



**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES**

**TRABAJO DE TITULACION PREVIO A LA OBTENCIÓN DEL TÍTULO
DE INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES**

**TEMA: “SISTEMA DE ESTIMACIÓN DE NÚMERO DE PERSONAS EN
TIEMPO REAL DURANTE MISIONES DE RECONOCIMIENTO DEL
EJÉRCITO ECUATORIANO UTILIZANDO VEHÍCULOS AÉREOS NO
TRIPULADOS MULTIROTOR”**

AUTOR:

AGUAIZA GUERRERO, CÉSAR HERNÁN

DIRECTOR: DR. AGUILAR CASTILLO, WILBER GEOVANNY

SANGOLQUI, 24 de AGOSTO

2018



DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES
CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

CERTIFICACIÓN

Certifico que el trabajo de titulación *"SISTEMA DE ESTIMACIÓN DE NÚMERO DE PERSONAS EN TIEMPO REAL DURANTE MISIONES DE RECONOCIMIENTO DEL EJÉRCITO ECUATORIANO UTILIZANDO VEHÍCULOS AÉREOS NO TRIPULADOS MULTIROTOR"* fue realizado por el señor *Aguaiza Guerrero, César Hernán* el mismo que ha sido revisado en su totalidad, analizado por la herramienta de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 24 de Agosto de 2018

Dr. Wilbert Geovanny Aguilar Castillo

CI: 0703844696



DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

AUTORÍA DE RESPONSABILIDAD

Yo, *Aguaiza Guerrero, César Hernán*, declaro que el contenido, ideas y criterios del trabajo de titulación: *SISTEMA DE ESTIMACIÓN DE NÚMERO DE PERSONAS EN TIEMPO REAL DURANTE MISIONES DE RECONOCIMIENTO DEL EJÉRCITO ECUATORIANO UTILIZANDO VEHÍCULOS AÉREOS NO TRIPULADOS MULTIROTOR*, es de mi autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz.

Sangolquí, 24 de agosto de 2018.

Aguaiza Guerrero César Hernán

C.C.: 1713415154



DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES
CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

AUTORIZACION

*Yo, Aguaiza Guerrero, César Hernán autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **SISTEMA DE ESTIMACIÓN DE NÚMERO DE PERSONAS EN TIEMPO REAL DURANTE MISIONES DE RECONOCIMIENTO DEL EJÉRCITO ECUATORIANO UTILIZANDO VEHÍCULOS AÉREOS NO TRIPULADOS MULTIROTOR**, en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.*

Sangolquí, 24 de agosto de 2018

Aguaiza Guerrero César Hernán

C.C.: 1713415154

DEDICATORIA

Este trabajo quiero dedicar de manera especial, al amor de mi vida Liliana Granda, por ser la amiga y esposa que me motivó a conseguir esta meta, por acompañarme durante estos años de estudio, por cuidar de nuestras hijas mientras yo no estaba, por brindarme la fortaleza para seguir adelante y estar a mi lado en los buenos y malos momentos.

A mis preciosas hijas Karlita y Aby, por comprender todas las veces que no pude salir a jugar con ustedes, con su amor e inocencia fueron mi motor e inspiración para alcanzar este objetivo.

A mi madre Leonor Guerrero por ser un ejemplo de lucha y constancia, con sus consejos y apoyo incondicional siempre pendiente de sus hijos y a mis hermanas Paola y Cristina por el apoyo brindado durante todo este tiempo.

César Hernán Aguaiza Guerrero

AGRADECIMIENTO

A Dios por darme la vida, a la vida por darme una familia y a mi familia por estar siempre a mi lado, gracias a ustedes en estos años de estudio he recibido la fuerza para seguir adelante y conseguir la meta propuesta.

A mis amigos militares del arma de comunicaciones que supieron darme un consejo en momentos difíciles, a mis profesores que impartieron sus conocimientos de manera desinteresada para mi formación profesional, a mis amigos electrónicos con quienes pasamos días y noches en las aulas o laboratorios terminando los proyectos.

Al Ing. Rubén León y Dr. Wilbert Aguilar, por enseñarme que la educación superior se adquiere en base al esfuerzo y sacrificio que el alumno se proponga.

Gracias a todos ustedes he culminado una etapa profesional en mi vida, con los buenos consejos y frases de aliento que he recibido durante estos años de estudio, he alcanzado mi objetivo, de todo corazón muchas gracias.

César Hernán Aguaiza Guerrero

ÍNDICE DE CONTENIDOS

CERTIFICACIÓN.....	i
AUTORIA DE RESPONSABILIDAD.....	ii
AUTORIZACIÓN.....	iii
DEDICATORIA	iv
AGRADECIMIENTO	v
ÍNDICE DE CONTENIDOS	vi
ÍNDICE DE TABLAS.....	ix
ÍNDICE DE FIGURAS.....	x
RESUMEN.....	xii
ABSTRACT	xiii
CAPITULO I.....	1
1. INTRODUCCION.....	1
1.1 Antecedentes	1
1.2 Justificación e importancia.....	2
1.3 Alcance del proyecto.....	3
1.4 Objetivos	4
1.4.1 Objetivo General.....	4
1.4.2 Objetivos Específicos	4
1.5 Descripción del proyecto de investigación	5
CAPITULO II.....	7
2. MARCO TEÓRICO.....	7

2.1 Vehículo Aéreo no Tripulados Multirroto (UAV).....	7
2.2 DJI Phantom 4.....	8
2.3 Video Streaming.....	9
2.3.1. Formatos de video streaming	9
2.3.2. Métodos de transmisión	10
2.3.3 Protocolos más utilizados.....	10
2.4 Protocolo de Mensajería en Tiempo Real (RTMP).....	12
2.5 Servidor Web Nginx	13
2.6 Visión por computadora.....	14
2.7 Redes Neuronales Convolucionales.....	16
CAPÍTULO III	25
3. ANALISIS DE ESTANDARES DE COMUNICACIÓN INALÁMBRICA	25
3.1 Estándares de comunicaciones inalámbricas	25
3.2 Implementación de la tecnología inalámbrica.....	27
3.3 Arquitectura de despliegue.....	29
3.4 Resultados de las pruebas experimentales	31
CAPITULO IV	38
4. DESARROLLO DEL ALGORITMO DE DETECCION Y CONTEO DE PERSONAS 38	
4.1 Detección de objetos en tiempo real	38
4.2 Desarrollo y entrenamiento del sistema	40
4.3 TensorFlow adaptado a YOLO	41
4.4 Instalación e implementación del sistema.....	43
4.4.1 Anaconda distribución Python 3.6.6	43
4.4.2 TensorFlow versión con GPU	45

4.4.3 OpenCV (Open Source Computer Vision Library).....	48
4.4.4 Repositorio DarkFlow	48
4.4.5 Procesamiento de video en tiempo real.....	49
CAPITULO V	50
5. EXPERIMENTACION, RESULTADOS Y DISCUSION DEL SISTEMA DE CONTEO DE PERSONAS	50
5.1 Métricas de evaluación.....	50
5.2 Diseño del experimento.....	51
5.3 Análisis de resultados.....	67
CAPITULO VI.	71
6. CONCLUSIONES Y RECOMENDACIONES	71
6.1 CONCLUSIONES	71
6.2 RECOMENDACIONES	72
REFERENCIAS BIBLIOGRÁFICAS	74
Bibliografía	74

ÍNDICE DE TABLAS

Tabla 1 <i>Origen y desarrollo de Redes Neuronales Convolucionales</i>	17
Tabla 2 <i>Tecnologías Inalámbricas</i>	25
Tabla 3 <i>Factores y criterios del modelo FURPS</i>	28
Tabla 4 <i>Rendimiento del Servidor Streaming Nginx</i>	37
Tabla 5 <i>Frames capturados para encontrar la sensibilidad y especificidad del sistema a una altura de 10 metros.</i>	58
Tabla 6 <i>Frames capturados para encontrar la sensibilidad y especificidad del sistema a una altura de 20 metros.</i>	59
Tabla 7 <i>Frames capturados para encontrar la sensibilidad y especificidad del sistema a una altura de 40 metros.</i>	61
Tabla 8 <i>Frames capturados para encontrar la sensibilidad y especificidad del sistema a una altura de 60 metros.</i>	62
Tabla 9 <i>Frames capturados para encontrar la sensibilidad y especificidad del sistema a una altura de 60 metros con zoom</i>	64
Tabla 10 <i>Frames capturados para encontrar la sensibilidad y especificidad del sistema a una altura de 100 metros.</i>	65
Tabla 11 <i>Frames capturados para encontrar la sensibilidad y especificidad del sistema a una altura de 100 metros con zoom</i>	67
Tabla 12 <i>Resultados de las pruebas en sensibilidad y especificidad</i>	69

ÍNDICE DE FIGURAS

Figura 1. Sistema de detección de personas Dron – Estación Remota	6
Figura 2. Video Streaming con protocolo RTMP	11
Figura 3. Representación de estados cliente servidor mediante RTMP.....	13
Figura 4. Representación de una imagen en píxeles	14
Figura 5. Comparación píxeles vs features	15
Figura 6. Red neuronal con 3 neuronas de entrada, 4 ocultas y 2 de salida.....	18
Figura 7. Red LeNet-5 arquitectura	20
Figura 8. Estructura de funcionamiento YOLO.....	22
Figura 9. Cuadros delimitadores	23
Figura 10. Configuración protocolo RTMP y HLS	30
Figura 11. Estado de transmisión en vivo	31
Figura 12. Datos del ancho de banda utilizado	32
Figura 13. Tráfico Estimado de Red	33
Figura 14. Rendimiento de los Recursos del CPU.....	34
Figura 15. Rendimiento de Memoria, Cores y CPU	35
Figura 16. Transmisión en Tiempo Real del Servidor Nginx	36
Figura 17. Proceso de confianza, cuadros delimitadores, probabilidad de clase en una imagen..	40
Figura 18. Instalación Anaconda con distribución Python e importación de librerías	44
Figura 19. Plataforma de instalación CUDA Toolkit 8.0.....	46
Figura 20. Comprobación de TensorFlow importado a Python.....	47
Figura 21. Diagrama de flujo del sistema de identificación de personas.....	49

Figura 22. Relación de sensibilidad y especificidad	51
Figura 23. Sistema de detección a 20 m de altura del dron y 45° de inclinación de la cámara	52
Figura 24. Sistema de detección a 20 m de altura del dron y 90° de inclinación de la cámara	53
Figura 25. Sistema de detección a 20 m de altura del dron, 45° de inclinación, las 08:00.....	54
Figura 26. Sistema de detección a 20 m de altura del dron, 45° de inclinación, siendo 17:30.....	55
Figura 27. Sistema de detección a 20 m de altura del dron, 45°, siendo las 14:00, zona congestionada	56
Figura 28. Detección a 10m de altura del dron, ángulo cámara 45°, frame 2.....	57
Figura 29. Detección a 20m de altura del dron, ángulo cámara 50°, frame 1	59
Figura 30. Detección a 40m de altura del dron, ángulo cámara 55°, frame 5	60
Figura 31. Detección a 60m de altura del dron, ángulo cámara 60°, frame 7.....	62
Figura 32. Detección a 60m de altura del dron con zoom, ángulo cámara 65°, frame 10	63
Figura 33. Detección a 100m de altura del dron, ángulo cámara 70°, frame 4.....	65
Figura 34. Detección a 100 m de altura del dron con zoon, ángulo cámara 75°, frame 8	66
Figura 35. Diagrama posicional del dron en referencia al objetivo	68
Figura 36. Comportamiento del Sistema a diferentes alturas	70

RESUMEN

En el presente proyecto de investigación, se desarrolla y propone un sistema de identificación de personas utilizando un vehículo aéreo no tripulado o dron, sistema que se encuentra en una estación remota constituida por una computadora portátil con características de última generación. Este trabajo está dividido en dos partes: la primera, analiza los estándares de comunicaciones inalámbricas para establecer la comunicación dron – estación remota, empleando un servidor streaming de código abierto, con arquitectura asíncrona basada en eventos, en base al servicio bajo demanda de envío punto a punto (unicast) y mediante la interacción del protocolo RTMP y HLS a fin de mantener la calidad de servicio durante la transmisión del video: la segunda parte se basa en el procesamiento digital de imágenes enviadas por el dron a través del algoritmo YOLO (solo se observa una vez), el mismo que utiliza redes neuronales convolucionales para la clasificación de imágenes y su entrenamiento. YOLO se complementa con la interacción de librerías como OpenCV, Anaconda, CUDA, TensorFlow y está desarrollado utilizando programación orientada a objetos con el intérprete Python. Finalmente, se realiza las pruebas del sistema completo, bajo diferentes condiciones de altura de vuelo del dron, iluminación, características del terreno, posición del dron y resolución de la cámara, a fin de evaluar objetivamente el desempeño del sistema de identificación de personas en tiempo real.

PALABRAS CLAVES:

- **VIDEO STREAMING**
- **PROTOCOLOS DE COMUNICACIÓN**
- **REDES NEURONALES CONVOLUCIONALES**
- **VEHÍCULOS AEREOS NO TRIPULADOS**
- **PROCESAMIENTO DIGITAL DE IMÁGENES**

ABSTRACT

In the present research project, a system of identification of people using an unmanned aerial vehicle is developed and proposed, the system is located in a remote station constituted by a portable computer with characteristics of the latest generation. This work is divided into two parts: the first one, analyzes the standards of wireless communications to establish remote communication, the remote station, the server of an open source transmission server, the asynchronous architecture based on events, the base of the service on demand from a single-point point (unicast) and through the interaction of the RTMP and HLS protocol in order to maintain the quality of service during the transmission of the video: the second part is based on the processing of digital images sent by the map through the YOLO algorithm (only observed once), the same one that uses convolutional neural networks for the classification of images and their training. YOLO is complemented with the interaction of libraries like OpenCV, Anaconda, CUDA, TensorFlow and is late for the orientation of a target with the Python interpreter. Finally, the tests of the complete system, lighting, terrain characteristics, radar position and camera resolution are carried out, in order to objectively evaluate the performance of the people identification system in real time.

KEYWORDS:

- **VIDEO STREAMING**
- **COMUNICATION PROTOCOLS**
- **CONVOLUTIONAL NEURAL NETWORKS**
- **NON-CREATED AIR VEHICLES**
- **DIGITAL PROCESSING OF IMAGES**

CAPITULO I

1. INTRODUCCION

1.1 Antecedentes

Las misiones de reconocimiento que hasta el momento realiza el Ejército Ecuatoriano se orientan a identificar las características del terreno, sin embargo, el avance tecnológico permite optimizar, modernizar e incrementar estas tareas, de acuerdo al nuevo escenario de combate, por lo que es necesario incorporar algoritmos avanzados que permitan estimar un determinado número de personas durante tareas de reconocimiento con Vehículos Aéreos No Tripulados multirotor (drones). Un sistema con estas características, permitirá suministrar con información veraz y oportuna para la toma de decisiones en el teatro de operaciones.

Bajo estas premisas el Centro De Investigación de Aplicaciones Militares (CICTE), desarrolla el proyecto “Sistemas de percepción y localización para la navegación autónoma de vehículos aéreos multirotores de micro escala en entornos GPS-denegados (Visual Nav Drone)” bajo la dirección del Dr. Wilbert Aguilar. Este proyecto está orientando a la investigación y fusión de datos inerciales e identificación de movimiento estimado, con algoritmos de visión por computadora.

El Ejército Ecuatoriano posee drones de la casa comercial DJI que no pueden ser manipulados tecnológicamente, debido a que pierden la garantía, siendo el principal problema obtener las secuencias de imágenes en una estación remota. Estos drones no poseen un sistema que permita establecer el número de personas que se encuentran durante una misión de reconocimiento, limitando el cumplimiento de las tareas del escalón superior.

1.2 Justificación e importancia

El CICTE gestiona proyectos de investigación científica y desarrollo tecnológico que permiten aportar al mejoramiento y crecimiento técnico operativo del Ejército Ecuatoriano. Una de las falencias presentadas por unidades militares que cumplen misiones de reconocimiento con drones, es que no disponen de un sistema que permita obtener mayor información en tiempo real de las actividades realizadas durante un movimiento o aglomeración de personas en determinados sectores de la urbe o fuera de ella.

Las unidades militares al disponer de drones con licencia de propiedad intelectual (Kendoul, 2012) (Copyright © 2018 DJI, 2018), se ven limitados en sus acciones, limitándose a las prestaciones que brinda el fabricante sin posibilidad de agregar o disminuir funciones importantes (DJI, 2016). Se puede mejorar las características de los drones a costos que sobrepasan los presupuestos asignados por el Estado al Ejército (Jara-Olmedo A. , y otros, 2018), (Jara-Olmedo A. , Medina-Pazmiño, Tozer, Aguilar, & Pardo, 2018), (Pardo, Aguilar, & Toulkeridis, 2017), (Andrea, Byron, Jorge, Inti, & Aguilar, 2018), (Basantes, y otros, 2018), (Orbea, y otros, Math Model of UAV Multi Rotor Prototype with Fixed Wing Aerodynamic Structure for a Flight Simulator, 2017), (Orbea, y otros, Vertical take off and landing with fixed rotor, 2017). Este proyecto de investigación con un fuerte componente de desarrollo tecnológico presenta una propuesta innovadora, endógena y costo cero, de un sistema que permite estimar el número de personas en tiempo real, existentes en la secuencia de imágenes capturadas y enviadas desde un dron hacia la estación remota (Valdez, 2016), (Aguilar, Cobeña, Rodriguez, Salcedo, & Collaguazo, 2018), (Aguilar, Manosalvas, Guillén, & Collaguazo, 2018), (Aguilar, y otros, On-Board Visual SLAM on a UGV Using a RGB-D Camera, 2017), (Aguilar, y otros, Real-Time 3D

Modeling with a RGB-D Camera and On-Board Processing, 2017), (Aguilar, y otros, Visual SLAM with a RGB-D Camera on a Quadrotor UAV Using on-Board Processing, 2017).

El proyecto se basa, primeramente, en implementar un sistema de comunicación inalámbrica que permita transmitir los datos obtenidos por el dron, hacia la estación remota (Perez, 2016) (Galiano, 2014). Debido a que en la actualidad la información obtenida por el dron llega a dispositivos móviles cuya capacidad de procesamiento de imágenes es limitado, seguidamente se requiere que el algoritmo de detección y conteo de personas utilizando las secuencias de las imágenes obtenidas por el dron (Pussiol, 2014) sea implementado en un ordenador con mayores capacidades de procesamiento, de esta forma optimizar el cumplimiento de las misiones de reconocimiento aéreo (Amaguaña, Collaguazo, Tituaña, & Aguilar, 2018).

1.3 Alcance del proyecto

El presente proyecto de investigación consiste en elaborar un sistema de estimación de número de personas en tiempo real, con drones que poseen licencia de propiedad intelectual, para ser aplicado en misiones de reconocimiento del Ejército Ecuatoriano. Para las pruebas de campo se utiliza la transmisión inalámbrica con tecnología WiFi (Salvador, 2016) (Cordeiro, Akhmetov, & Park Minyoung, 2010), la comunicación dron – estación remota se materializa con el servidor Nginx multiplataforma y el protocolo RTMP (Real-Time Messaging Protocol), que utiliza TCP a nivel de la capa de transporte y el flujo de datos multimedia esta codificado en formato FLV (Flash Video), lo que permitirá obtener el video capturado por la cámara del dron en una computadora portátil a través de la tecnología streaming (Pozueco Alvarez, 2014).

Con los datos en la estación remota, se procede al procesamiento de imágenes mediante el sistema de detección de personas en tiempo real YOLO (your only look once) (Redmon, Divvala,

Girshick, & Farhadi, 2016), que utiliza una red neuronal a la imagen completa, dividiéndola en regiones y cuadros de límite para obtener las probabilidades de cada región (Redmon & Farhadi, You Only Look Once, 2015). Los cuadros delimitadores estarán ponderados por las probabilidades o priori, optimizando el modelo en comparación con los sistemas basados en clasificadores. De esta forma se materializa un entrenamiento mejorado y de alto rendimiento, que permitirá realizar la detección de personas desde la cámara del dron.

Al ser parte de un proyecto de investigación del CICTE, el alcance del presente esta direccionado por una necesidad del Ejército Ecuatoriano que es la identificación de personas en tiempo real desde un dron en movimiento, para lo que se ha definido límites de altura para realizar las pruebas de detección y obtener el mejor desempeño posible del sistema propuesto.

1.4 Objetivos

1.4.1 Objetivo General

Desarrollar un sistema de estimación de número de personas en tiempo real, durante misiones de reconocimiento militares con vehículos aéreos no tripulados multirotor.

1.4.2 Objetivos Específicos

- Implementar un sistema de comunicación inalámbrica, que permita la obtención de imágenes RGB del dron en una estación remota, en tiempo real.
- Desarrollar un sistema de procesamiento de imágenes usando diferentes modelos de reconocimiento de objetos o personas.
- Investigar un sistema basado en la detección de objetos mediante la clasificación y entrenamiento con redes neuronales convolucionales.
- Desarrollar un sistema de extracción de características basado en las librerías de visión por computadora.

- Optimizar los sistemas de comunicación en tiempo real entre el dron y estación remota para la detección de personas en diferentes ambientes.

1.5 Descripción del proyecto de investigación

El proyecto se basa en una metodología de investigación que analiza los estándares actuales de comunicaciones inalámbricas como son bluetooth, Xbee, Zigbee, Wifi u otras posibles (Perez, 2016) (Vasilios & Dimitrios, 2015), se trabajó con el dron Phantom 4 de la casa comercial DJI para obtener las imágenes, un router CISCO que permite direccionar la ruta adecuada del tráfico de datos recibidos desde el dron (Ariganello, 2015), una computadora con sistema operativo Ubuntu 16.04 core i7, con tarjeta de video nVidia y procesamiento paralelo con CUDA que permitirá el manejo de imágenes a un alto rendimiento (NVIDIA C. , 2018).

Inicialmente se estableció la comunicación dron – estación remota mediante Nginx que es un servidor web de código abierto (Ruoyu & Li, 2018), eficiente en la entrega de datos con arquitectura asíncrona basada en eventos, ofreciendo contenido en formato MP4 y video FLV (flash video) (Nginx.com), el video streaming se basa a un servicio bajo demanda identificado por el envío punto a punto (unicast), asegurando la interacción dron – estación remota utilizando el protocolo RTMP (Real Time Messaging Protocol) compatible con una aplicación del dron Panthom 4 (PANTHOM 4, 2016), para obtener resultados confiables en calidad de servicio (QoS) se realizó un monitoreo de funcionalidad del servidor streaming mediante comandos de Linux.

Con la transmisión de video a la estación remota todos los datos obtenidos ingresan a un procesamiento de imágenes mediante la aplicación del algoritmo YOLO, las librerías OpenCv, Anaconda, CUDA son el complemento para la programación en Python, este lenguaje es de fácil comprensión con alta estructura de datos y enfoque simple (Python Software, 2001 2018), pero

efectivo gracias a la programación orientada a objetos, su interprete contiene funciones y tipos de datos implementados en C o C++, u otros idiomas que se puedan llamar desde C. El modelo YOLO ha sido adaptado para TensorFlow evitando dependencias con terceros para lo cual se agrega una función a Darkflow que permite guardar puntos de control con el meta – gráfico, importando los pesos pre entrenados de DarlKNet a TensorFlow a fin de obtener la detección de personas con elevado puntaje de precisión (Abadi & Barham, 2016).

En la figura 1 se indica la estructura del proyecto, especificando la comunicación entre ESTACION REMOTA – DRON.

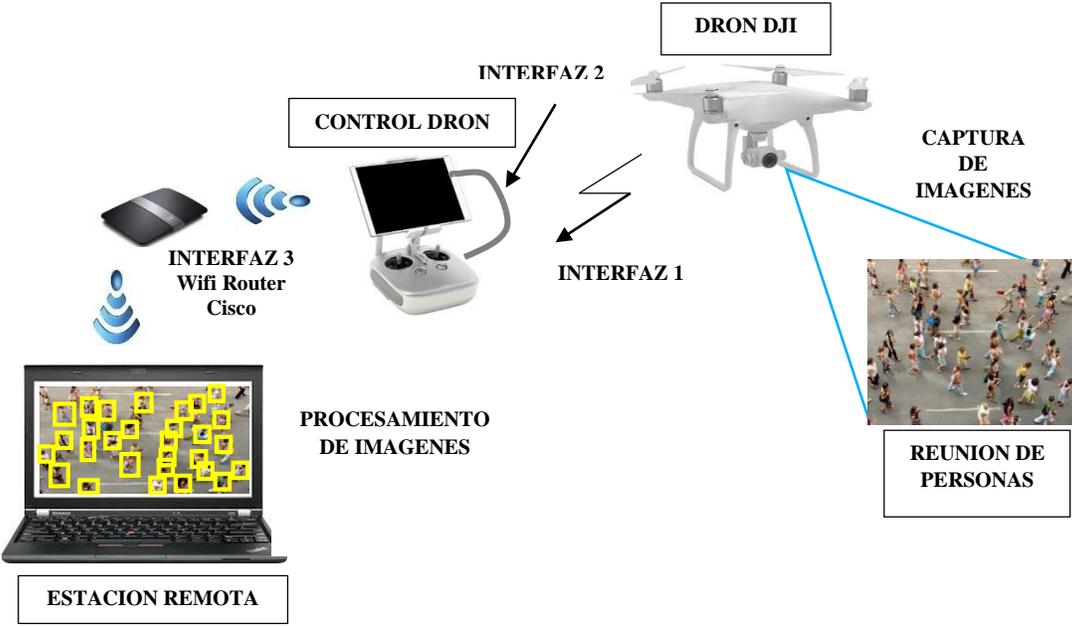


Figura 1. Sistema de detección de personas Dron – Estación Remota

CAPITULO II

2. MARCO TEÓRICO

2.1 Vehículo Aéreo no Tripulados Multirrotor (UAV)

El UAV multirrotor, mejor conocido como DRON, es un helicóptero motorizado que vuela sin necesidad de un piloto y es controlado de forma remota (Kendoul, 2012) (Huang, 2008), o mediante algoritmos de navegación autónoma como planificadores y controladores de ruta (Aguilar & Morales, 3D Environment Mapping Using the Kinect V2 and Path Planning Based on RRT Algorithms, 2016), (Aguilar, Angulo, & Costa-Castello, Autonomous Navigation Control for Quadrotors in Trajectories Tracking, 2017), (Aguilar, Morales, Ruiz, & Abad, RRT* GL Based Optimal Path Planning for Real-Time Navigation of UAVs, 2017), (Aguilar, Abad, Ruiz, Aguilar, & Aguilar-Castillo, 2017), (Aguilar, Morales, Ruiz, & Abad, RRT* GL Based Path Planning for Virtual Aerial Navigation, 2017). Estos sistemas de aeronaves se encuentran en varias aplicaciones civiles como militares, ofreciendo grandes ventajas en vigilancia, reconocimiento o inspección de extensas áreas geográficas en ambientes peligrosos donde el ser humano ve limitado su ingreso.

Los drones se clasifican en cinco categorías que varían de acuerdo a: la escala, carga útil, resistencia y rango de operatividad. La categoría I son los de mayor escala, la presente investigación se basa en el dron categoría IV. Estos drones son portátiles, vuelan al aire libre, con un peso inferior a los 2 kg y una autonomía de vuelo de 5 a 60 minutos (Kendoul, 2012), (Salcedo, 2018), (Aguilar, Casaliglla, & Pólit, Obstacle Avoidance Based-Visual Navigation for Micro Aerial Vehicles, 2017), considerando el costo, fácil mantenimiento y operación segura se han convertido en los más utilizados para investigación y desarrollo tecnológico en los tiempos modernos (DJI, 2016).

2.2 DJI Phantom 4

El dron Phantom 4 es un cuadricoptero de la casa comercial DJI (Dajiang Innovation Technology), empresa que nació en la Universidad de Ciencia y Tecnología de Hong Kong con Frank Wang como su actual director general. DJI es la mayor fabricante y distribuidora de todo tipo de drones tanto profesionales como recreativos, actualmente su sede se encuentra en Shenzhen–China, lugar en donde se crea la mejor tecnología para drones (Copyright © 2018 DJI, 2018).

La característica principal del dron Phantom 4 es su cámara de 4k, permite grabar video a 30 fps (fotogramas por segundo), toma fotos de 12 megapíxeles y posee un sensor CMOS que proporciona nitidez y reduce el ruido. La transmisión se realiza en HD de baja latencia con un alcance de 5 km en formato MOV o MP4, la retransmisión en vivo se basa en la tecnología DJI Lightbridge de hasta 5 km de distancia en áreas libres de interferencia, proporcionando transmisión en vivo FHD (full high definition) de 1080 x 120 fps (DJI, 2016).

La tecnología DJI Lightbridge permite transmitir video digital 2.4G Full HD, capacidad de 1920 x 1080 p a 30 fps, con una antena omnidireccional de 14 dBi patch (Hiaolong, Jiang, & Yongjia, 2017). Esta tecnología alcanza una distancia de transmisión de 5 km. La cámara del Phantom 4 posee un robusto sistema de estabilización (Aguilar & Angulo, Real-Time Model-Based Video Stabilization for Microaerial Vehicles, 2016), (Aguilar & Angulo, Real-time video stabilization without phantom movements for micro aerial vehicles, 2014), (Aguilar & Angulo, Robust video stabilization based on motion intention for low-cost micro aerial vehicles, 2014), (Aguilar, Angulo, & Pardo, Motion intention optimization for multirotor robust video stabilization, 2017), una combinación de lentes de 3,5 aumentos y un zoom digital de 2x y dependiendo la altura a la que se

encuentre el dron se pueden obtener imágenes en HD para su posterior análisis (DJI, 2016), (Cedeño, Marcillo, & Pereira, 2017).

2.3 Video Streaming

En la actualidad el video streaming permite observar contenidos durante la descarga de información a través de un buffer de datos en donde se almacena contenido dando fluidez a una reproducción sin interrupciones (Soni, 2016). Para alcanzar este objetivo se debe considerar el formato de video, métodos y protocolos utilizados para la transmisión (Intriago, 2016) (Nginx.com)

2.3.1. Formatos de video streaming

El formato se define como una estructura en donde se codifica la información que será transmitida de forma optimizada, permitiendo la captura, edición, entrega y distribución de video en diferentes plataformas (Abdullah, 2018). Debe existir coordinación entre elementos de audio, imágenes y metadatos, estos últimos poseen información sobre idioma, elenco, año de producción entre otras. Se requiere de un software para que codifique y decodifique una señal de video mejor conocido como “códec”, cuya función principal será la de comprimir y descomprimir un conjunto de datos para ser almacenados y transmitidos en archivos más pequeños (Abdullah, 2018), (Constanzo, 2014).

Los codecs de la familia H.264 o mejor conocido como MPEG-4 o AVC son más versátiles, ofrecen excelente compresión con alta calidad, soporta video 3D y varios esquemas de codificación de audio (Facchini, Pérez, Hidalgo, & Cárdenas, 2018). Los videos de alta resolución 4K, ultra HD, superan la transmisión de 1080p mejorando la calidad, tiempo de codificación y requerimientos de CPU en varias plataformas que pertenecen a la familia de codecs H.265. (Pozueco Alvarez, 2014), (Galiano, 2014).

2.3.2. Métodos de transmisión

Streaming Tradicional. Se caracteriza por descarga y reproducción, donde el usuario baja un video mostrando la URL de acceso. Este método debe considerar el tamaño del video por cuanto requiere de tiempo de descarga considerable, la ventaja que una vez obtenido todo el video se puede adelantar, pausar o retroceder (Pozueco Alvarez, 2014).

Video Streaming o Live Streaming. Presentación de video en vivo, en donde la tasa de transmisión es constante sin tiempos de espera a través de internet (Vasilios & Dimitrios, 2015). Se debe considerar: una fuente (cámara del dron), un encoder, un editor de medios y una red de distribución de contenidos. Para una transmisión en tiempo real considerar: un protocolo de streaming, un servidor y un protocolo de entrega como RTMP (Real Time Media Protocol) (Intriago, 2016) (Vasilios & Dimitrios, 2015).

2.3.3 Protocolos más utilizados

Es fundamental elegir el protocolo para la reproducción de video, de esto dependerá el tipo de dispositivo a utilizarse mediante aplicaciones existentes como por ejemplo la plataforma Adobe Media Server que utiliza flujos RTMP como formato estándar, de esta manera se dividen en dos clases (Campaña, 2015):

- **Protocolos tipo Push**

Se establece la conexión entre cliente y servidor. El servidor transmite paquetes al cliente de forma continua hasta que se detiene o se interrumpe la sesión. Mantiene el enlace y espera comandos por parte del cliente para que se produzca algún cambio en el estado de la sesión. En este tipo se destaca la familia RTP (Real-time Transport Protocol) sobre UDP, pero no está orientado a la conexión limitando los mecanismos de control (Campaña, 2015).

- **Protocolos tipo Pull**

Para este tipo de protocolos la tasa de bits y condiciones de red disponibles depende del cliente, ya que debe realizar toda la gestión solicitando contenido al servidor. En este tipo se destaca Adobe RTMP (Real Time Messaging Protocol) protocolo usado por Flash sobre TCP (Transmisión Control Protocol), soporta video MP4 y FLV, audio ACC y MP3. El tipo pull realiza el soporte mediante servicios multiplexados de mensajes bidireccionales, flujo de transporte confiable e información de temporización entre pares (Adobe Developer, 2018).

La figura 2 representa el flujo de información desde una cámara hacia los usuarios de video streaming mediante el protocolo RTMP, constituyéndose en un medio de transporte para audio y video.

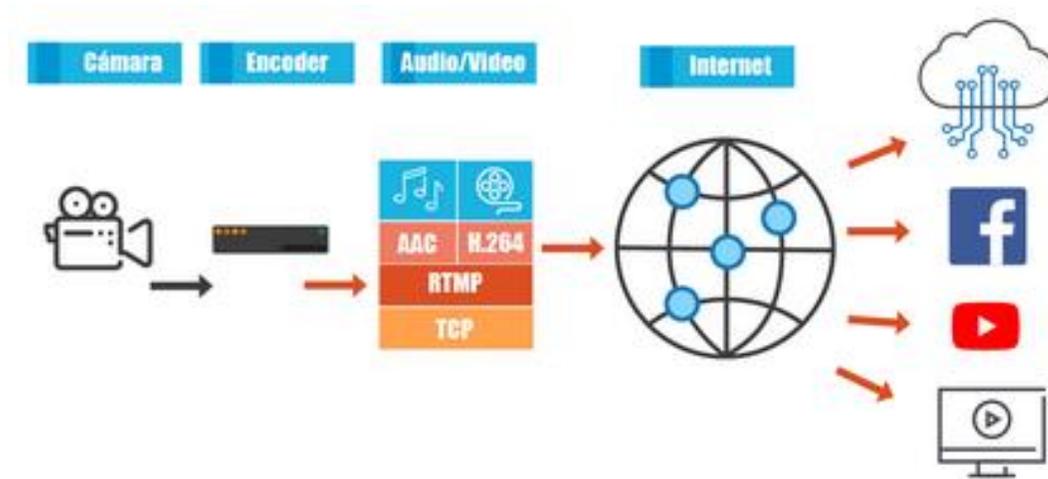


Figura 2. Video Streaming con protocolo RTMP

Fuente: (Galiano, 2014)

2.4 Protocolo de Mensajería en Tiempo Real (RTMP)

El protocolo RTMP permite transmitir audio y video, entre un codificador y una plataforma streaming. RTMP utiliza TCP para garantizar el envío de la información a través del control, orden y entrega de paquetes, se proporciona una URL con el número de puerto por ejemplo:

```
rtmp://192.168.1.144/hls
```

En este caso la URL se ha cambiado por una dirección IP, que será proporcionada por un router que establezca el enlace hacia el servidor streaming (Nginx.com).

Las plataformas de streaming reciben un bitrate en RTMP para posterior generar otros bitrates, con la finalidad de proporcionar al espectador un streaming adaptativo. Esta técnica permite codificar el video original en varias calidades, de tal forma que el cliente adapte la transmisión según su ancho de banda. TCP establece una conexión RTMP con el intercambio de 3 paquetes del servidor y otros 3 del cliente, iniciando la conexión al enviar el paquete con un valor constante de 0x03 indicando la versión del protocolo y establece la conexión entre cliente y servidor (Parmar & Thornburgh, 2012).

En la figura 3, se describe los estados de comunicación entre cliente – servidor. Este estado está sin inicializar, el cliente envía la versión del protocolo (RTMP) en el paquete c0, si es aceptada por el servidor se envía un s0 y s1 en respuesta, caso contrario el servidor responde con una acción adecuada. En el estado de envío el cliente espera el paquete s1 mientras el servidor se queda esperando al paquete c1, al recibir los paquetes esperados el cliente envía el paquete c2 y el servidor envía el paquete s2 convirtiendo al estado en Ack Sent. Servidor y cliente esperan s2 y c2 respectivamente pasando a handshake en donde cliente – servidor intercambian mensajes

(Facchini, Pérez, Hidalgo, & Cárdenas, 2018) (Ruoyu & Li, 2018), (Qifa Ke, Deng, & Baker, 2016).

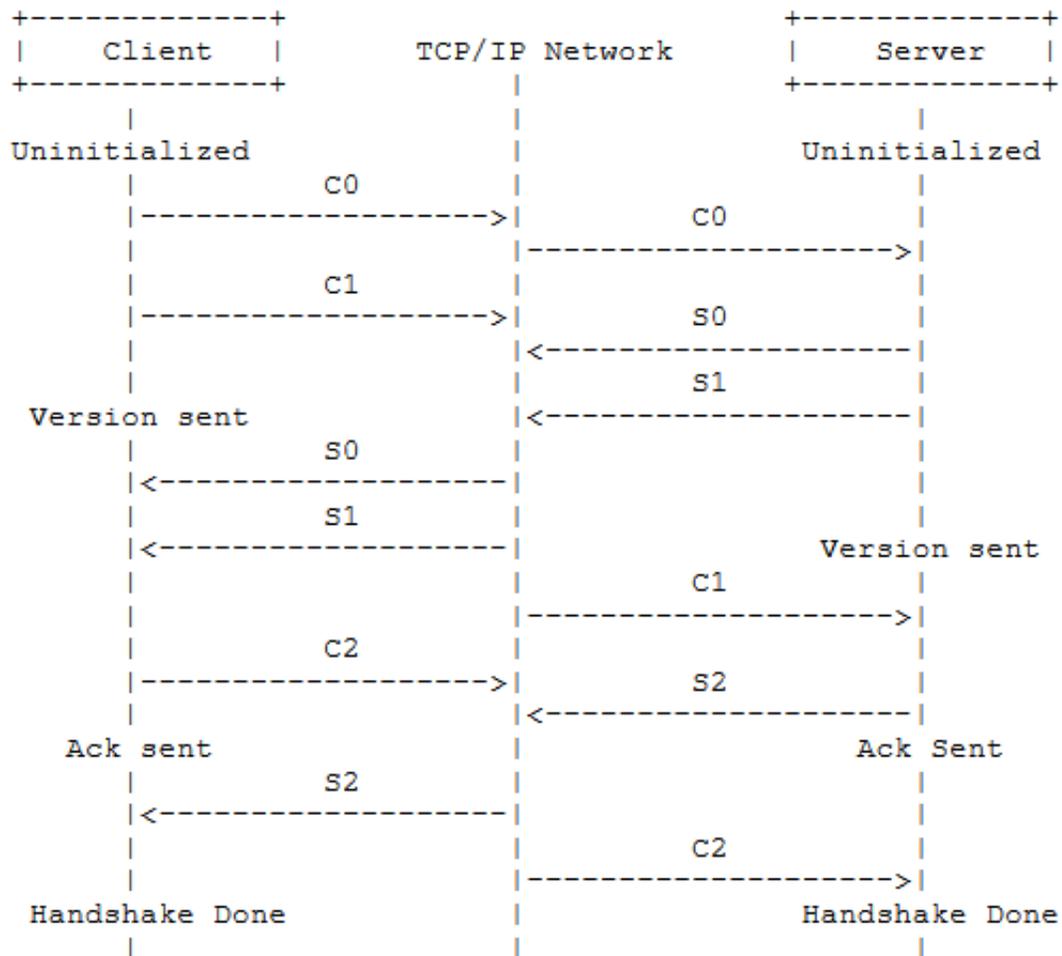


Figura 3. Representación de estados cliente servidor mediante RTMP

Fuente: (Qifa Ke, Deng, & Baker, 2016)

2.5 Servidor Web Nginx

Nginx se caracteriza por ser un servidor robusto de varias funcionalidades, ofreciendo contenido en formato MP4 o video FLV, confiable en la entrega de datos gracias a la arquitectura asíncrona basada en eventos (Nginx.com). Este servidor soporta varias conexiones simultaneas con gran

velocidad, para materializar el servidor streaming, Nginx utiliza el servicio bajo demanda (unicast) mediante el envío punto a punto, es decir el servidor dispone de un canal para cada cliente garantizando la calidad de servicio durante la transmisión (Ruoyu & Li, 2018).

A diferencia de otros servidores web, Nginx ofrece acoplamiento con el módulo streaming para una transmisión en vivo, permitiendo integrar protocolos que trabajen en modo dúplex. El servidor evalúa el rendimiento reduciendo los recursos utilizados, y permite actualizar software sin detener programas (Soni, 2016).

2.6 Visión por computadora

Una imagen puede ser interpretada desde el punto de vista de una persona y el de una computadora abarcando problemas inversos. Una imagen es un conjunto de valores captados por un sensor en un determinado punto espacial que serán representados como píxeles. La figura 4 se observa una imagen original observada por el ser humano en escala de grises y como la misma imagen es interpretada por un software (Aguilar, Casaliglla, Pólit, Abad, & Ruiz, Obstacle Avoidance for Flight Safety on Unmanned Aerial Vehicles, 2017), (Aguilar, Salcedo, Sandoval, & Cobeña, 2017).

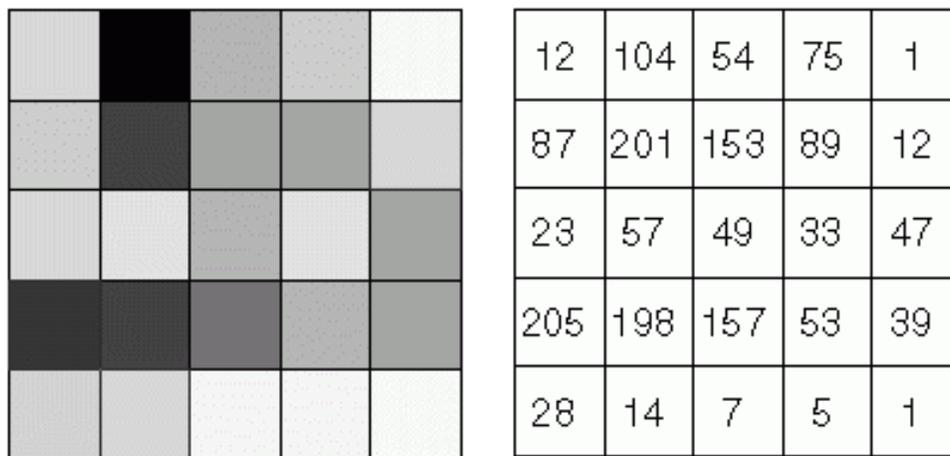


Figura 4. Representación de una imagen en píxeles

Fuente: (Pussiol, 2014)

Para iniciar el análisis de visión por computadora se necesita que la imagen representada en píxeles tenga una función que entienda el significado de la imagen numérica. Función que se represente y describa las características de objetos de interés, a esta definición se la va a conocer como características, en inglés *features*. En la actualidad existen diferentes niveles de abstracción representados por features que permiten identificar: bordes, color, ángulos, concepto de una persona o varios objetos en una imagen (Qifa Ke, Deng, & Baker, 2016) (Pussiol, 2014).

En la figura 5, se realiza la comparación de píxel a píxel sin una estructura definida. Se compara features y se obtiene una función que clasifique en categorías una imagen, para este caso dos funciones definidas con color rojo y azul.

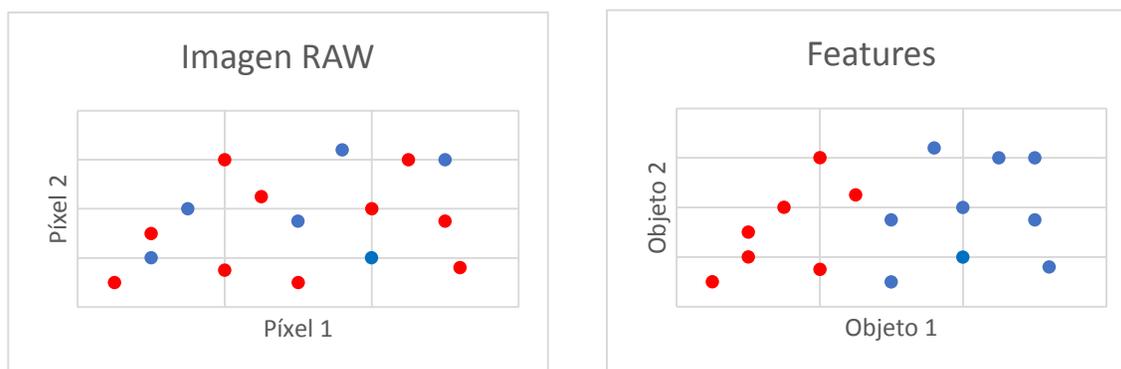


Figura 5. Comparación píxeles vs features

Fuente: Elaboración propia

La función features necesita de herramientas para ponerlas en funcionamiento, generando descriptores que puedan reconocer estas features, mediante dos enfoques: basados en técnicas superficiales y aprendizaje no supervisado (Pussiol, 2014), (Aguilar W. G., y otros, 2017).

El *enfoque superficial* es la forma tradicional de visión por computadora, genera una representación invariante a la posición, iluminación, fondo entre otras características. Mediante

técnicas estadísticas en base al objeto anterior, este enfoque procesa una imagen con extractor de features prefijas, luego esta nueva imagen pasa por un clasificador entrenado siendo dos procesos diferentes (Pussiol, 2014).

El *enfoque no supervisado* se fundamenta en un único algoritmo de aprendizaje, comprendiendo las diferentes entradas mediante técnicas, algoritmos y estructuras similares al cerebro. Tienen la capacidad de aprender features por sí mismas, pasando a ser parte del componente entrenable del detector y el clasificador es independiente a la estructura del enfoque no supervisado (Pussiol, 2014).

El presente trabajo está basado en el aprendizaje de features con enfoque no supervisado y serán soportadas por Redes Neuronales Convolucionales que se analizarán a continuación.

2.7 Redes Neuronales Convolucionales

Conocidas como ConvNets o CNN son redes neuronales de la familia deep learning y están diseñadas para aprender y descubrir features de imágenes. Las CNN incorporan operaciones no-lineales de manejo de imágenes. Las CNN a través de la historia han evolucionado a grandes pasos, en la tabla 1 se indica su origen y desarrollo hasta la actualidad (Pussiol, 2014), (Cebrián, 2015-2016), (Qifa Ke, Deng, & Baker, 2016):

Tabla 1
Origen y desarrollo de Redes Neuronales Convolucionales

AÑO	AUTOR	NOMBRE	MODELO
1980	Kunihiko Fukushima	Neocognitron, percepción visual por células en la corteza primaria de visión. S y C.	Varias células interconectadas en cascada, las features son extraídas por células S y las deformaciones por células C, se integran gradualmente y se clasifican en la capa superior de la cascada.
1998	Yann LeCun	Backpropagation, algoritmo para reconocimiento de dígitos.	Inicio de las redes neuronales mediante identificación de dígitos en documentos de texto de forma supervisada. Aprende features de la imagen.
2012	Geoff Hinton	ConvNets	Modelo de clasificación con 1000 categorías.

A continuación se realiza el modelo de trabajo de una red neuronal conectadas entre sí para transferir información. Las redes deben estar organizadas en capas formando una gráfica cíclica, es decir la salida son la entrada de las neuronas a las cuales se encuentran conectadas. La figura 6 se presenta un ejemplo conocido como redes *fully-connected*. La capa de entrada representa el vector de datos de entrada con igual número de neuronas como elementos del vector para el caso de imágenes son píxeles. En la capa oculta se encuentra un número arbitrario de capas que serán

probadas empíricamente para obtener la mejor respuesta a la interrogante planteada. Las capas ocultas convierten los datos de entrada de acuerdo a los pesos que tienen las neuronas y con la función no-lineal se procede a la extracción de características y en la capa de salida se identifica las etiquetas o clases (Cebrián, 2015-2016) (Aguilar, Manosalvas, Guillén, & Collaguazo, 2018).

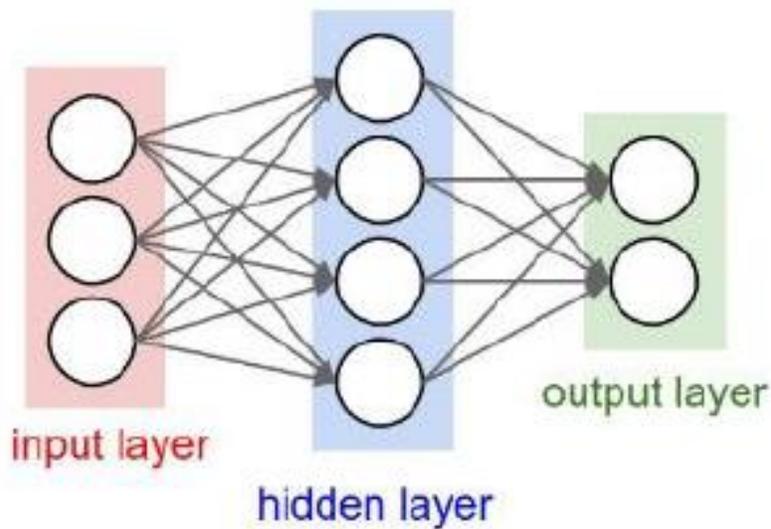


Figura 6. Red neuronal con 3 neuronas de entrada, 4 ocultas y 2 de salida

Fuente: (Cebrián, 2015-2016)

Matemáticamente las transformaciones entre capas se modelan por:

$$f_l(v) = g(W_v + b) \quad (1)$$

W es la matriz de pesos de las conexiones entre neuronas, cada fila corresponde a la capa $l-1$ en relación al resto de neuronas de la capa l , v es el vector de entrada y b es el vector oculto. La función de activación se representa por g que es una función no lineal y define el tipo de neuronas a trabajar.

En relación a la figura 6 las transformaciones que tendría el vector de datos de entrada y la capa 0 sería:

$$f(v) = f_2(f_1(v)) \quad (2)$$

Mediante extrapolación para un número de L capas:

$$f(v) = f_L(f_{L-1}(\dots f_1(v))) \quad (3)$$

Con la función sigma (σ) se transformará el valor x de entrada en uno de salida entre 0 y 1:

$$\sigma(v) = \frac{1}{(1+e^{-x})} \quad (4)$$

Pero si el valor de x es muy pequeño o muy grande se produce saturación, haciendo más lento el proceso de aprendizaje, se ha solucionado con otras funciones como la función lineal rectificadora (ReLU) mediante:

$$f(x) = \max(x, 0) \quad (5)$$

Un gradiente alto pasado por una neurona ReLU, hace que los pesos se actualicen dejando en 0 la neurona y se desactive. Se reduce la probabilidad que sea 0 ajustando el factor de aprendizaje:

$$f_j(z) = \frac{e^{z_j}}{\sum_{k=1}^K e^{f_k}} \quad (6)$$

Con esta función se designa un valor a la neurona j proporcional a la salida sumando 1, siendo la probabilidad de la entrada introducida en la red.

La anterior descripción hace referencia al modelo de una red neuronal convencional. A continuación se detalla la arquitectura de una red neuronal convolucional. Las ConvNets se componen de varias etapas entrenables en donde la entrada y salida representa un conjunto de arreglos conocidos como “*mapas de features*”. La etapa está compuesta por tres capas: capa de

convolución, capa no lineal y capa de sub-muestreo. La figura 6 se describe la arquitectura de la red LeNet-5 de Yann LeCun que cuenta con dos etapas:

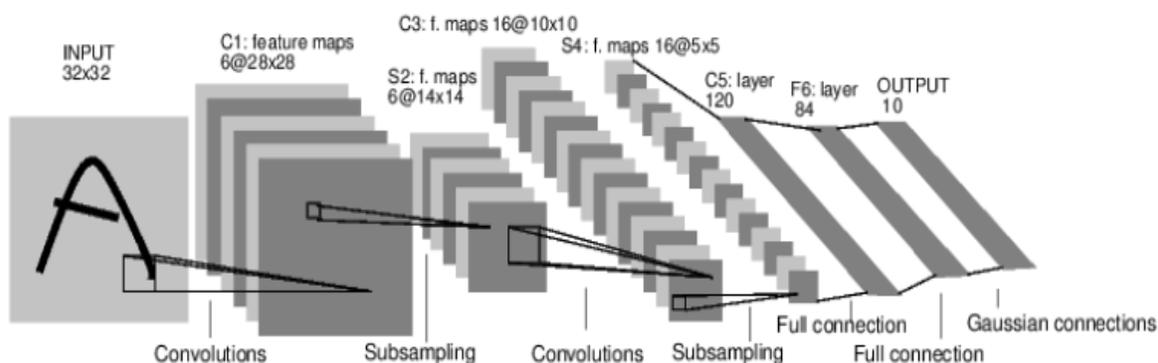


Figura 7. Red LeNet-5 arquitectura

Fuente: Aprendizaje basado en gradientes, Yann LeCun 1998 (Pussiol, 2014)

La entrada es representada por el cuadro de la letra A en píxeles, la primera etapa presenta la capa de convolución con una secuencia de características, seguidas de la capa de submuestreo (pooling). En la primera capa C1 cada uno de los 6 mapas de features, tiene pequeñas features con sus pesos agrupados. En la capa S2 se realiza el pooling agrupando las salidas de features de C1, se obtiene entonces un mapa de menor tamaño que será la entrada para la siguiente etapa en donde se encontrará features de mayor abstracción. Debido al sub-muestreo que se realiza en cada etapa se detectan características más detalladas. El sistema conectado evalúa las posibles combinaciones de las features aprendidas y a su salida se obtiene imágenes fiables para cada clase (Redmon, Divvala, Girshick, & Farhadi, 2016), (Pussiol, 2014), (Hiaolong, Jiang, & Yongjia, 2017).

De esta forma antes de iniciar el entrenamiento se debe elegir los pesos iniciales, cada neurona tiene un peso según el número de entradas utilizando una distribución uniforme:

$$W \sim U \left[-\sqrt{\frac{2}{N}}, \sqrt{\frac{2}{N}} \right] \quad (7)$$

Para el entrenamiento se debe minimizar la función de pérdida:

$$\min L(x, y, w) \quad (8)$$

En donde la función de pérdida es el error de aproximación y depende del trabajo que se realiza:

$$L(x, y, w) = E[(y - f(x, w))^2] \quad (9)$$

Para redes feed-forward, se utiliza el método del gradiente estocástico con:

$$w_t = w_{t-1} - \lambda * \frac{dL(x, y, w_{t-1})}{dw} \quad (10)$$

Donde λ representa la tasa de aprendizaje, pero a la función de pérdida se incrementa más términos esto se conoce como regularización, mejorando la capacidad de generalización:

$$L_{reg}(x, y, w) = L(x, y, w) + \alpha * |w|^2 \quad (11)$$

Finalmente el dropout optimiza la red desconectando algunas neuronas por cada entrenamiento.

2.8 Solo Se Mira Una Vez (YOLO)

YOLO es un sistema de detección de objetos en tiempo real, que utiliza deep learning y redes neuronales convolucionales. YOLO se fundamenta en ver una imagen solo una vez, mejorando el tiempo para detectar objetos con un procesamiento de imágenes de 30 FPS (fotogramas por segundo) en tiempo real.

La imagen de entrada se divide en una matriz de $S \times S$ elementos, en cada celda formada se predice n posibles cuadros delimitadores calculando su nivel de probabilidad. Se calcula $S \times S \times n$ diferentes cajas con una probabilidad muy baja en su mayoría. Con los resultados de las predicciones se elimina cajas que se encuentren bajo un determinado límite (umbral), a las cajas restantes se aplica la supresión no máxima para eliminar las detecciones duplicadas, identificando

los objetos de una determinada imagen. En la imagen 8 se puede observar el proceso que realiza el sistema de identificación de objetos YOLO, para una imagen con tres características particulares (Redmon, Divvala, Girshick, & Farhadi, 2016).

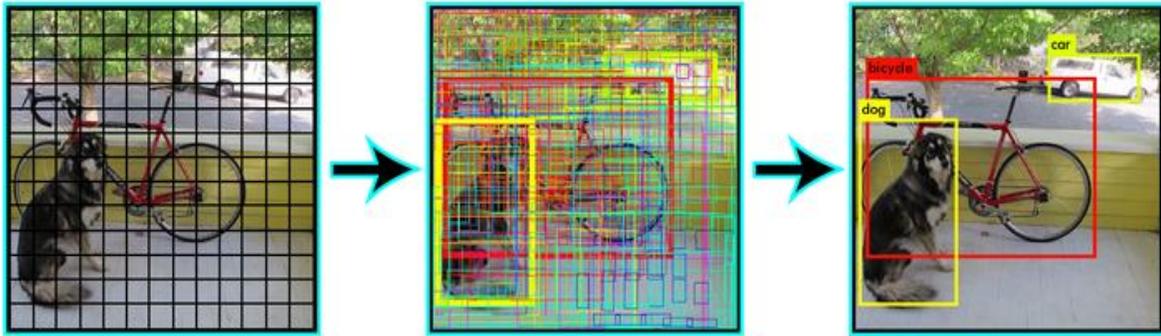


Figura 8. Estructura de funcionamiento YOLO

FUENTE: (Redmon & Farhadi, You Only Look Once, 2015)

Los cuadros delimitadores se predicen utilizando clústers de 4 coordenadas para cada uno, t_x , t_y , t_w , t_h . Para identificar la celda desplazada desde la esquina superior izquierda considerar c_x , c_y con un ancho y alto definido por p_w , p_h , de tal manera que las predicciones corresponden a:

$$b_x = \sigma(t_x) + c_x \quad (12)$$

$$b_y = \sigma(t_y) + c_y \quad (13)$$

$$b_w = p_w e^{t_w} \quad (14)$$

$$b_h = p_h e^{t_h} \quad (15)$$

En el entrenamiento se utiliza la suma de la pérdida del error al cuadrado con una puntuación de objetividad para cada recuadro delimitador. Este entrenamiento utiliza la regresión lógica que predice la puntuación de objetividad, es decir si es 1 el recuadro delimitador se solapa con un objeto verdadero antes que otro recuadro anterior. Si no es el mejor cuadro pero se superpone a un objeto por encima del umbral no se realiza la predicción. La figura 9 determina una caja de delimitación,

prioridad a la dimensión y predicción de la ubicación. Se predice el ancho y altura del cuadro como desplazamientos de los centroides del grupo. Se predice las coordenadas del centro de la caja en relación a la ubicación de la aplicación del filtro, utilizando una función σ (Redmon & Farhadi, You Only Look Once, 2015).

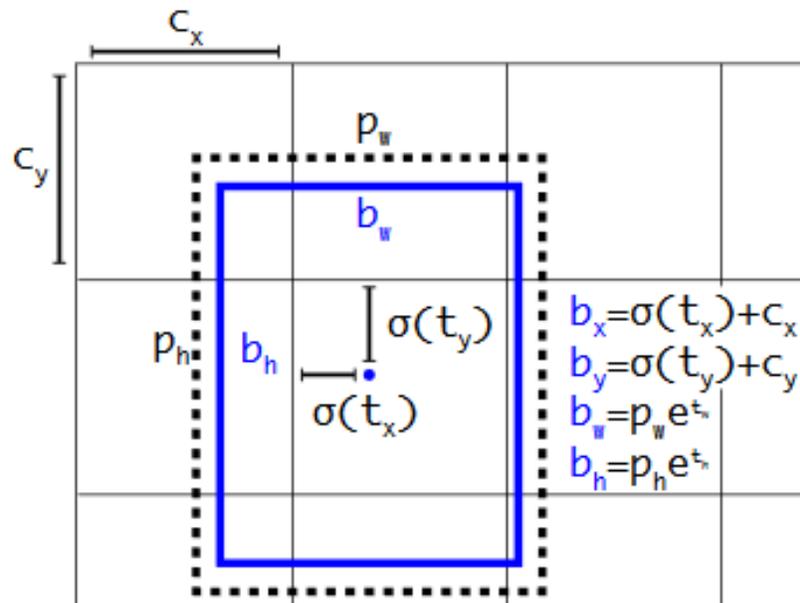


Figura 9. Cuadros delimitadores

Fuente: (Redmon & Farhadi, You Only Look Once, 2015)

Un aspecto fundamental en el manejo del sistema YOLO es la predicción a través de escalas. Maneja 3 escalas diferentes extrayendo las características de cada una, desde el extractor de características base se agrega varias capas convolucionales, la última predice un tensor 3-d que codifica cada cuadro delimitador (Redmon & Farhadi, You Only Look Once, 2015).

Una ventaja que incrementa la velocidad de procesamiento en imágenes, es que YOLO está creado en Darknet que es un marco de red neuronal de código abierto escrito en C y CUDA (Compute Unified Device Architecture). CUDA soporta cómputo con GPU (Unidad de Procesamiento Gráfico), y para la materialización del proyecto se ha utilizado Darkflow que es el

port de Darknet a Tensorflow. Darkflow es una biblioteca de código abierto para aprendizaje automático a través de un rango de tareas desarrollado por Google, para construir y entrenar redes neuronales (Tensorflow, 2018).

CAPÍTULO III

3. ANALISIS DE ESTANDARES DE COMUNICACIÓN INALÁMBRICA

3.1 Estándares de comunicaciones inalámbricas

Los sistemas de comunicaciones inalámbricos surgen ante la necesidad de enlazar dos dispositivos electrónicos, integrando un sistema de comunicación mediante software y hardware compatible, para alcanzar este propósito a lo largo de la historia se han destacado varias tecnologías inalámbricas que se detallan en la tabla 2. (Cordeiro, Akhmetov, & Park Minyoung, 2010), (Vera, Barbosa, & Pabon, 2017), (Valdez, 2016):

Tabla 2
Tecnologías Inalámbricas

TECNOLOGÍAS	PROPOSITO	CARACTERÍSTICAS	FRECUENCIA DE TRABAJO
RFID (Radio Frequency Identification)	Identificación y seguimiento de objetos mediante etiquetas, tarjetas o transpondedores RFID	Y Hasta 2 Kbytes de almacenamiento, necesita de un lector que lea los datos almacenados, necesita de varias antenas para una mejor recepción.	Frecuencias de trabajo baja, alta, ultra alta y microonda.
Bluetooth	Optimizar la comunicación entre dispositivos móviles,	Bajo consumo de energía, no necesitan de línea de vista.	ISM (Industrial Scientific and Medical) de 2,4 Ghz

CONTINÚA

	eliminando cables, Debe ser utilizado en facilitando la creación distancias cortas entre de redes pequeñas, dispositivos. mediante la sincronización de datos.
Xbee y Zigbee	Crea topología de red Bajo costo y consumo Banda libre de 2,4 jerarquica para de potencia, usa bandas Ghz conexión entre libres, fácil instalación, dispositivos mediante sus redes son flexibles y autenticación y extensibles. encriptación. Topología punto a punto o estrella.
WiFi	Acceso para varios Distancias hasta 100 m, Banda libre de 2,4 dispositivos sin gasto tasa de datos 150 Mbps, Ghz de infraestructura, estándar IEEE 802.11, compatibilidad salida a internet a través extendida. de un punto de acceso

Fuente: (Salvador, 2016), (Vera, Barbosa, & Pabon, 2017).

3.2 Implementación de la tecnología inalámbrica

El estándar IEEE 802.11 representa a la conexión inalámbrica WiFi, en la actualidad es el más utilizado para establecer redes inalámbricas entre varios usuarios, utiliza dos modos de funcionamiento: modo infraestructura, para una determinada zona de cobertura se configura un punto de acceso (AP) a manera de un dispositivo central para interconectar con todos los miembros de la red, y tener salida por el AP a una red externa. Modo ad-hoc facilita la conexión de equipos entre si formando una red punto a punto actuando como cliente y enrutador, buscando constantemente equipos disponibles materializando una malla que permite interconectar a todos los usuarios de la red (Perez, 2016), (Intriago, 2016), (Pozueco Alvarez, 2014).

La implementación del presente proyecto se basa en una red inalámbrica Wifi en modo infraestructura en donde se dispone de un router CISCO como punto de acceso, para recibir los datos proporcionados por el dron Phantom 4, mediante el protocolo RTMP hacia el servidor streaming Nginx.

Mediante la investigación cuantitativa se obtendrá mediciones del tiempo de respuesta, la velocidad de procesamiento y consumo del recurso, con herramientas de código abierto que permitirá constatar las funcionalidades y características del servidor streaming Nginx. Además se implementará el modelo FURPS (funcionalidad, facilidad de uso, confiabilidad, rendimiento, capacidad de soporte) desarrollado por Hewlett-Packard en 1987. FURPS permite establecer métricas de calidad en diseño y validación de interfaces para el cliente final (Callejas & Alarcón, 2017), para la evaluación de los factores de calidad se ha tomado en cuenta los criterios descritos en la tabla 3:

Tabla 3
Factores y criterios del modelo FURPS

FACTORES	CRITERIOS
Funcionalidad (Functionality)	Se refiere a los requisitos específicos del sistema determinando las características, funciones y seguridad del mismo.
Usabilidad (Usability)	Determina la facilidad de uso del sistema, a través del esfuerzo del usuario, se evalúa factores humanos, estética, consistencia y documentación.
Confiabilidad (Reliability)	El sistema debe ser robusto y sólido durante su ejecución, presentando recuperabilidad, precisión y predicción.
Desempeño (Performance)	Los recursos deben ser rápidos y eficientes, considerando la velocidad de procesamiento, el tiempo de respuesta y el consumo de recursos.
Capacidad de soporte (Supportability)	Se considera factores en la implementación del sistema, a través de la adaptabilidad, compatibilidad, configuración y mantenimiento.

Fuente: (Constanzo, 2014), (Callejas & Alarcón, 2017):

Para cumplir con los criterios del modelo FURPS, el proyecto se ha desarrollado bajo el estándar de codificación H.264 o MPEG-4 (Moving Picture Experts Group), definido como un códec de video de alta compresión, buena calidad de imagen, pequeñas tasas binarias y de fácil diseño. Este codec fue probado con varias tecnologías como AVC (Códec de Video Avanzado) y reconocido

actualmente como el estándar H.264/AVC, por la Organización Internacional de Estándares ISO, (Standardization, 2008).

EL protocolo HLS (HTTP Live Streaming) es utilizado en transmisión de audio y video a través de internet, compatible con varios dispositivos y firewalls, de fácil configuración. HLS divide en segmentos cortos los archivos de video, lo que permite tener una continuidad en la transferencia de datos en tiempo real, se ha configurado bajo RTMP con la finalidad de optimizar la transmisión en tiempo real, entregando el video a la plataforma creada por el router independiente hacia los usuarios que utilicen el protocolo HLS, (Galiano, 2014)

3. 3 Arquitectura de despliegue

El servidor Nginx está instalado en una computadora portátil TOSHIBA Core i7, tarjeta gráfica Nvidia GeForce con CUDA, memoria RAM de 4Gb; con sistema operativo Ubuntu 16.04 LTS. Como punto de acceso inalámbrico se ha configurado un router Cisco Linksys E4200. Este router permitirá el acceso a la red establecida entre la Aplicación DJI – router Cisco - servidor Nginx, mediante la siguiente interfaz:

- Dirección IP: 192.168.1.144
- Máscara de red: 255.255.255.0
- Puerta de enlace predeterminada: 192.168.1.1
- Seguridad router: WPA2

La configuración e instalación del servidor Nginx se realizó desde la página oficial <https://www.nginx.com/>, el servidor RTMP debe ser agregado al archivo de *nginx.conf* al final del código (Parmar & Thornburgh, 2012), en la figura 9 se describe la configuración en mención:

rtmp {	Inicia el módulo rtmp.
server {	Llama al servidor.
listen 1935;	Puerto de escucha para conexiones rtmp.
chunk_size 8192;	Tamaño de fragmento en tx de 8192 bits
application hls {	Crea la aplicación hls.
live on;	Activación del stream.
meta copy;	Información estructurada páginas web
hls on;	Activa el protocolo hls
hls_path /tmp/hls;	Direcciona el protocolo hls
}	
}	
}	

Figura 10. Configuración protocolo RTMP y HLS

Fuente: (Parmar & Thornburgh, 2012)

Para poner en funcionamiento el servidor se debe reiniciar Nginx y poner en funcionamiento mediante los siguientes comandos en el terminal de Ubuntu:

- `sudo /usr/local/nginx/sbin/nginx -s stop`
- `sudo /usr/local/nginx/sbin/nginx`

El servidor Nginx está listo para ser utilizado, en la aplicación DJI 4.0, seleccionar la plataforma de transmisión broadcast, protocolo RTMP, proporcionar la dirección URL que reconoce el servidor Nginx de la siguiente forma:

- `rtmp://192.168.1.144/hls`

RTMP es el protocolo que permitirá la transmisión en tiempo real, a continuación la dirección IP configurada en el router, y HLS proporciona continuidad en el envío de datos hacia el cliente evitando la pérdida de paquetes y reduciendo considerablemente el retardo.

El video capturado por el dron con el servidor Nginx, podrá ser reproducido en la computadora portátil por un reproductor de video que permita conexiones de red como VLC, o con un programa en python.

3.4 Resultados de las pruebas experimentales

Cuando la conexión se establece, mediante la aplicación DJI se puede controlar la información que se está transmitiendo. La figura 11 se describe que en promedio se tiene 30 FPS, este valor debe mantenerse constante para no tener pérdidas de paquetes. El *video Bitrate* representa el flujo de datos que el ordenador reproduce por cada segundo, considerando el tamaño de video de 1080x720 píxeles a una tasa de 1385 kbps que proporciona la cámara del Phantom 4 se obtiene buena calidad de video (DJI, 2016):

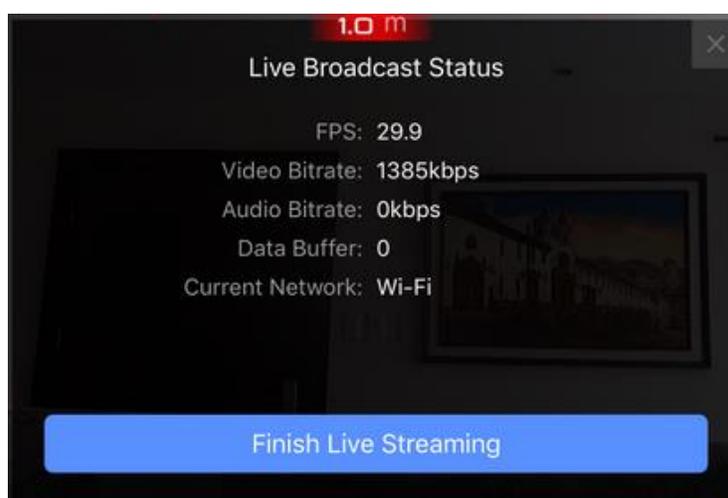


Figura 11. Estado de transmisión en vivo

Fuente: Aplicación Phantom 4 (DJI, 2016)

Mediante el comando **iftop** ejecutado en el terminal de Ubuntu se realizó el monitoreo del ancho de banda. Iftop determinar de forma continua el tráfico entre las interfaces de la red, la figura 12 proporciona información del transmisor y receptor identificados como cesar-Satellite-L750 y iPad-de-Fuerzas respectivamente, los valores de la esquina superior izquierda corresponden al tráfico de datos en intervalos de 2, 10 y 40 segundos. En promedio la transmisión de video representa un tráfico de datos de 304 kb, los datos al final de la figura representan el tráfico total de tx, rx y total en general; con un total de 36,6 MB de datos acumulados (cum) desde la ejecución y con una tasa máxima pico (peak) de 561 kb.

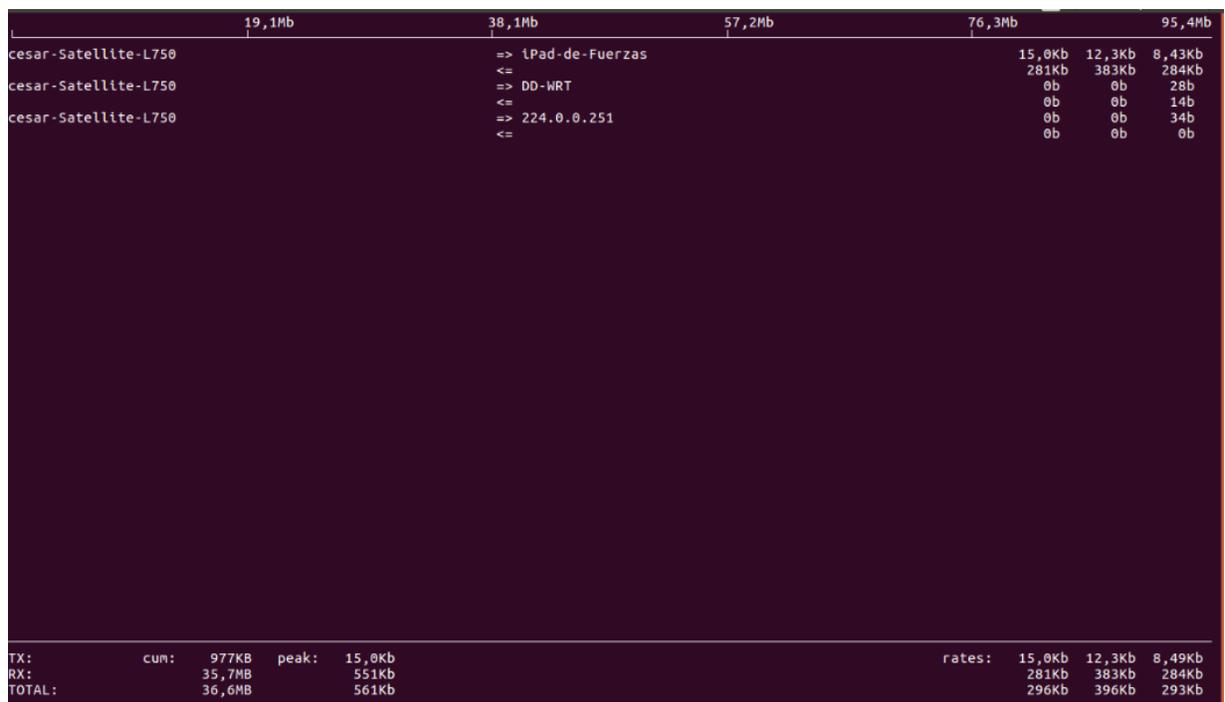


Figura 12. Datos del ancho de banda utilizado

Fuente: Sistema de video streaming implementado

Mediante el comando *vnstat* se tiene control en tiempo real del tráfico enviado y recibido, puede ser definido por un determinado tiempo, para este caso se ha tomado el tráfico del día de pruebas para la red wlp9s0, en recepción se tiene 335,70 MiB y transmisión 20,51 MiB bajo estos parámetros se estima que la red tendrá un tráfico de datos aproximado de 943 MiB en el día como se indica en la figura 13.

```
cesar@cesar-Satellite-L750:~$ vnstat
          rx      /      tx      /      total      /      estimated
enp10s0: Not enough data available yet.
wlp9s0:
  ago '18    1,89 GiB / 160,84 MiB / 2,05 GiB / 5,12 GiB
  yesterday  1,53 GiB / 136,48 MiB / 1,66 GiB
  today     335,70 MiB / 20,51 MiB / 356,21 MiB / 943 MiB
```

Figura 13. Tráfico Estimado de Red

Fuente: Sistema de video streaming implementado

Un factor de análisis es el rendimiento del CPU para lo cual se utilizó el comando `sar -u 2 10`, `sar -u` identifica al comando, tiempo 2 segundos para obtener los resultados y 10 la cantidad de muestras de acuerdo a la figura 14.

```
cesar@cesar-Satellite-L750:~$ sar -u 2 10
Linux 4.15.0-30-generic (cesar-Satellite-L750) 13/08/18      _x86_64_      (8 CPU)

09:09:59      CPU      %user      %nice      %system      %iowait      %steal      %idle
09:10:01      all       0,76       0,00       0,70       0,00       0,00       98,54
09:10:03      all       1,62       0,00       1,31       0,00       0,00       97,08
09:10:05      all       0,70       0,00       0,44       0,13       0,00       98,74
09:10:07      all       0,83       0,00       1,53       0,00       0,00       97,64
09:10:09      all       0,83       0,00       0,89       0,13       0,00       98,15
09:10:11      all       1,26       0,00       0,95       0,06       0,00       97,73
09:10:13      all       0,64       0,00       0,76       0,00       0,00       98,60
09:10:15      all       0,88       0,00       1,26       0,19       0,00       97,66
09:10:17      all       1,31       0,00       1,49       0,00       0,00       97,20
09:10:19      all       0,63       0,00       1,27       0,19       0,00       97,91
Media:        all       0,95       0,00       1,06       0,07       0,00       97,92
```

Figura 14. Rendimiento de los Recursos del CPU

Fuente: Sistema de video streaming implementado

El parámetro %user indica el porcentaje de tiempo que el procesador permanece en modo usuario, %system es el tiempo en porcentaje donde el procesador esta modo de sistema, %iowait procesador inactivo en espera de la finalización de entrada/salida y %idle procesador inactivo y no en espera de finalización.

Para la memoria RAM se utilizó el comando *htop*, que proporciona información de todos los procesos que se están ejecutando en ese momento, en la figura 15 el rendimiento de los 8 cores que posee la computadora están en condiciones normales por cuanto no sobrepasan el 20% de capacidad, a continuación se tiene el rendimiento de la memoria RAM que se encuentra a la mitad de su capacidad, el 22,4 % es el porcentaje al cual se encuentra trabajando el CPU en un periodo de tiempo de 35 minuto y 07 segundos, con estos valores el video streaming no presenta retraso ni pérdida de paquetes durante su transmisión.

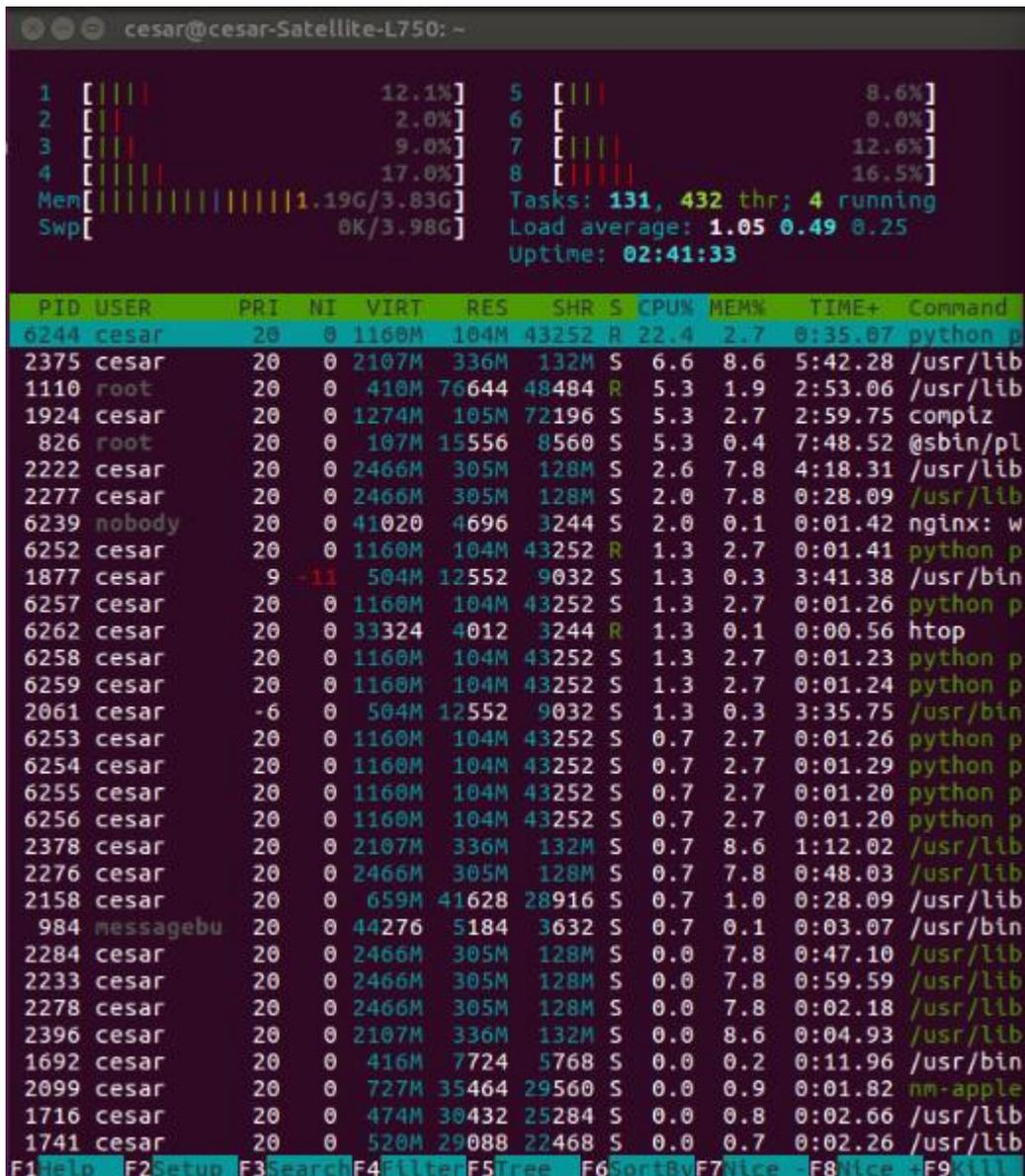


Figura 15. Rendimiento de Memoria, Cores y CPU

Fuente: Sistema de video streaming implementado

El sistema se ha implementado de forma correcta evitando perdida de paquetes o retraso en la transmisión de video según la figura 16. Se puede observar una captura de imagen durante la transmisión en tiempo real:



Figura 16. Transmisión en Tiempo Real del Servidor Nginx

Fuente: Sistema de video streaming implementado

Los datos obtenidos en el terminal de Ubuntu, ha permitido monitorear el rendimiento del ancho de banda, establecer el control de tráfico de red, el rendimiento de los recursos del CPU y la utilización de los mismos durante la transmisión entre el dron y servidor streaming, la tabla 4 establece cada uno de los valores obtenidos considerados estables para la transmisión en tiempo real:

Tabla 4
Rendimiento del Servidor Streaming Nginx

PARAMETRO	DESTINO	VALOR PROMEDIO	
Ancho de banda	Servidor Nginx – Ipad Dron Tx	11,86 kb	
	Ipad Dron – Servidor Nginx Rx	316 kb	
	Datos acumulados desde el inicio de la transmisión en Tx y Rx totales		336,6 MB
	Tasa máxima de transmisión pico Tx y Rx total		561 kb
Control de tráfico en tiempo real	Servidor – Dron Tx	20,51 MB	
	Dron – Servidor Rx	335,70 MB	
	Estimación diaria		943 MB
Rendimiento de recursos del CPU en porcentajes	Procesador en modo usuario		0,95 %
	Procesador en modo sistema		1,06 %
	Procesador inactivo en modo espera de finalización tx/rx		0,07 %
	Procesador inactivo en modo NO espera de finalización tx/rx		97,92 %
Utilización de recursos	Cores de procesamiento (8)		9,72 %
	Memoria RAN utilizada de 4 Gb		1,19 G
	Uso de CPU en una muestra de 35 minutos		22,4 %

De acuerdo a los valores obtenidos el servidor streaming funciona bajo sus mejores estándares, sin consumir mayor ancho de banda, con un control de tráfico estable para la transmisión en tiempo real, optimizando el rendimiento de los recursos del CPU garantizando de esta forma la calidad del servicio durante la transmisión dron – estación remota.

CAPITULO IV

4. DESARROLLO DEL ALGORITMO DE DETECCION Y CONTEO DE PERSONAS

4.1 Detección de objetos en tiempo real

La detección de objetos ha sido un reto para científicos desde la visión artificial (Aguilar W. G., y otros, Cascade Classifiers and Saliency Maps Based People Detection, 2017), (Aguilar W. G., y otros, Pedestrian Detection for UAVs Using Cascade Classifiers and Saliency Maps, 2017), (Aguilar W. G., y otros, 2017), (Aguilar W. G., y otros, 2017). Distinguir en una imagen los elementos que la componen, por medio de una computadora con un alto nivel de precisión, ha tenido grandes avances tecnológicos para alcanzar este objetivo. Este proceso se ha definido como clasificación de imagen, luego de analizar e investigar varios métodos de detección de objetos y enfocados en el objetivo de la detección de personas el presente trabajo se basa en entrenar una sola red neuronal. La red neuronal producen cajas delimitadoras ordenando las probabilidades simultáneamente, es decir solo se mira una vez (YOLO). Otros sistemas de detección toman la imagen dividiéndola en regiones luego se ejecuta un clasificador para cada región en donde las puntuaciones más altas se consideran detecciones. Este proceso tiene sus desventajas respecto a la velocidad de procesamiento en especial dificulta trabajar con varias escalas, es decir si se requiere que el sistema funciones con el dron a varias alturas la imagen se distorsiona y limita totalmente la identificación (Aguilar, Casaliglla, & Pólit, Obstacle Avoidance for Low-Cost UAVs, 2017), (Fernández Morcillo, 2016).

El sistema YOLO se basa en darflow que es un framework de redes neuronales, por ser de código abierto y dominio público libre se constituye en la herramienta principal de investigación para el desarrollo del presente proyecto. Las CNN generan cuadros de limitación en la imagen para

luego ejecutar el clasificador en cada cuadro (Pussiol, 2014), después se origina un nuevo procesamiento para definir los cuadros delimitadores, eliminar las detecciones duplicadas recalculando los cuadros en otros objetos de la imagen; de esta manera YOLO entrena imágenes completas optimizando el rendimiento en la detección observando toda la imagen durante el entrenamiento codificando la información contextual sobre las clases (Redmon, Divvala, Girshick, & Farhadi, 2016).

$$\Pr(Class_i|Object) * \Pr(Object) * IOU_{pred}^{truth} = \Pr(Class_i) * IOU_{pred}^{truth} \quad (16)$$

El sistema toma la imagen dividiendo en una cuadrícula de $S \times S$, cada celda predice B cuadros delimitadores generando puntajes de confianza definidos por: $\Pr(Object) * IOU_{pred}^{truth}$, será 0 si no existen objetos en la celda, también cada celda predice probabilidades de clase condicional $\Pr(Class_i|Object)$, se multiplica las probabilidades dando como resultado puntajes de confianza específicos en cada celda. Estos valores codifican la probabilidad de que la clase este en el recuadro y que tan bien se ajusta al objeto. En la figura 17 se puede observar el proceso de predicción y detección sobre una imagen predeterminada, donde una red neuronal convolucional esta evaluada en PASCAL VOC que es un conjunto de datos de detección, para lo cual las CNN iniciales extraen las características de la imagen y las caspas totales conectadas predicen las probabilidades y coordenadas de salida (Pussiol, 2014).

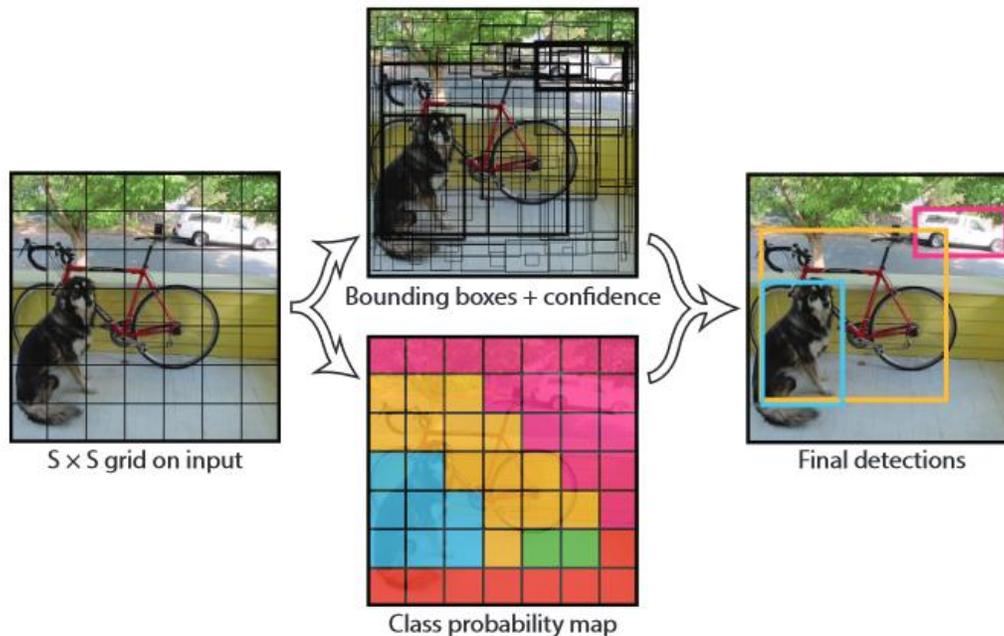


Figura 17. Proceso de confianza, cuadros delimitadores, y probabilidad de clase en una imagen

Fuente: (Redmon, Divvala, Girshick, & Farhadi, 2016)

4.2 Desarrollo y entrenamiento del sistema

El pre entrenamiento se realiza con las 20 primeras capas convolucionales, seguido de una capa de agrupación promedio y una capa conectada. El marco de Darknet se usa para el entrenamiento; como la detección requiere de visualización detallada la resolución de entrada a la red es de 608 x 608 p, donde la capa final predice las probabilidades de clase y las coordenadas del cuadro delimitador. Se normaliza la caja delimitadora entre 0 y 1, se parametriza las coordenadas x, y, para que sean desplazamientos de una posición particular, para la capa final se toma la función de activación lineal, mientras las otras capas usan la función de activación lineal rectificadas (Redmon & Farhadi, You Only Look Once, 2015) mediante:

$$\phi(x) = \begin{cases} x, & \text{si } x > 0 \\ 0, & \text{de otra manera} \end{cases} \quad (17)$$

Durante este proceso se presenta el error de suma cuadrada tanto en cajas grandes como pequeñas, la métrica de error debe reflejar la predicción de la raíz cuadrada del ancho y la altura del cuadro delimitador, al momento del entrenamiento un predictor será responsable de cada objeto, para lo cual se debe optimizar la función de pérdida de acuerdo a:

$$\begin{aligned}
& \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \\
& + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} (c_i - \hat{c}_i)^2 \\
& + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{noobj} (c_i - \hat{c}_i)^2 \\
& + \sum_{i=0}^{S^2} \mathbb{1}_i^{obj} \sum_{c \in \text{clases}} (p_i(c) - \hat{p}_i(c))^2 \tag{18}
\end{aligned}$$

Donde $\mathbb{1}_i^{obj}$ indica si el objeto se encuentra en la celda i , mientras que $\mathbb{1}_{ij}^{obj}$ indica que el factor de predicción de cuadro j -ésimo en la celda i es el responsable de esa predicción, esta función de pérdida se presenta si un objeto está en la celda de cuadrícula o al detectar un error de coordenadas en el cuadro delimitador.

4.3 TensorFlow adaptado a YOLO

El modelo YOLO está desarrollado en DarkNet pero presenta inconvenientes ya que está programado en C y no dispone de otra interfaz de programación limitando la integración en otras plataformas. Un sistema manejable, flexible y transferible a otras plataformas es tensorflow, proporciona una API para Python, C++, Java, Go entre otros lenguajes de programación compatibles (Abadi & Barham, 2016), además permite ejecutar en una GPU con solo disponer de

una tarjeta gráfica NVIDIA CUDA e instalación del software TensorFlow – GPU, incrementando el rendimiento del sistema y la agrupación para informática distribuida.

Para el sistema de detección de personas se adapta el modelo YOLO con TensorFlow, repitiendo el modelo capa por capa con un convertidor de código abierto conocido como DarkFlow. Este es un generador de redes bajo Darknet que permite construir redes TensorFlow desde archivos de configuración y cargar pesos entrenados, mediante una función en DarkFlow que permita guardar puntos de control con un meta-gráfico según la siguiente línea de programación:

```
# darkflow / net / build.py
def saveckpt ( self ):
    self .saver.save ( self .sess, ' ./ {} _model.ckpt ' .format ( self .meta [ ' name ' ]))
```

Esta importación de TensorFlow a YOLO permite crear una interfaz que toma la imagen de entrada (608 x 608 píxeles) la normalice y alimente a la red neuronal, funcionando más rápido que otros sistemas.

Los datos entrenados se encuentran en la página oficial de YOLO, entonces se importó los pesos de Darknet a TensorFlow mediante las siguientes instrucciones:

- Desde el archivo de configuración de Darknet lea la información de la capa.
- Desde el archivo ponderaciones de Darknet lea los datos entrenados de acuerdo con la definición de la capa.
- Según la capa Darknet prepare la capa TensorFlow.
- Agregar etiquetas a la nueva capa,
- Repetir el procedimiento para cada capa

En este momento el modelo YOLO divide a la imagen en varias celdas cada una de estas predice los recuadros delimitadores con puntajes de confianza y probabilidades de clase, este puntaje individual se multiplica por un mapa de probabilidad de clase, al final se obtiene un puntaje de detección de clase final.

4.4 Instalación e implementación del sistema

La demanda tecnológica se centra en la detección de objetos de forma rápida, precisa y con una flexibilidad de distinguir varias características de una imagen. YOLO posee varios modelos de entrenamiento para este trabajo se ha escogido a YOLO v2 por utilizar una combinación de un conjunto de datos y algoritmo de entrenamiento en tiempo real, permitiendo alcanzar el objetivo de detectar personas desde el dron a diferentes alturas, gracias a su escalabilidad se realizó pruebas de varias posiciones, como primer paso es tener instalado los siguientes requerimientos:

4.4.1 Anaconda distribución Python 3.6.6

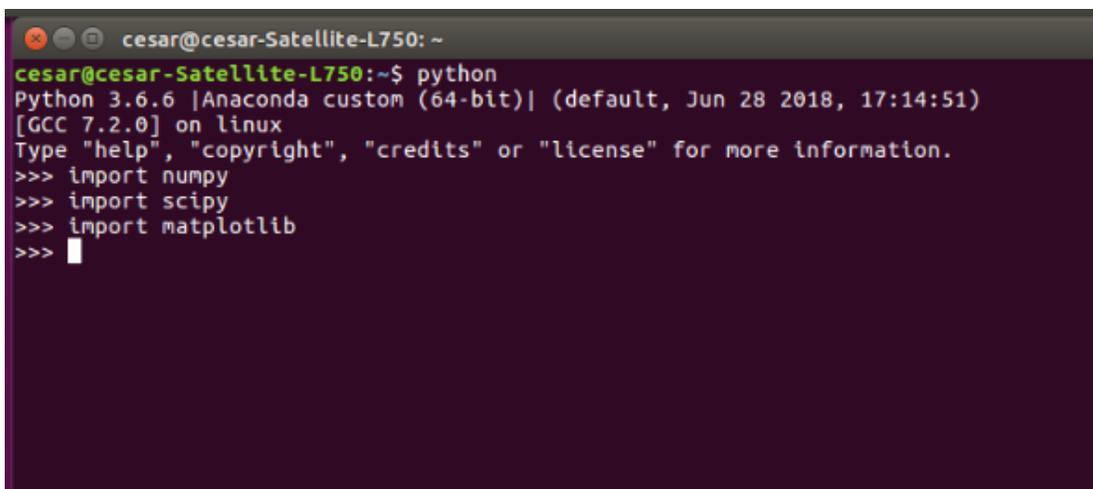
Anaconda es un administrador de paquetes y entornos de python con más de 1000 paquetes de código abierto de fácil instalación, si se requiere de un entorno gráfico de usuario está disponible la aplicación Navigator, o su vez se puede utilizar prompt de Anaconda en el terminal de Ubuntu, dispone de más de 150 paquetes que se instalan automáticamente y más de 250 paquetes de código abierto que se pueden instalar individualmente desde el repositorio de Anaconda. (Anaconda, 2018)

En el repositorio de Anaconda se encuentra la versión para Python 3.6.6 con un peso de 564 MB para un procesador de 64-Bit. Python es considerado como un lenguaje de programación de propósito general que permite almacenar y manejar datos para el análisis de acuerdo a las necesidades del usuario (Python Software, 2001 2018). Este lenguaje de programación es el más

adecuado para Anaconda por cuanto facilita el aprendizaje automático mediante el procesamiento de grandes cantidades de información, permite el análisis predictivo y ejecuta cómputos científicos.

Una vez instalado Anaconda 6 con distribución Python 3.6.6, se debe verificar los contenidos de la carpeta actual y que sus extensiones se encuentren instaladas mediante la importación, según la figura 18:

- Import numpy. Es una biblioteca matemática que proporciona soporte para vectores y matrices.
- Import scipy. Biblioteca de herramientas y algoritmos matemáticos con módulos que permiten optimizar álgebra lineal, interpolación, integración, procesamiento de señales e imágenes y otras funciones.
- Import matplotlib. Biblioteca que genera gráficos con datos obtenidos de la biblioteca Numpy.



```
cesar@cesar-Satellite-L750: ~  
cesar@cesar-Satellite-L750:~$ python  
Python 3.6.6 |Anaconda custom (64-bit)| (default, Jun 28 2018, 17:14:51)  
[GCC 7.2.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> import numpy  
>>> import scipy  
>>> import matplotlib  
>>> █
```

Figura 18. Instalación Anaconda con distribución Python e importación de librerías

Fuente: Sistema de detección de personas implementado

4.4.2 TensorFlow versión con GPU

TensorFlow es una biblioteca de código abierto direccionado al aprendizaje de máquina utilizado en investigación y producción. Una de las herramientas que más bondades presenta es la API keras de alto nivel ya que construye bloques para crear y capacitar modelos de aprendizaje profundo. Para continuar con la instalación de TensorFlow 1.4 versión GPU, se debe tener en cuenta los siguientes requisitos para su correcto funcionamiento:

- Tarjeta gráfica Nvidia GPU 650 o superior
- Anaconda con python 3.6
- CUDA ToolKit v8
- CuDNN v6

CUDA Toolkit es un entorno de desarrollo para crear aplicaciones por GPU de alto rendimiento. Es una biblioteca que permite la aceleración en varios dominios como algebra lineal, procesamiento y análisis de imágenes o video, constituye un soporte fundamental en el proceso de detección de personas, desde su página oficial se descarga e instala según el sistema operativo que se disponga según la figura 19:

Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown.

Operating System	Windows	Linux	Mac OSX
Architecture	x86_64	ppc64le	
Distribution	Fedora	OpenSUSE	RHEL
	SLES	Ubuntu	CentOS
Version	16.04	14.04	
Installer Type	runfile [local]	deb [local]	deb [network]
	cluster [local]		

Download Installers for Linux Ubuntu 16.04 x86_64

The base installer is available for download below.
There is 1 patch available. This patch requires the base installer to be installed first.

> Base Installer	Download (1.4 GB)
----------------------------	-----------------------------------

Installation Instructions:

1. Run `sudo sh cuda_8.0.61_375.26_linux.run``
2. Follow the command-line prompts

Figura 19. Plataforma de instalación CUDA Toolkit 8.0

Fuente: (NVIDIA A. C., 2017)

Una vez instalado el kit de herramientas CUDA Toolkit 8.0 se debe establecer las variables de entorno. Estas variables son un conjunto de información del sistema y usuario para indicar la ubicación de los archivos temporales, con los siguientes comandos:

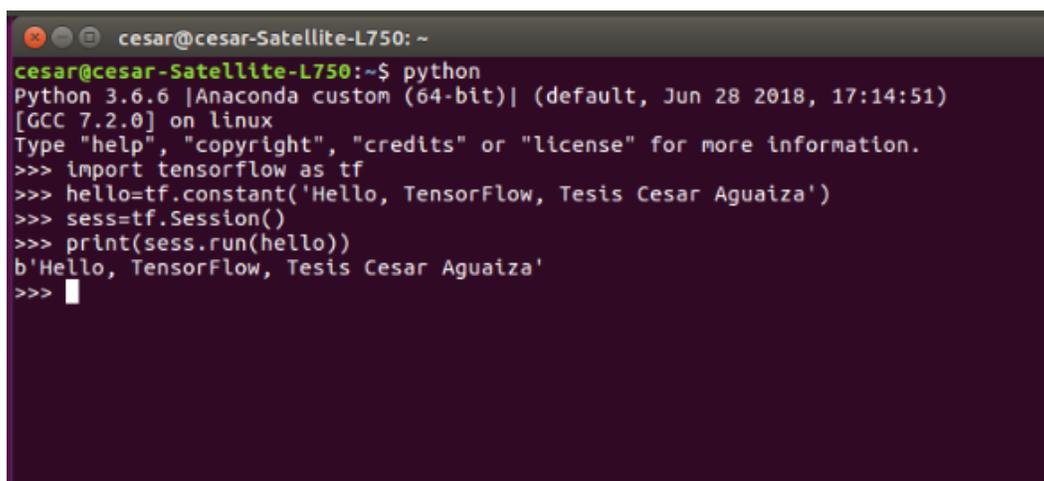
- `export LD_LIBRARY_PATH="$LD_LIBRARY_PATH:/usr/local/cuda/lib64:/usr/local/cuda/extras/CUPTI/lib64"`
- `export CUDA_HOME=/usr/local/cuda`

La biblioteca cuDNN permite trabajar con redes neuronales, proporciona implementaciones sincronizadas para convoluciones, normalización o capas de activación y es parte del SDK de Nvidia Deep Learning. CuDNN enfoca su trabajo en entrenar redes neuronales desarrollando aplicaciones de software acelerando la GPU de alto rendimiento, desde el sitio oficial se puede realizar la instalación en pocos y sencillos pasos, (NVIDIA D. , 2018).

Se debe tener en cuenta el driver de la GPU, tiene que estar instalado caso contrario se puede descargar desde el sitio oficial de NVIDIA de acuerdo a la versión que disponga la CPU, <http://www.nvidia.com/Download/index.aspx>. Una vez ejecutado los requisitos anteriores se procede a instalar TensorFlow mediante el siguiente comando:

- `pip install --ignore-installed --upgrade tensorflow-gpu`

Para comprobar que se ha instalado correctamente TensorFlow desde python importamos la biblioteca y ejecutamos un pequeño programa según el gráfico 20.



```
cesar@cesar-Satellite-L750: ~  
cesar@cesar-Satellite-L750:~$ python  
Python 3.6.6 |Anaconda custom (64-bit)| (default, Jun 28 2018, 17:14:51)  
[GCC 7.2.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> import tensorflow as tf  
>>> hello=tf.constant('Hello, TensorFlow, Tesis Cesar Aguaza')  
>>> sess=tf.Session()  
>>> print(sess.run(hello))  
b'Hello, TensorFlow, Tesis Cesar Aguaza'  
>>> █
```

Figura 20. Comprobación de TensorFlow importado a Python

Fuente: Sistema de detección de personas implementado

4.4.3 OpenCV (Open Source Computer Vision Library)

OpenCV es una librería de código abierto para visión por computadora, usada en ámbito académico como comercial, con interfaces en C++, Python o Java y compatible con varios sistemas operativos. El diseño de OpenCV está orientado a aplicaciones en tiempo real aprovechando el procesamiento de varios núcleos. Este proyecto utiliza el módulo `opencv_dnn` con `yolo_object_detection` que realiza la captura de una imagen o video en tiempo real, al ser de código abierto se utilizó el que se encuentra en el repositorio GitHub, `opencv/samples/dnn/object_detection.py` (OpenCV, 2018).

4.4.4 Repositorio DarkFlow

YOLO posee tres implementaciones, una de ellas es DarkFlow, al ser el puerto de unión entre DarkNet y TensorFlow, se ha implementado con el fin de mejorar el procesamiento en el manejo de imágenes convirtiéndose en el más utilizado por mejorar la velocidad de DarkNet incluso sin la utilización de GPU. Para instalar DarkFlow debe dirigirse al Github en donde encontrará cada una de las librerías, archivos ejecutables o dependencias que permiten ejecutar la detección de personas, mediante la descarga del directorio se podrá acceder al repositorio de DarkFlow. Luego de descargar el repositorio de DarkFlow, construir la librería y crear el entorno de trabajo con python:

- `python setup.py build_ext --inplace`

Para finalizar se debe descargar los archivos de peso de la página oficial de YOLO v2 para una entrada de imagen de 608 x 608 p, este archivo posee un modelo entrenado para detectar 82 objetos diferentes pero la investigación se basa solo en la detección de personas.

4.4.5 Procesamiento de video en tiempo real

El sistema de identificación de personas puede ejecutarse en tiempo real o mediante un archivo pregrabado, permitiendo al usuario tener un mejor análisis del reconocimiento. Para implementar esta capacidad en el sistema se debe considerar la figura 21, que representa el proceso del sistema elaborado para su ejecución en Python:

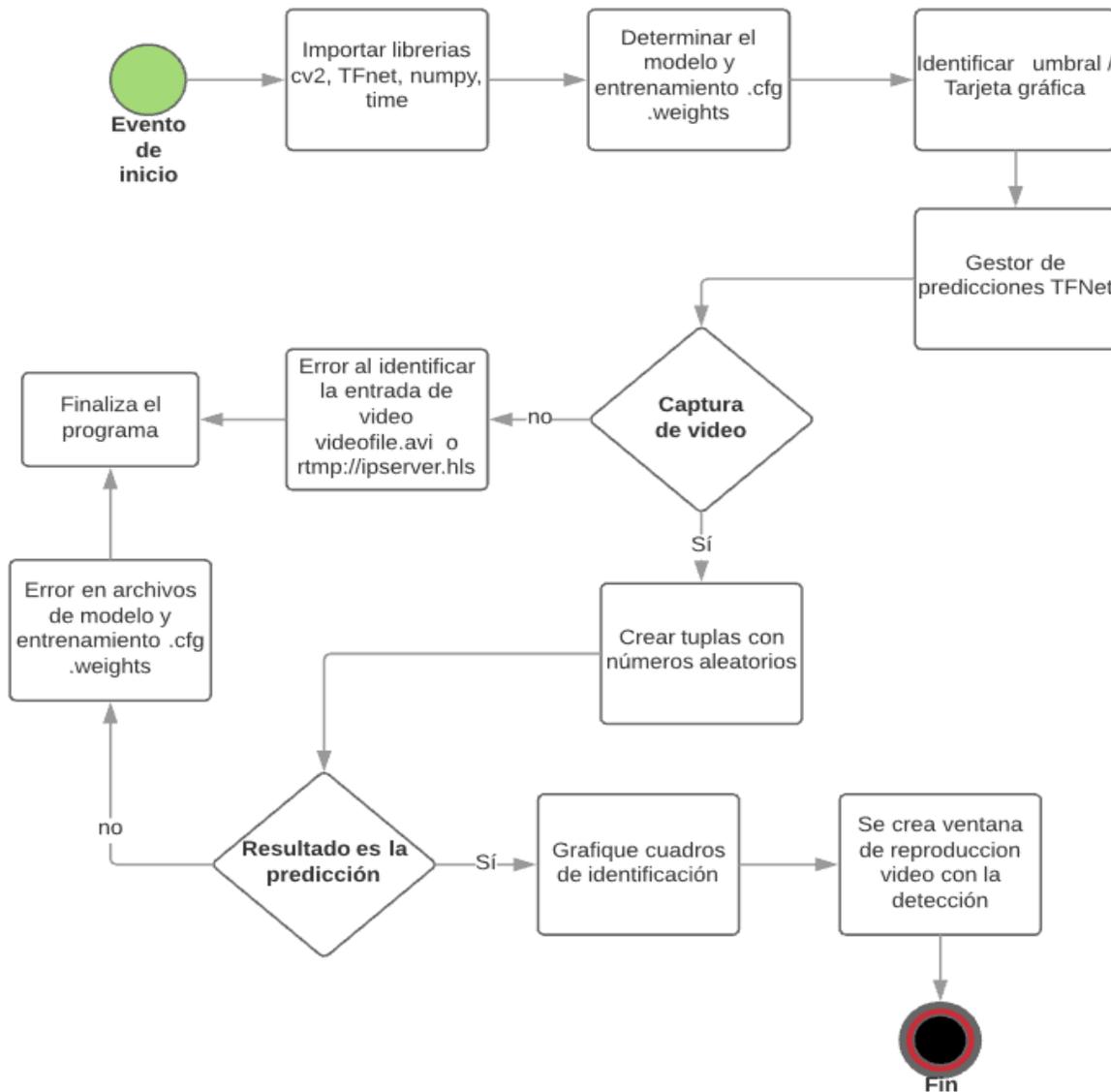


Figura 21. Diagrama de flujo del sistema de identificación de personas

Fuente: Autoría propia

CAPITULO V

5. EXPERIMENTACION, RESULTADOS Y DISCUSION DEL SISTEMA DE CONTEO DE PERSONAS

5.1 Métricas de evaluación

Las métricas son la correspondencia de un mundo real a un mundo formal matemático. Son representadas por valores numéricos asignados a un determinado ente, de esta forma las métricas a evaluar en el proyecto, están representadas por la capacidad del sistema para identificar a personas en una determinada área. Utilizan un estimador con una variable estadística que permita obtener la sensibilidad y especificidad del sistema.

La sensibilidad se define como la capacidad del sistema para identificar personas, considerando dos variables: los verdaderos positivos que son todas las personas identificadas y los falsos negativos vendrían a ser las personas que no fueron identificadas por el sistema, obteniendo la siguiente relación:

$$\text{Sensibilidad} = \frac{VP}{VP+FN} \quad (19)$$

Donde:

- VP → Verdadero positivo
- FN → Falso negativo

La especificidad se define como la capacidad del sistema para identificar objetos que no son personas. Se considera dos variables: los verdaderos negativos que son todos los objetos de una imagen que no han sido identificados como personas y los falsos negativos son objetos erróneamente identificados como personas, estableciendo la siguiente relación:

$$\text{Especificidad} = \frac{VN}{VN+FP} \quad (19)$$

Donde:

- VN → Verdaderos negativos
- FP → Falsos positivos

Para una mejor comprensión de la relación que existe entre sensibilidad y especificidad se detalla el gráfico 22:

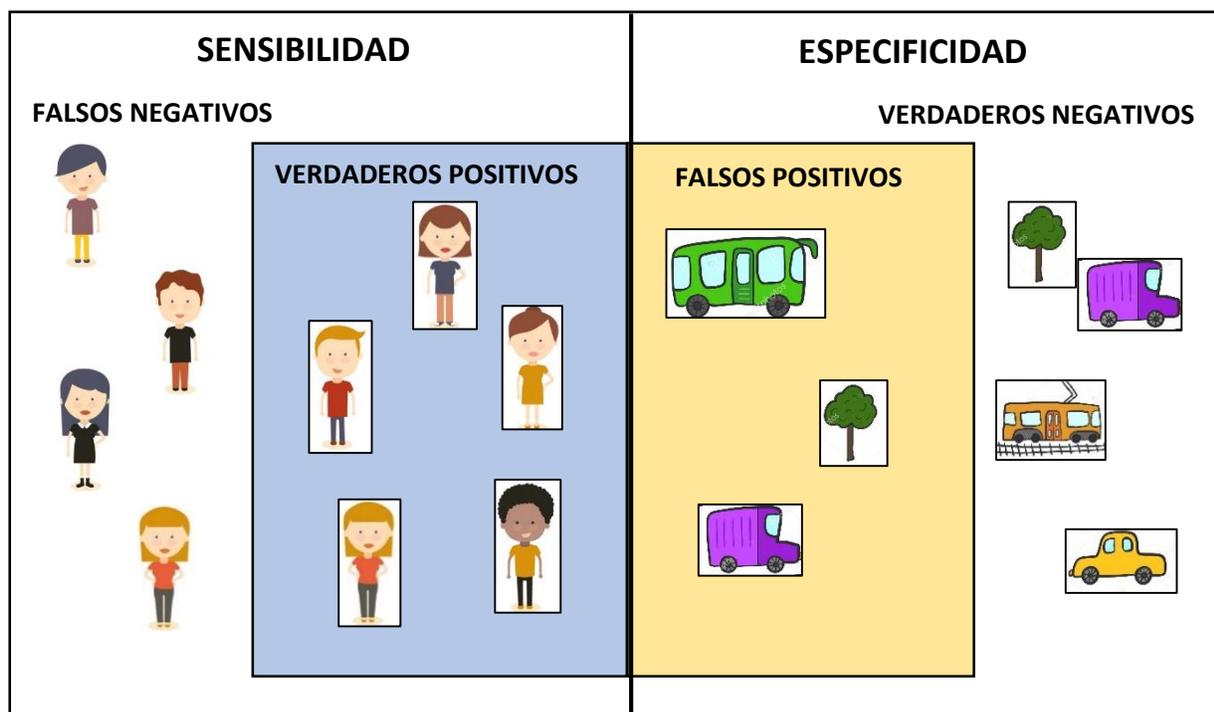


Figura 22. Relación de sensibilidad y especificidad

Fuente: Autoría propia

5.2 Diseño del experimento

Las pruebas se realizaron con el dron en movimiento, considerando diferentes variables como iluminación, sector, altura, características de la superficie y ángulo de enfoque de la cámara del

dron. Cada toma aérea fue procesada en el sistema de detección de personas, a fin de identificar cuáles son las mejores condiciones para su buen desempeño, detallando a continuación las pruebas iniciales realizadas:

1. Toma aérea a 20 metros de altura, cámara con un ángulo de 45° de inclinación, ingreso peatonal a la Universidad de las Fuerza Armadas ESPE, siendo las 13:00 horas. En la figura 23 se detecta personas y varios objetos, la inclinación de la cámara permite tener mayor detalle del perfil de una persona, pero las líneas del paso cebra tienden a confundirse con otros objetos que pueden ser detectados, además no hay mayor tráfico de transeúntes lo que limita el análisis de la eficiencia del sistema.

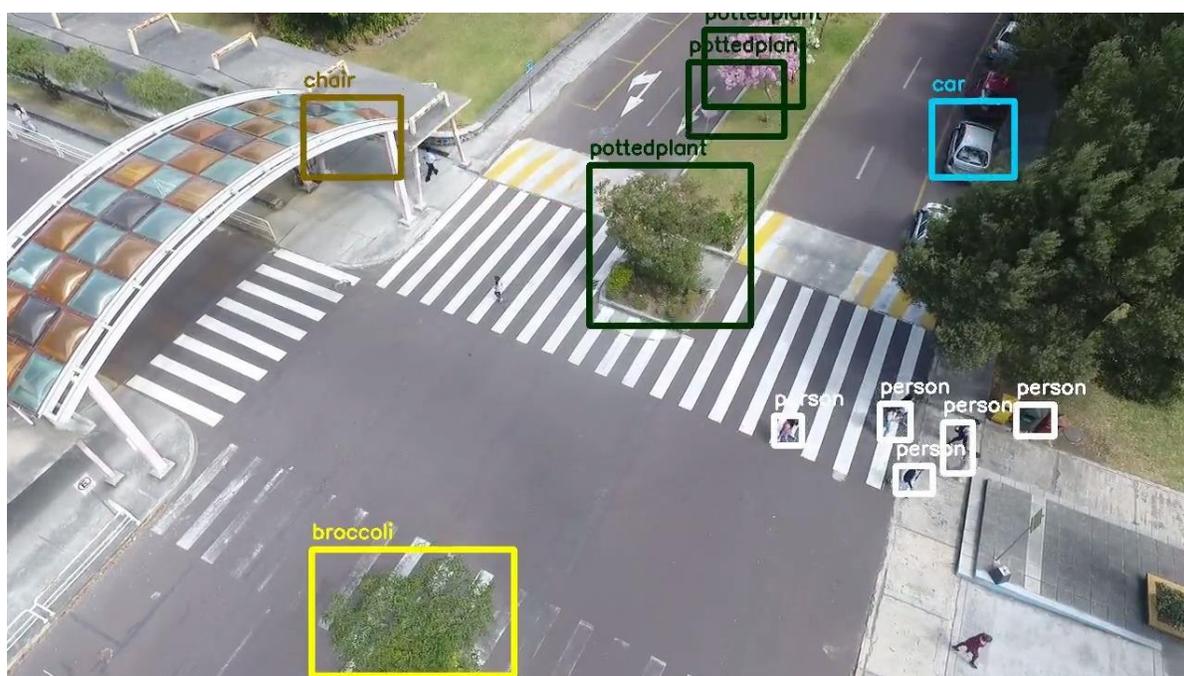


Figura 23. Sistema de detección a 20 m de altura del dron y 45° de inclinación de la cámara

Fuente: Sistema de detección de personas implementado

2. Toma aérea a 20 metros de altura, cámara con un ángulo de 90° de inclinación, ingreso por la puerta principal al Centro Comercial El Recreo, siendo las 15:00 horas. En la figura 24 se observa que el sistema reconoce apenas tres personas debido a la posición de la cámara, la figura humana no se distingue o puede ser confundida con otros objetos como un pájaro, desde esta posición el sistema no podría tener su mejor rendimiento.



Figura 24. Sistema de detección a 20 m de altura del dron y 90° de inclinación de la cámara

Fuente: Sistema de detección de personas implementado

3. Toma aérea a 20 metros de altura, cámara con un ángulo de 45° de inclinación, parque paseo escénico Sangolquí, siendo las 08:00 horas. Hasta el momento se ha comprobado que el sistema tiene un mejor rendimiento desde una ubicación alejada del objetivo con la cámara a 45° de

inclinación. La figura 25 se realizó el video siendo las 08:00 de la mañana, como se puede observar a esta hora el sol produce sombra pronunciada sobre las personas, esta particularidad hace que el sistema confunda a las personas como pájaros, reduciendo el rendimiento del sistema.

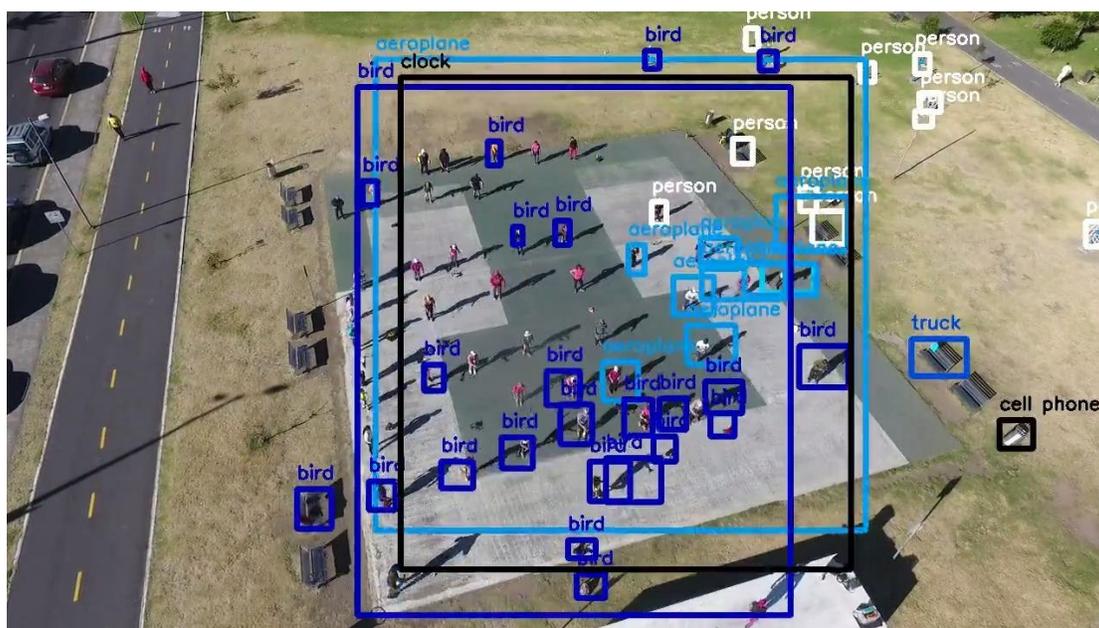


Figura 25. Sistema de detección a 20 m de altura del dron, 45° de inclinación, siendo las 08:00

Fuente: Sistema de detección de personas implementado

4. Toma aérea a 20 metros de altura, cámara con un ángulo de 45° de inclinación, parque paseo escénico Sangolquí, siendo las 17:30 horas. En la figura 26 se tomó las mismas condiciones de la figura 25, pero en diferente horario, es decir cuando el sol no produce mayor sombra sobre las personas, teniendo buenos resultados en el desempeño del sistema.



Figura 26. Sistema de detección a 20 m de altura del dron, 45° de inclinación, siendo las 17:30

Fuente: Sistema de detección de personas implementado

5. Toma aérea a 20 metros de altura, cámara con un ángulo de 45° de inclinación, parque del agua Sangolqui, siendo las 16:00 horas. En la figura 27 se ha tomado como referencia las anteriores pruebas buscando el mejor rendimiento del sistema.

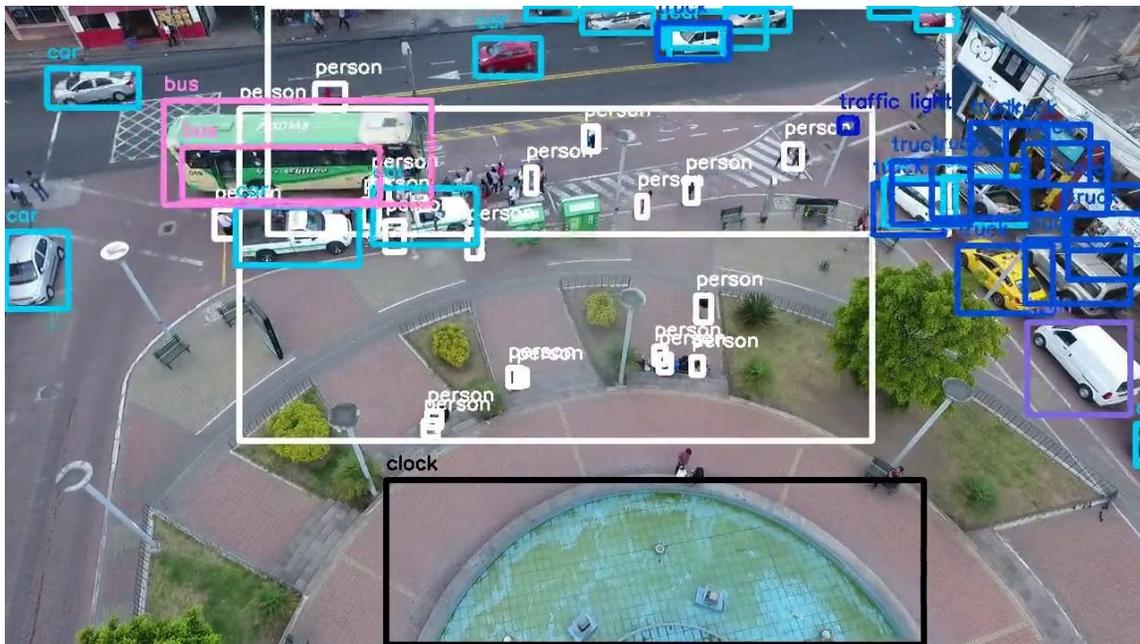


Figura 27. Sist. detec. 20 m de altura del dron, 45°, siendo las 14:00, zona congestionada

Fuente: Sistema de detección de personas implementado

De los 5 escenarios analizados y considerando que el objetivo del proyecto es la identificación de personas, se escoge las pruebas de videos realizadas en la figura 26. Esta imagen tiene mayor cantidad de personas, bajo las características de buscar una posición que permita una mejor resolución de los objetos y procurando evitar las sombras sobre personas.

Para las siguientes pruebas se tomó variables referenciando la altura del dron con respecto a la tierra y en forma vertical. Las alturas consideradas son de: 10, 20, 40, 60, 60 + zoom, 100 y 100

metros con zoom, cada video tiene una duración de 30 segundos a 30 FPS teniendo 900 FPS de los cuales se ha escogido 10 para establecer la sensibilidad y especificidad del sistema de identificación de personas. El ángulo de inclinación de la cámara varía conforme se sigue elevando el dron, para mantener en cuadro inicial partiendo desde los 45°. Se obtiene los siguientes resultados:

- Prueba 1. Altura 10 metros del dron respecto a la tierra.



Figura 28. Detección a 10m de altura del dron, ángulo cámara 45°, frame 2

Fuente: Sistema de detección de personas implementado

Tabla 5

Frames capturados para encontrar la sensibilidad y especificidad del sistema a una altura de 10 metros.

FRAMES	VP	FN	SEN	FRAMES	VN	FP	ESP
F 1	23	2	0,92	F 1	5	2	0,71
F 2	17	2	0,89	F 2	5	4	0,56
F 3	18	3	0,86	F 3	6	4	0,60
F 4	19	2	0,90	F 4	5	3	0,63
F 5	21	3	0,88	F 5	6	4	0,60
F 6	22	2	0,92	F 6	4	3	0,57
F 7	20	3	0,87	F 7	7	4	0,64
F 8	22	2	0,92	F 8	5	3	0,63
F 9	19	4	0,83	F 9	5	4	0,56
F 10	21	2	0,91	F 10	5	3	0,63
MEDIA	202	25	0,89	MEDIA	53	34	0,61

De la primera prueba se obtiene una sensibilidad del 89% y una especificidad del 61% a una altura de 10 metros, es decir el sistema funciona satisfactoriamente para la detección de personas a bajas alturas de igual forma permite distinguir otros objetos detectados.

- Prueba 2. Altura de 20 metros del dron respecto a la tierra.



Figura 29. Detección a 20m de altura del dron, ángulo cámara 50°, frame 1

Fuente: Sistema de detección de personas implementado

Tabla 6

Frames capturados para encontrar la sensibilidad y especificidad, a una altura de 20 metros.

FRAMES	VP	FN	SEN	FRAMES	VN	FP	ESP
F 1	24	1	0,96	F 1	5	3	0,63
F 2	19	2	0,90	F 2	8	5	0,62
F 3	21	2	0,91	F 3	5	4	0,56
F 4	22	3	0,88	F 4	7	5	0,58
F 5	18	3	0,86	F 5	7	5	0,58
F 6	20	3	0,87	F 6	4	5	0,44
F 7	20	3	0,87	F 7	7	4	0,64
F 8	23	2	0,92	F 8	4	3	0,57
F 9	22	2	0,92	F 9	5	5	0,50
F 10	21	2	0,91	F 10	5	3	0,63
MEDIA	210	23	0,90	MEDIA	57	42	0,58

En esta prueba el dron se encuentra a 20 metros de altura obteniendo una sensibilidad del 90% y una especificidad del 58%, en relación a la primera prueba no existe mayor diferencia el sistema continua con la identificación de personas y objetos a pesar de encontrarse a 10 metros más de altura.

- Prueba 3. Altura de 40 metros del dron respecto a la tierra.



Figura 30. Detección a 40m de altura del dron, ángulo cámara 55°, frame 5

Fuente: Sistema de detección de personas implementado

Tabla 7

Frames capturados para encontrar la sensibilidad y especificidad del sistema a una altura de 40 metros.

FRAMES	VP	FN	SEN	FRAMES	VN	FP	ESP
F 1	10	11	0,48	F 1	11	3	0,79
F 2	11	12	0,48	F 2	12	5	0,71
F 3	9	10	0,47	F 3	9	4	0,69
F 4	13	11	0,54	F 4	10	5	0,67
F 5	12	9	0,57	F 5	11	4	0,73
F 6	13	8	0,62	F 6	9	5	0,64
F 7	12	10	0,55	F 7	10	3	0,77
F 8	14	11	0,56	F 8	8	4	0,67
F 9	13	12	0,52	F 9	9	3	0,75
F 10	14	12	0,54	F 10	11	3	0,79
MEDIA	121	106	0,53	MEDIA	100	39	0,72

En la tercera prueba se obtiene una sensibilidad del 53% y una especificidad del 72% a una altura de 40 metros, debido al incremento de la elevación el sistema ha reducido su capacidad de distinguir a personas mediante la sensibilidad, pero aumentó su capacidad para detectar objetos.

- Prueba 4. Altura de 60 metros del dron respecto a la tierra.

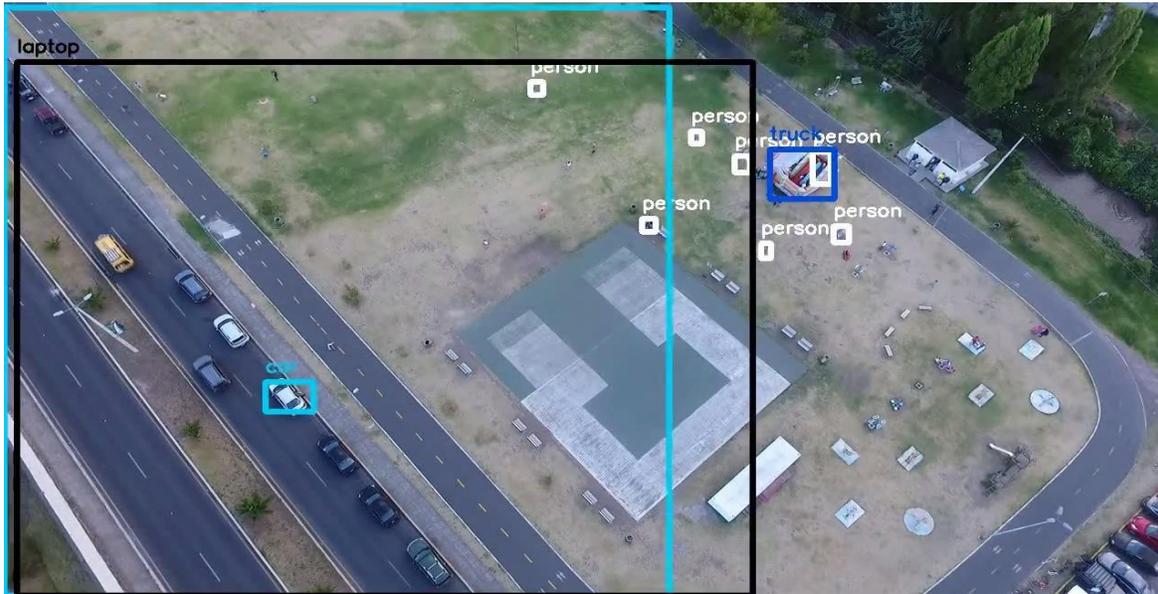


Figura 31. Detección a 60m de altura del dron, ángulo cámara 60°, frame 7.

Fuente: Sistema de detección de personas implementado

Tabla 8

Frames capturados para encontrar la sensibilidad y especificidad a una altura de 60 metros.

FRAMES	VP	FN	SEN	FRAMES	VN	FP	ESP
F 1	6	17	0,26	F 1	4	1	0,80
F 2	6	16	0,27	F 2	6	2	0,75
F 3	5	18	0,22	F 3	6	2	0,75
F 4	6	17	0,26	F 4	5	3	0,63
F 5	5	19	0,21	F 5	5	2	0,71
F 6	2	21	0,09	F 6	6	1	0,86
F 7	6	19	0,24	F 7	5	1	0,83
F 8	4	20	0,17	F 8	7	2	0,78
F 9	5	18	0,22	F 9	5	1	0,83
F 10	4	19	0,17	F 10	5	1	0,83
MEDIA	49	184	0,21	MEDIA	54	16	0,77

Para esta prueba la sensibilidad se ha reducido a un 21% y la especificidad ha incrementado al 77% a una altura de 40 metros, es decir mientras va incrementando la elevación, el sistema tiende a disminuir su eficiencia para detectar a personas, mientras que la especificidad para reconocer objetos se incrementa.

- Prueba 5. Altura de 60 metros con zoom del dron respecto a la tierra.



Figura 32. Detección a 60m de altura del dron con zoom, ángulo cámara 65°, frame 10

Fuente: Sistema de detección de personas implementado

Tabla 9

Frames capturados para encontrar la sensibilidad y especificidad del sistema a una altura de 60 metros con zoom

FRAMES	VP	FN	SEN	FRAMES	VN	FP	ESP
F 1	9	7	0,56	F 1	6	3	0,67
F 2	13	4	0,76	F 2	4	4	0,50
F 3	14	5	0,74	F 3	1	4	0,20
F 4	13	6	0,68	F 4	3	5	0,38
F 5	15	4	0,79	F 5	2	5	0,29
F 6	14	3	0,82	F 6	2	4	0,33
F 7	16	6	0,73	F 7	3	3	0,50
F 8	17	3	0,85	F 8	2	4	0,33
F 9	13	4	0,76	F 9	6	4	0,60
F 10	17	7	0,71	F 10	4	3	0,57
MEDIA	141	49	0,74	MEDIA	33	39	0,46

En esta prueba se aplicó la capacidad que dispone la cámara del dron para realizar un acercamiento en este caso el zoom realizado es de 2x optimizando la detección del sistema en cuanto a la sensibilidad pero la especificidad es relativamente baja.

- Prueba 6. Altura de 100 metros del dron respecto a la tierra.

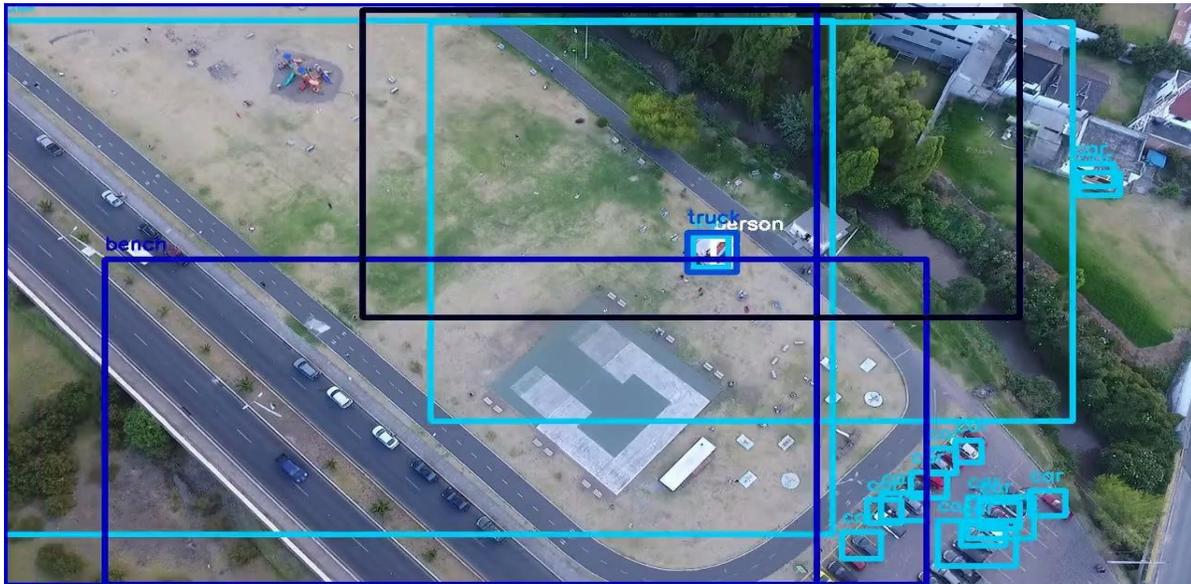


Figura 33. Detección a 100m de altura del dron, ángulo cámara 70°, frame 4

Fuente: Sistema de detección de personas implementado

Tabla 10

Frames capturados para encontrar la sensibilidad y especificidad a una altura de 100 metros.

FRAMES	VP	FN	SEN	FRAMES	VN	FP	ESP
F 1	0	20	0,00	F 1	15	0	1,00
F 2	0	20	0,00	F 2	16	0	1,00
F 3	0	20	0,00	F 3	15	0	1,00
F 4	0	20	0,00	F 4	17	1	0,94
F 5	0	20	0,00	F 5	18	2	0,90
F 6	0	20	0,00	F 6	16	0	1,00
F 7	0	20	0,00	F 7	15	0	1,00
F 8	0	20	0,00	F 8	17	0	1,00
F 9	0	20	0,00	F 9	15	0	1,00
F 10	0	20	0,00	F 10	18	0	1,00
MEDIA	0	200	0,00	MEDIA	162	3	0,98

En esta prueba la sensibilidad del sistema para detectar personas se encuentra al 0% es decir a esta altura no detecta a personas pero la especificidad se incrementó al 98% lo cual demuestra que es un valor alejado de la realidad por cuanto confunde los objetos detectados con los reales de la imagen.

- Prueba 7. Altura de 100 metros con zoom del dron respecto a la tierra.



Figura 34. Detección a 100 m de altura del dron con zoon, ángulo cámara 75°, frame 8

Fuente: Sistema de detección de personas implementado

Tabla 11

Frames capturados para encontrar la sensibilidad y especificidad del sistema a una altura de 100 metros con zoom

FRAMES	VP	FN	SEN	FRAMES	VN	FP	ESP
F 1	7	18	0,28	F 1	4	3	0,57
F 2	7	17	0,29	F 2	3	3	0,50
F 3	6	16	0,27	F 3	4	2	0,67
F 4	5	15	0,25	F 4	8	2	0,80
F 5	9	12	0,43	F 5	11	4	0,73
F 6	10	13	0,43	F 6	10	3	0,77
F 7	12	14	0,46	F 7	12	4	0,75
F 8	13	12	0,52	F 8	10	3	0,77
F 9	12	13	0,48	F 9	8	5	0,62
F 10	12	10	0,55	F 10	9	4	0,69
MEDIA	93	140	0,40	MEDIA	79	33	0,71

Para esta última prueba se elevó el dron a los 100 metros pero se ha aplicado un zoom de 2x lo que ha permitido incrementar un 40% la sensibilidad del sistema al detectar personas, si bien el valor es bajo pero considerando la altura se puede determinar que es óptimo el sistema.

5.3 Análisis de resultados

Como punto de partida se debe considerar las variables iniciales, buscando el mejor rendimiento del sistema de detección de personas, para lo cual se ha aplicado en un lugar abierto con varios objetos y transeúntes, en donde las condiciones climáticas permitan obtener una imagen sin sombras, a una distancia horizontal de 5 metros en relación al objetivo y con una inclinación inicial

de 45° de la cámara considerados desde la posición paralela 0° hacia abajo según se detalla en la figura 35:

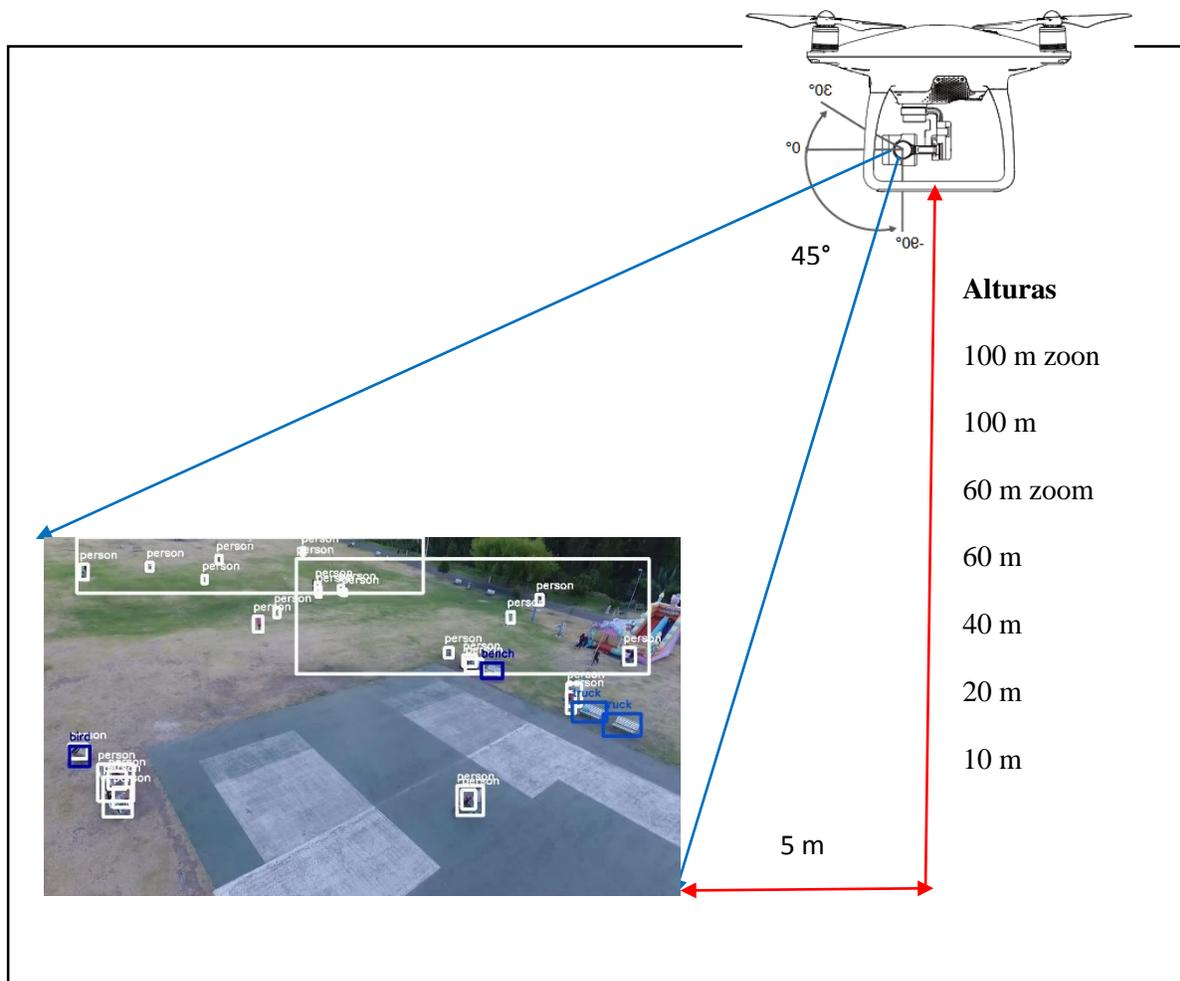


Figura 35. Diagrama posicional del dron en referencia al objetivo

El sistema de video del dron Phantom 4 permite ser configurado en varios tamaños y fps, para una mejor calidad se configuró con un tamaño de 1280×720 píxeles, a 30 fps, con una tasa de bits de 60 Mbps. Estas características permiten utilizar el zoom digital que posee la cámara con una

distancia focal de 35 mm reduciendo el ángulo de visión e incrementando el aumento de la imagen a ser analizada.

Para el cálculo de la sensibilidad y especificidad se consideraron variables en alturas de 10, 20, 40, 60 y 100 metros, tomando como muestra a 10 frames de una total de 900 frames por cada prueba, en videos de 30 segundos, obteniendo los resultados de la tabla 12:

Tabla 12
Resultados de las pruebas en sensibilidad y especificidad

PRUEBAS	ALTURA DEL DRON	ANGULO CAMARA	SENSIBILIDAD	ESPECIFICIDAD
1	10 m	45°	89%	61%
2	20 m	50°	90%	58%
3	40 m	55°	53%	72%
4	60 m	60°	21%	77%
5	60 m zoom	65°	74%	46%
6	100 m	70°	0%	98%
7	100 m zoom	75°	40%	71%

De los datos obtenidos durante las siete pruebas, se ha podido comprobar el rendimiento del sistema de detección de personas, para lo cual la sensibilidad a 10 metros es alta, al igual que la especificidad. Conforme sigue elevándose el dron estos valores varían por cuanto la distancia focal disminuye, el ángulo de visión se incrementa, disminuyendo el aumento, es decir los detalles de la imagen serán cada vez más pequeños, esto se corrigió en parte con las pruebas 5 y 7 que se utilizó el zoom digital de la cámara incrementando la distancia focal.

Finalmente el sistema está en funcionamiento óptimo para distancias de 20 metros de altura alcanzando los 100 metros con zoon digital para obtener buenos resultados. En la figura 36 se puede observar el rendimiento del sistema durante las siete pruebas:

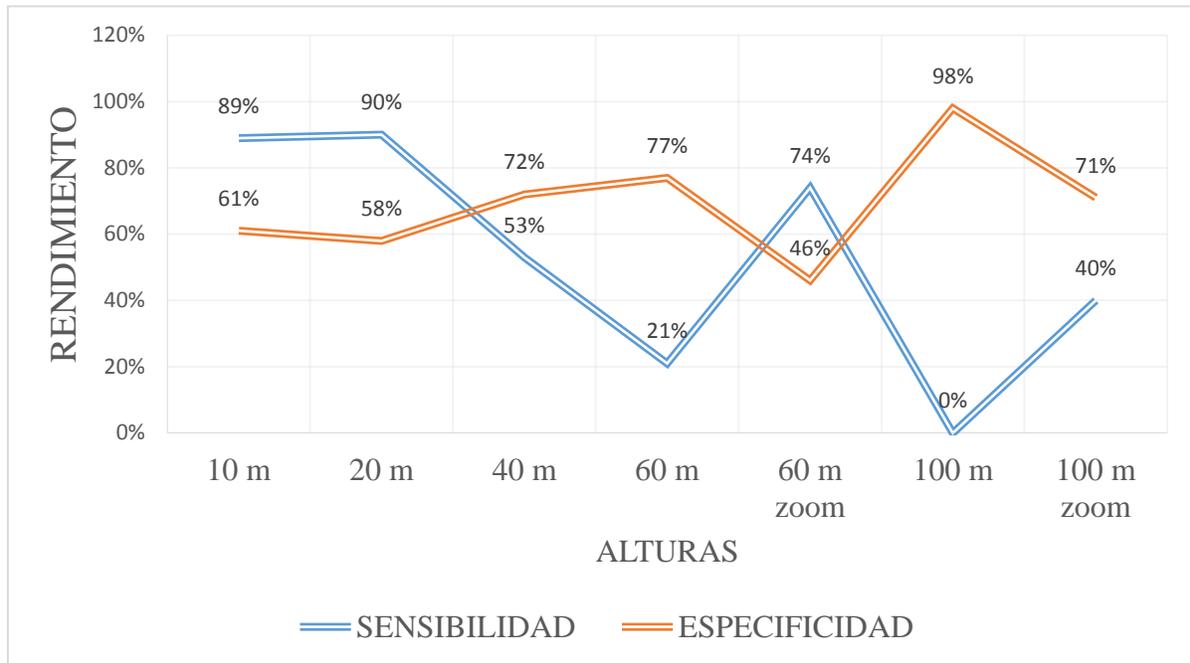


Figura 36. Comportamiento del Sistema a diferentes alturas

CAPITULO VI.

6. CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

El sistema de estimación del número de personas en tiempo real se ha implementado de manera óptima, estableciendo inicialmente la comunicación entre el dron y la estación remota, mediante la plataforma de video streaming Nginx. Esta plataforma es de código abierto utilizando la comunicación punto a punto bajo demanda, y la interacción de protocolos de comunicación como RTMP y HLS, garantizando de esta forma la calidad de servicio durante la transmisión del video.

La implementación del video streaming ha sido optimizada con la configuración de un router que direcciona el tráfico de datos desde la tablet hacia la computadora portátil en donde se encuentra el servidor streaming. La sincronización de los protocolos de comunicación RTMP y HLS evitan el retardo y la pérdida de paquetes de datos durante la transmisión de video del dron. Se consideró las características de la cámara del dron Phantom 4, enviando imágenes a 30 fotogramas por segundo, con un tamaño de 1280 x 720 píxeles a una tasa de bits de 60 Mbps.

Los sistemas de detección tradicionales reutilizan los clasificadores, a la imagen se aplica un modelo en varias ubicaciones y escalas donde las puntuaciones más altas destacan como detecciones. El presente trabajo aplica una sola red neuronal a la imagen, dividiéndola en regiones, prediciendo cuadros de límite y probabilidades para cada región donde los cuadros delimitadores son ponderados por las probabilidades a priori. Las predicciones se basan en el contexto global de la imagen, así se optimiza la velocidad de procesamiento de las imágenes durante la detección,

además se puede trabajar con tarjetas gráficas GPU para seguir mejorando el rendimiento del sistema.

Las pruebas de funcionamiento del sistema se realizaron en diferentes ambientes considerando variables como altura, iluminación y distancia focal. Se obtuvo una sensibilidad alta para detectar personas en alturas bajas. Las misiones de reconocimiento del Ejército Ecuatoriano deben estar sobre los 100 metros de altitud del dron respecto a la tierra, entonces se optimizó el sistema mediante el manejo de las características técnicas que posee la cámara al incrementar la distancia focal de 2x con el zoom digital. Se obtuvo valores de especificidad para detectar objetos que no sean personas mediante la relación de falsos positivos y verdaderos negativos con buenos resultados.

6.2 RECOMENDACIONES

Durante la investigación, para obtener el mejor rendimiento en la detección de objetos se encontró que los nuevos sistemas se basan en redes neuronales. Razón por la cual es recomendable continuar con estos estudios basados en la predicción de cuadros de límite y probabilidades.

La transmisión de video streaming debe estar soportado por hardware confiable que garantice la entrega de datos, proporcione seguridad a la comunicación dron – estación remota y asegure la calidad de servicio, evitando la pérdida de paquetes o el retardo en la comunicación entre estaciones.

Las variables utilizadas para evaluar el desempeño, son de vital importancia ya que de esto depende el rendimiento del sistema de identificación de personas. Se debe continuar con pruebas

en misiones reales de reconocimiento, para que sea confiable la entrega de información y el mando pueda tomar decisiones acertadas y oportunas.

La visión por computadora es un tema actual y extenso, requiere de tiempo y constancia para escoger el modelo que más se ajuste a las necesidades de la investigación. El sistema YOLO al ser de código abierto permite adaptar varios lenguajes de programación. Se recomienda para trabajos futuros se pueda diseñar clases individuales para el reconocimiento específico de algún objeto, que sea de interés para el cumplimiento de las misiones del Ejército Ecuatoriano.

El sistema propuesto se encuentra en funcionamiento, las pruebas se han desarrollado en ambientes urbanos, se debe continuar con el desarrollo del proyecto para optimizar el algoritmo y pueda ser utilizado en diferentes terrenos como la frontera norte.

REFERENCIAS BIBLIOGRÁFICAS

Bibliografía

- Abadi, M., & Barham, P. C. (2016). TensorFlow: A System for Large-Scale Machine Learning. *The Advances Computing Systems Association*, 21.
- Abdullah, M. (2018). *Smart Client-Server Protocol and Architecture for Adaptive Multimedia Streaming*. Valencia: Universidad Politécnica de Valencia.
- Adobe Developer, C. (2018). *Adobe*. Obtenido de <https://www.adobe.com/devnet/rtmp.html>
- Aguilar, W. G., & Angulo, C. (2014). Real-time video stabilization without phantom movements for micro aerial vehicles. *EURASIP Journal on Image and Video Processing*, 1, 1-13.
- Aguilar, W. G., & Angulo, C. (2014). Robust video stabilization based on motion intention for low-cost micro aerial vehicles. *11th International Multi-Conference on Systems, Signals & Devices (SSD)*. Barcelona, Spain.
- Aguilar, W. G., & Angulo, C. (2016). Real-Time Model-Based Video Stabilization for Microaerial Vehicles. *Neural Processing Letters*, 43(2), 459-477.
- Aguilar, W. G., & Morales, S. (2016). 3D Environment Mapping Using the Kinect V2 and Path Planning Based on RRT Algorithms. *Electronics*, 5(4), 70.
- Aguilar, W. G., Abad, V., Ruiz, H., Aguilar, J., & Aguilar-Castillo, F. (2017). RRT-Based Path Planning for Virtual Bronchoscopy Simulator. En *Lecture Notes in Computer Science* (págs. 155-165).
- Aguilar, W. G., Angulo, C., & Costa-Castello, R. (2017). Autonomous Navigation Control for Quadrotors in Trajectories Tracking. En *Lecture Notes in Computer Science* (págs. 287-297).
- Aguilar, W. G., Angulo, C., & Pardo, J. A. (2017). Motion intention optimization for multirotor robust video stabilization. *Chilean Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*. Pucón, Chile.
- Aguilar, W. G., Casaliglla, V. P., & Pólit, J. L. (2017). Obstacle Avoidance Based-Visual Navigation for Micro Aerial Vehicles. *Electronics*, 6(1), 10.
- Aguilar, W. G., Casaliglla, V. P., & Pólit, J. L. (2017). Obstacle Avoidance for Low-Cost UAVs. *IEEE 11th International Conference on Semantic Computing (ICSC)*. San Diego.

- Aguilar, W. G., Casaliglla, V. P., Pólit, J. L., Abad, V., & Ruiz, H. (2017). Obstacle Avoidance for Flight Safety on Unmanned Aerial Vehicles. En *Lecture Notes in Computer Science* (págs. 575-584).
- Aguilar, W. G., Cobeña, B., Rodriguez, G., Salcedo, V. S., & Collaguazo, B. (2018). SVM and RGB-D Sensor Based Gesture Recognition for UAV Control. *International Conference on Augmented Reality, Virtual Reality and Computer Graphics* (págs. 713-719). Springer.
- Aguilar, W. G., Luna, M. A., Moya, J. F., Abad, V., Ruiz, H., & Parra, H. (2017). *Real Time Detection and Simulation of Abdominal Crowd Behavior*. Lecture Notes in Computer Science.
- Aguilar, W. G., Luna, M. A., Moya, J. F., Abad, V., Ruiz, H., Parra, H., & Angulo, C. (2017). Pedestrian Detection for UAVs Using Cascade Classifiers and Saliency Maps. En *Lecture Notes in Computer Science* (págs. 563-574).
- Aguilar, W. G., Luna, M. A., Moya, J. F., Abad, V., Ruiz, H., Parra, H., & Lopez, W. (2017). Cascade Classifiers and Saliency Maps Based People Detection. En *Lecture Notes in Computer Science* (págs. 501-510).
- Aguilar, W. G., Luna, M. A., Ruiz, H., Moya, J. F., Luna, M. P., Abad, V., & Parra, H. (2017). Statistical Abnormal Crowd Behavior Detection and Simulation for Real-Time Applications. En *Lecture Notes in Computer Science* (págs. 671-682).
- Aguilar, W. G., Luna, M., Moya, J., Abad, V., Parra, H., & Ruiz, H. (2017). Pedestrian Detection for UAVs Using Cascade Classifiers with Meanshift. *IEEE 11th International Conference on Semantic Computing (ICSC)*. San Diego.
- Aguilar, W. G., Manosalvas, J. F., Guillén, J. A., & Collaguazo, B. (2018). Robust Motion Estimation Based on Multiple Monocular Camera for Indoor Autonomous Navigation of Micro Aerial Vehicle. *International Conference on Augmented Reality, Virtual Reality and Computer Graphics* (págs. 547-561). Springer.
- Aguilar, W. G., Morales, S., Ruiz, H., & Abad, V. (2017). RRT* GL Based Optimal Path Planning for Real-Time Navigation of UAVs. En *Lecture Notes in Computer Science* (págs. 585-595).
- Aguilar, W. G., Morales, S., Ruiz, H., & Abad, V. (2017). RRT* GL Based Path Planning for Virtual Aerial Navigation. En *Lecture Notes in Computer Science* (págs. 176-184).
- Aguilar, W. G., Rodríguez, G. A., Álvarez, L., Sandoval, S., Quisaguano, F., & Limaico, A. (2017). On-Board Visual SLAM on a UGV Using a RGB-D Camera. En *Lecture Notes in Computer Science* (págs. 298-308).

- Aguilar, W. G., Rodríguez, G. A., Álvarez, L., Sandoval, S., Quisaguano, F., & Limaico, A. (2017). Real-Time 3D Modeling with a RGB-D Camera and On-Board Processing. En *Lecture Notes in Computer Science* (págs. 410-419).
- Aguilar, W. G., Rodríguez, G. A., Álvarez, L., Sandoval, S., Quisaguano, F., & Limaico, A. (2017). Visual SLAM with a RGB-D Camera on a Quadrotor UAV Using on-Board Processing. En *Lecture Notes in Computer Science* (págs. 596-606).
- Aguilar, W. G., Salcedo, V. S., Sandoval, D. S., & Cobeña, B. (2017). Developing of a Video-Based Model for UAV Autonomous Navigation. En *Communications in Computer and Information Science* (págs. 94-105).
- Amaguaña, F., Collaguazo, B., Tituaña, J., & Aguilar, W. (2018). *Simulation System Based on Augmented Reality for Optimization of Training Tactics on Military Operations*. International Conference on Augmented Reality, Virtual Reality and Computer Graphics (págs. 394-403). Springer.
- Anaconda, D. (2018). *Copyright 2018 Anaconda, Inc. All Rights Reserved*. Obtenido de <https://docs.anaconda.com/anaconda/>
- Andrea, C. C., Byron, J. Q., Jorge, P. I., Inti, T. C., & Aguilar, W. G. (2018). Geolocation and Counting of People with Aerial Thermal Imaging for Rescue Purposes. *International Conference on Augmented Reality, Virtual Reality and Computer Graphics* (págs. 171-182). Springer.
- Ariganello, E. (2015). *Redes Cisco, Guía de estudio para la certificación CCNP Routing y Switching*. Madrid: Ra-Ma.
- Basantes, J., Godoy, L., Carvajal, T., Castro, R., Toulkeridis, T., Fuertes, W., . . . Addison, A. (2018). Capture and processing of geospatial data with laser scanner system for 3D modeling and virtual reality of Amazonian Caves. *IEEE Ecuador Technical Chapters Meeting (ETCM)*. Samborondón, Ecuador.
- Callejas, M., & Alarcón, A. (2017). *Modelos de Calidad del Software, un estado del arte*. Colombia: Colombia.
- Campaña, J. (2015). *Estudio de mecanismos de distribución de contenidos en redes vehiculares urbanas*. Madrid: Universidad Carlos III de Madrid. Departamento de Ingeniería Telemática.
- Cebrián, L. (2015-2016). *Reconocimiento de emociones mediante técnicas de aprendizaje profundo*. Valencia: Universidad Politécnica de Valencia.

- Cedeño, G., Marcillo, D., & Pereira, A. (2017). Drone automaton for identification of clandestine airstrips. *IEEE Xplore Digital Library*, 7.
- Constanzo, M. (2014). *Comparación de Modelos de Calidad, Factores y Métricas en el Ambito de la Ingeniería de Software*. Río Gallegos: Universidad Nacional de la Patagonía Austral .
- Copyright © 2018 DJI, T. 1. (27 de jul de 2018). *DJI TM Official Website* . Obtenido de www.dji.com/
- Cordeiro, C., Akhmetov, D., & Park Minyoung, P. (2010). *Ieee 802.11ad: introduction and performance evaluation of the first multi-gbps wifi technology*. Chicago: Chicago, Illinois, USA.
- DJI. (2016). *Manual de Usuario PHANTOM 4*. Madrid: DJI, All Rights Reserved.
- Facchini, H., Pérez, S., Hidalgo, F., & Cárdenas, A. (2018). *Análisis de tráfico multicast de video H264*. Red de Universidades con Carreras en Informática.
- Fernández Morcillo, A. (2016). *Estudio de las tecnologías, mercado, soluciones, normativa y regulación para el uso de dispositivos voladores autónomos*. Linares, España: Escuela Politécnica de Linares.
- Galiano, V. (2014). *Análisis y Comparativa de los Protocolos de Transmisión de Video Adaptativo por Internet*. Madrid: Universidad Carlos III.
- Gonzalez, W. (2017). *Diseño y construcción de un vehículo aéreo no tripulado, del tipo drone cuadricóptero de carreras*. Institución Universtaria Politécnico Grancolombiano.
- Hiaolong, H., Jiang, B., & Yongjia, Y. (2017). *A novel autonomous navigation approach for UAV power line inspection*. IEEE Xplore Digital Library.
- Huang, H. M. (2008). *Autonomy levels for unmanned systems, Terminology, version 2.0* . Contributed by the Ad Hoc ALFUS Working Group Participants. NIST Special Publication 1011-I-2.0.
- Intriago, E. (2016). *Análisis de Tecnologías de Streaming*. Madrid: Universidad Politécnica de Madrid.
- Jara-Olmedo, A., Medina-Pazmiño, W., Mesías, R., Araujo-Villaroel, B., Aguilar, W. G., & Pardo, J. A. (2018). Interface of Optimal Electro-Optical/Infrared for Unmanned Aerial Vehicles. En *Smart Innovation, Systems and Technologies* (págs. 372-380).
- Jara-Olmedo, A., Medina-Pazmiño, W., Tozer, T., Aguilar, W. G., & Pardo, J. A. (2018). E-services from Emergency Communication Network: Aerial Platform Evaluation.

International Conference on eDemocracy & eGovernment (ICEDEG) (págs. 251-256). IEEE.

Kendoul, F. (2012). *Survey of Advances in Guidance, Navigation of Unmanned Rotocraft Systems*. Australian Research Centre for Aerospace Automation.

Nginx.com, O. S. (s.f.). *Nginx Open Source*. Obtenido de <https://docs.nginx.com/>

NVIDIA, A. C. (Febrero de 2017). *CUDA Toolkit 8.0*. Obtenido de <https://developer.nvidia.com/cuda-80-ga2-download-archive>

NVIDIA, C. (2018). *NVIDIA, Procesamiento paralelo CUDA*. Obtenido de <http://www.nvidia.es/object/cuda-parallel-computing-es.html>

NVIDIA, D. (2018). *CUDNN 6.0*. Obtenido de <https://developer.nvidia.com/rdp/cudnn-download>

Oñate De Mora, M. (2015). *Los Drones y sus aplicaciones a la ingeniería civil, Tipología de aeronaves pilotadas por control remoto*. Madrid, España: Consejería de Economía y Hacienda, Comunidad de Madrid.

OpenCV, T. (2018). *OpenCV*. Obtenido de <https://opencv.org/>

Orbea, D., Moposita, J., Aguilar, W. G., Paredes, M., León, G., & Jara-Olmedo, A. (2017). Math Model of UAV Multi Rotor Prototype with Fixed Wing Aerodynamic Structure for a Flight Simulator. En *Lecture Notes in Computer Science* (págs. 199-211).

Orbea, D., Moposita, J., Aguilar, W. G., Paredes, M., Reyes, R. P., & Montoya, L. (2017). Vertical take off and landing with fixed rotor. *Chilean Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*. Pucón, Chile.

PANTHOM 4, D. (2016). *Manual de Usuario PHONTOM 4*. Madrid: DJI, All Rights Reserved.

Pardo, J. A., Aguilar, W. G., & Toulkeridis, T. (2017). Wireless communication system for the transmission of thermal images from a UAV. *Chilean Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*. Pucón, Chile.

Parmar, H., & Thornburgh, M. (2012). *Adobe RTMP*. San José: Adobe Systems Incorporated.

Perez, G. (2016). *Redes WiFi sobre drones: Estudio Teórico*. Cartagena - España: Universidad Politécnica de Cartagena.

Pozueco Alvarez, L. (2014). *Diseño y Evaluación de Sistemas de Estimación de Ancho de Banda Disponible para Servicios Adaptativos de Video Streaming*. Madrid, España: UNED Escuela de Doctorado.

- Pussiol, P. (2014). *Redes Convolucionales en Comprensión de Escenas*. Argentina: Creative Commons Atribución-CompartirIgual.
- Python Software, F. (2001 2018). *PYTHON*. Obtenido de <https://www.python.org/>
- Qifa Ke, C., Deng, J., & Baker, S. (2016). *Image Retrieval Using Discriminative Visual Features*. USA: Microsoft Technology.
- Redmon, J., & Farhadi, A. (2015). *You Only Look Once*. Washington: University of Washington.
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *Unified, Real-Time Object Detection*. Washintong: University of Washintong.
- Ruoyu, L. Y., & Li, W. (2018). *An Integrated Load-balancing Scheduling Algorithm for Nginx-Based Web Application Clusters*. Journal of Physics: Conference Series.
- Salcedo, V. (2018). *Aterrizaje Automático de un Vehículo Aéreo No Tripulado Basado en Seguimiento de Puntos de Interés para Superficies Móviles*. Quito: Universidad de las Fuerzas Armadas ESPE.
- Salgado, M. F., Tierra, A., & Aguilar, W. G. (2017). Travel Planning in Public Transport Networks Applying the Algorithm A* for Metropolitan District of Quito. *Analysis*, 4, 11.
- Salgado, M. F., Tierra, A., Sandoval, D. S., & Aguilar, W. G. (2017). Travel Time Estimation of Public Transport Networks Based on Commercial Incidence Areas in Quito Historic Center. *Analysis*, 294, 78448.
- Salvador, C. V. (2016). *Diseño y Desarrollo de Modelos y Sistemas de Comunicaciones para Vehículos Aéreos no Tripulados y Drones*. Madrid - España: Escuela Técnica Superior de Ingenieros de Telecomunicaciones.
- Soni, R. (2016). *Nginx From Beginner to Pro*. Kolkata: West Bengal.
- Standardization, I. O. (2008). *Standards Codec H.264*. Obtenido de <https://www.iso.org/standards.html>
- Tensorflow, G. (2018). *Tensorflow*. Obtenido de <https://www.tensorflow.org/tutorials/?hl=es>
- Valdez, A. D. (2016). *Antenas en Sistemas Inalámbricos de Comunicaciones Digitales*. Corrientes - Argentina: Revistas Universidad Nacional del Nordeste.
- Vasilios, S., & Dimitrios, D. (2015). Multi-source mobile video streaming with proactive caching and D2D communication. *2015 IEEE 16th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM) (2015)* (págs. 1 -6). Boston, MA, USA: ISBN.

Vera, C., Barbosa, J., & Pabon, D. (2017). *La Tecnología ZigBee Estudio de las Características de la Capa Física*. Colombia: Universidad Tecnológica de Pereira.