

Implementación de un sistema de comunicación entre un *drone* y una computadora en tierra para el seguimiento de objetos en tiempo real

Agustín A. Giovanazzi, Sebastián Verrastro

Universidad Tecnológica Nacional, Facultad Regional Buenos Aires, Departamento de Ingeniería Electrónica y Grupo de Inteligencia Artificial y Robótica (GIAR), Medrano 951, (C1179 AAQ), Ciudad Autónoma de Buenos Aires, Argentina

sebasverra@gmail.com

Recibido el 5 de diciembre de 2023, aprobado el 15 de febrero de 2024

Resumen

Los vehículos no tripulados o *drones* controlados por inteligencia artificial cuentan con muchas ventajas en el ámbito de la seguridad y seguimiento inteligente de personas u objetos. Aun así, sus competencias se ven limitadas debido a su reducida capacidad de procesamiento de datos. Por este motivo, se propone incorporar una estructura de comunicación entre un sensor RGB-D que se encuentra a bordo del *drone* y la computadora en tierra capaz de ejecutar los algoritmos de inteligencia artificial en tiempo real para poder entregar los resultados al sistema de vuelo y hacer posible un seguimiento e identificación de un sujeto en tiempo real. Es por esto que se cuenta con un algoritmo de codificación y transmisión de datos de baja latencia junto con ROS2 para comunicar, tanto al *drone* con su estación en tierra como, el procesamiento de datos.

PALABRAS CLAVE: DRONE - SEGUIMIENTO AUTÓNOMO - RASTREO - TRACKEO - RASTREO E IDENTIFICACIÓN DE PERSONAS - PERSECUSIÓN - CÁMARA KINECT

Abstract

Unmanned vehicles or drones controlled by artificial intelligence have many advantages in the field of security and intelligent monitoring of people or objects. Even so, its powers are limited due to its reduced data processing capacity. For this reason, it is proposed to incorporate a communication structure between an RGB-D sensor that is on board the drone and the ground computer capable of executing the artificial intelligence algorithms in real time in order to deliver the results to the flight system and make it possible to track and identify a subject in real time. This is why there is a low latency data transmission and coding algorithm together with ROS2 to communicate both the drone with its ground station and the data processing.

KEYWORDS: TRACKING - DRONE - DRONE COMMUNICATION SYSTEM - DRONE CONTROLLED BY AI - DRONE TRACKING - ROS2

Introducción

En la actualidad, los vehículos no tripulados (UAVs) o *drones* dependen de un piloto que dirige su movimiento o de un programa preconfigurado mediante el cual se le asigne la misión a cumplir (Dardoize, *et al.*, 2019). Sin embargo, cuando es necesario el seguimiento de un objeto o persona, esta tarea se vuelve muy complicada para el piloto que debe mantener una distancia adecuada, evitar obstáculos, entre otras cosas. Si bien existen dispositivos que permiten que los *drones* sean autónomos en sus movimientos y desplazamientos, estos deben posicionarse con anterioridad en la zona donde el *drone* se desplazará (Pavlenko *et al.*, 2019). Así mismo, existen vehículos no tripulados que incorporan algoritmos de inteligencia artificial para identificar y seguir objetos. A pesar de ello, estos algoritmos no son lo suficientemente eficientes para cumplir el objetivo y poseen un rendimiento insuficiente para desempeñar un control autónomo en tiempo real (Smyczyński, *et al.*, 2017), (Amato, *et al.*, 2019). Este problema surge debido a la poca capacidad de procesamiento con la que cuenta el *drone*. Es por ello que una alternativa es ceder dichos cálculos a una computadora externa que pueda mantener un elevado rendimiento en el procesamiento de los datos y entregar una respuesta en tiempo real con un reducido nivel de error.

El objetivo de este trabajo es desarrollar un entorno simulado dentro del cual se pueda disponer de un vehículo no tripulado capaz de adquirir la información necesaria para realizar el seguimiento en tiempo real de una persona mediante una comunicación entre el *drone* simulado y una computadora capaz de ejecutar los algoritmos de inteligencia artificial.

Metodología

Para lograr nuestro objetivo, el *drone* debe contar con una conexión inalámbrica capaz de transmitir los datos necesarios en tiempo real. Esta dificultad se solventa gracias a la utilización de un algoritmo de transmisión de imágenes (Promonet, 2023) junto a la codificación de las mismas en el formato que permita una transmisión eficiente (Jian-Wen Chen, *et al.*, 2006).

A su vez, para administrar el vuelo del *drone* se utiliza un software de control de vuelo llamado PX4 *autopilot* (Dronecode Foundation, 2021a), el cual se encarga de controlar sus movimientos, supervisar su estado y mantener una comunicación estable con el control de tierra (GCS). Para llevar a cabo la comunicación entre el control de tierra y el vehículo no tripulado, se utiliza un sistema de comunicación llamado Mavlink (Dronecode Foundation, 2021b) el cual, junto a la librería Mavros (Ermakov, 2021), permiten el intercambio de información entre el sistema de vuelo y el sistema de control en tierra llamado *QgroundControl* (Dronecode Foundation, 2021c). Además, Mavros se encarga de traducir la información a nodos de ROS2 (*Robot Operating System*) (Macenski, *et al.*, 2022) lo cual agiliza y mejora la manera en la cual se administra y distribuye la información.

Los algoritmos de inteligencia artificial sobre el seguimiento de objetos o personas (YOLO y DEEPSORT) (Redmon, *et al.*, 2022), (Khokhar, 2023) junto con los datos de sensores de profundidad juegan un papel fundamental. Mediante el uso de un sensor de imagen y profundidad (sensor RGB-D), los algoritmos de la inteligencia artificial junto con la información de profundidad ubican e identifican al objeto o persona que se está siguiendo. Para nuestro sistema, el sensor encargado de recolectar la información de imagen y profundidad será la cámara Kinect (Zhang, 2012).

YOLO+DEEPSORT

YOLO (*You Only Look Once*)

El algoritmo de YOLO (*You Only Look Once*) es un algoritmo de inteligencia artificial el cual cuenta con algunas características que lo convierten en la opción más acertada

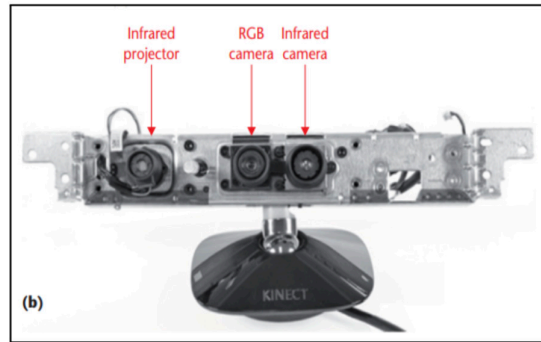


Fig. 1. Kinect. La cámara RGB junto con su sistema de detección de profundidad. Se proyecta un láser infrarrojo que rebota sobre una superficie y vuelve para ser detectado por la cámara infrarroja.

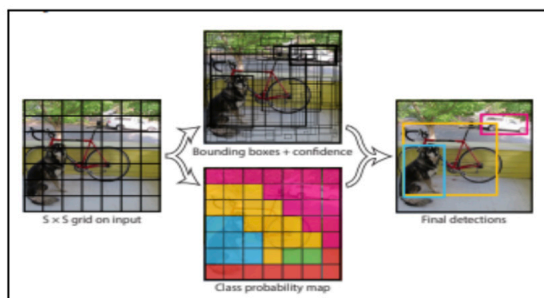


Fig. 2. Modelo. El sistema de YOLO para detectar objetos dividiendo la imagen en matrices de tamaño SxS sobre las cuales se realiza una predicción para determinar si es posible que dicha matriz contenga un objeto.

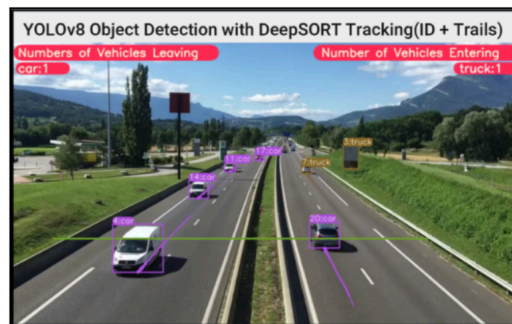


Fig. 3. Tracking. La inteligencia artificial de trackeo detectando y enumerando los objetos detectados por YOLOv8.

para lograr el objetivo buscado gracias a ser uno de los más rápidos que existen hoy en día y contar con un excelente nivel de precisión (Mahendrakar *et al.*, 2022). Estos dos aspectos son fundamentales a la hora de mantener un seguimiento en tiempo real de un objeto o persona debido a que se minimiza el tiempo que tarda el *drone* en reaccionar a los eventos que pueden suceder durante el seguimiento. Este sistema parte de una imagen proporcionada por una cámara, que se divide en una cantidad determinada de cuadros (SxS cuadros) (Girshick, *et al.*, 2014). Estos cuadros son analizados por separado para determinar, por un lado, si dentro del mismo se encuentra un objeto o persona asignándole un valor que representa la probabilidad y, por otro lado, a qué clase pertenece dicho objeto. Finalmente, se le aplica un pos-procesado para eliminar múltiples detecciones sobre el mismo objeto y recalcular la probabilidad.

DEEPSORT

Para lograr el seguimiento de un objeto o persona es necesaria la capacidad de enumerar al objeto o persona sobre la cual se lleva a cabo el seguimiento (Rakotoniaina, *et al.*, 2023).

Este algoritmo utiliza a YOLO para identificar objetos o personas y los enumera siendo capaz de diferenciar a un objeto de otro con apariencia similar, así como a un sujeto de otro. Esto lo logra gracias a que mantiene un registro de las detecciones previas de estos objetos las cuales compara para determinar si se trata del mismo objeto o de uno nuevo. Asimismo, permite identificar a la persona aun si esta salió del rango de visión de los sensores o se interpuso algún objeto en la detección.

Sistema de control de vuelo PX4

Para este proyecto se utilizó PX4 siendo el piloto automático profesional (Meier, *et al.*, 2015). Para cualquier vehículo no tripulado es esencial que posea la capacidad de controlar su posición y movimiento siendo capaz de trasladarse por medio de algoritmos de control, los cuales son usualmente ejecutados por un microcontrolador.

PX4 integra ROS2 (*Robot Operating System*). Permitiendo que los datos de los sensores y el control de vuelo puedan comunicarse con los demás procesos que se están ejecutando dentro del *drone*, de una manera más eficiente.

ROS2 (Robot Operating System)

ROS2 es un conjunto de librerías y herramientas de código abierto diseñadas para el armado de robots (Berger, *et al.*, 2009). Este nos permite comunicar distintos procesos de manera eficiente y sencilla. Para lograr esto, cada programa se comunica por medio de nodos. Los programas pueden asignarse a cada nodo, principalmente de dos maneras, como publicador o como suscriptor. De esta manera, el sistema de control de vuelo y los algoritmos de inteligencia artificial pueden comunicarse eficientemente. (Zhao, *et al.*, 2015)

Cada programa puede ser asignado como publicador o suscriptor de un tópico (también pueden ser publicador y suscriptor de uno o varios tópicos). Esto permite que pueda recibir información de uno o varios tópicos, suscribiéndose a los mismo, procesar la información y, luego, publicarla en otro tópico para que cualquier otro programa que la necesite pueda recurrir a ella.

Comunicación entre los algoritmos de inteligencia artificial y el *drone*

Dado que el *drone* no cuenta con la potencia computacional para ejecutar los algoritmos con suficiente eficacia, esta tarea la debemos reservar para un dispositivo externo. Es por esto que una computadora en tierra se encargará de realizar los cálculos necesarios y enviar por medio de una comunicación vía internet los resultados de los algoritmos. Para ello, un algoritmo a bordo del *drone* se encargará de recibir la información y distribuirla por medio de los nodos de ROS2. Aun así, para lograr el objetivo propuesto es necesaria la información de profundidad proporcionada por una cámara RGB-D, siendo en este caso la Kinect (Di Filippo, *et al.*, 2015). Este sensor es capaz de, entre otras cosas, otorgar dos imágenes de la misma escena, pero con distinta información. Por un lado, otorga la imagen a color (RGB) y, por otro lado, proporciona una matriz con los valores de profundidad. Todo esto ocurre gracias a la utilización de un driver que se comunica con la cámara y extrae la información anteriormente mencionada. Finalmente, la imagen extraída de la cámara es enviada a los algoritmos de inteligencia artificial por medio de una conexión vía internet.

Esta comunicación se logra gracias a la utilización de un algoritmo, llamado v4l2rtspserver, que se encarga de tomar la imagen de una cámara y enviarla por medio de un protocolo de comunicación llamado RSTP (*Rapid Spanning Tree Protocol*) (Pallos, *et al.*, 2016). Sin embargo, para mejorar el rendimiento de la transmisión de video, se utiliza la librería v4l2tools (Promonet, 2023) junto al módulo v4l2loopback (Umlaeute, 2023). Estas se encargan de crear cámaras virtuales y codificar la imagen de la cámara a un formato más eficiente para

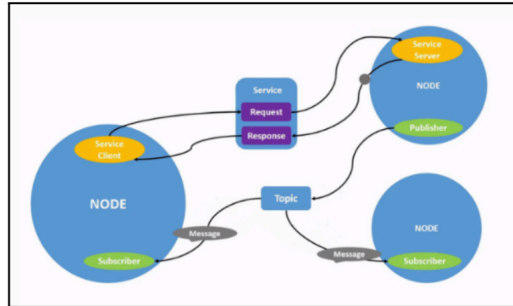


Fig. 4. Comunicación con ROS2. Se puede observar el funcionamiento de los nodos y los servidores de ROS2 permitiendo conectar varios procesos entre sí.

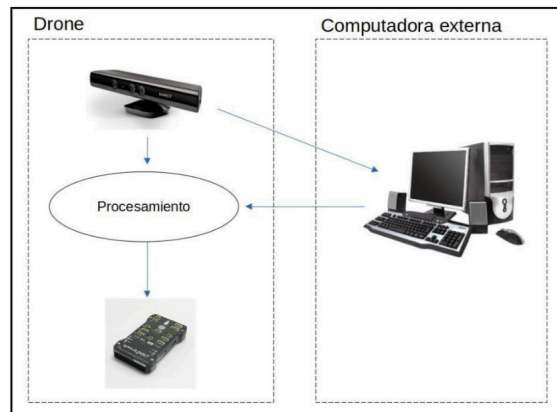


Fig. 5. Diagrama explicativo. Se presenta un diagrama simplificado de la comunicación que se pretende lograr entre los dispositivos del drone y la computadora en tierra.

su transmisión. La imagen se codifica en formato H.264, el cual permite una transmisión más eficiente (Yoon, *et al.*, 2012).

Dado que el video transmitido se encuentra codificado, es necesario decodificarlo. De este paso se encarga el software FFMPEG (FFMPEG Team, 2023) el cual, decodifica la imagen recibida y la coloca en una cámara virtual generada por la librería antes mencionada. Finalmente, el algoritmo de inteligencia artificial (DEEPSORT) analiza la imagen y otorga una respuesta por medio de una comunicación vía internet con el *drone*.

De esta manera, el *drone* procesa la información y le indica al sistema de vuelo (PX4), por medio de los nodos de ROS2, que movimiento es necesario realizar para mantener el seguimiento del objetivo.

Simulación

Para llevar a cabo la simulación se hará uso del simulador Gazebo (OPEN ROBOTICS, 2023) dentro del cual, se dispondrá del *drone* junto con la cámara Kinect montada sobre el mismo y algunos modelos de personas que cumplirán la función de sujetos de prueba para desempeñar el seguimiento.

En lo que respecta al procesamiento de información por parte de los algoritmos de inteligencia artificial, se dispondrá de una computadora que cuenta con una placa de video Nvidia GTX 1060.

Por su parte, las imágenes de la cámara son extraídas y transmitidas en una resolución de 640x480 a una velocidad de 15 fotogramas por segundo.

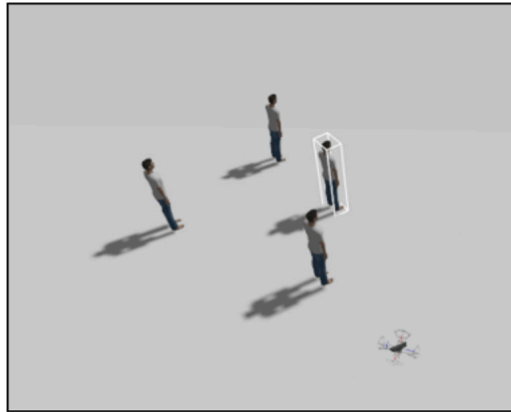


Fig. 6. Escenario. Se dispone de varios modelos de personas junto con el *drone* realizando el seguimiento de uno de ellos.

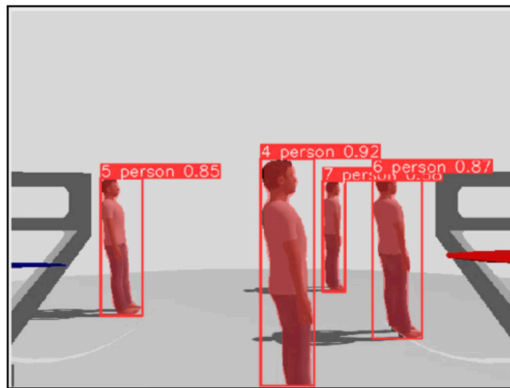


Fig. 7. Detecciones. Se observa el resultado de las detecciones de los modelos de inteligencia artificial a partir de los datos recolectados por la cámara a bordo del *drone*.



Fig. 8. Seguimiento. Se muestra el funcionamiento del *drone* el cual, se eleva para acomodarse a una distancia de un metro de su objetivo.

Los datos de desplazamiento del *drone* son recibidos en coordenadas cilíndricas (r , θ , z), siendo la distancia obtenida del sensor de profundidad de la cámara, el valor de la variable r . Por su parte, para la obtención de los valores de tita (θ), se optó por desarrollar un cálculo basándose en los datos del campo de visión de la cámara. De esta manera, el *drone* recibe los datos procesados por la inteligencia artificial a través de la conexión inalámbrica y, junto con los datos extraídos del sensor de la cámara, se transforman los datos de desplazamiento de coordenadas cilíndricas a coordenadas cartesianas ya que, con dichas coordenadas es con las que trabaja el control de vuelo PX4.

Resultados

Como resultado de la simulación, se logró una comunicación estable entre el sistema de vuelo y procesamiento de datos y los algoritmos de inteligencia artificial. Estos últimos alcanzaron un rendimiento que varió entre 60 m y 120 m por procesamiento de imagen dependiendo de la cantidad de sujetos detectados. Por su parte, el sistema de comunicación no presentó problemas siendo capaz de transmitir hasta 15 fotogramas por segundo sin presentar un retraso en la comunicación y decodificación que pudiera generar un problema a la hora de controlar los movimientos del *drone* en tiempo real.

En lo que respecta al *drone*, este logró situarse a una distancia preestablecida de tres metros de distancia delante del sujeto de pruebas.

Cabe destacar que el *drone*, a la hora de desplazarse, puede causar que el sujeto desaparezca de la imagen tomada por la cámara a bordo. Esto debido a la inclinación del propio *drone* mientras lo hace. Sin embargo, este problema fue fácilmente solucionado gracias a la propia inteligencia artificial de seguimiento (DEEPSORT), que demostró ser capaz de reconocer al mismo sujeto, aun si este desaparece de la imagen.

Conclusiones

Se introdujo un sistema de comunicación que es capaz de comunicarse en tiempo real con algoritmos de inteligencia artificial y controlar al *drone* en base a los datos recibidos por los mismos algoritmos.

Si bien el sistema cuenta con la utilización de los algoritmos YOLO y DEEPSORT, este admite cualquier otro algoritmo de seguimiento e identificación de objetos que proporcione la información necesaria. De esta manera, es posible aumentar el rendimiento del sistema mediante algoritmos más rápidos y precisos. Asimismo, el sistema permite la utilización de cualquier sensor RGB-D y no se limita a la precisión y velocidad de la cámara Kinect utilizada.

Es por ello que el sistema desarrollado cuenta con un elevado rendimiento y compatibilidad que lo transforman en una de las mejores opciones a la hora de realizar el seguimiento de un objeto.

Referencias

AMATO, G.; CIAMPI, L.; FALCHI, F. y GENNARO, C., (2019). "Counting Vehicles with Deep Learning in Onboard UAV Imagery," 2019 IEEE Symposium on Computers and Communications (ISCC), Barcelona, Spain, 2019, pp. 1-6, doi: 10.1109/ISCC47284.2019.8969620.

BERGER, E.; WHEELER, R. y NG, A.; (2009). ROS: An open-source Robot Operating System. ICRA Workshop on Open Source Software, 3, 1-6

CHEN, J-W.; KAO, C-Y. y LIN, Y- L., (2006). "Introduction to H.264 advanced video coding," Asia and South Pacific Conference on Design Automation, 2006., Yokohama, Japan, pp. 6 pp.-, doi: 10.1109/ASPAC.2006.1594774.

DARDOIZE, N.; CIOCHETTO, J.; H. HONG, H y SHIN, H. S., (2019). "Implementation of Ground Control System for Autonomous Multi-agents using QGroundControl," 2019 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED UAS), Cranfield, UK, 2019, pp. 24-30, doi: 10.1109/REDUAS47371.2019.8999717.

DIFILIPPO, N. M. y JOUANEH, M. K., (2015). "Characterization of Different Microsoft Kinect Sensor Models," in IEEE Sensors Journal, vol. 15, no. 8, pp. 4554-4564, Aug. 2015, doi: 10.1109/JSEN.2015.2422611.

DRONECODE FOUNDATION (2021a). PX4 Autopilot. Disponible en: <https://px4.io/h>

DRONECODE FOUNDATION (2021b). MAVLink: Micro Air Vehicle Communication Protocol. Disponible en: <https://mavlink.io/en/>.

DRONECODE FOUNDATION (2021c). QGroundControl Ground Control Station. Disponible en: <http://qgroundcontrol.com/>.

ERMAKOV V., (2021). Mavros: MAVLink extendable communication node for ROS. Disponible en: <http://wiki.ros.org/mavros>.

FFMPEG Team (2023). A complete, cross-platform solution to record, convert and stream audio and video. Disponible en: <https://ffmpeg.org>

GIRSHICK, R.; DONAHUE, J.; DARRELL, T. y MALIK, J., (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Computer Vision and Pattern Recognition. (CVPR), 2014 IEEE Conference on, pages 580–587. IEEE.

KHOKHAR, N., (2023). YOLOv8 OBJECT DETECTION WITH DEEPSORT TRACKING. Disponible en: <https://github.com/noorkhokhar99/YOLOv8-Object-Detection-with-DeepSORT-Tracking>

MACENSKI, S.; FOOTE, T.; GERKEY, B.; LALANCETTE, C. y WOODALL, W., (2022). "Robot Operating System 2: Design, architecture, and uses in the wild," Science Robotics vol. 7, May 2022. Disponible en: <https://docs.ros.org/en/humble/index.html>

MAHENDRAKAR, T. et al., (2022). "Performance Study of YOLOv5 and Faster R-CNN for Autonomous Navigation around Non-Cooperative Targets," 2022 IEEE Aerospace Conference (AERO), Big Sky, MT, USA, pp. 1-12, doi: 10.1109/AERO53065.2022.9843537.

MEIER, L.; HONEGGER, D. y POLLEFEYS, M., (2015). PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms. IEEE International Conference on Robotics and Automation (ICRA), 6235-6240.

OPEN ROBOTICS (2023). GAZEBOSIM. Disponible en: <https://gazebo.org/home>

PAVLENKO, T.; SCHÜTZ, M.; VOSSIEK, M.; WALTER, T. y MONTENEGRO, S., (2019). "Wireless Local Positioning System for Controlled UAV Landing in GNSS-Denied Environment," 2019 IEEE 5th International Workshop on Metrology for AeroSpace (MetroAeroSpace), Turin, Italy, 2019, pp. 171-175, doi: 10.1109/MetroAeroSpace.2019.8869587.

PALLOS, R; FARKAS, J.; MOLDOVAN, I. y LUKOVSKI, C., (2007). "Performance of rapid spanning tree protocol in access and metro networks,"

- Second International Conference on Access Networks & Workshops, Ottawa, ON, Canada, 2007, pp. 1-8, doi: 10.1109/ACCESSNETS.2007.4447112.
- PROMONET, M., (2023). V4L2TOOLS.
Disponibile en: <https://github.com/mpromonet/v4l2tools>
- PROMONET, M., (2023). V4L2RTSPSERVER.
Disponibile en: <https://github.com/mpromonet/v4l2rtspserver>
- RAKOTONIAINA, Z. A. T.; CHELBI, N. E.; GINGRAS; D. y FAULCONNIER, F., (2023). "LIV-DeepSORT: Optimized DeepSORT for Multiple Object Tracking in Autonomous Vehicles Using Camera and LiDAR Data Fusion," 2023 IEEE Intelligent Vehicles Symposium (IV), Anchorage, AK, USA, pp. 1-7, doi: 10.1109/IV55152.2023.10186759.
- REDMON, J.; DIVVALA, S.; GIRSHICK, R. y FARHADI, A., (2016). "You Only Look Once: Unified, Real-Time Object Detection," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.
- SMYCZYŃSKI, P.; STARZEC, T. y GRANOSIK, G., (2017). "Autonomous drone control system for object tracking: Flexible system design with implementation example," 2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR), Miedzyzdroje, Poland, 2017, pp. 734-738, doi: 10.1109/MMAR.2017.8046919.
- UMLAEUTE (2023). V4L2LOOPBACK.
Disponibile en: <https://github.com/umlaeute/v4l2loopback>
- YOON, Y.; KIM, M.; LEE, S.; LEE, B.; HYUN, S. J. y LEE, K., (2012). "Performance analysis of H.264/AVC, H.264/SVC, and VP8 over IEEE 802.11 wireless networks," 2012 IEEE Symposium on Computers and Communications (ISCC), Cappadocia, Turkey, 2012, pp. 000151-000156, doi: 10.1109/ISCC.2012.6249284.
- ZHANG, Z., (2012). "Microsoft Kinect Sensor and Its Effect," in IEEE MultiMedia, vol. 19, no. 2, pp. 4-10, Feb. 2012, doi: 10.1109/MMUL.2012.24.
- ZHAO, T. y JIANG, H., (2016). "Landing system for AR.Drone 2.0 using onboard camera and ROS," 2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC), Nanjing, pp. 1098-1102, doi: 10.1109/CGNCC.2016.7828941.

