

SISTEMA DE DETECCIÓN Y RECONOCIMIENTO DE PERSONAS MEDIANTE DRON TELLO EDU

SYSTEM FOR DETECTION AND RECOGNITION OF PEOPLE USING DRONE TELLO EDU

Ramiro Fernando Isa Jara ^{1*}

¹ Ingeniero en Electrónica, Control y Redes. Doctor en Ingeniería Electrónica. Docente. Escuela Superior Politécnica de Chimborazo. Riobamba, Chimborazo, Ecuador. ORCID: <https://orcid.org/0000-0001-5624-5822>. Correo: ramiro.isa@esepoch.edu.ec

Jhonnatan Israel Cedeño Duchi ²

² Investigador externo. Ingeniería en Electrónica y Automatización. Riobamba, Chimborazo, Ecuador. ORCID: <https://orcid.org/0000-0003-3803-0283>. Correo: israelcd123@gmail.com

Bryan Germán Iza Valle ³

³ Investigador externo. Ingeniería en Electrónica y Automatización. Riobamba, Chimborazo, Ecuador. ORCID: <https://orcid.org/0000-0002-7442-993X>. Correo: lmlbryger@gmail.com

Diego Ramiro Ñacato Estrella ⁴

⁴ Ingeniero en electrónica control y redes industriales. Magister en Sistemas de Control y Automatización Industrial. Docente. Escuela Superior Politécnica de Chimborazo. Riobamba, Chimborazo, Ecuador. ORCID: <https://orcid.org/0000-0002-7233-9076>. Correo: diego.nacato@esepoch.edu.ec

William Geovanny Zavala Ruiz ⁵

⁵ Investigador externo. Ingeniería en Electrónica y Automatización. Riobamba, Chimborazo, Ecuador. ORCID: <https://orcid.org/0000-0002-9119-3834>. Correo: geova1609@hotmail.com

* Autor para correspondencia: ramiro.isa@esepoch.edu.ec

Resumen

En el presente artículo de investigación se desarrolla el entrenamiento de un algoritmo de reconocimiento e identificación de usuarios, utilizando un vehículo aéreo no tripulado con una cámara de video para el enfoque del rostro, específicamente con el Drone Tello EDU. Dentro del entrenamiento del algoritmo en el software Python con el desarrollador Pycharm se utilizó tres modelos de instrucción FISHER, EIGEN y LBPH los mismos que presentaron resultados eficientes en la detección de patrones de identificación. El modelo EIGEN

y FISHER presentan un mínimo margen de error por otra parte, el modelo de LBPH presenta un menor error de detección, así como un mejor tiempo de respuesta de resultado, en las pruebas realizadas a los usuarios, en función del número de muestras.

Palabras clave: Aprendizaje automático; algoritmo; dron; histograma; reconocimiento; python; imágenes.

Abstract

In this research article, the training of an algorithm for user recognition and identification is developed, using an unmanned aerial vehicle with a video camera for face focusing, specifically with the Tello EDU Drone. Within the training of the algorithm in the Python software with the developer Pycharm, three instruction models FISHER, EIGEN and LBPH were used, which presented efficient results in the detection of identification patterns. The EIGEN and FISHER model have a minimum margin of error on the other hand, the LBPH model has a lower detection error, as well as a better response time of result, in the tests carried out to users, depending on the number of samples.

Keywords: Machine learning; algorithm; drone; histogram; recognition; python; images.

Fecha de recibido: 24/11/2022

Fecha de aceptado: 13/01/2023

Fecha de publicado: 26/01/2023

Introducción

En la presente investigación se aborda un tema que en los últimos años ha tenido un crecimiento notable en cuanto a necesidades y mejoras específicamente en el ámbito de seguridad, el surgimiento de aeronaves pilotadas remotamente ha permitido que el mundo sufra una transformación en la manera de concebir estos vehículos, los drones son un tipo de vehículo autónomo no tripulado capaz de mantener un nivel de vuelo controlado y sostenido. Actualmente los drones con el desarrollo tecnológico han sido llevados a varios usos principalmente en el área de investigación, delivery, cine y militar (Jones, 2017). Un drone con un sistema de control es capaz de realizar posicionamientos milimétricos que permiten realizar acciones en cuestión de segundos y sin el riesgo de que el operador sufra algún tipo de daño, en ese sentido es importante contar con un controlador confiable, seguro y eficiente (Custers, 2016).

Como reto principal para la investigación se tiene diseñar e implementar un sistema de detección y reconocimiento de personas mediante vehículos no tripulados, el interés nace de la posibilidad de aprovechar las características del Tello EDU puesto que es un drone impresionante y programable, perfecto para la educación (Subash et al., 2020). La versatilidad de la utilización de lenguajes de programación como Scratch, Python y Swift permiten abordar la solución desde diferentes perspectivas con una gama de posibilidades muy amplia para cumplir con el objetivo (Saini, 2021).

El Tello EDU cuenta con comandos avanzados e interfaces de datos eficientes. Cuenta con tecnología de control de vuelo DJI, el Tello EDU también admite estabilización de imagen electrónica. Mediante código se

puede comandar varias acciones para que vuele en un enjambre o posiciones complejas y desarrolle increíbles funciones de IA (Lee et al., 2021). Aprovechando sus increíbles características además de su alta definición de toma de imágenes podemos fructificar para crear el sistema de reconocimiento mediante la incorporación de software y hardware (Maalouf et al., 2022).

Para la detección se utilizará el algoritmo de Histogramas de gradientes orientados el mismo que permite la determinación eficiente de las características de forma de las imágenes, considerando que, la técnica en mención no cuenta los valores absolutos de intensidad (Espinal, 2018). Permitiendo una mayor invarianza a brillos, sombras y demás espectros de iluminación, además que presenta un gran nivel de detalle en bordes y texturas de la imagen y al mismo tiempo permite cambios moderados de posición proporcionando resultados eficientes en los sistemas de detección (Morcillo Martínez, 2018).

Estado del arte

En los últimos años el uso de vehículos aéreos no tripulados (UAV) comúnmente conocidos como drones, se ha tornado gran utilidad debido a sus aplicaciones de manera especial la industria, defensa militar, detección y reconocimiento de personas. El estudio de vehículos aéreos no tripulados conlleva un análisis en constante evolución, dentro de la navegación de los drones es trascendental considerar el estudio de evasión de obstáculos ya que de esto depende una navegación autónoma. Dentro de la evasión de obstáculos se parte del análisis del entorno donde se desplegará el UAV, concretamente en la estructura de un cuadricóptero se adapta un servomotor que lleva acondicionado dos sensores láser VL53L0X, el servomotor permite realizar un barrido sobre el eje horizontal de 360 grados alrededor del drone para que los sensores puedan tomar las distancias con los posibles obstáculos. Con los datos obtenidos de distancia y radio de detección del obstáculo se puede representarlos gráficamente para su análisis y validación (Morcillo Martínez, 2018).

Dadas sus características los vehículos aéreos no tripulados pueden ser utilizados para aplicaciones civiles enfocadas desde el punto de vista de monitoreo en tiempo real, seguridad, comunicación y conectividad, donde se evalúa la eficiencia, calidad de servicio, recolección de datos y escalabilidad y confiabilidades de transmisión de datos utilizando los vehículos aéreos no tripulados (Hayat et al., 2016).

Dentro del método de detección y reconocimiento existe un enfoque basado en el registro de imágenes tomado por los UAV a través de la plataforma Raspberry Pi, donde se parte de un algoritmo de entrenamiento para el aprendizaje autónomo por medio de redes neuronales convolucionales. Una vez entrenado el modelo dicho algoritmo se transfieren a la tarjeta de desarrollo cuyas pruebas fueron evaluadas en dos arquitecturas propias del método de aprendizaje profundo, alcanzado una tasa de precisión del 91% con la arquitectura Faster-RCNN y un 88% en la arquitectura SSD MobileNet V2, siendo estos resultados favorables dentro de la detección y reconocimiento de objetos (Sabry, 2021).

A nivel de seguridad los vehículos aéreos no tripulados se utilizan en el monitoreo en tiempo real para la búsqueda y rescate de personas en lugares de difícil acceso, con la ayuda del cuadricóptero Parrot Bebop 2 se implementó dos metodologías para el sistema de patrullaje autónomo con detección de personas, el primero consiste en determinar y seguir la posición actual del drone por medio de GPS obteniendo así el modelo matemático el cual describe la trayectoria de movimiento seguida por el drone. En el segundo método se usa un sistema de percepción el cual captura las imágenes con la cámara monocular. Finalmente, en ambos casos

para la detección de personas se extrae la información y se realiza la clasificación de características en OpenCV un software para aplicaciones de visión artificial y Machine Learning de código abierto (Mar Cornelio, 2019).

Los sistemas de rescate tradicional marítimo generalmente suelen estar supervisados por personas, en la actualidad gracias al desarrollo tecnológico se están realizando investigaciones para el uso de nuevas tecnologías como visión artificial y Deep Learning con el fin de mejorar los resultados de búsqueda de naufragos sobrevivientes. Los vehículos aéreos no tripulados UAV son elementos ideales al momento de realizar una búsqueda y detección de personas, según el análisis bibliográfico se planteó un sistema cuyo objetivo permita la detección de personas para rescate marítimo, cuyo sistema está basado en Deep Learning el mismo que utiliza un modelo YOLOv3 que es un algoritmo de detección de objetos muy utilizado en aplicaciones de visión artificial que ofrece buenas prestaciones de imágenes y video en tiempo real, además se creó un datasheet propio para el entrenamiento de la red. Dicho modelo se implantó en un dron autónomo que navega determinadas áreas marítimas en búsqueda y detección de personas naufragantes logrando resultados con un acierto del 63.90% (Calatayud Ferre, 2022). Lo que demuestra el cumplimiento del objetivo así mismo se plantea una base para futuras investigaciones a fin de contar con un sistema de detección eficiente para salvar vidas.

Materiales y métodos

La metodología para el sistema de detección consiste en el entrenamiento de un algoritmo de reconocimiento donde el sistema está entrenado para adquirir datos por medio de la cámara, con el método del histograma de gradiente se procesan los datos y son cotejados a fin de enviar un resultado de identificación del usuario. Los materiales y herramientas utilizadas para la presente investigación se detallan a continuación:

- Vehículo aéreo no tripulado Drone.
- Cámara de grabación.
- Computadora.
- Software Python, Anaconda.
- Editor de código desarrollador Pycharm.

A. Vehículo aéreo no tripulado Drone

El dron definido por el acrónimo UAV (Unmanned Aerial Vehicle) que traducido al español podría entenderse como Vehículo Aéreo No Tripulado, da nombre a las diferentes clases de aeronaves que tienen la capacidad para volar sin la necesidad de que un piloto se encuentre a bordo. Estos drones pueden ser tripulados de forma remota o pueden volar de forma autónoma siguiendo instrucciones preestablecidas en tierra (Chamayou, 2016).

Principio de funcionamiento:

Las operaciones de la mayoría de los drones no parecen más complejas que jugar un videojuego. Sin embargo, detrás de la sencilla interfaz de usuario hay acelerómetros, giroscopios y otras tecnologías complejas que trabajan para que la mecánica del vuelo del dron sea lo más fluida posible (Reymondin et al., 2019). En la figura 1 se muestran las partes del Dron Tello Edu.

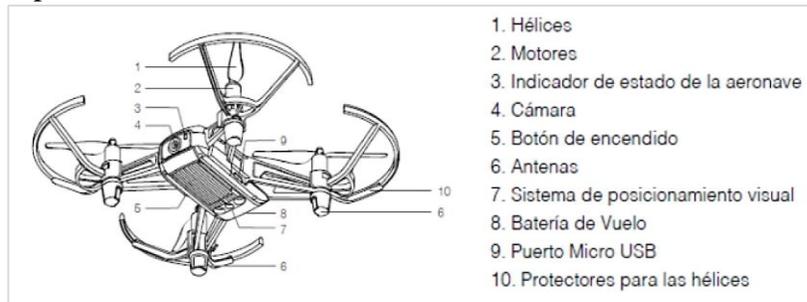


Figura 1. Componentes del Dron Tello Edu

A continuación, se describe el funcionamiento de los principales componentes del Dron mostrados en la figura 1.

- **Indicador de estado de la aeronave.** El indicador de estado de la aeronave de Tello comunica el estado de los sistemas de control de vuelo y las baterías de vuelo de la aeronave. El indicador de estado de la aeronave está ubicado en la parte delantera de la aeronave, al lado de la cámara, como se muestra en la figura 2 (Reymondin et al., 2019).

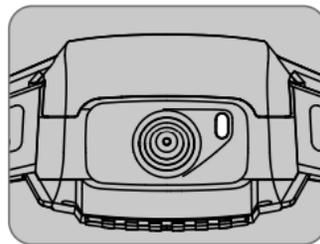


Figura 2. Indicador de estado de la aeronave.

- **Rotores y motores.** Los drones dependen de rotores y para el movimiento vertical. Los drones flotan usando sus rotores, que consisten en hélices conectadas a motores, lo que significa que el empuje hacia abajo del dron es igual a la fuerza de gravedad que actúa sobre él. Para ascender, el piloto aumenta la velocidad hasta que el rotor produce una fuerza ascendente mayor que la gravedad; para descender, el piloto invierte la marcha y reduce la velocidad. Para flotar, dos de los cuatro rotores del dron se mueven en el sentido de las agujas del reloj mientras que los otros dos se mueven en el sentido contrario, lo que garantiza que el impulso lateral del dron permanezca equilibrado. El mismo principio se aplica al movimiento hacia delante y hacia atrás: los rotores del dron deben producir empuje al tiempo que garantizan que la rotación de los rotores mantiene el equilibrio del dron (Reymondin et al., 2019).

- **Cámaras.** Las cámaras montadas en drones ayudan a los usuarios a ver lugares de difícil acceso y podrían ser una herramienta innovadora para los socorristas, especialmente en escenarios de búsqueda y rescate [16].
- **Sistema de posicionamiento visual.** El sistema de posicionamiento visual ayuda a la aeronave a mantener su posición actual. Con la ayuda de este sistema, Tello puede flotar con mayor precisión, así como volar en interiores o exteriores en condiciones de calma. Los componentes principales del sistema de posicionamiento visual son la cámara y el módulo infrarrojo 3D ubicado en la parte inferior de la aeronave (Reymondin et al., 2019).

Drone Tello Edu

Tello EDU es un pequeño cuadricóptero fabricado por Shenzhen Ruizhi Technology, equipado con un procesador Intel y un controlador de vuelo DJI integrado. Este drone es una versión mejorada del drone Tello que no permite el modo Swarm y tiene un SDK limitado. Gracias a la facilidad con la que se puede programar utilizando diferentes lenguajes de programación (Scratch, Python y Swift), esta aeronave es ideal para desarrollar infinidad de aplicaciones de forma educativa (Jaume Busquets, 2022).

Especificaciones técnicas: Dentro de las especificaciones técnicas en la tabla I se muestran las características del vehículo aéreo no tripulado (Reymondin et al., 2019).

Tabla 1. Especificaciones técnicas Drone Tello Edu.

Drone	Modelo	TLW004
	Peso	87 g
	Dimensiones	98x92,5x41 mm
	Velocidad máxima	28,8 km/h (8 m/s)
	Tiempo de vuelo máx.	13 minutos
	Distancia de vuelo máx.	100 m
	Altura de vuelo máx.	30 m
	Intervalo de temperatura de funcionamiento	De 0°C a 40°C
	Intervalo de frecuencias de funcionamiento	De 2,4 a 2,4835 GHz
	Transmisor (PIRE)	20 dBm (FCC) 19 dBm (CE) 19 dBm (SRRC)
Cámara	Tamaño imagen máximo	2592x1936
	Formato de imagen	JPG
	Modos de grabación de vídeo	HD: 1280x720 30p
	Formato de vídeo	MP4
Batería	Capacidad	1100 mAh
	Voltaje	3,8 V
	Tipo de batería	LiPo
	Energía	4,18 Wh
	Peso neto	25 ± 2 g
	Intervalo de temperatura de carga	de 5°C a 45°C
	Potencia de carga máx.	10 W

B. Procesamiento de imágenes con PYTHON

Python es un lenguaje de programación de alto nivel muy popular. Puede resolver una amplia variedad de tareas de programación, como computación numérica, desarrollo web, programación de bases de datos, programación de redes, procesamiento paralelo y más (Chityala & Pudipeddi, 2020). Python además es popular por varias razones, entre ellas:

- Es gratis.
- Está disponible en todos los sistemas operativos populares como Windows, Mac o Linux.
- Es un lenguaje interpretado. Por lo tanto, los programadores pueden probar partes del código en la línea de comandos antes de incorporarlo a su programa.
- No hay necesidad de compilar o vincular.
- Da la posibilidad de programar más rápido.
- Es sintácticamente más simple que C/C++/Fortran. Por lo tanto, es altamente legible y más fácil de depurar.
- Los programas escritos en Python se pueden importar a varios sistemas operativos o plataformas con poco o ningún cambio.
- Es un lenguaje tipado dinámicamente. Por lo tanto, el tipo de datos de las variables no tiene que declararse antes de su uso.
- Tiene una comunidad dedicada de desarrolladores y usuarios y se mantiene actualizada.

Principios básicos:

El procesamiento de imágenes es un método para realizar ciertas operaciones en una imagen para mejorar su calidad o extraer información útil de ella. Hoy en día, es una de las tecnologías de más rápido crecimiento en la actualidad. Además, forma parte de las áreas de investigación del Centro Disciplinario de Ingeniería e Informática (Mar et al., 2020). El procesamiento de imágenes incluye básicamente los siguientes pasos

- Importación de la imagen a través de herramientas de adquisición de imágenes.
- Analizar y manipular la imagen.
- Obtener una salida en la que se puede alterar el resultado de la imagen.

Por otro lado, el desarrollo de algoritmos de procesamiento ha cobrado impulso en el ámbito científico, especialmente en el campo de la inteligencia artificial para aplicaciones relacionadas con la visión artificial. Este conocimiento se nutre de la digitalización de datos en dispositivos que capturan video, fotos y más. La imagen se convierte entonces en el centro de la descripción de la realidad (Llulluna Llumiñana, 2014).

Anaconda:

Anaconda es una distribución de los lenguajes de programación Python y R para computación científica (ciencia de datos, aplicaciones de aprendizaje automático, procesamiento de datos a gran escala, análisis predictivo, etc.) Tiene como ventaja simplificar la gestión e implementación de paquetes. La distribución incluye paquetes de “data science” adecuados para Windows, Linux y macOS (Kadiyala & Kumar, 2018).

PhyCharm

PyCharm ofrece finalización de código inteligente, inspecciones de código, informes instantáneos de errores y soluciones rápidas, así como refactorización automática de código y navegación integral. PyCharm se parece a cualquier otro editor de código que haya usado en el pasado. Incluye un editor de código inteligente que reconoce fácilmente todos los componentes de un programa en ese idioma (Hu et al., 2018).

C. Técnicas para procesamiento de imágenes

La fase de procesamiento comienza con la secuencia de captura de imágenes y su preprocesamiento (Gutiérrez, 2017). Si el análisis se enfoca en el procesamiento secuencial de imágenes (como es el caso del video), las técnicas de captura de enfoque y nitidez serán más relevantes para los métodos de aprendizaje de IA (Kaur, 2016). A continuación, se presentan las principales técnicas para el procesamiento de imágenes.

Histogramas de gradientes: Es un descriptor de características utilizado en la visión artificial y el procesamiento de imágenes para detectar objetos. Esta técnica cuenta el número de ocurrencias de direcciones de degradado en partes locales de una imagen. El enfoque es similar a los histogramas de orientación de borde, los descriptores de transformación de características de escala invariable y los contextos de forma, pero se diferencia en que se calcula en una cuadrícula densa de celdas distribuidas uniformemente y utiliza la normalización de contraste local superpuesta para mejorar la precisión (Calero et al., 2015).

Algoritmo de detección LBPH: Además del método de rostro característico FisherFace, OpenCV también proporciona otro algoritmo clásico de reconocimiento de rostro LBPH. KBPH es la abreviatura de histogramas de patrones locales binarios, que se traduce como histogramas de codificación binaria locales. El algoritmo extrae las características de la imagen según el operador LBP. Si la imagen codificada por LBP se usa directamente para el reconocimiento facial. De hecho, no es muy diferente de no extraer características de LBP. Por lo tanto, en aplicaciones prácticas de LBP, los histogramas estadísticos de imágenes codificadas con LBP se usan generalmente como vectores de características para clasificación y reconocimiento (Gómez & Mayumi, 2016).

Algoritmo de detección EIGEN: En la visión por computadora, un conjunto de vectores de características utilizados para reconocer un rostro humano se denomina cara propia. El enfoque de caras propias fue propuesto por (Sirovich & Kirby, 1987) para el reconocimiento y desarrollado unos años más tarde por Matthew Turk y Alex Pentland (Turk & Pentland, 1991) para la clasificación de caras. El término "eigen" se refiere a un conjunto de vectores propios. La principal ventaja de este enfoque es que un conjunto de imágenes se puede representar utilizando una base compuesta de imágenes "intrínsecas", que son mucho más pequeñas en dimensionalidad que el conjunto de imágenes original. El vector de características proviene de la matriz de covarianza de la distribución de probabilidad en el espacio vectorial de alta dimensión de la imagen de la cara. Las propias caras propias forman el conjunto base de todas las imágenes utilizadas para construir la matriz de covarianza.

Algoritmo de detección FISHER: En general, la programación dinámica requiere optimizar un criterio numérico adicional W que depende de m pasos. Por lo general, el proceso de optimización bajo programación dinámica se ejecuta de principio a fin: primero, se planifica el último (m) paso (Gonzalez-Reyna et al., 2013).

Al planificar el último paso, es necesario hacer suposiciones sobre cómo termina el penúltimo paso (m-1) y tomar la mejor decisión en el último paso para cada situación diferente obtenida (Croux et al., 2005).

Machine Learning: El aprendizaje automático (ML) es un término general que se refiere a algoritmos que hacen predicciones inteligentes basadas en un conjunto de datos. Estos conjuntos de datos suelen ser grandes y pueden contener millones de puntos de datos únicos (Dietterich, 1990). Los avances recientes en el aprendizaje automático parecen haber alcanzado niveles humanos de comprensión semántica y extracción de información, a veces incluso siendo capaces de detectar patrones abstractos con mayor precisión que los expertos humanos. Ampliando las técnicas clásicas en el modelado estadístico, el aprendizaje automático moderno se ha convertido en una herramienta poderosa debido al gran aumento de los volúmenes de datos, el crecimiento exponencial en el poder computacional y los avances en el diseño de algoritmos, impulsados por las necesidades de las industrias web (Nichols et al., 2019).

Marco metodológico

Una vez identificada la metodología, los materiales y herramientas a continuación se procede a analizar el algoritmo de funcionamiento del sistema de detección y la conceptualización del algoritmo de entrenamiento.

A. Diagrama de flujo del funcionamiento del sistema

En la figura 3 muestra el algoritmo de funcionamiento del sistema de detección y reconocimiento facial. El algoritmo describe la siguiente secuencia de funcionamiento:

1. Inicio
2. Verifica los parámetros de entrada
3. Comprueba que el estado de la batería y el tiempo de vuelo sean los adecuados.
4. Si no cumple la etapa 3 el dron aterriza y finaliza el ciclo.
5. Si cumple la etapa 3 realiza los siguientes pasos: Adquisición de frame desde la cámara, procesa las imágenes, detecta e identifica el rostro finalmente, realiza un seguimiento del rostro y la ruta y repite el ciclo.

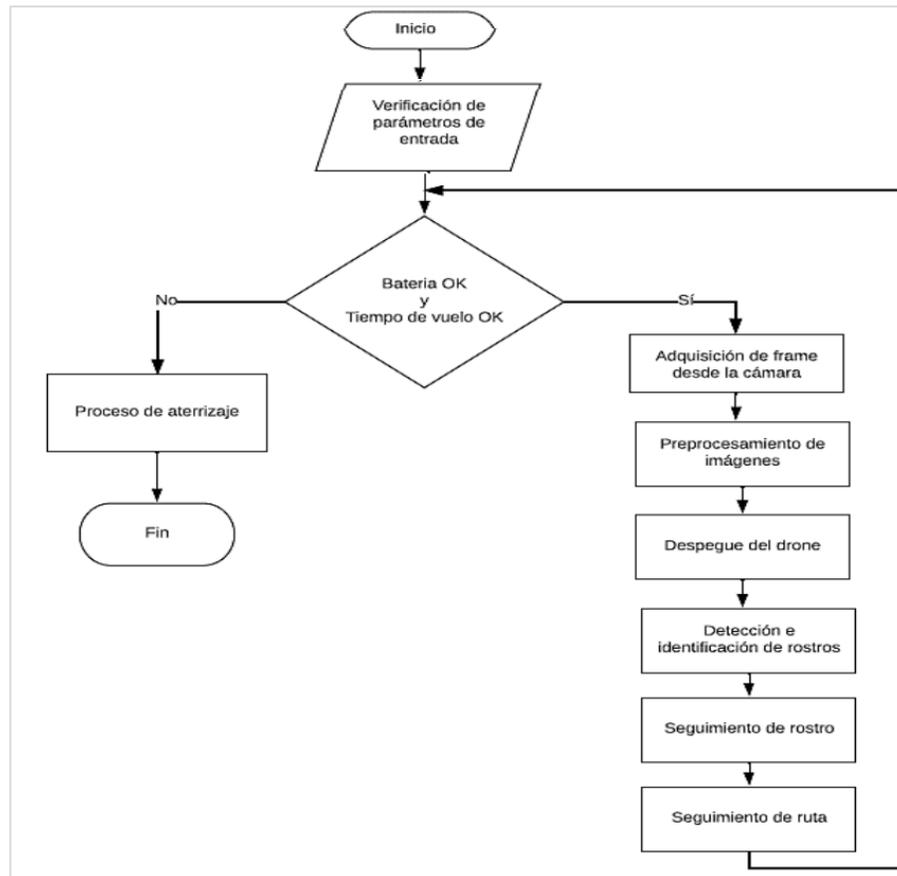


Figura 3. Diagrama de flujo del sistema de reconocimiento.

B. Conceptualización de la interfaz gráfica para el control del programa.

Dentro de la conceptualización se procedió con la creación de la interfaz gráfica para control del programa, este script usa principalmente la librería PySimpleGUI.

Además, dentro del código fuente del script para la detección se declararon todas las variables globales para el ingreso de parámetros y presentación de resultados, cuyas características se presentan a continuación:

- 1) **Administración de usuarios:** es la primera sección donde se permite el ingreso de usuarios usando la cámara de la PC para fines de pruebas y con la cámara del dron para la validación del sistema.
- 2) **Configuraciones:** permite seleccionar los modos de operación del dron. Se puede definir algunos modos de operación entre ellos manual y automático.
- 3) **Modelo de rostros:** permite actualizar el modelo machine learning para el reconocimiento de usuarios detectados por medio de la cámara. Cada vez que se desee agregar un nuevo usuario el modelo debe ser actualizado.
- 4) **Configuraciones de vuelo:** son los parámetros que se usarán para el control autónomo del dron en el espacio aéreo.

- 5) **Panel de resultados:** interfaz donde se muestran algunos parámetros relevantes del funcionamiento del sistema de detección.

C. Conceptualización de la interfaz de detección e identificación

Los métodos o directorios utilizados para la detección, identificación y control de imágenes se detallan a continuación:

- 1) **Método camera_idx:** permite la confirmación para la detección de la cámara en la red.
- 2) **Método update_dir:** permite actualizar los directorios de acuerdo con el Sistema operativo.
- 3) **Método frames_camera:** permite obtener cada uno de los frames desde la webcam o desde la cámara de un vehículo aéreo no tripulado según sea el caso.
- 4) **Método image_users:** permite detectar los rostros de personas durante el ingreso de nuevos usuarios. Estos rostros se guardarán en el directorio asignado previamente.
- 5) **Método model_faces:** permite entrenar el modelo de machine learning para clasificar los usuarios de acuerdo con los datos ingresados que finalmente se guardará en un archive .xml para leer el modelo durante la ejecución del control.
- 6) **Método read_model:** permite leer el modelo que se ha guardado previamente. Este proceso se realiza durante todas las ejecuciones de manera automática.
- 7) **Método distance_finder:** permite calcular la distancia aproximada en cm. desde la cámara hasta el rostro detectado.
- 8) **Método image_faces:** permite detectar rostros en cada uno de los frames durante la ejecución. Además, permite definir un objetivo para el seguimiento del rostro durante un tiempo específico.
- 9) **Método on_press, on_release:** permite leer las pulsaciones de las teclas del computador para ingreso de los comandos para el control manual del dron.
- 10) **Método control_manual:** permite activar el método de eventos para detectar las teclas de control.
- 11) **Método tracking_faces:** permite el seguimiento de un rostro detectado y cuando existiera más de un rostro en la imagen solo se enfocará el que más cerca de la cámara se encuentre.
- 12) **Método control_auto_dron:** permite generar tareas automáticamente para el dron, entre ellas, elevarse, moverse hacia adelante, control de altura, de batería, etc.
- 13) **Método finish_camera:** permite cerrar el proceso de adquisición de imágenes de manera segura.

D. Conceptualización de la interfaz de entrenamiento y predicción.

A nivel de programación esta etapa permite realizar el proceso de entrenamiento y predicción de rostros de acuerdo con los datos ingresados por usuario. El algoritmo de entrenamiento actualiza los valores de los pesos equitativamente.

Dentro del proceso de entrenamiento se permite la evaluación de los 3 modelos de machine learning para la detección de rostros cuyos valores de entrenamiento serán analizados en el capítulo de resultados. Dentro del entrenamiento de la red neuronal se considerarán los siguientes aspectos:

- 1) Durante cada iteración, el script presentará los resultados mediante la matriz de confusión y los tiempos de entrenamiento y predicción.

- En este script, se debe indicar la cantidad de muestras de imágenes que serán tomadas por cada usuario y el path donde se encuentran guardados los datos como se muestra en la figura 4.



Figura 4. Número de muestras y path de ubicación.

Para la ubicación del path es recomendable guardarlo dentro del mismo directorio del Proyecto y es fundamental considerar que la cantidad de imágenes o muestras por usuario debe ser igual para todos los casos registros, caso contrario no se puede entrenar el modelo de Machine Learning, debido a que surgiría un error de inconsistencia por la cantidad de elementos de la matriz de inicio.

E. Integración del código fuente en el software Pycharm

Una vez detallados el diagrama de flujo del sistema (A), los modelos de interfaz gráfica para control del programa (B), definición de los métodos para la detección e identificación del dron (C) y el modelo de entrenamiento y predicción de rostros de acuerdo con los datos ingresados por usuario (D), se procede a integrar dichos scripts en software PyCharm que se basa en el lenguaje de simulación Python como se muestra en la figura 5.

```

# Adjust size screen
# -----
Screen_size = 10
# -----
sg.theme('LightGray3')
path_r = os.getcwd()
idx = ['0', '1']

Layout1 = [[sg.Radio('Windows', 'RADIO1', enable_events=True, default=True, key='_SYS_'), sg.Text(' ', size=(5, 1)),
sg.Text('Path user: ', size=(10, 1)), sg.InputText(path_r, size=(27, 1), key='_DES_'), sg.FolderBrowse(...)],
[sg.Radio('Linux / Mac', 'RADIO1', enable_events=True, key='_LINUX_'), sg.Text(' ', size=(3, 1)),
sg.Text('User name: ', size=(10, 1)), sg.InputText(' ', size=(10, 1), key='_NAME_'),
sg.Text('Video input: ', size=(10, 1)), sg.Combo(values=idx, size=(5, 1), default_value='0', key='_VIN_'),
[sg.Text(' ', size=(16, 1)),
sg.Text('Total images: ', size=(10, 1)), sg.InputText('200', size=(10, 1), key='_TOTL_'),
sg.Checkbox('REPLACE IMAGES', default=True, key='_SIR_', font=('Arial', 9, 'bold'))],
[sg.Text(' ', size=(28, 1)), sg.Button('Save user', key='Save', size=(8, 1), button_color=('white', 'Blue')),
sg.Button('Cancel', size=(8, 1), button_color=('white', 'Red'))]]

Layout2 = [[sg.Radio('Indoor ', 'RADIO2', default=True, key='_IND_'), sg.Text(' ', size=(10, 1)),
sg.Radio('Manual ', 'RADIO3', default=True, key='_MAN_'), sg.Text(' ', size=(7, 1)),
sg.Radio('Dron camera', 'RADIO4', default=True, key='_DRO_'), sg.Text(' ', size=(3, 1))],
[sg.Radio('Outside', 'RADIO2', key='_END_'), sg.Text(' ', size=(9, 1)),
sg.Radio('Automatic', 'RADIO3', key='_AUT_'), sg.Text(' ', size=(6, 1)),
sg.Radio('Web camera', 'RADIO4', key='_WEB_'), sg.Text(' ', size=(3, 1))]]

Layout3 = [[sg.Text(' ', size=(4, 1)), sg.Text('Fly line: ', size=(7, 1)), sg.InputText('2', size=(5, 1), key='_TVU_'),
sg.Text('min.', size=(6, 1)), sg.Text(' ', size=(1, 1)),
sg.Text('Fly distance: ', size=(10, 1)), sg.InputText('2', size=(5, 1), key='_FDL_'),
sg.Text('m.', size=(6, 1)), sg.Text(' ', size=(1, 1))]]
    
```

Figura 5. Pseudocódigo de integración de módulos de entrenamiento en Pycharm.

Posterior se corrió la simulación a fin de comprobar el entrenamiento de la red para la detección de rostros, cuyos pasos generalizados se presentan a continuación:

- En primera instancia se debe seleccionar el tipo de cámara, sea esta de la PC o del dron, en este caso específico se utilizó la cámara del dron Tello Edu como se muestra en la figura 6.



Figura 6. Drone Tello Edu.

- Para ingresar un nuevo usuario se define el nombre y el número de muestras a tomar (la misma debe coincidir con todos los usuarios previamente registrados) como se muestra en la figura 7.



Figura 7. Registro de usuario para el entrenamiento del algoritmo.

- Se procede a enfocar la cámara hacia el usuario para la toma de muestras y se guarda el modelo.
- Posterior se ejecuta el Guide principal, se enfoca el rostro hacia la cámara, empieza el entrenamiento y reconocimiento del usuario mostrando el nombre del usuario y la distancia respectiva, cuyas pruebas se muestran en la figura 8 y figura 9.

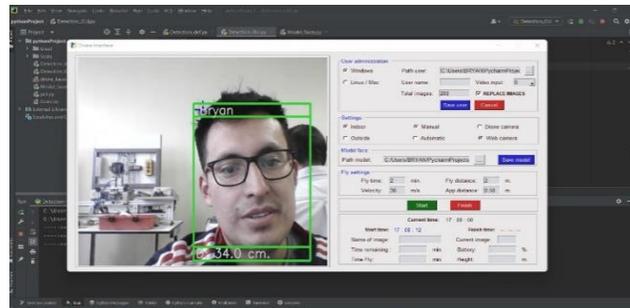


Figura 8. Entrenamiento del algoritmo para el reconocimiento del usuario 1.

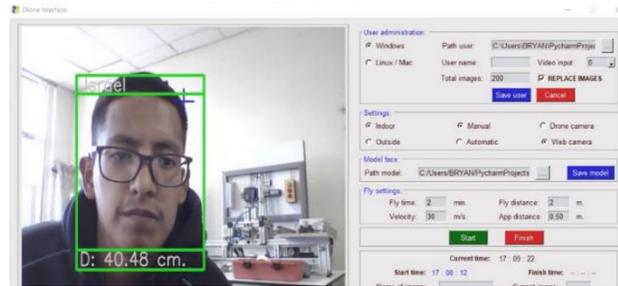


Figura 9. Entrenamiento del algoritmo para el reconocimiento del usuario 2.

Resultados y discusión

Los datos se recuperan a través de la interfaz desarrollada para el control del dron Tello. El conjunto de datos consta de 5 clases que representan a las personas que el sistema puede reconocer. Cada clase se compone de 200 imágenes que proporcionan la información necesaria para entrenar y evaluar los algoritmos de reconocimiento facial.

El conjunto de datos ha sido dividido en el 80% para datos de entrenamiento y 20% para datos de evaluación del modelo. Este detalle se muestra en la tabla 2:

Tabla 2. Tabla especificaciones técnicas Dron Tello Edu.

Conjunto de datos	Porcentaje	Cantidad de datos
Total	100%	1000
Entrenamiento	80%	800
Evaluación	20%	200

Para la creación del modelo de Machine Learning se requirió de un modelo, un proceso de entrenamiento y finalmente un proceso de evaluación. Los algoritmos evaluados fueron el Algoritmo de detección LBPH, Algoritmo de detección EIGEN, Algoritmo de detección FISHER. Para la evaluación se utilizó la matriz de confusión LBPH, matriz de confusión EIGEN y la matriz de confusión FISHER, dichas matrices se muestran en la figura 10 y 11 respectivamente.

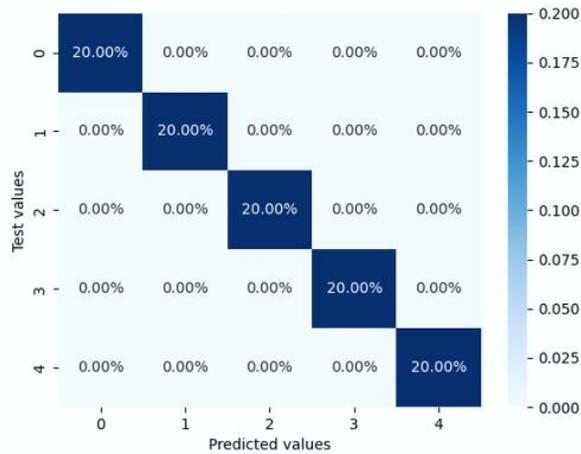


Figura 10. Entrenamiento del algoritmo LBPH para el reconocimiento de usuario.

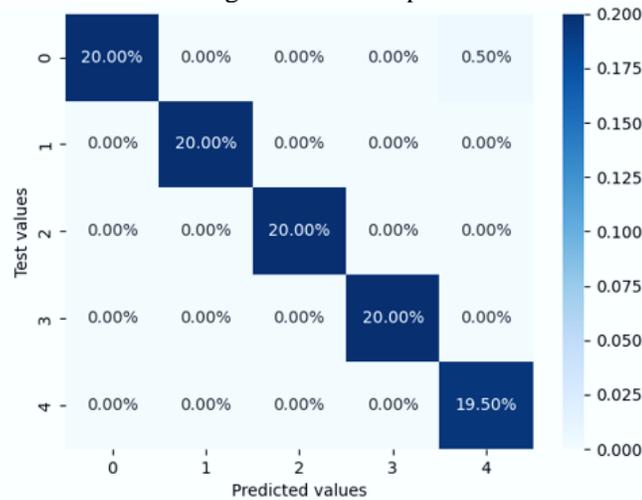


Figura 11. Entrenamiento del algoritmo EIGEN para el reconocimiento de usuario.



Figura 12. Entrenamiento del algoritmo FISHER para el reconocimiento de usuario.

Como se evidencia el entrenamiento del algoritmo presenta resultados favorables y eficientes en la detección de los usuarios puestos a prueba, en ese sentido el algoritmo del modelo EIGEN presenta un margen de error mayor al de los modelos FISHER y LBPH, siendo este último el que mejores tiempos de respuesta proporciona al ser puesto a prueba.

Conclusiones

El algoritmo de entrenamiento permitió la detección y reconocimiento de usuarios cuyo registro de imágenes es tomado a través de la cámara del vehículo aéreo no tripulado o por medio de la cámara de un ordenador donde dichos datos son procesados y cotejados a través del algoritmo de entrenamiento para el aprendizaje autónomo por medio de redes neuronales.

El entrenamiento de redes neuronales enfocado en entornos de detección y reconocimiento de imágenes se basa en el desarrollo de un algoritmo de control y modelado de las interfaces de comunicación para el proceso de identificación. Dentro del desarrollo se analizó los algoritmos de detección LBPH, EIGEN y FISHER y el método de entrenamiento model.train fue definido como identificador para la lectura de datos.

Dentro de las pruebas realizadas el modelo EIGEN presenta un rango de identificación cercano a la media de -2.2. Por otra parte, el modelo LBPH y FISHER presentan un rango de error con una media cercana a cero, considerando que este último requiere de un mayor tiempo de entrenamiento, donde el modelo LBPH resulta el algoritmo más idóneo para el entrenamiento de la red neuronal debido a su baja tasa error y su corto tiempo de entrenamiento.

Dentro de la evaluación del modelo predictivo se identificó la tasa de precisión del modelo de identificación y comparación por medio de la matriz de confusión model.predict de los algoritmos LBPH, EIGEN. Según los resultados el algoritmo EIGEN presenta un porcentaje de error de predicción del 19.50% al clasificar la clase 4. El algoritmo FISHER presenta errores de predicción en la clasificación de las clases 0, 1, 3, 4 donde no alcanza la clasificación correcta cuyo resultado se refleja en un valor de 20% según la cantidad de casos.

El modelo LBPH alcanza un porcentaje de clasificación correcta en el 100% de los casos donde se valida el funcionamiento de identificación y clasificación de valores predictivos para el caso de estudio.

Recomendaciones

Dentro de la evaluación del modelo predictivo al crear path del usuario es recomendable guardarlo dentro del mismo directorio del proyecto para la correcta validación de datos, además es importante considerar que la cantidad de imágenes o muestras por usuario debe ser igual para todos los casos de registro, caso contrario no se procederá con el entrenamiento del modelo de Machine Learning, debido a que surgiría un error de inconsistencia por la cantidad de elementos de la matriz de datos.

Dentro de la conceptualización de la interfaz de control del programa es importante identificar con claridad las variables globales y parámetros de funcionamiento para el correcto proceso de reconocimiento e identificación de usuarios. Para el desarrollo de la interfaz de detección y control del dron es importante identificar los métodos o directorios utilizados por el prototipo para su uso dentro del entorno de desarrollo software a fin de obtener resultados eficientes y optimizar los procesos.

Para la validación de resultados es fundamental considerar el número de muestras, número de datos de predicción, el tiempo de entrenamiento y el tiempo de predicción como factores relevantes del modelo, a fin de comprobar el tiempo y eficiencia de respuesta de cada algoritmo dependiendo el número de muestras.

Referencias

- Calatayud Ferre, R. (2022). Detección de personas para rescate marítimo. https://rua.ua.es/dspace/bitstream/10045/124653/1/Deteccion_de_personas_para_rescate_maritimo_Calatayud_Ferre_Raul.pdf
- Calero, M. J. F., Jácome, D. J. R., & Escobar, D. A. S. (2015). Histograma del gradiente con múltiples orientaciones (hog-mo) detección de personas. *Revista vínculos*, 12(2), 138-147. <http://revistas.udistrital.edu.co/ojs/index.php/vinculos/article/download/10991/11830>
- Croux, C., Filzmoser, P., & Joossens, K. (2005). Robust linear discriminant analysis for multiple groups: influence and classification efficiencies. Available at SSRN 876896. <https://lirias.kuleuven.be/retrieve/18130>
- Custers, B. (2016). *Future of Drone use*. Springer. <https://www.asser.nl/upload/documents/20161025T135125-Frontmatter%20Custers.pdf>
- Chamayou, G. (2016). *Teoría del dron: Nuevos paradigmas de los conflictos del siglo XXI*. Ned ediciones. https://www.google.com/books?hl=es&lr=&id=NOVSDAAAQBAJ&oi=fnd&pg=PT5&dq=G.+Chamayou,+Teor%C3%ADa+del+dron:+nuevos+paradigmas+de+los+conflictos+del+siglo+XXI.+Ned+ediciones,+2016.+Accedido:+6+de+noviembre+de+2022.+%5BEn+1%C3%ADnea%5D.+Disponible+en:+https://elibro.net/es/lc/epoch/titulos/59924&ots=8sLu8jlEUY&sig=XcRMWPmx8ZVkoZjmsv_O5oKR010
- Chityala, R., & Pudipeddi, S. (2020). *Image processing and acquisition using Python*. Chapman and Hall/CRC. <https://www.taylorfrancis.com/books/mono/10.1201/9780429243370/image-processing-acquisition-using-python-ravishankar-chityala-sridevi-pudipeddi>

- Dietterich, T. G. (1990). Machine learning. *Annual review of computer science*, 4(1), 255-306. <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=417f55320080fea71d491a18318dad6572b2fa8c>
- Espinal, E. K. (2018). HISTOGRAMA CIRCULAR 3D. *Pesquimat*, 15(2). <https://revistasinvestigacion.unmsm.edu.pe/index.php/matema/article/download/9610/8423>
- Gómez, T., & Mayumi, C. (2016). Análisis comparativo de los algoritmos Fisherfaces y LBPH para el reconocimiento facial en diferentes condiciones de iluminación y pose, Tacna–2015. https://alicia.concytec.gob.pe/vufind/Record/UNJB_109808fa5dc207c12655d88db63e454b/Description
- Gonzalez-Reyna, S. E., Avina-Cervantes, J. G., Ledesma-Orozco, S. E., & Cruz-Aceves, I. (2013). Eigen-gradients for traffic sign recognition. *Mathematical Problems in Engineering*, 2013. <https://www.hindawi.com/journals/mpe/2013/364305/>
- Gutiérrez, J. A. T. (2017). El procesamiento de imágenes y su potencial aplicación en empresas con estrategia digital. *Interfases*(10), 11-29. <https://dialnet.unirioja.es/descarga/articulo/6230450.pdf>
- Hayat, S., Yanmaz, E., & Muzaffar, R. (2016). Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint. *IEEE Communications Surveys & Tutorials*, 18(4), 2624-2661. <http://mobile.aau.at/publications/hayat-2016-COMST-surveyUAV.pdf>
- Hu, Q., Ma, L., & Zhao, J. (2018). DeepGraph: A PyCharm tool for visualizing and understanding deep learning models. 2018 25th Asia-Pacific Software Engineering Conference (APSEC),
- Jaume Busquets, J. (2022). *Programación de un drone Tello EDU* Universitat Politècnica de Catalunya]. <https://upcommons.upc.edu/bitstream/handle/2117/362149/memoria.pdf?sequence=1>
- Jones, T. (2017). *International commercial drone regulation and drone delivery services*. <https://picture.iczhiku.com/resource/paper/whkfUTaAzZGRFNxX.pdf>
- Kadiyala, A., & Kumar, A. (2018). Applications of python to evaluate the performance of bagging methods. *Environmental Progress & Sustainable Energy*, 37(5), 1555-1559. <https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/ep.13018>
- Kaur, S. (2016). An automatic number plate recognition system under image processing. *International Journal of Intelligent Systems and Applications*, 8(3), 14. <http://www.mecspress.net/ijisa/ijisa-v8-n3/IJISA-V8-N3-2.pdf>
- Lee, W., Alkouz, B., Shahzaad, B., & Bouguettaya, A. (2021). Package delivery using autonomous drones in skyways. Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers,
- Llulluna Llumiquinga, F. R. (2014). *Procesamiento de imágenes mediante software libre python para el análisis metalográfico en aceros de bajo contenido de carbono* QUITO, 2014.]. <http://bibdigital.epn.edu.ec/bitstream/15000/7171/1/CD-5361.pdf>

- Maalouf, A., Gurfinkel, Y., Diker, B., Gal, O., Rus, D., & Feldman, D. (2022). Deep Learning on Home Drone: Searching for the Optimal Architecture. *arXiv preprint arXiv:2209.11064*. <https://arxiv.org/pdf/2209.11064>
- Mar Cornelio, O. (2019). Modelo para la toma de decisiones sobre el control de acceso a las prácticas de laboratorios de Ingeniería de Control II en un sistema de laboratorios remoto.
- Mar, O., Gulín, J., Ivan Santana Ching, & Fonseca, B. B. (2020). Remote Laboratory System for Automatic Engineering. *International Journal of Wireless and Ad Hoc Communication*, 1(2), 55-63.
- Morcillo Martínez, L. (2018). *Sistema de detección de obstáculos para drones basado en sensor láser* [Universitat Politècnica de València]. <https://riunet.upv.es/bitstream/handle/10251/105633/MORCILLO%20-%20SISTEMA%20DE%20D ETECCI%3%93N%20DE%20OBST%3%81CULOS%20PARA%20DRONES%20BASADO%2 0EN%20SENSOR%20L%3%81SER.pdf?sequence=4>
- Nichols, J. A., Herbert Chan, H. W., & Baker, M. A. (2019). Machine learning: applications of artificial intelligence to imaging and diagnosis. *Biophysical reviews*, 11(1), 111-118. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6381354/>
- Reymondin, L., Paz-García, P., Tello, J. J., Coca-Castro, A., Bautista, O., & Menzinger, B. (2019). User Manual. Training Manual Terra-i. *CIAT Publication*. <https://cgspace.cgiar.org/handle/10568/102219>
- Sabry, F. (2021). *Robótica Autónoma: ¿Cómo aparecerá un robot autónomo en la portada de la revista Time?* (Vol. 3). One Billion Knowledgeable. https://www.google.com/books?hl=es&lr=&id=uP6TEAAQBAJ&oi=fnd&pg=PT654&dq=Detecc i%3%B3n+y+reconocimiento+de+objetos+de+veh%3%ADculos+a%3%A9reos+no+tripulados +mediante+la+plataforma+Raspberry+Pi%22,+Quinto+Simposio+Internacional+sobre+Estudios+M ultidisciplinarios+y+Tecnolog%3%ADas+Innovadoras&ots=V5Qr9bA1Ro&sig=w0djXbQHBU8 Qrj3n8oVOEjPc_28
- Saini, T. (2021). *Manoeuvring drone (Tello ans Tello EDU) using body poses or gestures* [Universitat Politècnica de Catalunya].
- Sirovich, L., & Kirby, M. (1987). Low-dimensional procedure for the characterization of human faces. *Josa a*, 4(3), 519-524. <https://opg.optica.org/abstract.cfm?uri=josaa-4-3-519>
- Subash, K., Srinu, M. V., Siddhartha, M., Harsha, N. S., & Akkala, P. (2020). Object detection using Ryze Tello drone with help of mask-RCNN. 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA),
- Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1), 71-86. <https://direct.mit.edu/jocn/article-pdf/3/1/71/1932018/jocn.1991.3.1.71.pdf>