CEA Comité Español de Automática
XLVI JORNADAS AUTOMÁTICA CARTAGENA 2025
UNIVERSIDADE DA CORUÑA

Jornadas de Automática

Sistema de visión embarcado para detección de emergencias desde UAVs

Jiménez Herrera, Nicole[a], Smith Ballester, Laura[b], Blanes Noguera, Francisco[b], Brenes Torres, Juan Carlos[a]

[a] *Instituto Tecnológico de Costa Rica, Cartago, Costa Rica.*
[b] *Instituto de Automática e Informática Industrial, Universitat Politècnica de València, Camino de Vera, s/n, 46022, Valencia, España.*

To cite this article: Jiménez-Herrera, Nicole, Smith-Ballester, Laura, Blanes-Noguera, Francisco, Brenes-Torres, Juan Carlos. 2025. Onboard Vision System for Detecting Emergency Situations on UAVs. Jornadas de Automática, 46. https://doi.org/10.17979/ja-cea.2025.46.12169

Resumen

Uno de los principales desafíos en los sistemas embebidos de Vehículos Aéreos No Tripulados (UAV) es ejecutar tareas de alta carga computacional, como por ejemplo redes neuronales y visión artificial, en tiempo real y de forma autónoma, sin depender de procesamiento externo. Este artículo propone una arquitectura descentralizada que permite implementar un sistema de detección de gestos de la mano a bordo de un dron. Los resultados iniciales muestran que, a alturas de hasta 3.5 metros, se alcanzan exactitudes del 89 % o más en gestos como OK, STOP o SOS (combinación de los gestos CUATRO y PUÑO). Además, los tiempos de detección de gestos simples como STOP y OK no superan, en promedio, los 370 milisegundos, lo que evidencia la eficiencia del sistema a pesar de la carga computacional. Estos resultados demuestran la viabilidad de ejecutar tareas de reconocimiento en tiempo real a bordo de un UAV, manteniendo autonomía de cómputo y capacidad de respuesta para aplicaciones en escenarios reales.

Palabras clave: Arquitecturas de computación embebidas, Robótica embebida, Robots voladores, Interacción humano-vehículo, Internet de las cosas, Algoritmos en tiempo real, Navegación, programación y visión robótica.

Onboard Vision System for Detecting Emergency Situations on UAVs

Abstract

One of the main challenges in embedded systems for Unmanned Aerial Vehicles (UAVs) is executing high-computational-load tasks, such as those using neural networks and computer vision, in real time and autonomously, without relying on external processing. This article proposes a decentralized architecture that enables the implementation of a hand gesture recognition system onboard a drone. Initial results show that, at altitudes of up to 3.5 meters, recognition accuracies of 89 % or higher are achieved for gestures such as OK, STOP or SOS (a combination of FOUR and FIST). Additionally, detection times for simple gestures like STOP and OK average no more than 370 milliseconds, demonstrating the system's efficiency despite the computational demands. These results demonstrate the feasibility of performing real-time recognition tasks onboard a UAV while maintaining computational autonomy and an adequate response time for real-world applications.

Keywords: Embedded computer architectures, Embedded robotics, Flying robots, Human and vehicle interaction, Internet of things, Real-time algorithms, Robos Navigation, Programming and Vision.

1. Introducción

En los últimos años, el uso de Vehículos Aéreos No Tripulados (*Unmanned Aerial Vehicle* o UAVs) ha crecido de forma significativa en diversas áreas, como la vigilancia, la agricultura y, especialmente, la gestión de emergencias. Gracias a su capacidad para acceder a zonas de difícil alcance y recopilar información en tiempo real, los UAVs se han convertido en herramientas clave para la identificación y monitoreo de situaciones críticas.

*Autor para correspondencia: pblanes@ai2.upv.es

No obstante, el diseño de los sistemas embebidos para UAVs presenta varios desafíos. Entre ellos destacan las limitaciones de hardware y procesamiento, especialmente cuando se requiere ejecutar modelos de visión por computadora (*Computer Vision* o CV) y redes neuronales convolucionales (*Convolutional Neural Networks* o CNNs) en tiempo real y de forma autónoma. Además, el espacio físico y el consumo energético dentro de un UAV son reducidos, y se requiere una alta robustez ante posibles fallos del sistema.

En este contexto, se propone un sistema embebido modular que incorpora una cámara con capacidad de cómputo, permitiendo detectar situaciones de emergencia mediante gestos de la mano. Esta propuesta busca apoyar a los recolectores de frutos silvestres en los bosques de Finlandia, quienes suelen enfrentar condiciones climáticas adversas, terrenos difíciles y una conectividad celular limitada, lo que complica la notificación de emergencias (Fletcher et al., 2023).

Esta iniciativa forma parte del proyecto FEROX (Fostering and Enabling AI, Data and Robotics Technologies for Supporting Human Workers in Harvesting Wild Food), el cual busca desarrollar una solución integral que combine inteligencia artificial, visión por computadora y drones para optimizar las actividades de cosecha y asistir a los trabajadores en escenarios de emergencia (Trybała et al., 2024).

2. Enfoque de la solución

La solución propuesta se basa en el desarrollo de una arquitectura capaz de integrar CV, CNNs y computación en el borde. El objetivo es detectar y clasificar gestos manuales sin afectar el rendimiento de otras tareas, como el control de vuelo del UAV, y al mismo tiempo reportar situaciones de emergencia hacia una base en tierra. A continuación, se describe el enfoque adoptado para la arquitectura del sistema, el flujo de trabajo para la detección de gestos, así como el mecanismo de mensajería utilizado para la transmisión de alertas a la estación base.

2.1. *Arquitectura de la solución*

La arquitectura general de la solución se presenta en la Figura 1. Los componentes integrados en el dron incluyen una cámara OAK-D S2, responsable de la captura y procesamiento de imágenes; una Jetson Nano, que actúa como unidad de procesamiento embebido; y un Pixhawk 6C, que funciona como Unidad de Control de Vuelo (*Flight Control Unit* o FCU). El UAV utilizado es el dron Holybro X500 V2.

Esta arquitectura se caracteriza por una *descentralización de los procesos computacionales*, lo que permite distribuir eficientemente las cargas de trabajo entre los diferentes dispositivos manteniendo la autonomía del sistema. Por un lado, el Pixhawk 6C está dedicado exclusivamente al control de vuelo del UAV, gestionando tareas críticas como la estabilización, navegación autónoma y lectura de sensores inerciales y de posicionamiento. La comunicación entre el Pixhawk y la Jetson Nano se establece a través de una interfaz serial UART, lo que permite el intercambio de datos de telemetría y comandos de control. Adicionalmente, el Pixhawk puede mantener conexión con estaciones de control terrestre mediante enlaces de radiofrecuencia, facilitando la supervisión y control del dron desde aplicaciones como *QGroundControl* o mediante un mando RC convencional.

Por otro lado, la cámara OAK-D S2, equipada con óptica de enfoque fijo (*fixed-focus*) —ideal para entornos con vibraciones como los generados por un dron en vuelo—, es responsable de la adquisición y procesamiento de imágenes en tiempo real. Esta cámara integra un sistema estereoscópico de tres sensores, que le permite generar información de profundidad con alta precisión (Luxonis, 2025a). Además, cuenta con una arquitectura basada en el núcleo RVC2 (*Robotics Vision Core 2*), que incluye un SoC (*System On a Chip*) de alto rendimiento optimizado para tareas de visión computacional con DepthAI (Luxonis, 2025b). La comunicación entre la cámara y la Jetson Nano se realiza mediante interfaces USB 2.0 o 3.0, que también sirven como medio de alimentación.

La utilización de la OAK-D S2 permite disminuir la carga computacional del procesador embebido, al procesar localmente las redes neuronales de detección y tareas de visión, lo que reduce el tiempo de procesamiento y mejora el rendimiento global del sistema. En esta propuesta, la Jetson Nano se encarga de recibir los *frames* e información procesada por la cámara (por ejemplo, los gestos detectados), ejecutar algoritmos adicionales para mejorar la robustez del sistema y establecer comunicación con la estación terrestre mediante Wi-Fi para el envío de mensajes de emergencia.

Además, la Jetson puede comunicarse bidireccionalmente con el Pixhawk, lo que permite desarrollar aplicaciones que, desde el entorno Linux embebido, envíen comandos de navegación al FCU o consulten parámetros críticos, como la posición GPS, velocidad o estado del sistema. Esta capacidad es fundamental para el funcionamiento autónomo y para la generación de alertas contextualizadas durante la misión.

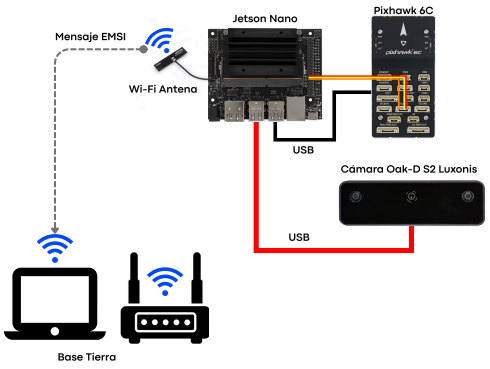


Figura 1: Arquitectura general del sistema para reconocimiento de gestos.

Los componentes de la base en tierra están conformados por un enrutador Wi-Fi, en específico se utiliza el *NFT Blizzard 2 ac-N* de LigoWave, y una computadora portátil. Considerando que el sistema está diseñado para operar en exteriores, donde el acceso a redes móviles o a internet puede ser limitado o inexistente, el enrutador actúa exclusivamente como un enlace de comunicación local entre el dron y la estación terrestre. Para ello, se emplea un modelo apto para condiciones de intemperie y que opera en la banda de 5 GHz, debido a su mayor capacidad de transmisión de datos y menor susceptibilidad a interferencias. La computadora en la base tiene como función principal la recepción, visualización y manejo de los

mensajes de emergencia generados por el UAV.

2.2. *Reconocimiento de gestos*

El subsistema de reconocimiento de gestos está diseñado para operar de manera eficiente en entornos con restricciones de cómputo y energía, como los que impone un UAV. Uno de los principales desafíos al ejecutar modelos de visión por computadora basados en CNNs en unidades de procesamiento de bajo consumo, como la Jetson Nano, es la alta demanda computacional, que puede generar cuellos de botella en el rendimiento general del sistema.

Para mitigar esta limitación, se ha integrado la cámara Luxonis DepthAI OAK-D S2, la cual incorpora una unidad de procesamiento visual que permite la ejecución de tareas de visión y detección directamente en el dispositivo, liberando a la Jetson Nano de estas cargas. Esta arquitectura distribuida mejora la eficiencia energética y reduce la latencia en la detección de eventos críticos.

El reconocimiento de gestos manuales se implementa mediante un *pipeline* de aprendizaje automático (Machine Learning o ML) que combina dos modelos especializados: un detector de palmas y un estimador de puntos clave (*landmarks*) de la mano. Estos modelos, originalmente propuestos por (Zhang et al., 2020), han sido adaptados para operar sobre la plataforma *DepthAI* en dispositivos Luxonis, utilizando implementaciones optimizadas como la de (geaxgx, 2023). La detección se realiza directamente en la cámara, y los resultados se transmiten a la Jetson Nano para su evaluación contextual y posterior generación de alertas.

En la Figura 2, se presenta el flujo de procesamiento distribuido entre la cámara y el procesador principal del UAV. Inicialmente, la detección de la palma se realiza utilizando el modelo *MoveNet* de TensorFlow, mientras que los puntos clave de la muñeca son estimados mediante un modelo basado en *Single-Shot Detector* (SSD). La información extraída es posteriormente procesada por el modelo de reconocimiento de gestos implementado en el *framework MediaPipe*.

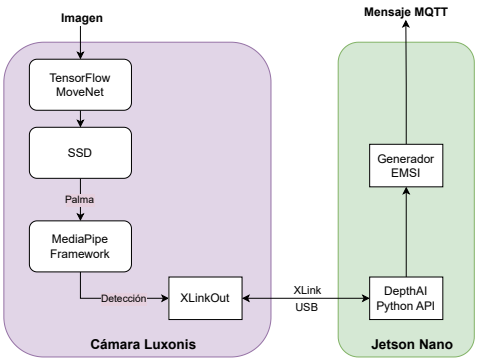
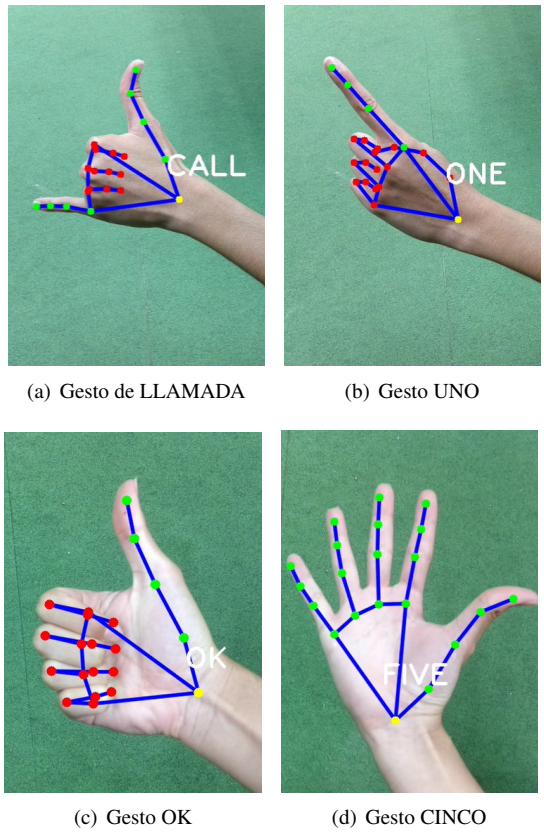


Figura 2: Flujo de trabajo para la detección de gestos con CV y ML pipeline con la Luxonis OAK-D S2.

En la Figura 3 se muestra con mayor detalle la detección de gestos, así como la representación gráfica de la ubicación de los puntos de la mano y la muñeca que utiliza el modelo para clasificar los gestos. El sistema puede detectar gestos tanto por el lado dorsal de la mano, como se observa en las Figuras 3(a) y 3(b); como por el lado palmar, como se muestra en las Figuras 3(c) y 3(d).



(a) Gesto de LLAMADA (b) Gesto UNO

(c) Gesto OK (d) Gesto CINCO

Figura 3: Ejemplos de los gestos que reconoce el pipeline de ML.

Los resultados de la detección generados por la cámara son transmitidos al procesador Jetson Nano mediante el protocolo XLink a través de la conexión USB. En el lado del CPU, se utiliza la API de DepthAI para gestionar la comunicación con la cámara y manipular los datos recibidos. Los resultados se almacenan en estructuras tipo cola, las cuales son evaluadas a través de un umbral de detecciones consecutivas para validar la presencia de un gesto. Esta estrategia, detallada en (Haas, 2024), permite reducir la ocurrencia de falsos positivos y mejorar la confiabilidad del sistema. Luego, se procede con la generación del mensaje de alerta.

El sistema de detección es capaz de reconocer diez gestos manuales: UNO, DOS, TRES, CUATRO, CINCO, PAZ, PUÑO, OK, LLAMAR y LET'S ROCK. No obstante, para efectos de identificación de situaciones de emergencia, se seleccionaron únicamente tres señales clave: el gesto OK como señal de estar bien (véase Figura 3(c)), el gesto CINCO como indicación de STOP (véase Figura 3(d)), y la combinación de CUATRO seguido de PUÑO como alerta de SOS.

En el caso de los gestos OK y STOP, se valida como gesto confirmado si el modelo detecta al menos *cinco instancias* del mismo gesto. Si durante la secuencia de validación se identifican más de *tres gestos distintos* al esperado antes de alcanzar las cinco detecciones, la cola se reinicia, descartando la secuencia.

En cuanto al gesto compuesto SOS, el sistema primero detecta el gesto CUATRO y entra en un estado de espera para

validar la aparición del gesto PUÑO. A partir de esa detección inicial, se comienza a contar el número de gestos detectados subsecuentemente. Si dentro de un máximo de *diez gestos distintos* al PUÑO no se detecta el gesto esperado, la cola se reinicia y se descarta la señal de SOS. En cambio, si el gesto PUÑO es identificado dentro de ese umbral, se valida correctamente la alerta de SOS.

2.3. Mensajería entre el dron y la base

La transmisión de mensajes de emergencia entre el dron y la estación terrestre se implementa utilizando el protocolo de mensajería ligera MQTT (Message Queuing Telemetry Transport). Este protocolo se basa en una arquitectura de publicación-suscripción gestionada a través de un bróker central, lo que lo hace especialmente adecuado para sistemas embebidos y aplicaciones IoT debido a su bajo consumo de ancho de banda y eficiencia en el manejo de mensajes.

En esta arquitectura, el dron actúa como publicador de eventos, mientras que la computadora en la base terrestre cumple el rol de suscriptor y aloja el bróker encargado de coordinar la comunicación. La Figura 4 ilustra la interacción entre los tres elementos principales: publicador, suscriptor y bróker.

MQTT ha sido seleccionado debido a su bajo tamaño de carga útil (payload), soporte para calidad de servicio (QoS) y mecanismos de confirmación de recepción por parte del bróker, lo cual incrementa la confiabilidad en la entrega de mensajes críticos en entornos con conectividad limitada o variable.

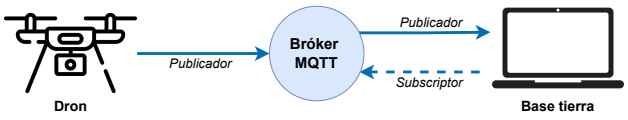


Figura 4: Interacción del protocolo de aplicación para envío de mensajes entre el dron y la base.

Además, con el fin de mantener compatibilidad con sistemas de gestión de emergencias existentes y garantizar la interoperabilidad entre plataformas, los mensajes enviados por el dron se estructuran siguiendo el estándar EMSI (Emergency Message Standard Interface) definido por la norma ISO/TR 22351:2015 (International Organization for Standardization, 2015). Este estándar define una gramática común para representar información crítica en contextos de emergencia, lo que facilita su integración con infraestructuras ya implementadas en centros de mando o sistemas de respuesta rápida, como por ejemplo los centros de servicio de emergencias.

Los mensajes EMSI se codifican utilizando el formato XML, lo que permite una representación jerárquica, legible y validable del contenido. Cada evento detectado por el sistema (en este caso OK, STOP o SOS) genera un archivo XML con los campos requeridos por el estándar, tales como identificador del incidente, ubicación, tipo de evento, prioridad, y marca temporal. Este archivo se transmite como contenido del mensaje MQTT, donde el payload del mensaje contiene el XML completo como cadena codificada en UTF-8.

Esta integración asegura que, además de una comunicación ligera y eficiente proporcionada por MQTT, los datos transmitidos cumplan con criterios de interoperabilidad, trazabilidad y estructuración que exigen las aplicaciones de misión crítica.

No obstante, esta arquitectura es escalable y adaptable, por lo que podría integrarse con otros medios de comunicación más robustos en aplicaciones futuras. En este caso, debido a las limitaciones de conectividad con el sistema embebido, MQTT es el protocolo más viable. Pero por ejemplo, en entornos donde se disponga de conectividad celular o redes más complejas - y con drones con más opciones y capacidad de comunicación - el sistema podría modificarse para establecer una comunicación directa con servicios de emergencia como el 112, permitiendo el envío automatizado de alertas críticas con información contextual sobre la emergencia detectada por el UAV.

2.4. Pruebas

Para validar la propuesta, se realizaron dos pruebas. La primera evaluó exclusivamente el sistema de visión. El dron voló en un entorno exterior mientras una persona realizó gestos OK, STOP y SOS dentro de su campo visual. Se registraron los gestos válidos detectados junto con marcas de tiempo, y luego se compararon con el video para clasificar los resultados como verdaderos positivos, falsos positivos y falsos negativos. Dado el umbral bajo de validación, se esperaban secuencias repetidas del mismo gesto; eliminando la primera detección, se calculó el tiempo promedio de respuesta del sistema para validar un gesto. Se realizó la prueba a dos diferentes alturas del dron: 2.5 m y 3.5 m. La posición de la cámara se encontraba en un ángulo de 45°.

La segunda prueba evaluó la confiabilidad de la comunicación. Con la estación base fija, el dron fue desplazado manualmente entre 100 y 150 m, generando cada 5 s un mensaje EMSI en formato XML para simular detecciones. Se midieron el tiempo de envío y la tasa de éxito en la entrega de los mensajes.

3. Resultados

A continuación, se presentan los resultados obtenidos en cada una de las pruebas, evaluados bajo métricas de desempeño como la tasa de verdaderos positivos, tiempo de detección, la latencia de detección y la tasa de entrega de mensajes.

3.1. Detección de gestos con dron volando

En las Figuras 5(a) y 5(b) se observa cómo se mostraron los gestos al dron. De forma general, se mostraba el gesto OK y STOP por aproximadamente 3 s, y el gesto SOS se mostraba en una secuencia de entre 3 a 5 veces seguidas. Para tener claridad entre el inicio y final de un gesto, se escondió la mano por aproximadamente 3 s entre los diferentes gestos.

Respecto a los resultados de la detección de gestos, en la Tabla 1 se presentan los resultados de la cantidad de verdaderos positivos (TP), falsos positivos (FP) y falsos negativos (FN), así como la exactitud del sistema por cada gesto, calculado por $TP/(TP + FN)$. Los resultados se muestran por cada altura en la que se posicionó el dron.

(a) Gesto CUATRO (b) Gesto PUÑO

Figura 5: Ejemplo de prueba de gestos a una altura de 3.5 m del dron.

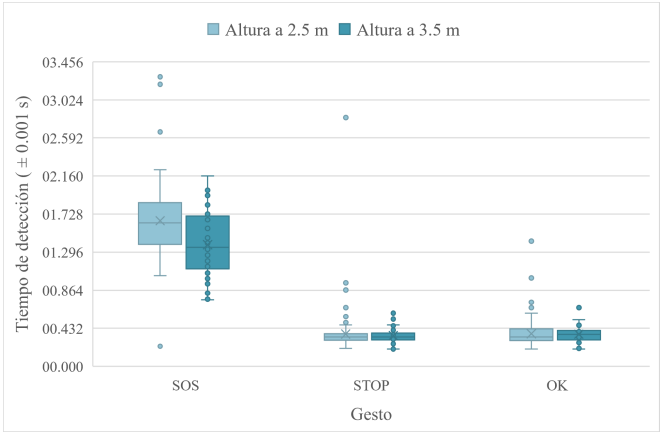


Figura 6: Distribución de los datos en caja de bigotes para la detección de gestos según su altura.

Los resultados de la Tabla 1 sugieren que el sistema presenta una alta exactitud general en la detección de gestos. Sin embargo, en el caso del gesto SOS —considerado el más crítico dentro del contexto de emergencia— se observa una menor exactitud. El análisis del video de la transmisión y del registro de gestos evidencia que los falsos negativos en la detección del SOS están asociados a la confusión del modelo entre los gestos PUÑO y OK, especialmente cuando la mano aparece cerrada. Esta efecto también se refleja en la cantidad elevada de falsos positivos para el gesto OK. Estos resultados podrían sugerir la necesidad de ajustar el filtro de validación para mejorar la robustez del sistema.

Tabla 1: Resultados de porcentaje de exactitud, positivos verdaderos (TP), falsos positivos (FP) y falsos negativos (FN) en las pruebas de detección para cada tipo de gesto en vuelos del dron.

Altura	Gesto	TP	FP	FN	Exactitud
2.5 ± 0.5 m	OK	20	11	1	95.24 %
	STOP	19	2	0	100.00 %
	SOS	63	3	3	95.45 %
3.5 ± 0.5 m	OK	20	19	0	100.00 %
	STOP	22	4	0	100.00 %
	SOS	66	1	8	89.19 %

Por otro lado, la Tabla 2 presenta los tiempos de detección del sistema para cada gesto. A diferencia de la Tabla 1, esta incluye todas las ocurrencias dentro de las secuencias que conforman un gesto válido, por lo que el número de muestras (n) por gesto es mayor, específicamente en los casos de OK y STOP. Para cada gesto, se calcula el tiempo promedio entre detecciones consecutivas dentro de una misma secuencia (excluyendo la primera detección, que se toma como referencia), así como la desviación estándar (Desv. Est.), el valor máximo y mínimo de estos intervalos. Estos valores se obtienen a partir de las marcas de tiempo registradas en el historial de detección, y permiten estimar el tiempo típico requerido por el sistema para validar un gesto. De forma más gráfica, se observa en la Figura 6 la distribución de los datos en formato caja de bigotes.

Es importante destacar que el tiempo de detección del gesto SOS tiende a ser mayor, ya que su validación depende de una secuencia compuesta por dos gestos consecutivos: CUATRO y PUÑO. Este tiempo varía según la duración con la que el operador mantiene el gesto CUATRO tras una detección válida de SOS, dado que, para el análisis de tiempos de reacción, se descarta la primera detección. Adicionalmente, la aparición de gestos distintos al PUÑO durante la validación —como el gesto OK, que suele confundirse visualmente con el puño cerrado— no reinicia de inmediato la cola de validación si su frecuencia se mantiene por debajo del umbral, pero sí prolonga el proceso de detección al retrasar la validación del SOS.

Lo anterior sugiere que, aunque el sistema tiene cierto nivel de robustez ante interrupciones leves, existe un compromiso entre tolerancia a errores y velocidad de detección en gestos compuestos.

3.2. *Comunicación entre dron y base en tierra*

En cuanto a la comunicación entre el dron y la base mediante Wi-Fi y el protocolo MQTT, la Tabla 3 presenta los resultados de la tasa de éxito en la entrega de mensajes, así como el tiempo promedio de transmisión —medido desde el envío del mensaje en el dron hasta la confirmación de recepción por parte del bróker— y su desviación estándar (Desv. Est.). Las pruebas se realizaron en un entorno exterior con mínima presencia de vegetación y sin interferencias urbanas, como edificios o viviendas cercanas, con el fin de acercarse más a las condiciones de una aplicación final.

Tabla 3: Resultados de la prueba de mensajería por MQTT entre el dron y la base terrestre a una distancia de hasta 100 m: cantidad de mensajes enviados y recibidos, porcentaje de tasa de éxito, tiempo promedio y desviación estándar del tiempo de envío.

Prueba	Enviados	Recibidos	Tasa éxito	Tiempo promedio (±0,001 ms)	Desv. Est. (±0,001 ms)
1	196	196	100 %	89.378	89.144
2	186	186	100 %	83.446	83.980
	Promedio		100 %	86.412	86.562

Los resultados obtenidos sugieren que, para distancias de hasta 100 m entre el dron y la base en tierra, se mantiene una conexión segura sin pérdida de mensajes en ninguna de las pruebas. Este comportamiento sugiere que el protocolo

Tabla 2: Tiempo de detección de gestos a diferentes alturas con el dron volando.

Altura a 2.5 ± 0.5 m					
Gesto	**n**	**Promedio (± 1 ms)**	**Desv. Est. (± 1 ms)**	**Máximo (± 1 ms)**	**Mínimo (± 1 ms)**
OK	125	370	150	1 421	198
STOP	119	366	249	2 824	204
SOS	46	1 653	500	3 285	228
Altura a 3.5 ± 0.5 m					
Gesto	**n**	**Promedio (± 1 ms)**	**Desv. Est. (± 1 ms)**	**Máximo (± 1 ms)**	**Mínimo (± 1 ms)**
OK	89	365	88.5	666	196
STOP	82	344	69.5	567	197
SOS	46	1 381	397	2 161	664

MQTT y la conexión Wi-Fi empleados son efectivos en distancias moderadas, lo que garantiza la fiabilidad de la transmisión en el entorno evaluado.

El tiempo promedio de envío y confirmación de recepción del mensaje es de 86.4 ms, lo que es un valor razonablemente bajo, lo que resalta la eficiencia del sistema de comunicación. Sin embargo, la desviación estándar promedio de 86.56 ms sugiere que existen fluctuaciones en el tiempo de transmisión, lo cual podría deberse a varios factores. Entre estos, se destacan posibles interferencias en la señal Wi-Fi, pequeñas variaciones en la capacidad de procesamiento del dron y la base en tierra, o fluctuaciones en la calidad de la conexión dependiendo de las condiciones ambientales.

A pesar de estas variaciones, los resultados obtenidos en términos de confiabilidad y tiempos de respuesta son adecuados para aplicaciones donde se requiere una comunicación en tiempo real para la detección y notificación de emergencias.

4. Conclusiones

Se concluye que el sistema propuesto demuestra ser viable y eficiente para la detección de gestos de la mano desde un sistema embebido a bordo de un UAV. La arquitectura implementada ha mostrado ser capaz de manejar procesamiento de imágenes y ejecución de CNNs en tiempo real, incluso durante el vuelo, lo cual es un avance significativo en el desarrollo de aplicaciones de visión computacional en drones de bajo consumo.

Uno de los principales aportes de esta solución es su capacidad de operar bajo las limitaciones propias de sistemas embebidos, manteniendo una respuesta rápida y con un nivel de exactitud de más del 89 %, lo cual es aceptable. Esto permite su aplicación en escenarios reales como la asistencia a recolectores de frutos o la detección de emergencias mediante gestos humanos, sin depender de procesamiento en la nube ni de enlaces de alta capacidad.

La integración de sistemas de CV y CNNs en UAVs para detectar emergencias mediante gestos de la mano, junto con el uso de MQTT para la comunicación, ha mostrado ser funcional en las primeras pruebas. Sin embargo, se deben poner a prueba factores como la iluminación, el contraste y las variaciones anatómicas de la mano.

El modelo reconoce gestos hasta 5 metros de distancia. Aunque no se alcanzó este límite en las pruebas, es importante considerar la distancia de seguridad entre el UAV y la persona. Un soporte a 45° ofrece ventajas sobre un soporte vertical, ya que permite una mayor distancia horizontal de seguridad, beneficiando la interacción con el sistema.

Finalmente, se recomienda incluir un sistema de alertas, ya sea visual o auditivo, para alertar a la persona usuaria que su gesto fue detectado y reconocido correctamente, lo que aumentaría la confiabilidad en situaciones de emergencia.

Agradecimientos

El proyecto FEROX ha recibido financiación del Programa Marco *Horizon* de la Unión Europea para la Investigación y la Innovación, en virtud del Acuerdo de Subvención nº 101070440, correspondiente a la convocatoria HORIZON-CL4-2021-DIGITALEMERGING-01-10: *IA, Data and Robotics at work (IA)*.

Referencias

Fletcher, S., Oostveen, A. M., Chippendale, P., Couceiro, M., Turtiainen, M., Ballester, L. S., 2023. Developing unmanned aerial robotics to support wild berry harvesting in finland: Human factors, standards and ethics. In: Proceedings of the International Conference on Robot Ethics and Standards (ICRES). pp. 109–122, available online, accessed May 14, 2025.
URL: https://clawar.org/icres2023/wp-content/uploads/2024/01/ICRES2023-Proceedings-Manuscript.pdf

geaxgx, 2023. depthai_hand_tracker. https://github.com/geaxgx/depthai_hand_tracker.

Haas, P., 2024. Developing enhanced drone operations: Enabling gesture recognition with integration of a custom fleet management system based on rooster. Tesis de máster, Escuela de Ingeniería Industrial, Universitat Politècnica de València, Valencia, España.

International Organization for Standardization, September 2015. ISO/TR 22351:2015 societal security — emergency management — message structure for exchange of information. Technical Report ISO/TR 22351:2015, ISO, Geneva, Switzerland, first edition.
URL: https://www.iso.org/standard/57384.html

Luxonis, 2025a. Oak-d s2 fixed-focus. https://shop.luxonis.com/products/oak-d-s2?variant=42455432265951, accedido el 15 de mayo de 2025.

Luxonis, 2025b. Rvc2 nn performance. https://docs.luxonis.com/hardware/platform/rvc/rvc2/#RVC2%20NN%20Performance, accedido el 15 de mayo de 2025.

Trybała, P., Morelli, L., Remondino, F., Farrand, L., Couceiro, M. S., 2024. Under-canopy drone 3d surveys for wild fruit hotspot mapping. Drones 8 (10).
URL: https://www.mdpi.com/2504-446X/8/10/577
DOI: 10.3390/drones8100577

Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C., Grundmann, M., 2020. Mediapipe hands: On-device real-time hand tracking. CoRR abs/2006.10214.
URL: https://arxiv.org/abs/2006.10214