Simposio CEA de Robótica, Bioingeniería, Visión Artificial y Automática Marina 2025

RBVM
Simposio CEA de Robótica, Bioingeniería, Visión por Computador y Automática Marina

Sistema robotizado de clasificación con Deep Learning y visión artificial

Kalmutskyy, G.[a], Otálora, P.[b,*], Castilla, M.[b], Moreno, J.C.[b]

[a] *University of Almería, Department of Informatics, CIESOL, ceia3, Ctra. Sacramento s/n, 04120, Almería, Spain. kurtco1996@gmail.com*
[b] *University of Almería, Department of Informatics, CIESOL, ceia3, Ctra. Sacramento s/n, 04120, Almería, Spain. {p.otalora, m.castilla, jcmoreno}@ual.es*

Resumen

En el presente trabajo se desarrolla un sistema de clasificación robotizado de propósito general, integrando técnicas de aprendizaje profundo y visión artificial en un entorno de simulación industrial. Se emplea el simulador *CoppeliaSim* para recrear una célula robotizada de almacenamiento y manejo de objetos, donde un brazo robot industrial equipado con cámara identifica piezas en un estante y localiza posiciones de depósito sobre un robot móvil. Se diseñan redes neuronales convolucionales para resolver dos tareas: (1) la clasificación de objetos en imágenes del almacén, y (2) la localización de un objeto en una zona de entrega. Mediante un conjunto de datos sintéticos, las redes se entrenan y validan alcanzando una precisión del 99,99 % en la identificación de objetos y un error medio inferior al 5 % en la localización. Los resultados demuestran la viabilidad de aplicar estas soluciones de visión artificial basadas en *Deep Learning* en entornos industriales modernos, aportando alta fiabilidad en tareas de inventario y manipulación autónoma.

Palabras clave: Robótica industrial, Visión por computador, Aprendizaje profundo, Simulación

Robotic sorting system with Deep Learning and artificial vision

Abstract

The present work develops a general-purpose robotic classification system by integrating deep learning and computer vision techniques within an industrial simulation environment. The *CoppeliaSim* simulator is used to recreate a robotic cell for object storage and handling, where an industrial robotic arm equipped with a camera identifies items on a shelf and locates drop-off positions on a mobile robot. Convolutional neural networks are designed to solve two tasks: (1) object classification in warehouse images, and (2) object localization in a delivery zone. Using a synthetic dataset, the networks are trained and validated, achieving 99.99 % accuracy in object identification and an average localization error of less than 5 %. The results demonstrate the feasibility of applying these *Deep Learning*-based computer vision solutions in modern industrial environments, offering high reliability for inventory and autonomous handling tasks.

Keywords: Industrial robotics, Computer vision, Deep Learning, Simulation

1. Introducción

En la actualidad nos encontramos inmersos en la llamada cuarta revolución industrial (Industria 4.0), caracterizada por la introducción masiva de inteligencia artificial en los procesos productivos. La automatización industrial demanda sistemas flexibles capaces de percibir y tomar decisiones – por ejemplo, identificar y manipular distintos productos en líneas de producción o almacenes automatizados (Ghobakhloo, 2020). En este contexto, la visión artificial y el aprendizaje profundo se han convertido en herramientas clave para dotar a los robots de capacidades de reconocimiento de objetos y entendimiento del entorno (Villalba-Diez et al., 2019). Un sistema robotizado que pueda clasificar objetos de forma genérica tendría un alto valor en aplicaciones de logística y manufactura.

La motivación de este trabajo es precisamente explorar el diseño de un sistema general de clasificación robotizado apoyado en algoritmos de *Deep Learning*, evaluando su desempeño y potencial aplicabilidad industrial. Se busca estudiar y diseñar redes neuronales artificiales que puedan implementarse en una célula robotizada, para mejorar la autonomía y versatilidad de los procesos de manipulación. Para ello, se ha optado por traba-

*Autor para correspondencia: p.otalora@ual.es

jar en un entorno simulado que reproduce una célula industrial real, lo que permite generar datos de entrenamiento de forma controlada y probar los modelos de visión sin incurrir en riesgos ni costos físicos.

Los objetivos concretos del proyecto son: (1) desarrollar un entorno de simulación de una célula robotizada con cámara y robot manipulador que realice tareas de almacén; (2) diseñar e implementar dos modelos de red neuronal profunda, uno para clasificar objetos en imágenes y otro para predecir la ubicación de un objeto; (3) entrenar y validar dichas redes usando datos sintéticos obtenidos de la simulación; y (4) evaluar el rendimiento obtenido y discutir la viabilidad de trasladar la solución a un sistema real en la industria. A continuación se describe la configuración del sistema simulado, la metodología de diseño y aprendizaje de las redes neuronales, los resultados de validación obtenidos, y finalmente se discute su aplicabilidad en entornos industriales, seguido de conclusiones.

2. Célula robotizada

El sistema se ha implementado íntegramente en el simulador *CoppeliaSim* (antes conocido como *V-REP*) (Rohmer et al., 2013), reproduciendo una célula robotizada de almacenamiento automático con entrada y salida de objetos. En la Figura 1 se muestra una vista general de la célula en la simulación, incluyendo el robot manipulador, el estante de objetos y la zona de entrega sobre el robot móvil.

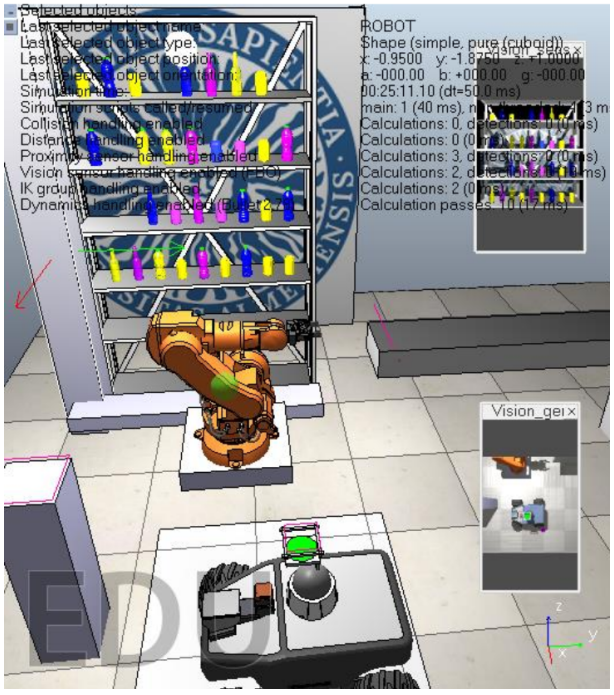

Figura 1: Entorno de simulación en *CoppeliaSim*: brazo *ABB IRB140* con cámara frente al estante de objetos (centro), base de depósito en robot móvil Summit (abajo), y cinta transportadora para entrada de objetos (derecha).

La célula se encuentra constituida por una serie de componentes. En primer lugar, dispone de una estantería industrial multi-nivel donde se colocan los objetos a clasificar. Fue seleccionada de la biblioteca de modelos de *CoppeliaSim*. Consta de múltiples baldas con posiciones definidas para hasta 36 objetos en total (incluyendo espacios vacíos). Una cámara fija se enfoca frontalmente a esta estantería para capturar imágenes de todos los objetos visibles. Detrás del estante se ubicó una pared con el logo de la empresa (Universidad de Almería) para proporcionar variabilidad de fondo y ayudar a la red a diferenciar los objetos del entorno.

Como manipulador, se emplea un robot industrial de 6 grados de libertad modelo *ABB IRB140* montado sobre una base fija dentro de la célula. El brazo está equipado con una pinza de dos dedos *ROBOTIQ 85*, adecuada para agarrar objetos de tamaño pequeño-medio. Este manipulador realiza las acciones de tomar objetos de la cinta o estantería y depositarlos en la ubicación deseada. El alcance del brazo define una zona de aparcamiento frente al estante donde puede colocar o recoger objetos sin colisionar con el entorno. El conjunto base-brazo-elemento terminal se puede observar en la Figura 2.

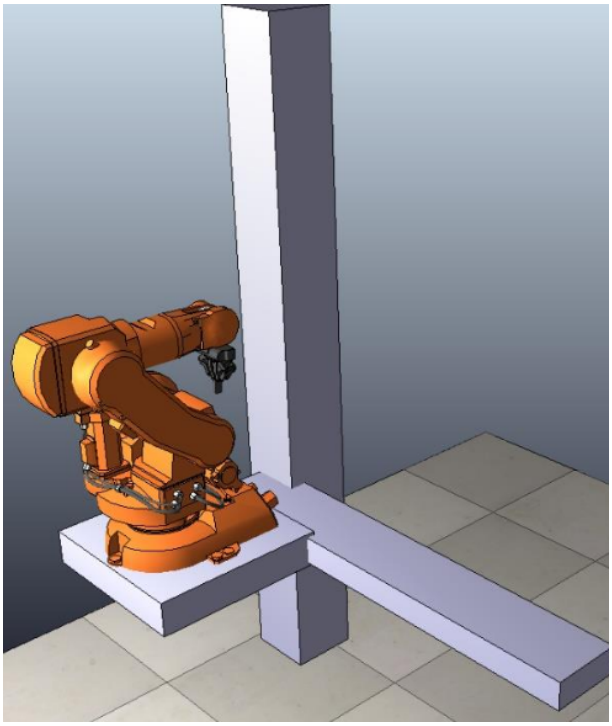

Figura 2: Estructura con el brazo robot y el elemento terminal.

Para simular la salida de objetos del almacén, se incluyó un robot móvil modelo *Summit XL* de *Robotnik*, posicionado en una zona de parada junto al brazo. Sobre el *Summit* se diseñó una base de depósito: una pequeña plataforma acoplada en la parte superior del robot móvil, diseñada con formas cuboidales simples, donde el brazo puede dejar los objetos recogidos. Esta base tiene una marca visual (un panel de color verde) que sirve como referencia visual para localizar su posición. Un sensor de área en la base detecta cuando un objeto ha sido depositado, comunicando al sistema que el objeto está listo para ser retirado, simulando la salida del sistema. Este conjunto se puede ver en la Figura 3 La ubicación y orientación del *Summit* pueden variarse en la simulación para entrenar la red de localización en distintos escenarios.

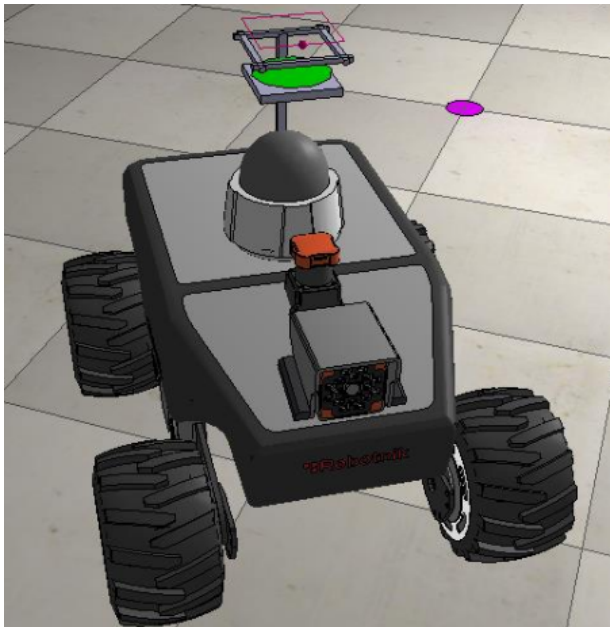

Figura 3: Robot *Summit* con plataforma superior.

Como mecanismo de entrada de nuevos objetos al sistema, se añadió una cinta transportadora virtual modelo *conveyor belt* eficiente de la librería de *CoppeliaSim*. La cinta genera objetos duplicando modelos predefinidos en uno de sus extremos y trasladándolos hasta el alcance del brazo robot. Un sensor lineal al final de la cinta detecta la presencia de un objeto; cuando un objeto llega, la cinta se detiene para que el robot pueda tomarlo. Este ciclo permite alimentar continuamente el almacén con nuevos artículos a clasificar.

Se utilizaron dos cámaras virtuales para proporcionar la información visual a las redes neuronales. La primera cámara está fija enfocando la estantería completa, capturando imágenes de resolución 128×128 píxeles de los objetos dispuestos en las baldas. La resolución se mantuvo baja para reducir la carga de procesamiento y simular imágenes sencillas para la red de clasificación. La segunda cámara se ubica enfocando la zona de aparcamiento del *Summit*, aportando una vista superior de la base de depósito. Esta cámara también genera imágenes de 128×128 de la ubicación donde debe depositarse el objeto. Para añadir realismo, la posición de esta cámara varía ligeramente en cada ejecución, emulando posibles imprecisiones o vibraciones en un sistema real. Además de las cámaras, se integraron sensores de proximidad y de área como el mencionado en la base del *Summit*, que permiten automatizar el flujo: el sensor de la cinta indica cuándo el robot debe recoger un objeto, y el sensor de la base del *Summit* confirma la entrega del objeto para su retirada.

En conjunto, esta célula robotizada simula un almacén automatizado donde el brazo robot puede clasificar objetos entrantes, almacenándolos en la estantería, y extraer objetos para colocarlos en el robot móvil de salida cuando sea necesario. Toda la lógica de control, incluyendo movimientos del robot, sincronización con sensores y generación de datos de cámara se implementó mediante *scripts* en *Python* utilizando la API remota de *CoppeliaSim*. Esto permitió automatizar la toma de miles de imágenes y la ejecución de las redes neuronales en bucle con la simulación, reproduciendo así un ciclo operativo autónomo.

3. Redes neuronales convolucionales

Para dotar al sistema descrito de inteligencia en la clasificación, se han desarrollado dos modelos de red neuronal convolucional (CNN) (O'Shea and Nash, 2015) entrenados específicamente para las tareas de visión artificial planteadas: identificación del tipo de objeto en el estante, y determinación de la posición de la base del *Summit*. Ambos modelos fueron diseñados y entrenados con la biblioteca *TensorFlow*/*Keras* en *Python* (Abadi et al., 2015), aprovechando los datos generados en la simulación. A continuación se detalla el proceso seguido para cada red.

3.1. Red neuronal de clasificación de objetos

Para la tarea de identificación se definió un problema de clasificación multiclase. Cada objeto que puede aparecer en la estantería pertenece a una de N clases predefinidas (distintos tipos de piezas, contenedores, etc., incluyendo la clase "hueco vacío"). En nuestro caso se trabajó con 10 clases, correspondientes a 9 tipos de objetos distintos más la clase vacío. El objetivo de la red es, dada la imagen de un objeto (o hueco) tomada por la cámara del estante, predecir correctamente a qué clase corresponde.

Mediante la simulación se generó un extenso *dataset* de imágenes etiquetadas. Para obtenerlas, se tomaron 280 imágenes distintas del estante completo, variando aleatoriamente la configuración de objetos en los 36 huecos (presencia/ausencia de cada objeto) así como las condiciones de iluminación general. A partir de cada imagen completa, se segmentaron automáticamente las 36 sub-imágenes correspondientes a cada posición de la estantería, asignando a cada una la etiqueta de la clase de objeto presente en esa posición (o vacío). De esta forma se recopiló un total de 10107 muestras individuales (8164 para entrenamiento y 1943 para validación), asegurando múltiples ejemplos por cada clase en diversas condiciones, con cambios en la iluminación, fondo variable gracias a la pared con logo o distintos objetos vecinos alrededor. Este enfoque de generar datos sintéticos permitió entrenar el modelo sin necesidad de imágenes reales, cubriendo una amplia variabilidad de escenarios de forma rápida.

Respecto a la estructura del modelo, se optó por una arquitectura CNN sencilla, adecuada al tamaño reducido de las imágenes (128×128 bajadas a 30×15 para cada objeto recortado, mediante preprocesamiento). La red de clasificación cuenta con dos capas convolucionales principales: la primera con 32 filtros de tamaño 3×3 y la segunda con 64 filtros de 2×2, cada una seguida de una capa de *pooling* 2×2 (Gholamalinezhad and Khosravi, 2020). Estas capas extraen características visuales relevantes de la imagen de entrada. A continuación, las características son aplanadas y pasadas a una parte densamente conectada: una capa totalmente conectada de 256 neuronas con activación *ReLU* (Dubey and Jain, 2019), seguida de la capa de salida de 10 neuronas *Softmax* (Gold et al., 1996), una por cada clase. La configuración se resume en la Tabla 1. Se utilizó función de pérdida de entropía cruzada categórica y el optimizador *Adam* (Kingma and Ba, 2017), con un *learning rate* inicial de 5×10^{-4} para el entrenamiento. No se recurrió a arquitecturas excesivamente profundas dado que el problema tiene complejidad moderada, priorizando un modelo ligero que pueda ejecutarse en tiempo real en el lazo de control.

Antes del entrenamiento, se aplicaron técnicas de aumento de datos sobre las imágenes (leve variación de escala, pequeñas rotaciones, espejado horizontal) para mejorar la robustez del modelo. El entrenamiento se llevó a cabo durante 30 épocas sobre los 8164 recortes de imagen de entrenamiento, con *batch size* de 10. Tras cada época se evaluaba la exactitud en el conjunto de validación (1943 muestras). Al finalizar, la red alcanzó más del 99 % de precisión tanto en entrenamiento como en validación, sin indicios de sobreajuste pronunciado. La mejor versión del modelo basada en la métrica de validación se guardó para las pruebas en la simulación.

Tabla 1: Arquitectura de la CNN de clasificación

Capa	Dimensiones	Parámetros	Activación
Entrada	30x15x3	-	-
Conv2D	30×15×32	896	ReLU
MaxPooling2D	15×7×32	0	–
Conv2D	15×7×64	8256	ReLU
MaxPooling2D	7×3×64	0	–
Flatten	1344	0	–
Dense	256	344,320	ReLU
Dropout (20 %)	256	0	–
Dense (salida)	10	2570	Softmax

3.2. Red neuronal de localización de objetos

La segunda red neuronal aborda un problema de regresión: dado un imagen de la cámara superior que muestra la base verde del *Summit* (zona donde se depositará el objeto), la red debe predecir las coordenadas de posición donde se encuentra dicha base u objeto objetivo dentro de la imagen. En esencia, se busca que el robot "vea"la plataforma del *Summit* y determine su posición relativa para colocar con precisión el objeto. Este es un caso de localización visual en 2D.

Para entrenar este modelo, se generaron 8997 muestras simuladas que relacionan imagen con posición. En cada iteración de simulación, se posicionó el robot móvil *Summit* de forma aleatoria dentro de una zona predefinida de aparcamiento, variando ligeramente su orientación y desplazamiento, y se capturó la imagen desde la cámara superior. Junto a cada imagen se almacenó la posición real en coordenadas del mundo simulado del centro de la base verde del *Summit*. Para aumentar la diversidad, también se alteraron aleatoriamente las condiciones de iluminación global en cada escena. De esta manera, la red aprende a inferir la ubicación de la base a partir de distintas perspectivas y luces, generalizando más allá de un caso fijo. Las 8997 muestras se dividieron en 5000 para entrenamiento, 2000 para validación durante la etapa de ajuste de hiperparámetros, reservando las 1997 restantes para pruebas finales.

A diferencia del clasificador, aquí la salida es continua, siendo esta las coordenadas X e Y relativas. La red de localización se diseñó inicialmente similar a la de clasificación en sus primeras capas, pero con una sección densa más profunda para lograr mayor precisión en la regresión. En concreto, la arquitectura (resumida en la Tabla 2) cuenta con dos capas convolucionales (32 filtros 3×3 y 64 filtros 3×3, cada una seguida de *MaxPooling* 2×2) que reducen la imagen de entrada de 128×128×3 a un mapa de características de 32×32×64. Luego, tras aplanar (*Flatten*) estas características, se emplea una serie de capas densas plenamente conectadas, todas con activación *ReLU*. Finalmente, la capa de salida tiene 2 neuronas lineales sin función de activación que producen las coordenadas (X, Y) predichas. En total, este modelo tiene del orden de 8.5 millones de parámetros, dominados por las conexiones de la parte densa debido al *flatten* de tamaño grande. Se utilizó la función de pérdida de error cuadrático medio (MSE) para ajustar la salida continua a las posiciones objetivo, y se empleó nuevamente el optimizador *Adam* con un *learning rate* inicial de $1 \times 10^{-}5$, ajustado más bajo que en clasificación para favorecer convergencia suave. Durante los experimentos se probaron variantes del modelo, realizando ajustes en el número de neuronas, capas o técnicas de regularización, buscando minimizar el error de localización.

La red se entrenó con *batch size* de 32 durante un número variable de épocas ajustando hiperparámetros según la métrica de validación. Se monitorizaron dos métricas: el error absoluto medio en píxeles y el MSE (función de coste). En las gráficas de evolución se observó una disminución estable de ambos errores hasta estabilizarse, tras lo cual se detenía el entrenamiento para evitar sobreajuste. Mediante la validación se refinaron parámetros como la tasa de aprendizaje y se introdujo *dropout* del 50 % en alguna capa intermedia para mejorar la generalización. La versión final elegida de la red fue la que arrojó menor error relativo en validación.

Tabla 2: Arquitectura de la CNN de localización

Capa	Dimensiones	Parámetros	Activación
Entrada	128x128x3	-	-
Conv2D	128x128×32	896	ReLU
MaxPooling2D	64x64×32	0	–
Conv2D	64x64x64x	8256	ReLU
MaxPooling2D	32x32x64	0	–
Flatten	65536	0	–
Dense	128	8,388,736	ReLU
Dense	256	33,025	ReLU
Dense	512	131,584	ReLU
Dense	256	131,328	ReLU
Dense	128	32,896	ReLU
Dense (salida)	2	258	Lineal

Al finalizar el entrenamiento de ambos modelos, se integraron las redes en el bucle de simulación para realizar pruebas de validación en la escena virtual: el brazo robot ejecuta secuencias de clasificación realimentándose de las predicciones de las redes (por ejemplo, identificar un tipo de objeto en cierto estante y luego posicionar ese objeto en la base del *Summit* utilizando la coordenada predicha). De esta forma, se pudo evaluar el desempeño de las redes de forma dinámica y no solo sobre *dataset* estático.

4. Resultados de validación

Se realizaron numerosas pruebas de validación para cuantificar el rendimiento de las redes en las tareas previstas. A continuación se resumen los resultados más destacados, separando la parte de clasificación y la de localización.

El modelo de CNN logró una precisión muy alta en la identificación de objetos. En las pruebas finales, el porcentaje de

acierto fue superior al 99.9 %. En términos de error, la tasa de clasificación errónea fue de solo 0.0132 %, lo cual implica prácticamente cero confusiones entre las 10 clases. Este excelente resultado se debe en parte a las condiciones controladas de las imágenes y al hecho de que los objetos a distinguir tenían diferencias visuales claras. La red fue capaz de aprender estas diferencias con los datos generados, abarcando variaciones de iluminación y posiciones aleatorias de cada objeto. Cabe destacar que incluso la clase "hueco vacío"fue reconocida sin problemas, permitiendo detectar la ausencia de objeto en una posición dada. En escenarios de prueba donde se simuló un recorrido completo de almacenaje, el sistema identificó correctamente todos los objetos entrantes antes de acomodarlos en la estantería. Esto demuestra la fiabilidad de la red de clasificación, esencial para que en un entorno industrial no se produzcan errores de inventario.

Para la tarea de localización de la base del robot móvil *Summit*, la red obtuvo también resultados satisfactorios. La métrica principal evaluada fue el error relativo de posición, definido como el error euclídeo en la predicción respecto al tamaño total de la zona de aparcamiento considerada. El modelo final alcanzó un error medio relativo de 4.93 %. Esto corresponde, en la escala física simulada, a un desvío típico del orden de pocos centímetros, dado que la zona de aparcamiento tenía aproximadamente unas decenas de centímetros de lado. Además, se verificó un criterio de precisión absoluta: en el 100 % de los casos de prueba, el error de predicción fue menor a un umbral de seguridad de 5cm, es decir, no se registró ningún caso de "fallo.[en] la colocación del objeto fuera del área prevista. En la práctica, esto significa que el brazo robot siempre logró dejar el objeto dentro de la plataforma del *Summit*. El tiempo de inferencia de la red es casi instantáneo, unos pocos milisegundos por imagen, por lo que el proceso de localización no introduce retardo perceptible en la operación del robot.

En resumen, el modelo seleccionado demuestra ser lo suficientemente preciso para el objetivo industrial: permite posicionar objetos con exactitud dentro del área designada sobre el robot móvil. Esta precisión se logró a costa de un modelo relativamente grande, pero aún así ejecutable en tiempo real. Un aspecto importante es que la red aprendió a generalizar distintas orientaciones del robot móvil gracias al entrenamiento con variadas posiciones: en pruebas, aunque el *Summit* se colocara en una orientación distinta a la habitual, la red pudo localizar correctamente la plataforma verde.

5. Discusión

Los resultados obtenidos avalan la viabilidad de aplicar este sistema de clasificación robotizado en un entorno industrial real, aunque conviene analizar algunas consideraciones prácticas. En primer lugar, la fiabilidad muy alta lograda en la identificación de objetos, cercana al 100 %, es un indicio positivo para la implementación en procesos industriales: un sistema de visión con ese nivel de certeza puede encargarse de verificar referencias de productos, detectar piezas correctas o equivocadas, y en general reducir errores humanos en la gestión de almacenes automatizados. En un almacén real, esto se traduciría en menores tasas de envíos erróneos o mezclas de producto, mejor control de stock y trazabilidad. Además, la capacidad de la red para distinguir también cuándo un hueco está vacío le permitiría, por ejemplo, llevar un inventario automatizado de existencias en cada ubicación de la estantería.

En cuanto a la localización con visión, haber garantizado un margen de error por debajo de 5 cm en todas las pruebas significa que el sistema es seguro y efectivo para guiar movimientos de precisión del robot. En la práctica, un brazo robot industrial suele tener repetibilidad del orden de 0.1–0.5cm, por lo que un error de unos pocos centímetros en visión sería inaceptable sin calibración adicional. Nuestro sistema, sin embargo, mantuvo errores dentro de un rango tolerable: ningún objeto quedó fuera de la plataforma de destino. Esto sugiere que la integración de la red de localización con el control del robot puede cumplir requerimientos industriales de precisión para tareas de *pick-and-place*. No obstante, para una aplicación real se debería realizar una calibración extrínseca cuidadosa entre la cámara y el mundo real, y posiblemente una transformación de las coordenadas predichas a coordenadas del robot mediante un sistema de visión calibrado. En este trabajo, al ser simulado, se asumió alineación conocida; en planta habría que añadir ese componente.

Un aspecto clave es el volumen de datos de entrenamiento requerido. Se necesitó generar del orden de 10^4 imágenes para entrenar cada red con buen rendimiento. En un proyecto industrial, obtener tal cantidad de datos reales etiquetados puede ser costoso. Aquí hemos demostrado que la simulación puede ser una aliada poderosa: utilizando *CoppeliaSim* se generaron datos sintéticos con variaciones de iluminación y posiciones, ahorrando tener que capturar miles de fotos reales. Esta estrategia de simulación para entrenamiento (*sim-to-real*) es cada vez más usada en robótica, aunque luego suele requerir técnicas de adaptación al dominio real. En nuestro caso, los modelos entrenados podrían beneficiarse de un ajuste fino (*fine-tuning*) posterior con algunas imágenes reales del sistema final para compensar las diferencias entre simulación y realidad, por ejemplo, distinto ruido en la cámara, diferencias en color de objetos, iluminación no ideal, etc.

También merece mención la flexibilidad del sistema: está concebido como propósito general porque las redes pueden reentrenarse para distintas familias de objetos o escenarios. Por ejemplo, se podrían introducir nuevos tipos de piezas en la estantería simplemente incorporando sus imágenes al *dataset* sintético y volviendo a entrenar la CNN de clasificación. De igual modo, la red de localización podría recalibrarse a otra zona o incluso a un sensor distinto, como coordenadas 3D de una cámara RGB-D, con los datos apropiados. En un entorno industrial cambiante, esta capacidad de reajuste es valiosa.

Desde el punto de vista de integración, la implementación en *Python* con *TensorFlow* demuestra que es posible ejecutar inferencias en tiempo real dentro del ciclo de control robótico. En las pruebas, ambas redes corrían en una fracción de segundo sobre hardware de PC convencional (CPU Ryzen 5 y GPU GTX 1660) – un rendimiento que en un sistema comercial podría mejorarse usando dispositivos dedicados, como una GPU industrial o un acelerador tipo TPU. Esto significa que añadir visión inteligente no compromete la velocidad de ciclo de la célula robotizada, que pudo mantener ritmos de varios *picks* por minuto sin espera por procesamiento.

En contraste con un enfoque puramente tradicional, basado en sensores determinísticos y programación explícita, el uso de

aprendizaje profundo aportó robustez ante variabilidad: las redes aprendieron de situaciones variadas, y por tanto el sistema es menos sensible a desviaciones de las condiciones nominales. Sin embargo, una limitación inherente es que el sistema actuará dentro de lo aprendido; situaciones completamente nuevas, como un objeto de clase desconocida, o una obstrucción inesperada en la cámara, podrían requerir re-entrenamiento o generar fallos. Por ello, en entornos críticos se podría combinar esta solución con técnicas tradicionales de verificación; sensores redundantes que detecten anomalías, parar el sistema si la red da una salida incoherente, etc.

Por otro lado, el simulador desarrollado puede cobrar gran importancia como herramienta de apoyo a la docencia, en las asignaturas de Robótica Industrial del Grado en Ingeniería Electrónica Industrial y Automática, y en la de Sistemas Robotizados del Máster en Ingeniería Informática de la Universidad de Almería, permitiendo a los alumnos trabajar de forma remota con un modelo del sistema de almacenamiento disponible en la misma universidad. Del mismo modo, el robot *ABB IRB140* y el *Summit*, así como la cinta transportadora y los sensores, forman también parte de la infraestructura de la universidad, pudiendo combinar los conceptos afianzados en simulación con su posterior aplicación en los sistemas reales.

En general, los resultados apoyan que un sistema robotizado con visión por computador basada en *Deep Learning* puede integrarse exitosamente en operaciones industriales de manipulación. Este proyecto, si bien realizado en simulación, sienta las bases para una implementación física: el siguiente paso lógico sería trasladar el modelo a una célula real, usando el mismo software de redes neuronales para guiarlo. Dado el desempeño mostrado en virtual, se esperaría una alta probabilidad de éxito en el mundo real, siempre que se realice la debida calibración y ajuste al entorno real.

6. Conclusiones

Se ha presentado un sistema de clasificación robotizado que combina robótica y *Deep Learning*, validado en un entorno de simulación con orientación industrial. El proyecto permitió diseñar dos redes neuronales convolucionales especializadas: una para la identificación de objetos en imágenes de un almacén automatizado, y otra para la localización de la posición de un objeto en el entorno. A través de la simulación en *CoppeliaSim* se generaron los datos necesarios y se integraron las redes en la célula virtual, logrando demostraciones exitosas de clasificación y manipulación autónoma.

Las redes desarrolladas alcanzaron rendimientos muy altos: 99.9 % de confianza en la clasificación de objetos y error nulo en la colocación de objetos dentro del área objetivo. Esto confirma que las técnicas de visión artificial basadas en aprendizaje profundo pueden dotar a un robot industrial de la capacidad de percibir e identificar su entorno con precisión y fiabilidad equiparables o superiores a sistemas tradicionales. La utilización de simulación resultó ser una herramienta eficaz para reducir el coste de desarrollo, permitiendo probar rápidamente distintas configuraciones y generar grandes volúmenes de datos sintéticos.

En conclusión, el trabajo demuestra la viabilidad de un sistema robotizado de propósito general guiado por visión inteligente en contextos industriales modernos. Un robot equipado con estas redes neuronales puede adaptarse a diferentes tareas de almacén (clasificación de productos, *pick-and-place*) sin reprogramación manual exhaustiva, simplemente entrenando con ejemplos. Ello redunda en mayor flexibilidad y eficiencia en la intralogística. Como trabajos futuros, se propone ampliar el alcance del sistema: por ejemplo, incrementar el número de clases de objetos identificables, extender la zona de localización más allá de la actual requiriendo imágenes de mayor resolución y redes más complejas, e investigar el uso de técnicas de preentrenamiento no supervisado (autoencoders) para mejorar aún más la fase de clasificación. Asimismo, la transición del modelo simulado al real será un paso crucial, en el cual se validará la robustez de las redes frente a las imperfecciones del mundo físico.

En síntesis, este proyecto refleja cómo la sinergia entre simulación robótica y *Deep Learning* puede acelerar el desarrollo de soluciones de visión artificial aplicadas a la robótica industrial, alcanzando sistemas inteligentes con potencial para integrarse en la fábrica del futuro.

Referencias

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.
URL: `https://www.tensorflow.org/`

Dubey, A. K., Jain, V., 2019. Comparative study of convolution neural network's relu and leaky-relu activation functions. In: Mishra, S., Sood, Y. R., Tomar, A. (Eds.), Applications of Computing, Automation and Wireless Systems in Electrical Engineering. Springer Singapore, Singapore, pp. 873–880.

Ghobakhloo, M., 2020. Industry 4.0, digitization, and opportunities for sustainability. Journal of Cleaner Production 252, 119869.
DOI: `https://doi.org/10.1016/j.jclepro.2019.119869`

Gholamalinezhad, H., Khosravi, H., 2020. Pooling methods in deep neural networks, a review.
URL: `https://arxiv.org/abs/2009.07485`

Gold, S., Rangarajan, A., et al., 1996. Softmax to softassign: Neural network algorithms for combinatorial optimization. Journal of Artificial Neural Networks 2 (4), 381–399.

Kingma, D. P., Ba, J., 2017. Adam: A method for stochastic optimization.
URL: `https://arxiv.org/abs/1412.6980`

O'Shea, K., Nash, R., 2015. An introduction to convolutional neural networks.
URL: `https://arxiv.org/abs/1511.08458`

Rohmer, E., Singh, S. P. N., Freese, M., 2013. V-rep: A versatile and scalable robot simulation framework. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 1321–1326.
DOI: `10.1109/IROS.2013.6696520`

Villalba-Diez, J., Schmidt, D., Gevers, R., Ordieres-Meré, J., Buchwitz, M., Wellbrock, W., 2019. Deep learning for industrial computer vision quality control in the printing industry 4.0. Sensors 19 (18).
DOI: `10.3390/s19183987`