POLITÉCNICO DE LEIRIA | ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

Disertación

Maestría en Ingeniería Informática – Computación Móvil

Reconocimiento de humanos en imágenes de búsqueda y rescate con UAV usando una interfaz de realidad mixta

Raúl Homero Llasag Rosero

Leiria, *junio* de *2019*

Disertación

Maestría en Ingeniería Informática – Computación Móvil

Reconocimiento de humanos en imágenes de búsqueda y rescate con UAV usando una interfaz de realidad mixta

Raúl Homero Llasag Rosero

Disertación de master realizado con la orientación de la Doctora Catarina Silva y el Doctor Carlos Grilo, Profesores de la Escuela Superior de Tecnología y Gestión del Instituto Politécnico de Leiria, y la coorientación del Doctor Diego Marcillo, Profesor de la "Universidad de las Fuerzas Armadas – ESPE".

Leiria, *junio* de *2019.*

Esta página fue intencionalmente dejada en blanco

Agradecimientos

Agradezco la apertura social brindada por la cultura portuguesa que me acogió durante mi estadía en Leiria como estudiante internacional, especialmente a mis profesores y a la familia de Alice Cruz, quienes fueron mi apoyo emocional durante mi estadía en el maravilloso país llamado Portugal.

Agradezco a mi familia por su paciencia y cariño desmedido que supieron alentarme en mis momentos de soledad y mucho trabajo, especialmente a mi padre, quien tuvo una experiencia muy similar cuando realizó su doctorado en la ciudad de Coimbra.

A mi madre, que con su vínculo de madre supo mantenerme enfocado en la meta que me propuse.

A mis hermanos Diego y Leo, Diego por ser mi ejemplo por seguir y a Leo por mantenerme siempre jovial con su alegría e inocencia de niño.

Agradezco la sabiduría compartida por mis profesores del Instituto Politécnico de Leiria, especialmente a mis orientadores de este trabajo, Catarina Silva y Carlos Grilo no solo reforzaron mis conocimientos técnicos, sino también elevaron mi sentido de autocrítica y mi capacidad para mantener una posición firme ante toda adversidad.

Es un honor para mí agradecer el cariño incondicional y apertura que me ha brindado el personal docente del departamento de Ciencias de la Computación de la Universidad de las Fuerzas Armadas, especialmente a los profesores Diego Marcillo y Graciela Guerrero, quienes confiaron en mis capacidades para cumplir con los objetivos de este trabajo, confiándome el cuidado de los equipos y el espacio físico que se me asignó.

Esta página fue intencionalmente dejada en blanco

Resumen

El uso de los vehículos aéreos no tripulados ha sido una importante herramienta en aplicaciones de búsqueda y rescate, porque estas aplicaciones son usadas para la preservación de vidas humanas al realizar rápidas exploraciones de las áreas afectadas por desastres naturales. Por otro lado, la detección humana ha sido posible gracias a algoritmos que analizan imágenes obtenidas desde las cámaras instaladas en los vehículos aéreos. Sin embargo, restricciones en la visión artificial y en la interactividad de las interfaces mantienen desafíos. Uno de estos desafíos se relaciona con las técnicas de visión por computador, a partir de la necesidad de transmitir tramas de video en tiempo real desde las cámaras de los drones hacia las estaciones de tierra. Adicionalmente, las interfaces de las estaciones de tierra de este tipo de aplicaciones no son usualmente fáciles de usar por pilotos inexpertos. En este trabajo, nosotros propusimos e implementamos una arquitectura para operaciones de búsqueda y rescate, integrando una interfaz interactiva que puede ser usada con vehículos aéreos de origen comercial y dispositivos montados en la cabeza, dispositivos que proveen realidad mixta a las estaciones de tierra.

El resultado de este trabajo es el prototipo de una aplicación que puede ejecutar misiones de vuelo basadas en puntos de referencia. El planeamiento y ejecución de la misión son presentadas en una inmersiva interfaz de realidad mixta que proyecta la detección de humanos sobre video en tiempo real. Este trabajo presenta también el proceso del ajuste fino de un modelo de detección de objetos con el uso de imágenes aéreas e imágenes de búsqueda y rescate.

Palabras-Clave: detección humana, vehículos aéreos no tripulados, búsqueda y rescate, visión por computador, realidad mixta, dispositivos montados en la cabeza.

Esta página fue intencionalmente dejada en blanco

Abstract

The use of unmanned aerial vehicles has become an important tool in search and rescue applications, because these are used for human lives preservation and quickly exploring natural disaster affected areas. On the other hand, human detection has been possible through algorithms which analyse images obtained from the installed cameras. Nevertheless, constraints on artificial vision and interactive interfaces remain a challenge, namely the ones related to computer vision techniques, since drones' cameras need to transmit video streams in real-time to ground stations. Additionally, ground stations interfaces of this type of applications are usually not easy to use for unexperienced pilots. In this work, we propose and implement an architecture for search and rescue operations, integrating an interactive interface that can be used with commercial unmanned aerial vehicles and head-mounted display devices, which provide mixed reality for ground stations.

The result is an application prototype able to execute flight missions based on reference points. The mission planning and execution is presented on an immersive mixed reality interface that displays human detection over real-time video. This work also shows the fine-tuning process of an object detection model with aerial and search and rescue images.

Keywords: Human Detection, Unmanned Aerial Vehicles, Search and Rescue, Computer Vision, Mixed Reality, Head-Mounted Display.

Esta página fue intencionalmente dejada en blanco

Artículos publicados

Del trabajo efectuado se realizaron las siguientes publicaciones:

- Raul Llasag, Diego Marcillo, Carlos Grilo, Catarina Silva. “Human Detection for Search and Rescue Applications with UAVs and Mixed Reality Interfaces”. CISTI'2019 - 14ª Conferencia Ibérica de Sistemas y Tecnologías de Información. Publicada en el 2019.
- Raul Rosero, Diego Marcillo, Carlos Grilo, Catarina Silva. “SAR missions with commercial UAVs over Mixed-Reality interfaces”. RECPAD 2018 - 24ª Conferencia portuguesa de Reconocimiento de Patrones (Poster). Publicado en octubre de 2018.

Esta página fue intencionalmente dejada en blanco

Lista de figuras

Esta página fue intencionalmente dejada en blanco

Lista de tablas

Lista de fórmulas

Índice

Lista de siglas

AdaBoost: Adaptative Boosting

ANN: Artificial Neural Network

API: Application Programming Interface

AR: Augmented Reality

CA: Cellular Automata

CNN: Convolutional Neural Network

CV: Computer Vision

DJI: Dà-Jiāng Innovations

DNN: Deep Neural Networks

FAA: Federal Aviation Administration

FLIR: Forward Looking InfraRed

FPV: First Person Vision

GB: GigaByte

GPU: Graphic Processing Unit

GPS: Global Position System

GS: Ground Station

HBD: Human Body Detection

HCI: Human Computer Interaction

HD: Human Detection

HMD: Head-mounted Display

HOG: Histogram of Oriented Gradients

HTTP: Hypertext Transfer Protocol

IDE: Integrated Development Environment

IP: Internet Protocol

JSON: JavaScript Object Notation

LTS: Long Term Support

MB: Megabyte

MOS: Mean Opinion Score

MR: Mixed Reality

MS-COCO: MicroSoft Common Objects in COntext

NMS: Non-Maximum Suppression

OD: Object Detection

OpenCV: Open Source Computer Vision Library

PRS: Pedestrian Recognition Systems

QoE: Quality of Experience

QoS: Quality of Service

RAM: Random Access Memory

REST: REpresentational State Transfer

ROI: Region of Interest

RPN: Region Proposal Network

RPV: Remote Piloted Vehicle

RTMP: Real-Time Message Protocol

R-CNN: Regin Convolutional Neural Network

SAR: Search And Rescue

SDK: Software Development Kit

SSD: Single Shot multiBox Detector

TCP: Transmission Control Protocol

UAV: Unmanned Aerial Vehicles

USAR: Urban Search And Rescue

USB: Universal Serial Bus

UWP: Universal Windows Platform

UX: User eXperience

VR: Virtual Reality

YOLO: You Only Look Once

Esta página fue intencionalmente dejada en blanco

1. Introducción

Desastres naturales a gran escala han causado impredecibles y masivas pérdidas de vidas humanas. De acuerdo con (Madeleen & Dorothea, 2006) y (Rainer & Kristie, 2012), existe un incremento en el número de desastres naturales gracias al calentamiento global; desastres que afectan mayormente zonas de origen geofísico desfavorable (ubicación geográfica, condición climática, factor geológico o tectónico).

En el mundo, la zona con más complejidad tectónica se encuentra entre las placas "Nazca" y "Sudamericana" (FAO, 2014). Esta zona geográfica es llamada "El cinturón de Fuego del Pacífico" porque está situada una serie de volcanes activos que producen sismos y erupciones volcánicas. Por otro lado, una zona de convergencia tropical que rodea el globo terráqueo se denomina "Calma Ecuatorial", esta zona se caracteriza por su baja presión atmosférica.

En la intersección de estas dos zonas de riesgo se encuentra el país del Ecuador, donde existen áreas de convergencia intertropical sensibles a experimentar: inundaciones, sequías o efectos del fenómeno de "El Niño" (Pierre & Gustavo, 1998). Adicionalmente, la explotación no controlada de recursos naturales como son la deforestación, la minería y el alto riesgo de actividad volcánica convierten al Ecuador en un país con un elevado índice de riesgo de sufrir desastres naturales, y por consecuencia con gran número de víctimas.

En las últimas cuatro décadas se han registrado mas de 4932 muertes, y 187601 heridos en los desastres naturales del fenómeno de "El Niño" registrados en 1982, 1997 y 1998, el terremoto en la Región Amazónica en 1987, el deslave "La Josefina" en 1993, las erupciones del volcán "Guagua Pichincha" en 1999, los deslizamientos de lava de la erupción del volcán "Tungurahua" desde 1999 hasta la fecha, las inundaciones en la Región Costa en 2008 y el terremoto en la provincia de "Manabí" el 16 de abril del 2016 (FAO, 2014) (El Comercio, 2016).

En estos desastres, algunas personas no pudieron ser localizadas a tiempo y se mantivieron esperando por ayuda, pero finalmente murieron porque sus rescatistas no consiguieron llegar a tiempo (Egred, 2018). Los principales problemas de estos grupos de rescate están relacionados usualmente con el tiempo y la dificultad en cruzar zonas de riesgo, sin contar que muchos de estos rescatistas no llegan a su objetivo, muren cruzando zonas de riesgo, haciendo todo lo posible por localizar y rescatar víctimas (Long, 2016).

1.1. Motivación

Grupos de rescate no usan usualmente drones o vehículos aereos no tripulados (UAV, del inglés Unmanned Aerial Vehicles) como herramienta de exporación y localización de víctimas, a pesar de su gran potencial en operaciones de búsqueda y rescate (SAR, del inglés

Search And Rescue). Estos vehículos son muchas veces usados por companías de televisión y empresas privadas que únicamente los usan para realizar un reconocimiento de daños causados a la infraestructura. Únicamente desde el segundo semestre de 2016, la Institución de Servicio Integrado de Seguridad "ECU911" que actúa a nivel del Ecuador, cuenta con dos drones comerciales como parte de su grupo de rescate, los cuales proveen de video en tiempo real en un dispositivo, control remoto, localización GPS y programación de rutas (ECU911, 2016). Servicios que son solo una pequeña parte de los beneficios que pueden ofrecer este tipo de vehículos. Pues en (Pólkaa, Ptaka, & Kuzioraa, 2017), menciona la importancia de un sistema de localización de personas en terrenos con difíciles condiciones de acceso. En (Rapee Krerngkamjornkit, 2013) y (Piotr Rudol, 2008) se exponen métodos optimizados de reconocimiento de humanos con cámaras de vehículos aéreos, los cuales podrán ser usados dependiendo del alcance del sistema.

De acuerdo con (Gartner, 2017), UAVs comerciales se ubican en el top de tendencias tecnológicas gracias a su desarrollo en distintas áreas. En (Milan Erdelj, 2017), su uso es recomendado para operaciones de búsqueda y rescate por su disponibilidad, asequibilidad y facilidad de uso. Sin embargo, (Xiaoyue Ji, 2017) menciona que el pilotaje de drones desde las estaciones de tierra (Groud Station - GS) en operaciones de búsqueda y rescate ha tenido dificultades, debido a la dificultad de entendimiento de sus interfaces, pues han sido pensadas para pilotos expertos.

Por esta razón, (Xiaoyue Ji, 2017) propone el uso de gafas de realidad virtual (VR, del inglés Virtual Reality) montadas en la cabeza del piloto para proyectar dos realidades, la realidad virtual y la realidad aumentada (AR, del inglés Augmented Reality), obteniendo interfaces más inmersivas que faciliten a los usuarios no expertos el control de los vehículos aéreo. La primera realidad permite obtener en todo momento la imagen capturada por la cámara de la aeronave y la segunda permite estar conciente de la ubicación física real de la aeronave, pero impidiéndole observar el mundo real, aún cuando el vehículo esté a poca distancia del piloto.

Con la combinación de estas dos realidades (VR+AR) nace la realidad mixta (MR, del inglés Mixed Reality). MR genera nuevos entornos, pues coexisten objetos reales con objetos virtuales en tiempo real (Khalaf, y otros, 2018). Al usar gafas de proyección de hologramas y crear este tipo de entornos, es posible que el piloto esté conciente de la realidad física y del mundo virtual al mismo tiempo. Además, estos dispositivos podrían proveer una experiencia más inmersiva al controlar una interfaz con los gestos de la mano, obteniendo conciencia del estado actual del vehículo aéreo cuando esté a la vista del piloto y la ubicación del mismo cuando no se encuentre a la vista, tal y como lo propone (Khalaf, y otros, 2018) con su arquitectura para video juegos de entrenamiento de pilotos. Este concepto aún no ha sido desarrollado para una aplicación SAR de misiones reales con UAVs.

1.2. Objetivos y contribuciones

El objetivo general de este trabajo es desarrollar un sistema de búsqueda y rescate (SAR) que sea controlado con un dispositivo montado en la cabeza, dispositivo que provea una interfaz de realidad mixta MR. Las proyecciones sobre la realidad mixta deben constar de información correspondiente al estado del vehículo aéreo y de imágenes analizadas para localizar víctimas de desastres naturales en zonas de riesgo. Estas imágenes deben ser capturadas desde una cámara óptica instalada en el vehículo.

El trabajo se define como prototipal, por lo que un primer objetivo es proponer e implementar una arquitectura de comunicación de doble vía entre unas gafas inteligentes y un vehículo aéreo comercial a través de un servidor intermediario que actúe como interpretador de eventos. Seguidamente, un segundo objetivo define controlar la ruta de vuelo del vehículo aéreo no tripulado desde una interfaz de gafas inteligentes, control realizado mediante el análisis deductivo de las condiciones y ubicación del vehículo por parte del piloto. Estas condiciones del dron pretenden ser proyectas al frente de sus ojos del piloto como realidad aumentada y realidad virtual. Proyecciones que intregran datos correspondientes a los sensores del UAV, configuraciones del sistema, comandos de control e imágenes capturadas desde su cámara.

Finalmente, como tercer objetivo se tiene la implementación de un método de análisis de imágenes capturadas por la cámara óptica para la identificación y localización de humanos. Estos resultados deben ser proyectados al piloto en tiempo real, indicando si existe la presencia de un ser humano.

1.3. Estructura del documento

De acuerdo con los objetivos definidos en este trabajo, la primera fase consiste en realizar un estudio de los sistemas actuales de búsqueda y rescate con drones, especialmente en los sistemas que utilizan interfaces de realidad virtual VR y/o realidad aumentada AR, así como también, los sistemas que detectan humanos mediante la tecnología de detección de objetos.

El segundo capítulo de este trabajo presenta la evolución de los sistemas de búsqueda y rescate que han utilizado drones y dispositivos de montados en la cabeza HMD de VR y/o AR en sus estaciones de control de tierra, valorando especialmente drones de origen comercial ideales para búsqueda y rescate tanto urbana como de montaña. Una revisión de literatura sobre los métodos de detección de objetos (humanos) utilizados en los sistemas SAR están descritos en el tercer capítulo. En este capítulo se destacan a los métodos tradicionales *Machine Learning* y a los métodos *Deep Learning*. De igual manera, se destacan a los conjuntos de datos (imágenes) y las métricas de evaluación utilizadas para la construcción y evaluación de un modelo de detección de objetos.

Con la base teórica presentada en el segundo y tercer capítulo, el cuarto capítulo presenta la propuesta de la aplicación SAR, haciendo énfasis en sus requerimientos funcionales y no

funcionales, escenarios, arquitectura del sistema y las herramientas software y hardware disponibles. Este cuarto capítulo presenta con mayor detalle la propuesta a la solución del primer objetivo de este trabajo, objetivo que pretende proponer una arquitectura de comunicación de doble vía entre unas gafas inteligentes y un vehículo aéreo.

El quinto capítulo de este trabajo presenta la implementación de la propuesta del primer objetivo, así como también el desarrollo del segundo y tercer objetivo, objetivos que abarcan el control de la la ruta de vuelo del UAV desde el dispositivo montado en la cabeza HMD y la implementación del método para la detección de humanos.

Los resultados del desarrollo de los objetivos de este trabajo son presentados en el sexto capítulo. Estos resultados fueron obtenidos mediante pruebas de calidad de servicio y pruebas subjetivas con usuarios expertos e inexpetos en el pilotaje de drones y en el manejo de dispositivos de realidad mixta. Adicionalmente, este capítulo presenta los índices de detección de humanos en una expedición experimental.

2. Vehículos aereos no tripulados (UAV)

El presente capítulo provee la base teórica pertinente para comprender el estado actual del desarrollo tecnológico de los sistemas de vehículos no tripulados (UAS, del inglés Unmanned Aerial Systems). En la sección introductoria se describe una reseña histórica sobre investigaciones relacionadas con el desarrollo de sistemas de drones, desde la creación del primer dron hasta la creación de los drones comerciales.

Posteriormente, la siguiente sección se centra en una de las investigaciones realizadas con drones para aplicaciones de búsqueda y rescate, identificando tanto sus aportes a la comunidad científica, como los problemas que no se consiguieron atender.

Una limitación a la implementación del sistema con modelos de vehículos aéreos disponibles en el mercado es descrita en la siguiente sección, donde (Milan Erdelj, 2017) limita el uso de cierto tipo de drones, por su accequibilidad, disponibilidad y facilidad de uso. Sin embargo, esta sección está orientada para abarcar el mayor mercado posible para la implementación del prototipo de este proyecto.

Una siguiente sección describe la importancia del uso de la realidad mixta y dispositivos montados en la cabeza como mejora de la interactividad de las estaciones de control de tierra de los sistemas de vehículos aéreos no tripulados. Finalmente, este capítulo finaliza con las limitaciones legales aplicadas en Ecuador y una síntesis del uso de drones para aplicaciones de búsqueda y rescate.

2.1. Introducción

El vehículo aéreo no tripulado (del inglés, Unmanned Aerial Vehicle UAV) hace referencia a un dron o una aeronave que puede volar sin un piloto a bordo. Según (Collins Dictionary, 2018), un dron puede ser una aeronave o un barco que puede navegar autónomamente fuera de la línea de visión del piloto, mientras que en (Oxford Dictionaty, 2018), un dron consta como una aeronave o misil controladamente remotamente. Sin embargo, términos como: torpedos aéreos, vehículos controlados por radio, vehículos pilotados remotamente (RPV, del inglés Remote Piloted Vehicles), vehículos autónomos controlados o sistemas de aeronaves no tripuladas (UAS, del inglés Unmanned Aerial Systems) pueden usarse para describir un objeto volador sin piloto a bordo.

La historia de estos objetos voladores data de 1849, cuando los austriacos atacaron la ciudad de Venecia con globos no tripulados cargados de explosivos, sin embargo, no consta este evento en la historia en algunos relatos de la historia de los drones, por no cumplir con la definición de (Oxford Dictionaty, 2018). La invensión del primer avión exitoso se atribuye a los hermanos Wright en 1900, sin embargo, la creación del primer avión no tripulado o "The Ruston Proctor Aerial Target" se desarrolló en Gran Bretaña 16 años después, el cual

se basó en los diseños de Nikola Tesla (Jaysen A. Yochim, 1998). En la Figura 1, se ilustra un breve resumen del desarrollo de los prototipos de los drones, a partir de la invención de 1900.

Figura 1: Evolución de los vehículos aéreos no tripulados

El "Aerial Target" era una bomba que sobrevolaba Inglaterra que se esperaba usar para contrarrestar el ataque de los dirigibles alemanes. Pero, luego de varios intentos fallidos se decidió abandonar el proyecto, pensando que no tenía un potencial militar. Error que notaría Estados Unidos y solo un año después se aprobó la producción en masa del avión "Hewitt Sperry" con el nombre de "Kettering Bug" (Sullivan, 2006), una tecnología avanzada para el año de 1918, pero que no pudo actuar en combate, pues la primera Guerra Mundial acabó antes.

A partir de ahí, los estudios sobre UAVs continuaron, la Armada del país de los Estados Unidos experimentó con un avión controlado por radio, dando como resultado el "Curtiss N2C-2" en 1937. Simultáneamente los británicos crearon en 1935 el "Queen Bee", que era también controlado por radio (Sullivan, 2006). La primera producción en masa de estos aviones controlados por radio fue el "Radioplane OQ-2", donde 15000 drones fueron creados por los Estados Unidos (EE. UU.) para la Segunda Guerra Mundial. Sin embargo, quien patentó y permitió que estas aeronaves fueran pilotadas desde un control de tierra y alcanzara distancias fuera del rango de visión del piloto fue Edward M. Sorensen (FPO, 2018).

En esa época, la producción era costosa y poco fiable. Por lo que EE. UU. lanzó un programa para construir un dron barato en 1980. Más tarde, en 1982 las fuerzas israelíes utilizaron vehículos aéreos no tripulados para obtener la victoria sobre las fuerzas sirias. Demostrando así su potencial, por lo que, en 1986 EE. UU. e Israel impulsaron el desarrollo del avión "Q2 Pionner" (Kashyap, 2018), usado hasta 2007 para la localización de la artillería enemiga.

La producción de mini y micro drones se introdujeron en 1990, pero según (Ford, 2018) no es hasta el 2006 que drones de venta al público se usaron para aplicaciones no militares, pues fue el primer año en que la Administración Federal de Aviación (FAA, del inglés Federal Aviation Administration) emitió permiso para volar un dron comercial. Las empresas empezaron a utilizarlos en el área agrícola, mientras que agencias del gobierno utilizaron para aplicaciones de búsqueda y rescate, vigilancia fronteriza y lucha contra incendios.

2.2. UAV en Desastres Naturales

Diferentes desastres naturales pueden ser asistidos y gestionados mediante UAVs con el fin de planear misiones de búsqueda que optimicen el tiempo, recursos humanos y materiales, disminuyendo el número de víctimas de muerte en desastres naturales. Este es el caso de los desastres de origen geográfico como lo son terremotos, erupciones volcánicas, tsunamis, deslizamientos de tierra y avalanchas; de origen hidrológico como inundaciones y flujos de escombros; de origen climatológico a causa de temperaturas extremas o meteorológicos como lluvias fuertes, huracanes o tormentas de arena (Milan Erdelj, 2017). Sin embargo, el uso de estos vehículos no es recomendado en desastres meteorológicos debido a que condiciones climáticas inestables no permiten actuar confiablemente durante la evaluación, haciendo su uso limitado.

Existen varias aplicaciones de los drones para la gestión de desastres, sin embargo, las más conocidas son búsqueda y rescate, reconstrucción de infraestructura, servicios médicos, cobertura mediática, evaluación de daños, comunicación autónoma, soporte de evacuación, conciencia situacional, conciencia logística, monitoreo y pronósticos. Debido a la necesidad de una respuesta emergente, misiones de búsqueda y rescate tienen una alta prioridad de ejecución, pues las primeras 72 horas son cruciales para salvar las vidas de personas, heridas, atrapadas o perdidas (Milan Erdelj, 2017).

2.2.1. Búsqueda y rescate

Búsqueda y rescate (SAR, del inglés Search And Rescue) es la operación en respuesta a desastres con el objetivo de localizar personas en peligro inminente para trasladarlos a un lugar seguro (Defense, 2006). Estas operaciones se pueden clasificar en búsqueda y rescate urbano (USAR, del inglés Urban Search And Rescue) y rescate de montaña (Khalaf, y otros, 2018).

El uso de UAVs en aplicaciones SAR es un tema que ha sido tratado en las dos últimas décadas. Una prueba de ello son las investigaciones del laboratorio de la Universidad de Linkoping en Siuza (UAVTech Lab, por sus siglas en inglés Autonomous Unmanned Aerial Vehicle Technologies Laboratory) ha realizado un gran esfuerzo por construir sistemas de aviación con UAVs de gran autonomía (Doherty & Rudol, 2007). La motivación de sus investigaciones con drones no comerciales ha iniciado con los proyectos (Doherty, 2004) (Ali Haydar, 2005).

En el primer proyecto se ha creado una arquitectura escalable en software y hardware para un sistema inteligente sobre drones y en el segundo se usa la realidad virtual (VR, del inglés Virtual Reality) a través del uso de sensores virtuales para comunicar múltiples vehículos aéreos mediante agentes heterogéneos, obteniendo así un modo de vuelo controlado por humanos y otro automático. Estos proyectos, que se han inspirado en un proyecto previo

llamado “WITAS” (Doherty, y otros, 2000), crearon en un plazo de 3 años un helicóptero totalmente autónomo listo para encender y monitorear el tráfico vehicular en Suiza.

Posteriormente, gracias a la emisión de permisos de pilotaje de drones comerciales en el año 2006 por parte de la (FAA, del inglés Federal Aviation Administration), el número de contribuciones científicas en operaciones SAR se han incrementado, aunque paralelamente se han seguido utilizando drones no comerciales. En (Doherty & Rudol, 2007), por ejemplo, se ha conseguido un sistema que sea consiente a la situación con un UAV no comercial. Aun así, brinda un gran aporte, pues implementa un método de detección de cuerpos humanos mediante el análisis de imágenes ópticas y el uso del sensor de posicionamiento global (GPS, del inglés Global Positioning System).

Dando continuidad al proyecto, (Piotr Rudol, 2008) adiciona a éste un análisis del estudio sobre imágenes térmicas para la localización y detección de cuerpos humanos, incluyendo al estudio de los drones otra área de investigación como la Detección de Cuerpos Humanos (HBD, del inglés Human Body Detection), área que pertenece al campo de Visión por Computador (CV, del inglés Computer Vision).

La visión por computador se dedica a la adquisición, procesamiento, análisis y comprensión de imágenes o video. Su gran importancia en aplicaciones de búsqueda y rescate con UAVs ha continuado presentando posteriores mejoras al proyecto de (Piotr Rudol, 2008). Un ejemplo de estas mejoras son los proyectos de (Blondel, Potelle, Pégard, & Lozano, 2014), (Martins, Groot, Stokkel, & Wiering, 2016) y (de Oliveira & Wehrmeister, 2018), trabajos que se abordarán en ítems siguientes de este capítulo.

2.3. UAVs comerciales

El desarrollo tecnológico en drones se ha venido dando a la reducción de costos en la fabricación de vehículos aéreos, el incremento en su demanda, así como también al incremento en la inversión en los drones de carácter civil y una disminución en el carácter militar (Oñate, 2017). Una acelerada competencia en el mercado de consumo hace que estos vehículos sean más baratos, fiables y con mejores características en comparación con los drones militares. Otro aspecto importante en su desarrollo han sido las grandes capacidades de los ingenieros que día a día mejoran sus sistemas, sin embargo, regulaciones como el “Part 107” de EE. UU. y la restricción de peso a 4 kilogramos y altura máxima de vuelo a 500 metros en los Países Bajos ralentizan su desarrollo.

Según el tipo de ala, los drones se pueden clasificar en: ala fija o *Fixed-wing* y ala rotatoria o *Rotary-wing* y según el número de hélices, los *Rotary-wing* se pueden diferenciar al helicóptero o *Helicopter* de un dron con varias hélices, *Multicopter* o *Multi-rotor*. En (Milan Erdelj, 2017) se propone el uso de drones *Fixed-wing* para inspecciones rápidas de la zona afectada en aplicaciones SAR, mientras que para unas inspecciones más detalladas se recomienda el uso de *Multicopters*, debido a su versatilidad al cambio de dirección en el vuelo.

Hoy en día, la empresa número uno en la producción de drones *Multicopter* es la empresa china (DJI, Dà-Jiāng Innovations), que tiene el 70% del mercado mundial consumo (Oñate, 2017). Al igual que en las plataformas iOS[1] y Android[2], es más fácil acceder al mercado desarrollando aplicaciones para DJI[3], que intentar construir una propia (Oñate, 2017). Además, los productos DJI proveen kits de desarrollo sobre estas plataformas, dejando en libertad a los usuarios sobre el desarrollo de sus aplicaciones (DJI, 2016).

El mercado de drones es enorme, y seguirá creciendo tanto, que el crecimiento del mercado para el final del 2018 se cree que será del 19,6% (IDC, 2018). Según un análisis realizado por (DRONEII, 2017) e ilustrado en la Figura 2, para la distribución del mercado en el tercer trimestre del 2016, DJI superó a Parrot por primera vez con su modelo Phantom 4, al establecer alianzas con Apple y la aerolínea Lufthansa.

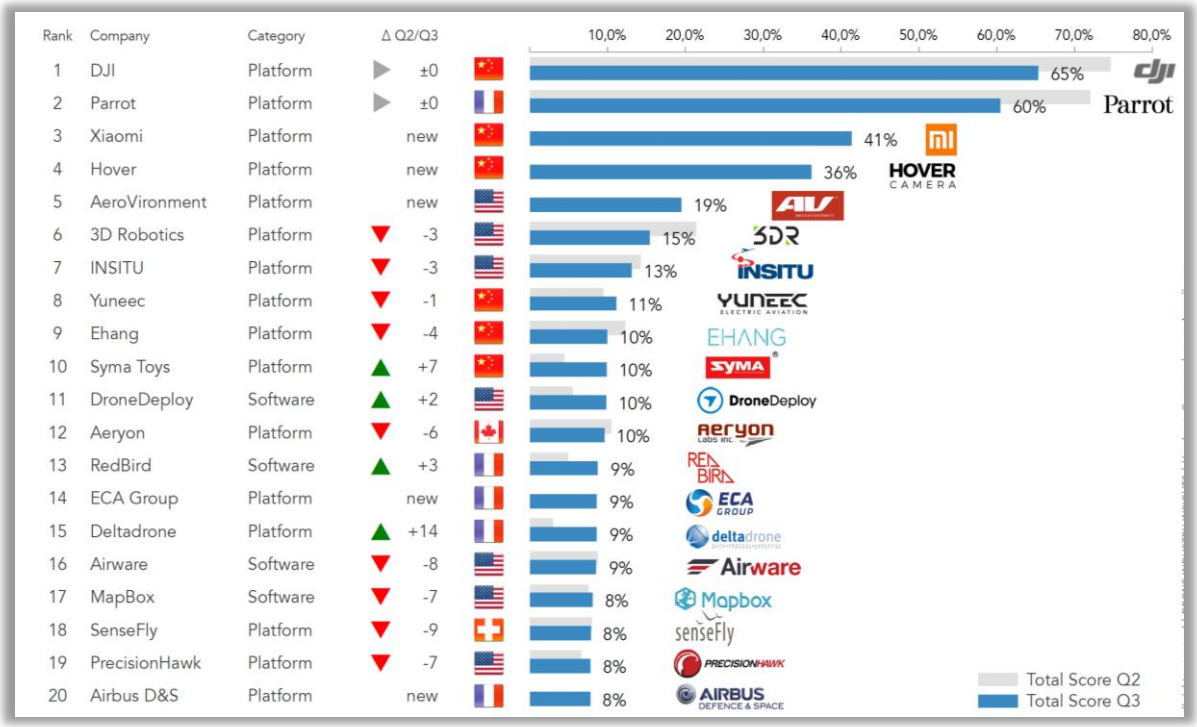


Figura 2: Top de fabricantes de drones a septiembre del 2016 (DRONEII, 2017).

El modelo *Multicopter* BEBOP-2 de Parrot que provee visualización en realidad aumentada y otros modelos de drones descendieron de posición en este cuadro gracias a las bondades cinematográficas de los drones DJI Phantom. Sin embargo, pese a estas variantes en el mercado, el cual puede modificarse en los próximos años, sus características de autonomía durante el vuelo, despegue y aterrizaje inteligente y su integración con herramientas de desarrollo de software multiplataforma los hacen ideales para el desarrollo de aplicaciones de búsqueda y rescate más intuitivas, tal y como lo mencionaba (Xiaoyue Ji, 2017).

[1] https://material.io/develop/ios/
[2] https://developer.android.com
[3] https://developer.dji.com/

2.4. Estaciones de tierra

La estación de tierra (GS, del inglés Ground Station) es el centro de control de un sistema de UAVs (Xiaoyue Ji, 2017). Las principales funciones de este centro de control incluyen tareas como: visualización del entorno, planificación de la misión, comando, control, entre otras. Las estaciones de tierra son clasificadas básicamente en tres tipos: portables, cabinas y las de visualización en primera persona (FPV, First-Person Vision), tal y como lo ilustra la Figura 3.

Figura 3:Estaciones de Tierra, portables, cabinas y FPV (Xiaoyue Ji, 2017).

Las estaciones de tierra portables usualmente están equipadas por una pequeña pantalla, un ratón, un teclado y la aplicación de adquisición de datos instalada sobre un sistema operativo. Por otro lado, las estaciones de tierra de cabina para grandes drones están equipadas con un teclado, un ratón, un sistema operativo y múltiples pantallas para fusionar y diagnosticar fallas. Sin embargo, en (Xiaoyue Ji, 2017), se menciona que la interfaz de las pantallas de cabina tiene un pobre sentido de inmersión, no son interactivas ni flexibles y están orientadas más para pilotos expertos.

2.4.1. Realidad virtual y aumentada en HMD

En solución a este problema y gracias al desarrollo tecnológico de fabricantes de drones como la compañía Parrot, las interfaces FPV son posibles. En este tipo de interfaces se requiere de una cámara instalada en el UAV, la cual transmite imagen a una pantalla montada en la cabeza (HMD, del inglés Human-Mounted Display), similar a los dispositivos que se ilustran en la Figura 4.

Figura 4: Dispositivos montados en la cabeza HMD (Microsoft, 2018)

Los dispositivos montados sobre la cabeza o (HMD, del inglés Head-Mounted Display) se derivaron de la realidad virtual o (VR, del inglés Virtual Reality). La realidad virtual crea un

entorno de escenas de apariencia real que simula el mundo real. Su primer dispositivo fue creado de 1968 y desde ahí ha existido una evolución continua que permitió en 1990 lanzar al mercado un primer modelo de VR con el nombre de “i-glasses” (Chou, y otros, 1999).

Las funcionalidades de estos dispositivos HMD evolucionarían y nacerían más adelante conceptos como: realidad aumentada (AR, del inglés Augmented Reality) y realidad mixta (MR, del inglés Mixed Reality). Sin embargo, la definición de estos conceptos está estrechamente relacionada, por lo que podría ser confusa y requeriría de una explicación ejemplificada para identificar sus límites.

El uso de la realidad virtual VR en sistemas de UAVs es usada para pilotar de una forma más realista. Su potencial fue tan reconocido que la empresa Parrot sacó al mercado en el año 2015 el modelo “Bedop2”, modelo que incluye en su paquete su propio dispositivo HMD y tuvo gran éxito de ventas alrededor del mundo. Por otro lado, en (Prexl, Struebig, Harder, & Hoehn, 2017) utilizó el dispositivo “Vive” de la empresa HTC para simular escenarios de entrenamiento para operaciones de búsqueda y rescate en incendios forestales. Pero, el límite de los HMD de realidad virtual VR se encuentra en la separación de la realidad con el mundo virtual, puesto que una vez colocado el HMD, el piloto ya no puede ver la aeronave como se ilustra en la Figura 5, únicamente puede estar consciente de su posición a través de la imagen que transmite su cámara.

Figura 5: Realidad Virtual en el pilotaje de vehículos aéreos (Parrot, 2018).

El concepto que intenta solventar este problema visual es el uso de la realidad aumentada AR en conjunto con la realidad vitual VR, que recientemente es tratada en sistemas con UAVs. En (Xiaoyue Ji, 2017) trata esta conjunción como una integración del mundo virtual con el mundo real, proveyendo así una experiencia de control más inmersiva, que hace que el operador conozca el entorno de tareas del UAV más directamente. Este proyecto utiliza un HMD de VR de nombre “Oculus”, pero también simula una realidad aumentada AR. A este dispositivo se le transmite imagen proveniente de dos cámaras, la primera es la instalada en la aeronave y la segunda son imágenes precargadas en el sistema que fueron obtenidas mediante satélite. De esta manera el piloto puede estar consciente de la ubicación geográfica del dron y de lo que captura su cámara, tal y como lo ilustra la Figura 6. Aun así, el dispositivo “Oculus” limita al usuario a ver únicamente un entorno virtual, pues el piloto no ve directamente la posición física del dron.

Figura 6: Resultados de la operación del sistema de (Xiaoyue Ji, 2017).

Tras este dilema, se desarrolla el concepto llamado la realidad mixta o MR, que según (Khalaf, y otros, 2018), se puede considerar MR si integra la realidad virtual VR con la realidad física de una manera significativa. En ese mismo proyecto se ha creado una arquitectura que permite la simulación de UAVs en un video juego de SAR en una interfaz de MR para el entrenamiento de pilotos, arquitectura que requiere de dispositivos de realidad aumentada para su implementación.

Los dispositivos montados en la cabeza HMD de MR nace con el modelo "HoloLens" en el 2015, el cual integra la realidad virtual VR con la realidad aumentada AR a través de hologramas 2D y 3D. Su ventaja principal es que puede el usuario puede visualizar un entorno virtual en conjunto con un entorno real. La primera versión de "HoloLens" fue lanzada por primera vez a la venta en marzo del 2016 (Windows, 2016).

2.4.2. Dispositivos de realidad mixta

El precursor y líder de la realidad mixta es "HoloLens". Sus hologramas están soportados únicamente sobre el sistema operativo Windows 10. La Figura 7 se ilustra una magnífica integración del mundo real con el mundo virtual a través del control de gestos con los dedos de la mano, control por voz y proyección de hologramas que son desarrollados sobre el motor de desarrollo de juegos Unity. Sin embargo, este modelo de HMD no es el único en el mercado, pues empresas manufactureras de computadores como Dell, HP, Asus, Samsung, Lenovo, entre otras se unieron a Microsoft para impulsar su creación.

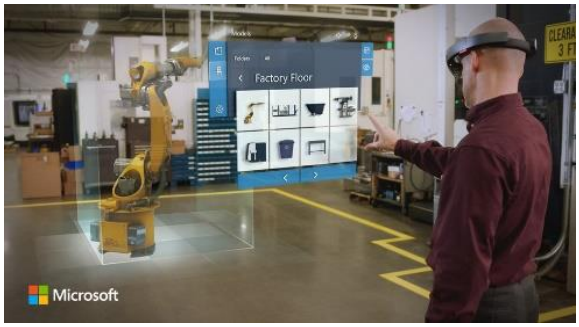


Figura 7: Aplicación de Realidad Mixta sobre HoloLens (Microsoft, 2018)

Estos dispositivos comúnmente se encuentran en el mercado en conjunto con unos controles manuales de movimiento como los de la Figura 8 y tienen en común la restricción del uso

del sistema operativo Windows 10. Hasta el momento no se registra algún sistema de UAVs en aplicaciones SAR reales que use el motor de desarrollo de videojuego Unity para la creación de plataformas universales de Windows o (UWP, del inglés Universal Windows Platform) (Microsoft, 2018).

Figura 8: Samsung HMD Odysssey (Microsoft, 2018)

2.5. Regulaciones para UAS en Ecuador y Portugal

La Dirección General de Aviación Civil (DGAC) de la República del Ecuador emitió en el mes de septiembre del año 2015 un reglamento para sistemas para aeronaves no tripuladas UAS, en donde se indica que cualquier violación de este documento será analizado y resuelto por la Autoridad Aeronáutica Civil.

Este reglamento restringe a usuarios del transporte aéreo y público en general a pilotear drones a una distancia mayor o igual a 9 kilómetros de un aeródromo o cualquier base aérea militar, a una altura máxima de 122 metros sobre el terreno, a horas comprendidas entre la salida y puesta del sol, a pilotear en condiciones meteorológicas de vuelo visual, a pilotear en estado de sobriedad y libre de fatiga, a pilotear en modo automático, a pilotear modo manual únicamente en caso de emergencia, a sostenerse a las restricciones de la empresa fabricante del dron y a contratar un seguro que cubra daños que puedan causar a terceros.

Por otro lado, en Portugal, la Autoridad Nacional de la Aviación Civil (ANAC del portugués Autoridade Nacional da Aviação Civil) emitió un decreto en el mes de julio del 2018, que restringe el uso de drones que pesen más de 250 gramos. Este decreto obliga al piloto a obtener una matrícula del dron con una validez de tres años, además de obtener un seguro de responsabilidad.

Al igual que Ecuador, en Portugal se restringe el uso de drones a una altura mayor a 120 metros y horas del día. Sin embargo, la ley portuguesa es más estricta al fijar multas que van desde los 151 euros a los 4027 euros. Adicionalmente, en Portugal, si un piloto es menor a menor a los 16 años, éste es obligado a manejar el dron con una figura paternal.

2.6. Síntesis

Este capítulo presentó una resumida historia del arte en cuanto a los sistemas de búsqueda y rescate realizado con drones y dispositivos de montados en la cabeza de realidad virtual y aumentada. Hasta aquí se han valorado a los drones y HMD de origen comercial que se han de tomar en cuenta en los capítulos siguientes.

Se valora el uso de vehículos aéreos de ala rotatoria para SAR de montaña y urbana, pues pueden cambiar de ruta en pocos instantes, proveyendo de inspecciones más detalladas. Por otro lado, se descarta el uso de drones para aplicaciones SAR en desastres naturales de origen meteorológico debido a la inestabilidad climática. Aunque se encontró que existen drones comerciales que proveen VR sobre una estación de tierra de tipo FPV, sus dispositivos montados en la cabeza no proveen interfaces interactivas como lo hace la MR. Adicionalmente, los sistemas comerciales de este tipo de drones no implementan visión artificial para la detección de humanos en escenarios de búsqueda y rescate, por lo que el siguiente capítulo aborda técnicas de detección que se pueden adoptar.

3. Técnicas de análisis de detección para búsqueda y rescate (SAR)

Este capítulo presenta las técnicas usadas para la detección en las aplicaciones SAR. Posteriormente, profundiza en la tecnología de detección de objetos, donde, uno de estos objetos puede ser humanos o personas. Para comprender la tecnología detección de objetos y sus técnicas, la evolución del aprendizaje computacional es provisto para diferenciar los métodos tradicionales de *Machine Learning* (ML, con su traducción al español como Aprendizaje Automático) de los métodos *Deep Learning* (DL, con su traducción al español como Aprendizaje Profundo).

Una vez que el lector haya diferenciado las características de aprendizaje de los métodos *Machine Learning* y *Deep Learning* provisto por las redes neuronales artificiales (ANN, del inglés Artificial Neural Network) inspiradas en las neuronas cerebrales del ser humano, se describen las herramientas y técnicas de detección que pueden adoptarse en cada uno de los métodos.

Seguidamente se describen conjuntos de datos (del inglés datasets*)* usados en la detección de humanos con drones con métodos inteligentes tradicionales y métodos de *Deep Learning*. Finalmente, se describen las métricas de evaluación de los modelos de detección de objetos, sus fórmulas de cálculo y la síntesis del capítulo.

3.1. Introducción

La visión por computador (CV, del inglés Computer Vision) es una aplicación de la Inteligencia Artificial (AI, del inglés Artificial Intelligence). La visión por computador es la encargada de la adquisición, procesamiento, análisis y comprensión de las imágenes, con el fin que las máquinas interpreten imágenes matemáticamente y/o simbólicamente. Su objetivo es simular la interpretación realizada por los ojos y cerebro humano para emitir acciones en base al contenido de una imagen o secuencia de imágenes procesadas por las aplicaciones. Las aplicaciones de la visión por computador pueden ser, detección de objetos (OD, del inglés Object Detection), detección de eventos, restauración de imágenes, entre otras. La detección de humanos o personas (HD, del inglés Human Detection) es un caso especial de la detección de objetos, caso que puede ser abordado usando métodos tradicionales de *Machine Learning* y los métodos *Deep Learning*.

El aprendizaje computacional, aprendizaje automático o *Machine Learning*, es un mecanismo usado para la construcción de inteligencia de las máquinas para ser capaces de aprender a través de la experiencia y mejorar progresivamente el desenvolvimiento de una tarea (Gollapudi, 2016). Por otro lado, el aprendizaje profundo o *Deep Learning* es una extensión de *Machine Learning* que se basa en representaciones más eficientes de

aprendizaje en lugar de mejorar continuamente el cumplimiento de una tarea específica (Schmidhuber, 2015).

Gracias a la evolución de un aprendizaje bio-inspirado en las neuronas del cerebro humano, la detección de objetos ha evolucionado significativamente. Las primeras redes neuronales artificiales ANN usadas en los métodos tradicionales de *Machine Learning* no tenían las grandes capacidades de aprendizaje de las redes artificiales utilizadas por los métodos *Deep Learning* (Han, Zhang, Cheng, Liu, & Xu, 2018).

Los métodos tradicionales de detección de objetos se basan en la evaluación estadística de las carácterísticas del objeto, por lo que requieren inicialmente un pre procesamiento de los conjuntos de imágenes. El pre procesamiento de imágenes tiene por objetivo extraer un conjunto reducido de características del objeto (humanos para este caso), como lo hacen por ejemplo los histograma de gradientes orientados (HOG, del inglés Histograms of Oriented Gradient) y las cascada Haar, que se presentan en la Sección 3.2.

Posterior a la extracción de las características de los objetos, las técnicas de clasificación de *Machine Learning* utilizadas para la detección de humanos con drones son presentadas en la Sección 3.3. En esta sección se presentan a los clasificadores AdaBoost (del inglés Adaptative Boosting), la máquina de vectores de soporte (SVM, del inglés Support Vertor Machine) y las redes neuronales artificiales (ANN, del inglés Artificial Neural Network).

3.2. Extractores de características

Los extractores de características que han demostrado mejores resultados en la detección de humanos en las aplicaciones de búsqueda y rescate con drones han sido HOG y Haar. HOG es un descriptor de características o técnica que cuenta las apariciones de la orientación del degradado en partes específicas de la imagen. Un descriptor de característica es la representación de una imagen simplificada mediante la extracción de información útil y la eliminación de la información basura (Mallick, 2016). En la Figura 9, los vectores graficados al contorno del atleta son los que caracterizan al cuerpo humano en esta imagen.

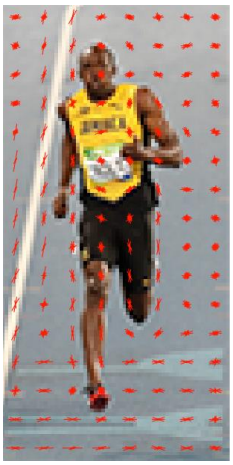


Figura 9: Visualizando HOG (Mallick, 2016)

El algoritmo Haar propuesto por (Viola & Jones, 2001) utiliza las características por su forma y la posición dentro de la región de interés y la escala. Este algoritmo es muy útil para la detección de rostros y seguimiento (OpenCV, 2018). En la Figura 10 se ilustran a las características de borde, lineales y envolvimiento que permiten caracterizar diferentes partes del rostro de una persona.

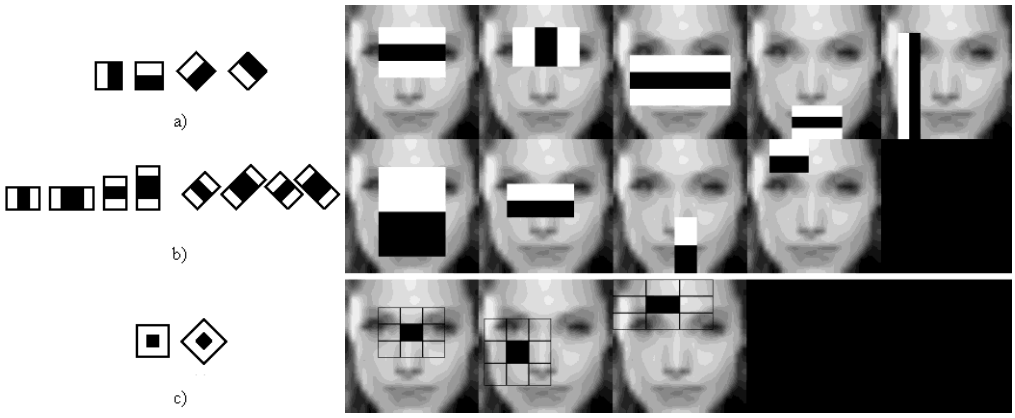


Figura 10: Características Haar (OpenCV, 2018). En el literal a) se visualizan las características de borde, en el literal b) las características lineales y en el literal c) las características de envolvimiento central.

3.3. Clasificadores

Los clasificadores que han demostrado mejores resultados en la detección de humanos con drones han sido SVM, las redes neuronales artificiales ANN y AdaBoost.

3.3.1. SVM

Las máquinas de vectores de soporte SVM son uno de los mejores métodos de clasificación de imágenes (Le, Le, Tran, Tran, & Thanh, 2012). SVM está diseñado para separar dos diferentes clases de objetos de un conjunto de imágenes (x_1, y), (x_2, y_2), ..., (x_n, y_n), donde x_i pertenece a un plano R^d de dimensión d, dimensión que corresponde al espacio de características. Por otro lado, y_i está definido entre {-1, +1}, correspondiente a la clase del objeto, con valores para i = {1, …, n} (Lu & Weng, 2007).

La clasificación de SVM consiste en obtener un hiperplano de separación óptima (OSH, del inglés Optimal Separation Hyperplane) con un largo margen de separación entre las dos clases de objetos, como lo muestra la Figura 11 (Surantha, Isa, Lesmana, & Setiawan, 2017). OSH se construye entonces con una óptima separación entre los hiperplanos basados en una función kernel *K*.

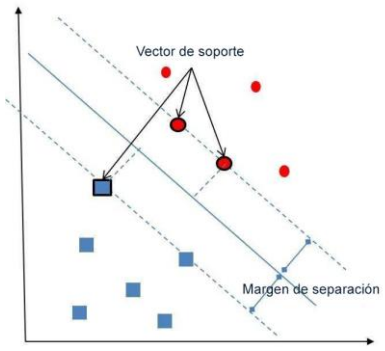


Figura 11: Support Vector Machines (Surantha, Isa, Lesmana, & Setiawan, 2017)

3.3.2. Redes neuronales

El aprendizaje computacional se basa en algoritmos bio-inspirados que son una simplificación del funcionamiento de las redes neuronales biológicas del cerebro humano, que está formado por 10^{11} células nerviosas denominadas neuronas (Silva & Ribeiro, 2018).

La conjunción de estas neuronas se da al simular conexiones entre ellas, función que es provista por los axones de las neuronas del cerebro humano (Gerven & Bohte, 2017). A los conjuntos de estas neuronas artificiales se las denomina redes neuronales artificiales (ANN, del inglés Artificial Neural Networks). Estas redes neuronales también son usadas por los métodos *Deep Learning*, pero en ese caso, las redes son mucho más amplias.

Las redes neuronales brindan apoyo a la solución de problemas en áreas de estudio donde la programación convencional no ofrece un óptimo abordaje. Se habla de un procesamiento cooperativo porque las neuronas del cerebro humano forman una red neuronal al estar sus neuronas intensamente conectadas. Científicos coinciden en que el aprendizaje se debe a la modificación y al establecimiento de nuevas conexiones entre las neuronas.

Una neurona humana está compuesta de tres elementos principales: el cuerpo celular, el axón, y las dendritas como se ilustra en la Figura 12. Se puede apreciar que el axón transmite información mediante el mecanismo de sinapsis hacia las dendritas de otras neuronas, produciendo así acciones o conocimiento.

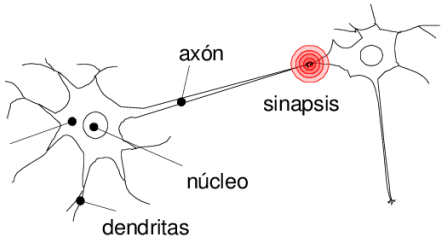


Figura 12: Neurona cerebral humana (Vélez, 2019)

Las ANN son una aproximación de la red neuronal biológica, donde una neurona recibe información de una capa anterior de neuronas y transmite información que es procesada

hacia una posterior capa de neuronas, produciendo el cumplimiento de una tarea al analizar la información, almacenarla y transmitirla.

El análisis de la información se produce en el cuerpo de la neurona, mientras que el almacenamiento de información o peso ω se da dentro de las conexiones. Cada neurona puede tener una función delimitadora que calcula el resultado en función de la información de entrada e impone un límite al valor de salida, emitiendo dicho valor a las siguientes neuronas. La emisión de un resultado con un valor límite es realizado por un perceptrón (Rosenblatt, 1957) y se puede visualizar en la Figura 13.

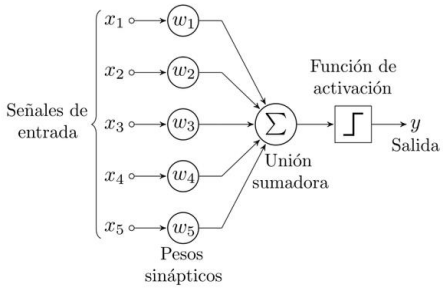


Figura 13: Perceptrón (Rosenblatt, 1957)

El perceptrón simple resuelve únicamente problemas linealmente separables (Minsky & Papert, 1969). Un problema linealmente separable puede ser la clasificación binaria de un objeto con su entorno. Los métodos tradicionales que incluían a las ANN muchas veces han sido sustituidos por el desempeño de SVM como, por ejemplo, en (Moreno, Santiago, Lazaro, & Moreno, 2013), pero se ha demostrado mejores resultados de clasificación con el uso de métodos *Deep Learning*, como por ejemplo en (de Oliveria, 2016).

3.3.2.1. Deep Learning

Los métodos de *Deep Learning* principalmente se basan en el uso de las ANN, pero interconectando las capas de entrada y salida con múltiples capas ocultas de procesamiento no linear. Cada capa oculta puede diferir en su nivel de abstracción para extraer y procesar información necesaria para conseguir un aprendizaje minucioso que se basa en la jerarquía de las características del objeto.

Las redes neuronales artificiales de aprendizaje profundo o comúnmente llamadas redes neuronales profundas (DNN, del inglés Deep Neural Networks) aparecieron con el acuñamiento del concepto de *Deep Learning* (Hinton, 2006). Pero, las DNN fueron implementadas gracias al aumento de la potencia de cálculo de los microprocesadores. De esta manera, los computadores tuvieron la capacidad suficiente para manejar el tiempo de ejecución de múltiples capas de perceptrones. En la Figura 14 se ilustra la transición de las ANN simples hacia las DNN al interconectar múltiples capas ocultas.

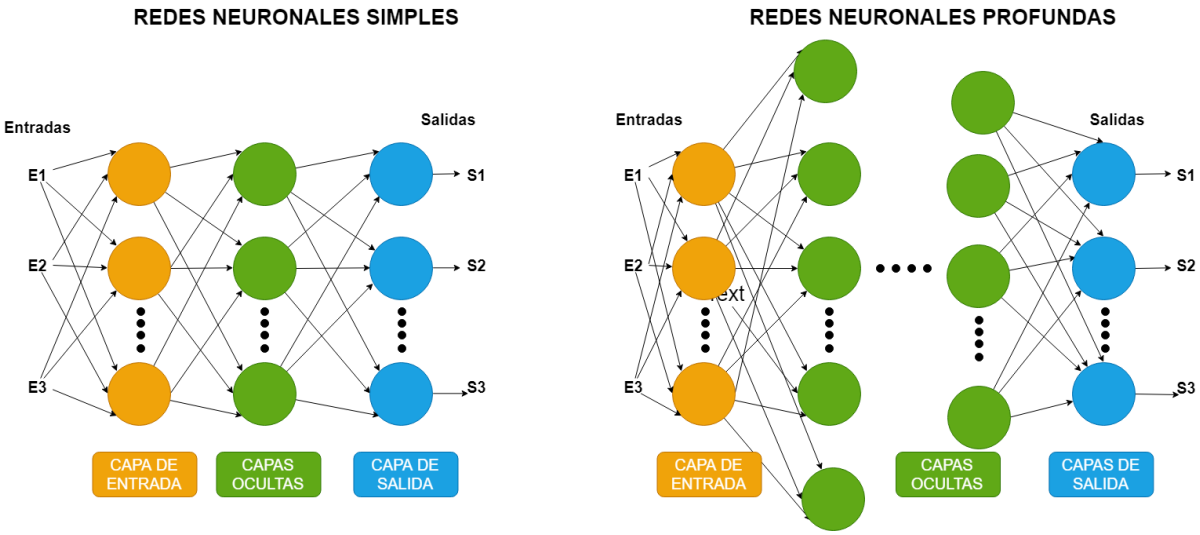


Figura 14: Transición de las redes neuronales simples a las profundas

En la Figura 15 se ilustra como un conjunto de imágenes son la base de conocimiento para que cuatro grupos de capas de una red neuronal aprendan a distinguir rostros mediante un aprendizaje profundo. En este ejemplo, cada grupo de capas mejora el reconocimiento de rostros, pues el aprendizaje se basa en la jerarquización de las características del objeto. La calidad del conocimiento es directamente proporcional al número de muestras y al número de repeticiones de entrenamiento, tal y cual aprendería la mente humana.

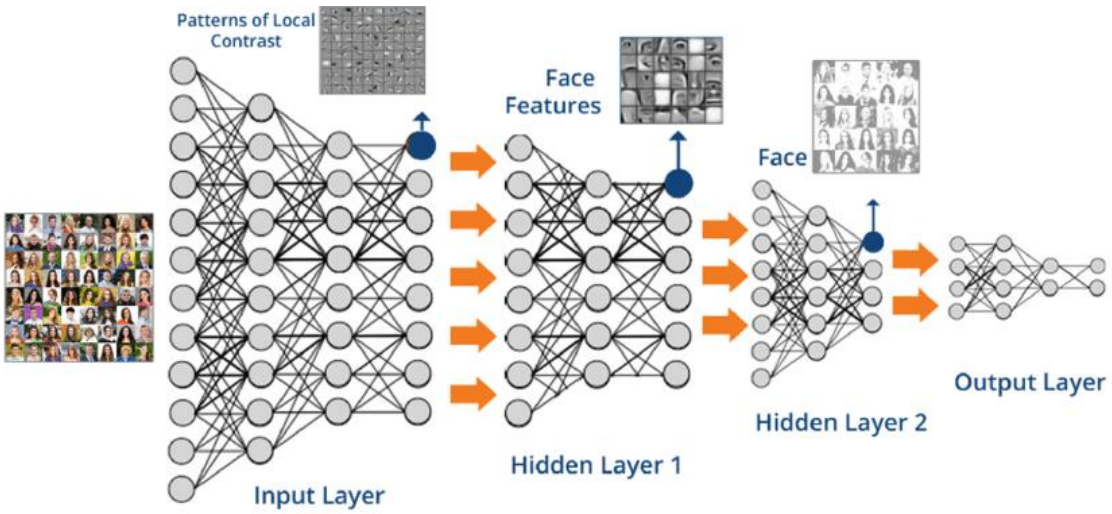


Figura 15:Deep Learning en visión por computador (Grigsby, 2018)

Gracias al aprendizaje profundo, fue posible entrenar a las DNN para detectar múltiples clases de objetos y objetos que pueden tener varias representaciones. Cuando existe más de una instancia de un tipo de objeto en la imagen, la tarea de la localización es mucho más compleja, por lo que estos algoritmos de detección dividen a la imagen por regiones, con el objetivo de clasificar la instancia de un objeto por región, tal y como se puede visualizar en la Figura 16, donde se han remarcado a los objetos identificados.

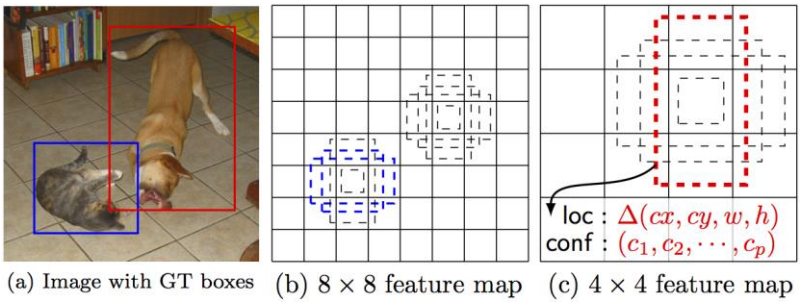


Figura 16: Regiones para detección de objetos (Liu, Anguelov, Erhan, & Szegedy, 2016)

Para optimizar la clasificación de objetos en cada región se han desarrollado ciertos algoritmos basados en regiones de la imagen, donde algunos que se comparan en la Tabla 1. Inicialmente, investigadores desarrollaron los algoritmos (R-CNN, del inglés Region Convolutional Neural Network) (Girshick, Donahue, Darrell, & Malik, 2014) y Fast-R-CNN (Ross Girshick M. R., 2015) que utilizaban redes neuronales convolucionales (CNN, del inglés Convolutional Neural Netowrks) y SVM para potenciar la clasificación.

Las redes CNN operan sobre matrices bidimensionales, por lo que son efectivas en las aplicaciones de visión artificial, clasificación y segmentación de la imagen. Una CNN comprende la fase de extracción de características y la clasificación. La fase inicial de extracción está compuesta de neuronas convolucionales y de reducción de muestreo. El operador de convolución tiene el efecto de filtrar la imagen de entrada con un núcleo pre entrenado, transformando los datos para que las características determinadas por el núcleo se vuelvan más dominantes en la imagen de salida, obteniendo mayores valores numéricos en píxeles que representan la imagen. Finalmente, en la fase de clasificación de la red CNN se encuentran neuronas de perceptrón sencillas que realizan la clasificación sobre las características extraídas.

Aunque R-CNN y Fast-R-CNN no proveen una clasificación en tiempo real, aportaron con la definición de las redes de propuestas regionales (RPN del inglés Region Proposal Network), en la cual se basa su sucesor Faster R-CNN (Ren, He, Girshick, & Sun, 2015).

Figura 17:Faster R-CNN combina a RPN, Selective Search y el detector Fast R-CNN que utiliza las regiones propuestas (Ren, He, Girshick, & Sun, 2015) (Uijlings, Sande, Gevers, & Smeulders, 2013).

La detección de objetos en tiempo real basada en regiones de Faster R-CNN que es ilustrada en la Figura 17 nace al combinar Fast-R-CNN con RPN. Esta detección agrega capas convolucionales que simultáneamente regresan los límites de una región de interés (ROI, del inglés Region Of Interest) y su puntaje de objetividad. Los límites de las regiones son obtenidos gracias al algoritmo de propuesta de regiones llamado Búsqueda Selectiva (del inglés Selective Search) (Uijlings, Sande, Gevers, & Smeulders, 2013). Este algoritmo realiza una segmentación excesiva de la imagen según la intensidad de los píxeles y agrupa regiones similares en función de la compatibilidad de color, textura, tamaño y forma.

Los índices de detección y velocidad de procesamiento se incrementaron con los métodos de detección de objetos de escenario único (del inglés single-stage). Las investigaciones sobre las redes convolucionales CNN (YOLO, You Only Looks Once) (Redmon, Divvala, Girshick, & Farhadi, 2016), YOLO v2 y YOLO 9000 (Redmon & Farhadi, 2017) incrementan la velocidad en la detección de múltiples objetos en tiempo real al usar una única CNN basada en la detección por regiones.

La primera versión de YOLO ofreció un óptimo abordaje al reproducir video a 45 cuadros por segundo (fps, del inglés frames per second). Esto se debe a las limitaciones espaciales propuestas a las regiones propuestas por Selective Search, mitigando la detección múltiple de un mismo objeto (Redmon, Divvala, Girshick, & Farhadi, 2016). YOLO v2 y YOLO 9000 disminuyeron progresivamente el tiempo de procesamiento y mejoraron parcialmente los índices de detección.

En 2016, nace el detector de un solo tiro (SSD, del inglés Single Shot MultiBox Detector) (Liu, Anguelov, Erhan, & Szegedy, 2016) como competidor directo de YOLO 9000. La red alimentada hacia adelante (del inglés feed-forward) utilizada en SSD adhiere al final algunas capas de características que decrecen progresivamente en tamaño y permiten predecir detecciones en múltiples escalas.

La red de SSD es simple de entrenar e integrar en los módulos de detección de las aplicaciones computacionales. Estas características destacan a SSD con otros modelos, pues elimina la generación de posibles regiones, encapsulando todo el cálculo en una sola red. Otra ventaja de SSD sobre otros modelos de detección de objetos, es que, los otros modelos requieren la definición de un umbral de coincidencia basado en el índice de Jaccard (Real & Vargas, 1996) para el proceso de entrenamiento e inferencia, índice que se explica en la Sección 3.6. Mientras que, SSD supone un umbral de 0.5 que permite a la red predecir puntuaciones altas para múltiples casillas predeterminadas que se superponen en lugar de requerir que elija solo la que tenga la superposición máxima (Liu, Anguelov, Erhan, & Szegedy, 2016). El proceso de selección de detecciones de alta puntuación y eliminación de su superposición y vecinos de baja confianza es realizado por el algoritmo NMS (Non-Maximum Suppression) (Goldman, y otros, 2019), algoritmo que es usado por los métodos de detección descritos en la Tabla 1.

La tercera versión de YOLO v3 (Redmon, Ali, 2018) se introduce para reemplazar a YOLO 9000, pues promete ser tres veces más rápido y preciso. YOLO v3 usa predicciones de múltiples escalas, las mismas que proveen un mejor clasificador de la red troncal, pero

presenta mejores resultados en viejas métricas de precisión basadas en la intersección y unión de los cuadros predichos y los cuadros de verdad que se explican en la Sección 3.6.

Tabla 1: Algoritmos de detección de objetos basados en Deep Learning

Algoritmo	Tiempo real	Características
R-CNN	No	**(del inglés, Region Convolutional Neural Networks)**
		Para detectar se necesita, clase de objeto, tamaño y ubicación dentro del cuadro delimitador. Genera 2K propuestas de región, luego se aplica la clasificación CNN para cada región. Finalmente, cada región se puede refinar con regresión.
Fast R-CNN	No	Es más rápido que R-CNN y contiene más eficiente clasificación. Utiliza SVM para potenciar la clasificación y no requiere almacenamiento en disco. Búsqueda selectiva para generar propuestas regionales.
Faster R-CNN	Si	Contiene dos redes RPN y red de detección de objetos. Costo en generar propuestas regionales menores en RPN que en búsqueda selectiva. RPN predice la posibilidad de que un anclaje sea de fondo o de primer plano, y refina el ancla (caja).
YOLO	Si	**(del inglés, You Only Look Once)**
		Conceptualiza a la detección como un problema de regresión de cuadros delimitadores separados espacialmente y probabilidades de clases asociadas. Detección a 45 fps en su versión sencilla. Comete más errores de localización, pero es menos probable que encuentre falsos positivos.
YOLO v2	Si	Se entrena una red clasificadora como para luego reemplazar las capas totalmente conectadas con una capa de convolución y la volver a entrenar de extremo a extremo.
YOLO 9000	Si	Detección en tiempo real de 9000 categorías de objetos. Trabaja con los datasets ImageNet y MS-COCO.
SSD	Si	**(del inglés Single Shot Multibox Detector)**
		Mucho más fácil de entrenar que otros modelos, no requiere la definición de un umbral para el proceso de entrenamiento y presenta dos versiones: SSD300: resolución más baja, más rápido. SSD512: mayor resolución, más precisa.
YOLO v3	Si	Tres veces rápido y preciso a comparación de YOLO 9000.

3.3.3. AdaBoost

En algunos casos, los clasificadores SVM, las redes neuronales artificiales simples u otros clasificadores no mencionados en este capítulo pueden no proveer los resultados de clasificación deseada. Para minimizar estos errores de clasificación se recurre al uso de ensambladores que difieren de su forma de construir el entrenamiento; Bagging (Brieman, 1996) y Boosting (Freund, 1995) (Freund & Schapire, 1996).

En Bagging, cada clasificador es entrenado en paralelo, por lo que cada entrenamiento es diferente. Por otro lado, en Boosting, el ensamble es compuesto por clasificadores

entrenados secuencialmente, donde en cada clasificador se profundizan detalles de los objetos a clasificar (Schwenk & Bengio, 2000). AdaBoost o impulso adaptativo (del inglés, Adaptative Boosting) es el algoritmo más reciente de Boosting, el cual es una gran opción en las aplicaciones de detección de objetos (Schwenk & Bengio, 2000) (Wei, Wei, & Chareonsak, 2005).

AdaBoost es un ensamble sensible al ruido (Freund & Schapire, 1999). En la Figura 18 se ilustran las etapas de entrenamiento de tres clasificadores c1, c2, c3 que dan lugar a un clasificador mejorado. Con un ajuste en cada entrenamiento, los resultados de su clasificación son multidimensionales, dimensionalidad que es proporcional al número de características de los objetos.

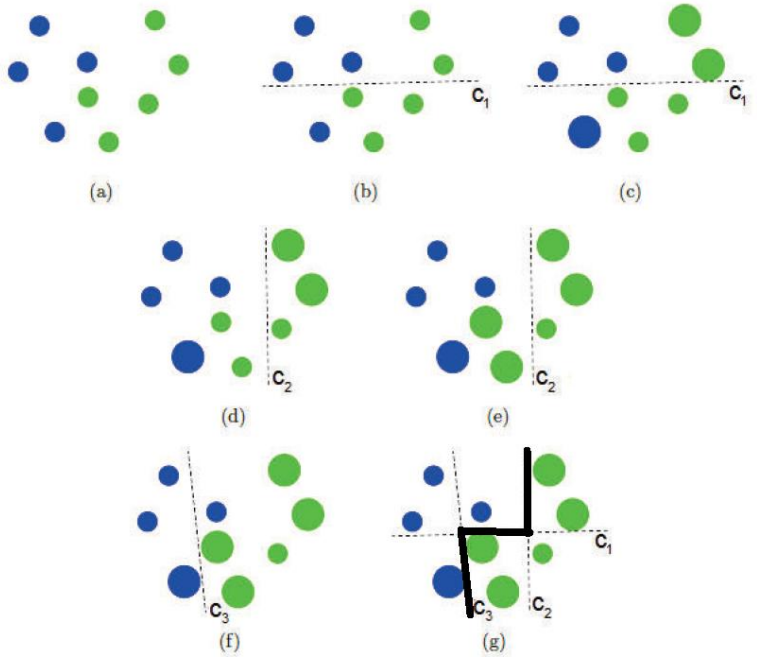


Figura 18: Funcionamiento de AdaBoost (de Oliveria, 2016)

3.4. Herramientas de detección

Una de las primeras implementaciones de redes neuronales artificiales dentro de la visión por computador fue el reconocimiento de números escritos a mano (LeCun, 1989). Posteriormente su desarrollo se estancó a la mitad de la década de los años 90, al igual que las investigaciones de inteligencia artificial. Sin embargo, una de las razones del crecimiento acelerado en el siglo XXI fue debido al lanzamiento de la biblioteca multiplataforma libre para visión artificial (OpenCV, del inglés Open Source Computer Vision) desde su lanzamiento oficial en enero de 1999 y la definición de las redes neuronales profundas o DNN 2006.

Hoy en día existe una gran variedad de herramientas para visión por computador. Por un lado, existen herramientas tradicionales de *Machine Learning* como DLib[4], SciPy[5], entre otros. Por otro lado, están las herramientas con enfoque *Deep Learning* como Tensor Flow[6], Theano[7], Caffe[8], entre otros. También se incluyen bibliotecas de procesamiento de imágenes como SimpleCV[9], BoofCV[10], OpenCV[11], OpenGL[12], FastCV[13], entre otros. A esta lista extensa se suman las herramientas de estabilización y seguimiento de video, marcos de trabajo, reconstrucción de la escena, segmentación, reconocimiento óptico de caracteres, entre otras. Por lo que el uso de estas herramientas depende de los requerimientos del proyecto.

Una biblioteca líder en el desarrollo de aplicaciones de visión por computador es OpenCV (Alliance, 2017). Esta se posiciona por encima de las herramientas que se ilustran la Figura 19. Las razones se deben a su integración con modelos pre entrenados para la detección de objetos como: YOLO, RCNN, SSD, entre otros. También por la integración del módulo (dnn) de *Deep Learning* a partir de su versión 3.3 (OpenCV, 2018). Sin embargo, (dnn) no proporciona una fiabilidad del aprendizaje de redes neuronales como varios marcos de *Deep Learning*, como lo son Caffe, TensorFlow y Torch / PyTorch[14].

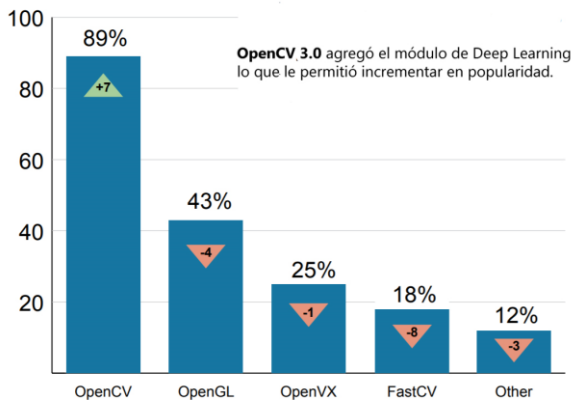


Figura 19: Bibliotecas líderes en desarrollo de visión por computador (Alliance, 2017).

La visión por computador puede enfocarse al uso de clasificadores entrenados mediante *Machine Leaning*, para posteriormente aplicar métodos de identificación de objetos mediante las herramientas descritas en la Figura 19. Pero, también puede enfocarse al uso de herramientas de *Deep Learning* para que el sistema aprenda a identificar objetos. Sin

[4] http://dlib.net/
[5] https://www.scipy.org/
[6] https://www.tensorflow.org/
[7] https://pypi.org/project/Theano/
[8] http://caffe.berkeleyvision.org/
[9] http://simplecv.org/
[10] https://boofcv.org/
[11] https://opencv.org/
[12] https://www.opengl.org/
[13] https://developer.qualcomm.com/software/fastcv-sdk
[14] https://pytorch.org/get-started/locally/

embargo, OpenCV se caracteriza por el primer enfoque, pues su integración con *Deep Learning* llegó únicamente en su versión 3.3, y hasta entonces el *framework* Tensor Flow lideró en la fiabilidad de aprendizaje desde el 2017, según la referencia de la Figura 20.

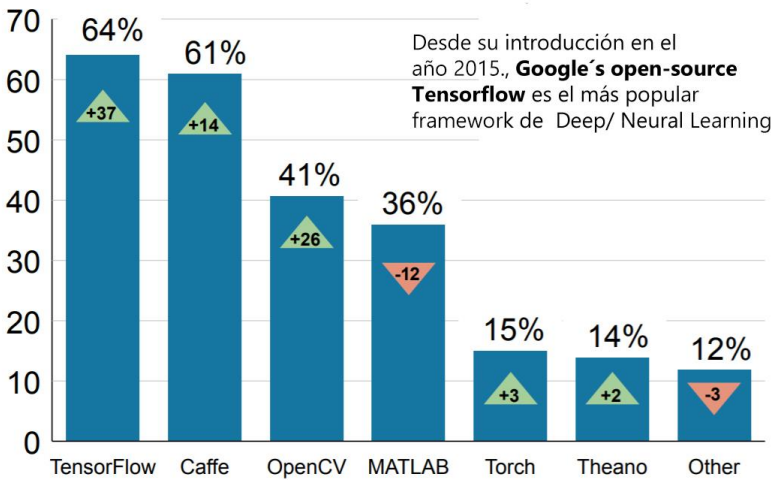


Figura 20: Popularidad de herramientas de Deep Learning (Alliance, 2017).

Adicionalmente, la desición de adoptar una herramienta de visión por computador debe considerar las características de hardware que lo procesará en la construcción del modelo de detección. Por ejemplo, si se cuenta con un equipo que integra una unidad gráfica de procesamiento (GPU, del inglés Graphic Processing Unit) OpenCV es una buena opción, pues ofrece un buen rendimiento de detección de objetos sobre lenguajes nativos como C++[15] o Python[16]. Pero, si no se cuenta con un equipo con GPU, lo más adecuado sería usar herramientas de entrenamiento provistas en la nube (del inglés cloud) para *Deep Learning* como TensorFlow Object Detection API[17], YOLO, entre otros.

Adicionalmente al uso de una unidad GPU, es importante también destacar el número de objetos que se requieren detectar, clasificar, segmentar y/o localizar. En la Figura 21 se puede diferenciar mediante ejemplos gráficos estos términos. Por ejemplo, OpenCV provee modelos de detección de objetos de una sola clase que pueden ser mejorados, mientras que herramientas como YOLO o TensorFlow proveen modelos de detección de múltiples clases de objetos.

Los modelos de detección pueden ser reentrenados para elevar sus índices de detección de objetos en alguna clase de objeto específica (West, Ventura, & Warnick, 2007). Escencialmente, los modelos de detección de provistos por OpenCV, YOLO o TensorFlow son preentrenados con un conjuntos de datos específicos y son capaces de elevar sus índices de detección al incluir nuevos conjuntos de imágenes.

[15] http://www.cplusplus.com/
[16] https://www.python.org/
[17] https://github.com/tensorflow/models/tree/master/research/object_detection

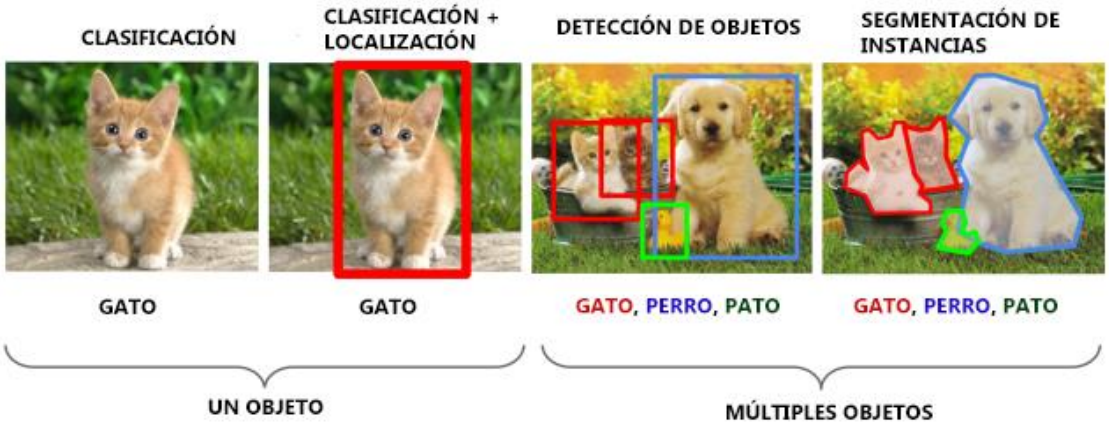


Figura 21: Detección, clasificación y localización (Li, Johnson, & Yeung, 2017)

3.5. Conjuntos de datos

En esta sección se describe a los conjuntos de datos (datasets) que han sido utilizados para la detección de humanos con drones con métodos *Machine Learning*. Sin embargo, también se describen *dataset* con múltiples clases de objetos que pueden servir como base para un reentrenamiento del modelo para la detección de humanos, basándose en el uso de herramientas *Deep Learning*.

3.5.1. Detección de humanos con drones

Se mencionó en la sección anterior que la detección de objetos enfocada en métodos tradicionales requería de la obtención de las características del objeto que se pretende detectar, pero no se mencionó de donde se obtienen dichas características, ni mucho menos a las características que deben tener estos *datasets*.

Esta sección describe a las dificultades de clasificación de humanos con drones y detalla las consideraciones que se han tomado en cuenta para componer *datasets* especializados para las aplicaciones de búsqueda y rescate.

Diferentes restricciones existen en la detección de humanos desde el aire porque la detección con drones no es una tarea fácil. Entre esas restricciones se encuentran el movimiento de la cámara, la distancia de captura de la imagen, el punto de vista, la iluminación y la posición del cuerpo humano debida a una cierta actividad. La Tabla 2 compara los *datasets* usados en los trabajos utilizados por (Blondel, Potelle, Pégard, & Lozano, 2014), (Martins, Groot, Stokkel, & Wiering, 2016), (de Oliveira & Wehrmeister, 2018) y *(Andriluka, Pishchulin, Gehler, & Schiele, 2014)*.

Tabla 2: Datasets para detección de humanos con UAVs

Dataset	Positivos	Negativos	Características
MPII Human Pose	25000 imágenes 40000 anotaciones Ilustrados en la Figura 22		-Contiene actividades de la vivencia diaria de las personas. -Contiene 410 actividades humanas. -Se presentan imágenes de rotación de torso y cabeza.
INRIA	350 000 anotaciones 250 000 imágenes Ilustrados en la Figura 22		-No existe ninguna rotación en la captura de la imagen humana. -Contiene imágenes de humanos con una rotación menor a 40 grados. -Contiene imágenes normalizadas.
GMVRT-V1	4223 Ilustrados en la Figura 23	8461	-Imágenes capturadas de diferentes ángulos de personas relajadas, caminando y con expresiones de angustia. -Imágenes capturadas con movimientos de los ejes del *gimbal* del dron "cabeceo", "inclinación" y "vuelco" (del inglés roll, pitch and yaw).
GMVRT-V2	3846 Ilustrados en la Figura 24.	13821	-Contiene imágenes capturadas en diferentes sitios como playas, calles, aceras, hierba, entre otros. -(Blondel, Potelle, Pégard, & Lozano, 2014) integra este dataset con imágenes térmicas.
Dataset por (Martins et al. , 2016).	1451	1056	-Integra imágenes con zonas de aterrizaje para UAVs.
UCF-ARG	12 actores en 10 acciones grabadas en secuencias de video. Ilustrados en la Figura 25, literal c.		Video capturado desde una cámara aérea a 100 pies de altura. Humanos realizando acciones de boxeo, transporte, palmas, excavación, trote, troncal abierto, correr, lanzar, caminar y agitar.

En la Figura 22 se ilustran imágenes del *dataset* INRIA y MPII Human Pose. Estos dos *datasets* no están orientados a la detección de humanos desde el aire, pero están orientados a la detección de personas. MPII Human Pose, contiene instancias de personas con diferentes posiciones del cuerpo humano, debido a las actividades cotidianas.

En (Blondel, Potelle, Pégard, & Lozano, 2014) se comparó INRIA con el *dataset* GMVRT-V1 o (Generalized Multi-View Real Time – first version) que se ilustra en la Figura 23. Esta comparación fue realizada con entrenamiento basado en HOG y SVM y sugiere que se puede incrementar la efectividad del reconocimiento de persona desde el aire al incrementar el número de ejemplos.

Figura 22: Imágenes de MPII Human Pose e *INRIA (Massera, 2018) (Andriluka, Pishchulin, Gehler, & Schiele, 2014)*

Figura 23: Imágenes del Dataset GMVRT-v1 (de Oliveira & Wehrmeister, 2018)

Mejorando esta propuesta, (Blondel, Potelle, Pégard, & Lozano, 2014) realizan un mecanismo para menorar el espacio de búsqueda adiriendo imágenes térmicas al análisis. Se integra también al *dataset* GMVRT-V2 ilustrado en la Figura 24 y se han utilizado técnicas Haar, HOG y algoritmos basados en AdaBoost, con lo que se consiguió disminuir la velocidad de procesamiento.

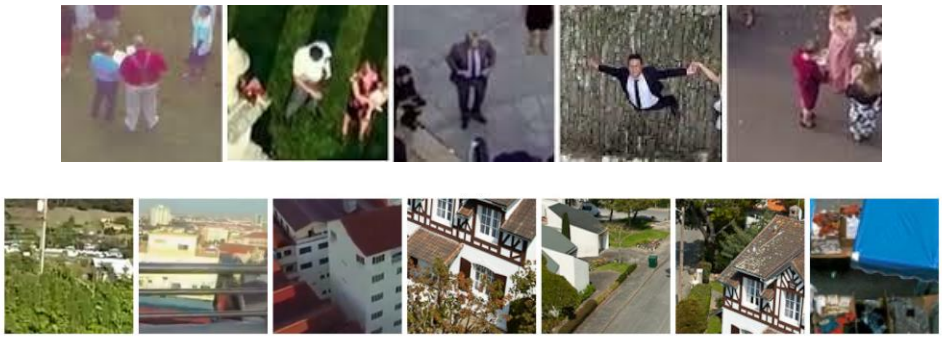


Figura 24: Dataset GMVRT-v2 (Blondel, Potelle, Pégard, & Lozano, 2014)

Por otra parte, (Martins, Groot, Stokkel, & Wiering, 2016) intregran imágenes de sitios de aterrizaje a las imágenes humanas, obteniendo como resultado detección humana y seguridad en el aterrizaje al entrenar la red mediante HOG y SVM. Finalmente, (de Oliveira & Wehrmeister, 2018) obtienen un nivel de precisión en la detección de humanos del 92.3%. Resultado que se debe a la integración de los *datasets* GMVRT-V1(literales a y d), GMVRT-V2 (literales b y e) y UCF-ARG (literal c) (Nagendran, Harper, & Shah, 2010) y las imágenes aéreas provenientes de cámaras UAV ilustradas en la Figura 25.

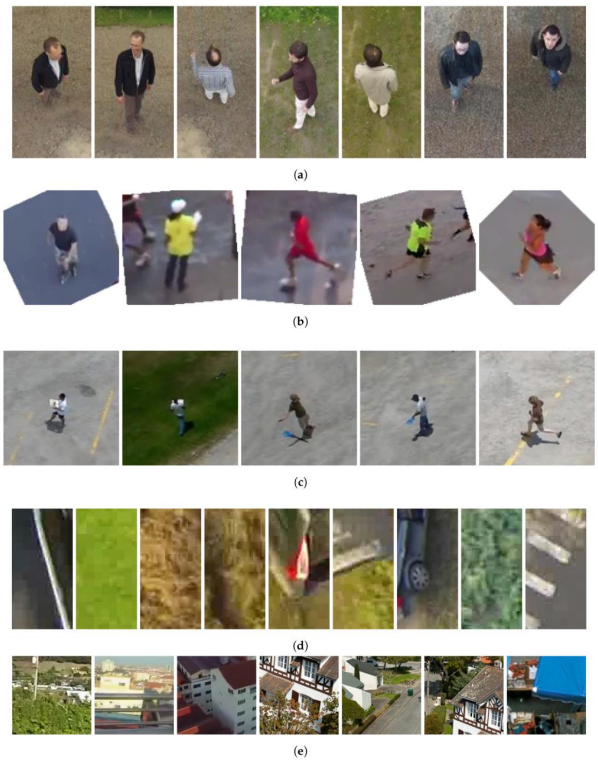


Figura 25: Imágenes de GMVRT y UCF-ARG (de Oliveira & Wehrmeister, 2018)

3.5.2. Detección de múltiples clases de objetos con Deep Learning

Los *datasets* mencionados en la Tabla 3 están valorados en función del número de imágenes, instancias de los objetos y sus características principales. En una primera instancia se tiene al *dataset* MS-COCO que contiene 80 clases de objetos. Se basa en la segmentación individual de objetos para sus etiquetados y es el *dataset* con más instancias de objetos por imagen. Seguidamente se tiene a PASCAL VOC con 200 clases de objetos con un mayor número de imágenes, pero con menor número de instancias de objetos por imagen.

Un gran aporte para detección múltiple de objetos se da con ImageNet gracias al uso una variante del amplio esquema para categorizar objetos y a las 22000 clases de objetos. Por otro lado, SUN cubre una gran variedad de escenas ambientales, lugares y objetos de interiores.

Tabla 3: Datasets para detección de objetos

Dataset	Objectos	Imágenes e intancias	Características
MS-COCO	80	328000 imágenes 2.5 millones instancias etiquetadas Categorías de objetos con más de 5000 etiquetas	(del inglés, Microsoft Common Objects in COntext) Más instancias de objetos por imagen (Lin, et al., 2014).
PASCAL VOC	200	4000000 imágenes con 350000 instancias de objetos etiquetadas en cajas.	Se publican varias versiones de PASCAL VOC desde 2005 hasta 2012. Cuenta con conjuntos de datos para clasificación, detección y diseño de personas (PASCAL2, 2018).
ImageNet	22k	1000 imágenes, donde 14 millones de instancias de objetos han sido etiquetadas.	Presenta varias versiones de ILSVRC (del inglés, Large Scale Visual Recognition Challenge) desde el 2010 hasta el 2017 (StanfordVisionLab, 2018).
SUN	3819	908 categorías de escenas con 16,873 imágenes.	Es una base de datos completa de imágenes anotadas que cubren una gran variedad de escenas ambientales, lugares y los objetos en su interior (Xiao, Hays, Ehinger, Oliva, & Torralba, 2010).

3.6. Evaluación de la detección

En esta sección se describen a los factores más importantes en la detección de objetos. Inicialmente se tienen a las técnicas que permiten evaluar imparcialmente el entrenamiento de los modelos de detección, dado un conjunto de imágenes. Finalmente, las métricas de clasificación binaria y de múltiples clases de objetos son utilizadas para evaluar el modelo.

En la Sección 3.6.1 se describen a las técnicas más simples de construcción de los conjuntos de imágenes para su entrenamiento y evaluación, por ejemplo, divisiones (del inglés splits) y validación cruzada (del inglés cross-validation). Mientras que, en la Sección 3.6.2, se describen a las métricas de clasificación binaria útiles para la evaluación de modelos que clasifican una sola clase de objetos del entorno. Pero, adicionalmente, la Sección 3.6.2 trata las métricas de evaluación de modelos que clasifican múltiples clases de objetos.

3.6.1. Técnicas de evaluación

Para que la evaluación pueda ser dada de forma imparcial, debe ser realizada en datos nuevos, normalmente designados por datos de prueba (Silva & Ribeiro, 2018). Se acostumbra a dividir un conjunto de datos en dos partes: los datos para el entrenamiento y los datos para evaluación. Pueden existir diferentes porciones para estas divisiones o *splits*, pero se restringe la toma del mayor porcentaje para el aprendizaje. Una de las proporciones más recomendada 80/20 por el principio de Pareto (Kiremire, 2011).

Otra opción popular de evaluación, validación cruzada (cross-validation en inglés), consiste en dividir el *dataset* en *k* partes (k-folds) y probar *k* veces con una iteración *k* diferente, como se ilustra en la Figura 26. En esta figura, los datos de prueba se encuentran agrupados de forma ordenada, pero se pueden tomar de forma aleatoria los datos de prueba y los datos de aprendizaje dando lugar a la validación cruzada aleatoria o (random cross-reference en inglés). En el caso de tomar un solo dato de prueba se da lugar a la validación cruzada dejando uno fuera (Leave-one-out cross-validation en inglés).

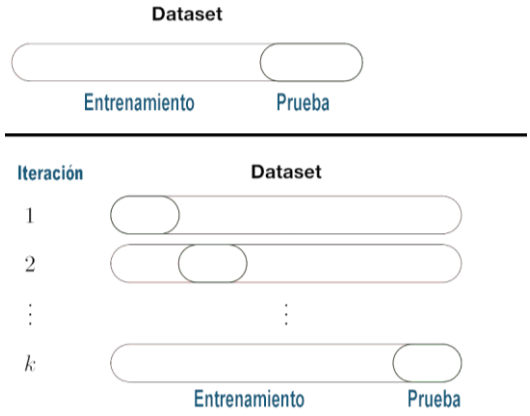


Figura 26: Validación cruzada en k divisiones (Autoría Propia)

Teniendo estos tipos de divisiones, se requiere la definición de un tipo de métricas para evaluar el modelo de clasificación. A pesar de tener la evaluación *cross-reference* una definición para calcular su error, existen otras métricas de evaluación de los modelos de clasificación que son mucho más utilizadas. A estas métricas se las conoce como las métricas de clasificación binaria y las métricas de clasificación múltiple de objetos.

3.6.2. Métricas para clasificación binaria

Para explicar estas métricas vamos a asumir que existe una necesidad de detectar humanos y se tiene de un *dataset* de 100 imágenes, donde existen 80 positivos *P* (contienen un humano) y 20 negativos *N* (no contienen un humano). La matriz de contingencia (confusion matrix en inglés) de la Tabla 4, detalla un caso de prueba del clasificador que detecta correctamente 50 imágenes con humanos y 12 imágenes que no tienen humanos.

Por cuestiones de cálculo de la clasificación se da lugar a los siguientes conceptos:

- **Verdaderos-Positivos VP:** imágenes que realmente **SI** tenían humanos y fueron clasificadas como que **SI** tenían humanos.
- **Verdaderos-Negativos VN:** imágenes que realmente **NO** tenían humanos y fueron clasificadas como que **NO** tenían humanos.
- **Falsos-Negativos FN:** imágenes que realmente **SI** tenían humanos y fueron clasificadas como que **NO** tenían humanos.

- **Falsos-Positivo FP:** imágenes que realmente **NO** tenían humanos y fueron clasificadas como que **SI** tenían humanos.

Tabla 4: Matriz de contingencia (confusion matrix en inglés) (Silva & Ribeiro, 2018).

	En clasificación tiene humano	**En clasificación no tiene humano**
En la realidad tiene humano	Verdadero Positivo VP 50	Falso Negativo FN 30
En la realidad no tiene humano	Falso Positivo FP 8	Verdadero Negativo VN 12

Entonces, la definición y la fórmula de cálculo para cada una de las métricas de evaluación del algoritmo de aprendizaje para este ejemplo corresponden en la siguiente Tabla 5, métricas que al calcularse toman un valor de entre 0 y 1, pero corresponden a un porcentaje del total de ejemplos datos para la evaluación.

Tabla 5: Métricas de evaluación de error de los algoritmos clasificadores de clases.

Métrica	**Descripción**	**Fórmula**	**Valor**
Error	Porcentaje de ejemplos errados sobre el total de ejemplos.	$\frac{FP+FN}{VP+VN+FP+FN}$	$\frac{38}{100} \rightarrow 38\%$
Exactitud (del inglés Acurracy)	Porcentaje de ejemplos acertados sobre el total de ejemplos.	$\frac{VP+VN}{VP+VN+FP+FN}$	$\frac{62}{100} \rightarrow 62\%$
Precisión (P, del inglés Precision)	Porcentaje de ejemplos escogidos como verdaderos que realmente son verdaderos.	$\frac{VP}{VP+FP}$	$\frac{50}{58} \rightarrow 86\%$
Sensibilidad (R, del inglés Recall)	Porcentaje de verdaderos escogidos como verdaderos.	$\frac{VP}{VP+FN}$	$\frac{50}{80} \rightarrow 52{,}5\%$
Especificidad (del inglés Specificity)	Porcentaje de negativos escogidos como negativos.	$\frac{VN}{VN+FP}$	$\frac{12}{20} \rightarrow 60\%$
Razón (FPR, del inglés False Positive Ratio)	Porcentaje de verdaderos escogidos como negativos	$\frac{FP}{VN+FP}$	$\frac{8}{20} \rightarrow 40\%$
F1	Media armónica de la precisión y de la sensibilidad con el fin de dar una medida única que valorice de igual modo los errores cometidos por la precisión y la sensibilidad.	$\frac{2*P*R}{P+R}$	$\frac{65}{100} \rightarrow 65\%$

Una alternativa es también el cálculo del área bajo la curva (AUC, del inglés Area Under Curve) de la característica operativa del receptor o (ROC del inglés Receiver Operator Characteristic). Una curva ROC es una representación gráfica de la *sensibilidad* frente a la *razón* (1- e*specificidad*) para un sistema clasificador binario según se varía el umbral de discriminación.

Al calcular el AUC de la curva ROC se consigue una buena aproximación del error en resultados con decisión binaria. Esto quiere decir que, si una imagen no es clasificada como positiva *P*, es clasificada como negativa *N*. En la Figura 27 se ilustran a diferentes tipos de curvas ROC, donde, la mejor detección está definida por un valor de AUC más cercano a 1.

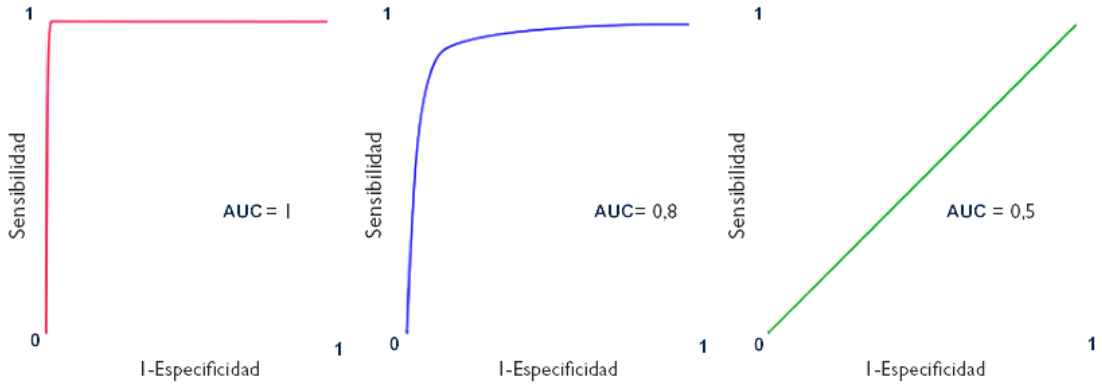


Figura 27: Tipos de curvas ROC

El resultado se complica entonces, si se pretender detectar no solo personas, sino también, animales, automóviles, aviones, entre otros objetos. En este caso, existen otras métricas para la detección de objetos por visión por computador que se detallan en la siguiente subsección.

3.6.3. Métricas para clasificación de múltiples objetos

Existen varias métricas de clasificación y localización para la evaluación el error y el rendimiento de la detección de múltiples objetos. Entre las más populares se encuentran la precisión por curva de precisión (Precision x Recall curve en inglés). Esta métrica es buena para calcular el rendimiento de la detección, que se cambia trazando una curva para cada clase de objeto.

Se considera que un detector de una clase de objeto es bueno si su precisión se mantiene alta a medida que va aumentando la sensibilidad. Un detector de objetos deficiente debe aumentar el número de objetos detectados (lo que aumenta los falsos positivos = menor precisión) para poder recuperar todos los objetos. Es por eso por lo que la curva *Precision x Recall* generalmente comienza con valores de alta precisión (*precision)*, disminuyendo a medida que aumenta la sensibilidad (*recall)* (Padilla, 2019).

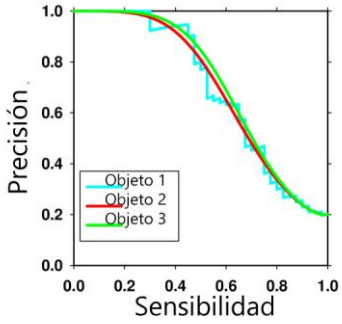


Figura 28: Precisión vs Sensibilidad o Precision x Recall (Autoría Propia).

Otra métrica bastante usada es la precisión media o *AP* (del inglés Average Precision). En esta métrica interviene el concepto de *IoU* (del inglés Intersection over Union), que es una medida basada en el índice de Jaccard (Real & Vargas, 1996) que evalúa la intersección de dos cuadros delimitadores (cuadro real y el cuadro detectado).

Esta evaluación de *IoU* requiere de un cuadro delimitador de la verdad fundamental B_{gt} y un cuadro delimitador predicho B_p. Al aplicar el *IoU* se puede saber si una detección es aceptada o rechazada (Padilla, 2019). En la Figura 29 se ilustra a *IoU* como el área de intersección dividida para el área de unión del recuadro de la verdad fundamental de B_{gt} color verde y el recuadro predicho B_p en color rojo. Este valor *IoU* puede configurarse como un umbral, en donde un determinado valor entre 0 y 1 puede definir a partir de que valor se puede considerar que un objeto haya sido detectado correctamente.

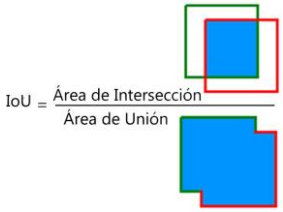


Figura 29: Intersección sobre Unión IoU (Padilla, 2019).

En la precisión media *AP*, interviene también el cálculo del área debajo de la curva (*Precision x Recall*) para una clase de objetos. Cuando se habla de varias clases de objetos se usa la media de las distintas *AP*, dando como resultado al promedio de las precisiones medias (mAP, del inglés mean Average Precision).

El cálculo de varias áreas bajo la curva que se mueven de arriba hacia abajo no es una tarea relativamente fácil porque tienden a cruzarse como en la Figura 28. Sin embargo, la *mAP* ayuda a comparar la detección con distintas clases de objetos. En la práctica, una *AP* es la *precision* promediada entre todos los valores de *recall* desde un umbral *IoU* hasta 1 para una clase de objeto.

Los valores entre el umbral y *IoU* y 1 pueden ser infinitos, pero una inicial alternativa definió un valor de 0 para *IoU*, de esta forma se pudo construir una interpolación de once puntos [0, 0.1, 0.2, ..., 1] que se define en la Fórmula 1. Donde *AP*, es la suma de los distintos valores de *precision P* de un *recall r* que toma 11 diferentes valores.

$$AP = \frac{\sum_{r \in \{0,0.1,\ldots 1\}} P_{interp}(r)}{11}$$

Fórmula 1: Cálculo de AP con interpolación de 11 pasos

En la Fórmula 2, la $P_{interp}(r)$ o *precision P* del *recall r* se define como el máximo valor de *precision* medida sobre el *recall* $\tilde{r}$.

$$P_{interp}(r) = max_{\tilde{r}:\tilde{r} \geq r}\, p\,(\tilde{r})$$

Fórmula 2: Precisión de la interpolación con recuperación r

El error de cálculo de la interpolación en 11 puntos se puede mitigar al tomar en cuenta todos los valores de *recall*. La Fórmula 3 define a *AP* como el área bajo la curva como una sumatoria de todos los valores de *recall* r desde el umbral *IoU* a 1.

$$AP = \sum_{r=umbral\ IoU}^{1} (r_{n+1} - r_n) P_{interp}(r_{n+1})$$

Fórmula 3: Cálculo de AP con todos los puntos de recuperación r.

Con el objetivo de mitigar el error lo máximo posible, la diferencia entre $r_{n+1} - r_n$ debe tender a cero y su interpolación r_{n+1} está definida en la Fórmula 4 como la *precision* máxima cuyo valor de recuperación es mayor o igual que r_{n+1}.

$$P_{interp}(r_{n+1}) = max_{\tilde{r}:\tilde{r} \geq r_{n+1}}\, p\,(\tilde{r})$$

Fórmula 4: Precisión de la interpolación con recuperación r_{n+1}

Un breve ejemplo de cálculo supone que se quiere obtener *AP* para cuatro valores distintos para r que son superiores a un umbral *IoU* dado. Esto quiere decir que se debería calcular con las fórmulas 3 y 4 la sumatoria de las cuatro áreas bajo las curvas dados cuatro valores para la *precision* y el *recall*, tal y como se muestra en la Figura 30.

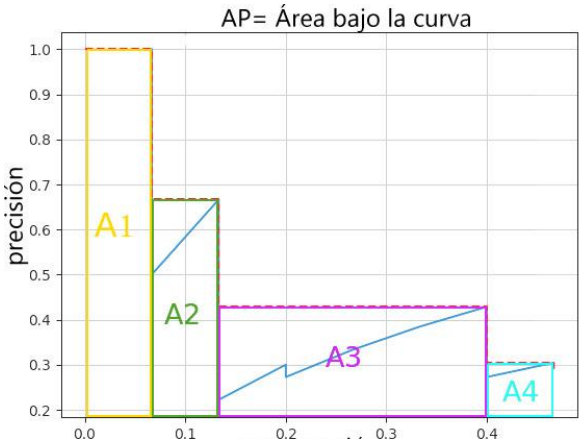


Figura 30: Cálculo de AP con áreas bajo la curva

En la detección de múltiples clases objetos se realiza una clasificación por cada tipo de objeto, donde generalmente el valor del umbral para *IoU* para todas las clases de objetos se

define como mayor o igual al 50%, pero existe una notación en la cual se indica específicamente los umbrales que se han tomado en el entrenamiento del modelo.

Si se toma más de un umbral en una sola clase de objetos, o si se evalúa más de una clase de objetos ya no se habla de *AP*, sino el promedio de la precisión media *mAP* (del inglés mean Average Precision). Este cálculo hace referencia a calcular el promedio de distintas áreas bajo la curva como en la Figura 31, donde se han tomado distintos umbrales *IoU* para la detección de personas.

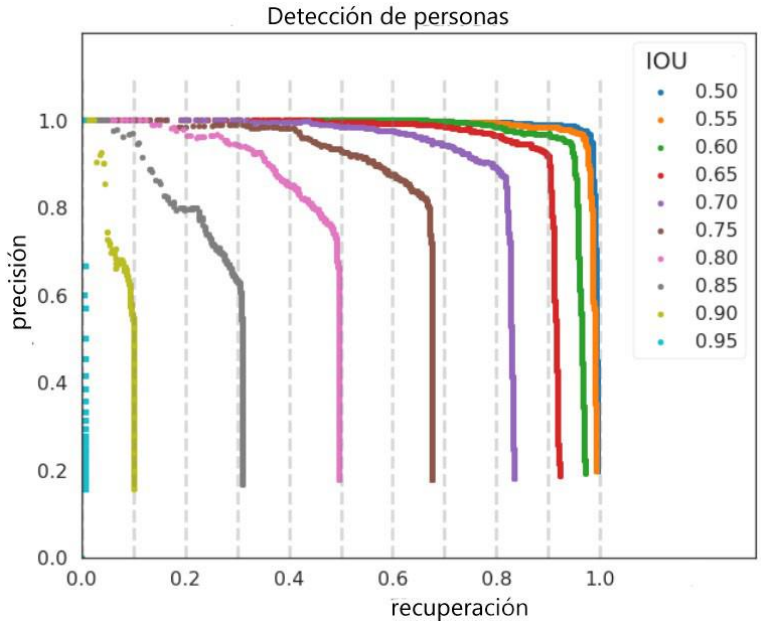


Figura 31: mAP con umbrales mayores a 0.5 con paso de 0.05

La notación para la detección múltiple de objetos se escribe como *mAP@*, donde @ hace referencia al umbral de intersección *IoU*. Considerando que, para que un objeto corresponda a un verdadero-positivo *VP* con un umbral de 50% se escribe como *mAP [0.5]*, pero si se quiere decir que se han tomado los umbrales de 50% en adelante, con un paso de 5% y hasta 95%, como en caso de la Figura 30, se escribe como *mAP [0.5:0.05:0.95]*.

3.6. Aplicaciones SAR con detección humana

Uno de los primeros grandes aportes en la detección de humanos se debe a (llnnrell, 2004). En este proyecto se han utilizado máquinas de vectores de soporte (SVM, del inglés Support Vertor Machine). En este caso, SVM es usada para clasificar características del movimiento, las cuales fueron valoradas estadísticamente en el espacio-tiempo. Este estudio aporta con la diferenciación de humanos y animales mediante el análisis de cuadros (del inglés frames) de video térmico, pero menciona su dificultad de análisis debido al ruido de la luz del día.

Posteriormente, el estudio de rendimiento sobre SVM de (Navneet Dalal, 2005) demuestra que la técnica (HOG, Histograms of Oriented Gradient) supera con mucho rendimiento el conjunto de características para la detección de peatones a la luz del día. Seguidamente, la

incursión de detección de humanos en aplicaciones SAR se da con (Doherty & Rudol, 2007) en los laboratorios de UAVTech, al desarrollar un modelo matemático que busca siluetas humanas en imágenes ópticas y térmicas. Pero su modelo requiere de mejoras, por lo que esta investigación es mejorada en un segundo proyecto. En (Piotr Rudol, 2008), se mejora el modelo para identificar siluetas de humanos en transmisión de video en tiempo real, pero además se integran los métodos de entrenamiento tradicionales orientados a características Haar y el algoritmo AdaBoost.

El proyecto de (Dong Ming, 2009) da seguimiento a los aportes de detección de humanos al lograr la identificación de personas en imágenes de baja resolución capturadas con una cámara térmica. El éxito de esta investigación se debe al estudio de la locomoción humana con (GIF, Gait Feature Image) y a una base de datos (ITGD, Image Thermal Gait Database).

La integración de estos modelos en aplicaciones SAR se da en (Rapee Krerngkamjornkit, 2013). Trabajo que, provee un marco de trabajo o *framework* para la detección de cuerpos humanos. Este *framework* integra HOG y un modelo basado en partes para la detección con drones. Sin embargo, su implementación puede ser costosa, debido a que no provee una fácil integración con modelos de aprendizaje pre entrenados.

En (Lee, et al., 2015), a pesar de su efectivo método de localización y ubicación de objetos en una imagen con una implementación de AdaBoost, SVM y HOG, el uso de las cámaras térmicas y ópticas de captura cercana y lejana no brinda una gran velocidad de respuesta. El uso de la combinación de métodos de detección y *datasets* recaen sobre la velocidad de procesamiento, especialmente en dispositivos móviles.

Finalmente, el más reciente aporte sobre el tema lo hace (de Oliveira & Wehrmeister, 2018). En su trabajo, se ha creado y evaluado diferentes implementaciones de sistemas de reconocimiento de patrones (PRS, del inglés Pedestrian Recognition Systems) para la detección automática de peatones en cámaras térmicas y cámaras ópticas. Estas implementaciones están orientadas a plataformas de computación de bajo costo y utilizan cuatro técnicas de *Machine Learning* en los pasos de extracción y clasificación de características como cascada de Haar, HOG + SVM y redes neuronales convolucionales (CNN, Convolutional Neural Networks).

Los resultados de la clasificación realizada por (de Oliveira & Wehrmeister, 2018) muestran que la implementación de métodos no tradicionales con CNN es la mejor técnica con 99.7% de precisión, seguida con la implementación con métodos tradicionales HOG + SVM con 92.3%.

3.7. Síntesis

Este capítulo abordó el entrenamiento de modelos de detección de una y varias clases de objetos. En primer lugar se mencionó que la detección de objetos puede implementarse al usar métodos tradicionales de *Machine Learning* o con métodos *Deep Learning*. *Machine*

Learning usa redes neuronales como técnica de clasificación binaria para identificar un objeto en especial en una imagen, pero en el caso de *Deep Learning*, la red neuronal se encuentra intensamente interconectadas al formar múltiples capas que proveen grandes capacidades de aprendizaje computacional.

Una vez explicados los métodos de *Machine Learning* y *Deep Learning*, se abordaron las herramientas tecnológicas con las que pueden implementarse, destacando a OpenCV como la más usada en métodos *Machine Learning* y a TensorFlow como la herramienta más usada por *Deep Learning*, gracias a sus capacidades de aprendizaje computacional.

Dentro de las técnicas de deteccción se describió que se requiere de la definición de características del objeto a identificar, características que se obtienen de los *datasets* de detección de humanos, para posteriormente aplicar alguna técnica de clasificación. Por otro lado, se describió que la detección de múltiples clases de objetos con *Deep Learning* requiere del uso de *datasets* para clasificación varias clases objetos mediante algoritmos basados en regiones de la imagen.

Teniendo las técnicas computacionales, herramientas y datasets, se describieron las técnicas de construcción y evaluación de los modelos de detección. En la étapa de construcción se destaca la construcción de los conjuntos de imágenes para entrenamiento y evaluación. Mientras que, en la etapa de evaluación se destacaron las métricas de clasificación binaria para la evaluación de modelos de una sola clase de objetos y las métricas *AP*, *IoU* y *mAP* para la evaluación del aprendizaje de los modelos de detección de múltiples clases de objetos.

Este capítulo finalizó con una revisión de trabajos que se enfocaron en aplicaciones de búsqueda y rescate con detección de humanos en imágenes. El aportes más recientes realizado por (de Oliveira & Wehrmeister, 2018) tiene altos índices de precisión al usar métodos de detección *Machine Learning* y combinar varios *datasets*.

4. Presentación de la propuesta

En este capítulo se detallan los requerimientos funcionales, los requerimientos no funcionales, los escenarios de rescate, la arquitectura del UAS para aplicaciones SAR con interfaces MR y las herramientas software y hardware disponibles para su implementación. Tomando como punto de partida a las limitaciones de los requerimientos funcionales y no funcionales, describiendo escenarios en los que un UAV podría estar involucrado.

Posteriormente se define una propuesta de arquitectura que no limita al uso de un solo dispositivo HMD ni mucho menos un modelo de UAV. En esta arquitectura se describe el flujo de información de tres aplicaciones que se comunican entre sí a través de sus módulos de conexión. Pero, internamente, estas aplicaciones contienen otros módulos que también son detallados. Finalmente, este capítulo concluye con la descripción de las herramientas de hardware que se dispone para el proyecto, así como también las herramientas de software disponibles para su implementación.

4.1. Requerimientos no funcionales

Para la implementación del sistema es necesario contar con un dron comercial que como mínimo integre una cámara óptica. Además, debería proveer a los desarrolladores un kit de desarrollo de software (SDK, del inglés Software Development Kit) de licencia libre para dispositivos móviles (Android o iOS) que permita la programación de la ruta de vuelo, aterrizaje y despegue automático, retorno inteligente, configuraciones de su sistema, sistema anticolisión y vuelo inteligente.

El sistema también debe contar con un dispositivo HMD que provea una interfaz de realidad mixta MR, la cual permita como mínimo la entrada de comandos por gestos de dedos de la mano. Finalmente, se debe contar con un equipo computacional que provea de comunicación inalámbrica entre el dron comercial y el HMD.

4.2. Requerimientos funcionales

El sistema UAS con visualización en primera persona con HMD cuenta con dos requerimientos funcionales esenciales. Requerimientos que son constituidos por sub-requerimientos con un nivel de detalle más específico. Como requerimiento esencial se tiene a la configuración, programación, control y visualización del recorrido de una misión de búsqueda y rescate con el vehículo aéreo en la cual se debe asegurar la seguridad física del UAV, mediante un uso correcto de sus sensores. Como requerimiento de valor agregado se tiene a la visualización en el HMD el video capturado por la cámara óptica montada en el

vehículo aéreo. El video tiene que ser previamente analizado para detectar e indicar la presencia de humanos en la zona de búsqueda.

El requerimiento esencial cuenta con los siguientes sub-requerimientos, clasificados en orden de prioridad:

- Accionar un evento de despegue automático;
- Accionar un evento de aterrizaje automático en caso de que el dron se encuentre en vuelo;
- Visualizar en la interfaz de MR el tiempo restante de la batería, estado de vuelo (tiempo, descripción y distancia), estado de conexión de inalámbrica y la ubicación de la aeronave (longitud y latitud);
- Configurar el punto de origen de vuelo (longitud y latitud), modo de vuelo inteligente, misión, altitud máxima (metros), radio máximo de vuelo (metros) y el estado de activación del sistema de anticolisión (habilitado o deshabilitado y distancia en metros de objetos próximos);
- Configurar un límite de nivel de batería para el regreso automático al punto de origen;
- Configurar un límite de nivel de batería como medio preventivo, el cual permita al usuario decidir si desea acceder al retorno automático al punto de origen;
- Programar una ruta de vuelo del UAV mediante definición de coordenadas geográficas (longitud y latitud) del origen y el destino;
- Accionar un evento de retorno automático hacia el punto de origen en el caso de que se encuentre el dron en vuelo;
- Retornar inteligentemente al punto de inicio a la aeronave en caso de que el límite de batería baja se haya alcanzado o cuando el usuario accione el evento de retorno automático.

4.3. Escenarios de pilotaje

Durante una misión de búsqueda y rescate se pueden tener distintos escenarios, iniciando por la experticia del piloto, la cual puede ser principiante o experimentado. Se recomienda utilizar respectivamente un modo de vuelo inteligente para el primer caso y un control manual para el segundo caso. El diagrama de flujo de la Figura 32 ilustra las diferentes alternativas que puede tomar una misión de búsqueda y rescate.

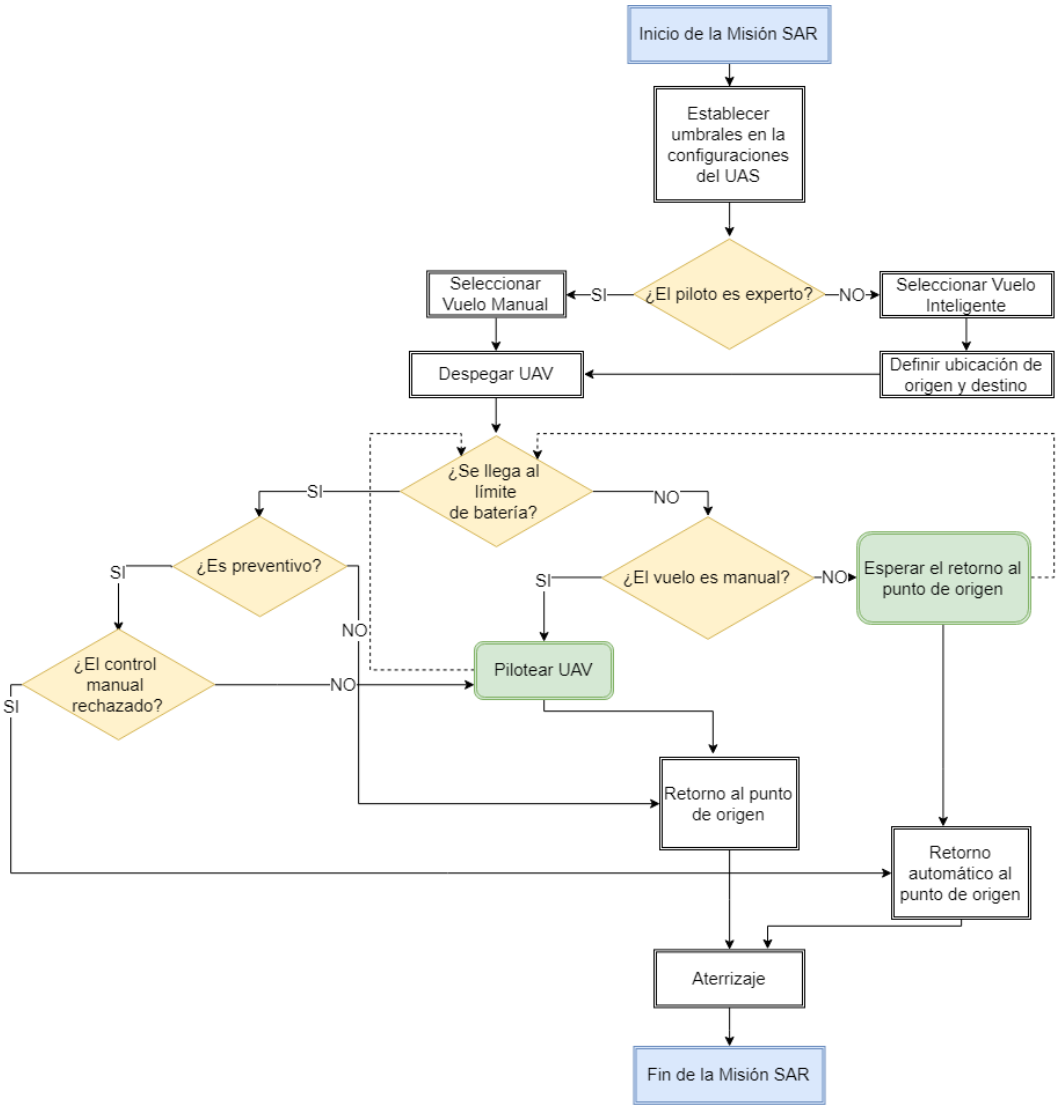


Figura 32: Escenarios del UAS en una misión de búsqueda y rescate.

Cada misión inicia con una configuración del sistema, posteriormente con una selección de vuelo de piloto experto o principiante. En el caso de que el piloto seleccione el modo principiante se definen las coordenadas geográficas del origen y el destino. Una vez definidas estas coordenadas, el piloto debe accionar la función de despegue para que el dron recorra la ruta. En una misión exitosa, donde la batería del dron permita completar la ruta sin problemas, el dron retorna al punto de origen y el piloto debe aterrizar el UAV para finalizar la misión. Pero, en el caso de presenciar un umbral en el nivel de batería, se debe dar la posibilidad al usuario si desea tomar el control manual, en el caso de que este umbral se haya definido como medio preventivo. Pero, en el caso de que este umbral no sea preventivo, un retorno automático al punto de origen es activado.

En el caso de que el piloto sea experto, éste solo debe despegar el dron y pilotear según su criterio. Pero si la batería del dron llega a su límite máximo, el UAV toma el control

automático y retorna el dron hacia su punto de origen. Y si el nivel de batería no lo permite, este aterrizará automáticamente.

4.4. Escenarios de búsqueda y rescate

Los escenarios de las operaciones de búsqueda y rescate se dividen en urbano USAR y de montaña. En el primer caso, el UAV sobrevuela zonas rodeadas por edificaciones y vehículos terrestres, en donde existe una mayor probabilidad de identificar personas, si estas se encuentran en la calle. En los vuelos de montaña, el pilotaje debe tener otros cuidados como, la elevación, las condiciones atmosféricas y la evasión de obstáculos, que puede ser, la floresta, aves, cableado de energía de alta potencia, entre otros.

Las escenas que dejan los desastres naturales son distintas unas de otras, por lo que para el piloto es recomendable hacer el máximo uso de los sensores y del vuelo inteligente del vehículo aéreo para prevenir daños. Por otro lado, la identificación de humanos puede verse afectada debido a obstáculos y el ángulo de captura de la imagen tanto en zonas urbanas y como de montaña, por lo que es necesario implantar la mejor alternativa para su solución.

4.5. Arquitectura

En la Figura 33 se ilustra la propuesta la arquitectura UAS para SAR con una estación de tierra con interfaz de tipo FPV con MR, donde el UAV actúa como explorador y es controlado por un piloto remoto a través de una aplicación instalada en un dispositivo HMD de MR. El piloto es una persona que visualiza hologramas e imágenes en las gafas inteligentes, imágenes que son recolectadas por las cámaras del UAV y transmitidas a través de flujos de datos sobre el protocolo de control y transmisión (TCP, del inglés Transmission Control Protocol), proveyendo al piloto de una conciencia de la ubicación del dron en el espacio para controlar la ruta de una expedición. Previo a la visualización en pantalla del video, las secuencias de video que lo componen son analizadas con el fin de detectar personas. En el caso de que exista la presencia de humanos, los resultados del análisis se integran a las secuencias de video.

Esta comunicación se da gracias a un servidor de comunicaciones que actúa como intérprete entre el dispositivo HMD y la aplicación que controla el UAV. En esta arquitectura la comunicación entre dron y su aplicación controladora se realiza estableciendo un puente mediante su control remoto a través de radio frecuencia. Al conectarse la aplicación mediante una conexión USB o Bluetooth al control remoto, este delega sus funciones a la aplicación que use el kit de desarrollo de software.

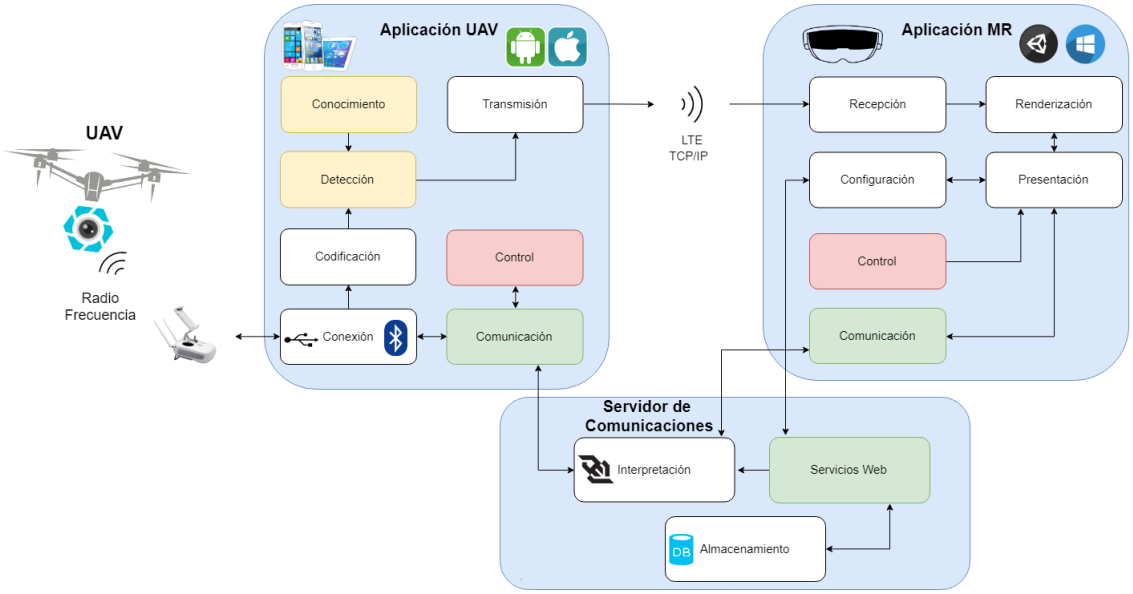


Figura 33: Arquitectura con una GS de tipo FPV con MR

La información del UAS fluye en tres aplicaciones con características diferentes, las mismas que en la Figura 33 se ilustran con color azul y se detallan a profundidad en los siguientes ítems. La primera aplicación obtiene el control del UAV y la información de sus sensores mediante una conexión directa con su control remoto. La segunda aplicación conforma el despliegue de la información mediante hologramas en el dispositivo HMD de MR. Finalmente, la tercera aplicación es un servidor de comunicaciones que interpreta los eventos de las dos aplicaciones anteriores y almacena las configuraciones del UAS.

Esta arquitectura puede estar sujeta a cambios sobre los módulos de *Detección* y *Conocimiento*, correspondientes al análisis de imágenes y reconocimiento de humanos. Estos módulos pueden ser parte de la aplicación MR si el receptor requiere analizarlos.

4.5.1. Aplicación móvil UAV

Un dispositivo móvil (Android o iOS) al correr una aplicación desarrollada con el SDK del dron y establecer conexión con su control remoto a través del puerto USB o Bluetooth tiene la capacidad de controlarlo. En su interfaz contiene únicamente configuraciones para el establecimiento de la conexión con el UAV, el servidor central de comunicaciones y las gafas inteligentes. Una vez establecida esta conexión, la aplicación emite el video capturado desde su cámara óptica directamente hacia la aplicación instalada en el HMD. También permite compartir información con el servidor central de comunicaciones mediante eventos.

El módulo en desarrollarse primero debe ser el *Módulo de Conexión*, pues tiene la tarea de validar si la conexión con el control remoto y el dron es exitosa. En el caso de obtener un resultado satisfactorio, este módulo permite la comunicación bidireccional con el *Módulo de*

Comunicación y la comunicación unidireccional con el *Módulo de Codificación* de video. El *Módulo de Comunicación* es quien tiene una comunicación de dos vías con el servidor de comunicaciones y el *Módulo de Control* del UAV.

Estos dos módulos, *Comunicación* y *Control* son los más importantes entre las aplicaciones, pues el *Módulo de Control* toma acciones sobre datos compartidos por el *Módulo de Comunicación.* El módulo de *Comunicación* de la aplicación móvil implementa un mecanismo de ahorro de energía al transmitir periódicamente información hacia el servidor central mediante la tecnología de *WebSockets,* recibe con la misma tecnología información únicamente cuando se disparen eventos producidos en la aplicación de MR.

La información que se envía periódicamente sin restricción corresponde al estado de su sistema, al estado de la batería y al estado de conexión. Adicionalmente, en el caso de que el UAV se encuentre en una misión de rescate, se puede requerir el envío de las coordenadas de longitud y latitud de la posición actual, la altura alcanzada, el radio alcanzado o la distancia recorrida en kilómetros. Estos datos corresponden a las necesidades de la misión de vuelo y dependen de las capacidades de cada dron.

El *Módulo de Conexión* en el móvil recibe datos del servidor de comunicaciones cuando el servidor escucha la emisión de eventos por parte de la aplicación MR. Estos eventos son relacionados con las configuraciones del sistema como altura máxima, radio máximo, coordenadas geográficas del punto de origen y de la ruta de expedición. Esta información es agregada cuando se programa una ruta para algún modo de vuelo inteligente.

El *Módulo de Conexión* comparte esta información correspondiente a configuraciones con el *Módulo de Control*, el cual tomará acciones si se producen umbrales en el nivel de batería, límites de vuelo o señal de la conexión con el control remoto. Las acciones correspondientes son obligar o dar la opción al piloto para tomar el control manual en el caso de estar en un vuelo automático, activar el retorno automático al punto de origen o aterrizar automáticamente.

Por otro lado, la transmisión de video tiene origen en el *Módulo de Codificación*, el cual recibe los datos sin procesar de las cámaras y comprime los flujos de datos mediante el códec H264. Esta información es transmitida al *Módulo de Detección*, que puede alojar uno o varios algoritmos de detección basados en *Machine Learning* o *Deep Learning* para la clasificación y localización de humanos en imágenes ópticas.

Sin embargo, el *Módulo de Detección* no puede actuar sin la información proporcionada por el *Módulo de Conocimiento.* Este último módulo contiene el conocimiento entrenado por redes neuronales en el caso de usar *Deep Learning*, así como también de características de objetos que han sido obtenidas con métodos de *Machine Learning.* Este conocimiento puede utilizar SVM para acelerar el proceso de clasificación y localización, pero sin este conocimiento obtenido de *datasets*, la detección de objetos es imposible.

Con este conocimiento, el *Módulo de Detección* transforma los flujos de datos de imágenes a mapas de bits (Bitmaps) obtenidos del *Módulo de Codificación* y utiliza algoritmos de detección para remarcar con algún indicador la presencia a una persona dentro de una imagen. Finalmente, las secuencias de video o mapas de bits son enviadas como flujos de bytes al *Módulo de Transmisión,* donde éste a través de una conexión inalámbrica con un *Socket* TCP de tipo servidor envía la información al *Módulo de Recepción* de la aplicación de realidad mixta.

4.5.2. Aplicación de realidad mixta

La aplicación MR obtiene las secuencias de bytes en el *Módulo de Recepción* mediante la conexión inalámbrica con un Socket TCP de tipo cliente. Una vez receptado el primer flujo de bytes, esta información y la que le precede es tratada por un *Módulo de Renderización* que permite cargar en orden secuencial las imágenes obtenidas para ser proyectadas en texturas que son parte del *Módulo de Presentación*.

La *Presentación* de hologramas en MR requiere de texturas para presentar los datos enviados desde la aplicación móvil a través del servidor de comunicaciones. El *Módulo de Comunicación* de esta aplicación, es el encargado de receptar la información que es enviada periódicamente sin restricciones por el UAV, así como también información requerida para el monitoreo de una misión de rescate. Estos datos son compartidos con el *Módulo de Control* que gestiona la entrada por gestos por el piloto.

Por otro lado, la información correspondiente a las configuraciones del sistema es manejada directamente por el *Módulo de Configuración,* pues intercambia los datos a través de recursos web con el servidor de aplicaciones, el mismo que los almacena en una base de datos, para una recuperación del estado del sistema.

4.5.3. Servidor central de comunicaciones

El *Módulo de Interpretación* del servidor de comunicaciones está ilustrado por la Figura 34, donde se visualiza la emisión y recepción de información a través del *Módulo de Comunicación* de la aplicación UAV y los módulos de *Comunicación* y *Configuración* de la aplicación MR. En la figura se distingue la emisión de datos periódicamente sin restricciones con una emisión restringida cuando el UAV se encuentre en vuelo. Se distingue también la recepción de datos correspondientes a una acción o configuración del UAS producida desde la interfaz MR. Por otro lado, la emisión de información por parte de la aplicación MR se realiza mediante dos módulos, debido a que los eventos de pilotaje se deben separar de las configuraciones del sistema.

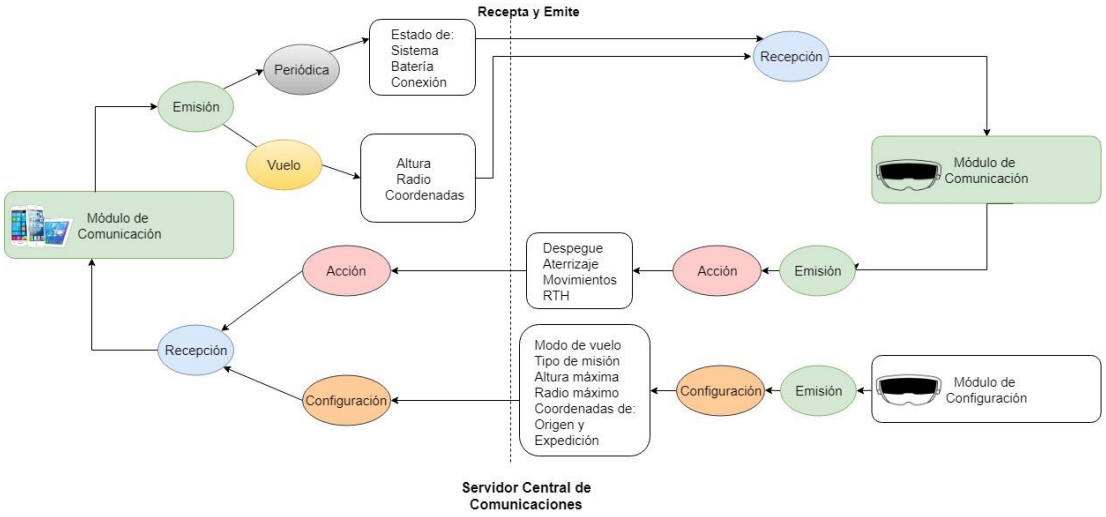


Figura 34: Interpretación de eventos del Servidor Central de Comunicaciones

La información correspondiente a la configuración del sistema es enviada desde el *Módulo de Configuración* de la aplicación MR hacia el *Módulo de Servicios Web* del servidor mediante recursos web. Esta información es almacenada en una base de datos por el *Módulo de Almacenamiento*.

Posteriormente, a este proceso de almacenamiento, la información es transmitida al *Módulo de Comunicación* del UAV y compartida con el *Módulo de Control* para sus respectivas validaciones. Pero en el caso de una recuperación del estado del sistema, el *Módulo de Configuración* solicita al de *Servicios Web* leer la última configuración de la base de datos del *Almacenamiento* y posteriormente esta información también es compartida por el *Módulo de Control* del UAV para sus respectivas validaciones.

4.6. Hardware

Este proyecto cuenta con un vehículo aéreo no tripulado Phantom 4 de la compañía DJI. Este dron está integrado por defecto con una cámara óptica, un cardán para su cámara, un módulo de localización satelital GPS, sensores anticolisión, un control remoto, tres baterías recargables inteligentes, entre otras (DJI, 2016), tal y como lo ilustra la Tabla 6.

Tabla 6: Especificaciones de hardware del dron Phantom 4

Phantom 4	
Cámara óptica	12,4 megapíxeles / 60fps / 5.6 GHz
Almacenamiento Micro SD	64GB
Control Remoto y Batería	20 min de vuelo y Conexión USB con aplicación móvil

Para la interfaz gráfica MR este proyecto cuenta con el primer modelo de gafas inteligentes "HoloLens" proveniente de la compañía (Microsoft, 2018), el cual integra sensores de fusión que permite captar información de eventos, altavoces y un micrófono. Adicionalmente se cuenta con un computador NUC que hace las veces de servidor y cumple con el rol de intérprete entre el UAV y la interfaz del HMD. Finalmente, se cuenta con los teléfonos inteligentes Sony Xperia A2 y Samsung Neo Duos, los cuales cumplen con el rol de intérpretes del UAV al conectarse a la interfaz USB de su control remoto en un ambiente de pruebas de software. Las especificaciones del hardware a usarse en este proyecto y sus costos en euros se detallan en la Tabla 7.

Tabla 7: Especificaciones del hardware y costos del proyecto

Equipo	**Especificaciones**	**Costo**
Dron Phantom 4	Descritos anteriormente en la Tabla 7	1413€
HoloLens	Intel Atom 4 núcleos Almacenamiento interno de 54GB usables Memoria RAM de 980MB Memoria de video de 110MB Cámara de fotos de 2.3 megapíxeles Cámara de video de 1.1 megapíxeles	3200€
Computador Intel NUC	Sistema Operativo Ubuntu 18 LTE Intel NUC Core i5 -6260u Memoria Interna de 500GB / RAM 16GB	300€
Sony Xperia A2	Sistema Operativo Android 7.0 Memoria Interna de 16GB / RAM de 2GB	200€
Samsung Neo Duos	Sistema Operativo Android 7.0 Memoria Interna de 8GB / RAM de 1GB	60€
TOTAL		5173€

4.7. Software

DJI dispone a sus clientes desarrolladores un kit de desarrollo de software SDK llamado Mobile SDK[18]. Este kit está soportado por los sistemas operativos iOS y Android, a partir

[18] https://developer.dji.com/mobile-sdk/

de las versiones 9.0 y 5.0 respectivamente (DJI, 2016). Para el sistema operativo Android, es recomendado el uso del entorno de desarrollo integrado o IDE Android Studio 3.1.4[19], con un kit de desarrollo de Java o JDK 1.7 o superior y la herramienta de compilación de código Grandle o Maven, tal y como provee su proyecto ejemplo en Github[20]. Para un sistema operativo iOS es recomendable el uso de DJI iOS SDK 4.7.1[21] en un entorno de programación sobre el IDE Xcode 8.0 o superior. Por disponibilidad de dispositivos móviles Android, el uso de herramientas de Apple es descartado.

Por el otro lado del sistema, aplicaciones MR sobre HoloLens utilizan tecnologías compatibles con Windows 10. En (Microsoft, 2018) se menciona que no existe SDK para desarrollo de aplicaciones de realidad mixta para este dispositivo HMD, por lo que se requiere instalar en conjunto Visual Studio con el SDK de Windows 10. Se recomienda instalar Visual Studio 2015[22] a partir de la tercera actualización en las versiones Enterprise o PRO de Windows 10 (Microsoft, 2018) con el fin de incluir el Emulador de Hololens a partir de la versión 10.0.14, la cual debe soportar virtualización de Microsoft Hyper-V.

La construcción de interfaces de MR requiere la instalación del motor de videojuegos multiplataforma Unity[23], a partir de su versión 2017.4, la cual debe ser instalada en conjunto con la secuencia de comandos para la Plataforma Universal de Windows – UWV o Windows Store .NET (Microsoft, 2018). Unity soporta scripts en Lenguajes de Programación Javascript[24], C#[25] y otros lenguajes de .NET, sólo si estos compitan un DLL compatible. Adicionalmente y gracias a la bondad de los scripts, Microsoft dispone en Github[26] un conjunto de scripts y componentes llamado *Mixed Reality Toolkit "HoloToolKit"*, con el fin de agilitar el desarrollo de interfaces de aplicaciones de realidad mixta en Windows.

Construir un proyecto desde Unity para UWP y desplegar éste desde Visual Studio hacia el emulador de HoloLens es un proceso que lleva tiempo, por lo cual es recomendada la instalación de *Holographic Remoting Player*[27] en el HMD desde Microsoft Store. De esta manera, las pruebas de la aplicación se realizan mediante una conexión Wi-Fi configurada en la interfaz de Unity y en la de HoloLens. Así, una aplicación puede ser probada en mucho menos tiempo y transmitiendo contenido en tiempo real (Microsoft, 2018).

Para integrar estas tecnologías de fabricantes diferentes, un servidor de aplicaciones orientada a eventos es requerida. El lenguaje de programación sobre el servidor es indiferente, pero para este caso se propone el entorno de ejecución multiplataforma de

[19] https://developer.android.com/studio/releases/
[20] https://github.com/dji-sdk/Mobile-SDK-Android
[21] https://github.com/dji-sdk/Mobile-SDK-iOS
[22] https://visualstudio.microsoft.com/es/vs/older-downloads/
[23] https://unity3d.com/
[24] https://www.javascript.com/
[25] https://docs.microsoft.com/en-us/dotnet/csharp/
[26] https://github.com/Microsoft/MixedRealityToolkit
[27]https://docs.microsoft.com/en-us/windows/mixed-reality/holographic-remoting-player

código abierto Node.js[28] y el lenguaje de código abierto TypeScript[29], pues reducen errores de código gracias a su inteligencia y correcciones de mecanografía.

Bibliotecas disponibles para TypeScript como, *socket.io*, *restify*, *rxjs* y de bases de datos, proveen respectivamente la comunicación orientada a eventos mediante sockets web, desarrollo y comunicación de REST API's, construcción de programas asíncronos orientados a eventos a través de observadores, y conexión a base de datos. Se puede hacer uso de cualquier base de datos para esta arquitectura, sin embargo, se propone el uso de la base de datos orientada a documentos MongoDB[30] que se conecta a través de la biblioteca mongodb.

Para la transmisión de las tramas de video desde la aplicación móvil hacia la aplicación MR, se propone el envío directo de flujos de bytes mediante una conexión TCP/IP y el uso de la biblioteca FFmpeg[31] para transformaciones de formato de video. La conexión TCP/IP se establece mediante Sockets TCP, que, para el caso de Unity con programación de script basados en C#, se requiere del uso del espacio de bibliotecas como BestHTTP, espacios de nombres *System.Net.Sockets* y *System.Threading*. Para el caso de Android y iOS se requiere de las clases *ServerSocket* y *CFSocket* respectivamente.

Adicionalmente, se propone el uso de la biblioteca OpenCV en su versión estable 3.4.3 para la detección de humanos en imágenes ópticas sobre la aplicación móvil con *Machine Learning*. La misma que se integra con el kit de desarrollo nativo de (NDK, del inglés Native Development Kit[32]) en el caso de Android, ofreciendo una programación sobre su lenguaje nativo C++ para incrementar el rendimiento de los algoritmos de reconocimiento de objetos. Alternativas a esta herramienta es el uso YOLO[33] o de TensorFlow Object Detection API[34]. Tensorflow es compatible con la integración con varios algoritmos de detección de objetos orientados a *Deep Learning*.

Como alternativa, el análisis de imágenes puede ser realizado sobre las secuencias de bytes transmitidas a la aplicación MR, si este fuera el caso. La tienda de Unity provee la biblioteca OpenCV, la cual requiere de la instalación adicional de la herramienta CMake[35] 3.11 o superior para crear paquetes de código abierto que usen algoritmos de reconocimiento sobre en el lenguaje de programación C++.

[28] https://nodejs.org/
[29] https://www.typescriptlang.org/
[30] https://www.mongodb.com/es
[31] https://www.ffmpeg.org/
[32] https://developer.android.com/ndk/
[33] https://pjreddie.com/darknet/yolo/
[34] https://github.com/tensorflow/models/tree/master/research/object_detection
[35] https://cmake.org/

4.8. Síntesis

Este capítulo inicialmente presentó los requerimientos funcionales y no funcionales para el sistema de búsqueda y rescate. Seguidamente, se hace planteó una propuesta de arquitectura para SAR que pueda ser implementada con varios modelos de drones y dispositivos montados en la cabeza.

La arquitectura propuesta consta de tres aplicaciones, una aplicación móvil que se conecta directamente con el control remoto del dron, una aplicación que contiene las interfaces de MR, y finalmente la aplicación del servidor de comunicaciones, que actúa como intérprete entre las dos primeras aplicaciones. Los módulos de cada aplicación tienen un objetivo principal y pueden implementar mecanismos de ahorro de energía o sincronización de transmisión.

Aunque, no se limitan al uso de una sola tecnología o herramienta de desarrollo, como, por ejemplo, el uso de una determinada herramienta para detección de humanos en las secuencias de video. La implementación de esta aplicación se implementó usando el modelo de dispositivo HMD HoloLens creado por Microsoft, que provee MR y computación espacial. Adicionalmente, se usó un dron del modelo Phantom4 de la marcha DJI por ser un dron que provee versatilidad de vuelo y es semiautónomo, dando facilidades de uso a un piloto no experto. Finalmente, este capítulo detalla las múltiples herramientas de desarrollo que se podrían utilizar al usar dispositivos de los proveedores mencionados. Sin embargo, en los siguientes capítulos se profundiza en el uso de herramientas de desarrollo para dispositivos móviles Android, y el uso de la herramienta TensorFlow para incrementar la calidad de detección de objetos basados en métodos no tradicionales o *Deep Learning*, debido a investigaciones que lo aseveran y fueron mencionadas en el capítulo anterior.

5. Implementación de la propuesta

Este capítulo describe las etapas de implementación por las que ha pasado el sistema, etapas de las que se destacan, la integración de las aplicaciones, implementación de la seguridad física del dron, el análisis y transmisión del video y el entrenamiento del modelo de detección de humanos.

En las siguientes secciones de este capítulo se detallan a las diferentes etapas de implementación, destacando su problemática, los mecanismos que han sido implementados como solución y sus evaluaciones mediante simple observación. En la última sección de este capítulo que corresponde al reentrenamiento de un modelo de detección de humanos, se presentan estadísticas de detección que son los resultados del proceso de entrenamiento y evaluación del modelo de detección de humanos. Sin embargo, los resultados de evaluaciones sistemáticas al sistema se presentan con más detalle en el sexto capítulo de este trabajo.

5.1. Introducción

La primera etapa por la que ha pasado la implementación de la arquitectura corresponde a la integración de sus tres aplicaciones (UAV, MR y Servidor Central de Comunicaciones), integración que se ilustra en la Figura 33, ubicada en la Sección 4.5 del cuarto capítulo. La segunda etapa implementó mecanismos de seguridad física para evitar daños al vehículo aéreo. La tercera etapa profundizó la conexión directa de las aplicaciones UAV y MR, que, mediante un canal dedicado, transmite el video capturado desde la cámara del dron y proyecta en la interfaz de MR. Finalmente, una cuarta y última etapa usó métodos *Deep Learning* para reentrenar un modelo de detección de humanos con drones, donde, se ha seleccionado un conjunto de imágenes para el reentrenamiento de un modelo de detección provisto por Tensorflow.

5.2. Integración de aplicaciones

El diagrama de red ilustrado en la Figura 34 tiene por objetivo compartir internet de un proveedor de red telefónica celular hacia una red interna conformada por un computador portátil, un teléfono móvil y el dispositivo de MR. El computador portátil tiene instalado un Servidor Central, el dispositivo móvil la aplicación UAV y el dispositivo de MR la aplicación MR.

El teléfono móvil y el dispositivo de MR requieren de una conexión a internet para el registro de la aplicación en la plataforma de desarrolladores de aplicaciones para drones de la

empresa DJI y también para descargar las texturas de los mapas que son utilizados para la geolocalización del dron.

Se requiere crear una red de datos compartida mediante un teléfono móvil para poder descargar las texturas de lo mapas de zonas de difícil acceso como montañas, bosques y otros sitios, donde únicamente se tiene acceso a una red telefónica celular con servicio de red de datos móviles (3G, LTE y otros). Mediante un punto de conexión hacia una red telefónica, un teléfono inteligente Android o iOS puede compartir internet mediante la configuración de una zona de demanda de tráfico (del inglés hotspot), creando una red intermedia conformada por un dispositivo móvil y el computador portátil, ilustrada en la Figura 35.

El computador que tiene instalado el Servidor Central de Comunicaciones es un cliente de la red intermedia y a la vez es la puerta de enlace de la red interna al crear un segundo *hotspot*. El dispositivo móvil conectado al control remoto del dron y el dispositivo de MR se conectan como clientes del segundo *hotspot* para conformar la red requerida.

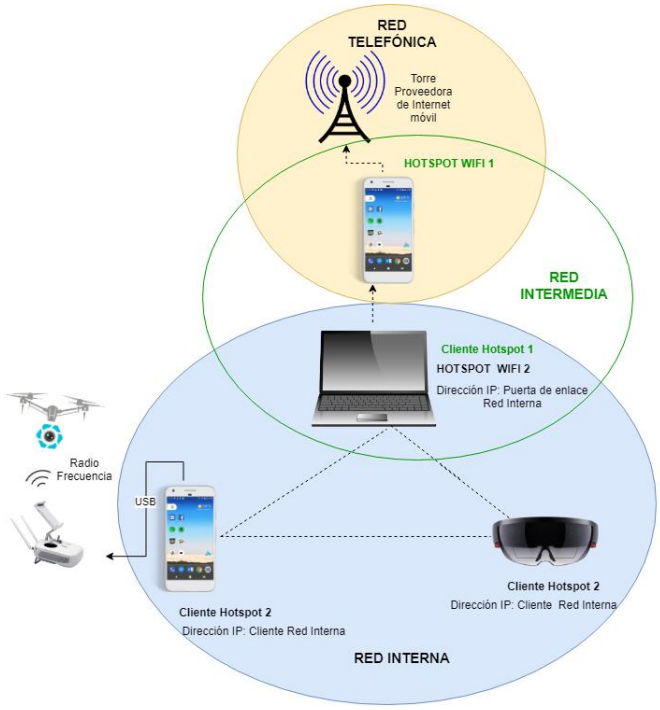


Figura 35: Diagrama de red del sistema

En este diagrama de red se utilizan dos dispositivos móviles porque el dispositivo ubicado en la red telefónica se utiliza para proveer internet, mientras que el otro dispositivo móvil ejecuta la aplicación UAV. Se puede utilizar el mismo dispositivo móvil de la red telefónica y la red interna, pero esto requeriría de un dispositivo con permisos de super usuario (del inglés super user) para configurar en su corta fuegos (del inglés firewall) el intercambio de la información por los puertos proveedores de internet.

Se configuran zonas de demanda de tráfico porque la comunicación se realiza entre clientes y servidores mediante el protocolo TCP/IP. Una vez establecida la comunicación y desplegadas las texturas de los mapas, se pueden deshabilitar la red de datos compartida en el teléfono móvil para evitar el uso innecesario de datos móviles.

5.2.1. Intercambio de información

El intercambio de información entre las aplicaciones que se describe en esta sección hace referencia a las configuraciones y control de la ruta del dron en la interfaz MR, mientras que el tratamiento de las imágenes y la sincronización de la transferencia de imágenes se tratan en la Sección 5.4. La configuración y control es inicialmente delegada a la aplicación UAV al usar un SDK provisto por la empresa DJI. Para permitir la comunicación entre el dron y el teléfono móvil, la aplicación UAV escucha constantemente mediante observadores y una conexión USB o Bluetooth si el control remoto y el dron están encendidos y se comunican entre ellos. Estos observadores se encuentran ubicados dentro del módulo de *Conexión* de la aplicación UAV que se ilustra en la Figura 36.

Los métodos de control de la ruta, como son, despegue, aterrizaje, retorno a casa, movimientos (adelante, atrás, arriba, abajo, izquierda y derecha), el giro, inclinación y cabezeo (del inglés raw, roll and pitch) de su cámara se encuentran dentro del módulo de *Control* de la aplicación UAV. Adicionalmente, este módulo de *Control* está compuesto por los métodos de configuración del dron, como son, evación de obstáculos, misión basada en coordenadas geográficas, altura máxima permitida, radio máximo permitido y coordenadas geográficas de origen.

El módulo de *Comunicación* recepta información que le envía la aplicación MR a través de de los módulos del Servidor Central de Comunicaciones que se explican más adelante. Según la información recibida, este módulo de *Comunicación* solicita la ejecución de los métodos del módulo de *Control*. Adicionalmente, el módulo de *Comunicación* implementa un mecanismo de ahorro de energía que escucha únicamente la información de los sensores del dron cuando hayan sufrido un cambio relevante en el módulo de *Conexión*.

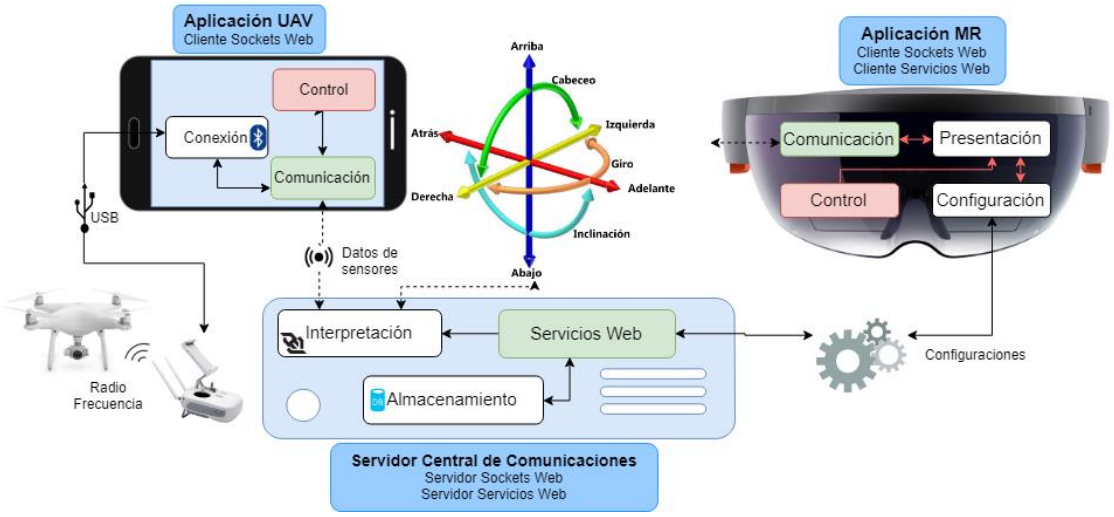


Figura 36: Módulos de integración de las aplicaciones del sistema SAR

La integración entre el módulo de *Comunicación* de la aplicación UAV y el módulo de *Interpretación* del servidor de comunicaciones es realizado mediante las tecnologías sockets web (del inglés Web Sockets) y sockets TCP, estableciendo un transporte de modo bidireccional (del inglés, duplex) sobre una arquitectura cliente – servidor. Esta conexión se establece en la pantalla de inicio en la interfaz de la aplicación móvil (Figura 37). De la misma manera, el módulo de *Comunicación* de la aplicación MR ilustrado en la Figura 36, se comunica bidireccionalmente como cliente de sockets para emitir acciones de control de ruta de vuelo y recibir información actualizada de los sensores del dron.

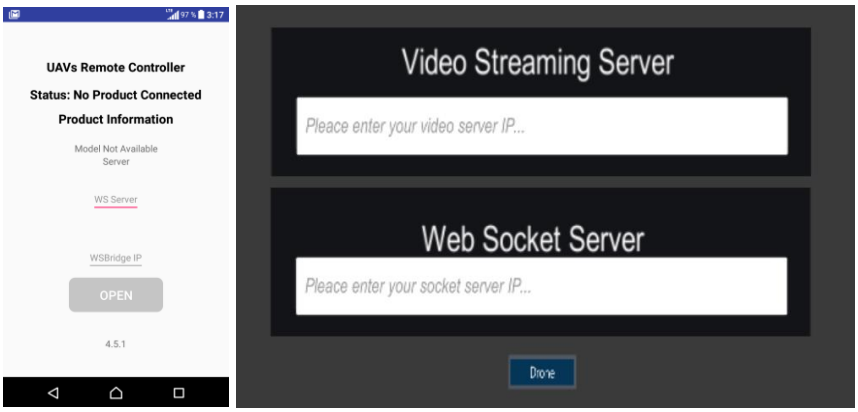


Figura 37: Integración de aplicaciones mediante direcciones IP

La información receptada por el módulo de *Comunicación* de MR trasmite sus datos a un módulo de *Presentación* que contiene las instancias de las texturas que componen la interfaz de MR. Adicionalmente, esta aplicación MR contiene los módulos de *Control* y *Configuración*. El primero de ellos implementa las funciones de control por gestos correspondientes a las texturas de la interfaz. El segundo módulo implementa otro canal de comunicación en modo de comunicación bidireccional como cliente de servicios web (del

inglés Web Services) para establecer configuraciones del dron, compartiendo información con el módulo de *Servicios Web* del servidor central de comunicaciones bajo una arquitectura de cliente - servidor.

Todo evento llamado del módulo de *Control o* el módulo de *Configuración* actualiza información en la interfaz MR a través de observadores del módulo de *Presentación.* En el caso de las funciones de control, la información de los sensores se verá forzada a ser actualizada en la pantalla, mientras que la información correspondiente a la configuración notifica en la pantalla que la información ha sido correctamente almacenada en base de datos instanciada por el módulo de *Almacenamiento* y accedida a través de los módulos *Servicios Web* e *Interpretación*.

5.2.2. Diagrama de flujo y control

Las misiones de búsqueda y rescate que proveen este sistema requieren de un intercambio ordenado de información de los equipos para establecer correctamente su comunicación, tal y como se propone en la Figura 38. Una vez configurada la red ilustrada en la Figura 35, se procede a encender el control remoto y conectar mediante el cable USB al dispositivo móvil. La conexión a internet en este punto es necesaria para el registro de la aplicación UAV, para la proyección de los mapas en las interfaces o para actualizar el firmware del dron. Si el proceso ha sido satisfactorio hasta este punto la aplicación móvil reconoce automáticamente al dron.

Una vez la aplicación se conecta con el dron, se requiere la configuración de la dirección IP del servidor de comunicaciones, el cual envía una respuesta satisfactoria en el caso de que la conexión mediante sockets se haya efectuado. Tras una respuesta satisfactoria, la aplicación MR debe configurarse para escuchar y emitir información por los canales de comunicación del servidor de comunicaciones y del servidor móvil de video.

Una vez establecida esta comunicación, la aplicación MR está apta para recibir video en tiempo real, configurar, planificar y ejecutar una misión de expedición de búsqueda y rescate. Se requiere despegar el dron para proceder a configurar una misión basada en localización por GPS. El cuadro de color rojo en la Figura 38 describe las acciones principales que se involucran para obtener una misión satisfactoria. Al ejecutar cada uno de estos eventos desde la aplicación MR, ésta es enviada al servidor central y posteriormente notificada a la aplicación móvil. El dispositivo móvil finalmente es quien controla al UAV y este retorna la información del estado actual de sus sensores mediante el mismo canal.

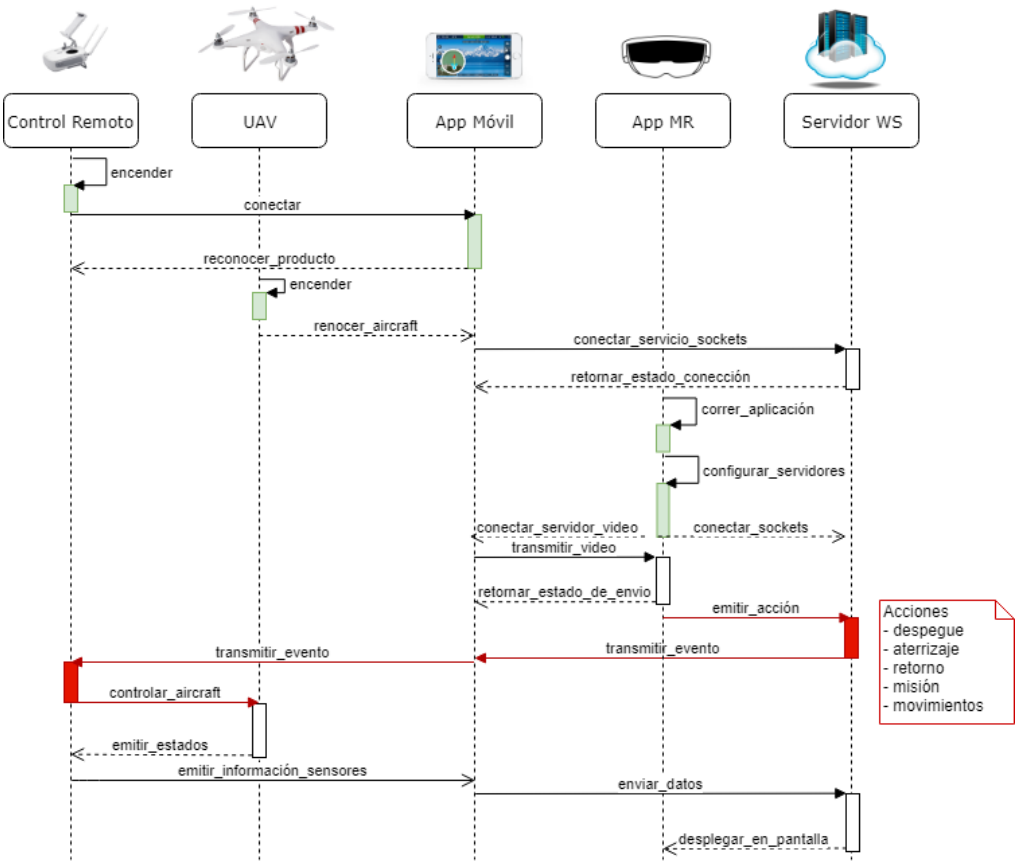


Figura 38: Diagrama de flujo y control de la aplicación SAR

5.3. Seguridad física

El piloto de un vehículo aéreo puede experimentar eventos que pongan en riesgo la seguridad física del dron. Por esta razón, en los ítems de esta sección se detallan las implementaciones de la seguridad física correspondientes a la geolocalización del dron (Sección 5.3.1), estado de la batería (Sección 5.3.2), distancia de vuelo permitida (Sección 5.3.3) y la evasión de objetos (Sección 5.3.4).

Con el fin de brindar seguridad física del dron, fue modificado el proveedor de mapas Mapbox[36] para desplegar en la interfaz MR la posición actual del dron. Por otro lado, la posición actual del dron (Sección 5.3.1) fue nuevamente revisada en la (Sección 5.3.3) para limitar la distancia y altura del vuelo. Adicionalmente, un análisis de consumo progresivo de la batería del dron (Sección 5.3.2) permite predecir el tiempo restante de vuelo. Finalmente, la activación de un mecanismo inteligente de evasión de objetos (Sección 5.3.4) permite al piloto evitar choques.

[36] https://docs.mapbox.com/mapbox-unity-sdk/docs/hololens-development.html

5.3.1. Geolocalización

El vehículo aéreo a través de su GPS provee al aplicativo móvil coordenadas geográficas en el formato de longitud y latitud. Sin embargo, para poder plasmar esta información visualmente en el dispositivo móvil y en el dispositivo MR se requiere del uso de servidores de aplicaciones de mapas como Google Maps[37], Mapbox, Bing Maps[38], OpenStreetMap[39] y otros. Se decidió usar Google Maps en el dispositivo móvil porque provee imágenes satelitales y aéreas de alta resolución con un acercamiento de hasta 23 niveles (Novapex, 2018). Sin embargo, en los dispositivos de MR HoloLens no existe soporte para Google Maps (Google Developers, 2019), mientras que Bing Maps está más orientado a la presentación de mapas 3D estáticos. Por esta razón, se escogió a Mapbox, al estar orientado a la personalización de los mapas OpenStreetMap de dominio público, mapas concebidos para aplicaciones de movilidad humana y con un acercamiento de hasta 22 niveles (Novapex, 2018).

Modificaciones en el kit mapbox-unity-sdk[40] relacionadas con la compilación para HoloLens hicieron posible que el sistema sea consciente de la posición del dron. La geolocalización se puede visualizar en las aplicaciones UAV y MR (Figura 39). Este kit no requiere ninguna modificación si un aplicativo del motor de video juegos Unity es construido para dispositivos Android (Google Developers, 2019).

Las modificaciones del kit mapbox-unity-sdk son parte de la transición de ambiente de laboratorio a un ambiente fuera de laboratorio, transición que implica que las etiquetas UNITY_EDITOR cambien a NETFX_CORE. De esta manera, la geolocalización es soportada por el editor de Unity en ambiente de laboratorio y soportada fuera de laboratorio por una cuarta versión o superior de .NET.

[37] https://cloud.google.com/maps-platform/?hl=es
[38] https://www.microsoft.com/en-us/garage/profiles/maps-sdk/
[39] https://www.openstreetmap.org/about
[40] https://github.com/mapbox/mapbox-unity-sdk

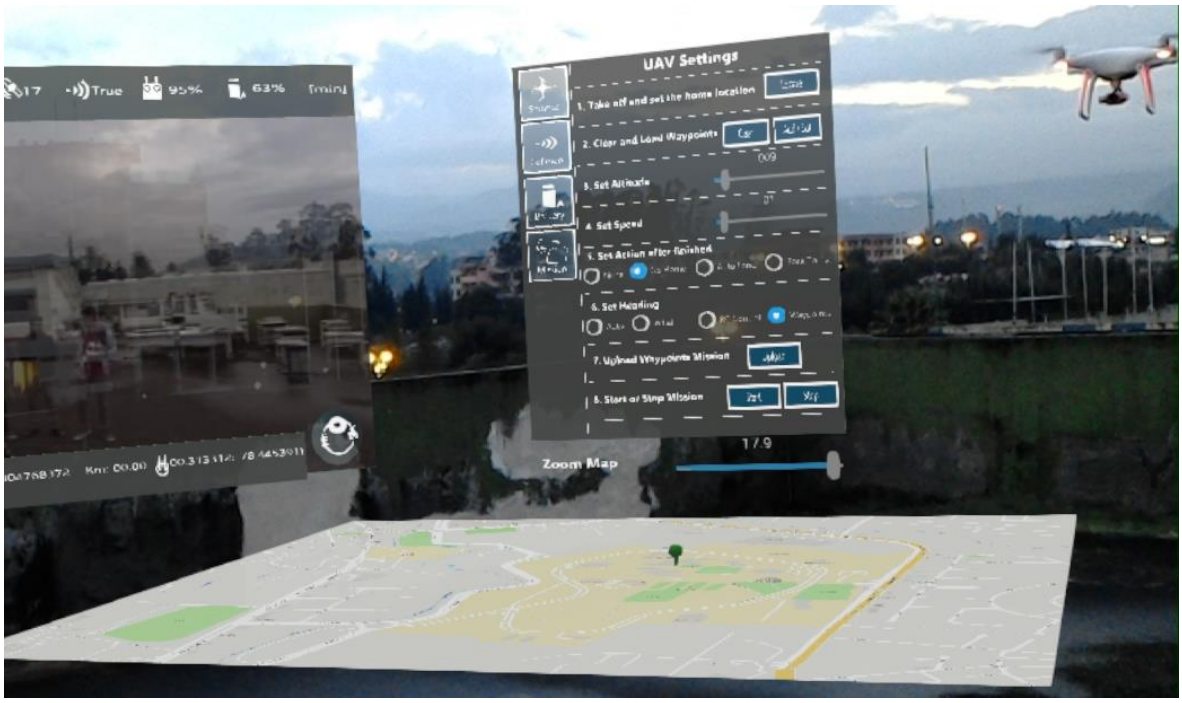


Figura 39: Ubicación actual del dron en vuelo

Adicionalmente, el cálculo de la posición actual del dron en el mapa y sobre la interfaz MR tridimensional está dada por las fórmulas 5 y 6, que proveen las coordenadas sobre los ejes *x* y *z*. Mientras que el eje *y* no presenta ninguna alteración porque no existe desplazamiento en ese eje.

$$x = \left(lat * RadioTierra * \frac{\pi}{180} - latIni\right) * metrosPorUnidad$$

Fórmula 5: Posición actual del dron en el eje x

$$z = \left(\log\left(\frac{\tan(90 + lat)}{2}\right) * \frac{\pi}{180} - longIni\right) * metrosPorUnidad$$

Fórmula 6: Posición actual del dron en el eje z

Donde *lat y long* es la latitud y longitud actual del dron; *latIni* y *longIni* es la latitud y longitud del sitio de donde despega el dron; *metrosPorUnidad* es la escala del mapa dentro de los ejes provistos por Unity. Finalmente, *RadioTierra* es el valor del radio de la tierra dado en metros, valor que equivale a 6378137 metros para el caso de la plataforma Mapbox.

5.3.2. Estado de la batería

La aplicación alerta al piloto en dos ocasiones sobre el nivel bajo de la batería del vehículo aéreo. Estas dos alertas son configurables y se han denominado prevención y riesgo. La alerta de prevención se configura entre un 15% y 50% del nivel de batería, mientras que la alerta de riesgo se configura entre un 10% y 45% de batería. Estas configuraciones se las puede realizar en el panel de la batería de la interfaz que es ilustrada en la Figura 40.

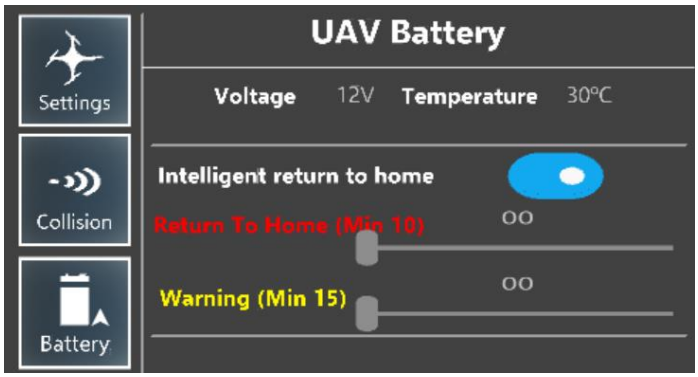


Figura 40: Configuración de alertas de batería baja

La alerta de prevención notifica al piloto que se encuentra en un nivel bajo de batería y brinda la posibilidad de retornar el dron a la posición de despegue. En el caso de no haber accionado el retorno al sitio de despegue, el dron continúa el vuelo. Pero, en el caso de recibir la notificación de riesgo en cuanto al nivel bajo de batería, el vehículo retorna al punto de despegue y notificando al piloto de que ha activado el retorno automático al punto inicial.

Complementariamente a estas alertas, la interfaz indica el tiempo de vuelo que puede soportar la batería restante. Este criterio también es usado para limitar la distancia de vuelo permitida en la programación de una ruta de vuelo. Este tiempo de vuelo está calculado con la Fórmula 7, donde la energía restante obtenida en miliamperios por hora *mAh* y el consumo de corriente actual obtenida en miliamperios *mA* son utilizados para calcular los segundos de vuelo restantes.

$$SegundosDeVuelo\ (s) = \frac{ConsumoActual\ (mA) * 3600\ (s)}{EnergíaRestante\ (mAh)}$$

Fórmula 7: Segundos de vuelo restantes

5.3.3. Distancia de vuelo permitida

El límite de distancia de vuelo permitido puede ser calculado mediante dos criterios. El primer criterio se basa en la restricción del vuelo mediante la configuración de un radio máximo que es activado en la conducción en modo principiante. Por otro lado, el segundo

criterio se basa en el cálculo de la distancia total a recorrer, el tiempo requerido para completar la ruta, la velocidad de conducción y el nivel de batería necesaria para recorrer la ruta definida mediante puntos de latitud y longitud.

El criterio del radio máximo dado por un modo principiante puede ser configurado en el panel de configuraciones del dron e ilustrado por la Figura 41, donde, es necesario activar el modo principiante y la distancia límite. Una vez activadas estas opciones, se puede configurar la distancia de vuelo permitida que va de entre 0 metros a 100 metros. Este rango ha sido definido por las especificaciones del kit de desarrollo de software del fabricante y especificado por la distancia en metros entre dos coordenadas geográficas por la fórmula de Haversine (Brummelen & Glen, 2013).

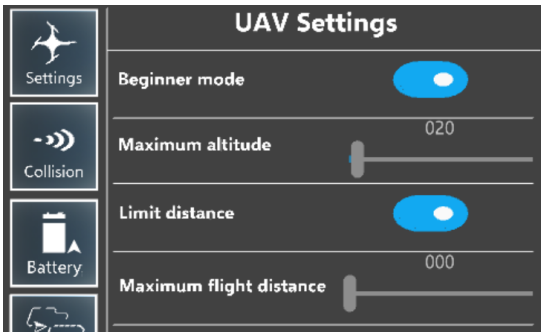


Figura 41: Configuración de vuelo del vehículo aéreo

El límite de distancia de vuelo del modo principiante puede ser usado en la programación de una ruta basada en coordenadas geográficas o en conducción manual mediante el uso del control remoto. Sin embargo, si el modo principiante no es activado, el cálculo de la distancia permitida de vuelo es calculada únicamente cuando se programa la ruta de vuelo.

El cálculo de la distancia de vuelo se realiza mediante las fórmulas 7, 8 (Haversine inverso) y 9. La Fórmula 7 define el tiempo restante de vuelo; la Fórmula 8 suma las distancias entre las coordenadas geográficas usando la fórmula inversa de Haversine, ilustradas en la Figura 42 que indican la distancia total a recorrer. Adicionalmente, la Fórmula 9 calcula el tiempo de recorrido de la ruta a una velocidad constante.

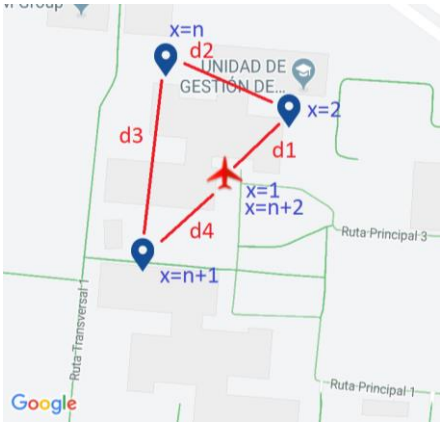


Figura 42: Puntos de referencia del vuelo

$$d = \sum_{x=1}^{x=n+2} \sqrt{sin^2\left(\frac{lat_{x+1} - lat_x}{2}\right) + \cos(lat_{x+1}) * \cos(lat_x) sin^2\left(\frac{long_{x+1} - long_x}{2}\right)}$$

Fórmula 8: Distancia total entre los puntos de referencia

En la fórmula 8, *n* indica el número de puntos de referencia graficados en el mapa, *x* el número de posición del punto de referencia, siendo *x* = 1 y *x* = n la ubicación de despegue, como lo indica la Figura 41. Adicionalmente *Lat* hace referencia a la latitud y *long* a la longitud.

La Fórmula 9 obtenida de la velocidad de Newton calcula el tiempo que toma recorrer la distancia establecida por la Fórmula 8 y la velocidad especificada para el vuelo automático. En el caso de que el tiempo que toma recorrer la distancia dada por las coordenadas geográficas sea menor a los segundos de vuelo restantes al calcular el estado de la batería por la Fórmula 7, la ejecución es aceptada, por de lo contrario, es denegada.

$$tiempo\ (s) = d/Velocidad$$

Fórmula 9: Tiempo obtenida de la fórmula de la velocidad de Newton

5.3.4. Evasión de objetos

La detección y evasión de obstáculos son características que dependen de cada dron, pues solo los drones más avanzados pueden ofrecer esta seguridad. En el caso del dron Phantom 4 usado para este trabajo, la detección y evasión de obstáculos se da únicamente cuando el dron vuela de frente, pues no cuenta con sensores en sus partes horizontales.

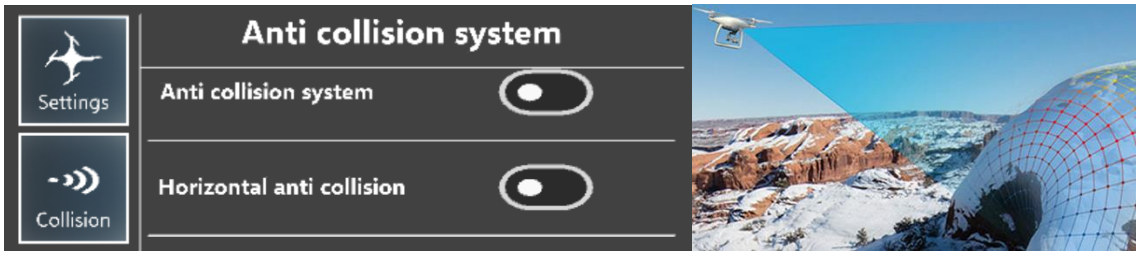


Figura 43: Configuración del sistema de anticolisión del dron Phantom 4 (DJI, 2016)

En la Figura 43, se muestra el panel de anticolisión de la aplicación MR, donde, únicamente los drones con estas características pueden habilitar la detección y la evasión de objetos a 4 metros de distancia. La evasión de obstáculos horizontalmente puede ser activado en drones más recientes de la marca DJI, como por ejemplo el Phantom 4 PRO[41].

[41] https://www.dji.com/phantom-4-pro

5.3.5. Misión geolocalizada

Todas las políticas de seguridad mencionadas fueron evaluadas al desplegar misiones de vuelo basado en puntos (Latitud, Longitud) de referencia. Estas políticas de seguridad fueron evaluadas en repetidas ocasiones en un ambiente de simulación para no comprometer el estado físico del dron en un ambiente real.

Los mecanismos de seguridad mostraron resultados satisfactorios, pues el dron no sufrió daños físicos. Evadió objetos como edificios y autos, informó al piloto que debe retornar al punto de origen cuando su batería presentó el rango de alerta de batería baja y retornó automáticamente cuando alcanzó el límite de batería baja. Por otro lado, la geolocalización permitió conocer la posición real del dron, aún cuando se encontraba fuera del rango de vista.

Cada planificación de la misión geolocalizada requirió seguir ocho pasos que se enumeran en el panel de la misión de vuelo en la interfaz MR, panel que se ilustra en la Figura 44. En un vuelo automático, inicialmente, se despega el dron para luego localizarlo con el fin de almacenar la posición actual como el punto de origen. Posteriormente, se agrega los puntos de recorrido en la aplicación móvil.

Una vez seleccionados estos puntos, se indica la altura y velocidad de vuelo. A continuación, se requiere de la configuración de la acción a realizarse cuando el dron haya recorrido todos estos puntos, así como el origen que corresponde a las coordenadas geográficas. A continuación, se cargan estas configuraciones en el dron, para finalmente iniciar la misión de búsqueda y rescate.

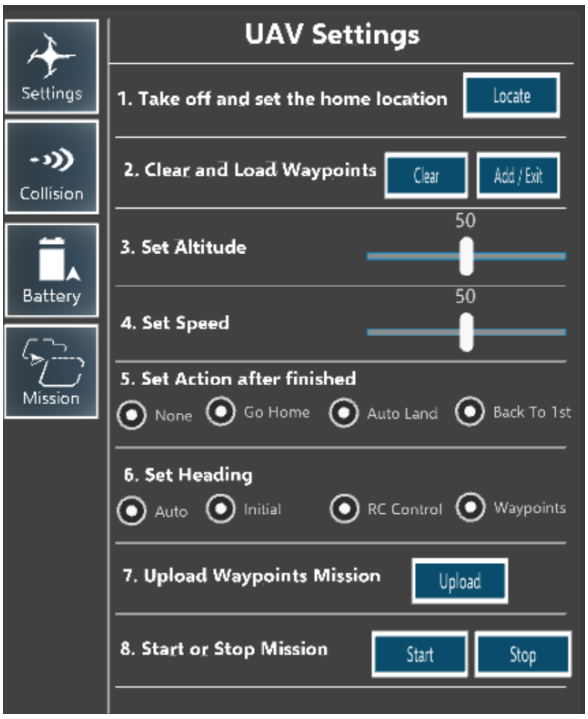


Figura 44: Configuración de una misión basada en coordenadas geográficas

5.4. Transmisión de video

Las aplicaciones UAV y MR contienen módulos especializados para la transmisión y análisis de video en tiempo real. Para este caso en especial, el período de tiempo ocurrido entre la captura y la clasificación de personas dentro de la imagen se lo denomina “retraso” y es utilizado para la sincronización de la transmisión de secuencias de video. Para poder acceder a este servicio es necesario establecer una conexión de socket TCP/IP, donde la dirección IP (del inglés Internet Protocol) del servidor video corresponde a la dirección del dispositivo móvil, como se ilustró en la Figura 36. Una vez establecida esta conexión, el video sigue el flujo que se ilustra en la Figura 45.

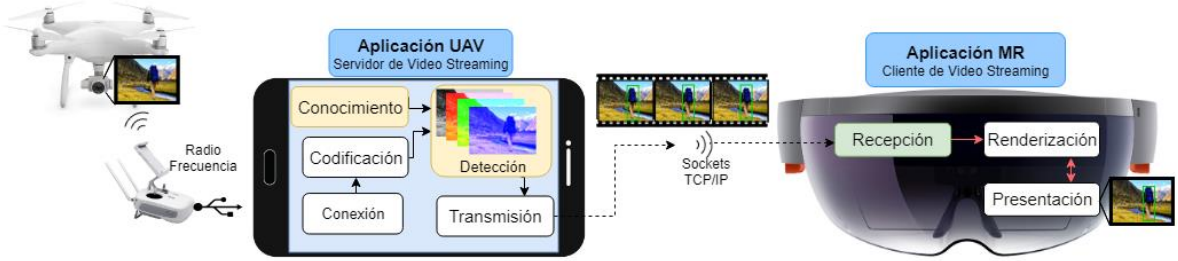


Figura 45: Integración de aplicaciones UAV y MR con el servidor de comunicaciones

Inicialmente, la cámara del dron captura imagen y con una conexión de radio frecuencia transmite la imagen sin procesar hacia el su control remoto. Mediante una conexión USB, el módulo de *Conexión* habilita al módulo de *Codificación* para receptar los datos en formato de imagen Raw. Posteriormente, esta imagen es transcodificada a los formatos superficie (del inglés Surface) y YUV240SP mediante la biblioteca de video FFmpeg. El formato superficie permite al desarrollador verificar que está recibiendo imagen al visualizarla en la pantalla del móvil, mientras que la imagen en formato YUV240SP posteriormente es reducido de tamaño y transformado formato de colores ARGB (del inglés Alpha, Red, Green and Blue), para luego ser pasado al módulo de *Detección* ilustrado en la Figura 45.

El módulo de *Detección* analiza la presencia de humanos en las imágenes y genera como salida a un mapa de bits (del inglés Bitmap). Las condiciones de coincidencia de humanos dentro de las imágenes está dado por el modelo de detección colocado en el módulo de *Conocimiento.* Entonces, si una imagen que pasa por el módulo de *Detección* contiene humanos, este módulo de detección se encarga de graficar un recuadro de color verde alrededor de cada silueta humana para transportarlo al módulo de Recepción.

Entre el canal de comunicaciones de sockets TCP/IP del módulo de *Detección* y el módulo de *Recepción* se encuenta implementado un mecanismo de reconexión del canal que permite reestablecer la conexión cuando la red inalámbrica sufra interrupciones debido a una baja intensidad de su señal, o cuando la memoria buffer haya usado todo el espacio disponible. Por esta razón, se encuentra además implementado un mecanismo de sincronización de envio y recepción de imagen basado en el PTS (del inglés Presentation Time Stamp) que evalua el tiempo en que el video es procesado por los módulos de *Codificación* y *Detección* para predecir el tiempo en que el módulo de recepción tiene que esperar para recibir la siguiente imagen, para evitar saturar el uso de la memoria buffer.

Finalmente, cuando se recibe un *Bitmap* en el módulo de *Recepción*, éste es procesado por el módulo de *Renderización* y proyectado sobre una textura instanciada en el módulo de *Presentación,* visualizando hologramas superpuestos a la realidad como en la Figura 46. En cualquier caso, una división por modulos permite mejorar aspectos en cada módulo. En el caso del módulo de *Conocimiento,* podrían mejorarse las condiciones de clasificación, especialmente en las características que definen la existencia humanos en una imagen.

Figura 46: Transmisión de video en tiempo real del sistema SAR

5.5. Detección de humanos

En esta sección se describen los cambios que se han producido dentro del módulo de *Conocimiento*. Estos cambios son fruto de un proceso de perfeccionamiento en la detección de humanos, incorporando en este módulo de *Conocimiento* a modelos de detección creados por métodos *Machine Learning* y por métodos *Deep Learning*.

En primeras instancias de este perfeccionamineto se experimetó con el uso de modelos creados con métodos *Machine Learning* como HOG, SVG y características Haar, modelos que fueron interpretados por la herramienta OpenCV. Posteriormente, fueron utilizados modelos creados com métodos *Deep Learning* e interpretados por la herramienta Tensorflow. Esta comparación de los modelos creados por métodos *Machine Learning* con los modelos creados con métodos *Deep Learning* fue realizada con el objetivo de comprobar la aseveración de (de Oliveira & Wehrmeister, 2018), quienes aseguraban que los métodos *Deep Learning* mejoraban la detección de cuerpos humanos al distinguir distintas posiciones del cuerpo.

En un inicio, se usó OpenCV con características Haar y un modelo pre entrenado por (Zhang, Bauckhage, & Cremers, 2014) que aseveraba una tasa detección de 90%. Luego, la detección mejoró al usar la combinación de HOG y SVG con un modelo preentrenado por (Meus, Kryjak, & Gorgon, 2017) que alcanzaba una *precision* del 99.4% y 93.8% de *recall* aproximandamente. Sin embargo, estos modelos pre entrenados entregaban un número significante de falsas detecciones en personas observadas desde el aire, respondiendo únicamente a la detección de cuerpos humanos captados horizontalmente como en la Figura 47.

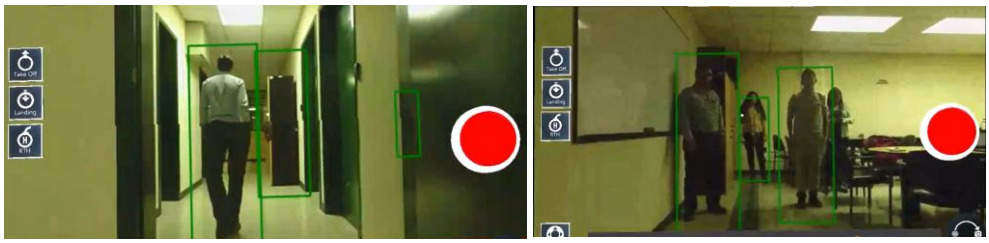


Figura 47: Detección de humanos con Haar y HOG-SVG

Posteriormente, los modelos entrenados por métodos *Deep Learning* tomados de Tensorflow Zoo[42] fueron utilizados para la comparación con los modelos de *Machine Learning*. Los modelos de TensorFlow Zoo que se describen en la Tabla 8 se escogieron por que fueron entrenados con el *dataset* MS COCO, *dataset* que incluye imágenes de personas, donde ssd_mobilnet_v1_coco presenta un valor de 0.21 en *mAP* y 30 *ms* en velocidad de

[42] https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md

clasificación. Mientras que tiny_yolo_voc_graph reconoce la presencia de personas con una probabilidad de 94% aproximadamente.

Tabla 8: modelos de referencia TensorFlow Zoo

Nombre del modelo	**Velocidad (ms)**	**mAP**	***Dataset con COCO***
ssd_mobilenet_v1_coco	30	0.21	MobileNet
tiny-yolo-voc-graph	Probabilidad ~94%		Pascal Voc

A simple vista (Figura 47), la detección mejoró considerablemente al usar modelos pre entrenados con *Deep Learning* que disponibiliza TensorFlow. Sin embargo, la detección de humanos con imágenes captadas desde las alturas y con movimientos en los ejes de rotación aún tenía fallas, como es en el caso de la imagen de la izquierda de la Figura 48.

Figura 48: Detección de objetos con el modelo pre entrenado ssd_mobilenet_v1_coco

Al no poseer un equipo físico con GPU o una CPU de gran velocidad para entrenar desde cero un modelo, se decidió utilizar herramientas disponibles en la nube (del inglés cloud), técnicas de transferencia de conocimiento (del inglés Transfer Learning) de *Deep Learning*, el algoritmo de detección de fácil entrenamiento SSD descrito en la Sección 3.3.2.1, la API de detección de objetos TensorFlow e imágenes de personas capturadas con drones.

5.5.1. Transferencia de conocimiento

La transferencia de conocimiento (del inglés Transfer Learning) es un tema de investigación de las métodos *Deep Learning* para la clasificación de múltiples clases objetos, donde se toma el conocimiento ya adquirido para aplicarlo en un tema más específico (West, Ventura, & Warnick, 2007). En este caso, se toma el conocimiento de detección de humanos con vista horizontal para incrementar la detección sobre la clase persona (del inglés person) con imágenes capturadas desde el aire.

Al mejorar la detección de una clase de objeto mediante trasferencia de conocimiento, la exactitud de otras clases de objetos puede verse afectadas. Es por eso que se recurre a técnicas complejas como la destilación del conocimiento (del inglés Distilling) (Shmelkov, Schmid, & Alahari, 2017). Sin embargo, como se requería mejorar la exactitud de una sola clase de objeto, se ha aplicado la técnica de ajuste (del inglés Fine-Tuning) sin destilación (del inglés Distilation), que, según (Ross Girshick J. D., 2014) elimina la última capa de la

red neuronal que lista las clases de objetos, y luego se reentrena la red neuronal con un subconjunto de categorías en la última capa, logrando incrementar la exactitud de la detección sobre este subconjunto de clases de objetos.

5.5.1.1. Ajuste Fino

El proceso de ajuste fine (del inglés Fine-Tuning) requiere crear conocimiento dado por nuevas imágenes y cambiar algunos aspectos de la arquitectura de la red neuronal, por lo que es una tarea que requiere capacidades de procesadores gráficos. Para ajustar finamente un modelo SSD basado en los conjunto de datos COCO y Mobilenet, utilizamos herramientas TensorFlow y recomendaciones obtenidas de (Lawrence, Malmsten, Rybka, Sabol, & Triplin, 2017), quienes mencionan que usar máquinas en la nube (del inglés cloud) integradas con GPU reducen significativamente el tiempo de entrenamiento, mientras que, al usar una máquina con CPU el entrenamiento puede tardar muchas horas, e incluso días.
Las recomendaciones de (Haridas & Sandhiya, 2018) ilustradas en la Figura 49 detallan los etapas que conlleva el proceso de ajuste fino de un modelo de TensorFlow Zoo[43].

Figura 49: Ajuste fino de modelos de Tensorflow Zoo (Haridas & Sandhiya, 2018)

En la etapa de anotación de imágenes ilustrada en la Figura 50 se utiliza la aplicación LabelImage[44]. Las etiquetas de las personas forman parte de las imágenes del *split* de entrenamiento en formato *Pascal VOC*, y son guardadas en formato XML (del inglés eXtensible Markup Language).

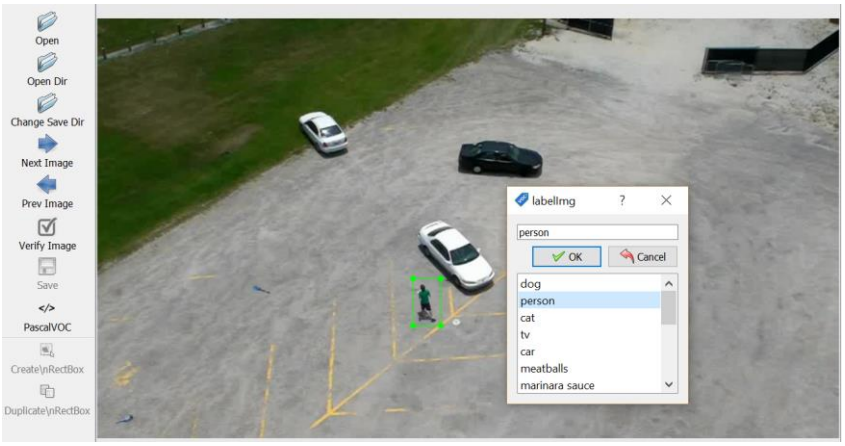


Figura 50: Anotación de instancias de la clase persona en imágenes de UCF-ARG

[43] https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md
[44] https://github.com/tzutalin/labelImg

En la preparación del mapa de etiquetas, se listan todas las imágenes y se etiqueta de acuerdo a la clase a la que pertenece. Para este caso en particular, como se utiliza una única clase (person), todos los nombres de las imágenes listadas en el archivo trainval.txt corresponden a la misma clase, tal y como se ilustra en la Figura 51.

Figura 51:Mapa de etiquetas

Con las etiquetas, el mapa de etiquetas y las imágenes de entrenamiento y evaluación se procede a la creación de los archivos en formato *TFRecord*[45]. Estos archivos son una compresión binaria propia de TensorFlow que puede tener un impacto significativo en la canalización de grandes conjuntos de información, dando por consecuencia una reducción del tiempo de entrenamiento. En una siguiente etapa, se configura la fuente de información (del inglés pipeline) que toma las direcciones de los archivos del modelo de origen para generar un nuevo modelo entrenado para las clases espefícas con las nuevas anotaciones e imágenes.

En la parte izquierda de la Figura 52 se ilustra las direcciones (*fine_tune_checkpoint, label_map_path e input_path*), además del número de clases que deben modificarse en el archivo *pipeline.config*, mientras que en la parte derecha se puede apreciar a la estructura de archivos creada para el proceso de ajuste fino.

[45] https://www.tensorflow.org/guide/extend/formats#writing_an_op_for_a_record_format

```
  fine_tune_checkpoint: "gs://drones-dataset/data/model.ckpt"
  from_detection_checkpoint: true
  num_steps: 200000
}
train_input_reader {
  label_map_path: "gs://drones-dataset/data/sw_label_map.pbtxt"
  tf_record_input_reader {
    input_path: "gs://drones-dataset/data/sw_train.record"
  }
}
eval_config {
  num_examples: 8000
  max_evals: 10
  use_moving_averages: false
}
eval_input_reader {
  label_map_path: "gs://drones-dataset/data/sw_label_map.pbtxt"
  shuffle: false
  num_readers: 1
  tf_record_input_reader {
    input_path: "gs://drones-dataset/data/sw_val.record"
```

Figura 52: Fuente de información y estructura de archivos para afinamiento del modelo

Al reentrenar un nuevo modelo de detección se configuran los parámetros (número de nodos e iteraciones) para el entrenamiento, tomando en cuenta el hardware a utilizar. Se pueden utilizar máquinas locales o máquinas virtuales en la nube con GPU, CPU o un circuito integrado específico para aprendizaje automático llamado TPU (del inglés Tensor Processing Unit) y creado por Google[46].

5.5.1.2. Configuración de hardware

En esta última etapa del ajuste fino del modelo de detección se puede optar por configurar recursos locales o recursos en la nube. En el primer caso, al utilizar recursos propios, se pueden tener instaladas GPU o únicamente CPU, siendo las CPU mucho más lentas que las GPU. Pero, si se desea acelerar el proceso de entrenamiento y reducir el proceso a pocas horas, es necesario usar por recursos en la nube que cuenten con GPU o TPU.

Al optar por el uso de recursos en la nube con TensorFlow se pueden tener dos opciones, utilizar recursos gratuitos como Google Colaboratory[47] y Kaggle[48], o utilizar recursos pagados como Google Cloud Platform[49], Amazon SageMaker[50] o Azure Notebooks[51], donde únicamente Google ML provee 300$ de saldo si se usa los servicios de la nube por primera vez.

[46] https://cloud.google.com/blog/products/gcp/google-supercharges-machine-learning-tasks-with-custom-chip
[47] https://colab.research.google.com
[48] https://www.kaggle.com/akashkr/tensorflow-tutorial
[49] https://cloud.google.com/ml-engine/
[50] https://aws.amazon.com/es/sagemaker/
[51] https://notebooks.azure.com/help/jupyter-notebooks

Las ventajas y desventajas de las plataformas en la nube compatibles con TensorFlow se detallan en la Tabla 9. En este trabajo se han experimentado con Google Cloud Platform y Google Colaboratory en dos entrenamientos que se detallan en la Sección 5.5.1.3. Un entrenamiento de 4 horas aproximadamente, que incluía el servicio de almacenamiento y procesamiento de imágenes con GPU en Google Cloud Platform tuvo un costo de 73$, mientras que el entrenamiento de 5 horas aproximadamente en Google Colaboratory no tuvo ningún costo. En Google Cloud Platform, el estado del aprendizaje se visualizó en tiempo real, mientras que en Google Colaboratory tenía ciertos retrasos. Sin embargo, Google Colaboratory y Google Cloud Platform muestran los resultados esperados al final del proceso de entrenamiento. Tras la experiencia obtenida en la Sección 5.5.1.3, podemos concluir que la mejor opción es utilizar la herramienta de Google Colaboratory, pues no tiene costo y requiere únicamente del almacenamiento de los datos en Google Drive cuando el entrenamiento dura más de 12 horas.

Tabla 9 : Plataformas para Deep Learning en la nube

Plataforma	**Ventajas**	**Desventajas**
Google Colaboratory	Disponibilidad de 12 GB de RAM y 358 GB de espacio. Tiene conexión con almacenamiento de Google Drive y viene con algunas bibliotecas preinstaladas de aprendizaje automático. Además, puede usar *dataset* públicos de Kaggle. No requiere la instalación de software controlador de GPU y TPU.	Cuando transcurren 12 horas, sus datos que no hayan sido almacenados en Google Drive se pierden, requiriendo descargarlos nuevamente.
Kaggle kernels	Provee 2 Cores de CPU y 14GB de memoria RAM cuando se usa GPU. Maneja versiones de *kernel* que pueden públicos o privados. Además, almacena y comparte *datasets*. No requiere la instalación de software controlador de GPU y TPU.	Requiere tunelización para visualizar las métricas de evaluación de Tensorboard. Solo provee recursos únicamente desde notebooks Jupyter. Solo puede utilizar un nodo para el entrenamiento.
Google Cloud Platform -ML Engine	Se puede ampliar los recursos GPU, TPU y almacenamiento. Permite también formar clústeres (más de un nodo) para acelerar el entrenamiento.	Para acceder a su salgo de 300$ por utilización de primera ocasión se requiere de una tarjeta de crédito válida. Pero si este saldo se ha agotado, sus precios son elevados.

5.5.1.3. Entrenamiento y evaluación

En este trabajo se experimentaron dos ajustes finos del modelo ssd_mobilnet_v1_coco disponibilizado por TensorFlow Zoo. El primer ajuste fino incluyó únicamente el uso de imágenes capturadas desde el aire del *dataset* UCF-ARG. Mientras que, el segundo ajuste fino incluyó imágenes con acciones de rescate, deportes acuáticos, actividades de montaña, personas cubiertas parcialmente por la floresta e imágenes de deportes aereos obtenidos del *dataset* MPII Human Pose.

En la obtención del *dataset* para el primer ajuste fino se eligieron 440 videos de 10 acciones humanas del *dataset* UCF-ARG, pues contenía imágenes capturadas en menos de 30.45 metros de altitud. Posteriormente, se reprodujo todos los videos que contenían una tasa de 29.97 fps con una calidad de 960x544 píxeles, de los cuales se capturó y guardó únicamente un *frame* por segundo en formato de imagen JPEG. Con este proceso se consiguieron 1960 etiquetas de personas dentro de 1947 imágenes para el *split* de entrenamiento y 493 imágenes para el conjunto de datos de evaluación.

Figura 53: Imágenes capturadas de los videos del dataset UCF-ARG

El primer ajuste fino utilizó máquinas virtuales de Google Cloud Plaftorm[52] para formar un clúster distribuido en la nube. Aquitectura que incluyó el uso de procesadores gráficos Nvidia Tesla K80 con software controlador CUDA y una instancia de almacenamiento (del inglés bucket) para almacenar los datos. Luego de transcurrir 4 horas aproximadamente de entrenamiento, el modelo obtuvo un valor de *precision* de 0.89. Sin embargo, el uso del modelo entrenado con las imágenes de la Figura 53 fue descartado, pues sus imágenes eran muy similares unas de otras y no incluía la detección de personas con diferentes posiciones del cuerpo o en zonas de riesgo.

Como solución a esta problemática se decidió realizar un segundo ajuste fino del modelo ssd_mobilnet_v1_coco. En la obtención de este segundo *dataset* se eligieron 292 imágenes con figuras humanas del *dataset* MPII Human Pose. Posteriormente, estas imágenes se redujeron la calidad de imagen a una resolución de 600x388 píxeles. Para asegurar que el modelo reconociera diferentes pocisiones del cuerpo humano se capturaron 763 imágenes

[52] https://cloud.google.com

con el dron a una altura máxima de 30 metros, con diferentes rotaciones de la cámara y reducidas a la misma calidad de 600x388 píxeles.

Figura 54: Imágenes del dataset MPII Human Pose e imágenes capturadas con el dron.

Se decidió capturar imágenes debido a no tener acceso a un *dataset* para el reconocimiento de humanos en escenarios de riesgo o parcialmente cubiertos. Al final del proceso de obtención de las imágenes, un total de 1055 imágenes en formato JPG fueron separadas en los *splits* de entrenamiento y evaluación. El *split* de entrenamiento se conformó por 855 imágenes con 5160 etiquetas de personas, etiquetas que incluían imágenes de personas con resoluciones superiores a 32x32 píxeles. Por otro lado, el *split* de evaluación contuvo 200 imágenes sin etiquetas de personas.

En este segundo ajuste fino se utilizó una máquina virtual de Google Colaboratory con 12.72 GB de memoria RAM, 358.27 GB de disco, una GPU Tesla K80. Esta máquina virtual contenía la versión 13.1 de TensorFlow con Python 3.6. Complementariamente, se instaló Tensorboard para ilustrar el proceso de aprendizaje que se ilustra en la Figura 55. Al transcurrir los 13390 pasos de entrenamiento (Figura 56) en un total de 4 horas y 50 minutos, el modelo alcanzó un 0.89 para *mAP* y un 0.5572 para *recall* promedio. Aunque el valor de *precision* no varía con el modelo obtenido en el anterior ajuste fino, éste modelo obtenido en el segundo ajuste fino incluyó el uso de imágenes con humanos que se encuentran en escenarios más parecidos a los escenarios de búsqueda y rescate.

Figura 55: Entrenamiento, detecciones (izquierda) y cuadros de verdad (derecha)

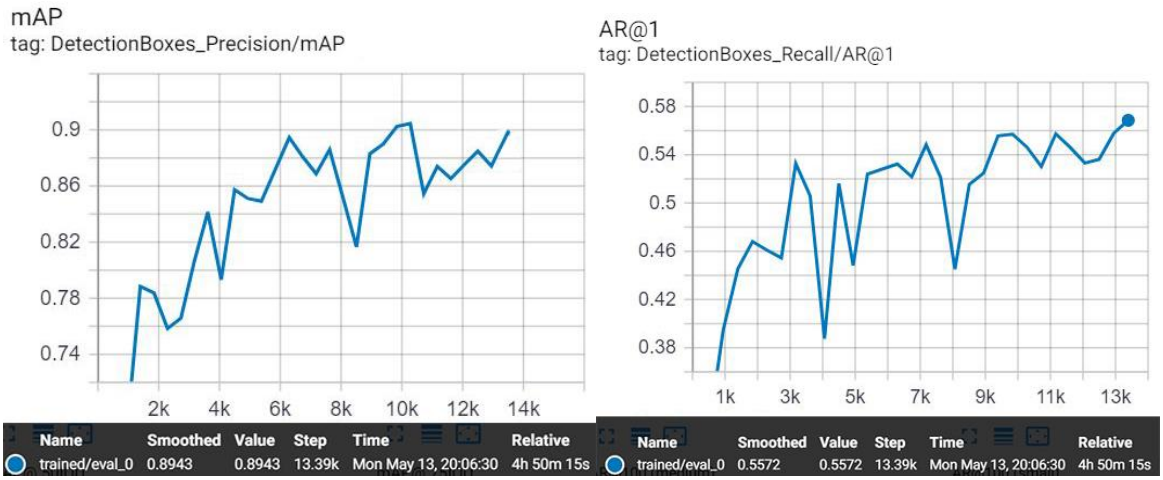


Figura 56: Resultados mAP y Recall del ajuste fino

5.6. Síntesis

Este capítulo describió las etapas que conllevó la implementación del sistema. En una primera etapa se integró las aplicaciones al configurar la red de comunicación, al implementar el intercambio de información entre los módulos de la arquitectura presentada en el capítulo anterior y al definir el flujo de configuración del sistema.

En una segunda etapa se implementaron medidas de seguridad física para conservar la integridad física del dron. Estas medidas de seguridad analizaron la posición actual del dron, estado de la batería, y el sistema inteligente de evasión. Las medidas de seguridad han sido evaluadas mediante la observación, donde, los resultados fueron satisfactorios al conservar la integridad física del dron cuando se ejecutaron misiones de vuelo basados en geolocalización dentro de una distancia permitida.

Una cuarta y quinta etapa abordaron la transmisión del video y el análisis de sus imágenes, en donde, fueron probados métodos de detección *Machine Learning* y *Deep Learning*. Pero, finalmente se optó por utilizar técnicas de transferencia de conocimiento de *Deep Learning* para afinar en dos ocasiones un modelo de detección de humanos provisto por TensorFlow Zoo. Como resultado de estas experiencias de ajuste fino, se tomó en cuenta el segundo ajuste, pues considera diferentes posiciones del cuerpo humano captados desde el aire y diferentes escenarios de rescate que no consideraba el *dataset* utilizado en el primer ajuste fino.

6. Evaluaciones y resultados

Este capítulo presenta resultados de evaluaciones de la calidad del servicio y calidad de experiencia de la transmisión de video. Sin embargo, las configuraciones correspondientes a los cuadros por segundo (fps, del inglés frames-per-second) y resolución de la imagen fueron previamente analizados para maximizar el uso de los recursos de *hardware*. Adicionalmente, este capítulo muestra un estudio de la percepción de usuarios expertos e inexpertos en el pilotaje de drones y con el uso de dispositivos montados en la cabeza HDM de realidad mixta MR. Finalmente, usando métodos de clasificación binaria, los índices de detección de humanos (personas) del modelo reentrenado (Sección 5.5.1.3) fueron evaluados al despegar una misión de vuelo y capturar imágenes del campus de la Universidad de las Fuerzas Armadas ESPE.

En la primera sección de este capítulo, se muestran las configuraciones de la transmisión de video. Estas configuraciones fueron obtenidas mediante un análisis de la resolución de la imagen y los *fps*, con la finalidad de obtener un cierto tiempo de reproducción que no presente interrupciones del canal de comunicaciones y que utilice al máximo los recursos hardware. En la segunda sección de este capítulo se describen las pruebas que permitieron incrementar la calidad de transmisión de video y la percepción de los usuarios. Posteriormente, estas configuraciones del video fueron analizadas con el conjunto de tecnologías de calidad de servicio (QoS, del inglés Quality of Service).

La calidad de servicio QoS es un conjunto de tecnologías que proporcionan la capacidad de administrar el tráfico de red de manera rentable y mejorar las experiencias de usuario (Murazzo, et al., 2013). Posteriormente, la percepción del servicio de transmisión de video fue evaluado mediante estudios de calidad de experiencia (QoE, del inglés Quality of Experience). Según la Unió Internacional de Telecomunicaciones UIT, la calidad de experiencia QoE es el grado de aceptabilidad de un servicio o aplicación, mediante la percepción subjetiva de los usuarios finales (UIT-T_G.1080, 2008).

Seguidamente, en la tercera sección de este capítulo se presentan los resultados de estudios de experiencia y percepción de los usuarios (estudiantes de informática y pilotos expertos de drones de la empresa DJI). Quienes, llenaron un cuestionario y evaluaron su accesibilidad a un HMD de realidad mixta MR, así como también, su percepción de la interfaz de la aplicación MR descrita en la Sección 4.5.2.

Finalmente, la cuarta sección de este capítulo presenta los resultados de la detección de humanos dentro del campus de la Universidad de las Fuerzas Armadas ESPE, con el uso de imágenes capturadas con el dron Phantom 4. Evaluación que fue realizada mediante métodos de clasificación binaria y con imágenes con humanos e imágenes sin humanos.

6.1. Servicio de transmisión de video

Los servicios de transmisión de video de las aplicaciones SAR presentan restricciones al presentar la imagen en tiempo real, pues el servicio incluye el análisis de las imágenes, servicio que puede demorar dependiendo de la unidad de procesamiento del dispositivo, resolución de la imagen y la velocidad del modelo de detección. Para examinar la calidad de los servicios de video se recurre a las evaluaciones de calidad QoS. El soporte de QoS se ha convertido en un factor importante dentro de los servicios emergentes de video actuales. De esta manera es posible trabajar los servicios de telecomunicaciones sobre una plataforma de red inalámbrica o móvil.

En una red de comunicación del servicio de transmisión de video, se requiere de una aceptable calidad video de un punto a otro, algunos factores estudiados por la calidad de video son, el retraso, la pérdida de información y la información fuera de orden (Murazzo, et al., 2013). Previamente a las evaluaciones de QoS, se han realizado una serie de pruebas de análisis y transmisión de video con el fin de maximizar el uso de los recursos hardware y proveer una mejor experiencia a los usuarios.

6.1.1. Análisis y transmisión de video

Las primeras pruebas realizadas al análisis y transmisión de video consistieron en variar la resolución de la imagen y los *fps* para obtener una transmisión con la mejor resolución, con el mayor valor de *fps* y una transmisión de 20 minutos sin interrupciones. Estos 20 minutos están limitados por el tiempo de vuelo máximo que provee la batería del dron Phantom 4 (DJI, 2016).

El dron utilizado en este proyecto provee varias configuraciones para *fps* y calidad de imagen. La resolución mínima de imagen corresponde a 1280x720 píxeles, mientras que la menor velocidad de captura de imagen corresponde a 40 *fps* (DJI, 2016). Sin embargo, esta cantidad de datos tuvo que ser reducida para poder analizarla con el modelo de detección de humanos.

En la Tabla 10 se puede visualizar 14 evaluaciones de análisis y transmisión de video. En estas evaluaciones se utilizó el modelo de detección ssd_mobilenet_v1_coco de TensorFlow Zoo[53], variando los valores de *fps* y resolución de la imagen con el fin de obtener una transmisión de video de 20 minutos.

[53] https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md

Tabla 10: Evaluaciones de fps y resolución de imagen

Evaluación #	Fps	Calidad de imagen	Tiempo
1	1.7142857	27%	04:19
2	1.7142857	25%	08:10
3	3	32%	01:28
4	3	30%	03:11
5	2	32%	15:27
6	2	23%	20:00
7	2	24%	20:00
8	2	25%	20:00
9	2	38%	00:21
10	2	37%	00:50
11	2	39%	00:54
12	2	37%	01:00
13	1.875	33%	01:16
14	1.875	34%	00:56

Los resultados de las evaluaciones de transmisión de la Tabla 10 se ilustran en la Figura 57, donde se puede apreciar que las evaluaciones 6, 7 y 8 obtuvieron una transmisión satisfactoria al obtener 1200 segundos (20 minutos) de reproducción.

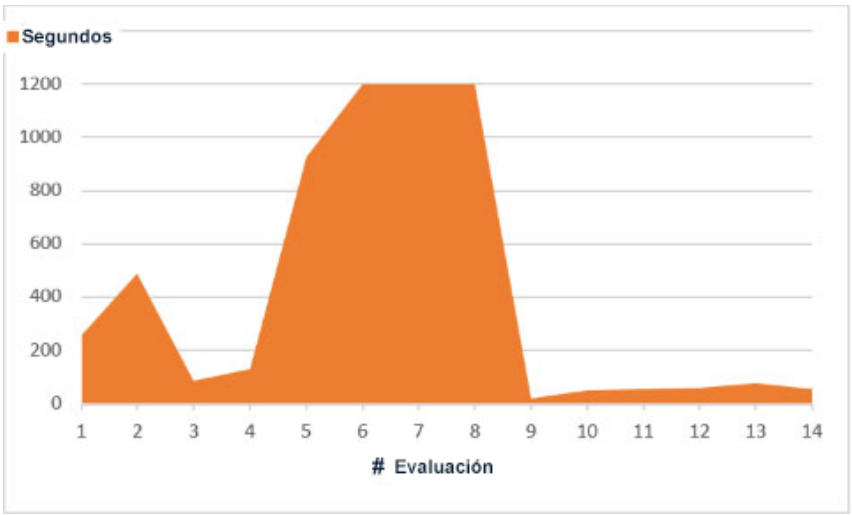


Figura 57: Segundos de transmisión por evaluación

Al analizar estas tres transmisiones satisfactorias se dedujo que las transmisiones con una resolución de imagen menor al 25% y con una velocidad menor a 2 *fps* no presentan interrupciones en canal de comunicaciones ni sobrecargan el trabajo de la unidad de

procesamiento del dispositivo móvil. Por esta razón, en futuras evaluaciones se usó una resolución de 320x180 píxeles para la imagen. Por otro lado, el módulo de *Codificación* de la aplicación móvil UAV presentada en la Sección 4.5.1, utilizará únicamente 2 *fps* para el análisis y transmisión de video.

6.1.2. Calidad de servicio

La calidad del servicio de transmisión de video requiere el análisis de algunos aspectos dentro de la red de comunicaciones, entre estos aspectos se encuentran al retraso de la transmisión de la imagen, la pérdida de información y la información fuera de orden (Murazzo, et al., 2013).

La red de comunicaciones que utilizamos para la transmisión de video está compuesta por dos nodos (emisor y receptor). La aplicación móvil UAV descrita en la Sección 4.5.1 cumple el rol de emisor, mientras que la aplicación MR descrita en la Sección 4.5.2 cumple el rol de receptor.

Esta comunicación entre el emisor y receptor es realizada mediante sockets TCP/IP (Sección 5.4), en donde cada paquete enviado corresponde a una secuencia de bytes que corresponden a la imagen y su dimensión. La sincronización de envío y recepción que se ha implementado se basa en un mecanismo basado en el protocolo (PTS, del inglés Presentation Time Stamp), por lo que se analizó manualmente el envío y recepción de datos en la aplicación móvil UAV y en la aplicación MR.

La sincronización de envío y recepción de datos se basa en el tiempo de retraso $T_{Retraso}$, tiempo comprendido entre la codificación, la compresión de la imagen en formato YUV420, la detección de humanos, el tiempo que dura la transmisión y el tiempo de renderización. Este tiempo de retraso $T_{Retraso}$ se ilustra en la Figura 58 y se calcula con la Fórmula 10.

Figura 58: Cálculo del Tiempo de Retraso

$$T_{Retraso} = T_{Codificación} + T_{Compresión} + T_{Detección} + T_{Transmisión} + T_{Renderización}$$

Fórmula 10: Cálculo del Tiempo de Retraso

Para calcular los tiempos de la Figura 58 se utilizaron variables de tiempo en la aplicación UAV. Estas variables definieron el inicio y el fin de cada actividad (codificación, compresión, detección, transmisión y renderización). De esta forma, el tiempo de cada

actividad fue definido por la Fórmula 11, como la diferencia entre el tiempo de conclusión y el tiempo inicial.

$$T_{Actividad} = T_{Conclusión} - T_{Inicial}$$

Fórmula 11: Tiempo de las actividades de codificación, compresión y detección

Datas estas definiciones, se calculó el valor de estas actividades al transmitir video desde la aplicación UAV hacia la aplicación MR durante 4 minutos, donde 506 imágenes fueron transmitidas. La media de los tiempos de cada actividad se definió en la Fórmula 11, donde la media de cada actividad está dada por la sumatoria de los tiempos todas sus *k* iteraciones dividida por *k*.

$$T_{mActividad} = \frac{\sum_{n=1}^{n=k} T_{Conclusión\_k} - T_{Inicial\_k}}{k}$$

Fórmula 12: Tiempo medio por actividad

La media de tiempo obtenida para la codificación fue de 35.12 ms, 65.93 ms para la compresión, 38.23 ms para la detección, 151.32 ms para la transmisión y 92.45 ms para la renderización. La media de tiempo para el retraso $T_{Promedio\_Retraso}$ definida en la Fórmula 11 se definió como la sumatoria de todas las actividades más un mínimo *Error* de cálculo, dando como resultado un valor de 383.05 ms aproximadamente.

$$T_{mRetraso} = T_{mCodificación} + T_{mCompresión} + T_{mDetección} + T_{mTransmisión} + T_{mRenderización} \pm Error$$

Fórmula 13: Tiempo medio del retraso

La calidad del servicio de transmisión de datos se encuentra dentro del rango recomendado y, en la prueba de 4 minutos, no se registraron pérdidas de información ni paquetes fuera de orden. Los valores para el retraso se encontraron dentro del rango recomendado por la (UIT-T_G.114, 2013), que menciona, que independiente del tipo de aplicación, la planificación de las redes de comunicación no debe superar un retraso máximo de 400 ms. Sin embargo, los colaboradores de Cisco (Lewis & Pickavance, 2019) especifican que el retraso debe ser imperceptible por parte del usuario, por lo que se acude a los estudios de la calidad de experiencia QoE.

6.2. Experiencia y percepción de usuarios

Las evaluaciones de experiencia y percepción de los sistemas computacionales son realizadas por pruebas con usuarios objetivo. Estas evaluaciones pueden estar centradas a la tecnología o centradas a la interacción del humano con la computadora (HCI, del inglés Human Computer Interaction). Cuando se evalúa a la tecnología, se habla de calidad de experiencia (QoE, del inglés Quality of Experience), mientras que, cuando se habla de HCI

se hace referencia a la experiencia de los usuarios al interactuar con la aplicación (UX, del inglés User eXperience).

Las evaluaciones de QoE se diferencian de UX básicamente por la forma de ser medidas. En UX predominan los métodos cualitativos, haciendo referencia a estudios psicológicos y de interpretación. Mientras que en QoE predominan los métodos cuantitativos (Möller & Raake, 2014). Existen tres enfoques de evaluación diferentes cuando se habla de calidad de experiencia de video, las cuales son, objetiva, subjetiva e híbrida (Amin & Rebeen, 2016).

Las pruebas objetivas son un método indirecto para medir la calidad de experiencia, pues los estudios de video se correlacionan con la media obtenida de las evaluaciones de los observadores. Por otro lado, las pruebas subjetivas, son un método directo de medición, pues la observación del resultado final es el resultado final. Finalmente, las evaluaciones híbridas son una combinación de las pruebas subjetivas con parámetros objetivos (Amin & Rebeen, 2016).

En este trabajo, se elaboró un cuestionario que incluyeron preguntas relacionadas con QoE y UX. Se usaron las evaluaciones objetivas de QoE para evaluar la nitidez de la imagen transmitida, mientras que, para UX, se desarrollaron preguntas que evaluaron la accesibilidad del usuario a un HMD de MR y su percepción de la interfaz de la aplicación de realidad mixta MR.

Las pruebas subjetivas de QoE se basan en el uso de la métrica de evaluación directa (MOS, del inglés Mean Opinion Score). El cálculo de MOS está dado por el promedio de la sumatoria de todas las calificaciones de los usuarios. En la Fórmula 14, *C* corresponde al nivel de una calificación individual para un estímulo dado y *N* corresponde al número de calificaciones.

$$MOS = \frac{\sum_{n=1}^{N} Cn}{N}$$

Fórmula 14: Cálculo de la Métrica Mean Opinion Score (Tavakoli, 2015)

MOS expresa un juicio de calidad promedio de un grupo que es parte de una misma condición de prueba. Su resultado se basa en una escala original de cinco niveles (Excelente, Buena, Regular, Pobre, Mala), niveles que son presentados de acuerdo con su calificación en la Tabla 11 (Tavakoli, 2015).

Tabla 11: Niveles, calificaciones y discapacidades usadas para MOS (Tavakoli, 2015)

Nivel	Calificación	Discapacidad
5	Excelente	Imperceptible
4	Buena	Perceptible pero no molesta
3	Regular	Ligeramente molesta
2	Pobre	Molesta
1	Mala	Muy molesta

6.2.1. Cuestionario

El cuestionario que fue utilizado para las evaluaciones de UX y QoE se basó principalmente en un estudio de percepción de realidad aumentada AR realizado por (Gandy, et al., 2010). Este cuestionario que es presentado en la Tabla 12 consta de 16 preguntas que evalúan diferentes temas de la aplicación. Entre estos temas se encuentran a la nitidez del video (parte de QoE), accesibilidad hacia un HMD de MR y sus grados de Inmersión, Usabilidad y Distracción con la interfaz de la ampliación de MR.

Tabla 12: Cuestionario de experiencia y percepción para usuarios

#	Pregunta
1	¿Qué tan apropiad@ estuviste del rol de piloto?
2	¿Necesitaste ayuda para manipular el dispositivo? **No la aplicación**.
3	¿Qué tan consciente estabas de tus acciones?
4	¿Pudiste anticipar qué sucedería? Antes de ejecutar
5	¿Qué tan bien pudiste manipular los objetos?
6	¿Qué tan bien te concentraste en la tarea asignada?
7	¿Cuán atraíd@ estuviste con la experiencia? Te sentiste alguien más o un piloto en formación.
8	¿Experimentaste demoras al accionar eventos?
9	¿Parece más interesante esta interfaz que una interfaz de un dispositivo móvil?
10	¿Pudiste estar consciente de los eventos del mundo real y mundo virtual al mismo tiempo?
11	¿La información proporcionada fue consistente?
12	¿Qué tan fácil fue aprender a ejecutar una misión de búsqueda y rescate?
13	¿Fue fácil ejecutar la misión de búsqueda y rescate?
14	¿Has tenido experiencia con otros dispositivos de realidad virtual y/o realidad aumentada?
15	¿Tuviste problemas físicos al ejecutar la tarea?
16	¿Qué te pareció la nitidez del video?

La accesibilidad hacia un HMD de MR analiza la capacidad o experticia de los usuarios para controlar las aplicaciones de MR. La inmersión evalúa la reacción emocional del usuario hacia el entorno virtual en términos de sentimiento, como si este utilizador fuera actualmente parte de la realidad que se la presenta (Cavalcante, 2017).

De todos los objetivos de usabilidad de (Preece, Rogers, & Sharp, 2005), esta evaluación comprende a los objetivos de facilidad de aprendizaje, utilidad y el grado de placer de interacción con la interfaz. Finalmente, las preguntas relacionadas con distracción obtenidas de (Gandy, et al., 2010), son evaluadas en contra posición al grado de inmersión. Las preguntas de la Tabla 12 están agrupadas de acuerdo con los temas propuestos en la Tabla 13.

Tabla 13: Temas de evaluación y preguntas del cuestionario

Grupo	Temas de evaluación	Preguntas
1	Inmersión	1, 3, 4, 7, 8, 9, 10
2	Accesibilidad a un HMD de MR	2, 14, 15

3	Usabilidad	12, 13
4	Distracción	6, 11
5	Nitidez del video (QoE)	16

Las respuestas del cuestionario tienen cinco opciones (1 = Nada, 2 = Mínimo, 3 = Poco, 4 = Ligeramente y 5 = Mucho) que permiten realizar un cálculo inspirado en la métrica de calidad MOS, exceptuando las preguntas 3, 15 y 16. La pregunta 3 presenta su mejor resultado en la respuesta (1 = Nada). Por otro lado, la pregunta 15 hace referencia a los problemas físicos (dolor de cabeza, mareo, fatiga visual u otros) que pueda sufrir el usuario. Por último, la pregunta 16 corresponde al grupo de QoE que tiene las opciones de respuesta de métrica de calidad MOS de la Tabla 11.

6.2.2. Escenario de pruebas

Previo a la ejecución de las pruebas, se capacitó en un tiempo aproximado de dos horas a 20 estudiantes de la carrera de Tecnologías de la Información de la Universidad de las Fuerzas Armadas ESPE sobre el uso del dispositivo HMD HoloLens y las acciones (aterrizaje, despegue y retorno a casa) correspondientes al pilotaje de drones DJI. Con la ayuda del video juego RobotRaid de Microsoft, la capacitación del uso del dispositivo HMD tuvo por objetivo practicar el control de la interfaz mediante gestos de la mano (florecimiento, listo, tocar y mantener) ilustrados en la Figura 59.

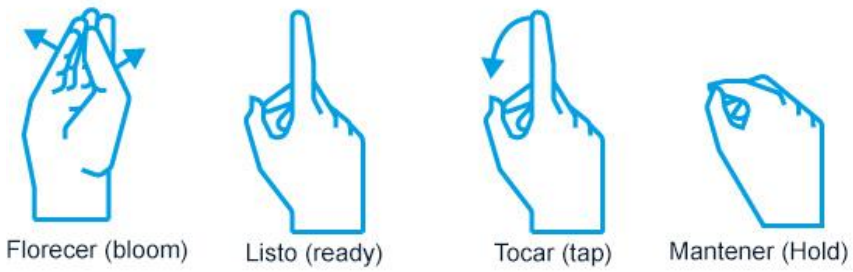


Figura 59: Gestos para la manipulación de la aplicación MR

Posterior a la capacitación, se estableció un escenario de prueba dentro de un cuarto de aproximadamente $8m^2$, con entrada parcial de luz y la conexión del dron con su simulador de vuelo, como se ilustra en la Figura 60. No se realizaron pruebas con vuelos reales con los 20 estudiantes, pues no tenían experiencia en el pilotaje de drones. Sin embargo, si se ejecutaron misiones de vuelo en ambiente real con dos pilotos expertos de drones de la empresa DJI y con experiencia en el uso del HMD HoloLens. Uno de los pilotos expertos que realizó la prueba es ingeniero graduado de la carrera de Informática y Sistemas de la universidad ESPE, mientras que la otra prueba fue realizada por una estudiante de PhD de la Universidad de Castilla – La Mancha y actual profesora de la universidad ESPE.

Figura 60: Pruebas con usuarios expertos e inexpertos

El objetivo de la prueba para cada usuario fue evaluar la experiencia del usuario al simular un escenario de misión automática basada en geolocalización. En esta prueba el usuario tenía la tarea de programar y ejecutar una misión de vuelo basada en puntos de referencia, fijándose en la información proporcionada en la interfaz de MR. Posteriormente, el usuario debía responder con total sinceridad al cuestionario de la Tabla 12.

6.2.3. Resultados

En esta sección se presentan las interpretaciones a las respuestas de los usuarios expertos e inexpertos sobre las preguntas del cuestionario de la Tabla 12. Las respuestas (Nada, Mínimo, Poco, Ligeramente y Mucho) de los usuarios inexpertos y sus valores correspondientes (1, 2, 3, 4 y 5) basados en la métrica MOS se encuentran distribuidas de acuerdo con la Figura 61, mientras que, las respuestas de los usuarios expertos se encuentran distribuidas de acuerdo con la Figura 62, excepto la pregunta 16 que se describe en la Sección 6.3.3.5.

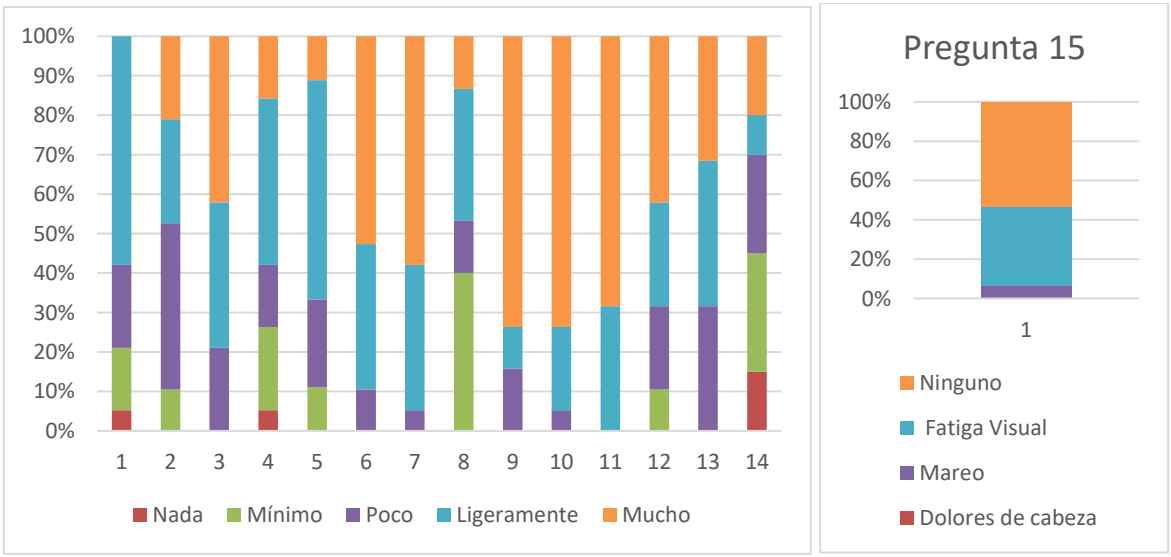


Figura 61: Distribución de las respuestas de los usuarios inexpertos al cuestionario de evaluación

Figura 62: Distribución de las respuestas de los usuarios expertos al cuestionario de evaluación

Todos los evaluadores cumplieron con el objetivo de la misión y respondieron a las preguntas del cuestionario (Anexo 1), evaluando inconscientemente la accesibilidad (Sección 6.3.3.1), inmersión (Sección 6.3.3.2), usabilidad (Sección 6.3.3.3), concentración (Sección 6.3.3.4) y la nitidez de la imagen del video (Sección 6.3.3.5) que provee la aplicación de búsqueda y rescate SAR.

6.2.3.1. Accesibilidad

De los 20 usuarios inexpertos, 6 presentaron problemas físicos de fatiga visual y otro sintió mareos. Mientras que los usuarios expertos no presentaron ningún tipo de problema físico. En la Figura 63 se ilustra la distribución de las calificaciones (Nada, Mínimo, Poco, Ligeramente y Mucho) de accesibilidad de los usuarios inexpertos hacia el HMD de MR. El estudio de accesibilidad comprendió al grado de experticia y la ayuda necesaria para utilizar el dispositivo HMD HoloLens.

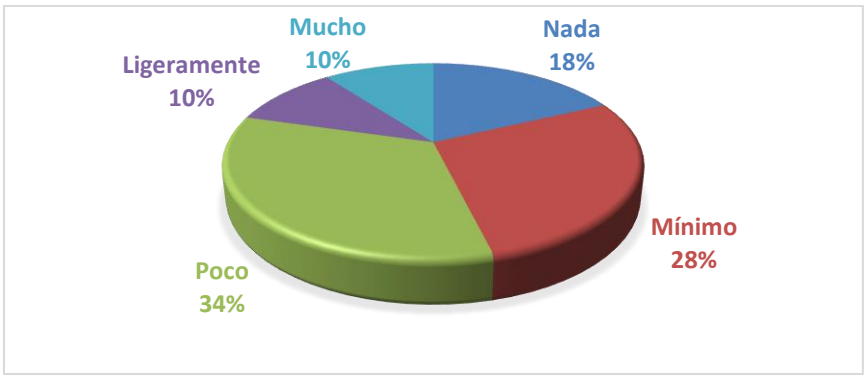


Figura 63: Accesibilidad de los usuarios inexpertos a los HMD de MR

Al realizar un promedio de las calificaciones inspirado en la métrica MOS, se obtuvo un valor de 2.666. Interpretando este resultado numérico se puede inferir que los usuarios inexpertos tienen poca experticia en el control de las aplicaciones de MR, sin embargo,

necesitan poca ayuda para el control de sus interfaces. Por otro lado, los usuarios expertos coinciden en no presentar problemas físicos y no tener dificultar en el control de aplicaciones de MR.

6.2.3.2. Inmersión

Este estudio de inmersión comprende factores como, sentimiento hacia el rol de piloto, conciencia de las acciones ejecutadas por el piloto, grado de interés por una interfaz de MR para el pilotaje de drones y la conciencia entre el mundo virtual y el mundo real. Los resultados de este estudio por parte de los usuarios inexpertos se encuentran distribuidos de acuerdo con la Figura 64.

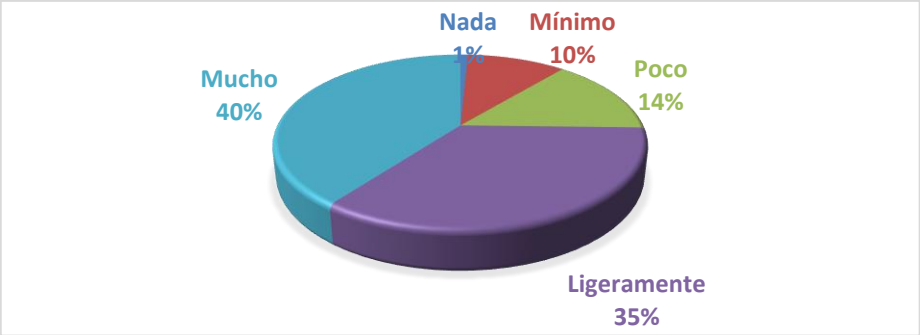


Figura 64: Inmersión de los usuarios inexpertos hacia la aplicación SAR

Al realizar un promedio de las calificaciones de los usuarios inexpertos con la ayuda de la métrica MOS, se obtuvo un valor de 4.039. Este valor infiere que los usuarios se sintieron ligeramente inmersos dentro del entorno virtual que se le ha sido presentado. Por otro lado, las calificaciones de los usuarios expertos variaron entre estar ligeramente y muy inmersos dentro de la realidad que se les presentó, coincidiendo en estar muy apropiados en el rol de piloto, estar muy conscientes de las acciones que tomaron y mostrando mucho más interés por una interfaz MR que una interfaz móvil.

6.2.3.3. Usabilidad

El estudio de usabilidad analizó la facilidad de aprendizaje y ejecución del vuelo automático basado en puntos de georreferencia. La percepción de los usuarios inexpertos se distribuyó de acuerdo con la Figura 65. Mientras que las respuestas de los usuarios expertos se encuentran entre (Ligeramente y Mucho).

Al calcular la opinión media MOS de los usuarios inexpertos se obtuvo un valor de 3.91. Este valor infiere que los usuarios inexpertos percibieron que es ligeramente fácil aprender y ejecutar un vuelo automático. Por otro lado, los usuarios expertos coincidieron en que fue muy fácil ejecutar una misión de vuelo automático, sin embargo, sintieron que se requiere ligeramente un esfuerzo para aprender a hacerlo.

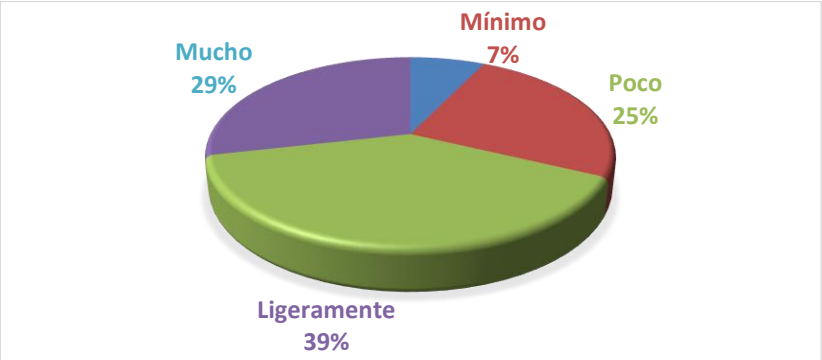


Figura 65: Percepción de usabilidad por los usuarios inexpertos

6.2.3.4. Concentración

La concentración de los usuarios inexpertos y la calidad de la información presentada en la aplicación SAR es distribuida de acuerdo con la Figura 66. El valor de la opinión media MOS de 4.55 indica que los usuarios inexpertos se encontraron muy concentrados o ligeramente concentrados en cumplir con la tarea asignada y que la información presentada fue útil muy útil o ligeramente útil para cumplir con la tarea. Sin embargo, la percepción de los usuarios expertos se fortaleció al realizar un vuelo real, pues indicaron estar muy concentrados en la programación del vuelo automático y que la información provista les fue muy útil.

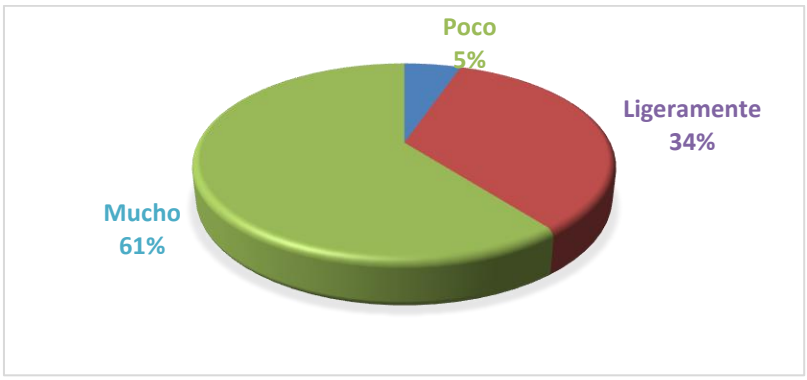


Figura 66: Concentración por parte de los usuarios inexpertos

6.2.3.5. Calidad del video

El factor de calidad del servicio de transmisión de video que fue evaluado por los usuarios expertos e inexpertos fue la nitidez. Las calificaciones disponibles de QoS para la nitidez del video fueron las calificaciones de la métrica MOS que se detallan en la Tabla 11. La opinión de los usuarios expertos e inexpertos se encontraron distribuidas de acuerdo con la Figura 67, donde la opinión media de 4.05 infiere que la nitidez de la imagen presentada en pantalla es buena.

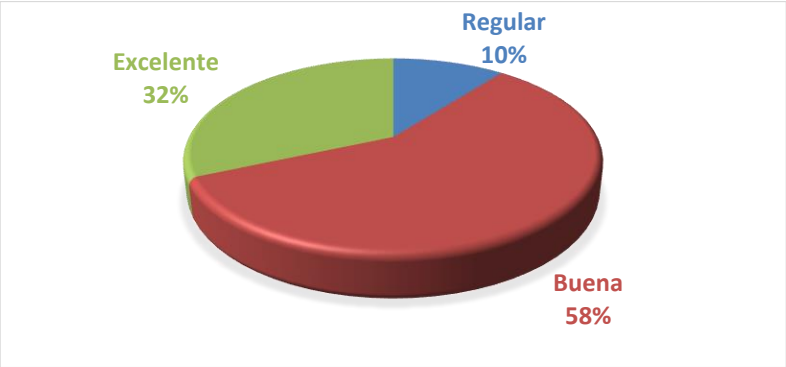


Figura 67: Percepción de los usuarios de la nitidez del video

6.3. Detección de humanos

Los índices de detección de humanos del modelo reentrenado de la Sección 5.5.3.1 fue nuevamente evaluado con imágenes capturadas de una misión de vuelo a una altura máxima de 30 metros de altura. Esta misión de vuelo fue realizada dentro del campus de la Universidad de las Fuerzas Armadas ESPE.

El objetivo de esta misión fue recolectar imágenes de personas y sin personas para calcular los valores de *precision*, *recall* y *F1* mediante una clasificación binaria. La misión fue ejecutada en dos etapas. La primera etapa se ejecutó durante las primeras horas de la mañana y en un día parcialmente nublado, mientras que, la segunda etapa se ejecutó durante las últimas horas de la tarde y con cielo despejado.

Durante la primera etapa de la misión de búsqueda de humanos se contó con el apoyo de 5 estudiantes de la carrera de Tecnologías de la Información, quienes tuvieron la tarea de adoptar distintas posiciones (acostados, de pie, sentados) del cuerpo humano y ocultarse entre la floresta del campus universitario.

Por otro lado, la segunda etapa de la misión de búsqueda comprendió un recorrido de distintos espacios del campus (edificios, canchas, pasillos, parqueaderos, jardines). Al concluir estas dos etapas, se seleccionaron imágenes con (uno, dos, tres, cuatro y cinco) humanos y sin humanos, imágenes que se ilustran en la Figura 68.

Figura 68: Imágenes obtenidas de la misión de búsqueda dentro del campus Matriz ESPE con (uno, dos, tres, cuatro y cinco) humanos y sin humanos

Se recolectaron un total de 150 imágenes con una resolución de 1280x720, pero fueron reducidas a una resolución de 320x180 píxeles. Esta resolución fue definida por el análisis de transmisión de video realizado en la Sección 6.2.1 y es utilizada para inferir sobre la eficiencia del modelo de detección de humanos que fue colocado en el módulo de *Conocimiento* descrito en la Sección 4.5.

De las 150 imágenes obtenidas, 118 tienen humanos y 32 no contienen humanos, que corresponde al 78.6% y 21.4% aproximadamente. De las 118 imágenes que tienen humanos, 30 tienen una persona, 17 tienen dos personas, 20 tienen tres personas, 16 tienen cuatro personas y 32 tienen cinco personas. En el total de 150 imágenes se cuentan con un total de 348 instancias de personas, tal y como se detalla en la Tabla 14.

Tabla 14: Agrupación de imágenes para el proceso de inferencia

Imágenes	***Cantidad de imágenes***	***Instancias de personas***
Sin personas	*32*	*0*
Una persona	*30*	*30*
Dos personas	*17*	*34*
Tres personas	*20*	*60*
Cuatro personas	*16*	*64*
Cinco personas	*32*	*160*
TOTAL	*150*	*348*

6.3.1. Resultados de la inferencia

El cálculo de los índices de detección mediante la clasificación binaria fue realizado mediante la inferencia de la detección sobre las 150 imágenes que se detallan en la Tabla 14 y se ilustra en la Figura 69. La primera etapa del proceso de la clasificación binaria consistió en un sumar el número de Verdaderos-Positivos VP (literales c y d), Falsos-Positivos FP (literal a) y Falsos-Negativos FN (persona no detectada en el literal b).

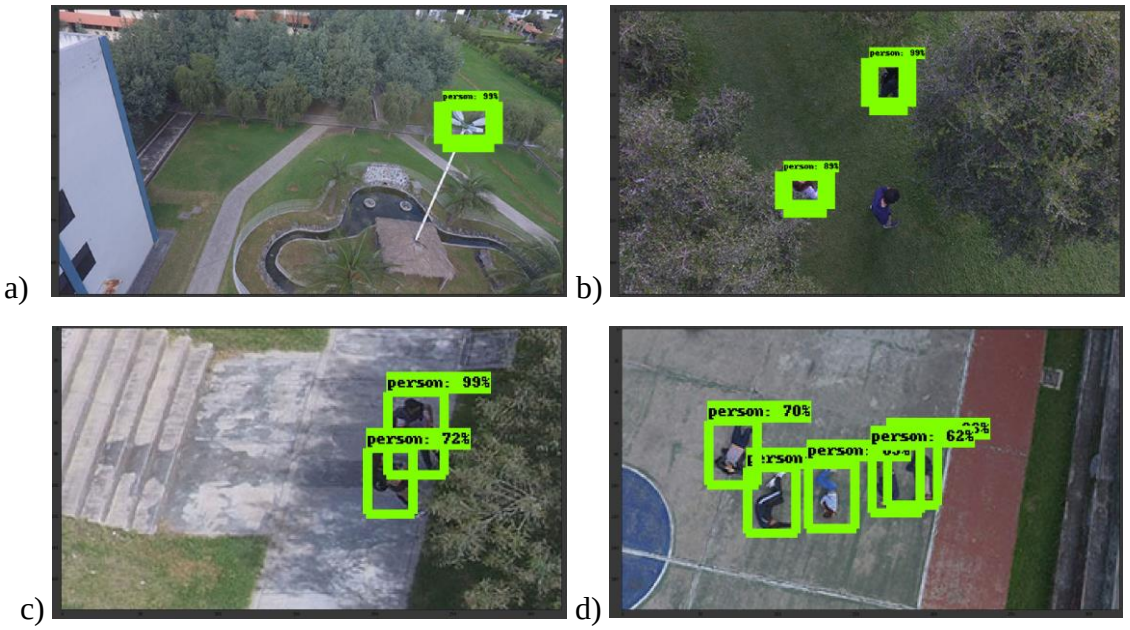


Figura 69:Inferencia de la detección de personas con imágenes con humanos y sin humanos

Tras obtener en la Tabla 15 los valores de VP, FP y FN de las 150 imágenes, la segunda etapa de la clasificación binaria consistió en calcular los valores de *precision*, *recall* y *F1* mediante las fórmulas descritas en la Tabla 5 de la Sección 3.6.2.

Tabla 15: Agrupación de imágenes para el proceso de inferencia

Imágenes	***Instancias de Personas***	***VP***	***FP***	***FN***	***Precision***	***Recall***	***F1***
150	*348*	*267*	*30*	*81*	*0.89899*	*0.767241*	*0.827907*

Entre los valores de 0.89 para *precision* (*mAP)* y 0.5572 para *recall* obtenidos en el entrenamiento del modelo detallado en la Sección 5.5.1.3, el valor de *precision* no presentó una variación significativa. Por otro lado, el valor de *recall* presentó una variación de 0.21 aproximadamente, que posiblemente se deba a las características de las imágenes utilizadas para la inferencia.

6.4. Síntesis

En la segunda sección de este capítulo se presentó un análisis de las configuraciones del servicio de transmisión de video. Las configuraciones ideales para la resolución de la imagen y los cuadros por segundo *fps* fueron obtenidas gracias a evaluaciones de transmisiones de video con un tiempo de transmisión de 20 minutos, tiempo en que la batería del dron Phantom 4 se agota cuando se encuentra volando.

Posteriormente, la resolución de la imagen (320x180 píxeles) y los cuadros por segundo (2 *fps)* del servicio de transmisión de video fueron analizadas en términos de QoS con otra transmisión de video. Esta evaluación de calidad de video principalmente estudió en el retraso, la pérdida de paquetes y los paquetes fuera de orden, presentado resultados que se encuentran dentro de las recomendaciones de la Unión Internacional de Telecomunicaciones.

La tercera sección presentó las evaluaciones de experiencia realizadas con usuarios expertos y con usuarios inexpertos en el manejo de dispositivos MR y pilotaje de drones de la marca DJI. Las evaluaciones del servicio de transmisión y percepción de la aplicación SAR se inspiraron en la métrica de evaluación subjetiva MOS de QoE. Las evaluaciones con los usuarios presentaron resultados favorables en cuanto su grado de accesibilidad a los dispositivos HMD de MR, su grado de inmersión con la interfaz MR, el grado de usabilidad de la aplicación, el grado de distracción a lo largo de la evaluación y la nitidez del video.

Finalmente, la cuarta sección de este capítulo presentó un valor de 0.89899 para *precision* y 0.767241 para *recall* de una clasificación binaria con imágenes capturadas de una misión de vuelo de prueba. Las imágenes capturadas para esta clasificación fueron capturadas dentro del campus de la Universidad de las Fuerzas Armadas ESPE.

7. Conclusiones y trabajos futuros

El uso de vehículos aéreos no tripulados es una importante herramienta en aplicaciones de búsqueda y rescate, porque con estos vehículos se pueden realizar rápidas exploraciones de las áreas afectadas por desastres naturales. Por otro lado, mediante un caso especial de la aplicación de la visión por computador llamada detección de humanos, los sistemas de vehículos no tripulados tienen la capacidad de analizar las imágenes transmitidas desde la cámara instalada en el dron hacia las estaciones de tierra.

A pesar de existir trabajos anteriores sobre sistemas de búsqueda y rescate con drones que incluyeron la detección de humanos, estos trabajos manifiestan la existencia de desafíos en cuando a la complejidad de la interacción de los pilotos con las interfaces presentadas en las estaciones de tierra y el análisis de imágenes en tiempo real. Estos desafíos fueron implementados y evaluados en este trabajo, donde, las conclusiones en base a los resultados obtenidos se describen en la siguiente Sección 7.1. Sin embargo, en el desarrollo de este trabajo se ha encontrado que las aplicaciones de búsqueda y rescate presentan otros desafíos que no se alcanzaron a trabajar, por lo que son presentados como trabajos futuros en la Sección 7.2.

7.1. Conclusiones

Durante este trabajo se han encontrado diversos trabajos que incluyeron interfaces de realidad aumentada y de realidad virtual en las estaciones de tierra sobre dispositivos montados en la cabeza con la finalidad de brindar mejores experiencias a los pilotos. Por otro lado, también se descubrió que estas interfaces de realidad virtual se incluyeron en la comercialización de drones de la empresa Parrot. Aun así, aunque este tipo de interfaces permiten estar consciente de la posición del dron al visualizar en tiempo real las imágenes que captura su cámara, no le permitían al piloto estar consciente de su ubicación física cuando este dron se encuentra dentro y fuera de la línea de vista.

Por otro lado, aunque se encontró aplicaciones que usaron la realidad virtual y aumentada para el entrenamiento de pilotos, este trabajo desarrolló una interfaz en la cual todos los usuarios inexpertos con que se evaluó el prototipo consiguieron ejecutar una misión de vuelo y estar conscientes de la posición del dron, cuando se encontraba dentro y fuera de la línea de vista. Estos resultados fueron posibles gracias a la proyección de hologramas sobre objetos reales mediante el uso de la realidad mixta.

En la revisión de literatura, no se encontró una arquitectura de comunicación entre un dron y un dispositivo montado en la cabeza que incluyera la detección humana. Por este motivo, este trabajo propuso e implemento una arquitectura de comunicación que puede ser utilizada para la integración de un dron de origen comercial, un teléfono móvil y un dispositivo de

realidad mixta también de origen comercial, donde, el teléfono móvil incluye módulos para a la detección de humanos.

Durante el perfeccionamiento de los módulos correspondientes a la detección de humanos, se experimentó con la herramienta OpenCV para las métodos tradicionales de *Machine Learning* y con la herramienta TensorFlow para métodos de *Deep Learning*. Donde los modelos de detección de *Deep Learning* mostraron a simple vista mejores resultados en la detección de humanos con diferentes posiciones del cuerpo. Por otro lado, suponen ser mucho más fáciles de entrenar, debido a las capacidades de aprendizaje de las redes neuronales profundas.

De los algoritmos de detección de múltiples objetos de *Deep Learning*, SSD (del inglés, Single Shot Multibox Detector) presenta rapidez y la mayor facilidad de entrenamiento, al realizar todos cálculos en usa sola red, al no analizar regiones innecesarias y al entrenar el modelo basándose en un umbral de coincidencia definido en la red.

El uso adecuado de la plataforma en la nube Google Colaboratory presenta los mismos resultados en comparación a los servicios Google Cloud Platform, pero sin costo por el entrenamiento de un modelo de detección con TensorFlow. Pues en Google Colaboratory solo se requiere habilitar los recursos de GPU, tener paciencia para visualizar los resultados del entrenamiento y respaldar la información del entrenamiento en caso de superar el tiempo límite de la sesión. El modelo, resultado del entrenamiento, es capaz de detectar diferentes posiciones del cuerpo humano captados desde el aire, así como también diferenciar estos cuerpos de los escenarios de rescate.

Las evaluaciones subjetivas de calidad de experiencia de 22 usuarios expertos e inexpertos supieron manifestar que la nitidez de la imagen del video de nuestra aplicación es buena. Este video tiene una resolución de 320x180 píxeles y una reproducción a 2 cuadros por segundo, configuraciones que permiten una fluida transmisión de video durante 20 minutos, tiempo en que la batería del dron utilizado en este trabajo se agota cuando se encuentra en vuelo.

Finalmente, las pruebas subjetivas realizadas por usuarios expertos e inexpertos, pruebas que en este caso fueron inspiradas en la métrica de evaluación MOS de QoE para obtener una percepción media y evaluadas mediante la interacción de los usuarios con nuestra aplicación, presentan resultados favorables en cuanto al grado de accesibilidad a los dispositivos montados en la cabeza de realidad mixta, al grado de inmersión con la interfaz, al grado de usabilidad y al grado de distracción.

7.2. Trabajos Futuros

Mediante la experiencia obtenida en este trabajo, el análisis de resultados, las conclusiones y teniendo en cuenta la tendencia de crecimiento en cuanto a las investigaciones sobre los vehículos aéreos no tripulados, los dispositivos de realidad mixta y la visión por computador, se proponen las siguientes investigaciones que pueden ser realizadas a corto o largo plazo.

Dentro de las investigaciones a **corto plazo** se podría:

- Añadir el seguimiento de cuerpos a la detección de humanos, explotando funcionalidad del algoritmo SSD usado en este trabajo.
- Incluir mapas 3D de Bing Maps en la interfaz de realidad mixta para obtener un posicionamiento en *x*, *y* y *z* del dron, obteniendo una experiencia más real de la ubicación del vehículo.
- Controlar la interfaz de realidad mixta mediante comandos de voz, así como también recibir alertas de audio cuando se detecten personas.
- Experimentar con el uso de dispositivos móviles con mejores capacidades de procesamiento para conseguir un análisis y transmisión de video con mejor calidad de imagen.

En cuanto a las investigaciones a **largo plazo** se podría:

- Incluir la detección de personas en condiciones de luz desfavorables con la instalación de una cámara térmica sobre el dron, ya que lamentablemente en este proyecto no se pudo adquirir una cámara de la empresa FLIR.
- Crear un algoritmo de superposición de imágenes térmicas y ópticas que sea utilizado para validar si una detección sobre una imagen óptica corresponde a una persona, o, por el contrario, si esa persona se encuentra con vida.
- Crear un *dataset* con imágenes de personas y animales en desastres naturales de procedencia geográfica, climática, geológica o tectónica.

Referencias

Ali Haydar, S. S. (2005). An Augmented Reality System for Multi-UAV Missions. *SimTect'05 Asia Pacific Simulation Technology and Training*. SemanticScholar.

Amin, H., & Rebeen, R. (2016). Video QoS/QoE over IEEE802.11n/ac: A Contemporary Survey. Obtenido de Thesis of Master Rochester Institute of Technology: https://scholarworks.rit.edu/theses/9148.

Andriluka, M., Pishchulin, L., Gehler, P., & Schiele, B. (2014). 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. *IEEE Conference on Computer Vision and Pattern Recognition*. 1-8.

Blondel, P., Potelle, A., Pégard, C., & Lozano, R. (2014). Fast and viewpoint robust human detection for SAR operations. *IEEE International Symposium on Safety, Security, and Rescue Robotics*. 2-7.

Blondel, P., Potelle, A., Pégard, C., & Lozano, R. (2014). Human Detection in Uncluttered Environments: from Ground to UAV View. *IEEE International Conference on Control Automation Robotics & Vision*. 2-7.

Brieman, L. (1996). Bagging predictors. *Machine Learning*. Springer. 24(2):123-140.

Brummelen, V., & Glen, R. (2013). *Heavenly Mathematics: The Forgotten Art of Spherical Trigonometry*. New Jersey, NJ: Princeton University.

Cavalcante, R. (2017). Virtual Reality Immersive user interface for first person view games. Obtenido de Master Thesis of the Univerity of Beira: DOI:10.13140/RG.2.2.21989.91368.

Chou, W., Chen, J., Huang, T., King, H., Chen, C., & Wang, S. (1999). Head mounted display (HMD) commercialized product practice research. *Proceedings of 5th Asian Symposium on Information Display. ASID '99 (IEEE Cat. No.99EX291)*. 1-4.

Collins Dictionary. (18 de 09 de 2018). *Unmanned Aerial Vehicle*. Obtenido de Collins Dictionary: https://www.collinsdictionary.com/dictionary/english/uav.

de Oliveira, D., & Wehrmeister, M. (2018). Using Deep Learning and Low-Cost RGB and Using Deep Learning and Low-Cost RGB and Images Captured by Multirotor UAV. Sensors, 18 (7), 22-44.

de Oliveria, D. (2016). Uma abordagem para detecção de pessoas em imagens de veículos aéreos não tripulados. Obtenido de Dissertação de Mestrado: http://repositorio.utfpr.edu.br/jspui/handle/1/2036.

Defense, D. o. (2006). DoD Support to Civil Search and Rescue (SAR) . United States of America. Recuperado el 3 de marzo del 2019 de: https://www.hsdl.org/?view&did=459986.

DJI. (2016). *User manual Phantom 4*. Recuperado el 4 el 10 de octubre del 2018 del manual de usuario del Phantom de : https://dl.djicdn.com/downloads/phantom_4/en/Phantom_4_User_Manual_en_v1.0.pdf.

Doherty, P. (2004). Advanced Research with Autonomous Unmanned Aerial Vehicles. *KR'04 Proceedings of the Ninth International Conference on Principles of Knowledge Representation and Reasoning*. 731-732.

Doherty, P., & Rudol, P. (2007). A UAV Search and Rescue Scenario with Human Body Detection and Geolocalization. *AI 2007: Advances in Artificial Intelligence*. Springer, 1-13.

Doherty, P., Granlund, G., Kuchcinski, K., Sandewall, E., Nordberg, K., Skarman, E., & Wiklund, J. (2000). The WITAS Unmanned Aerial Vehicle Project. ECAI, 1-9.

Dong Ming, Z. X. (2009). Identification of Humans Using Infrared Gait Recognition. *IEEE VECIMS 2009 - International Conference on Virtual Environments*.

DRONEII. (2017). *Las 20 compañías que dominan el mercado de los drones*. Recuperado el 23 de julio del 2018 de DRONEII: http://www.ticbeat.com/tecnologias/las-20-companias-que-dominan-el-mercado-de-los-drones/.

ECU911. (2016). *Drones Ecu911*. Recuperado el 11 de diciembre del 2018 del Servicio Integrado de Seguridad ECU911: http://www.ecu911.gob.ec/drones-ecu-911/

Egred, J. (2018). *El Terremoto de Riobamba 1797.* Quito: Ecuador. Recuperado el 26 de julio del 2018 del Instituto Geográfico de la Escuela Politécnica del Ejército: https://www.igepn.edu.ec/publicaciones-para-la-comunidad/comunidad-espanol/23-el-terremoto-de-riobamba-de-1797/file.

El Comercio. (2016). *A 9 días de ocurrido el terremoto, 655 personas fallecieron y 48 están desaparecidas*. Recuperado el 13 de julio del 2018 de El Comercio el: http://www.elcomercio.com/actualidad/terremoto-manabi-fallecidos-desaparecidos-damnificados.html.

FAO. (2014). *En Tierra Segura*. Obtenido de: http://www.fao.org/docrep/013/i1255b/i1255b02.pdf

Ford, J. (2018). *History of Drones*. Recuperado el 25 de octubre del 2018 de Dronethusiastic: https://www.dronethusiast.com/history-of-drones/.

FPO. (2018). *Radio Remote Control System*. Recuperado el 25 de octubre del 2018 de Free Patent Online: http://www.freepatentsonline.com/2408819.html.

Freund, Y. (1995). Boosting a weak learning algorithm by major. *Journal of Information and Computation.* AC, 121 (2), 256-285.

Freund, Y., & Schapire, R. (1996). A decision theoretic generalization of online learning and a application of boosting. *Journal of computer and system sciences.* ACM. 55(1): 119-137.

Freund, Y., & Schapire, R. E. (1999). A Short Introduction to Boosting. *Artificial Intelligence*. ACM. 14 (5), 771-780.

Gandy, M., Catrambone, R., MacIntyre, B., Alvarez, C., Eiriksdottir, E., Hilimi, M., . . . Collins, A. (2010). Experiences with an AR Evaluation Test Bed: Presence, Performance, and Physiological Measurement. *IEEE International Symposium on Mixed and Augmented Reality*.

Gartner. (2017). *Top Trends in the Gartner Hype Cycle for Emerging Technologies*. Recuperado el 15 de agosto del 2018 de Gartner: https://www.gartner.com/smarterwithgartner/top-trends-in-the-gartner-hype-cycle-for-emerging-technologies-2017.

Gerven, M. V., & Bohte, S. (2017). *Artificial Neural Networks as Models of Neural Information Processing.* Comput Neurosci. Recuperado el 11 de noviembre del 2018 de Research Topics: https://www.frontiersin.org/research-topics/4817/artificial-neural-networks-as-models-of-neural-information-processing.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Conference on Computer Vision and Pattern Recognition*. ACM, 580-587.

Goldman, E., Herzig, R., Eisenschtat, A., Ratzon, O., Goldberger, I. L., & Hassner, T. (2019). *Precise Detection in Densely Packed Scenes.* CoRR, abs/1904.00853.

Gollapudi, S. (2016). Practical Machine Learning. Obtenido de: https://www.amazon.es/Practical-Machine-Learning-Sunila-Gollapudi/dp/178439968X.

Google Developers. (4 de 6 de 2019). *Maps Unity SDK*. Obtenido de Google Developers: https://developers.google.com/maps/documentation/gaming/overview_musk.

Grigsby, S. (2018). Artificial Intelligence for Advanced Human-Machine Symbiosis. *Augmented Cognition: Intelligent Technologies*. Springer, 255-266.

Han, J., Zhang, D., Cheng, G., Liu, N., & Xu, D. (2018). Advanced Deep-Learning Techniques for Salient and Category-Specific Object Detection: A Survey. *IEEE Signal Processing Magazine*. 35(1)

Haridas, N., & Sandhiya, S. (2018). *Intel AI Academy*. Recuperado el 26 de enero del 2019 de Intel: https://software.intel.com/en-us/articles/traffic-light-detection-using-the-tensorflow-object-detection-api.

IDC. (2018). *International Data Corporation*. Recuperado el 18 de octubre del 2018 de IDC: https://www.idc.com/getdoc.jsp?containerId=US45091919.

Jaysen A. Yochim, M. U. (1998). *The Vulnerabilities of Unmanned Aircraft Systema Common Data*. Obtenido de Master of Military Art and Science of the U.S. Army Command and General Staff College: https://fas.org/irp/program/collect/uas-vuln.pdf.

Kashyap, V. (2018). *A Brief History of Drones: The Remote Controlled Unmanned Aerial Vehicles (UAVs)*. Recuperado el 25 de agosto del 2018 de Interesting Engineering: https://interestingengineering.com/a-brief-history-of-drones-the-remote-controlled-unmanned-aerial-vehicles-uavs.

Khalaf, A., Alharthi, S., Torres, R., Dolgov, I., Pianpak, P., NaminiMianji, Z., . . . Toups, Z. (2018). An Architecture for Simulating Drones in Mixed Reality Games to Explore Future Search and Rescue Scenarios. SemanticScholar.

Kiremire, A. R. (2011). *The application of the Pareto principle in Software Engineering.* Obtenido de: http://www2.latech.edu/~box/ase/papers2011/Ankunda_termpaper.PDF.

Krerngkamjornkit, R., & Simic, M. (2013). Human Body Detection in Search and Rescue Operation Conducted by Unmanned Aerial Vehicles. *Conference on Advances in Materials and Manufacturing*. 1077-1085. Obtenido de DOI: 10.4028/www.scientific.net/AMR.655-657.1077

Lawrence, J., Malmsten, J., Rybka, A., Sabol, D., & Triplin, K. (2017). Comparing TensorFlow Deep Learning Performance Using CPUs, GPUs, Local PCs and Cloud. CUNY Academic Works.

Le, H., Le, T., Tran, S., Tran, H., & Thanh, T. (2012). Image Classification using Support Vector Machine and Artificial Neural Network. *International Journal of Information Technology and Computer Science.* Obtenido de DOI:10.5815/ijitcs.2012.05.05

LeCun, Y. (1989). Generalization and network design strategies. *Connectionims in Perspective.* Zurich, Switzerland: Elsevier.

Lee, J., Choi, J.-S., Jeon, E., Kim, Y., Le, T., Shin, K., . . . Kang, P. (2015). Robust Pedestrian Detection by Combining Visible and Thermal Infrared Cameras. Sensors, 15 (5), 10580-615.

Lewis, C., & Pickavance, S. (2019). *Implementing Quality of Service Over Cisco MPLS VPNs*. Recuperado el 11 de abril del 2019 de Csco: www.ciscopress.com/articles/article.asp?p=471096&seqNum=6.

Li, F.-F., Johnson, J., & Yeung, S. (2017). *Lecture 8: Deep Learning Software*. Recuperado el 27 de abril del 2019 de Deep Learning: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture8.pdf.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., . . . Dollár, P. (2014). Microsoft COCO: Common Objects in Context. *Computer Vision – ECCV* . Springer, 740-755.

Liu, W., Anguelov, D., Erhan, D., & Szegedy, C. (2016). SSD: Single Shot MultiBox Detector. *Computer Vision – ECCV*. Springer, 21-37.

llnnrell, Q. J. (2004). Recognition of human and animal movement using infrared video stream. *IEEE International Conference on Image Processing*.

Long, G. (2016). *9 fallecidos extranjeros confirmados; 654 rescatistas* Recognition of human and animal movement using infrared video stream *foráneos en el país*. Recuperado el 25 de julio del 2018 de El Telégrafo: https://www.eltelegrafo.com.ec/noticias/ecuador/1/9-fallecidos-extranjeros-confirmados-654-rescatistas-foraneos-en-el-pais.

Lu, D., & Weng, Q. (2007). A Survey of Image Classification Methods and Techniques for Improving Classification Performance. *IEEE International Journal of Remote Sensing*. 28 (5), 823-870.

Madeleen, H., & Dorothea, H. (2006). Natural Disasters and Climate Change. 30 (1), 1-4. Obtenido de DOI: 0.1111/j.1467-9523.2006.00302.x.

Martins, F., Groot, M., Stokkel, X., & Wiering, M. (2016). Human Detection and Classification of Landing Sites for Search and Rescue Drones. *European Symposium on Artificial Neural Networks*.

Massera, C. (2014). Pedestrian Detection using K-means and Random Decision Forest. *IEEE Latin American Robotics Symposium.* 103-108.

Meus, B., Kryjak, T., & Gorgon, M. (2017). Embedded vision system for pedestrian detection based on HOG+SVM and use of motion information implemented in Zynq heterogeneous device. *IEEE Signal Processing: Algorithms, Architectures, Arrangements, and Applications*.

Microsoft. (2016). *Announcing Microsoft HoloLens Development Edition open for pre-order, shipping March 30*. Recuperado el 20 de febrero del 2018 de Microsoft Windows https://blogs.windows.com/devices/2016/02/29/announcing-microsoft-hololens-development-edition-open-for-pre-order-shipping-march-30/.

Microsoft. (2018). *HoloLens hardware details*. Recuperado el 21 de marzo del 2019 de Microsoft: https://docs.microsoft.com/en-us/windows/mixed-reality/hololens-hardware-details.

Microsoft. (2018). *Install the tools*. Recuperado el 26 de marzo del 2019 de Microsoft: https://docs.microsoft.com/en-us/windows/mixed-reality/install-the-tools.

Milan Erdelj, E. N. (2017). Help from the Sky: Leveraging UAVs for Disaster Management. *IEEE Pervasive Computing.* 24-32.

Min Zuo, G. Z. (2010). Research and Improvement of Face Detection Algorithm Based on the OpenCV. *IEEE International Conference on Information Science and Engineering*.

Minsky, M. L., & Papert, S. (1969). Perceptrons: An Introduction to Computational Geometry. IEEE, 18 (6), 572-572.

Möller, S., & Raake, A. (2014). Quality of Experience. *T-Labs Series in Telecommunication Services*. Springer, 11-33.

Moreno, A. P., Santiago, O. L., Lazaro, J. M., & Moreno, E. G. (2013). Estudio comparativo de clasificadores empleados en el diagnóstico de fallos de sistemas industriales . *IEEE Latin America Transactions*. 87-98.

Murazzo, M., Rodríguez, N., Vergara, R., Carrizo, F., González, F., & Grosso, E. (2013). Administración de QoS en ambientes de redes de servicios convergentes. *XV Workshop de Investigadores en Ciencias de la Computación*. Red de Universidades con Carreras en Informática, 53-57.

Nagendran, A., Harper, D., & Shah, M. (2010). New system performs persistent wide-area aerial surveillance. *Conference on Computer Vision and Pattern Recognition*. Obtenido de SPIE: http://arjunnagendrancom.ipage.com/uploads/3/3/9/9/3399323/aerial_surveillance.pdf.

Navneet Dalal, B. T. (2005). Histograms of Oriented Gradients for Human Detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Novapex. (2018). *Google maps vs mapbox*. Recuperado el 11 de noviembre del 2018 de NovapexTech: http://www.novapextech.com/v2/2018/12/28/google-maps-vs-mapbox.

Oñate, E. F. (2017). EL mercado de los UAV un dejà vu (por los ingenieros de telecomunicación). *COIT*, 49-51.

OpenCV. (2018). *OpenCV* . Recuperado el 26 de octubre del 2018 de OpenCV: https://opencv.org/about.html.

Padilla, R. (2019). *Object-Detection-Metrics*. Recuperado el 21 de enero del 2019 de Github: https://github.com/rafaelpadilla/Object-Detection-Metrics.

Parrot. (2018). *Parrot Blog*. Recuperado el 23 de octubre del 2018 de Parrot: http://blog.parrot.com/2015/11/19/introducing-parrot-bebop-2-flying-companion.

Pierre, P., & Gustavo, G. (1998). El Ecuador al cruce de varias influencias climáticas. Una situación estratégica para el estudio del Fenómeno del Niño. *Bulletin de l'Institut français d'études andines*. Redalyc,1-10.

Piotr Rudol, P. D. (2008). Human Body Detection and Geolocalization for UAV Search and Rescue Missions Using Color and Thermal Imagery. *IEEE Aerospace Conference*.

Pólkaa, M., Ptaka, S., & Kuzioraa, L. (2017). The use of UAV's for search and rescue operations. Science Direct, 748-752.

Preece, J., Rogers, Y., & Sharp, H. (2005). Design de interação: além da interação do homemcomputador. Bookman.

Prexl, M., Struebig, K., Harder, J., & Hoehn, A. (2017). User Studies of a Head-Mounted Display for Search and Rescue Teleoperation of UAVs via Satellite Link. *IEEE Aerospace Conference*.

Rainer, S., & Kristie, E. (2012). Climate change and natural disasters – integrating science and practice to protect health. *Current Environmental Health Reports*, 5 (1), 170-178.

Real, R., & Vargas, J. (1996). The Probabilistic Basis of Jaccard's Index of Similarity. Systematic Biology, 45 (3), 380-385.

Redmon, Ali. (2018). Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *CoRR, abs/1804.02767*.

Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, Faster, Stronger. *IEEE Conference on Computer Vision and Pattern Recognition*. 6517-6525.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *IEEE Conference on Computer Vision and Pattern Recognition*. 779-788.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39, 1137-1149.

Rosenblatt, F. (2016). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65, 386-408.

Ross Girshick, J. D. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *IEEE Conference of Computer Vision and Pattern Recognition*. 580-587.

Ross Girshick, M. R. (2015). Fast R-CNN. *IEEE International Conference on Computer Vision*. 1440-1448.

Samuel, A. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*. IEEE, 3 (3), 210-229.

Schmidhuber, J. (2015). Deep Learning in Neural Networks: An Overview. *Neural and Evolutionary Computing*. Neural Networks, 85-117.

Schwenk, H., & Bengio, Y. (2000). Boosting Neural Networks. *Neural computation*, 12 (8), 1869-1887.

Shmelkov, K., Schmid, C., & Alahari, K. (2017). Incremental Learning of Object Detectors without Catastrophic Forgetting. *IEEE International Conference of Computer Vision*.

Silva, C., & Ribeiro, B. (2018). Aprendizagem computacional em engenharia. Obtenido de DOI:10.14195/978-989-26-1508-0.

StanfordVisionLab. (2018). *A Large-Scale Hierarchical Image Database*. Recuperado el 26 de diciembre del 2018 de: http://image-net.org/challenges/LSVRC.

Sullivan, J. (2006). Evolution or revolution? the rise of UAVs. *IEEE International Symposium on Technology and Society*.

Surantha, N., Isa, S., Lesmana, T., & Setiawan, I. M. (2017). Sleep stage classification using the combination of SVM and PSO. *IEEE 1st International Conference on Informatics and Computational Sciences*.

Tavakoli, S. (2015). Subjective QoE Analysis of HTTP Adaptive Streaming Applications. Recuperado el 11 de noviembre del 2018 de Tesis Doctoral de Universidad Politécnica de Madrid: https://core.ac.uk/download/pdf/33178178.pdf.

Thattapon Surasak, I. T.-h. (2018). Histogram of Oriented Gradients for Human Detection in Video. *IEEE International Conference on Business and Industrial Research*.

Uijlings, J. R., Sande, K. E., Gevers, T., & Smeulders, A. W. (2013). Selective Search for Object Recognition. *International Journal of Computer Vision*, 154-171.

UIT-T_G.1080. (2008). *Quality of experience requirements for IPTV services*. Obtenido de Unión Internacional de Telecomunicaciones: https://www.itu.int/rec/T-REC-G.1080-200812-I.

UIT-T_G.114. (2013). *Sistemas y medios de transmisión, sistemas y redes digitales.* Recuperado el 4 de abril del 2019 de Unión Internacional de Telecomunicaciones: https://www.itu.int/rec/T-REC-G/es.

Vanderlaken, P. (2018). *Neural Networks 101*. Recuperado el 2 de marzo del 2019 de: https://paulvanderlaken.com/2017/10/16/neural-networks-101.

Vélez, J. (2019). Visión por computador / J.F. Vélez Serrano... [et al.]. Madrid: S.L. - DYKINSON.

Viola, P., & Jones, M. (2001). Rapid Object Detection using a Boosted Cascade of Simple Features. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 511-518.

Wei Zhang, K. I. (1990). Parallel distributed processing model with local space-invariant interconnections and its optical architecture. Applied Optics, 1-32.

Wei, Y., Wei, Y., & Chareonsak, C. (2005). FPGA implementation of AdaBoost algorithm for detection of face biometrics. *IEEE International Workshop on Biomedical Circuits and Systems*.

West, J., Ventura, D., & Warnick, S. (2007). DataSpring Research Presentation: A Theoretical Foundation for Inductive Transfer. *College of Physical and Mathematical Sciences.* Brigham Young University.

Xianghua Fan, F. Z. (2012). The System of Face Detection Based on OpenCV. *IEEE Chinese Control and Decision Conference.*

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., & Torralba, A. (2010). SUN database: Large-scale scene recognition from abbey to zoo. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 3484-3492.

Xiaoyue Ji, X. X. (2017). Data-driven augmented reality display and operations for UAV ground stations. *IEEE Data Driven Control and Learning Systems*.

Zdziarski, Z. (2019). *Why Deep Learning Has Not Superseded Traditional Computer Vision*. Recuperado el 6 de enero del 2019 de Zbigatron: https://zbigatron.com/has-deep-learning-superseded-traditional-computer-vision-techniques.

Zhang, S., Bauckhage, C., & Cremers, A. B. (2014). Informed Haar-like Features Improve Pedestrian. *IEEE Conference on Computer Vision and Pattern Recognition*.

Anexos

El cuestionario de evaluación de calidad de experiencia del servicio de transferencia de video y percepción que fue llenado por los usuarios al interactuar con el prototipo se encuentra en el Anexo 1, mientras que las evaluaciones de los dos usuarios expertos se encuentran en el Anexo 2. Finalmente, el Anexo 3 contiene las publicaciones realizadas durante este trabajo.

Anexo 1: Cuestionario

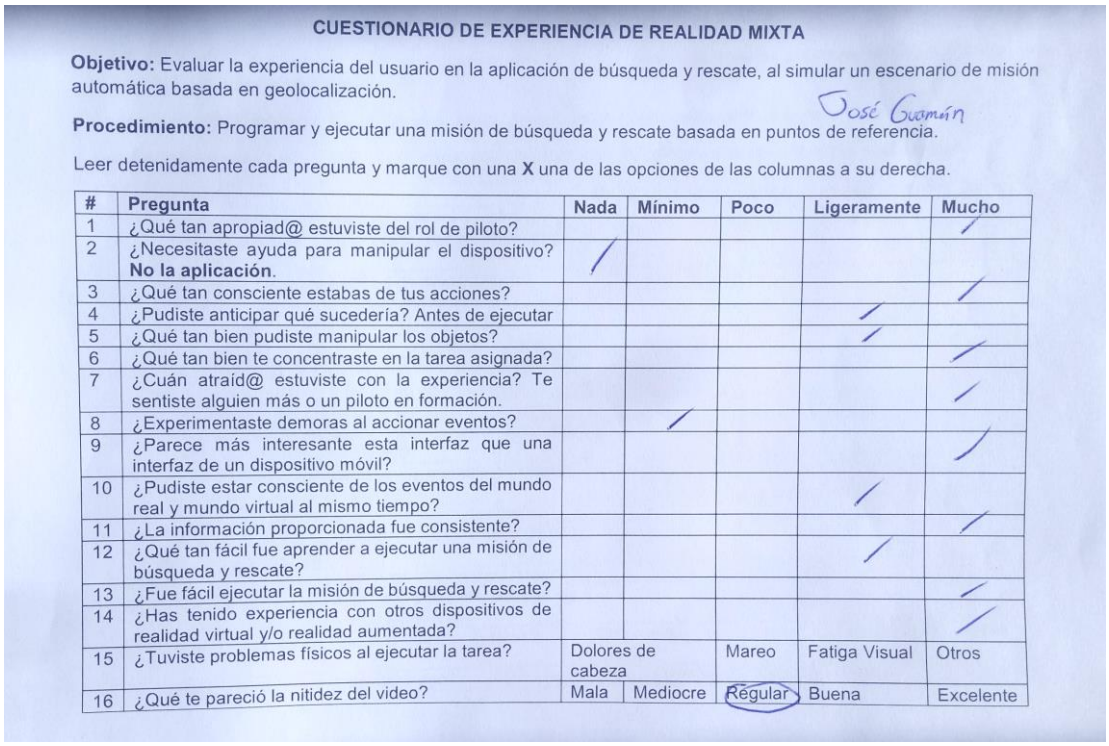
Objetivo: Evaluar la experiencia del usuario en la aplicación de búsqueda y rescate, al simular un escenario de misión automática basada en geolocalización.

Procedimiento: Programar y ejecutar una misión de búsqueda y rescate basada en puntos de referencia.

Leer detenidamente cada pregunta y marque con una **X** una de las opciones de las columnas a su derecha.

#	Pregunta	Nada	Mínimo	Poco	Ligeramente	Mucho
1	¿Qué tan apropiad@ estuviste del rol de piloto?					
2	¿Necesitaste ayuda para manipular el dispositivo? **No la aplicación**.					
3	¿Qué tan consciente estabas de tus acciones?					
4	¿Pudiste anticipar qué sucedería? Antes de ejecutar					
5	¿Qué tan bien pudiste manipular los objetos?					
6	¿Qué tan bien te concentraste en la tarea asignada?					
7	¿Cuán atraíd@ estuviste con la experiencia? Te sentiste alguien más o un piloto en formación.					
8	¿Experimentaste demoras al accionar eventos?					
9	¿Parece más interesante esta interfaz que una interfaz de un dispositivo móvil?					
10	¿Pudiste estar consciente de los eventos del mundo real y mundo virtual al mismo tiempo?					
11	¿La información proporcionada fue consistente?					
12	¿Qué tan fácil fue aprender a ejecutar una misión de búsqueda y rescate?					
13	¿Fue fácil ejecutar la misión de búsqueda y rescate?					
14	¿Has tenido experiencia con otros dispositivos de realidad virtual y/o realidad aumentada?					
15	¿Tuviste problemas físicos al ejecutar la tarea?	Dolores de cabeza		Mareo	Fatiga Visual	Otros
16	¿Qué te pareció la nitidez del video?	Mala	Mediocre	Regular	Buena	Excelente

Anexo 2: Evaluaciones de expertos



CUESTIONARIO DE EXPERIENCIA DE REALIDAD MIXTA

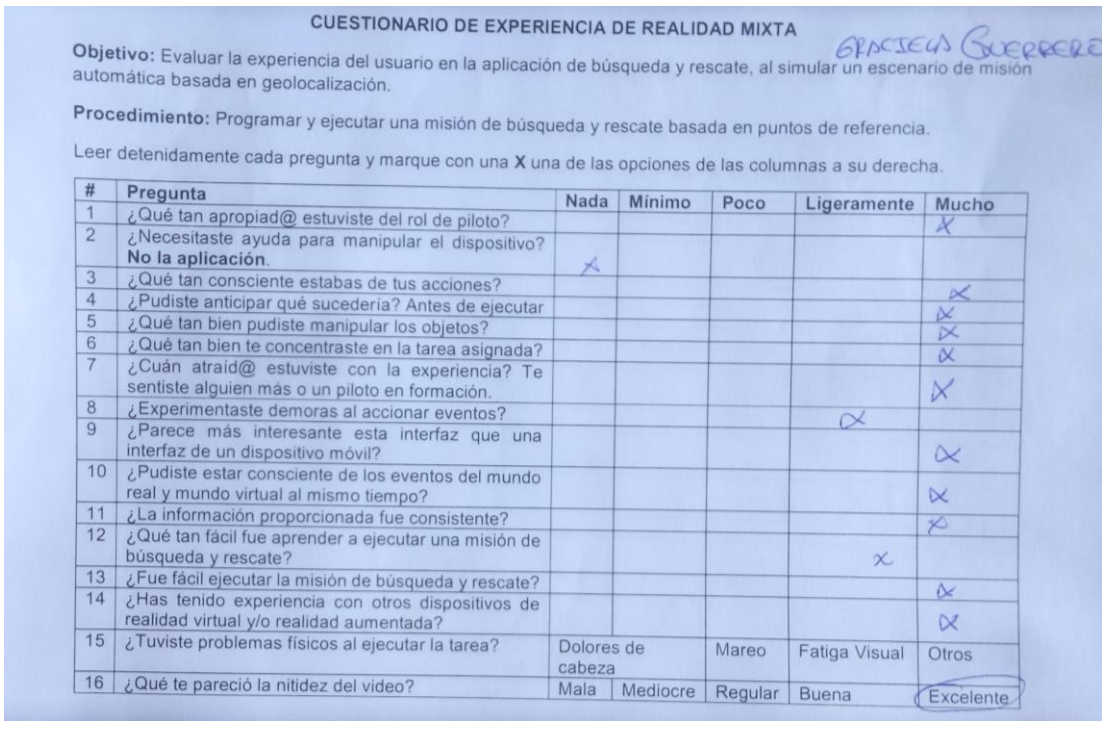
Objetivo: Evaluar la experiencia del usuario en la aplicación de búsqueda y rescate, al simular un escenario de misión automática basada en geolocalización.

José Guamán

Procedimiento: Programar y ejecutar una misión de búsqueda y rescate basada en puntos de referencia.

Leer detenidamente cada pregunta y marque con una **X** una de las opciones de las columnas a su derecha.

#	Pregunta	Nada	Mínimo	Poco	Ligeramente	Mucho
1	¿Qué tan apropiad@ estuviste del rol de piloto?					/
2	¿Necesitaste ayuda para manipular el dispositivo? **No la aplicación**.	/				
3	¿Qué tan consciente estabas de tus acciones?					/
4	¿Pudiste anticipar qué sucedería? Antes de ejecutar				/	
5	¿Qué tan bien pudiste manipular los objetos?				/	
6	¿Qué tan bien te concentraste en la tarea asignada?					/
7	¿Cuán atraíd@ estuviste con la experiencia? Te sentiste alguien más o un piloto en formación.					/
8	¿Experimentaste demoras al accionar eventos?		/			
9	¿Parece más interesante esta interfaz que una interfaz de un dispositivo móvil?					/
10	¿Pudiste estar consciente de los eventos del mundo real y mundo virtual al mismo tiempo?				/	
11	¿La información proporcionada fue consistente?					/
12	¿Qué tan fácil fue aprender a ejecutar una misión de búsqueda y rescate?				/	
13	¿Fue fácil ejecutar la misión de búsqueda y rescate?					/
14	¿Has tenido experiencia con otros dispositivos de realidad virtual y/o realidad aumentada?					/
15	¿Tuviste problemas físicos al ejecutar la tarea?	Dolores de cabeza		Mareo	Fatiga Visual	Otros
16	¿Qué te pareció la nitidez del video?	Mala	Mediocre	Regular	Buena	Excelente



CUESTIONARIO DE EXPERIENCIA DE REALIDAD MIXTA

GRACIAS GUERRERO

Objetivo: Evaluar la experiencia del usuario en la aplicación de búsqueda y rescate, al simular un escenario de misión automática basada en geolocalización.

Procedimiento: Programar y ejecutar una misión de búsqueda y rescate basada en puntos de referencia.

Leer detenidamente cada pregunta y marque con una **X** una de las opciones de las columnas a su derecha.

#	Pregunta	Nada	Mínimo	Poco	Ligeramente	Mucho
1	¿Qué tan apropiad@ estuviste del rol de piloto?					X
2	¿Necesitaste ayuda para manipular el dispositivo? **No la aplicación**.	X				
3	¿Qué tan consciente estabas de tus acciones?					X
4	¿Pudiste anticipar qué sucedería? Antes de ejecutar					X
5	¿Qué tan bien pudiste manipular los objetos?					X
6	¿Qué tan bien te concentraste en la tarea asignada?					X
7	¿Cuán atraíd@ estuviste con la experiencia? Te sentiste alguien más o un piloto en formación.					X
8	¿Experimentaste demoras al accionar eventos?				X	
9	¿Parece más interesante esta interfaz que una interfaz de un dispositivo móvil?					X
10	¿Pudiste estar consciente de los eventos del mundo real y mundo virtual al mismo tiempo?					X
11	¿La información proporcionada fue consistente?					X
12	¿Qué tan fácil fue aprender a ejecutar una misión de búsqueda y rescate?				X	
13	¿Fue fácil ejecutar la misión de búsqueda y rescate?					X
14	¿Has tenido experiencia con otros dispositivos de realidad virtual y/o realidad aumentada?					X
15	¿Tuviste problemas físicos al ejecutar la tarea?	Dolores de cabeza		Mareo	Fatiga Visual	Otros
16	¿Qué te pareció la nitidez del video?	Mala	Mediocre	Regular	Buena	Excelente

Anexo 3: Publicaciones

SAR missions with commercial UAVs over Mixed-Reality interfaces

Raul Rosero[1,4]
2170051@my.ipleiria.pt

Diego Marcillo[4]
dmmarcillo@espe.edu.ec

Carlos Grilo[1,2]
carlos.grilo@ipleiria.pt

Catarina Silva[1,3]
catarina@ipleiria.pt

[1]School of Technology and Management, Polytechnic Institute of Leiria, Portugal

[2]CIIC, Polytechnic Institute of Leiria, Portugal

[3] Center for Informatics and Systems of the University of Coimbra, Portugal

[4] Universidad de Las Fuerzas Armadas, ESPE, Ecuador

Abstract

The use of unmanned aerial vehicles (UAV) has become an important tool in Search And Rescue (SAR) applications, because these are used to human lives preservation and quickly exploring natural of disaster affected areas. Also, human body detection has been possible through algorithms which analyse optical and thermal images obtained from the installed cameras.

On the other hand, Ground Stations (GS) with First Person Vision (FPV) interfaces have been implemented with Augmented Reality (AR). Nevertheless, satisfactory projects with commercial UAVs are not common, because these drones are difficult to control for non-expert pilots. These are the reasons why we propose the creation and implementation of an architecture with these requirements. Also, we aim at providing a more user immersive interaction through Mixed Reality (MR) interface over Head-Mounted Display (HMD) glasses.

Keywords: Unmanned Aerial Vehicle, Search and Rescue, Mixed Reality, First Person Vision, Head-mounted Display

1 Introduction

The use of UAVs in SAR operations has protected rescue groups' lives, permitting pilots to be aware of the environment by remotely controlling only the aircraft flight around disaster zones. Injured people location on natural disasters affected areas has been possible thanks to the installation of optical and/or thermal cameras on aerial vehicles and to the transmission of their recorder images in real time to a GS-Ground Station.

In [1] and [2], optical images have been analysed, while in [3], [4], and [5], optical and thermal images have been explored over cross-reference algorithms, allowing the location of human bodies in dark spaces. Nevertheless, [6] mentions that providing operations through First Person Vision interfaces with smart-glasses, GS interfaces of these projects could be more immersive, allowing non-expert users to interact in an easier way.

FPV in smart-glasses provides an interaction of virtual reality with the physical reality throughout the use of networks, sensors, and data bases. On [3], the interaction is called Mixed Reality (Augmented Reality + Virtual Reality) when the pilot sends data to the physical world and does not see the UAV. In [7], the use of commercial UAVs is recommended for disaster management because of their availability, affordability and easy to use.

The target of this work is to implement an architecture that displays an FPV-MR interface for human detection over optical and thermal images in SAR missions, using a commercial UAV that provides mission planning, take-off, landing, intelligent flight modes, expertise modes and a Software Development Kit (SDK).

This paper is structured according to discussion points inside the proposed architecture. In Section II, SAR possible scenarios are described. In Section III, types, advantages, and disadvantages of commercial UAVs are discussed. Then, in Section IV, methods for human body detection and video transmission. Next, in Section V, devices for MR are present with their software provided for developers are presented. In Section VI, the prototype architecture is presented and, finally, future work for the project is delineated.

2 SAR Scenarios

SAR operations must be conducted quickly and efficiently during the first 72 first hours after the disaster hits [7], hours when injured people have more probability to resist extreme conditions. Geophysical, hydrological, climatological, hydrographical or human-induced disasters could be managed satisfactorily with AUVs because climate conditions have low interferences on the communication with the Ground Station.

In [2], a rescue scenario is described in two parts of a big problem. The first one mentions the identification and injured people location and the second one consists in delivering supplies or resources, but only the first problem is possible due to the physic characteristics of commercial AUVs.

There are two types of SAR operations, Urban SAR and Mountain Rescue [8]. Depending on the features of UAVs, one or both types could be possible. Also, the number of aerial vehicles is an important topic in this operation, because UAVs can organize in teams. Though, [7] mentions that the use of multiple drones in SAR missions does not ensures a satisfactory task.

3 Unmanned Aerial Vehicles

UAVs, also called drones, come in different sizes, from microdrones to large military UAVs [8]. Flight type (rotary-wing or fixed-wing impact) also affects drones' performance in SAR missions. Fixed-wing UAVs move quickly, being ideal for surveying areas and structural inspections, while rotary-wing (helicopters and multi-copters) can do more tedious tasks such as detailed aerial inspection, supply delivery, photography, and filmography.

In [1], [2], and [9] non-commercial helicopters were piloted by experts and had successful SAR missions, while in [4] and [6] commercial quadcopters were easier piloted, because these provide characteristics such as mission planning, take-off, landing and gimbal movements (roll, pitch). Table 1 shows the advantages and disadvantages of different types of UAVs.

UAV Type	Advantages	Disadvantages	
Fixed-wing	Large area coverage	Inconvenient launch and landing	Price
Helicopter	Hover flight Single rotor	Harder to flight than Multicopter	
Multicopter	Availability price Hover flight	Low payload Short flight duration	

Table 1: Comparison of Unmanned Aerial Vehicles for Search and Rescue operations.

4 Human Detection over Live Video

For Live Video with image analysis, this project proposes the use of a similar model to [6], because it defines an architecture to provide video frames to the Ground Station. Here, the video is passed from the encoder (UAV) to the computing module (GS), where the Computer Vision (CV) works.

Classic human detection algorithms can be implemented together over optical images and thermal image to detect humans in bad weather conditions [4]. Or, a better performance could be obtained with the use of the open source library OpenCV, because contains more than 2500 optimized algorithms for object detection and need only use pre-trained classic algorithms to detect specific objects. We are thinking to use the second approach for this project, with the use of Histogram of Oriented Gradients HOG for optical images and Haar Cascade based feature for thermal images like in [4] and [11].

5 Remote Controller and Mixed Reality

The difference between AR and VR can be confused. AR puts digital information in the real world such as HoloLens [10] devices, while VR creates a digital context that simulates the real world such as Oculus Rift and also HoloLens. Nevertheless, as MR systems are sensor-driven, the software architecture is based on data flow.

Mixed Reality glasses as FPV Ground Stations for amateur pilots or video-gamers could provide them with a similar experience to video games or simulator of SAR applications, such as in [8], making tasks easier to complete.

UAVs need to be controlled when pilots cannot see them. For this reason, display control tools are necessary inside MR interfaces. Nowadays, more than one model of glasses with these features can be acquired because Microsoft among others such as Dell, HP, Asus, Samsung and Lenovo also developed accessible Head-mounted Displays (HMD) over Windows 10 operating system. In order to make MR interfaces over that, the game developing platform Unity is required as illustrated in Figure 1, as well as an implementation of a central server interpreter between the drone and the glasses.

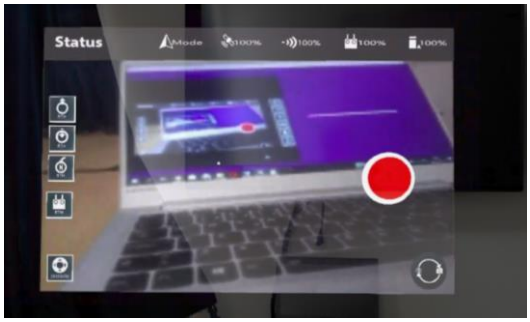


Figure 2: Remote controller prototype in Unity 2017

6 Architecture prototype

Figure 4 illustrates a prototype of the SAR architecture application to be developed along the project with some features described in the previous sections. Here, commercial drones can connect with an application throughout an SDK and a "Connection" module. This module connects with the "Video Encoder" module and the interpreter central server, which implements the "Human Detection" with one "Interpreter" module. In the FPV-MR interface, a MR application implements a "Media Receptor" and a "Remote Controller" module in order to control the drone at a distance.

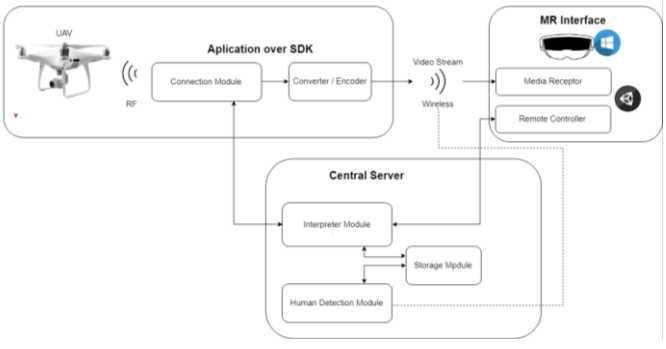


Figure 4: Proposed architecture prototype.

6.1 Real Time Data Sharing Definition

To implement an energy and battery saving mechanism, the UAV application defines a Data Sharing Definition in the "Connection Module". Exists data which is sent form the "Connection Module" to the server such as a response to remote control events or is shared at time intervals. Also, the "Interpreter Module" on the central Server and the "Remote Controller Module" on the MR application transport this data with the same concept.

System state, battery state and connection state are data sent at short time intervals to prevent the drone's health. While GPS position, altitude, distance, actual position and distance travelled are sent at short time intervals according the type of SAR mission. On another hand, data of state action events are sent from the drone to the remote controller through the Central Server, such as a response to remote controller events. These data correspond to the drone's configuration or action events and is sent at first instance from the "Remote Controller" such as: flight distance permitted, origin position, GPS mission coordinates, pilot experience, mission type, take-off, landing, copter movements, and gimbal camera movements.

6.2Data Persistence

Information corresponding to configurations such as: flight distance permitted, origin position, GPS mission coordinates, pilot experience and mission type need to be saved in storage through the "Interpreter Module" and the "Storage Module", to save state configurations of the applications and mission planning configurations. Also, data corresponding to the human detection results will be saved to emit mission reports. This, task is made through the connection of the "Storage Module" with the "Human Detection Module" and the Live Video.

7 Future Work

Improvements to the architecture prototype and its implementation will be made according to the limitations of commercial multi-copter drones, which provide the minimal features described before. Next, the modules of HMD application need to be defined to use the most possible MR glasses models. Models, which provide MR interfaces with Unity over Windows 10.

On the other hand, modules of the central server will be adjusted, integrating the human recognition over thermal and optical images transmitted and displayed in the MR interface in real time. Finally, the main goal of this project is to present a generic architecture, that is, agnostic regarding the commercial UAV or the smart-glasses model, which could be used in SAR systems.

References

[1] P. M. e. al., "Searching lost people with UAVs: The system and results of the Close-Search project," Infoscience, Melbourne, 2012.

[2] P. R. Patrick Doherty, "A UAV Search and Rescue Scenario with," Springer, Link¨oping, 2007.

[3] A. B. e. al., "Safety, Security, and Rescue Missions with an Unmanned Aerial Vehicle (UAV)," Springer, Netherlands, 2010.

[4] M. S. Rapee Krerngkamjornkit, "Human Body Detection in Search and Rescue Operation Conducted by Unmanned Aerial Vehicles," Trans Tech Publications, Switzerland, 2013.

[5] P. D. Piotr Rudol, "Human Body Detection and Geolocalization for UAV Search and Rescue Missions Using Color and Thermal Imagery," IEEE, Linkoping, 2008.

[6] X. X. T. H. Xiaoyue Ji, "Data-□riven Augmented Reality Display and Operations for UAV Ground Stations," IEEE, Chongqing, 2017.

[7] M. E. e. al., "Help from the Sky: Leveraging UAVs for Disaster Management," IEEE, Anaheim, 2017.

[8] A. S. K. e. al., "An Architecture for Simulating Drones in Mixed Reality Games to Explore Future Search and Rescue Scenarios," Open Track, New York, 2018.

[9] J. M. e. al., "A Surveillance Task for a UAV in a Natural Disaster Scenario," IEEE, Rio de Janeiro, 2012.

[10] Microsoft, "Microsoft HoloLens," Microsoft, 2018. [Online]. Available: https://www.microsoft.com/en-us/hololens. [Accessed 9 9 2018].

[11] R. Ribeiro, J. M. Fernandes and A. Neves, "Face Detection on Infrared Thermal Image," in *University of Aveiro*, Aveiro, 2016.

Human Detection for Search and Rescue Applications with UAVs and Mixed Reality Interfaces

Raúl Llasag[1,2], Diego Marcillo[1], Carlos Grilo[2, 3], Catarina Silva[2,4]

[1]Departamento de Ciencias de la Computación, Universidad, ESPE, Sangolquí, Ecuador

[2] Escola Superior de Tecnologia e Gestão, Instituto Politécnico de Leiria, Leiria, Portugal

[3] CIIC – Centro de Investigação em Informática e Comunicaçoes, Instituto Politécnico de Leiria, Leiria, Portugal

[4]CISUC - Centro de Informática e Sistemas da Universidade de Coimbra, Portugal

{rhllasag, dmmarcillo}@espe.edu.ec; {carlos.grilo, catarina}@ipleiria.pt

Abstract — **Human Detection (HD) in natural disaster areas with the use of Unmanned Aerial Vehicles (UAV) or drones is an important approach to preserve human lives in Search and Rescue (SAR) operations, allowing to rescue injured people through previous and fast exploration from safer zones. Nevertheless, constraints on artificial vision and interactive interfaces remain a challenge, namely the ones related to Computer Vision (CV) techniques, since HD cameras drones need to transmit video streams in real-time to Ground Stations (GS). Additionally, GS interfaces of this type of applications are usually not easy to use for unexperienced pilots. In this work, we propose an architecture for SAR operations, integrating an interactive interface that can be used with commercial UAV and Head-Mounted Display (HMD) devices, which provide Mixed Reality (MR) for GS. Results on an application prototype show that a fine tunning process of a non-traditional object detection model with SAR images can improve human detection precision with real-time video streaming.**

Keywords – Human Detection; Unmanned Aerial Vehicles; Search and Rescue; Computer Vision; Mixed Reality; Head-Mounted Display.

I. INTRODUCTION

Through the years, efforts have been made so that faster human assistance on natural disasters can be provided. In this regard, the use of Unmanned Aerial Vehicles (UAV) is being explored as a way of quickly providing information about the actual situation of affected zones. Additionally, for automatized people location, Computer Vision (CV) techniques can be used to search humans on video frames captured with the camera's UAV [1].

In general, the video is transmitted from the UAV to the Ground Station (GS), which then uses detection algorithms. However, Human Detection (HD) with UAVs is not an easy task because it has associated problems, such as poor detection when the cameras rotate over their axis, when people take different postures, when the human body is incomplete on the image, poor scene illumination, among others [2].

In [3], [4] and [5], poor detection issues were partially solved using traditional machine learning techniques and training models with single class datasets (see Section VI) combinations. [2] provide better detection results using the same single class datasets combinations and deep learning techniques. However, these authors mention that there is a tradeoff between accuracy and processing speed.

Besides that, [6] mentions that beginner pilots have trouble interacting with GS interfaces that usually provide 2D screens with complex tasks. In an attempt to mitigate this problem, Head-Mounted Displays (HMD) for GS interfaces with 3D augmented reality are used in [6]. In [2], a smart-glass is proposed to train beginner pilots, but the proposed approach was only tested on simulated environments. In [6] and [7] Virtual Reality (VR) interfaces display images captured from the UAV and its route over a map. However, with the last interface, pilots can not directly see the UAV, restring them to be aware of the UAV real position during the flight.

To our knowledge, until today, no project has proposed an architecture for SAR with HD that allows pilots to be aware of UAVs real-position and controlling them with a Mixed Reality (Augmented Reality + Virtual Reality) application installed on a MR headset or on a HMD. In this paper, we propose an architecture that addresses the problems mentioned above.

The rest of this paper is structured as follows: In Section II, SAR operations are described. In Section III, commercial drones and MR headsets models are introduced. Then, in Section IV, the main features of the system architecture are detailed. Next, in Section V, object detection approaches and their object detection metrics are presented. In Section VI, human detection over streaming video is detailed according to the constraints found. In Section VII, a fine-tuning process explains how detection accuracy can be improved. Finally, conclusions on the challenges tackled for a successful application and future works are present in Section VIII.

II. SEARCH AND RESCUE OPERATIONS

In [8], a rescue scenario is described in two parts of a larger problem. The first one is related to the identification and location of injured people and the second one consists in delivering supplies or resources.

However, it is mentioned that only the first problem is tackled due to the physic characteristics of commercial drones. Table 1 shows the advantages and disadvantages of different types of drones.

TABLE I. DRONE TYPES

Drone type	Advantages	Disadvantages	
Fixed-wing	Large area coverage	Inconvenient launch and landing	Price
Helicopter	Hover flight	Harder to flight than multicopter	
Multicopter	Availability price Hover flight	Low payload Short flight duration	

According to [9], the use of drones for SAR operations is recommended only on climate conditions with low communication interferences. It is also, mentioned that fixed-wing drones quickly provide knowledge about affected zones while rotary-wing drones can stabilize on the air for a change of flight direction, making more detailed inspections, for example, human detection.

Other considerations are the type of SAR application and the drone's flying complexity. In [8], [10] and [11] non-commercial helicopters were piloted by experts and had successful SAR missions, while in [6] and [12] commercial quadcopters were easily piloted, because these provide characteristics such as mission planning, take-off, landing and gimbal movements (roll, pitch, yaw). Our project is focused on Urban SAR operations with the use of commercial drones because they provide software development kits (SDK), assistance to beginner users and intelligent flight modes.

III. COMMERCIAL DRONES AND MR HEADSETS

We propose a general architecture to be implemented using tools of semi-autonomous drones and a MR headset because semi-autonomous drones minimize to effort needed by inexperienced pilots in order to accomplish complex tasks. Also, there are software development tools for MR headsets that allow the creation of interactive interfaces.

A. *Semi-autonomous drones*

In [13], DJI[1] and Parrot[2] are mentioned as the two first companies on sales position. Parrot offers only VR experiences over HMD with the BEBOP-2[3] model, while DJI with 70% of the global market does not have a HMD included in their model's boxes. The success factors of these drone models are semi-autonomous control, low prices and integration with the developer community thanks to software development kits for Android and iOS mobile applications.

B. *Mixed Reality headsets*

Mixed Reality interfaces for headsets started with the HoloLens[4] model created by Microsoft in 2015. The purpose of these devices was mixing Virtual Reality (VR) with Augmented Reality (AR) through 2D and 3D hologram projections. Today, other VR/AR models are present on the market such as the second version of HoloLens, Magic Leap[5] and Meta 2[6].

The HoloLens headset differs from other head-mounted displays (HMDs) on the fact that it provides spatial computing and user mobility. In other words, HoloLens runs its operating system on the display hardware and projects holographs based on a spatial mapping. On the other hand, Magic Leap handles the primary data and graphics processing with a spatial mapping approach on a small hip-mounted computer, while Meta 2 needs a computer connected for data processing. However, all these headset models allow creating MR interface applications with hand gestures and voice using the Unity engine.

IV. SYSTEM ARCHITECTURE

Figure 1 illustrates the SAR architecture system proposed and implemented in this work. It consists of three applications. The first one uses methods to configure and control the UAV (UAV-App), the second one is a Mixed Reality application (MR-App) and the third one is a Communication Central Server (CCS) that acts as an interpreter between the UAV-App and the MR-App.

Architecture features and mechanisms implemented are presented in three main parts. The first details the real-time data sharing and data persistence necessary to configure and control the drone flight. The second details the rendering and presentation modules on the MR-App. Finally, the third part details video transformations carried out to provide streaming video with computer vision.

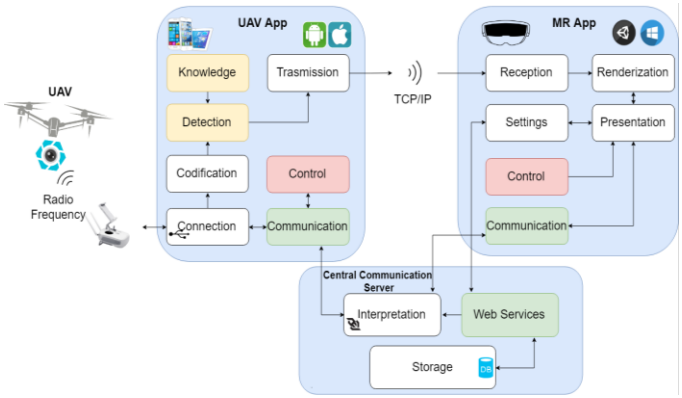


Figure 1. SAR architecture

A. *Real-Time Data Sharing and Data Persistence*

Real-time data sharing definitions are implemented on the communication and settings modules in the UAV-App and the

[1] https://www.dji.com
[2] https://www.parrot.com/global/drones
[3] https://www.parrot.com/global/drones/parrot-bebop-2-fpv
[4] https://www.microsoft.com/en-us/hololens
[5] https://www.magicleap.com
[6] https://www.metavision.com

MR-App. These conditions were defined in order to implement an energy and battery saving mechanism and are illustrated in Figure 2.

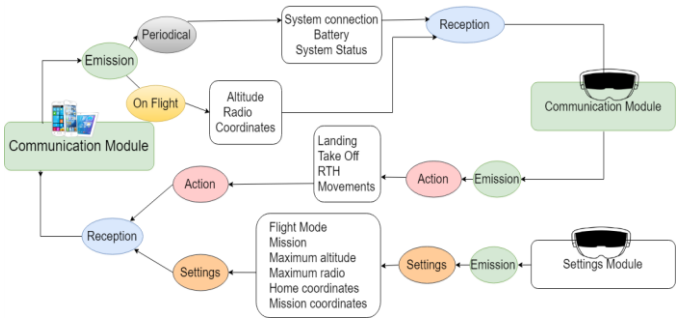


Figure 2. Real-Time Data Sharing

In this project, a battery saving mechanism specifies the emission and reception of data only when sensor data is updated or when an event is activated on the control and settings modules. The data emission from a UAV-App could be done only when the system or sensor status need to be updated after receiving data on the MR-App interface. However, data emission is also activated when the drone is flying, for drone's location transmission.

On the other hand, the MR-App emits data according to flight action events or setting events. In the first case, data come from the *Communication* module, while in the second case, data come from the *Settings* module. This module interchanges data with the *Storage* module located on the CCS throughout a *Web service* module in order to save system configurations.

Information corresponding to the configuration, such as, flight distance allowed, origin position, GPS mission coordinates, pilot experience and mission type need to be saved in storage through the *Web service* module and the *Storage* module to save drones configurations and mission planning routes. We tested with a Phantom 4 drone model and the first version of a MR headset called HoloLens, as illustrated in Figure 3.

Figure 3. Phantom 4 controlled by HoloLens

B. *Mixed Reality Interface*

The holograms projected in Figure 4 are texture instances located on the *Presentation* module that loads data from the *Control*, *Settings*, *Communication* and *Rendering* modules. This data is directly updated when the *Control* and *Settings* modules events are activated. However, this data may also be indirectly updated when it is received from the *Communication* module through an observer pattern implementation or when it is received from the *Rendering* module in bitmap format, providing video streaming service.

Figure 4. Mixed Reality or VR/AR interface

C. *Streaming video with Computer Vision*

Images projection on the reality, such as illustrated in Figure 4, renders bitmaps transmitted from the *Transmission* module to the *Reception* module throughout a TCP/IP socket connection. However, this connection type could experiment disconnections on a poor signal of a wireless network. For this reason, *Transmission* and *Reception* modules implement a channel reconnection mechanism. This mechanism is supported by connection and disconnection events implemented on the *Interpretation* module.

Besides disconnections, video transmissions could also suffer interruptions when buffers have used all the memory buffer space. For this reason, UAV-App and MR-App implement a synchronization mechanism based on Presentation Time Stamp (PTS) on the transmission and reception modules.

The synchronization process implemented is based on time video processing by the codification and detection modules. On the first step of this process, the video is stored on a UAV-App by a codification module only when a connection module has established that the application can receive data from the drone. Then, the H264 YUV240 video format caught by a codification module is downsized and converted to a four-colour channels video format to fit the detection module methods. Here, a classification method uses a pre-trained model for object (or human) detection that is stored on the knowledge module.

V. OBJECT DETECTION APPROACHES

The proposed architecture could use traditional and deep learning detection methods, and both were tested in this work. Each approach has their own advantages and disadvantages.

A. *Traditional methods*

Some of the most recommended ([4], [14] and others) traditional methods for human detection are Support Vector Machines (SVM) and Adaptive Boosting (AdaBoost). These are fed by features obtained with Histograms Oriented Gradients (HOG), Haar and others.

B. *Deep Learning models*

Regarding deep learning models, there are several algorithms that can be used to train real-time detection models as, for example, Faster Region Convolutional Neural Networks (Faster R-CNN) [15], Single Shot Detector (SSD) [16], and all the versions of You Only Look Once (YOLO) [17] [18] [19].

C. *Training and evaluation*

The training process is preceded by preparation step that has three main phases. The first one is where images are collected (dataset). In the second phase, object instances are labeled. In the third step, the dataset is split into training data and test data. In this last step, split combinations are usually used to increase the model knowledge.

The model is trained during some number of iterations. On each iteration, the model's output for all dataset object instances is computed and the number of true-positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) are counted. In the end, error metrics are computed using equations (1), (2) and (3) or using the Area Under a Curve (AUC) of the Precision and Recall graph.

$$Acurracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

In deep learning approaches, the Intersection Over Union (IoU) metric is used to determinate when a classification is accepted. The IoU is a division of the intersection versus the union of a Ground-Truth bounding box and a Predicted bounding box as illustrated in Figure 5 for a face detection sample. This value indicates de accuracy detection of a dataset and is used to establish correct detections.

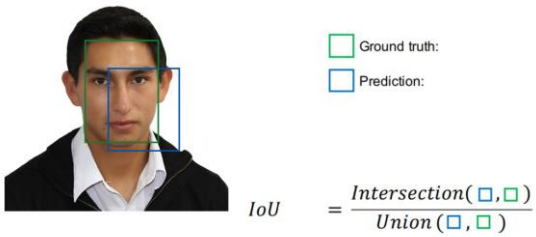


Figure 5. IoU metric

Works using deep learning based approaches, such as [16], [17], [18] and [19], also use a Precision and Recall variation called mean Average Precision (mAP) metric. This metric calculates the detection error with multiple object classes. To obtain the mAP value, an Average Precisions (AP) per object class using the same interpolation, is required. Also, it is possible to use multiple IoUs to specify more than one umbral, creating the mAP@ notation. The @ character specifies an initial umbral and the step between the next ones. For example, mAP [0.5, 0.05, 0.95] means that the model is tested with an umbral greater-than or equals to 50%. The Average Precision is computed as follows:

$$AP = \sum_{r=0}^{1}(r_{n+1} - r_n)P_{interp}(r_{n+1}) \quad (4)$$

$$P_{interp}(r_{n+1}) = max_{\tilde{r}:\tilde{r}\geq r_{n+1}} p(\tilde{r}) \quad (5)$$

where r is the recall, $p(\tilde{r})$ is the measured precision at recall $\tilde{r}$ and $P_{interp}(r_{n+1})$, is the maximum precision value that is greater-than or equal to r_{n+1}.

VI. Real-time Human Detection

Search and rescue applications present streaming video constraints to display images in real-time. We mention the "real-time" term to refer a predictable execution time of processing image used by the mechanism based on PTS. However, the streaming video also needs to be supported by an object classification model with good accuracy values. Usually, these two constraints depend on datasets features, training models, image quality and processing unit, topics that are exposed in the next items.

A. *Image Quality*

The drone cameras provided us 40 frames per second (fps) with an image resolution of 1280x720 pixels, but the resolution and fps were downsized to 2 fps and 320x180 pixels to provide a streaming video with fast image processing. These values were obtained making video transmission tests that evaluate fps with downsize percentage, obtaining twenty minutes of video transmission with these values. Obtaining twenty minutes of successfully video transmission is enough because it is the time supported by a flight mission.

In Figure 6 and in Table 2, 14 transmissions without wireless disconnects are illustrated. The sixth, seventh and eighth transmissions obtained a successful video transmission. We can see that transmissions with greater-than 2 fps or better-quality images could obtain fewer transmission times because channel disconnection happens, making reconnection necessary and losing sequences of images while disconnected.

TABLE II. FPS AND QUALITY TRANSMISSION TEST

Transmission #	Minutes	Fps	Quality
1	04:19	1.7142857	27%
2	08:10	1.7142857	25%
3	01:28	3	32%
4	03:11	3	30%
5	15:27	2	32%
6	20:00	2	23%
7	20:00	2	24%
8	20:00	2	25%
9	00:21	2	38%
10	00:50	2	37%
11	00:54	2	39%
12	01:00	2	37%
13	01:16	1.875	33%
14	00:56	1.875	34%

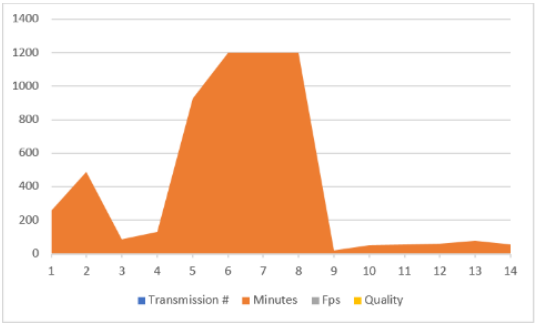


Figure 6. Time (seconds) per transmission

B. *Datasets features*

The main problems found in, [3], [4], [5] and [8] on datasets for human recognition with drones are related with camera movements (roll, pitch, yaw), capture distance and scene illumination. In our case, we do not just deal with the two first ones because we do not have thermal cameras installed.

The camera movements and capture distance could be solved using single class datasets such as Generalized Multi-View Real Training GMVRTv1 [3], GMVRTv2 [3], University of Central Florida-Aerial camera, Rooftop camera and Ground camera UCF-ARG [20], as well as multi-class datasets such as Microsoft Common Objects in Context MS-COCO [21], Visual Object Classes PASCAL-VOC [22], ImageNet [23], SUN [24], MobileNet [25] and others.

There are datasets thought for one object class identification (e.g. is there a person in the image?) and datasets thought for the detection of more than one object class, humans being one of them. In the first case, the dataset contains images with and without human instances. Usually, these images are called positives and negatives, respectively. On the other hand, images in multi-class datasets may contain several object instances.

C. *Detection models*

In [4], it was found that deep learning methods provide better accuracy values than traditional methods, but that the first ones require more execution time. We reached the same conclusion in our experiments because in the first steps of this work, human detection was implemented using the most popular library for traditional methods called Open Computer Vision (OpenCV), with poor accuracy. Then, human detection was improved using deep learning methods over TensorFlow Object Detection Application Program Interface (TensorFlow OB API), but with more processing time per frame.

At the beginning of this project, we tested OpenCV with Haar-like features and the trained model of [26] that provides an average of 90% detection rate. Then, the detection improved using HOG+SVG with the trained models of [27], which provides precision of ~99.4% and recall of ~93.87%. However, the datasets [27] and [26] are not focused on detecting people with drones and lead to a significant number of false-positives as illustrated in Figure 7.

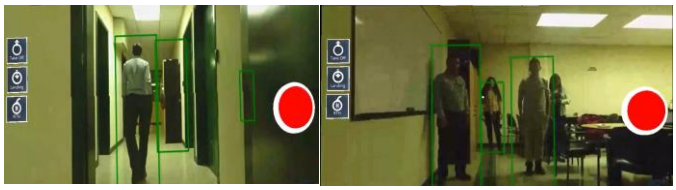


Figure 7. Detection with Haar-like features [27] and HOG+SVG [26]

Training a new model using the datasets proposed by [3], [4], [5] or [8] requires significant resources regarding training time, features definition, splitting, hardware with GPU or faster CPU. To solve this problem, we used deep learning methods such as SSD and YOLO, which can be used with TensorFlow OD API and datasets thought for multi-call detection.

The Pre-trained model *ssd_mobilenet_v1_coco* presents acceptable metric values according to mAP and speed presented in Table 3. On the other hand, *tiny-yolo-voc-graph* [28] is able to recognize the presence of a person with ~0.94% probability.

TABLE III. MAP AND SPEED OF TENSORFLOW MODELS [29]

Model Name	**Speed (ms)**	**mAP**	**Dataset with COCO**
ssd_mobilenet_v1_coco	30	21%	MobileNets
tiny-yolo-voc-graph	**Probability** ~0.94%		PASCAL VOC

Human detection was considerably improved using the faster model *ssd_mobilenet_v1_coco* presented in Figure 8, however, human recognition constraints over angle movements exist yet. In [30], it is shown that transfer learning processes, such as fine-tuning, a model can improve accuracy percentages in specific classes. Also, this theory was demonstrated by [31] when training a MS COCO dataset using a computer with a faster processing unit. For these reasons, we applied a fine-tuning process of deep learning models using traditional datasets to obtain better detections.

Figure 8. Human Detection with *ssd_mobilenet_v1_coco*

D. Processing Units

The type of processing unit could be the main factor that affects the data processing time to offer video in real-time. In [4], it is mentioned that laptops usually have more computational resources than mobile devices to process the data, for example, a faster Central Processing Unit (CPU) or a Graphics Processing Unit (GPU).

We used devices without GPUs for human recognition, but we also used a virtual machine with GPU to transfer knowledge from an object detection model to a human detection model. In [32], the authors found that, unsurprisingly GPUs significantly outperform CPUs machines on training object detection models with TensorFlow.

VII. Transferring Knowledge to a New Model

Transfer learning focuses on storing knowledge gained while solving one problem and applying it to a different but related problem [33]. Fine-tune an object detection model is a transfer learning process that improves the classification in a subset of categories [31]. This process requires the change of some aspects of the neural network architecture to improve the object classification of a new model. To fine-tune the *ssd_mobilenet_v1_coco* model that is based in COCO dataset, we used TensorFlow tools such in [31].

The preparation process was split in the three phases mentioned in Section V. In the first phase, we choose 440 videos of 10 human actions of the UCF-ARG [20] dataset. This dataset was chosen because it contains images caught in less than 30.45 meters of altitude. Then, we played all the videos, catching an image per second in jpeg format, which led to a training dataset of 1947 images and a test dataset of 493. In the second phase, illustrated in Figure 9, we labeled the training images in the PASCAL VOC format, which uses XML annotations.

In the third phase, the training and evaluation record files required to the fine-tuning were created. Here, we wrote a pipeline file and hyper-environments to train with GPU resources according to the file structure in Figure 9. The model was trained in a virtual machine with Nvidia CUDA GPUs such as recommended in [34].

Finally, the training process led to a model with an accuracy of 0.89 for human detection.

Figure 9. Labeling persons in UCF-ARG images and file structure

VIII. Results and Future Work

This paper presented an architecture for SAR operations which uses a Mixed Reality based interface. The human recognition process is also presented, as well as the training process of the detection model. The trained model has now a detection accuracy of 0.89. As future work we plan to include human recognition with thermal images.

References

[1] P. Doherty. e. al., "The WITAS Unmanned Aerial Vehicle Project," *14th European Conference on Artificial Intelligence*, Madeira, 2000.

[2] De Oliveira and M. Wehrmeister, "Using Deep Learning and Low-Cost {RGB} and Thermal Cameras to Detect Pedestrians in Aerial Images Captured by Multirotor UAV," Sensors, 2018.

[3] P. Blondel, A. Potelle, C. Pégard and R. Lozano, "Fast and viewpoint robust human detection for SAR operations," in *2014 IEEE International Symposium on Safety, Security, and Rescue Robotics*, Hokkaido, 2014.

[4] P. Blondel, A. Potelle, C. Pégard and R. Lozano, "Fast and viewpoint robust human detection for SAR operations," IEEE, Hokkaido, 2014.

[5] P. Blondel, A. Potelle, C. Pégard and R. Lozano, "Human detection in uncluttered environments: From ground to UAV view," IEEE, Singapore, 2014.

[6] X. Ji, X. Xiang and T. Hu, "Data-driven augmented reality display and operations for UAV ground stations," IEEE, Chongqing, 2017.

[7] M. Prexl, K. Struebig, J. Harder and A. Hoehn, "User studies of a head-mounted display for search and rescue teleoperation of UAVs via satellite link," IEEE, Big Sky, 2017.

[8] P. Doherty and P. Rudol, "A UAV Search and Rescue Scenario with Human Body Detection and Geolocalization," Springer, Berlin, 2007.

[9] M. Erdelj. e. al., "Help from the Sky: Leveraging UAVs for Disaster Management," IEEE, 2017.

[10] P. Molina. e. al., "Searching lost people with UAVs: The system and results of the Close-Search project," Infoscience, Melbourne, 2012.

[11] J. Montenegro. e. al., "A Surveillance Task for a UAV in a Natural Disaster Scenario," IEEE, Rio de Janeiro, 2012.

[12] R. Krerngkamjornkit and M. Simic, "Human Body Detection in Search and Rescue Operation Conducted by Unmanned Aerial Vehicles," Advanced Materials Research, Switzerland, 2013.

[13] A. Iglesias, "Las 20 compañías que dominan el mercado de los drones," 2017. [Online]. Available: https://www.ticbeat.com/tecnologias/las-20-companias-que-dominan-el-mercado-de-los-drones/. [Accessed 27 1 2019].

[14] Y. Wang, J. Xing, X. Luo and J. Zhang, "Pedestrian Detection Using Coarse-to-Fine Method with Haar-Like and Shapelet Features," IEEE, Ningbo, 2010.

[15] Ren Shaoqing. e. al., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE, 2016.

[16] Liu Wei. e. al., "SSD: Single Shot MultiBox Detector," Springer, 2016.

[17] Redmon Joseph. e. al., "You Only Look Once: Unified, Real-Time Object Detection," IEEE, 2016.

[18] Redmon J. and Farhadi A., "YOLO9000: Better, Faster, Stronger," IEEE, 2017.

[19] Redmon J. and Farhadi A, "YOLOv3: An Incremental Improvement," CoRR, 2018.

[20] Nagendran A., Harper D. and Shah M., "New system performs persistent wide-area aerial surveillance," SPIE Newsroom, 2010.

[21] Lin T. e. al., "Microsoft COCO: Common Objects in Context," in *Cornell University*, Springer, 2015.

[22] Everingham M. e. al., "The PASCAL Visual Object Classes Challenge: A Retrospective," Springer, 2015.

[23] Deng Jeng. e. atl., " ImageNet: A large-scale hierarchical image database," IEEE, Miami, 2009.

[24] Xiao J. e. al., "SUN Database: Large-scale Scene Recognition from Abbey to Zoo," IEEE, San Francisco, 2010.

[25] Howard Andrew. e. al, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision," 2017.

[26] Shanshan Zhang, "Informed Haar-like Features Improve Pedestrian Detection," IEEE, Columbus, 2014.

[27] Vision-ary.net, "BOOST THE WORLD: PEDESTRIAN DETECTION, [Online]. Available: http://www.vision-ary.net/2015/03/boost-the-world-pedestrian/. [Accessed 1 10 2018].

[28] TensorFlow, "Android YOLO with TensorFlow Mobile," TensorFlow, 18 5 2018. [Online]. Available: https://github.com/szaza/android-yolo-v2. [Accessed 2 2 2019].

[29] TensorFlow, "Tensorflow detection model zoo," [Online]. Available: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md. [Accessed 2 2 2019].

[30] Girshick R., Donahue J., Darrell T. and Malik J., "Rich feature hierarchies for accurate object detection and semantic segmentation," IEEE, 2014.

[31] Sonntag Daniel, "Fine-tuning deep CNN models on specific MS COCO categories," ResearchGate, 2017.

[32] Lawrence John. e. al., "Comparing TensorFlow Deep Learning Performance Using CPUs, GPUs, Local PCs and Cloud," Semantic Scholar, New York, 2017.

[33] West J., Ventura D. and Warnick S., "A Theoretical Foundation for Inductive Transfer," Spring, 2007.

[34] Lawrence John. e. al., "Comparing TensorFlow Deep Learning Performance Using CPUs, GPUs, Local PCs and Cloud," Semantic Scholar, New York, 2017.