

Jornadas de Automática

Robot autónomo lanzador y recogedor de pelotas de pádel para entrenamiento deportivo

Fuentes, Francisco, L., Rubio, Daniel, Tejada, Helio, Chicaiza, William D.*, Escaño, Juan Manuel

Departamento de Ingeniería de Sistemas y Automática, Universidad de Sevilla, Camino de los Descubrimientos s/n, 41092, Sevilla, España.

To cite this article: Fuentes, F. L., Rubio, D., Tejada , H., Chicaiza, W. D., Escaño, J.M. 2025. Autonomous robot for launching and collecting paddle tennis balls for sports training.
Jornadas de Automática, 46. <https://doi.org/10.17979/ja-cea.2025.46.12263>

Resumen

Este trabajo presenta el desarrollo de un sistema robótico autónomo para la detección, recogida y lanzamiento de pelotas, con aplicación en entornos deportivos. La arquitectura combina una *Raspberry Pi* 5 para el procesamiento de visión artificial con un microcontrolador *ESP32* encargado del control distribuido, optimizando la eficiencia computacional y la velocidad de respuesta. La detección se basa en el modelo *YOLOv8*, seleccionado por su elevada precisión en tiempo real. Para su ejecución eficiente en plataformas embebidas, se emplea el framework *ncnn*, lo que permite alcanzar una tasa estable de 6 fps en la *Raspberry Pi*. El sistema de control utiliza información visual —posición y distancia de la pelota— para generar órdenes de movimiento, ejecutadas mediante ruedas *Mecanum* que proporcionan maniobrabilidad omnidiireccional. Además, se implementa una máquina de estados para gestionar los modos de operación: recogida, manual y lanzamiento. La solución desarrollada resulta robusta, flexible y de bajo coste, constituyendo una base prometedora para aplicaciones en robótica autónoma deportiva o asistencial.

Palabras clave: Visión Artificial, Redes Neuronales Artificiales, Robot Autónomo, Sistemas mecatrónicos, Control inteligente, Robótica en el deporte.

Autonomous robot for launching and collecting paddle tennis balls for sports training

Abstract

This work presents the development of an autonomous robotic system for the detection, collection and launching of balls, with application in sports environments. The architecture combines a *Raspberry Pi* 5 for artificial vision processing with a *ESP32* microcontroller in charge of distributed control, optimising computational efficiency and response speed. The detection is based on the *YOLOv8* model, selected for its high real-time accuracy. For efficient execution on embedded platforms, the *ncnn* framework is used, allowing a stable rate of 6 fps to be achieved on the *Raspberry Pi*. The control system uses visual information - position and distance of the ball - to generate movement commands, executed by means of *Mecanum* wheels that provide omnidirectional manoeuvrability. In addition, a state machine is implemented to manage the operation modes: pick-up, manual and launch. The developed solution is robust, flexible and low-cost, constituting a promising basis for applications in autonomous sports or assistive robotics.

Keywords: Artificial Vision, Artificial Neural Networks, Autonomous Robots, Mechatronic Systems, Intelligent Control, Robotics in Sports.

1. Introducción

El pádel se ha consolidado en los últimos años como uno de los deportes con mayor crecimiento en España, tanto en

número de practicantes como en instalaciones habilitadas para su práctica (Statista, 2024). Este auge ha generado una creciente demanda de soluciones tecnológicas orientadas a optimizar la experiencia de entrenamiento, tanto para jugadores

* Autor para correspondencia: wchicaiza@us.es
Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

aficionados como para deportistas de alto nivel.

Uno de los principales desafíos en los entrenamientos individuales es la escasa autonomía de las sesiones: el jugador necesita un asistente que lance las pelotas y, adicionalmente, debe invertir parte del tiempo en su recogida manual, lo que interrumpe el flujo del ejercicio y disminuye su efectividad. Para paliar esta situación, han surgido en el mercado diversos dispositivos automatizados —lanzadores o recogepelotas— que abordan parcialmente el problema. Sin embargo, estos sistemas presentan importantes limitaciones: suelen ser voluminosos, de gran peso, con movilidad nula o limitada, y, sobre todo, con precios elevados que restringen su adopción generalizada.

La mayoría de las soluciones actuales están pensadas para deportes más consolidados como el tenis. Aunque algunos modelos han sido adaptados recientemente a disciplinas en expansión como el pádel y el pickleball (Marin, 2025), el número de lanzapelotas específicamente diseñados para pádel sigue siendo reducido, y sus características replican en gran medida las de sus predecesores: máquinas estáticas, orientadas a lanzar pelotas con configuraciones preestablecidas, sin capacidad de movilidad autónoma ni integración con sistemas de recogida.

En cuanto a la recogida de pelotas, esta sigue siendo una tarea predominantemente manual, a pesar de los intentos de automatización recientes. Algunos dispositivos, como el Tenibot (H.A.R. Eletrabi, 2024) para tenis o prototipos como PickBall (Pereira et al., 2012) en el ámbito del golf, han explotado la incorporación de visión artificial y navegación autónoma. No obstante, las soluciones disponibles comercialmente no ofrecen sistemas que combinen ambas funcionalidades —lanzamiento y recogida— en un solo robot operativo, y su precio sigue siendo una barrera para un uso extensivo.

Ante esta carencia, se propone el desarrollo de un sistema robótico autónomo, compacto y de bajo coste, capaz de integrar de forma simultánea las funciones de lanzamiento, recogida y almacenamiento de pelotas de pádel. Este proyecto, concebido en un entorno académico, no solo da respuesta a una necesidad práctica en el ámbito del entrenamiento deportivo, sino que también constituye una plataforma de aprendizaje multidisciplinar en la que convergen tecnologías de visión por computador, control embebido, diseño mecánico y comunicaciones inalámbricas. Además, su viabilidad técnica y económica lo posiciona como una solución potencialmente transferible a entornos reales.

El resto del artículo se organiza como sigue. La Sección 2 presenta la solución propuesta y los objetivos del sistema. En la Sección 3 se describe la arquitectura del robot, incluyendo el diseño mecánico, los subsistemas electrónicos y las comunicaciones. Las Secciones 4 y 5 abordan los algoritmos de detección de pelota y control de movimiento, basados en visión artificial e inteligencia artificial. La Sección 6 expone los resultados de la validación experimental. Finalmente, las conclusiones se recogen en la Sección 7, y las posibles mejoras futuras en la Sección 8.

2. Solución propuesta

Como respuesta a las limitaciones identificadas en las soluciones actuales para entrenamiento de pádel, se ha desarro-

llado un sistema robótico autónomo capaz de lanzar, recoger y almacenar pelotas de forma completamente automática. Esta triple funcionalidad, aún no integrada en ningún producto comercial, permite plantear sesiones de entrenamiento individuales más eficientes, sin necesidad de asistencia humana directa.

El robot está concebido como una herramienta versátil que puede ser empleada tanto por clubes y asociaciones deportivas —facilitando el acceso a entrenamientos estructurados a usuarios sin recursos para contratar entrenadores personales— como por profesionales del deporte, que podrán optimizar su tiempo de instrucción delegando ciertas tareas repetitivas al sistema.

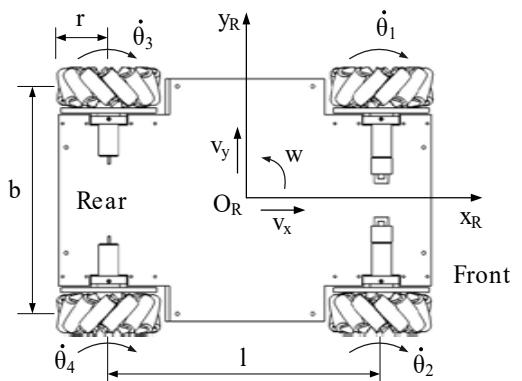


Figura 1: Modelo cinemático de la plataforma móvil omnidireccional con ruedas Mecanum (Peng et al., 2016). Se indican las variables geométricas (l , b , r), las velocidades lineales (v_x , v_y), la velocidad angular w respecto al centro de masas O_R , así como las velocidades angulares individuales de cada rueda $\dot{\theta}_i$.

La plataforma robótica cuenta con movilidad omnidireccional (ver Figura 1), basada en un conjunto de cuatro ruedas Mecanum, lo que le permite desplazarse por toda la pista de pádel con libertad de movimiento en todas direcciones. Este desplazamiento es clave para la localización y recogida de pelotas distribuidas aleatoriamente tras cada ejercicio. El modelado cinemático de este tipo de plataformas ha sido ampliamente estudiado en la literatura (por ejemplo, (Peng et al., 2016; Qian et al., 2017)), y permite establecer formalmente la relación entre las velocidades de las ruedas y los desplazamientos lineales y rotacionales del vehículo. En el presente trabajo, se asume dicho modelo como base para la implementación del control de navegación.

El sistema emplea tres baterías: una dedicada a los mecanismos de recogida y lanzamiento, y dos adicionales para la alimentación de los motores de tracción. Las pelotas se capturan mediante un conjunto de rodillos que las impulsan hacia un depósito de almacenamiento interno. Desde allí, se canalizan hacia el sistema de lanzamiento, compuesto por dos discos giratorios, capaces de proyectar la pelota con ángulos ajustables entre 20° y 70° y velocidades de hasta 15 m/s.

Todos los subsistemas están montados sobre un chasis ligero de perfiles de aluminio, que aporta robustez estructural y facilidad de montaje. El control del sistema se gestiona mediante una Raspberry Pi y un ESP32, que se reparten tareas de percepción, comunicación y actuación.

Tabla 1: Resumen de capacidades del sistema robótico

Movilidad y estructura	Sistema energético	Módulo de recogida y almacenamiento
Tipo: Omnidireccional (Mecanum) Vel. máxima: ~1.5 m/s Chasis: Aluminio modular	Baterías: 1 recogida + 2 tracción Autonomía: Dependiente del uso	Recogida: Rodillos motorizados Almacenamiento: Depósito interno
Módulo de lanzamiento	Control y procesamiento	Visión por computador
Tecnología: Discos motorizados Ángulo: 20°–70° Vel. disparo: hasta 15 m/s	Unidad central: Raspberry Pi 5 Microcontrolador: ESP32 Comunicación: UART, Wi-Fi	Detector: YOLOv8 + ncnn Capacidades: Posición y distancia

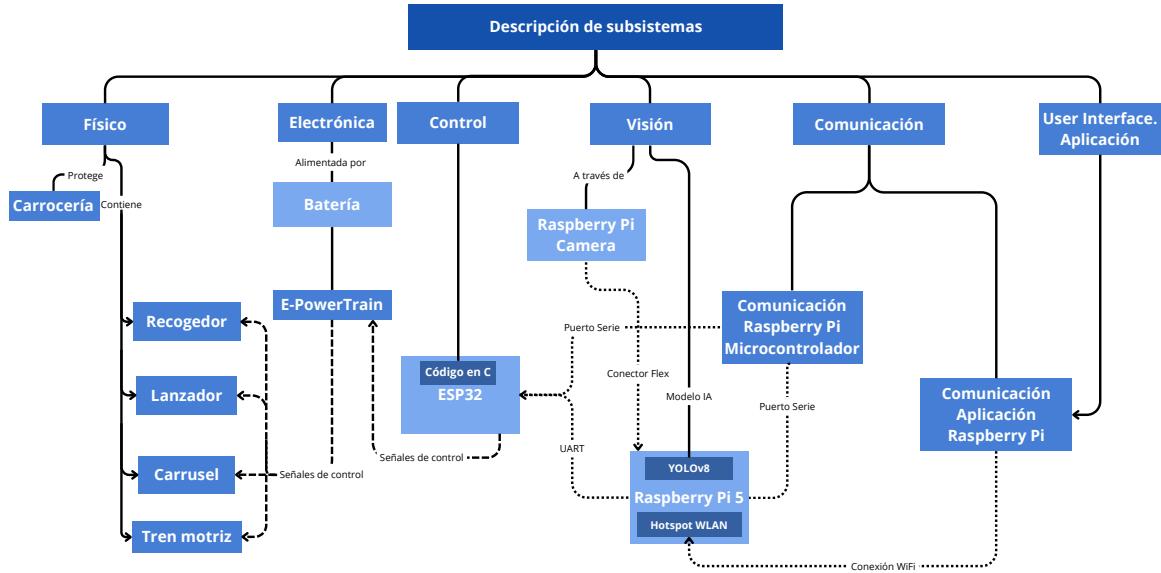


Figura 2: Subsistemas principales del sistema robótico

Para la detección de las pelotas, se ha integrado un modelo de visión por computador basado en YOLOv8, optimizado para su ejecución en Raspberry Pi mediante el framework ncnn. Este modelo permite localizar la posición relativa de cada pelota en la imagen capturada, estimar su distancia mediante triangulación y enviar esta información al microcontrolador encargado de calcular la trayectoria y controlar el desplazamiento del robot hacia el objetivo.

El resultado es un sistema compacto, funcional y de bajo coste, diseñado para integrarse en entornos reales de entrenamiento, y que representa una contribución original en el ámbito de la automatización aplicada al deporte. Las principales especificaciones técnicas del sistema se resumen en la Tabla 1.

3. Arquitectura del sistema

El sistema robótico desarrollado presenta una arquitectura modular que integra componentes mecánicos, electrónicos y de software, permitiendo la operación autónoma en tareas de lanzamiento y recogida de pelotas. Su diseño busca maximizar la funcionalidad, la eficiencia energética, la facilidad de implementación y minimizar los costes.

3.1. Hardware

Desde el punto de vista estructural, se emplean ruedas omnidireccionales Mecanum que permiten una alta maniobrabilidad. El mecanismo de manipulación incluye cuatro rodillos para succión y propulsión de pelotas, un depósito con capacidad para veinte unidades, un carrusel dosificador que regula la cadencia de lanzamiento, y un conducto que conecta el depósito con el sistema de lanzamiento.

El subsistema de control y visión se basa en dos unidades de procesamiento: una Raspberry Pi 5 y un microcontrolador ESP32. La captura de vídeo en tiempo real se realiza mediante una cámara compatible con la Raspberry Pi. La Raspberry Pi 5 se encarga del procesamiento de imagen, mientras que las tareas menos exigentes se delegan al ESP32, con el fin de reducir la latencia causada por la carga computacional. La comunicación entre ambos dispositivos es crítica, ya que cualquier retraso afecta directamente el rendimiento del sistema. Cada subsistema dispone de una fuente de alimentación independiente, con el fin de garantizar la estabilidad operativa ante los picos de corriente generados por los motores de lanzamiento y recogida, así como por la Raspberry Pi durante el procesamiento intensivo asociado a la detección y recogida de pelotas mediante redes neuronales.

3.2. Software

La comunicación entre módulos se establece a través de un servidor TCP, desarrollado en Python e implementado en la Raspberry Pi, que actúa como punto de acceso WLAN. La interfaz de usuario, desarrollada en Flutter (Dart), envía los datos al servidor, que a su vez los transmite mediante UART al ESP32, como se ilustra en la Figura 2. Este último decodifica las tramas y genera señales de control utilizando modulación por ancho de pulso (PWM), las cuales se emplean para accionar los motores.

4. Sistema de visión basado en inteligencia artificial

Para la tarea de detección de objetos en tiempo real se selecciona el modelo YOLOv8, debido a su eficacia demostrada en múltiples aplicaciones (Feng et al., 2019; Deepa et al., 2019; Jocher et al., 2023). Otros algoritmos más sencillos no han sido considerados debido a la alta tasa de falsos positivos. El entrenamiento, validación y prueba se realizan utilizando un conjunto de datos generado mediante herramientas de anotación y preprocesamiento ofrecidas por Roboflow (Nathan, 2025), el cual incluye múltiples escenarios representativos de la aplicación objetivo.

El modelo se evalúa en condiciones reales, considerando variaciones en la iluminación y la dinámica del entorno, con el objetivo de garantizar la robustez del sistema. La implementación se lleva a cabo sobre una Raspberry Pi 5, cuya capacidad computacional permite ejecutar el modelo a una tasa inicial de aproximadamente 3 fps.

La tasa de fotogramas obtenida resulta insuficiente para los requerimientos del sistema. Con el fin de mejorar el rendimiento, el modelo se convierte al framework ncnn, una solución optimizada para dispositivos móviles (Ni and ncnn contributors, 2017). Esta adaptación permite duplicar la velocidad de inferencia, alcanzando una media estable de 6 fps, lo que mejora significativamente la capacidad de detección en tiempo real.

4.1. Evaluación del rendimiento del modelo

La Figura 3 muestra la matriz de confusión normalizada correspondiente al modelo YOLOv8 tras su evaluación sobre el conjunto de prueba. Se consideran dos clases: *pelota de pádel* y *fondo*.

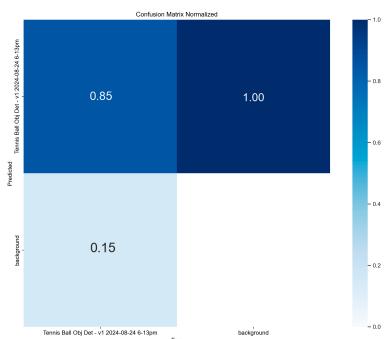


Figura 3: Matriz de confusión normalizada para el modelo YOLOv8 aplicado a la detección de pelotas de pádel

El análisis revela un comportamiento equilibrado del modelo, con un sesgo conservador hacia la clase *fondo*. En particular, el 85 % de las pelotas presentes en las imágenes se detectan correctamente, mientras que el 15 % restante se clasifica erróneamente como fondo, lo que representa los falsos negativos. Por otro lado, el modelo no comete errores de clasificación inversa: ninguna región correspondiente al fondo se clasifica erróneamente como pelota de pádel, lo que implica una tasa de falsos positivos nula.

Este comportamiento es deseable en sistemas autónomos, ya que minimiza el riesgo de acciones innecesarias provocadas por detecciones erróneas. Sin embargo, una tasa del 15 % de falsos negativos podría derivar en oportunidades perdidas de recogida, especialmente en entornos dinámicos. Para mitigar este efecto, se reforzó el conjunto de entrenamiento incorporando imágenes con variabilidad en la iluminación, occlusiones parciales y movimiento. Asimismo, se ajustó el umbral de detección con el fin de compensar los posibles sesgos derivados de una configuración conservadora.

En conjunto, la precisión obtenida en la clase *fondo* es del 100 %, mientras que la sensibilidad (recall) en la detección de pelotas alcanza el 85 %. Estos resultados posicionan al modelo como una solución robusta para aplicaciones en tiempo real, siempre que se contemple el margen de mejora en la cobertura de detección.

Como ejemplo representativo del rendimiento alcanzado, la Figura 4 muestra la detección de una pelota de pádel mediante el modelo entrenado, evidenciando su capacidad para identificar objetos en las condiciones previamente descritas. La escena simula una pista de pádel de color azul, tonalidad habitual en muchas instalaciones de este deporte.

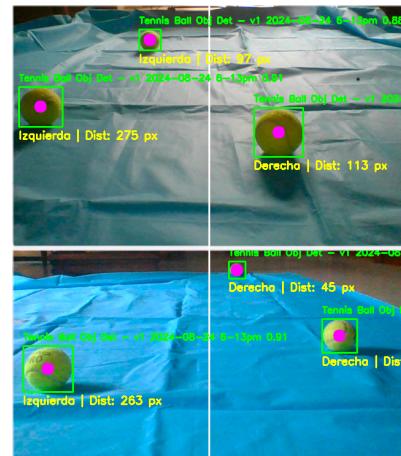


Figura 4: Resultado de inferencia del modelo YOLOv8: detección de una pelota de pádel en condiciones reales

5. Sistema de control distribuido

Una vez optimizado el modelo de visión, se implementa el sistema de control, evitando su integración directa en la Raspberry Pi para no comprometer el rendimiento del proceso de inferencia. La carga computacional adicional derivada de dicha integración reduce significativamente la tasa de fotogramas del sistema. Con el objetivo de mantener un funcionamiento eficiente, el control se externaliza mediante un micro-

controlador ESP32, que asume las tareas de interpretación de datos y generación de órdenes de movimiento.

El sistema de control se fundamenta en dos parámetros principales: la posición de la pelota respecto al centro de la imagen (expresada en píxeles) y su distancia al robot, estimada mediante técnicas de triangulación. Esta información se transmite al ESP32 a través de un enlace serie. El microcontrolador procesa los datos recibidos y genera los comandos de movimiento correspondientes. En función de la posición horizontal, se determina la dirección del desplazamiento; la distancia, por su parte, se emplea para ajustar dinámicamente la velocidad. La Figura 5 ilustra la lógica de control basada en la posición relativa de la pelota.

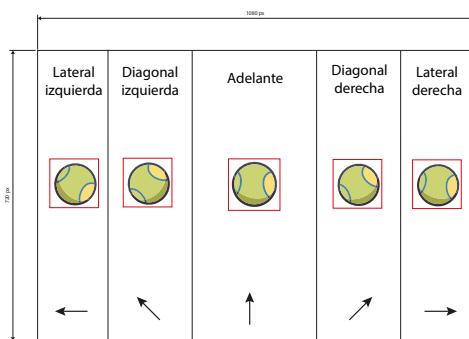


Figura 5: Lógica de control del movimiento en función de la posición horizontal de la pelota detectada

El robot emplea ruedas Mecanum, que permiten desplazamientos omnidireccionales sin necesidad de rotación sobre su eje, lo que incrementa la maniobrabilidad y precisión durante la recogida de pelotas. La comunicación entre la Raspberry Pi y el ESP32 se optimiza para activarse únicamente cuando se detecta una pelota, manteniendo el canal serie libre el resto del tiempo para comunicaciones adicionales, como la interacción con la aplicación de control o tareas de supervisión remota.

5.1. Control de actuadores y modos de operación

El control de los motores de recogida y lanzamiento, así como de los servos de orientación y el carrusel del depósito, se implementa mediante una máquina de estados con tres modos principales:

1. *Modo recogida*: Se activan exclusivamente los motores que accionan los rodillos de recogida, encargados de propulsar las pelotas hacia el depósito mediante fricción.
2. *Modo manual*: Permite el control directo del desplazamiento a través de la aplicación móvil, activando únicamente los motores de tracción.
3. *Modo lanzamiento*: Se activan los motores de lanzamiento y los servos de orientación, que permiten modificar el ángulo del disparo desde la aplicación. Además, se pone en funcionamiento el motor del carrusel, el cual regula el acceso de las pelotas al tubo de alimentación, evitando atascos cuando el sistema no se encuentra en modo de disparo.

Las variables de entrada utilizadas por el sistema de control se representan gráficamente en la Figura 6, donde se observan tanto la distancia estimada como la posición relativa de la pelota respecto al eje de referencia.

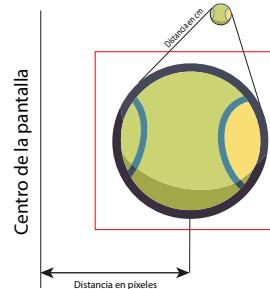


Figura 6: Parámetros de entrada proporcionados por el sistema de visión: posición horizontal y distancia estimada

Esta arquitectura basada en una máquina de estados permite un control estructurado y robusto del sistema, como se muestra la Figura 7, garantizando una operación precisa, segura y adaptable a distintas condiciones de funcionamiento.

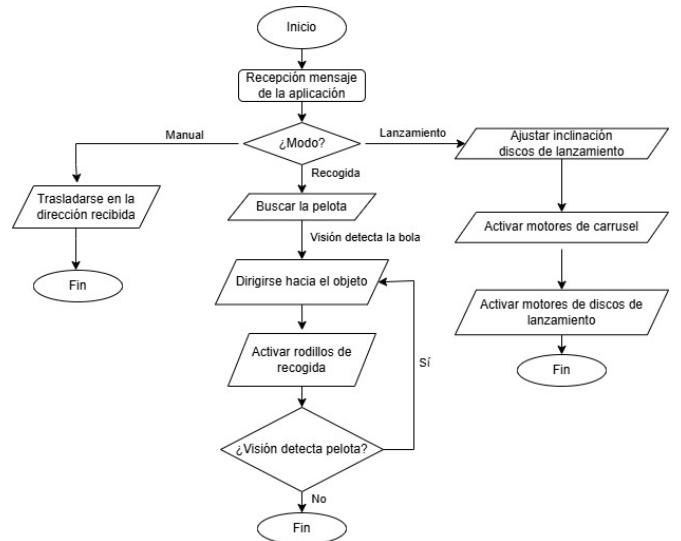


Figura 7: Diagrama de flujo del sistema de control

6. Validación experimental

El sistema diseñado e implementado se ha validado experimentalmente en un entorno controlado que simula las condiciones de operación reales. Las pruebas se llevaron a cabo en un espacio delimitado, con iluminación variable y presencia de obstáculos, a fin de evaluar la robustez del sistema de detección y la capacidad de respuesta del control distribuido.

Como se muestra en la Figura 8, durante las sesiones de prueba, el modelo YOLOv8 demostró un rendimiento consistente en la detección de pelotas de pádel, manteniendo una tasa media de inferencia de 6 fps tras su optimización con el framework ncnn. Esta frecuencia resultó suficiente para que el sistema de control ejecutara decisiones en tiempo real, permitiendo al robot detectar, acercarse y recoger pelotas de manera autónoma.



Figura 8: Fase de recogida automática del objeto mediante el sistema de succión, coordinado por el controlador autónomo tras la detección de la pelota.

Se realizaron múltiples ciclos de operación, alternando los distintos modos definidos en la máquina de estados. En el *modo recogida*, el sistema logró captar exitosamente más del 90 % de las pelotas presentes en el entorno, incluso en situaciones con oclusión parcial o cambios de iluminación. En el *modo lanzamiento*, el control remoto a través de la aplicación permitió modificar con precisión el ángulo y la dirección del disparo, validando la respuesta efectiva del sistema de orientación y del carrusel de alimentación.

La comunicación entre la Raspberry Pi y el ESP32 se mantuvo estable durante todas las pruebas, sin pérdidas significativas de datos, y la lógica de activación condicional del canal serie contribuyó a la eficiencia general del sistema.

En conjunto, los resultados obtenidos confirman la viabilidad del enfoque propuesto para aplicaciones autónomas de recogida y lanzamiento de objetos, con un diseño modular y escalable que permite futuras mejoras tanto en los algoritmos de percepción como en las estrategias de control.

7. Conclusiones

Este trabajo ha demostrado la viabilidad de un sistema robótico autónomo de recogida y lanzamiento de pelotas de pádel, combinando visión artificial, control distribuido y hardware de bajo coste. La adopción del modelo YOLOv8 y su conversión al framework ncnn ha permitido una detección precisa en tiempo real, con un balance adecuado entre sensibilidad (85 %) y precisión (100 %) en la clase *fondo*.

La separación entre inferencia (en Raspberry Pi 5) y control (en ESP32) garantiza un rendimiento fluido del sistema, mientras que la arquitectura basada en una máquina de estados y ruedas Mecanum proporciona versatilidad y eficiencia en la operación.

8. Mejoras futuras

Este desarrollo sienta las bases para futuras mejoras funcionales como la mejora del sistema de localización mediante

tecnologías como LiDAR, la integración de funcionalidades de entrenamiento personalizado mediante aplicación móvil, y el incremento de la capacidad de cómputo para habilitar nuevas funcionalidades avanzadas como la predicción del movimiento de la pelota o el aprendizaje adaptativo.

Agradecimientos

Los autores agradecen el apoyo recibido para la realización de este trabajo. Este proyecto ha recibido financiación por PID2022-142069OB-I00/AEI/10.13039/501100011033/FEDER, UE y por la Junta de Andalucía (Consejería de Transformación Económica, Industria, Conocimiento y Universidades) mediante la ayuda de investigación al Laboratorio de Ingeniería para la Sostenibilidad Energética y Medioambiental “ENGREEN”, referencia QUAL21 006 USE.

Referencias

- Deepa, R., Tamilselvan, E., Abrar, E., Sampath, S., April 2019. Comparison of yolo, ssd, faster rcnn for real time tennis ball tracking for action decision networks. In: 2019 International Conference on Advances in Computing and Communication Engineering (ICACCE). pp. 1–4.
DOI: [10.1109/ICACCE46606.2019.9079965](https://doi.org/10.1109/ICACCE46606.2019.9079965)
- Feng, X., Jiang, Y., Yang, X., Du, M., Li, X., 2019. Computer vision algorithms and hardware implementations: A survey. Integration 69, 309–320.
URL: <https://www.sciencedirect.com/science/article/pii/S0167926019301762>
DOI: <https://doi.org/10.1016/j.vlsi.2019.07.005>
- H.A.R. Eletrabi, X. W., 4 2024. Autonomous tennis assistant systems.
URL: <https://patents.google.com/patent/US11944876B2/en>
- Jocher, G., Chaurasia, A., Qiu, J., 2023. Ultralytics yolov8.
URL: <https://github.com/ultralytics/ultralytics>
- Marin, 2025. Rapid rise of pickleball and padel: Opportunities in the booming u.s. racquet sports market. <https://spainuschamber.com/en/rapid-rise-of-pickleball-and-padel-opportunities-in-the-booming-u-s-racquet-sports-market/>.
- Nathan, 2025. Tennis ball detection dataset. <https://universe.roboflow.com/nathan-4tlqa/tennis-ball-detection-vksde>, accessed: 2025-05-31.
- Ni, H., ncnn contributors, T., 2017. ncnn: High-performance neural network inference framework.
URL: <https://github.com/Tencent/ncnn>
- Peng, T., Qian, J., Zi, B., Liu, J., Wang, X., 2016. Mechanical design and control system of an omni-directional mobile robot for material conveying. Procedia CIRP 56, 412–415, the 9th International Conference on Digital Enterprise Technology – Intelligent Manufacturing in the Knowledge Economy Era.
URL: <https://www.sciencedirect.com/science/article/pii/S2212827116310605>
DOI: <https://doi.org/10.1016/j.procir.2016.10.068>
- Pereira, N., Ribeiro, F., Lopes, G., Whitney, D., Lino, J., Jan 2012. Autonomous golf ball picking robot design and development. Industrial Robot: An International Journal 39 (6), 541–550.
DOI: <https://doi.org/10.1108/01439911211268660>
- Qian, J., Zi, B., Wang, D., Ma, Y., Zhang, D., 2017. The design and development of an omni-directional mobile robot oriented to an intelligent manufacturing system. Sensors 17 (9).
URL: <https://www.mdpi.com/1424-8220/17/9/2073>
DOI: [10.3390/s17092073](https://doi.org/10.3390/s17092073)
- Statista, 2024. Ranking de federaciones deportivas con más licencias federativas en España en 2023. <https://es.statista.com/estadisticas/1051888/ranking-de-federaciones-con-mas-deportistas-federados-espana/>.