
- Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2014). *METODOLOGÍA DE LA INVESTIGACIÓN* (6th ed.).
- Hsu, A., Lo, A. J., Dorian, E. K., & Martinez, G. (2019). *Manual de vuelo de drones*.
- IDEAM. (2012). *AERÓDROMO EL DORADO SKBO-BOGOTÁ 1 2012 IDEA M CLIMATOLOGÍA AERONÁUTICA AERÓDROMO EL DORADO SKBO-BOGOTÁ COLOMBIA*.
- Isasi Vinuela, P., & Galván León, I. M. (2004). *REDES DE NEURONAS ARTIFICIALES: Un enfoque práctico*.
- Iván Jiménez Jiménez, S., Ojeda Bustamante, W., de Jesús, M., & Pablo, M. (2019). *INFORMACIÓN DE DRONES Y SU ANÁLISIS EN LA AGRICULTURA DE PRECISIÓN*. www.riego.mx
- Jocher, G. (2023). *GitHub - ultralytics/yolov5: YOLOv5*. <https://github.com/ultralytics/yolov5#pretrained-checkpoints>
- Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., NanoCode012, Kwon, Y., Michael, K., TaoXie, Fang, J., imyhxy, Lorna, Yifu), 曾逸夫(Zeng, Wong, C., V, A., Montes, D., Wang, Z., Fati, C., Nadar, J., Laughing, ... Jain, M. (2022, November 22). *Ultralytics*

- Yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation.*
<https://doi.org/10.5281/ZENODO.7347926>
- Lecun, Y., Eon Bottou, L., Bengio, Y., & Hinton, G. E. (1998). *Gradient-Based Learning Applied to Document Recognition.*
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9905 LNCS, 21–37. https://doi.org/10.1007/978-3-319-46448-0_2/FIGURES/5
- Lopez Poveda, E. (2002). *Las técnicas de análisis de imagen tienen aplicaciones en astronomía, teledetección, y también en neurociencias.* Popular Libros.
- Makesense.ai. (2022, January). *Makesense.ai.* <https://www.makesense.ai/>
- Nelson, J. (2020, October 28). *How to Create a Train Test Split.*
<https://blog.roboflow.com/train-test-split-with-roboflow/>
- OACI. (1983). *MANUAL DE SERVICIOS DE AEROPUERTOS PARTE 8 SERVICIOS OPERACIONALES DE AEROPUERTO.* www.afeonline.com
- OACI. (1986). *MANUAL DE SISTEMAS DE GUIA Y CONTROL, DEL MOVIMIENTO EN LA SUPERFICIE (SMGCS).* <https://pdfslide.net/documents/doc-9476-manual-de-sistemas-de-guia-y-control-del-movimiento-en-la-superficie.html?page=2>
- OACI. (2011). *Cir 328 AN/190 Unmanned Aircraft Systems (UAS).*
- OACI. (2015). *Doc 10019 AN/507 Manual on Remotely Piloted Aircraft Systems (RPAS).*
- OACI. (2016). *Anexo 14 - Aeródromos (7th ed., Vol. 1).*
- OACI. (2018). *Manual de gestión de la seguridad operacional.*
<https://www.aerocivil.gov.co/autoridad-de-la-aviacion-civil/biblioteca-tecnica/Gestin%20de%20Seguridad/Documento%20OACI%209859%20-%20tercera%20edici%C3%B3n%202013.pdf>
- OPAIN. (2017). *GSO-PG-0002 PROGRAMA DE PREVENCIÓN FOD AEROPUERTO INTERNACIONAL EL DORADO.*
- OPAIN. (2018). *PROCEDIMIENTO PARA EL CONTROL DE LAS NORMAS DE TRANSITO PARA CONDUCTORES Y PEATONES EN PLATAFORMA.*
- OPAIN. (2019). *GESTIÓN DE OPERACIONES PLAN MANEJO DE FOD.*
<https://www.opain.co/archivos/PLAN%20DE%20MANEJO%20FOD%20V1.0.pdf>
- OPAIN. (2020). *GOP-PN-003 PLAN DE MANEJO FOD PLAN DE MANEJO FOD AEROPUERTO EL DORADO DE BOGOTÁ.*

- Ortíz, J. (2020, March 30). *¿Qué es un F.O.D.? - aeronauticapv.com*.
<https://aeronauticapv.com/2020/03/30/que-es-un-f-o-d/>
- Panchal, S. (2019, December 3). *Why Google Colab?* <https://medium.com/analytics-vidhya/why-google-colab-c8ac16587dff>
- Perales Izquierdo, P. (2022). *Sistema de detección de submarinistas*. Escuela politécnica superior.
- Pino V., E. (2019). Los drones una herramienta para una agricultura eficiente: un futuro de alta tecnología. *Idesia (Arica), ahead*, 0–0. <https://doi.org/10.4067/s0718-34292019005000402>
- Python Software Fndn. (2022). *El tutorial de Python – documentación de Python - 3.11.1*.
<https://docs.python.org/es/3/tutorial/>
- Rath, S., & Gupta, V. (2022, November 29). *Performance Comparison of YOLO Object Detection Models – An Intensive Study*. <https://learnopencv.com/performance-comparison-of-yolo-models/>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015). *You Only Look Once: Unified, Real-Time Object Detection*. <http://pjreddie.com/yolo/>
- Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137–1149.
<https://doi.org/10.48550/arxiv.1506.01497>
- Reuter, F., & Pedenovi, A. (2019). *LOS DRONES Y SUS APLICACIONES A LA INGENIERÍA*.
- Roboflow. (2022). *Roboflow: Give your software the power to see objects in images and video*. <https://roboflow.com/>
- Rosebrock, A. (2017). *Deep Learning for Computer Vision with Python* (1st ed.). PYIMAGESEARCH.
- Rosenbacher, J. (2022, February 24). *How many images do you need to train a model?*
<https://blog.roboflow.com/images-train-model/>
- Ruiz De La, Á., & Molina, T. (2020). Drones y movilidad segura y conectada Drones and Safe, Connected Mobility. *Ingeniería Civil* 197, 12–23.
- Russell, S. J. (Stuart J., Norvig, Peter., Corchado Rodríguez, J. Manuel., & Joyanes Aguilar, Luis. (2004). *INTELIGENCIA ARTIFICIAL: un enfoque moderno*. Pearson Prentice Hall.

- Saha, S. (2018, December 15). *A Comprehensive Guide to Convolutional Neural Networks – the ELI5 way*. <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
- Shearer, C. (2000). The CRISP-DM Model: The New Blueprint for DataMining. *Journal of Fata Waewhousing*, 5, 13–22.
- SRVSOP. (2017). *Reglamento Aeronáutico Latinoamericano LAR 139 Certificación de aeródromos-Requisitos para operadores de aeródromos* (Vol. 1).
- SRVSOP. (2018). *Reglamento Aeronáutico Latinoamericano LAR 153 Operación de aeródromos* (Vol. 2).
- SRVSOP. (2020). *CA AGA 153 015 LAR 153 – Guía para el operador/explotador de aeródromo para control de objetos extraños (FOD) en el área operacional*.
- The FOD Control Corp. (2020). *TracSweep Operator’s Manual*. www.FODControl.com
- Torres, J. (2018). *Deep Learning Introducción práctica con Keras* (WATCH THIS SPACE, Vol. 1). https://torres.ai/deep-learning-inteligencia-artificial-keras/#Inteligencia_artificial_Machine_Learning_y_Deep_Learning
- Torres, J. (2020). DEEP LEARNING INTRODUCCIÓN PRÁCTICA CON KERAS. In *Artificial Intelligence: a Modern Approach*. Prentice Hall. <https://torres.ai/deep-learning-inteligencia-artificial-keras/>
- UAEAC, & Salazar Gómez, J. C. (2018). *Resolución 04201 Incorporación a la norma RAC 91 de los Reglamentos Aeronáuticos de Colombia*.
- Ukessays. (2020). *Foreign Object Damage: Prevention, Costs and Detectors*. <https://www.ukessays.com/essays/aviation/foreign-object-damage-prevention-costs-and-detectors.php>
- Ultralytics. (2022). *YOLOv5 Tutorial - Colaboratory*. <https://colab.research.google.com/github/ultralytics/yolov5/blob/master/tutorial.ipynb>
- Vara Castañeda, R. D. (2013). *Metodología para la identificación de peligros y gestión de riesgos en los motores debido a FOD*.
- Villalta Ayala, M. E., & Guerra Jara, G. S. (2016). *SISTEMA DE DISEÑO DE DRONES*.
- Villasenor, J. (2013). OBSERVATIONS FROM ABOVE: UNMANNED AIRCRAFT SYSTEMS AND PRIVACY. *Harvard Journal of Law & Public Policy*, 36, 457–517. <http://www.fas.org/irp/program/collect/uas-future.pdf>
- Yan, X., Liu, Y., Xu, Y., & Jia, M. (2020). Multistep forecasting for diurnal wind speed based on hybrid deep learning model with improved singular spectrum

decomposition. *Energy Conversion and Management*, 225.

<https://doi.org/10.1016/j.enconman.2020.113456>

Yuneec futurhobby. (2015). *Yuneec Q500 4K*. <https://yuneec-futurhobby.com/yuneec-q500-4k>

Yuneec Futurhobby. (2016). *Yuneec Typhoon H - Yuneec Futurhobby*. <https://yuneec-futurhobby.com/yuneec-typhoon-h-pro-realsense>



LOS LIBERTADORES
FUNDACIÓN UNIVERSITARIA