

Localización Basada en Visión por Computadora para Robots Móviles en Almacenes

Ingrid Yesenia Reyes Ramos¹, Dr. David Luviano Cruz², M.I. Luz Angelica Garcia Villalba³,
M.I Diana Yaziel Ortiz Muñoz⁴

Resumen— Los robots móviles autónomos se encuentran cada vez más presentes en la logística, realizando funciones en almacenes como es el transporte de mercancías y preparación de pedidos. Sin embargo, la eficiencia y precisión de estas operaciones dependen de la localización de los robots dentro del almacén, un desafío notable en entornos interiores complejos. En este documento se presenta un sistema de localización para la mejora de la navegación en entornos complejos que aborda el problema de localización de robots autónomos en sus áreas, en estudios recientes se ha encontrado variaciones en la precisión generando tiempos muertos, insatisfacción del cliente y riesgos de seguridad. Proponiendo de esta manera el uso de una red neural CNN conjunto al sistema de navegación y control de un robot en un entorno simulado. Se comparan resultados de simulación computacional en Webots enlazado a MATLAB ofreciendo una respuesta de mejora en términos de reconocimiento y eficiencia.

Palabras clave— Localización, automatización, visión, evasión de obstáculos, red neuronal convolucional.

Introducción

La industria manufacturera está al borde de una era de transformación impulsada por la automatización robótica móvil, que garantiza una mayor eficiencia, calidad y adaptabilidad (Kivnon, 2023). El informe de robots industriales de World Robotics 2021 revela un récord de 3 millones de robots industriales que funcionan en fábricas de todo el mundo, lo que se traduce en un aumento del 10% respecto al año anterior. Las ventas de nuevos robots crecieron ligeramente, un 0,5%, a pesar de la pandemia, con un total de 384.000 unidades vendidas en todo el mundo en 2020 (Robotics, 2021). Uno de los principales beneficios que se consiguen es automatizar simples tareas, como el transporte interno creando una trayectoria eficiente para el robot, pero para ello se necesita localizar cada movimiento del robot (Borenstein, 2018).

El saber la localización exacta del robot dentro del entorno previene graves consecuencias en términos de seguridad, eficiencia, navegación, mantenimiento e integración con otros sistemas (Robots, 2019), evitando de esta manera tiempos muertos y costos elevados. Para solucionar estos problemas, se utilizan diversos sistemas de localización para robots móviles, como sistema de posicionamiento global, sensores inerciales, odometría, SLAM (Localización y Mapeo Simultáneos) (Amjad, 2023), sin embargo, aún existe la problemática de localización de un robot móvil debido que su posición no puede ser medida directamente y debe ser inferida de datos sensoriales (Thrun, 2023).

Este trabajo propone un sistema para la optimización de navegación autónoma de un robot en un entorno simulado mediante visión por computadora y algoritmos de inteligencia artificial, sin depender de los sensores tradicionales como lo es el GPS. Utilizando MATLAB como controlador del robot y Webots como simulador del entorno donde el robot necesita esquivar obstáculos y reconocer obstáculos por medio de la red neuronal implementada, la cual es un preprocesamiento de imágenes para mejorar la visibilidad de las características clave.

Así mismo, se aborda aspectos clave, incluido el control de la navegación autónoma, procesamiento de imágenes, red neuronal para el reconocimiento de objetos, un mapeo del entorno en tiempo real e integración de sensores y visión artificial.

Se espera que este sistema mejore la eficiencia y productividad de los robots de entrega en almacenes, reduciendo costos operativos y abriendo nuevas oportunidades para la automatización en la industria logística. Por ende, no solo aborda una necesidad práctica en el campo de la logística, sino que también promueve el avance tecnológico y la optimización de procesos en almacenes, lo que lo convierte en un tema relevante y significativo para la investigación en este entorno.

¹ Ingrid Yesenia Reyes Ramos es Estudiante de la Ingeniera en mecatrónica en la Universidad Autónoma de Ciudad Juárez. al193850@alumnos.uacj.mx. (autor corresponsal)

² Dr. David Luviano Cruz es docente de Ingeniera en la Universidad Autónoma de Ciudad Juárez. david.luviano@uacj.mx.

³ M.I Luz Angelica Garcia Villalba es docente de Ingeniera en la Universidad Autónoma de Ciudad Juárez. lugarcia@uacj.mx.

⁴ M.I Diana Yaziel Ortiz Muñoz es docente de Ingeniera en la Universidad Autónoma de Ciudad Juárez. diana.ortiz@uacj.mx.

Metodología

Configuración del entorno

La configuración del entorno simulado en Webots es un proceso el cual comienza con el diseño y construcción de un modelo tridimensional del almacén dentro del simulador, el cual incluye la colocación estratégica de obstáculos relevantes para la operación del robot.

El objetivo principal de este entorno es proporcionar una representación desafiante que sirva como base para el análisis del desempeño del robot. Al reflejar fielmente las características del almacén, se asegura que las pruebas realizadas en el entorno simulado proporcionen datos relevantes y aplicables para mejorar el comportamiento del robot en el mundo físico.

Sistema de visión

Existen diferentes tipos de redes neuronales; en este caso, al ser un sistema de visión se optó por el uso de una CNN (siglas en inglés Convolutional Neuronal Network). Para implementar el sistema de visión, inicialmente se captura imágenes del entorno virtual. Este entorno se diseñó con obstáculos estratégicamente situados para la simulación en un escenario de navegación autónoma. Las imágenes capturadas deben almacenarse en una carpeta específica, como se puede observar en la figura 1, debido a que el código las extrae directamente. En este proyecto, la carpeta principal para el almacenamiento es "Imágenes". Dentro de esta carpeta se dividen en 2 secciones: sección 1 la cual contiene las imágenes de los obstáculos, mientras en la sección 2 almacena las imágenes de la pared que delimita el espacio del robot.

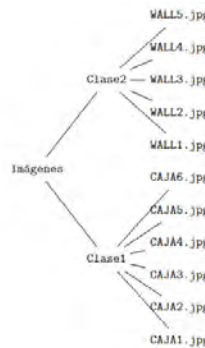
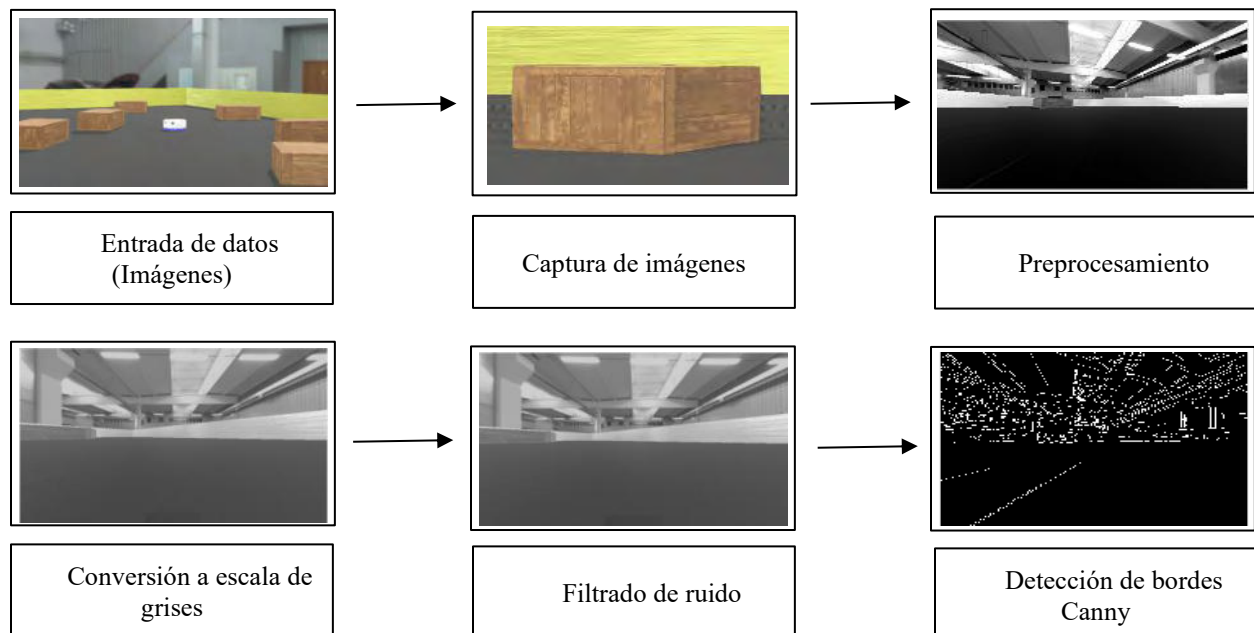


Figura 1. Diagrama de árbol sobre la clasificación de imágenes.

Entrada de datos

Se captura imágenes del entorno donde el robot realizará su navegación evadiendo obstáculos. El siguiente diagrama demuestra el proceso de preprocesamiento:



Arquitectura de red neuronal

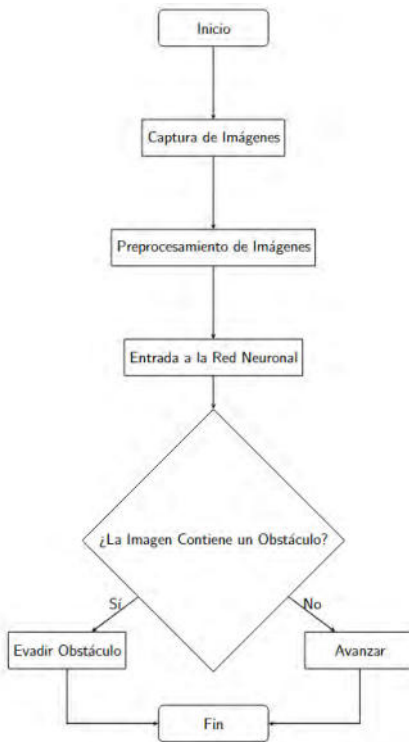


Figura 2. Diagrama de flujo de red neuronal en la cual toma decisiones de acuerdo con la configuración.

Captura de imágenes

Se utiliza un datatore de imágenes para el almacenamiento de las imágenes en 2 clases diferentes, que están clasificadas en carpetas (clase 1 y clase 2). Cada imagen se redimensiona en a tamaño estándar de 224x224 píxeles, garantizando la consistencia en las entradas a la red neuronal.

Preprocesamiento de imágenes

El objetivo principal del preprocesamiento es la mejora de calidad en las imágenes capturadas para facilitar la extracción de características relevantes:

- Conversión a escala de grises
- Filtrado de ruido, es la aplicación de filtros Gaussianos para suavizar las imágenes y eliminación de ruido mejorando la detección de características importantes.
- Detección de bordes, Canny edge detection permite resaltar los bordes de las imágenes, facilitando el reconocimiento de los objetos y sus contornos en las capas subsecuentes.

Entrada a la red neuronal

Las imágenes se dividen en dos subconjuntos: el 80% se usa para entrenar la red y el 20% para validar su desempeño y evitar sobreajuste. La red neuronal tiene varias capas: la de entrada acepta imágenes de 224x224 con tres canales RGB, cuatro capas convolucionales extraen características locales usando filtros crecientes (16, 32, 64 y 128), las capas de normalización por lotes estabilizan el entrenamiento, las capas ReLU introducen no linealidad, las de Max-Pooling reducen las dimensiones, una capa completamente conectada enlaza las características a dos salidas, y finalmente la capa Softmax asigna probabilidades a las clases.

Entrenamiento CNN

Se entrena la red utilizando el algoritmo de optimización SGDM (por sus siglas en ingles Stochastic Gradient Descent with Momentum); Así mismo se configura una tasa de aprendizaje inicial de 0.001, con un número máximo de 10 épocas.

Durante el entrenamiento, los datos se barajan en cada época para mejorar la generalización y se monitorea los progresos del entrenamiento a través de gráficos en tiempo real, mientras se valida el desempeño del modelo con el conjunto de validación.

El flujo procesa las imágenes capturadas, las prepara mediante preprocesamiento, las clasifica con una red convolucional, y ajusta los parámetros de la red para mejorar el rendimiento en la clasificación de imágenes de dos clases. Las opciones de entrenamiento utilizadas están enlistadas de la siguiente manera:

1. Algoritmo de optimización: Stochastic Gradient Descent (SGDM).
2. Tasa de aprendizaje inicial de 0.001.
3. 10 épocas.
4. Aleatorización en cada época.
5. Gráfica de progreso de entrenamiento.
6. Uso de los datos de validación.

Como se puede observar en la figura 2, empieza con el proceso de entrada de datos, al realizar el preprocesamiento de la información la red neuronal selecciona de las clasificaciones de obstáculos y lo detecta conforme navega en el entorno. Tomando decisiones de acuerdo con el obstáculo detectado conjunto a el umbral previamente programado de acción.

Mapeo de trayectoria por medio de Odometría

La odometría calcula la posición del robot a partir de las lecturas de los encoders en las ruedas. Primero, se obtienen las lecturas de los encoders para las ruedas izquierda y derecha:

$$left_wheel_position = wb_position_sensor_get_value(left_position_sensor)$$

$$right_wheel_position = wb_position_sensor_get_value(right_position_sensor)$$

Después, se calculan las distancias recorridas por cada rueda multiplicando estas lecturas por el radio de las ruedas r (por ejemplo, $r = 0.02$ m)

$$d_{left}t = left_wheel_position \times r$$

$$d_{right} = right_wheel_position \times r$$

$$d = \frac{2d_{left} + d_{right}}{2}$$

El cambio en el ángulo $\Delta\theta$ se calcula utilizando la diferencia de distancias de las ruedas y la longitud del eje L (por ejemplo, $L = 0.1$ m):

$$\Delta\theta = \frac{d_{right} - d_{left}}{L}$$

Para actualizar la orientación del robot, se suma el cambio de ángulo:

$$\theta_{new} = \theta_{old} + \Delta\theta$$

Finalmente, se actualiza la posición X_{new} y Y_{new} utilizando trigonometría:

$$y_{new} = y_{old} + d \times \sin(\theta_{new})$$

$$X_{new} = X_{old} + d \times \cos(\theta_{new})$$

Este conjunto de ecuaciones permite que el robot navegue de manera autónoma, calculando su posición y orientación en el entorno.

Resultados

El sistema de navegación autónoma desarrollado para el robot Khepera IV integra evasión de obstáculos, mapeo de trayectoria, procesamiento de imágenes en tiempo real y una red neuronal convolucional (CNN) para reconocimiento visual, permitiendo navegar en entornos complejos con mayor capacidad de decisión. Al aumentar la velocidad de desplazamiento y rotación, el robot mejora su eficiencia en el movimiento, manteniendo así la capacidad de evitar colisiones mediante una estrategia de retroceso y giro aleatorio, se puede observar en la figura 3, los sensores activos al momento de navegar por el entorno del robot, estos están controlados para detectar a un umbral de 100 mm el obstáculo para que el robot pueda tomar la decisión sin necesidad de topar con el obstáculo, como se puede observar en la figura número 4, el robot se encuentra con un obstáculo y este lo evade por medio de giro aleatorio.



Figura 3. Vista de dirección de sensores para evitar obstáculos.



Figura 4. Detección de obstáculo por medio de contorno.

La figura 5, muestra el progreso de entrenamiento de una red neuronal en MATLAB, destacando la precisión y la pérdida a lo largo de las épocas. En el panel superior se observa la precisión en el cual la curva azul representa la precisión del entrenamiento en cada iteración. Al final de la primera época, la precisión de entrenamiento sube rápidamente, alcanzando el 100% en la segunda época y manteniéndose constante. La línea de puntos negra muestra la precisión de validación, que va aumentando gradualmente con cada época, pero sin llegar a ser tan alta como la precisión de entrenamiento, lo que indica que el modelo aún podría beneficiarse de ajustes para mejorar la precisión en datos de validación. Mientras que en el panel inferior muestra la curva naranja que representa la pérdida en el conjunto de entrenamiento, que disminuye significativamente después de la primera época y se mantiene baja a lo largo del resto del entrenamiento.

- Se usó una CPU para entrenar el modelo, con una tasa de aprendizaje constante de 0.001.

El modelo muestra aceptación en los datos de entrenamiento, con una precisión de 100% desde la segunda época. Sin embargo, el aumento gradual en la precisión de validación y la diferencia en las curvas de pérdida entre el entrenamiento y la validación podrían indicar cierto sobreajuste, ya que el modelo parece aprender demasiado bien los datos de entrenamiento en comparación con su rendimiento en validación.

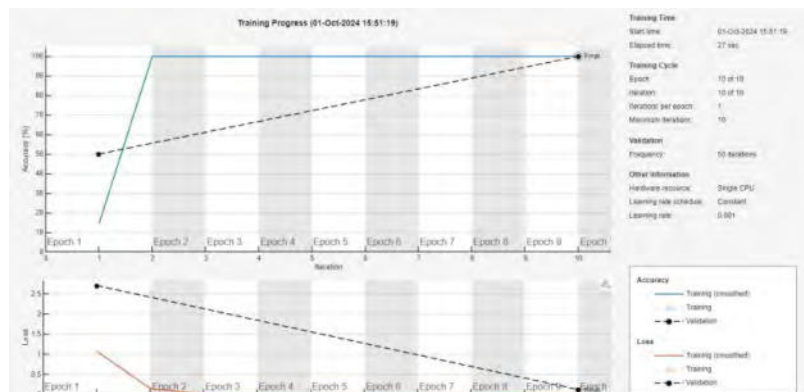


Figura 5. Resultado final de entrenamiento de red neuronal con 2 clasificaciones de imágenes.

Con los datos obtenidos de entrenamiento y control del robot se puede obtener la navegación que a tenido el robot en el entorno, esto es necesario ya que al tomar decisiones crea nuevas rutas para evadir obstáculos. Esto permite observar problemas a futuro de colisiones o detección de obstáculos.

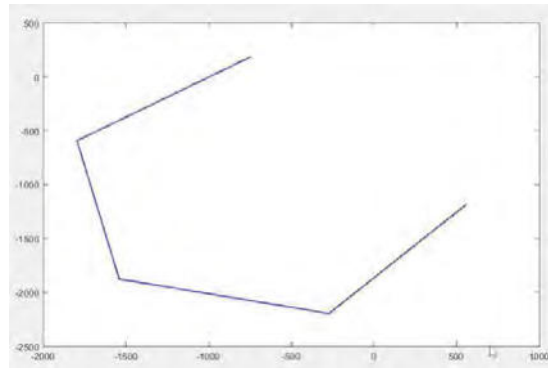


Figura 6. Mapeo de movimientos del robot trazado en color azul.

Como se observa en la figura 6, se traza el recorrido realizado por el robot con los giros aleatorios después de detectar los obstáculos del entorno, demostrando así un comportamiento lógico en su navegación. Así mismo se destaca que no hay pérdidas de mapeo en el recorrido del robot, comprobando la eficiencia en tiempo real de los cambios realizados durante su trayectoria.

Conclusiones

En conclusión, este proyecto representa una mejora significativa en la navegación de robots autónomos en entornos industriales, como son los almacenes, al integrar un sistema de navegación autónoma que combina evasión de obstáculos y procesamiento de imágenes en tiempo real. Por ende, proporciona al robot la capacidad de prevenir colisiones y accidentes, optimizando así el tiempo y minimizando tiempos muertos al encontrar obstáculos. Se resalta en este proyecto la importancia de combinar capacidades de navegación con visión por computadora, ya que permite que el robot no solo actúa ante obstáculos, sino que a la vez identifique elementos específicos de su entorno.

El proyecto enfrenta algunas delimitaciones, las cuales empiezan con la precisión de la odometría ya que puede disminuir en situaciones donde las ruedas pierdan tracción, afectando el mapeo. Otro factor limitante es que solo ha sido probado en una simulación para el entrenamiento y el ajuste fino del sistema de navegación, lo que podría haberse mejorado.

Los investigadores interesados en continuar esta línea de investigación podrían enfocarse en expandir la capacidad de clasificación de la red neuronal convolucional a un número mayor de clases de objetos relevantes en entornos industriales. Se recomienda investigar métodos de percepción robusta en tiempo real, para lograr que el robot no solo detecte y clasifique objetos de manera precisa, si no que tome directo de su entorno nuevos objetos y/o obstáculos sin necesidad de cargarlos previamente en su sistema, logrando adaptabilidad en su entorno.

Referencias

- Amjad, B. a. (2023). Radio SLAM: A Review on Radio-Based Simultaneous Localization and Mapping. *IEEE* , 9260-9278.
- Borenstein, J. a. (2018). A Survey of Methods for Mobile Robot Localization and Mapping in Dynamic Indoor Environments. *IEEE Transactions on Robotics and Automation*.
- kivnon. (2023). *Mobile Robots Revolutionizing the Manufacturing Industry*. Obtenido de <https://www.kivnon.com/es/mobile-robots-revolutionizing-the-manufacturing-industry/>
- Robotics, I. F. (November de 2021). *Service Robots Hit Double Digit Growth Worldwide*. Obtenido de <https://ifr.org/ifr-press-releases/news/service-robots-hit-double-digit-growth-worldwide>
- Robots, A. R. (2019). Inertial navigation systems for mobile robots. *IEEE Transactions on Robotics and Automation*.
- Thrun, S. a. (2023). simultaneous localization and mapping. *Artificial Intelligence*.